

HANSER



Vorwort

Mit der Erfindung der „Speicherprogrammierbare Steuerung (SPS)“ im Jahr 1968 wurde die dritte industrielle Revolution eingeläutet. Nun erleben wir im Verlauf der 4. industriellen Revolution, dass die SPS noch immer millionenfach in Produktionsbetrieben eingesetzt wird und in der Industrie 4.0 als Edge-Controller erheblich zu einer hocheffizienten, erfolgreichen Industrieproduktion beiträgt.

Das vorliegende *Lehrbuch* will den Lesern einen Leitfaden an die Hand geben, wie sie typische Aufgaben der Fabrik- und Prozessautomation mit speicherprogrammierbaren Steuerungen lösen können. Dabei wird sowohl der Systemaufbau als auch die Programmierung von speicherprogrammierbaren Steuerungen behandelt. Die Einbindung von SPSen in die digitale Fabrik mit Robotern und autonomen Systemen wird ebenso beschrieben wie Methoden der Industrie 4.0 zur Nutzung der von der SPS gesammelten Prozessdaten in der Cloud.

Das Buch versucht, den Stoff anwendungsorientiert zu vermitteln. Dabei wird nur am Rande auf die Programmiersysteme einzelner Hersteller und deren Programmiersyntax eingegangen, sondern im Mittelpunkt steht die *Entwurfsmethodik* für eine transparente und flexibel einsetzbare SPS-Software. Hierfür wird eine Systematik vorgestellt, die

- eine objektorientierte Softwarestrukturierung mit Hilfe von UML-Diagrammen vorschlägt,
- verschiedene Entwurfsverfahren aus der Informatik auf das SPS-Software-Engineering anwendet, und
- die Programmierung strukturiert oder objektorientiert vornimmt.

Diese neue *fünfte Auflage* betrachtet die Einbindung von SPSen in die Industrie 4.0. Dabei wird erläutert,

- wie SPSen für Cyber Physical Systems (CPS) entwickelt werden, die nicht individuell programmiert, sondern aus Modulen möglichst per Plug and Play zusammengesetzt werden können,
- wie die SPS Roboter und Kameras ansteuert und damit die Autonomie der CPS in der Smart Factory erhöht,
- wie die SPS-Software virtuell mit Hilfe digitaler Zwillinge projiziert und in Betrieb gesetzt wird,
- wie die Kommunikation zwischen den SPSen und der Cloud erfolgt,
- welche Sicherheitsmechanismen erforderlich sind,
- welche Cloud-Services zur integrierten Betriebsführung eingesetzt werden.

Außerdem wird für die 5. Auflage eine neue *SPS-Lern-und-Übungsseite* unter www.et-seitz.hs-mannheim.de bereitgestellt, auf der zahlreiche Beispiele, Übungen und

Wiederholungsfragen die Leser beim Erlernen der erläuterten Methoden und Werkzeuge unterstützen. Alle Beispiel- und Übungsprogramme sind systemneutral konzipiert, d. h. sie können prinzipiell in jedem Programmiersystem (Codesys, TIA-Portal o. a.) so wie im Text beschrieben umgesetzt werden. Da die Firma CODESYS ihr Programmiersystem zum kostenlosen Download zur Verfügung stellt, wurden die Beispiele und Übungsaufgaben damit erstellt. Sie stehen auf der SPS-Lern-und-Übungsseite zum Download und zur Simulation zur Verfügung ebenso wie Bibliotheken mit den im Buch besprochenen Funktionsbausteinen.

In diesem Zusammenhang bedanke ich mich bei den Firmen CODESYS, ABB, Siemens und Wonderware für die Bereitstellung von Software und Bildmaterial. Frau Natalia Silakova und Frau Christina Kubiak 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 Frau Prof. Dr. A. Weigl-Seitz sowie meine Kollegen von der Hochschule Mannheim Prof. Dr. K. Böhnke, Prof. Dr. M. Hauske, Prof. Dr. O. Wasenmüller, Prof. Dr. T. Weickert und ganz besonders Herr Dipl.-Ing. Hans Peter, Laborbetriebsleiter des Instituts für industrielle Automatisierungssysteme, mit dem mich eine enge Zusammenarbeit auf dem Gebiet der SPS-Technik verbindet.

Schließlich gilt mein *Dank* meinen Studierenden für ihre Mitarbeit in Vorlesung und Labor und den vielen Leserinnen und Lesern, die durch ihre Rückmeldungen zur Verbesserung der Darstellung und Korrektur von Fehlern beigetragen haben.

Mannheim, Juli 2021

Matthias Seitz

Inhaltsverzeichnis

Vorwort	5
Verzeichnis der Abkürzungen und Symbole	12
1 Einführung	16
1.1 Entwicklung der Automatisierungstechnik	16
1.2 Automatisierungssysteme	18
1.3 Aufgaben in der Industrie 4.0	19
1.3.1 Messen, Steuern, Regeln und Überwachen	20
1.3.2 Auswerten und Planen im Industrial Internet of Things	21
1.3.3 Virtualisierung und Testen mit digitalem Zwilling	23
Übungen zu Kapitel 1	23
2 Aufbau von Steuerungen für die Industrie 4.0	24
2.1 SPS-Aufbau	24
2.1.1 Zentralbaugruppe	24
2.1.2 Peripheriebaugruppen	24
2.1.3 Programmiergerät	25
2.1.4 Human Machine Interface	25
2.1.5 Vernetzte Automatisierungssysteme	26
2.2 SPS-Arten und IoT-Geräte	27
2.2.1 Hardware-SPS	27
2.2.2 Slot-SPS	27
2.2.3 Soft-SPS	27
2.2.4 Vor- und Nachteile PC-basierter SPSen	27
2.2.5 Edge-Controller und IoT-Gateways	28
2.2.6 Hochverfügbare und fehlersichere SPSen	29
2.3 Informationsverarbeitung in der SPS	29
2.4 Ein- und Ausgangsbaugruppen der SPS	30
2.4.1 Digitale Eingangsbaugruppen	30
2.4.2 Digitale Ausgangsbaugruppen	31
2.4.3 Analoge Eingangsbaugruppen	32
2.4.4 Analoge Ausgangsbaugruppen	34
2.4.5 Schnelle Zählerbaugruppen	34
2.4.6 Pulsausgabe-Baugruppen	35
2.5 Ankopplung der Sensoren und Aktoren an die SPS	36
2.5.1 Zwei-/Vierleitertechnik	36
2.5.2 Busankopplung der Feldgeräte	36
2.5.3 Intelligenter Feldverteiler (Remote-I/O)	37
2.6 Industrielle Feldbussysteme	38
2.6.1 Übertragungsmedien	38
2.6.2 Datenübertragung durch das Master-Slave-Verfahren	39
2.6.3 Ethernet-basierte Feldbusse	40

2.7	Bedienen und Beobachten	41
2.7.1	Elemente der Prozessvisualisierung	42
2.7.2	Datenaustausch zwischen HMI und SPS	45
2.7.3	Ankopplung der Prozessvisualisierung an SPSen	46
2.7.4	Prozessleitsysteme	46
2.8	Zusammenfassung	46
	Übungen zu Kapitel 2	47
3	Modulare SPS-Programmierung nach IEC 61131	51
3.1	Softwaremodell	51
3.1.1	Steuerungskonfiguration und Ressourcen	52
3.1.2	Variablen	54
3.1.3	Tasks	56
3.1.4	Programmorganisationseinheiten	58
3.1.5	Funktionen	60
3.1.6	Funktionsbausteine	61
3.1.7	Instanziierung von Funktionsbausteinen in Programmen	67
3.2	Kommunikationsmodell	68
3.2.1	Datenaustausch innerhalb eines Programms	68
3.2.2	Datenaustausch zwischen Programmen	70
3.3	Programmiermodell	71
3.3.1	SPS-Programmiersprachen	72
3.3.2	Anweisungsliste (AWL)	72
3.3.3	Strukturierter Text (ST)	72
3.3.4	Funktionsbausteinsprache (FUP)	73
3.3.5	Kontaktplan (KOP)	74
3.3.6	Ablaufsprache (AS)	75
3.3.7	Anwender-Datentypen	78
3.4	Zusammenfassung	79
	Übungen zu Kapitel 3	79
4	Entwurf von Verknüpfungssteuerungen	83
4.1	SPS-Software-Engineering	83
4.1.1	Analyse der User-Requirements	84
4.1.2	Objektorientierte Softwarestrukturierung	85
4.1.3	Entwurf der Feldgeräteklassen	87
4.1.4	Entwurf der Ansteuerprogramme	87
4.1.5	Implementierung in der SPS	89
4.1.6	Simulation und virtuelle Inbetriebnahme	90
4.2	Entwurf der Verknüpfungslogik	94
4.2.1	Entwurf von Schaltnetzen	94
4.2.2	Entwurf von Schaltwerken	98
4.3	Ansteuerung der Sensorik und Aktorik	106
4.3.1	Funktionsbausteine zum Einlesen von Sensordaten	106
4.3.2	Funktionsbausteine zum Ansteuern von Motoren	107
4.3.3	Funktionsbausteine zum Ansteuern von Ventilen	108
4.3.4	Schutzfunktionen	109
4.3.5	Betriebsarten	112

4.4	Regelungen	115
4.4.1	Schaltende Regler	116
4.4.2	Reglerbetriebsarten	118
4.4.3	Kontinuierliche Regler	119
4.4.4	Selbsteinstellende Regler	125
4.5	Zusammenfassung	129
Übungen zu Kapitel 4		129
5	Entwurf von Ablaufsteuerungen	133
5.1	Entwurf aus Zustandsfolge	133
5.2	Entwurf aus zeitlicher Abfolge	134
5.3	Modellierung durch Fluss- oder Aktivitätsdiagramm	135
5.4	Programmierung industrieller Abläufe	135
5.4.1	Verknüpfung von CFCs und SFCs	137
5.4.2	Schutzfunktionen und Betriebsarten	139
5.5	Modellierung durch anlagenneutrale Grundfunktionen	142
5.5.1	Prozessanalyse	142
5.5.2	Entwurf anlagenneutraler Grundfunktionen	143
5.5.3	Zusammensetzung der Ablaufsteuerung	144
5.6	Entwurf paralleler Prozesse mit Petri-Netzen	145
5.6.1	Analyse der Erreichbarkeit paralleler Abläufe	146
5.6.2	Modellierung paralleler Prozessabläufe durch Petri-Netze	147
5.6.3	Algebraischer Entwurf zur Koordination paralleler Prozesse	151
5.6.4	Programmmentwurf aus Petri-Netzen	152
5.7	Zusammenfassung	154
Übungen zu Kapitel 5		155
6	Objektorientierte SPS-Programmierung	159
6.1	Klassen und Objekte	159
6.2	Interfaces als abstrakte Klassen	160
6.3	Einsatz von Methoden und Eigenschaften	161
6.4	Vererbung	164
6.5	Objektorientierte Ansteuerung der Feldgeräte	167
6.5.1	Ablaufsteuerungen mit Methoden und Eigenschaften	168
6.5.2	Polymorphe Ansteuerung durch Schnittstellen	170
6.5.3	Anlagenneutrale Ablaufsteuerung	171
6.6	Flexible Ablaufsteuerung durch Rezeptfahrweise	173
6.6.1	Entwurf von Rezeptsteuerungen	173
6.6.2	Objektorientierte Programmierung des Prozessmodells	178
6.6.3	Objektorientierte Programmierung des Anlagenmodells	180
6.6.4	Ausführung von Steuerrezepten	181
6.7	Zusammenfassung	182
Übungen zu Kapitel 6		183
7	Bewegungssteuerungen für die digitale Fabrik	185
7.1	Entwurfsmethodik zur Steuerung von Fertigungsabläufen	185
7.2	Motion-Control in der SPS	190
7.2.1	Aufbau von Motion-Control-Systemen	190
7.2.2	Motion-Control durch SPS-Programmierung nach PLCopen	192

7.3	Steuerung einer Bewegungsachse	193
7.3.1	Interpolation	194
7.3.2	Lageregelung	197
7.4	Steuerung von Werkzeugmaschinen	199
7.4.1	Bahnplanung durch CNC-Programmierung	200
7.4.2	Bewegungsvorgaben durch Kurvenscheiben	202
7.4.3	Elektronisches Getriebe	205
7.5	Robotersteuerungen	206
7.5.1	Koordinatentransformation	207
7.5.2	Programmierung von Bewegungsabläufen	209
7.6	Bildverarbeitung zur Steuerung von Robotern	212
7.6.1	Machine-Vision-Systeme	212
7.6.2	Programmierung der Bildverarbeitung nach IEC 61131	214
7.6.3	3D-Positionsbestimmung	215
7.7	Zusammenfassung	217
	Übungen zu Kapitel 7	217
8	Digital Engineering zuverlässiger Steuerungen	221
8.1	Projektierung in der Cloud	221
8.1.1	CAE-Systeme	222
8.1.2	Gute Engineering-Praxis	222
8.1.3	Digitaler Zwilling	224
8.2	Planung der Automatisierung	225
8.2.1	Prozess- und Anlagenplanung	225
8.2.2	Automatisierungstechnisches Lastenheft	226
8.2.3	Automatisierungstechnisches Pflichtenheft	227
8.3	Realisierung des Automatisierungssystems	229
8.3.1	Automatische Codegenerierung	229
8.3.2	Cloud Engineering	230
8.3.3	Virtuelle Inbetriebnahme	231
8.4	Inbetriebnahme und Qualifizierung	234
8.4.1	Installationsprüfung	235
8.4.2	Funktionsprüfung	235
8.4.3	Produktionsprüfung	236
8.5	Wartung und Instandhaltung	237
8.5.1	Condition Monitoring mit Hilfe des digitalen Zwillings	237
8.5.2	Change Management	238
8.5.3	Fernwartung	239
8.6	Zusammenfassung	240
	Übungen zu Kapitel 8	241
9	Safety und Security in der Industrie 4.0	243
9.1	Funktionale Sicherheit	243
9.1.1	Sicherheitsgerichtete Steuerungen	243
9.1.2	Mehrkanalige SSPSen	245
9.1.3	Einkanalige SSPSen	246
9.1.4	Selbsttests in SSPSen	247
9.1.5	Sicherheitsgerichtete Feldbussysteme	248
9.1.6	Gefahren- und Risikoanalyse	249

9.2	Steuerungen für explosionsgefährdete Betriebe	254
9.2.1	Eigensichere Ansteuerung durch die SPS	254
9.2.2	Eigensichere Feldbussysteme	255
9.3	IT-Security	256
9.3.1	Analyse der Schwachstellen	256
9.3.2	Schutzmaßnahmen	257
9.4	Zusammenfassung	259
	Übungen zu Kapitel 9	259
10	Industrial IoT in der Prozessautomatisierung	261
10.1	Horizontale Vernetzung Cyber Physical Systems	262
10.1.1	Vernetzung mit Industrial Ethernet	262
10.1.2	Werkzeuge zur Netzwerkintegration	263
10.1.3	Datenaustausch zwischen SPSen	264
10.2	Vertikale Vernetzung in die Cloud	266
10.2.1	Interoperabilität durch OPC-UA	267
10.2.2	MQTT zur Anbindung von Kleingeräten an die Cloud	269
10.2.3	Plattformunabhängige Webvisualisierung	270
10.3	Betriebsdatenauswertung und Cloud Services	272
10.3.1	Betriebsdatenerfassung	272
10.3.2	Betriebsdatenauswertung	277
10.3.3	Asset Management	279
10.3.4	Qualitätsmanagement	281
10.4	Automatisierte Produktionsplanung und -steuerung	282
10.4.1	Produktionsplanung durch Enterprise-Ressource-Planning	283
10.4.2	Produktionssteuerung in Manufacturing Execution Systems	284
10.4.3	Logistic Execution Systems	285
10.4.4	Supply Chain Management	286
10.5	Zusammenfassung	286
	Übungen zu Kapitel 10	287
11	Die Zukunft der SPS in der Industrie 4.0	290
	Literaturverzeichnis	293
	Anhang	299
	A SPS-Lern-und-Übungsseite	299
	B Funktionsbaustein-Bibliotheken	300
	Stichwortverzeichnis	303

1 Einführung

Eine speicherprogrammierbare Steuerung (SPS) ist ein industrieller Rechner mit einfachen Schnittstellen zu Sensoren und Aktoren. Sie besteht wie in Bild 1.1 gezeigt aus einer Central Processing Unit (CPU), die Programme ausführt und über Ein-/Ausgangsbaugruppen oder Busverbindungen Sensordaten einliest und Befehle zur Ansteuerung von Aktoren, wie etwa Motoren oder Ventilen, ausgibt. Durch die Ansteuerung der Sensoren und Aktoren, die in einer Anlage oder Maschine eingebaut sind, ermöglicht die SPS, dass Produktionsprozesse automatisiert ablaufen.

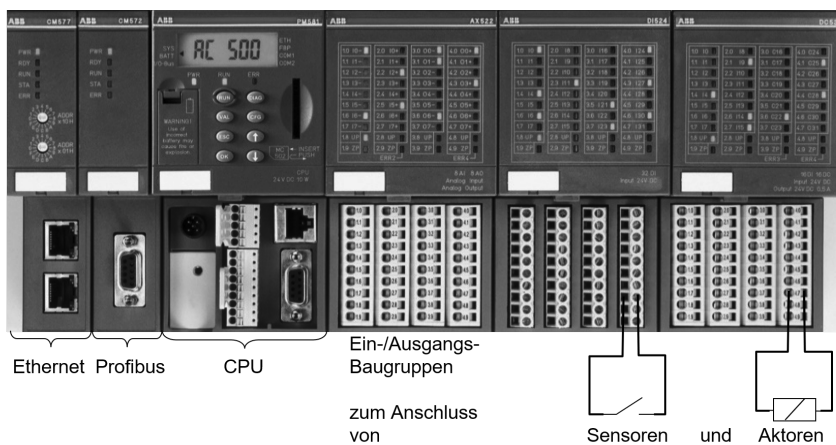


Bild 1.1: Speicherprogrammierbare Steuerung AC500 von ABB mit Baugruppen für Ethernet- und Profibus-Kabel, CPU sowie Ein-/Ausgangssignalkabel (von links nach rechts) [1]

Die SPS ist seit Jahrzehnten ein bewährtes, millionenfach eingesetztes Automatisierungssystem, dessen Funktionsumfang und Einsatzgebiete kontinuierlich erweitert wurden. Der Aufbau von SPS-basierten Automatisierungssystemen wird in Kapitel 2 dieses Buchs ausführlich behandelt.

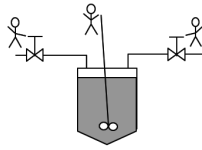
1.1 Entwicklung der Automatisierungstechnik

Die historische Entwicklung der industriellen Produktion wird gemäß Bild 1.2 in vier Etappen gegliedert: Durch die erste industrielle Revolution wurde die Mechanisierung von Maschinen und Anlagen erreicht, z. B. durch Handantriebe für Ventile und Rührwerke. In der zweiten industriellen Revolution wurden Antriebe elektrisch angesteuert und einfache Verknüpfungen durch Relaischaltungen realisiert. Die Erfindung der SPS im Jahr 1968 läutete die dritte industrielle Revolution und damit die Automatisierung von Produktionsanlagen ein.

Ziel der heutigen Digitalisierung in der Industrie 4.0 ist es, dass möglichst alle Maschinen und Anlagenteile ihre Daten zentral in einer Cloud speichern. Durch Auswertung dieser großen Datenmenge (Big Data) erhofft man sich neue und frühzeitigere Erkenntnisse über die Prozesssituation. Mit Hilfe der SPS können Maschinen und Anlagenteile

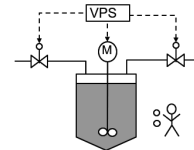
1. Industrielle Revolution Mechanisierung

um 1800



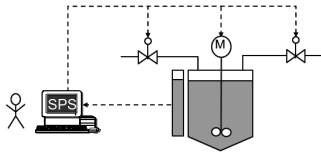
2. Industrielle Revolution Elektrifizierung

um 1900



3. Industrielle Revolution: Automatisierung

um 1970



4. Industrielle Revolution Digitalisierung

um 2010

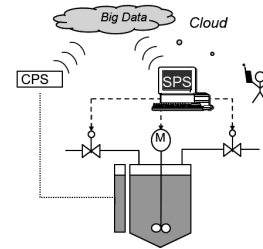


Bild 1.2: Die vier industriellen Revolutionen entlasten den Menschen zunehmend von manuellen und kleinteiligen Aufgaben

mit der Cloud verbunden werden und als sog. Cyber Physical Systems (CPS) direkt auf die in der Cloud erkannten Situationen reagieren und ggf. autonom Entscheidungen treffen [7, 11, 101].

Vor der Erfindung der SPS konnte eine Verknüpfungslogik nur als verbindungsprogrammierte Steuerungen (VPS) mit Hilfe von Relais-Schaltungen wie in Bild 1.3 realisiert werden. Ein Relais besteht aus einer Spule sowie einem Schalter, der vom Magnetfeld der Spule angezogen oder abgestoßen wird und somit einen anderen Stromzweig schließt bzw. öffnet.

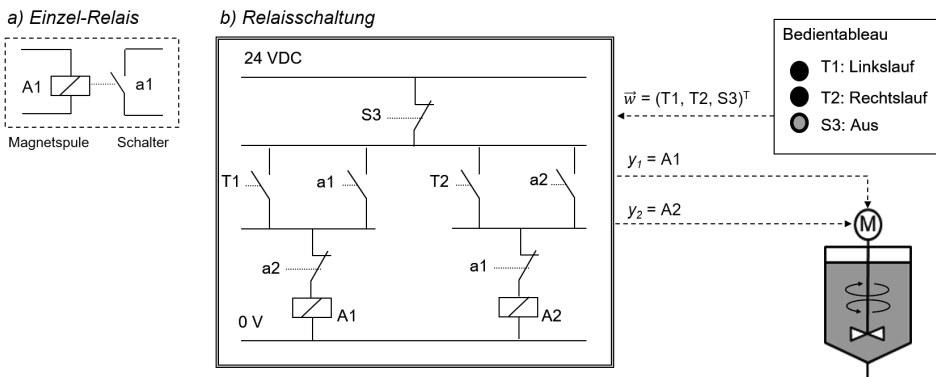


Bild 1.3: Verbindungsprogrammierte Steuerung (VPS) zur Ansteuerung eines Motors mit zwei Drehrichtungen, die gegenseitig verriegelt sind

Die Steuerungslogik wird so durch feste Draht- oder Leiterplattenverbindungen aufgebaut und ist dementsprechend unflexibel. Trotzdem werden VPSen bis heute für Schaltungen mit sehr hohen Sicherheitsanforderungen, die in Kapitel 9 näher erläutert werden, eingesetzt. In speicherprogrammierbaren Steuerungen wird die Verknüpfungslogik durch Software realisiert (siehe Bild 1.4), was den Vorteil hat, dass das Programm im Speicher der SPS hinterlegt ist und jederzeit verändert werden kann.


Beispiel 1.1: Steuerung eines Rührwerks mit VPS und SPS

Ein Koaxialrührwerk kann ein Gemisch in einem Behälter in zwei Drehrichtungen durchmischen. Jeweils ein Relais in Bild 1.3 aktiviert die beiden Drehrichtungen des Motors. An einem Bedientableau kann der Bediener vor Ort über die Taster T1 und T2 die Drehrichtung gegen den Uhrzeigersinn (Linkslauf) oder im Uhrzeigersinn (Rechtslauf) ansteuern und durch Betätigung des Aus-Schalters S3 den Motor abschalten.

Die VPS steuert über die Relais A1 und A2 den Links- bzw. Rechtslauf des Rührers an. Drückt der Bediener den Taster T1, wird das Relais A1 mit Strom versorgt und der Kontakt a1 schließt den linken Zweig der Schaltung bzw. öffnet den rechten Zweig. Springt der Taster T1 dann wieder auf, wird A1 trotzdem weiter mit Strom versorgt. Nun bewirkt ein Schließen von T2 nichts, weil a1 geöffnet ist und die Versorgung von A2 unterbricht. Somit wird ein direktes Umschalten vom Links- in den Rechtslauf und umgekehrt verriegelt. Durch Betätigung des Öffners S3 wird die Versorgung beider Relais unterbrochen, der Antrieb wird abgesteuert und das Rührwerk bleibt stehen. Danach kann wieder eine beliebige Bewegungsrichtung ausgewählt werden.

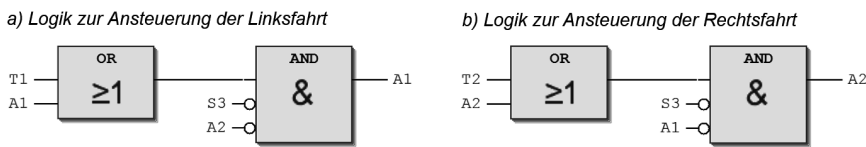


Bild 1.4: Ansteuerung des Antriebs mit zwei Drehrichtungen in einem SPS-Programm

Bei einer Speicherprogrammierung wird die Logik als Software in einer SPS eingegeben. Dabei kann das Programm direkt als Logik-Schaltplan wie in Bild 1.4 erstellt werden, was dem Papierentwurf für die Schaltung sehr nahe kommt. Ein Vergleich von VPS und SPS-Programm zeigt, dass eine Parallelschaltung von Schaltern bzw. Tastern in der Software durch ein ODER-Gatter und eine Reihenschaltung durch ein UND-Gatter nachgebildet wird. □

Im Lauf der Jahre wurde der Funktionsumfang der SPS über die rein binäre Logikverarbeitung hinaus weiterentwickelt, so dass auch *Analogwertverarbeitung* und *Regelungen* von der SPS ausgeführt werden konnten. Mit dem Erscheinen der IEC 61131 im Jahre 1993 wurde die SPS-Programmierung durch herstellerunabhängige Programmiersprachen genormt, was den Engineering- und Instandhaltungsaufwand der Software deutlich reduzierte. Der Einsatz grafischer Programmiersprachen wie der Funktionsbausteinsprache in Bild 1.4 ermöglicht es auch Laien, die SPS-Software aus Anwendersicht zu verstehen.

1.2 Automatisierungssysteme

Außer speicherprogrammierbaren Steuerungen gibt es noch andere Automatisierungssysteme für spezielle Anwendungen. So werden in der Verfahrenstechnik *Prozessleitsysteme* (PLSe) eingesetzt. Sie bestehen in der Regel aus mehreren Steuerungen (z. B. SPSen), die mit Bedien- und Beobachtungssystemen in einem Netzwerk verbunden sind. PLSe eignen sich für große Anlagen, denn sie bieten viele vorkonfektionierte Module und Elemente zur automatischen Codegenerierung und Prozessführung [39].

Für Bewegungssteuerungen von *Werkzeugmaschinen* werden CNC-Steuerungen (Computerized Numerical Controls) eingesetzt, die Motoren mit sehr schnellen Zykluszeiten synchron regeln und ihre Bewegungen koordinieren. Die Ansteuerung von Industrierobotern erfolgt meist durch herstellereigenspezifische *Robotersteuerungen* (Robot Controls, RC). *Mikrocontroller* ermöglichen eine preiswerte und energiesparende Automatisierung kleinerer Anwendungen. Doch die Anbindung der Sensoren und Aktoren erfordert

dabei meistens zusätzliche Entwicklungsarbeit, weil es nur wenige, nicht standardisierte Schnittstellen gibt. Auch ein *PC* kann als Steuerungsrechner eingesetzt werden.

Die meisten dieser Automatisierungssysteme können mit SPS-Programmiersprachen nach IEC 61131 programmiert werden, was in den Kapiteln 3-6 ausführlich behandelt wird.

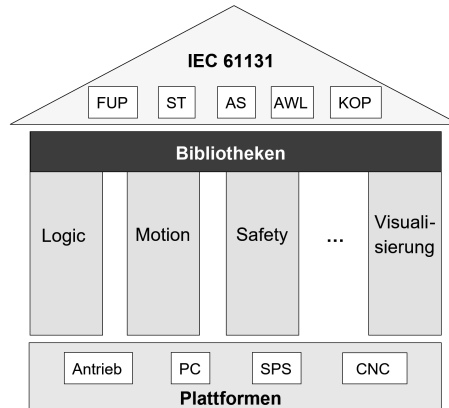


Bild 1.5: Klassische Logikschaltungen wachsen mit Motion- und Safety-Funktionen unter dem Dach der IEC 61131 zusammen. Die hardwareunabhängigen Programmiersprachen Funktionsbausteinsprache (FUP), Strukturierter Text (ST), Ablaufsprache (AS), Anweisungsliste (AWL) und Kontaktplan (KOP) ermöglichen die Implementierung auf unterschiedlichen Plattformen [116]

Da die Programmierung dieser Automatisierungssysteme standardisiert ist, wachsen die verschiedenen Automatisierungsaufgaben der numerischen und binären Steuerung und Regelung sowie Sicherheitsfunktionen und die Prozessbedienung und -beobachtung unter dem Dach der IEC 61131 zusammen (s. Bild 1.5). Was bisher in getrennten Systemen programmiert und implementiert war, wird nun in einem standardisierten Programmiersystem entwickelt und kann auf unterschiedlichen Hardwareplattformen implementiert werden. Statt der englischen Bezeichnung für SPS (Programmable Logic Controller, PLC) verwendet man nun häufig die Abkürzung PAC für Programmable Automation Controller [116, 155]. Durch Einsatz leistungsstarker und kostengünstiger Industrie-PCs wird der Funktionsumfang von SPSen um Motion-, Robotik- und Bildverarbeitungsfunktionen ausgedehnt. Diese SPSen werden auch als Motion-Control-Systeme bezeichnet und in Kapitel 7 beschrieben.

1.3 Aufgaben in der Industrie 4.0

In der Regel findet man zwei große Einsatzfelder industrieller Automatisierungssysteme, und zwar zum einen in der Verfahrenstechnik [39], wie z.B. Chemie oder Life Science, und zum anderen in der Fertigungstechnik, z. B. zur Fabrikautomatisierung in der Automobilindustrie.

Zielsetzung der Industrie 4.0 ist es, Rechner, Anlagen und Menschen miteinander zu vernetzen, um den Produktionsprozess zu optimieren [92]. Die SPSen spielen dabei eine wichtige Rolle, weil sie die von Sensoren gemessenen Daten über den Zustand der Anlage sammeln und über das Internet an eine Daten-Cloud übertragen. Durch Auswertung dieser Daten in der Cloud bekommt die SPS Befehle zurück, um die Automatisierung der Maschinen und Anlagen zu verbessern.

Das folgende Beispiel zeigt eine typische Produktionsanlage in der Industrie 4.0. Ihre automatisierten Komponenten werden als Cyber Physical Systems (CPS) bezeichnet. Sie tauschen Daten über die Cloud aber auch direkt miteinander aus.

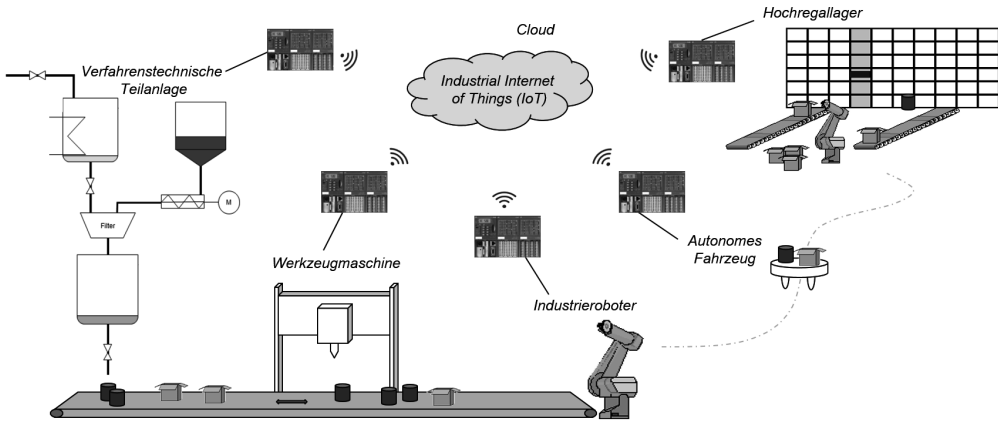


Bild 1.6: Produktionsanlage bestehend aus mehreren automatisierten Anlagenteilen (Cyber Physical Systems), die über die Cloud Daten austauschen

Beispiel 1.2: Produktionsanlage in der Industrie 4.0

In der Produktionsanlage nach Bild 1.6 werden Rohstoffe aus einem Hochregallager ausgelagert, von einem Roboterarm auf ein autonomes Fahrzeug gelegt und einer verfahrenstechnischen Anlage zur Herstellung eines Produkts zugeführt. Das Produkt wird in Fässer abgefüllt und über ein Förderband einer Werkzeugmaschine zum Verschließen zugeführt. Ein Roboterarm greift die Fässer und stellt sie auf dem autonomen Fahrzeug ab, das sie wieder zum Lager zurückbringt.

Die Ausführung dieser Prozesse erfolgt durch SPSen, die die Maschinen und Anlagenteile automatisch ansteuern. Die Daten über Art und Menge der hergestellten Waren und eingesetzten Rohstoffe sowie die Dauer der in Anspruch genommenen Anlagenteile werden in der Cloud, im sog. Industrial Internet of Things (IIoT), verwaltet. Die SPSen übertragen die Prozessdaten an die Cloud und bekommen von ihr Befehle, welche Waren wann ausgelagert oder produziert werden sollen. \square

1.3.1 Messen, Steuern, Regeln und Überwachen

Auch im Zeitalter von Industrie 4.0 besteht die Hauptaufgabe von SPSen in der Steuerung und Regelung von Maschinen und Anlagenteilen. Nach IEC 60050 versteht man unter der Steuerung eines Prozesses den Vorgang, bei dem durch Messung von Prozesszuständen über bestimmte Gesetzmäßigkeiten Stellwerte zur Beeinflussung des Prozesses erzeugt werden [52].

Im Steuerkreis in Bild 1.7 werden diese Gesetzmäßigkeiten z. B. in einer SPS programmiert. Dabei werden die Soll-, Mess- und Stellwerte nur zu diskreten Abtastzeitpunkten k eingelesen bzw. ausgegeben.

Kennzeichen einer Steuerung ist der *offene* Wirkungsweg, bei dem die Steuerung Stellwerte $\vec{y}(k)$ erzeugt, die den Prozess gemäß den vorgegebenen Sollwerten $\vec{w}(k)$ beeinflussen. Häufig berücksichtigt das Automatisierungssystem hierfür auch Messwerte $\vec{x}(k)$ von Sensoren, so dass ein *geschlossener* Wirkungsweg entsteht. Im Unterschied zur Regelung werden bei einer Steuerung die Stellwerte aber nicht kontinuierlich durch die Messwerte verändert.

Unter dem Oberbegriff Steuern (to control) wird häufig die gesamte Automatisierung verstanden, nämlich das Messen, Steuern, Regeln und Überwachen einer Anlage.

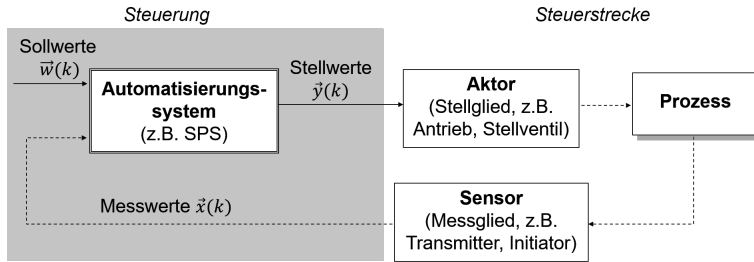


Bild 1.7: Aufbau eines Steuerkreises

Auch die Einsatzmöglichkeiten einer SPS erstrecken sich über diese Disziplinen, wie die folgenden Beispiele zeigen.

Beispiel 1.3: Steuern

Der Zufluss von Flüssigkeit in Bild 1.8 startet durch Öffnen des Zulaufventils. Ein Niveauschalter, z. B. in Form einer Schwinggabel, kann unterscheiden, ob sich Luft oder eine Flüssigkeit zwischen seinen Platten befindet, und sendet je nachdem ein binäres TRUE- oder FALSE-Signal an die SPS. Sobald die Steuerung also erfährt, dass die Flüssigkeit im Behälter bis zu den Platten der Schwinggabel angestiegen ist, schließt sie automatisch das Zulaufventil. □

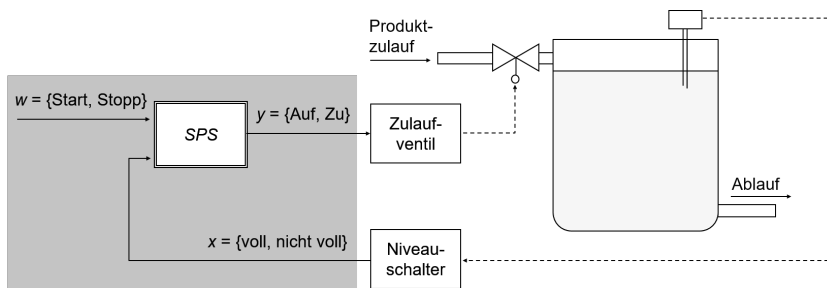


Bild 1.8: Steuerung des Produktzulaufs in einen Behälter

Beispiel 1.4: Regeln

Um ein Produkt auf eine gewünschte Temperatur einzustellen, führt die SPS die in Bild 1.9 dargestellte Regelung aus. Ein Thermometer misst die Produkttemperatur im Behälter. Je nach Abweichung des Temperatur-Istwerts x vom Sollwert w verändert das Regelungsprogramm in der SPS beim Heizen den Stellgrad y_1 des Warmwasserventils und beim Kühlen den Stellgrad y_2 des Kaltwasserventils. Dementsprechend fließt mehr oder weniger heizendes bzw. kühlendes Medium durch den Behältermantel und die Temperatur nähert sich so dem vorgegebenen Sollwert an. □

Beispiel 1.5: Überwachen

Viele Anwendungen zielen auch darauf ab, den Bediener der Anlage auf besondere Betriebszustände, wie z. B. Störungen, aufmerksam zu machen. Wenn der Vorratsbehälter in Bild 1.10 leer läuft, meldet dies der Niveauschalter an die SPS, die daraufhin eine Hupe ansteuert. Dadurch kann das Bedienpersonal rechtzeitig veranlassen, dass neues Produkt zugeführt wird. Somit handelt es sich um eine einfache Mensch-Maschine-Schnittstelle (Human Machine Interface, HMI). □

1.3.2 Auswerten und Planen im Industrial Internet of Things

In der Industrie 4.0 übertragen die von SPSen automatisierten Maschinen und Anlagen-teile als Cyber Physical Systems ihre gemessenen Daten an die Cloud [7]. Die Analyse

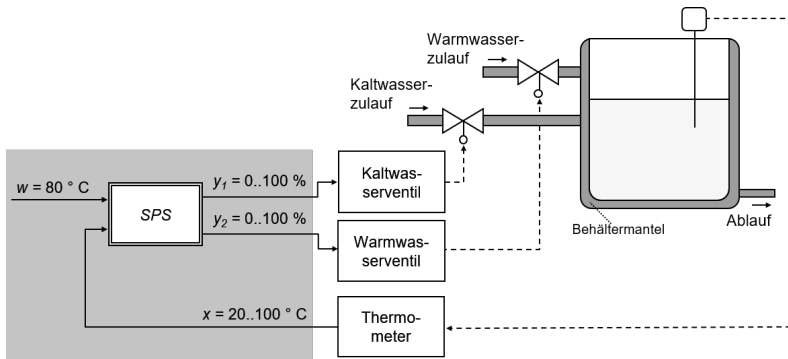


Bild 1.9: Regelung der Behältertemperatur in einer Chemieanlage

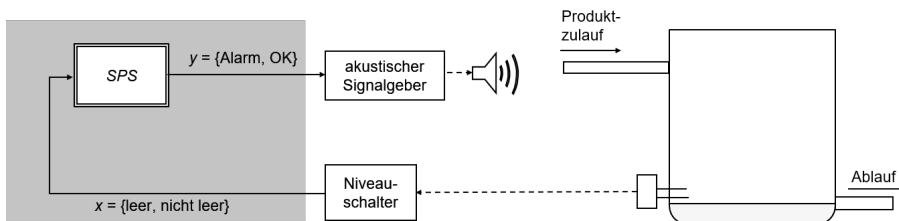


Bild 1.10: Füllstandsüberwachung eines Vorratsbehälters in einer Chemieanlage

dieser so erworbenen großen Datenmenge (Big Data) ermöglicht es, ein industrielles Internet der Dinge (Industrial IoT) aufzubauen, um Ressourcen und Produktionsabläufe genauer bilanzieren und besser planen zu können. Auf Grundlage der gesammelten Daten können Vorhersagen über Anlagenzustände und das Prozessverhalten gemacht werden. Diese dienen dem System dazu, autonome Entscheidungen zu treffen, um z. B. Störungen zu vermeiden und den Prozess zu optimieren [135].

Die Automatisierungsaufgaben sind hierarchisch organisiert. Wie in Bild 1.11 dargestellt ordnet man die Aufgaben der Feldebene, der Prozessleitebene sowie der Betriebs- und Produktionsleitebene zu. Im Unterschied zum zentralen Cloud-Computing der Produktionsleitebene werden die automatisierungstechnischen Aufgaben durch ein sog. Edge-Computing dezentral ausgeführt. Die hierfür eingesetzten SPSen, CPS, Web Clients, Human Machine Interfaces (HMI) befinden sich sozusagen am Rand (*engl.* Edge) des gesamten Netzwerks, in dessen Mittelpunkt die Cloud steht. Die Aufgaben der Betriebsleitebene werden teils zentral in der Cloud, teils in dezentralen Rechnern ausgeführt und als Fog-Computing bezeichnet.

Das Arbeiten mit Cloud-Systemen von Anbietern wie Google, Amazon oder Microsoft ist in der Büro- oder IT-Welt (Information Technology) schon etabliert [32], während Automatisierungssysteme, heute auch als OT-Systeme (Operational Technology) bezeichnet, bislang nicht an die Cloud angekoppelt waren. Durch Verbindung der IT- und OT-Systeme soll nun das industrielle IoT entstehen.

Das Zusammenspiel zwischen betriebswirtschaftlichen und automatisierungstechnischen Aufgaben im industriellen Internet of Things wird in Kapitel 10 behandelt.

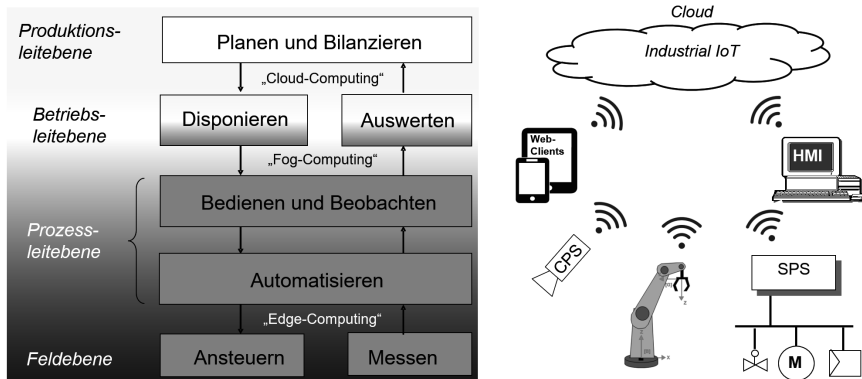


Bild 1.11: Hierarchie der Automatisierungsaufgaben im industriellen Internet of Things (IIoT)

1.3.3 Virtualisierung und Testen mit digitalem Zwilling

Die in der Cloud gespeicherten Daten stellen ein digitales Abbild der Anlage dar. Durch Vergleich des realen Anlagenverhaltens mit dem gewünschten Verhalten ihres digitalen Zwillings können Abweichungen vom Normalbetrieb erkannt und frühzeitig prognostiziert werden, um Störungen und Unfälle zu vermeiden [89].

Außerdem ermöglicht ein digitaler Zwilling eine virtuelle Inbetriebnahme der Anlage. Dabei wird die entwickelte Steuerungssoftware mit Hilfe von Simulationsmodellen getestet. Kapitel 8 beschreibt die Vorgehensweise beim Engineering der Automatisierungssysteme für die Industrie 4.0.

Wiederholungsfragen

1. Was ist eine SPS?
2. Welche Ziele verfolgt die Industrie 4.0?
3. Wie sind Steuerung und Regelung definiert?
4. Durch welche Ebenen werden die Aufgaben in der Industrie 4.0 hierarchisiert?
5. Wofür dient ein digitaler Zwilling?

Übung 1.1: Steuerkreis und Regelkreis

Die Innentemperatur T_i eines Raumes soll durch die Steuerung einer Heizungsanlage auf einen gewünschten Sollwert eingestellt werden. Hierfür ist der Öffnungsgrad des Warmwasserzulaufventils so einzustellen, dass die Heizkörper mit ausreichend Wärme versorgt werden, um den Temperaturunterschied zwischen Innentemperatur und Sollwert auszugleichen. Die Messung der Außentemperatur T_a erfolgt durch einen Außentemperaturfühler.

- a) Zeichnen Sie den Steuerkreis!
- b) Welches sind die Ein- und Ausgangsgrößen der Steuerung und ihre typischen Wertebereiche?
- c) Welchen Zusammenhang muss die Steuerung berechnen, um geeignete Stellwerte vorgeben zu können?
- d) Nun wird die Innentemperatur T_i in einem Raum gemessen. Sie soll durch einen Thermostat-Regler auf einen gewünschten Sollwert eingestellt werden. Zeichnen Sie den Regelkreis!
- e) Welchen Zusammenhang muss die Regelung berechnen?

2 Aufbau von Steuerungen für die Industrie 4.0

Automatisierungssystemen kommen in der Industrie 4.0 eine entscheidende Bedeutung zu, denn sie sammeln die Daten der Feldgeräte (Sensoren und Aktoren) und stellen sie als Edge-Controller für die Cloud zur Verfügung. Außerdem erhalten sie aus der Cloud Informationen über den Zustand der Gesamtanlage und versetzen damit Maschinen in die Lage, als Cyber Physical Systems *autonom* zu arbeiten.

Im Folgenden werden Aufbau und Strukturen von Automatisierungssystemen am Beispiel speicherprogrammierbarer Steuerungen (SPS) aufgezeigt. Im Anschluss daran werden Konzepte zur Ankopplung der Sensoren und Aktoren an die Steuerung vorgestellt. Schließlich werden Systeme zur Prozessvisualisierung erläutert, durch die ein Bediener die Anlage bedienen und beobachten.

2.1 SPS-Aufbau

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

2.1.1 Zentralbaugruppe

Das Kernstück einer SPS 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 einen Flashspeicher oder EEPROM (Electrically Erasable Programmable Read Only Memory), in dem alle Anwender- und Betriebssystemprogramme wie in einem Archiv *offline* gespeichert werden. Der EEPROM ist häufig als steckbare SD-Card 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 [128, 153].

Die Auswahl der CPU erfolgt gemäß der Größe und Anforderungen der zu automatisierenden Anlage. Auswahlkriterien sind z. B. Verarbeitungsgeschwindigkeit, Größe des Arbeitsspeichers, der Umfang an E/A-Adressen sowie Art und Anzahl der Busverbindungen.

2.1.2 Peripheriebaugruppen

Eine weitere Besonderheit einer SPS sind spezielle Ein-/Ausgangsbaugruppen, die das Einlesen von Sensorinformationen und Ausgeben von Befehlen an die Aktoren besonders einfach machen. Dabei wird ein Sensor oder Aktor mit zwei Kupferdrähten zum Aufbau eines Gleichstromkreises an einen binären bzw. analogen Ein-/Ausgangskanal

Stichwortverzeichnis

A

Ablaufreihenfolge, 185, 186
Ablaufsprache, 75, 133
Ablaufsteuerungen, 133, 145, 168, 171
Ablauftypicals, 142
Abtastzeit, 121
Abtastzyklus, 121
Access Specifier, 164
Achsguppe, 206, 209
Achsinterface, 192
Achskoordinatensystem, 206
Acknowledge, 249
ACTION, 178
Änderungsdatenbank, 239
Aktionen, 75, 133
Aktivitätsdiagramm, 135
Aktorik, 106, 161
Akzeptanzkriterium, 92
Alarmmeldungen, 43
Alternativverzweigung, 77
Ampelanlage, 129
Analog-/Digital-Umwandlung, 33
Analoge Ausgangsbaugruppen, 34
Analoge Eingänge, 81
Analoge Eingangsbaugruppen, 32
Analoger Sensor, 69
Anlagenmodell, 180, 224
Anlagenschema, 42, 226
Anti-Reset-Windup-Maßnahme, 122
Antriebstechnik, 192
Antriebswelle, 193
Anweisungsliste (AWL), 72
Anwender-Datentypen, 78
Anwender-Funktion, 60
Anwender-Funktionsbaustein, 62
Anzeige- und Bedienkomponente, ABK, 49
Arbeitsstromprinzip, 112
ASI-Bus, 39
Asset, 279
Asset Administration Shell, AAS, 225
Asset Management, 237, 279
Assets, 222, 261
Aufenthaltsdauer, 251
Auftragsüberwachung, 284
Aufwandsmaß, 96
Ausführungsplanung, 227
Ausgabeabbild, 30
Ausgangsschaltnetz, 99
Ausschaltverzögerung, 63, 103
Authentifizierung, 258
Automatenentwurf, 98, 129
Automatik, AUT, 113
AutomationML, 230
Automatische Codegenerierung, 229, 241
Automatisierungssysteme, 26

Autonomie, 290

Autotuning, 128

Azure, 275

B

Bahnlänge, 196
Bahnplanung, 200
Bahnsteuerung, 210
Basic Function, 171, 172
Bausteinbibliothek, 232, 241
Bedienphilosophie, 227
Behältersteuerung, 184
Beschleunigungszeit, 196
Bestimmungszeichen, 77
Betriebsart, 139, 163, 165
 Einzelsteuerfunktion, 112
 MAN, 122
 Regler, 118
Betriebsartenhierarchie, 115
Betriebsartenumschaltung, 113
Betriebsdatenauswertung, 272, 277
Betriebsdatenerfassung, 272, 288
Bewegungsachse, 193
Bewegungsprofil, 194
Bibliothek, 167
Big Data, 22, 272, 291
Bildaufbereitung, 214
Bildaufnahme, 214
Bildsegmentierung, 214
Bildverarbeitung, 212, 216, 220, 291
 Stereo, 220
Bimetallschalter, 111
Binärer Ausgang, 32
Binärer Eingang, 31
Binärer Sensor, 74
Black-Box-Test, 234
Bridge, 263
Buskommunikation, 248

C

CAE-System, 199, 222, 239
CAEX, 225, 228
CAM-Editor, 193, 203
CamIn, 204
CAM-Table, 202
Cause-and-Effect-Matrix, 89, 93
Central Processing Unit, 16
CFCs, 83, 133, 137, 154, 167
Change Management, 238
Change Order, 238
Charge, 282, 284
Chargenprotokoll, 277
Client-Server-Modell, 45, 267
Cloud, 16, 19, 21, 26, 28, 199, 221, 261, 269, 279, 280, 290

Cloud Engineering, 230
 Cloud Services, 272
 Cloud-Computing, 22
 CNC-Editor, 202
 CNC-Programmierung, 200, 218
 CNC-Steuerungen, 18
 Codesys, 25, 53, 299
 Codesys SoftMotion, 191
 Codesys-Store, 300
 Computerized Numerical Control, CNC, 190
 Condition Monitoring, 237, 279
 CPU, 24, 56, 58
 CRC-Wert, 249
 CSMA/CD-Verfahren, 262
 CTUD, 65
 Cyber Physical Systems, 17, 24, 83, 262, 283, 284
 Cyberattacken, 256
 Cyclic Redundancy Checks, 249

D

D-Anteil, 121
 Datenbank, 228, 238
 Datenbaustein, 70, 87, 137
 Datenkapselung, 163
 Datenmodelle, 225
 Datenspeicher, 30
 Datentyp, 55
 Standard- nach IEC 61131, 55
 Dauerschwingung, 126
 Deadlock, 147
 Deep Learning, 272
 Defense-in-Depth, 257
 Demilitarisierte Zone (DMZ), 257
 Determiniertheit, 262
 Device Type Manager, 52
 Dexpi, 225
 Diagnosemeldungen, 279
 Differenzial
 1. Ordnung, 121
 Digital Engineering, 221
 Digitale Ausgangsbaugruppen, 31
 Digitale Eingangsbaugruppen, 30
 Digitale Fabrik, 185, 216
 Digitaler Zwilling, 23, 224, 237, 240, 290
 Direkte Perspektivische Transformation, 215
 Disjunktion, 95
 Disposition, 285, 289
 Diversität, 246
 DNF, 95
 Dosieren, 177
 Drehgeber, 35
 Drehmaschine, 199, 203
 Drehzahlregler, 198
 Dreipunktregler, 116, 170, 183
 Dreitankanlage, 156, 183
 Dreiwegeventil, 109
 Dual-Port-RAM, 246
 Dualzahl, 96
 Durchflussregelung, 134
 Dynamisierung, 45

E

E/A-Baugruppen, 248
 E/A-Kanäle, 54, 89, 247
 E/A-Zuordnung, 89
 Edge-Computing, 22
 Edge-Controller, 24, 28, 272
 Edge-Gateway, 221
 EEPROM, 24, 247
 Eigenschaften (Properties), 160, 161, 176, 182
 Eigensicherheit, 36, 38, 254, 260
 Ein/Aus-Motor, 62
 Eingabeabbild, 29, 33
 Eingangsdatenwort, 33, 106
 Eingangskombination, 94
 Eingangsschaltnetz, 99, 104, 105
 Eingangsstrom, 33
 Einschaltverzögerung, 63
 Einstellregeln, 124
 Eintrittswahrscheinlichkeit, 251
 Einzelsteuerfunktion, 114, 163
 Electronic Batch Recording, 277, 282
 Electronic Device Description, 52
 Elektronisches Getriebe, 205
 Encoder, 35, 131, 206
 Endlagenüberwachung, 132
 Energieverbrauch, 277
 Engineering
 Detail-, 227
 Enterprise-Ressource-Planning (ERP), 283
 Entwurfsmethodik, 99, 185
 Erdschlussschleifen, 36
 Ereignisbaumanalyse, 249, 250
 Erreichbarkeitsgraf, 147, 152, 158
 EtherCAT, 41, 191
 Ethernet TCP/IP, 265
 Ethernet-basierte Feldbusse, 40
 Etikettierung, 285
 Euler-Winkel, 207
 EVA-Prinzip, 29, 56
 EVA-Zyklus, 133
 Explosionsgefahr, 36, 254
 Explosionsschutz, 47
 EXTENDS, 165

F

Faceplate, 42, 193
 Factory Acceptance Test (FAT), 93, 233
 Fail-Safe-Prinzip, 243
 Fehlerbaumanalyse, 250
 Fehlerbehebung, 280
 Fehlerdiagnose, 280
 Fehlererkennung, 280
 Fehlersichere SPS, 29
 Feldbus, 36, 38
 Feldbusbarrieren, 255
 Feldbussysteme, 39, 248
 Feldbustechnik, 36
 Feldgeräte, 279
 Feldgeräteklassen, 176
 Fernwartung, 239
 Fertigungsablauf, 185, 211, 217
 Fertigungstechnik, 19

Fertigungszelle, 189, 206
Firewall, 26, 258
Fliegende Säge, 203, 218
Flussdiagramm, 135
Fog-Computing, 22
Förderband, 159
Frames, 233, 242
Fräsmaschine, 199
Füllstandmessung, 81
Füllstandregelung, 122
Funktionale Sicherheit, 243
Funktionen, 60
Funktionsbaustein, 61, 159
 Anwender-, 62
 Bibliotheken, 300
Funktionsbausteinsprache (FBS), 73
Funktionsprüfung, 235

G

Galvanische Trennung, 31, 36, 255
GAMP-Leitfaden, 222
GASE-Risikoparameter, 251
Gateway, 264
G-Codes, 200
Gebinde, 285
Gedächtnis, 94, 99, 121
Gefahrenabwendung, 251
Gepäckanlage, 99
Gerätefehler, 109
Gerätemodell, 225
Geräte-Repository, 53
Gerätespezifikation, 226
Gerätestammdatei, 52
Gleichstrommotor, 36, 206
Global Variable List, 55
Globale Netzwerkvariablen, 264
Globale Variablen, 70
Greiferkoordinatensystem, 206
Grundfließbild, 225, 241
Grundfunktion, 157, 172, 173, 176, 177, 184
 anlagenneutral, 142, 174
 polymorph, 175, 176, 180
Grundfunktionsbausteine, 145
Grundoperation, 173, 175
Grundrezept, 173

H

Halteglied, 99, 106
Hardware-in-the-Loop (HIL), 232
Hardwarekonfiguration, 89, 231
Hardware-SPS, 27, 49
Hardwarestrukturplan, 52, 227, 229, 235
Hauptantrieb, 199
HAZOP-Methode, 249
Heizungsanlage, 23
HMI, 155
Hochregallager, 143, 151, 157, 186, 242, 285
Hochverfügbare SPS, 29
HTML5, 271
Hub, 263
Human Machine Interface, 26, 42
Hysterese, 116

I

I-Anteil, 121
Implizite Variablen, 157
Impulszähler, 66
Industrial Ethernet, 191, 262
Industrial IoT, 20, 23, 261, 291
Industrie 4.0, 20, 185, 259, 282
Industrielle Revolution, 16
Industrieroboter, 206
Informationsmodelle, 224
Infrastructure as a Service (IaaS), 272
Inkrementalweggeber, 197
In-Prozess-Kontrolle, 281
Installationsprüfung, 235
Instandhaltung, 237
Instanz, 67
Instanzvariable, 159
Integrationstest, 93, 235
Integrationswerkzeuge, 263
Interface, 160, 166
Internet of *Things* (IoT), 261
Interpolation, 194, 218
Interpolationszeit, 197
Inverse perspektivische Transformation, 215
IO-Controller, 41
IO-Devices, 41
IoT, 22
 -Gateway, 28, 221
 -Gerät, 28
 -Hub, 275
 -Plattform, 275
IP-Adresse, 239
IPSec-Protokoll, 239
Irreversibel, 147
Istwert, 20, 116, 120
Itemliste, 267
IT-Security, 256

J

JavaScripts, 271

K

Kameramodell, 215
Kanaladressen, 57, 88
Klasse, 67, 159, 162
 abstrakt, 171
Klassendiagramm, 86, 160
KNF, 96
Kommissionierung, 285
Kommunikationsdiagramm, 168
Kommunikationsmodell, 68
Kompakte Lösung, 152
Komplexität, 97
Komposition, 174, 178
Königswelle, 202
Konjunktion, 94
Konstanten, 55
Kontaktplan (KOP), 74
Koordinatentransformation, 207
Koordination, 188
Koordination paralleler Prozesse, 151, 153
Koordinationsentwurf, 158

Koordinationsprogramm, 153
Kosinussatz, 208
Künstliche Intelligenz, 287, 290
Kurvenscheibe, 202
KV-Diagramm, 96, 97

L

Lageregelung, 197, 217
Lagesollwert, 197
Lastenheft, 226
Laufmeldung, 112
Laufzeitfehler, 112
Laufzeitüberwachung, 111
Lebenszyklus, 222, 240
Leistungselektronikeinheit, 190
Lichtwellenleiter, 38
LIMIT, 60
LIMS, 281
Linearachse, 193
Lineargetriebe, 193
Linearinterpolation, 197
Logbuch, 238
Logikentwurf, 105
Logistic Execution System, 284, 285, 289
Loop-Check, 235
Losgröße 1, 282

M

Machine Learning, 291
Machine Vision, 291
Machine-to-Machine Kommunikation, 262
Machine-Vision-Systeme, 212
Manchestercodierung, 255
Manipulated Value, 116, 117
Manuell, MAN, 60, 113
Manufacturing Execution Systems, 272, 284
Markierung, 149
Maschinenkoordinatensystem, 206
Maschinenlaufzeit, 277
Masterachse, 202
Master-CPU, 245
Master-Slave-Verfahren, 39
Materialauftrag, 285
Materialbedarfsplanung, 284
Materialverbrauch, 277
Maximalgeschwindigkeit, 196
Maxterm, 95
MC_GearIn, 205
MC_MoveAbsolute, 193
MC_Power, 193
Mengenplanung, 283
Merkmalsextraktion, 214
MES, 281
Messumformer, 32
Methoden, 160, 161, 176, 182
Mikrocontroller, 18, 190
Mikroprozessor, 247
Mindsphere, 272, 275
Minterm, 94, 97
Mobiler Roboter, 220
mobiler Roboter, 220
Modbus, 39

MODES, 114
Modulare Lösung, 153
Modularisierung, 86
Modultests, 92, 235
Montagestationen, 186
MooN-Systeme, 245
Moore-Automat, 98
Motion-Control-Bausteine, 192
Motion-Control-System, 190
 Hardwareplattformen, 190
Motor, 62, 107
 2 Geschwindigkeitsstufen, 107, 130
 drehzahlveränderbar, 190
 Pulsweitenmodulation, 170
MQTT, 28, 269
Multiachsen, 192
Multi-Tasking, 57
Multi-User-Engineering, 230

N

Nachstellzeit, 123
Netzmatrix, 149
Netzwerkvariablen, 265
Neuronales Netz, 280, 281, 288
Niveauschalter, 42, 87
Nockenschalter, 203
Node-RED, 268
Normalform
 disjunktiv, 95
 konjunktiv, 95
NoSQL-Datenbanken, 276
Not-Aus-Schaltungen, 243

O

Objekt, 159, 162
Objektkoordinatensystem, 206
Objektorientierte Programmierung, 159, 301
 Vorteile, 182
Objektorientiertheit, 85
OPC-Client, 46, 267
OPC-Server, 45, 267
OPC-UA, 28, 266, 267, 287
Optik, 215
Optokoppler, 31
OT-Systeme, 22

P

PAC, 19
P-Anteil, 120
Parallele Werkstückhandhabung, 149
Parallelisierung, 186
Parallelverzweigung, 77, 146
Parallelzusammenführung, 146
Parametrierung, 236, 279
Patchmanagement, 259
PCE
 -Kategorie, 43
 -Kennzeichnung, 86
 -Stellen, 244
 -Stellenliste, 228
 -Verarbeitungsfunktion, 43
PC-Visualisierung, 46

- Performance Level, PL, 252
 - Persistent, 56
 - Petri-Netz, 145, 147, 151, 158, 187
 - Pflichtenheft, 227
 - Phase, 172
 - Pick-and-Place, 206
 - PID-Regler, 120, 170, 177
 - PI-Regler, 124
 - Plant-Asset-Management (PAM), 268, 279
 - Planung, 222
 - Platform as a Service (PaaS), 272
 - PLCopen, 192, 203, 206
 - PLCopenXML, 225
 - Plug-and-Play, 292
 - Point-to-Point, PTP, 209
 - Polling-Verfahren, 40
 - Polymorphismus, 171
 - Portalfräsmaschine, 201
 - Positioner, 34
 - Positionsermittlung, 215
 - Predictive Maintenance, 237, 279
 - Private Cloud, 275
 - Probability of Failure on Demand, PFD, 252
 - Probability of Failure per Hour, PFH, 252
 - Process Value, 69, 106, 116, 122
 - Produktionsauftrag, 284
 - Produktionsmenge, 277
 - Produktionsplanung und -steuerung, 268, 282
 - Produktionsprogrammplanung, 283
 - Produktionsprüfung, 236
 - Produktionssteuerung, 284
 - Produkt-Lifecycle-Management, 279
 - Produktqualität, 278
 - Produktsicherheit, 222, 243
 - Profibus, 39
 - Profibus PA, 255
 - ProfiNet, 41, 262
 - Interface, 54
 - IRT, 262
 - RT, 262
 - ProfiSafe, 248
 - Program Organization Units, 58
 - Programm, 162
 - Programmable Automation Controller, 213
 - Programmieraufwand, 182
 - Programmiergerät (PG), 25
 - Programmiersystem, 240
 - Programmspeicher, 30
 - Programmstrukturierung, 182
 - Projektierung, 221
 - Properties, 160, 161
 - Proportionalbeiwert, 123
 - Protokollierung, 236
 - Prozessablauf, 75, 144, 171
 - parallel, 147
 - Prozessanalyse, 142, 175
 - Prozessfehler, 109
 - Prozessgrafik, 42
 - Prozessleitsystem, 18, 46
 - Prozessmodell, 178, 224
 - Prozessphasen, 142
 - Prozessspezifikation, 226
 - Prozessverstärkung, 123
 - PTO-Ausgang, 35
 - PTP-Bewegung, 211
 - Publish-Subscribe, 269
 - Pulsausgabe-Baugruppen, 35
 - Pulstimer, 63
 - Pulsweitenmodulation, 131
 - Punkt-zu-Punkt-Bewegung, 210
 - PWM-Ausgang, 35
- Q**
- Qualifier, 76
 - Qualifizierung, 223
 - Qualitätskontrolle, 281
 - Qualitätsmanagementsystem, QMS, 268, 281
- R**
- R+I-Schema, 42
 - RAM, 24, 30, 56, 247
 - Randbedingung, 150, 158
 - Raspberry-Pi, 27
 - Raumbeleuchtung in Gebäuden, 184
 - Realisierung, 229
 - Redundanz, 245
 - passiv, 245
 - Regeldifferenz, 116
 - Regelgröße, 116
 - Regelkreis, 23
 - Regeln, 177
 - Regelung, 20, 21, 115
 - Regelventil, 34, 109, 122
 - Regler, 167, 180
 - kontinuierlich, 119
 - schaltend, 116
 - selbsteinstellend, 125
 - Reglerbaustein, 198
 - Reglerbetriebsarten, 118
 - Reglereinstellung, 123
 - automatisch, 127
 - Reglerparameter, 123, 198
 - Relais, 32
 - Relaisausgänge, 31
 - Relais-Schaltung, 17
 - Remote-I/O, 26, 38, 248
 - Reparaturschalter, 111
 - Repeater, 49, 263
 - Resolver, 206
 - Ressourcen, 52, 191
 - Restriktionen, 188
 - RETAIN, 56
 - Review, 228
 - Rezept, 181
 - Rezeptablauf, 277
 - Rezeptfahrweise, 173, 179, 184
 - Rezeptparameter, 173, 181, 285
 - Rezeptsteuerung, 183
 - Rezeptverwaltung, 181, 289
 - RIO, 53
 - Risiko, 251
 - Risikoanalyse, 251, 252, 260
 - Risikograf, 251, 260
 - Roboter, 151, 188, 216

Robotersteuerungen, 18, 206
Rohrleitungs- und Instrumentenschema, 42
Router, 264
RS-Flip-Flop, 61
Rückmeldung, 112
Rückwärtstransformation, 207, 208
Ruhestromprinzip, 74, 111

S

SADT-Methode, 174
Safety, 243
Safety Integrity Level (SIL), 251, 252
SCADA-Systeme, 45
SCARA-Roboter, 206, 211
Schadensausmaß, 251
Schaltfunktion, 94, 104
Schaltnetz, 94
Schaltungsentwurf, 96
Schaltwerk, 98
Schnelle Zählerbaugruppen, 34
Schnittstelle, 176
Schrittbaustein, 157
Schrittkette, 134, 135, 168, 171, 183, 189, 216
 Anhalten, 139
 Beenden, 139
 implizite Variablen, 138
 unerreichbar, 146
 unsicher, 146
Schrittmotor, 64, 170
Schrittvektor, 149
Schutzfunktionen, 109, 139
Schutzschalter, 110
Schwingungsanalyse, 126
Security, 221, 243, 268
Security Gateway, 258
Security Levels (SL), 256
SEL, 61
Selbsttests, 247
Sensoren, 106, 163
Sensorik, 106
Sequential Function Chart (SFC), 75, 133, 137
Sequenznummer, 249
Sercos, 191
Setpoint, SP, 117
SFCs, 155, 167
Sicherheit, 245
Sicherheitsanforderungen, 227
Sicherheitsgerichtete Steuerungen, SSPS, 243
Sicherheitszonen, 257
Signalliste, 228
Signalmodell, 225
Simulation, 90, 189, 198
Simulationsbausteine, 91, 232
Simulationsmodelle, 224
Single-Tasking, 57
Skalierbarkeit, 292
Slaveachse, 202
Slave-CPU, 245
Slot-SPS, 27
Smart-Camera, 213
Soft-SPS, 27
Software as a Service (SaaS), 272, 280
Software-in-the-Loop (SIL), 232
Softwaremodell, 51
Software-Quality, 51
Softwarestrukturierung, 85
Softwarestrukturplan, 228, 229
Sollwert, 20, 23, 116
Speicherkomparator, 247
Spezifikationsdatenbank, 239
Sprungantwort, 123
Sprünge, 146
SPS, 16, 290
 Schrank-SPS, 48
 Soft-SPS, 49
SPS-Lern-und-Übungsseite, 299
SPS-Programmierung, 80
 objektorientiert, 159
SPS-Software, 155
SQL-Datenbank, 276
SRIO, 248
Stellwert, 20, 23
Stellwertbegrenzung, 122
Step7, 25
Stereobildverarbeitung, 216
Steuerfunktionen, 143
Steuerkreis, 21, 23
Steuerrezept, 173, 181, 284, 285
Steuerung, 20
Steuerungskonfiguration, 52
STOP, 25
ST-Regler, 127
Stromregler, 198
Struktur, 78
Strukturierter Text (ST), 72
Subversion, 230
Supply Chain Management, 286
Swinging-Door-Algorithmus, 273
Switch, 262, 263
Systementwurf, 46
Systemspeicher, 30

T

Taktgenerator, 64
Taktschneideantrieb, 205
Tänzerwalze, 206
Tänzerwalzenregelung, 219
Target-Visualisierung, 46
Taskkonfiguration, 58
Tasks, 56
Taskzuordnung, 89
Taster, 62
Tätigkeiten, 186
TCP/IP-Protokoll, 262
Teamviewer, 240
TeKa-Anlage, 180, 278
Temperaturregelung, 132
Termin- und Kapazitätsplanung, 284
Testfahrt, 236
Thermoelement, 32
Time Sensitive Network (TSN), 263
Timer, 63, 104, 197
TLS-Protokoll, 221
Traceaufzeichnung, 43

Transistorausgänge, 31
Transitionen, 75, 133
Transitionsvektor, 149
Transport Layer Security (TLS), 258
Transportfahrzeug, 186
Trennverstärker, 36
Triggerstufe, 31
TSN, 266
Twin-Store, 291
TYP_2PT, 117
TYP_AIN, 69, 107, 166
TYP_AOUT, 109, 120, 123
TYP_BF, 172, 176
TYP_BIN, 74, 163
TYP_IDF1, 62, 71, 161
TYP_IDF2, 64, 165
TYP_PID, 120, 121, 166
TYP_PULSE, 67
TYP_SMOT, 65

U
Übergangsaktion, 104, 105
Übergangsbedingung, 104
Übertragungsmedien, 38
Überwachen, 21
Überwachungszeit, 249
Ultraschallsensor, 166
UML-Use-Case Diagramm, 85
UML-Zeitdiagramm, 100
Umrichter, 191, 198
Unified Modelling Language (UML), 83
Unsichere Kette, 146
URL, 270
Use-Case-Diagramm, 144
User Data Types, UDT, 78
User Datagram Protocol (UDP), 265
User-Requirements, 84

V
Variablen, 54
Ventile, 108
Ventiltypen, 108
Verbindungsprogrammierte Steuerungen, 17
Vererbung, 164, 182
Verfahrensfließbild, 226, 241
Verfahrenstechnik, 19
Verfügbarkeit, 245
Verfügbarkeitsanforderungen, 227
Verkehrssampel, 82
Verkehrskreuzung, 156
Verknüpfungslogik, 94
Verknüpfungssteuerungen, 83
Verriegelung, 110, 236
Verschlüsselung, 258
Vertikal-Knickarm-Roboter, 206
Verwaltungsschale, 225, 292
Vierleiterschaltung, 33, 48
Vierwegeventil, 109, 130
Virtual Private Network, VPN, 239
Virtualisierung, 291
Virtuelle Inbetriebnahme, 90, 224, 231
Visualisierung, 42

Visualisierungsframes, 233
V-Modell, 222, 235
Vor Ort, 113
Vorhaltzeit, 123
Vorranggraf, 186
Vorschubantrieb, 199
Vorwärtstransformation, 207
VPN/IPSec-Tunnel, 240

W
Wahrheitstabelle, 62, 94, 97, 105
WAMP-Server, 276
Waretransport, 285
Warenverwaltung, 78
Wärmetauscher, 237
Warteschritte, 188
Wartung und Instandhaltung, 237, 278
Wasserfahrt, 236
Watchdogschaltung, 247
Web-Client, 240
Web-Server, 270
Webvisualisierung, 46, 270, 288
Wendetangente, 123
Werkstück, 199
Werkzeug, 199
Werkzeugmagazin, 82
Werkzeugmaschine, 186, 199
White-Box-Test, 234
Wirkungslinien, 88

Z
Zähler, 65, 80
Zeitdiagramm, 134
Zeitredundanz, 246
Zeitreihen, 275, 277
Zenerbarrieren, 255
Zentralbaugruppe, 24
Zentrifuge, 158
Ziegler-Nichols, 126
Zirkularbewegung, 200
Zusatzlogik, 88, 236
Zusatz-Netzmatrix, 151, 152
Zusatzzustände, 151
Zustand, 98
Zustandsdiagramm, 103, 133
Zustandsgleichungen, 99, 152
Zustandsgraf, 113, 152
Zustandskodierung, 101
Zustandsübergangstabelle, 104
Zwei-/Vierleitertechnik, 36, 48
Zweileiterschaltung, 32, 48
Zweipunktregler, 116
Zykluszeit, 49, 57, 80, 121, 124, 190