

Matthias Seitz

Speicherprogrammierbare Steuerungen im Industrial IoT

Objektorientierter System- und Programmentwurf,
Motion Control, Sicherheit, Digital Engineering



6., aktualisierte Auflage

HANSER

Seitz
Speicherprogrammierbare Steuerungen im Industrial IoT



bleiben Sie auf dem Laufenden!

Hanser Newsletter informieren Sie regelmäßig über neue Bücher und Termine aus den verschiedenen Bereichen der Technik. Profitieren Sie auch von Gewinnspielen und exklusiven Leseproben. Gleich anmelden unter

www.hanser-fachbuch.de/newsletter

Matthias Seitz

Speicherprogrammierbare Steuerungen im Industrial IoT

Objektorientierter System- und Programmentwurf,
Motion Control, Sicherheit, Digital Engineering

6., aktualisierte Auflage

HANSER

Über den Autor:

Prof. Dr.-Ing. Matthias Seitz vertritt das Fachgebiet Elektronische Steuerungstechnik an der Hochschule Mannheim.



Print-ISBN: 978-3-446-48243-2

E-Book-ISBN: 978-3-446-48242-5

Alle in diesem Werk enthaltenen Informationen, Verfahren und Darstellungen wurden zum Zeitpunkt der Veröffentlichung nach bestem Wissen zusammengestellt. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund sind die im vorliegenden Werk enthaltenen Informationen für Autor:innen, Herausgeber:innen und Verlag mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor:innen, Herausgeber:innen und Verlag übernehmen infolgedessen keine Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Weise aus der Benutzung dieser Informationen – oder Teilen davon – entsteht. Ebenso wenig übernehmen Autor:innen, Herausgeber:innen und Verlag die Gewähr dafür, dass die beschriebenen Verfahren usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt also auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benützt werden dürften.

Die endgültige Entscheidung über die Eignung der Informationen für die vorgesehene Verwendung in einer bestimmten Anwendung liegt in der alleinigen Verantwortung des Nutzers.

Bibliografische Information der Deutschen Nationalbibliothek:

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

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Werkes, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Einwilligung des Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder einem anderen Verfahren), auch nicht für Zwecke der Unterrichtsgestaltung – mit Ausnahme der in den §§ 53, 54 UrhG genannten Sonderfälle –, reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Wir behalten uns auch eine Nutzung des Werks für Zwecke des Text- und Data Mining nach § 44b UrhG ausdrücklich vor.

© 2025 Carl Hanser Verlag GmbH & Co. KG, München

www.hanser-fachbuch.de

Lektorat: Dipl.-Ing. Natalia Silakova-Herzberg

Coverkonzept: Marc Müller-Bremer, www.rebranding.de, München

Covergestaltung: Thomas West

Titelmotiv: © Thomas West / Firefly-genAI

Satz: Matthias Seitz

Druck: Elanders Waiblingen GmbH, Waiblingen

Printed in Germany

Vorwort

Die vorliegende 6. Auflage behandelt die Rolle speicherprogrammierbarer Steuerungen (SPS) im Industrial Internet of Things (IoT). Hierfür wird einigen Trends nachgegangen, wie z. B.

- Nutzung *künstlicher Intelligenz* in der Automatisierungstechnik,
- *Virtualisierung* der SPS als Edge Controller und als übergeordnetes Automatisierungsgerät für Logik, Motion, Bildverarbeitung, Safety etc.,
- *OPC-UA* als standardisierter Datenaustausch von der Cloud bis ins Feld,
- Anwendung von *Asset Administration Shells* und digitalen Zwillingen beim Engineering der SPS,
- *Remote Engineering* mit der notwendigen IT-Security.

Schwerpunkt des Lehrbuchs ist aber nach wie vor die Methodik für den *System- und Programmwurf* speicherprogrammierbarer Steuerungen. Dabei wird nur am Rande auf die Programmiersysteme einzelner Hersteller eingegangen, sondern im Mittelpunkt steht die *Entwurfsmethodik* für eine transparente und flexibel einsetzbare SPS-Software durch:

- UML-Diagramme zur *Softwarestrukturierung*,
- *Entwurfsverfahren* aus der Informatik für das SPS-Software-Engineering,
- eine modulare oder objektorientierte *SPS-Programmierung*.

Es wird ein Leitfaden vorgestellt, wie typische Aufgaben der Fabrik- und Prozessautomation mit SPSen gelöst werden können. Dabei wird die Einbindung von SPSen in die digitale Fabrik mit Robotern und autonomen Systemen ebenso beschrieben wie die Nutzung der von der SPS gesammelten Prozessdaten im Industrial IoT.

Außerdem wird für die 6. Auflage eine überarbeitete „SPS-Lern-und-Übungsseite“ unter www.seitz.et.hs-mannheim.de bereitgestellt mit:

- Fragen und Antworten zum Einsatz speicherprogrammierbarer Steuerungen (SPS),
- Beispielprogramme mit dem SPS-Programmiersystemen Codesys oder Step7,
- Übungsaufgaben mit Lösungen,
- Links zu Downloads SPS-relevanter Software,
- Videos als Bedienungsanleitung der SPS-Software,
- Bibliotheken mit Funktionsbausteinen für Automatisierungsprojekte.

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 (s. Anhang).

In diesem Zusammenhang bedanke ich mich bei den Firmen CODESYS, ABB, Siemens und Wonderware für die Bereitstellung von Software und Bildmaterial. Frau N. Silakova und Frau C. 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 Frau Dr. R. Gao Hoffman für die Zusammenarbeit in den zahlreichen Laborprojekten am Institut für industrielle Automatisierungssysteme.

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, September 2024

Matthias Seitz

Inhalt

Verzeichnis der Abkürzungen und Symbole	13
1 Einführung.....	16
1.1 Entwicklung der industriellen Automatisierungstechnik	16
1.2 Automatisierungssysteme	18
1.3 Aufgaben im Industrial IoT	20
1.3.1 Messen, Steuern, Regeln und Überwachen	20
1.3.2 Auswerten und Planen im Industrial Internet of Things	22
1.3.3 Virtualisierung und Testen mit digitalem Zwilling	23
Übungen zu Kapitel 1	23
2 Aufbau industrieller Steuerungen.....	25
2.1 SPS-Aufbau	25
2.1.1 Zentralbaugruppe.....	25
2.1.2 Peripheriebaugruppen	26
2.1.3 Programmiergerät	26
2.1.4 Human Machine Interface	27
2.1.5 Vernetzte Automatisierungssysteme	27
2.2 SPS-Arten und IoT-Geräte	28
2.2.1 Hardware-SPS	28
2.2.2 Slot-SPS	28
2.2.3 Soft-SPS	28
2.2.4 Edge Controller und IoT-Gateways	29
2.2.5 Virtuelle SPS und Cloud-SPS	30
2.2.6 Hochverfügbare und fehlersichere SPSen	31
2.3 Informationsverarbeitung in der SPS	32
2.4 Ein-/Ausgangsbaugruppen der SPS	33
2.4.1 Digitale Eingangsbaugruppen	33
2.4.2 Digitale Ausgangsbaugruppen	34
2.4.3 Analoge Eingangsbaugruppen	35
2.4.4 Analoge Ausgangsbaugruppen.....	37
2.4.5 Schnelle Zählerbaugruppen	37
2.4.6 Pulsausgabe-Baugruppen.....	38

2.5	Ankopplung der Sensoren und Aktoren	39
2.5.1	Zwei-/Vierleitertechnik	39
2.5.2	Busankopplung der Feldgeräte	40
2.5.3	Intelligenter Feldverteiler (Remote-I/O)	41
2.6	Industrielle Feldbusysteme.....	41
2.6.1	Übertragungsmedien	41
2.6.2	Datenübertragung mit dem Master-Slave-Verfahren	42
2.6.3	Ethernet-basierte Feldbusse	44
2.7	Bedienen und Beobachten	45
2.7.1	Elemente der Prozessvisualisierung	46
2.7.2	Datenaustausch zwischen HMI und SPS	48
2.7.3	Ankopplung der Prozessvisualisierung an SPSen	51
2.7.4	Prozessleitsysteme	51
	Übungen zu Kapitel 2	52
3	SPS-Programmierung nach IEC 61131	56
3.1	Softwaremodell.....	56
3.1.1	Steuerungskonfiguration und Ressourcen	57
3.1.2	Variablen	59
3.1.3	Tasks	61
3.1.4	Programmorganisationseinheiten	64
3.1.5	Funktionen	65
3.1.6	Funktionsbausteine	66
3.1.7	Instanziierung von Funktionsbausteinen in Programmen	72
3.2	Kommunikationsmodell	74
3.2.1	Datenaustausch innerhalb eines Programms	74
3.2.2	Datenaustausch zwischen Programmen	75
3.2.3	Datenaustausch zwischen SPSen	76
3.3	Programmiermodell	78
3.3.1	SPS-Programmiersprachen	78
3.3.2	Anweisungsliste (AWL).....	78
3.3.3	Strukturierter Text (ST)	79
3.3.4	Funktionsbausteinsprache (FBS/FUP).....	80
3.3.5	Kontaktplan (KOP)	81
3.3.6	Ablaufsprache (AS)	82
3.3.7	Anwender-Datentypen	85
3.4	KI-gestützte SPS-Programmierung	87
	Übungen zu Kapitel 3	89

4	Entwurf kontinuierlicher Steuerungen	92
4.1	SPS-Software-Engineering	92
4.1.1	Analyse der User Requirements	93
4.1.2	Objektorientierte Softwarestrukturierung.....	94
4.1.3	Entwurf der Feldgeräteklassen	96
4.1.4	Entwurf der Ansteuerprogramme	97
4.1.5	Implementierung in der SPS	99
4.1.6	Simulation und virtuelle Inbetriebnahme	100
4.2	Entwurf der Verknüpfungslogik	103
4.2.1	Entwurf von Schaltnetzen	104
4.2.2	Entwurf von Schaltwerken	108
4.3	Ansteuerung der Sensorik und Aktorik	116
4.3.1	Funktionsbausteine zum Einlesen von Sensordaten	116
4.3.2	Funktionsbausteine zum Ansteuern von Motoren	117
4.3.3	Funktionsbausteine zum Ansteuern von Ventilen	118
4.3.4	Schutzfunktionen	119
4.3.5	Betriebsarten.....	122
4.4	Die SPS als Regler	125
4.4.1	Schaltende Regler	126
4.4.2	Reglerbetriebsarten	129
4.4.3	Kontinuierliche Regler	130
4.4.4	Selbsteinstellende Regler	135
	Übungen zu Kapitel 4	139
5	Entwurf von Ablaufsteuerungen	144
5.1	Entwurf aus Zustandsfolge	144
5.2	Entwurf aus zeitlicher Abfolge	145
5.3	Modellierung durch Aktivitätsdiagramm.....	146
5.4	Programmierung industrieller Abläufe	147
5.4.1	Verknüpfung von CFCs und SFCs	148
5.4.2	Schutzfunktionen und Betriebsarten.....	151
5.5	Ablaufmodellierung durch projektneutrale Grundfunktionen	154
5.5.1	Prozessanalyse.....	154
5.5.2	Entwurf anlagenneutraler Grundfunktionen	155
5.5.3	Zusammensetzung der Ablaufsteuerung	157
5.6	Entwurf paralleler Prozessabläufe	158
5.6.1	Analyse der Steuerbarkeit paralleler Abläufe	158
5.6.2	Modellierung paralleler Prozessabläufe mit Petri-Netzen	159

5.6.3	Algebraischer Entwurf zur Koordination paralleler Prozesse	163
5.6.4	Programmwurf aus Petri-Netzen	164
	Übungen zu Kapitel 5	166

6 Objektorientierte SPS-Programmierung 171

6.1	Klassen und Objekte	171
6.2	Interfaces als abstrakte Klassen.....	172
6.3	Methoden und Eigenschaften	172
6.4	Vererbung	177
6.5	Objektorientierte Ansteuerung der Feldgeräte	180
6.5.1	Ablaufsteuerungen mit Methoden und Properties	180
6.5.2	Polymorphe Ansteuerung durch Interfaces.....	182
6.5.3	Anlagenneutrale Ablaufsteuerung	184
6.6	Flexible Ablaufsteuerung durch Rezeptfahrweise	185
6.6.1	Entwurf von Rezeptsteuerungen	185
6.6.2	Objektorientierte Programmierung des Prozessmodells	191
6.6.3	Objektorientierte Programmierung des Anlagenmodells	193
6.6.4	Ausführung von Steuerrezepten	194
	Übungen zu Kapitel 6	195

7 Motion Control in der Fabrikautomatisierung 198

7.1	Entwurfsmethodik zur Steuerung von Fertigungsabläufen	198
7.2	Motion-Control in der SPS	203
7.2.1	Aufbau von Motion-Control-Systemen	203
7.2.2	Motion-Control durch SPS-Programmierung nach PLCopen	205
7.3	Steuerung einer Bewegungsachse	206
7.3.1	Interpolation	208
7.3.2	Lageregelung	211
7.4	Steuerung von Werkzeugmaschinen	213
7.4.1	Bahnplanung durch CNC-Programmierung	214
7.4.2	Bewegungsvorgaben durch Kurvenscheiben	216
7.4.3	Elektronisches Getriebe	218
7.5	Robotersteuerungen	220
7.5.1	Koordinatentransformation	221
7.5.2	Programmierung von Bewegungsabläufen	223
7.6	Bildverarbeitung in der SPS	226
7.6.1	Machine-Vision-Systeme	226
7.6.2	Programmierung der Bildverarbeitung nach IEC 61131	227
7.6.3	Bestimmung der 3D-Zielposition	228
	Übungen zu Kapitel 7	231

8	Industrial IoT in der Prozessautomatisierung	236
8.1	Horizontale Vernetzung der Cyber Physical Systems	237
8.1.1	Homogenes Netzwerk mit Industrial Ethernet	237
8.1.2	Werkzeuge zur Netzwerkintegration	238
8.1.3	Interoperabilität durch OPC-UA	239
8.2	Vertikale Vernetzung in die Cloud	241
8.2.1	Industrial IoT-Netzwerk mit OPC-UA	242
8.2.2	MQTT zur Anbindung von Kleingeräten an die Cloud	243
8.2.3	Plattformunabhängige Webvisualisierung	244
8.3	Betriebsdatenauswertung und Cloud Services	245
8.3.1	Betriebsdatenerfassung	247
8.3.2	Betriebsdatenauswertung.....	250
8.3.3	Asset Management	252
8.3.4	Qualitätsmanagement	254
8.4	Künstliche Intelligenz für die Anlagenautomatisierung	256
8.4.1	Deep Learning in der Cloud	256
8.4.2	Shallow Learning in der SPS	257
8.5	Produktionsplanung und -steuerung	260
8.5.1	Produktionsplanung durch Enterprise Resource Planning	260
8.5.2	Produktionssteuerung in Manufacturing Execution Systems	262
8.5.3	Logistic Execution Systems	263
8.5.4	Supply Chain Management.....	264
	Übungen zu Kapitel 8	266
9	Safety und Security im Industrial IoT	268
9.1	Funktionale Sicherheit	268
9.1.1	Sicherheitsgerichtete Steuerungen	268
9.1.2	Mehrkanalige SSPSen	270
9.1.3	Einkanalige SSPSen.....	271
9.1.4	Selbsttests in SSPSen	272
9.1.5	Sicherheitsgerichtete Feldbussysteme	274
9.1.6	Gefahren- und Risikoanalyse	274
9.2	Steuerungen für explosionsgefährdete Betriebe	279
9.2.1	Eigensichere Ansteuerung durch die SPS	279
9.2.2	Eigensichere Feldbussysteme	280
9.3	IT-Security	281
9.3.1	Analyse der Schwachstellen	281
9.3.2	Schutzmaßnahmen.....	282
	Übungen zu Kapitel 9	285

10 Digital Engineering zuverlässiger Steuerungen	287
10.1 Projektierung in der Cloud.....	287
10.1.1 Computer-Aided-Engineering-Systeme	287
10.1.2 Gute Engineering-Praxis	288
10.1.3 Verwaltungsschale und digitaler Zwilling	290
10.2 Planung der Automatisierung.....	293
10.2.1 Prozess- und Anlagenplanung	293
10.2.2 Automatisierungstechnisches Lastenheft	293
10.2.3 Automatisierungstechnisches Pflichtenheft	294
10.3 Realisierung des Automatisierungssystems	297
10.3.1 Automatische Codegenerierung.....	297
10.3.2 Cloud Engineering	299
10.3.3 Virtuelle Inbetriebnahme	300
10.4 Inbetriebnahme und Qualifizierung.....	303
10.4.1 Installationsprüfung	303
10.4.2 Funktionsprüfung	304
10.4.3 Produktionsprüfung	305
10.5 Wartung und Instandhaltung	305
10.5.1 Predictive Maintenance mithilfe des digitalen Zwillings.....	305
10.5.2 Change Management	307
10.5.3 Fernwartung	307
Übungen zu Kapitel 10	310
11 Die Zukunft der SPS im Industrial IoT	312
Anhang	316
A SPS-Lern-und-Übungsseite	316
B Funktionsbaustein-Bibliotheken	317
Literaturverzeichnis	319
Index	326

Abkürzungen und Symbole

-o	Negation	CRC	Cyclic Redundancy Check
&, ^	UND-Verknüpfung	CSMA/ CD	Carrier Sense Multiple Access/ Collision Detection
$\geq 1, \vee$	ODER-Verknüpfung	CSI	Camera Serial Interface
1oo2	eins von zwei System	CTD	Count-Down-Zähler
a	Beschleunigung	CTU	Count-Up-Zähler
\vec{a}	Vektor der Achsstellungen	CTUD	Count-Up-and-Down-Zähler
\vec{a}^T	transponierter Vektor	CV	Counted Value
A	Analysenmessung	ΔC	Drehwinkel um neue z-Achse
\bar{A}	Ausgangsschaltnetz	d	1. Durchmesser, 2. Schaltstufe beim Zweipunktregler
AA	analoger Ausgang	D	1. Delay, 2. Differenzierglied
AAS	Asset Administration Shell	D/A	Digital-/Analog
AC	Alternating Current	DB	Datenbank
ACS	Axis Coordinate System	DC	Direct Current (Gleichstrom)
A/D	Analog-/Digital	Dexpi	Data Exchange in the Process Industry
A_D	Amplitude der Dauerschwingung	DKT	Direct Kinematic Transformation
AE	analoger Eingang	DMZ	Demilitarisierte Zone
AH	Alarm bei oberem Grenzwert	DNF	disjunktive Normalform
AIN	Analog Input	DP	dezentrale Peripherie
AL	Alarm bei unterem Grenzwert	DPT	Direct Perspective Transformation
AOUT	Analog Output	DT#	Date and Time
APL	Advanced Physical Layer	DTM	Device Type Manager
AS	Ablaufsprache	\bar{E}	Eingangsschaltnetz
ASI	Aktor-Sensor-Interface	E/A	Ein-/Ausgang
AUT	Betriebsart Automatik	EDD	Electronic Device Description
AWL	Anweisungsliste	EEPROM	Electrically Erasable Programmable Read Only Memory
ΔA	Drehwinkel um z-Achse	EMV	Elektromagnetische Verträglichkeit
\vec{b}	Randbedingungsvektor	ϵ	1. Hystereseschwelle 2. Lernrate
B	Byte	ERP	Enterprise Resource Planning
[B]	Basiskoodinatensystem	ET	Elapsed Time
BA	binärer Ausgang	ETA	Event Tree Analysis
BDA	Betriebsdatenauswertung	EVA	Einlesen, Verarbeiten, Ausgeben
BDE	Betriebsdatenerfassung	Ex	Explosionsschutz
BE	binärer Eingang	EXT	Betriebsart externe Sollwertvorgabe
BF	Basic Function	f	1. Funktion, 2. Brennweite
BIN	Binary Input	F	1. Flow, 2. Fuse
b_k	Schwellwert Neuron	FAT	Factory Acceptance Test
b_x, b_y	Bildkoordinaten in mm	FB, f_B	Function Block
ΔB	Drehwinkel um y-Achse	FBS	Funktionsbausteinsprache
C	Kapazität eines Kondensators	FC, f_C	Function Code (Funktion)
c	Regler	F-CPU	fehlersichere CPU
\bar{c}	spezifische Wärmekapazität	FIC	Durchflussregler
[C]	Kamerakoodinatensystem	FQI	Durchflussmengen-zähler
CAE	Computer Aided Engineering	FTA	Fault Tree Analysis
CAEX	CAE Exchange	FTP	File Transfer Protocol
CAM	1. Computer Aided Manufacturing 2. Kurvenscheibe	FUP	Funktionsplan
CFC	Continuous Function Chart	G	Gauging
CPS	Cyber Physical System	GAMP	Good Automated Manufacturing Practice
CPU	Central Processing Unit		
CNC	Computerized Numerical Control		

G-Code	geometrische Bewegungsvorgabe	LIS	Level Indication Switching
GF	Grundfunktion	LOC	Betriebsart vor Ort (local)
GS	binärer Endschalter	LS	binärer Niveauschalter
GIC	Positionsregler	LSH	Überlaufabschalter
GIS	Gauging Indication	LT	lower than
GNVL	Global Network Variable List	LWL	Lichtwellenleiter
GOP	Grundoperation	m	1. Masse, 2. Anzahl von Nutzdatenbytes
GSD	Geräte-Stammdatei	m_i	Minterme
GT	greater than	M	Motor
h	Höhe	\mathcal{M}	erlerntes Prozessmodell
H	Auftrittshäufigkeit eines Schadens	M_i	Maxterme
\tilde{H}	Halteglieder im Moore-Automaten	MAN	Betriebsart Manuell
HAZOP	Hazards and Operability	MC, MCS	Motion Control (System)
H-CPU	hochverfügbare CPU	MES	Manufacturing Execution System
HIL	Hardware in the Loop	MCS	Machine Coordinate System
HMI	Human Machine Interface	μP	Mikroprozessor
HS	Handschalter	MQTT	Message Queuing Telemetry Transport
HTML	Hyper Text Markup Language	M2M	Machine-to-Machine-Kommunikation
HTTP	Hyper Text Transmission Protocol	MTP	Module Type Package
HW	Hardware	MTTF	Mean Time to Failure
I	1. elektrische Stromstärke, 2. Input 3. Integrier-Glied	MV	Manipulating Value (Stellwert)
IaaS	Infrastructure as a Service	n	1. Drehzahl, 2. Anzahl
IDF	Individual Drive Function	N	Petri-Netzmatrix
i. d. R.	in der Regel	N	nicht speichernd
IEC	International Electrotechnical Commission	NC	geregelter Antrieb
IKT	Inverse Kinematic Transformation	NS	Ein/Aus-Motor
INT	1. Datentyp Integer 2. Betriebsart intern	\vec{o}	Objektpositionsvektor
I/O	Input-/Output	ω	Winkelgeschwindigkeit
IIoT, IoT	(Industrial) Internet of Things	OOP	Objektorientierte Programmierung
IP	Internet Protocol	OPC-UA	Open Platform Communication Unified Architecture
IPT	Inverse Perspective Transformation	OT	Operational Technology
IPSec	Internet Protocol Security	P	Pressure (Druck)
IRT	Isochronous Real Time	PaaS	Platform as a Service
IT	Informationstechnik	PSH	Überdruckabschalter
k	diskreter Zeitpunkt	P	Power (elektrische Leistung)
K	Prozessverstärkung x_∞/y_∞	PA	Process Automation
K_I	Verstärkungsfaktor integrierender Prozess	PAC	Programmable Automation Controller
K_P	Verstärkungsfaktor des Reglers	PAM	Plant Asset Management
KNF	konjunktive Normalform	PCE	Process Control Engineering
KOP	Kontaktplan	PCS	Programmer's Coordinate System
KV	Karnaugh-Veitch-Diagramm	PEA	Process Equipment Assembly
KI	Künstliche Intelligenz	PFD	Probability of Failure on Demand
Λ	Komplexität einer Schaltung	PFH	Probability of Failure per Hour
L	1. Level (Füllstandsmessung), 2. limited, 3. Anzahl Zustände	PG	Programmiergerät
L	Randbedingungsmatrix	PID	Proportional-Integral-Differenzial-Glied
L1, L2, L3	Phasen der Drehstromversorgung	PLC	Programmable Logic Controller
LAN	Local Area Network	PLC_PRG	Hauptprogramm in CoDeSys
l	1. Länge, 2. Liter	PLM	Product Lifecycle Management
λ	1. Faktor in mm/Pixel, 2. Ausfallrate	PLS	Prozessleitsystem
LES	Logistic Execution System	PL	Performance Level
LIMS	Labordateninformationssystem	POL	Process Orchestration Layer
LIC	Füllstandsregler	POU	Program Organization Unit
		PPS	Produktionsplanung und -steuerung
		PS	Power Supply
		PT	Preset Time
		PtP	Point-to-Point Bewegung

PT1, PT2	Verzögerungsglied 1./2. Ordnung	TLS	Transport Layer Security
PTO	Pulstrain Output	TOF	Ausschaltverzögerung
PTt	Proportionalglied mit Totzeit	TON	Einschaltverzögerung
PV	Process Value (Istwert)	TP	Puls-Timer
PV	Preset Value	TS	Temperaturschalter
PWM	Pulsweitenmodulation	TSN	Time-Sensitive Networking
ODBC	Open Database Connectivity	T_0	Abtastzeit
Q	Wärmemenge	T_A	Anforderungszeit
Q	Ausgang	T_D	Differenzialzeit
QMS	Qualitätsmanagementsystem	T_g	Ausgleichszeit
R	ohmscher Widerstand	T_I	Integralzeit
R	rücksetzend	t_{IPO}	Interpolationszeit
[R]	Roboterkoordinatensystem	T_N	Nachstellzeit
RAM	Random Access Memory	T_P	Periodendauer
RC	Robot Control	T_u	Verzugszeit
REF	Reference Value	T_V	Vorhaltezeit
ρ	Dichte	TYP	Typical-Baustein
R&I	Rohrleitungs- und Instrumentenfließbild	u	1. Eingangsvariable, 2. Spalten-nummer in Bild
RIO	Remote I/O	\vec{u}	Vektor der Eingangsvariablen
RS	rücksetzdominantes Flip-Flop	U	elektrische Spannung
s	1. Weg, 2. Sekunde	UDP	User Datagram Protocol
\vec{s}	Schrittvektor	UDT	User Datagram Type
S	1. Speed, 2. Schaltvorgang (Switch)	UML	Unified Modeling Language
S	setzend	URL/URI	Universal Resource Locator/Identifier
SaaS	Software as a Service	USV	unterbrechungsfreie Spannungsversorgung
SADT	Structure Analysis and Design Technique	US, UC	in Steuerung programmierte Schaltung/Regelung
SAT	Site Acceptance Test	U	Reglerbaustein
SCM	Supply Chain Management	v	1. Geschwindigkeit, 2. Zeilennummer in Bild
SEL	Funktionsbaustein zur Auswahl	\vec{v}	Vektor der Ausgangsvariablen
SFC	Sequential Function Chart	V	Volumen
SH	Abschaltung bei oberem Grenzwert	VAR	Variable
SIL	1. Safety Integrity Level, 2. Software in the Loop	VBA	Visual Basic for Applications
SIM	Simulationsbaustein	V-Modell	Vorgehensmodell beim Engineering
SL	Security Level	VO	vor Ort
SL	Abschaltung bei unterem Grenzwert	VPN	Virtual Private Network
SMC	SoftMotion Control	VPS	verbindungsprogrammierte Steuerung
SO	binäre Schaltmeldung	w	Sollwert einer Steuerung
SP	Set Point (Sollwert)	W	Arbeit, Energie
SPS	speicherprogrammierbare Steuerung	[W]	Weltkoordinatensystem
SQL	Structured Query Language	w_k	Gewichtswert eines Neurons
SSPS	sicherheitsgerichtete SPS	WAMP	Windows Apache MySQL PHP
SR	setzdominantes Flip-Flop	WLAN	Wireless LAN
SRIO	sicherheitsgerichtete RIO	x	Messgröße, Istwert
SSL	Secure Sockets Layer	x_W	Eingangsdatenwort
ST, SCL	Strukturierter Text	XML	Extensible Markup Language
SW	Software	y	1. Stellwert, 2. Ventilöffnungsgrad
sW	Warteschritt	YC	Regelventil
$t, t\#$	Zeit	YS	Auf/Zu-Ventil
\vec{t}	Transitionsvektor	y_W	Ausgangsdatenwort
T	Temperatur	z	Zustandsgröße
TCP	Transmission Control Protocol	Z	Sicherheitsschaltung
TCS	Tool Coordinate System		
TeKa	Anlage für Tee und Kaffee		
TI	Temperature Indication		
TIC	Temperaturregler		
TIPP	Betriebsart Tippen für Ablaufketten		

1

Einführung

Eine speicherprogrammierbare Steuerung (SPS) ist ein seit Jahrzehnten bewährtes Automatisierungsgerät, das millionenfach in Produktionsbetrieben eingesetzt wird. Sie ist ein industrieller Rechner mit einfachen Schnittstellen zu Sensoren und Aktoren, die in eine Maschine oder Anlage eingebaut sind. Durch die Ansteuerung dieser Sensoren und Aktoren ermöglicht es die SPS, dass Produktionsprozesse automatisiert ablaufen können.

Eine SPS wie in Bild 1.1 besteht aus einer Central Processing Unit (CPU), die SPS-Programme ausführt, und aus Ein-/Ausgangsbaugruppen zum Einlesen von Sensordaten und Ansteuerung von Aktoren, wie etwa Motoren oder Ventilen. Die Sensoren und Aktoren können z. B. durch eine Ethernet- oder Profibusverbindung oder über einzelne Gleichstromkreise an die Baugruppen angeschlossen werden. Der Aufbau von SPS-basierten Automatisierungssystemen wird ausführlich in Kapitel 2 beschrieben.

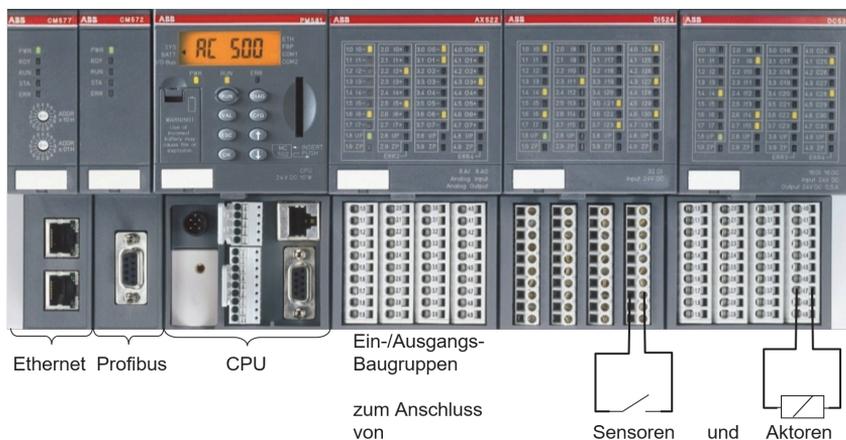


Bild 1.1 Speicherprogrammierbare Steuerung vom Typ AC500 der Fa. ABB mit CPU, Ein-/Ausgabebaugruppen und Feldbus-Schnittstellen zum Anschluss von Sensoren und Aktoren [2]

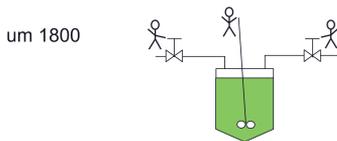
1.1 Entwicklung der industriellen Automatisierungstechnik

Die historische Entwicklung der industriellen Produktion durchlief bisher vier Etappen: In der ersten industriellen Revolution wurden mechanische Geräte eingeführt, um wie in Bild 1.2 z. B. Ventile und Rührwerke von Hand zu bedienen. Durch die Elektrifizierung konnten diese Ge-

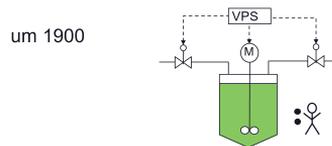
räte in der zweiten industriellen Revolution mit Antrieben und einfachen Relaisschaltungen elektrisch angesteuert werden. Die Erfindung der SPS läutete in der dritten industriellen Revolution das Zeitalter der Automatisierung ein. Dadurch war es möglich, ganze Produktionsabläufe zu programmieren und automatisch ausführen zu lassen.

Nun erleben wir in der vierten industriellen Revolution, dass Maschinen und Anlagen durch die SPS mit dem Internet verbunden werden und ihre Daten zentral in der Cloud abspeichern. Die dadurch entstehende große Datenmenge (Big Data) kann z. B. mit Methoden der künstlichen Intelligenz (KI) ausgewertet werden, um frühzeitig problematische Prozesssituationen zu erkennen und Gegenmaßnahmen zu ergreifen [8, 14, 117].

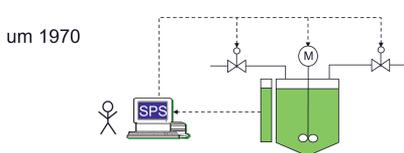
1. Industrielle Revolution *Mechanisierung*



2. Industrielle Revolution *Elektrifizierung*



3. Industrielle Revolution: *Automatisierung*



4. Industrielle Revolution *Internet of Things*

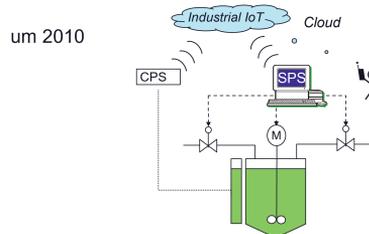


Bild 1.2 Die vier industriellen Revolutionen entlasten den Menschen zunehmend von manuellen und kleinteiligen Aufgaben

Vor der Erfindung der SPS konnte eine Verknüpfungslogik nur als verbindungsprogrammierte Steuerungen (VPS) mithilfe von Relais-Schaltungen wie in Bild 1.3 realisiert werden. Ein Relais besteht aus einer Spule und einem Schalter, der vom Magnetfeld der Spule angezogen oder abgestoßen wird und dadurch einen Stromkreis schließen oder öffnen kann.

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 jederzeit verändert werden kann.

Beispiel 1.1: Steuerung eines Rührwerks mit VPS und SPS

Das Koaxialrührwerk in Bild 1.3 soll verschiedene Substanzen in einem Behälter durchmischen. 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 und öffnet den rechten Zweig. Springt der Taster T1 durch seine Rückstellfeder wieder auf, wird A1 trotzdem weiter mit Strom versorgt.

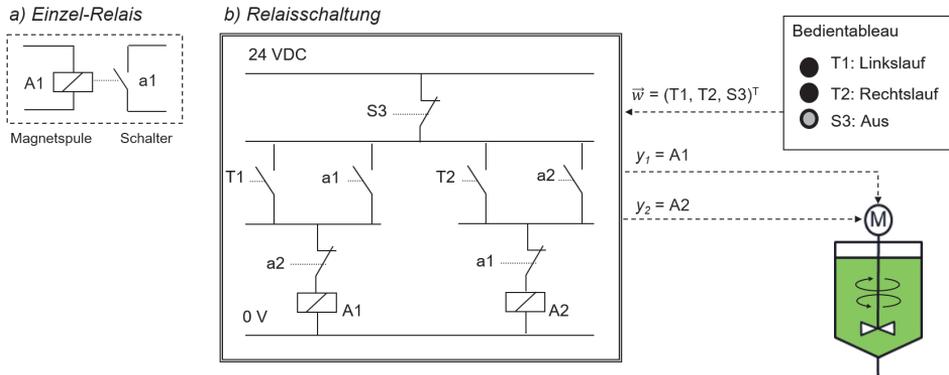


Bild 1.3 Verbindungsprogrammierte Steuerung (VPS) zur Ansteuerung eines Motors mit zwei Drehrichtungen, die gegenseitig verriegelt sind

Da a1 aber den rechten Stromzweig vor dem Relais A2 öffnet, kann der Rechtslauf durch Betätigen des Tasters T2 nicht ausgelöst werden. 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 angewählt werden.

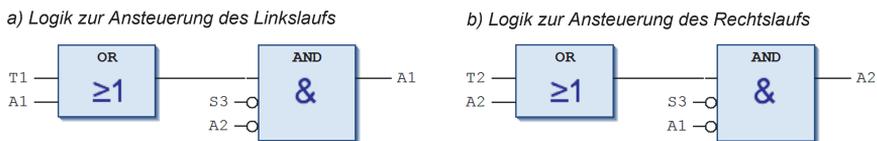


Bild 1.4 Ansteuerung des Antriebs mit zwei Drehrichtungen in einem SPS-Programm

In einer SPS wird das Programm direkt als Logik-Schaltplan wie in Bild 1.4 erstellt, 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. Die Software zu diesem Beispiel kann auf der SPS-Lern- und Übungsseite getestet werden.

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, die die Programmierung und die Prozessführung [47] vereinfachen.

Werkzeugmaschinen werden meist mit CNC-Steuerungen (Computerized Numerical Controls) automatisiert, weil sie Motoren mit sehr schnellen Zykluszeiten regeln und ihre Bewegungen synchron koordinieren können. Die Ansteuerung von Industrierobotern erfolgt meist durch herstellerspezifische *Robotersteuerungen* (Robot Controls, RC). Eine preiswerte und energiesparende Automatisierung von kleineren Anwendungen kann mit *Mikrocontrollern* realisiert werden. 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 industriellen Automatisierungssysteme können mit SPS-Programmiersprachen wie Funktionsbausteinsprache (FUP), Strukturierter Text (ST), Ablaufsprache (AS), Anweisungsliste (AWL) und Kontaktplan (KOP) nach IEC 61131 programmiert werden, was in den Kapiteln 3–6 ausführlich behandelt wird. Somit wachsen die verschiedenen Automatisierungsaufgaben der numerischen und binären Steuerung und Regelung sowie Sicherheitsfunktionen und Funktionen zur Prozessbedienung und -beobachtung unter dem Dach der IEC 61131 zusammen (s. Bild 1.5).

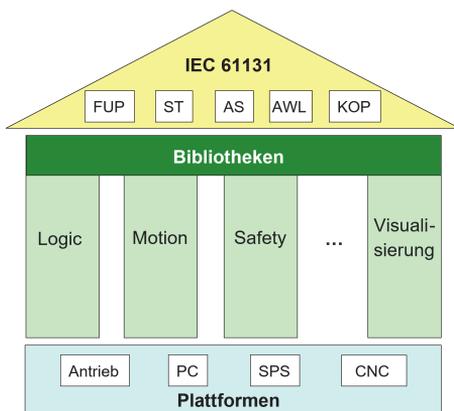


Bild 1.5 Klassische Logikschaltungen wachsen mit Motion- und Safety-Funktionen unter dem Dach der IEC 61131 zusammen. Die hardwareunabhängigen Programmiersprachen ermöglichen die Implementierung auf unterschiedlichen Plattformen [132]

Was bisher in getrennten Systemen programmiert und implementiert war, wird nun in einem standardisierten Programmiersystem entwickelt und kann auf unterschiedlichen Hardwareplattformen implementiert werden. Deshalb verwendet man statt der englischen Bezeichnung Programmable Logic Controller (PLC) nun häufig auch die Abkürzung PAC für Programmable Automation Controller [132, 181]. Dabei werden meist leistungsstarke und kostengünstige Industrie-PCs mit einer SPS-Programmiersoftware eingesetzt, die wie z. B. TwinCAT von der Fa. Beckhoff den Funktionsumfang von SPSen um Motion-, Robotik- und Bildverarbeitungsfunktionen erweitern. Solche SPSen arbeiten als Motion-Control-Systeme, die in Kapitel 7 beschrieben werden.

■ 1.3 Aufgaben im Industrial IoT

Das industrielle Internet of Things (IoT) vernetzt Rechner, Anlagen und Menschen, um Produktionsprozesse zu optimieren [108]. Die SPS spielt dabei eine wichtige Rolle, weil sie die von Sensoren gemessenen Daten über den Zustand der Anlage sammelt und über das Internet an eine Daten-Cloud überträgt. Durch Auswertung dieser Daten in der Cloud bekommt die SPS Befehle zurück, um die Automatisierung der Maschinen und Anlagen zu verbessern.

In der Regel findet man zwei große Einsatzfelder industrieller Automatisierungssysteme, zum einen in der Verfahrenstechnik [47], wie z. B. Chemie oder Life Science, und zum anderen in der Fertigungstechnik, z. B. zur Fabrikautomatisierung in der Automobilindustrie.

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.

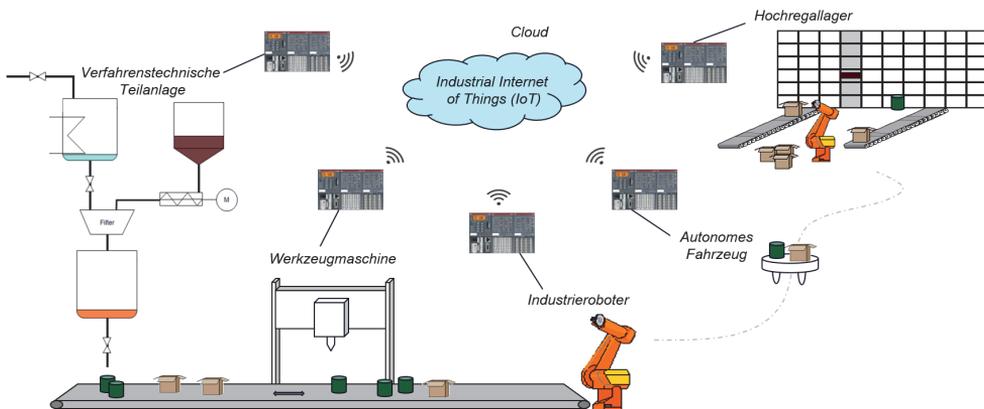


Bild 1.6 Produktionsanlage bestehend aus mehreren automatisierten Anlagenteilen (Cyber Physical Systems), die über die Cloud Daten austauschen

Beispiel 1.2: Produktionsanlage im Industrial IoT

In der Produktionsanlage nach Bild 1.6 werden Rohstoffe aus einem Hochregallager ausgelagert und von einem Fahrzeug, Roboterarm und Förderband zu einer verfahrenstechnischen Anlage transportiert. Die Anlage stellt ein Produkt her, das in Fässer abgefüllt wird, die von einer Werkzeugmaschine verschlossen werden. Danach hebt der Roboterarm die Fässer vom Band 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 SPSen übertragen die Prozessdaten an die Cloud und bekommen von ihr Befehle, welche Waren wann ausgelagert oder produziert werden sollen. Die Daten über Art und Menge der hergestellten Waren und eingesetzten Rohstoffe sowie die Dauer der in Anspruch genommenen Anlagenteile werden in der Cloud verwaltet.

1.3.1 Messen, Steuern, Regeln und Überwachen

Auch im Zeitalter des Internets 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 [64].

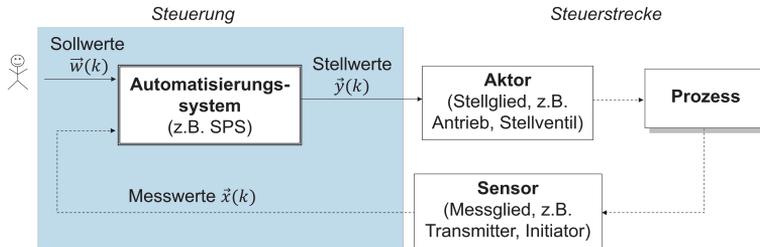


Bild 1.7 Aufbau eines Steuerkreises

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

Unter dem Oberbegriff Steuern (to control) wird häufig die gesamte Automatisierung verstanden, nämlich das Messen, Steuern, Regeln und Überwachen einer Anlage. 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 Flüssigkeit im Behälter bis zu den Platten der Schwinggabel angestiegen ist, schließt die SPS automatisch das Zulaufventil.

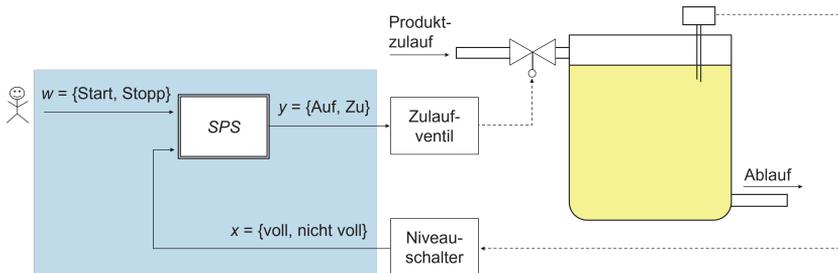


Bild 1.8 Steuerung des Produktzulaufs in einen Behälter

Beispiel 1.4: Regeln

Um die Flüssigkeit im Behälter nach Bild 1.9 auf eine gewünschte Temperatur einzustellen, führt die SPS eine Temperaturregelung durch. Hierfür misst ein Thermometer die Produkttemperatur im Behälter. Je nach Abweichung

des Temperatur-Istwerts x vom Sollwert w verändert das Regelungsprogramm in der SPS beim Heizen den Stellwert y_1 des Warmwasserventils und beim Kühlen den Stellwert 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.

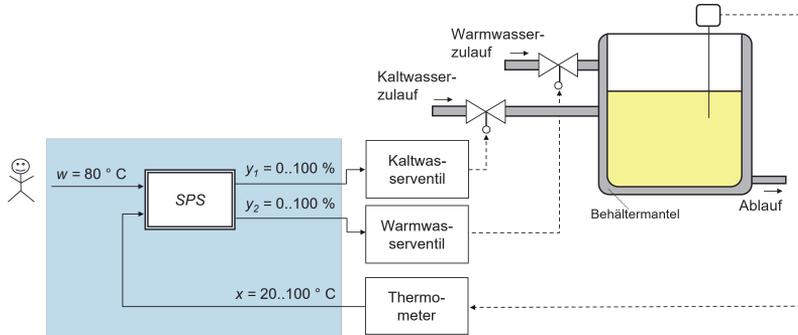


Bild 1.9 Regelung der Behältertemperatur in einer Chemieanlage

Beispiel 1.5: Überwachen

Viele Anwendungen zielen auch darauf ab, den Bediener auf besondere Betriebszustände, wie z. B. Störungen in der Anlage, 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).

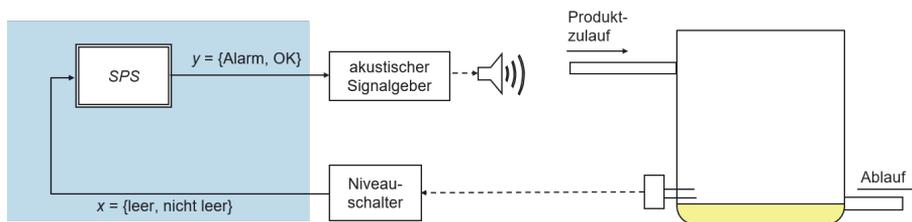


Bild 1.10 Füllstandsüberwachung eines Vorratsbehälters in einer Chemieanlage

1.3.2 Auswerten und Planen im Industrial Internet of Things

Die im Internet verwalteten *Dinge* in der Industrie 4.0 sind die Komponenten der Produktionsanlage, die sog. Assets. Hierzu zählen u.a. die Sensoren und Aktoren, aber auch Maschinen, Behälter und die eingesetzten Rohstoffe und hergestellten Produkte. Die Analyse der von den SPSen an die Cloud übertragenen großen Datenmenge (Big Data) über die Assets ermöglicht es, 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 [155].

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. Die Nutzung der an die Cloud übertragenen Daten für betriebswirtschaftliche und automatisierungstechnische Aufgaben wird in Kapitel 8 behandelt.

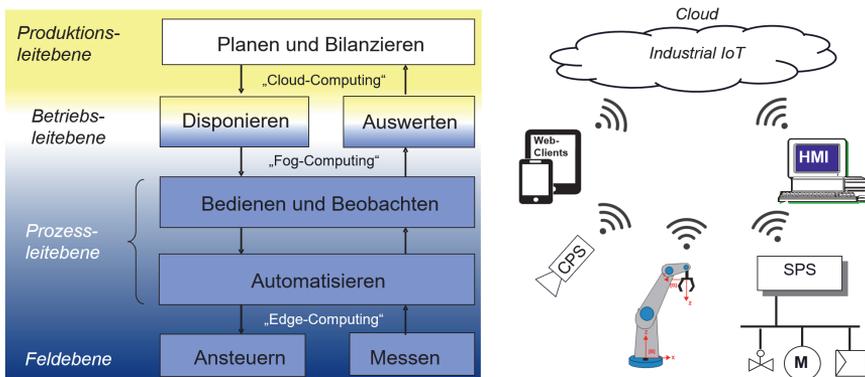


Bild 1.11 Hierarchie der Automatisierungsaufgaben im industriellen Internet of Things (IoT)

1.3.3 Virtualisierung und Testen mit digitalem Zwilling

Die in der Cloud gespeicherten Daten stellen ein digitales Abbild der Anlage und ihrer Assets dar. Durch Vergleich des Anlagenverhaltens mit dem idealen Verhalten ihres digitalen Zwillings können Abweichungen vom Normalbetrieb erkannt und frühzeitig prognostiziert werden, um Störungen und Unfälle zu vermeiden [105].

Außerdem ermöglicht der digitale Zwilling eine virtuelle Inbetriebnahme der Anlage. Dabei wird die entwickelte Steuerungssoftware mithilfe von Simulationsmodellen getestet. Kapitel 10 beschreibt die Vorgehensweise beim digitalen Engineering der Automatisierungssysteme.



Zusammenfassung

Eine SPS ist *das* Standard-Automatisierungsgerät, mit dem Prozesse in der Fertigungsindustrie oder der verfahrenstechnischen Industrie gesteuert, geregelt und überwacht werden. Sie kann wie die Wurzeln eines Baumes die Sensordaten in der Anlage einlesen und in übergeordneten Systemen oder im Internet zur Verfügung stellen. Dadurch entsteht ein digitales Abbild der Anlage in der Cloud, das für Simulationen, Fehlerfrüherkennung und Optimierung der Prozesse genutzt wird.

Wiederholungsfragen

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

Die Antworten finden Sie auf der SPS-Lern-und-Übungsseite.

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?

Eine Lösung zu dieser Übung finden Sie auf der SPS-Lern-und-Übungsseite.

2

Aufbau industrieller Steuerungen

Speicherprogrammierbare Steuerungen sind zentraler Bestandteil industrieller Automatisierungssysteme. Im Folgenden werden der SPS-Aufbau und die Strukturen von Automatisierungssystemen im Industrial IoT erläutert. Es wird gezeigt, wie die SPS Sensordaten einlesen und Befehle an die Aktoren ausgeben kann. Außerdem werden Systeme zum Bedienen und Beobachten vorgestellt.

■ 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.

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 sind. Der EEPROM ist häufig als steckbare SD-Card realisiert. Sein Speicherinhalt bleibt bei Spannungsausfall erhalten [149, 179].

Zur Ausführung eines Programms wird es vom EEPROM in den Arbeitsspeicher (Random Access Memory, RAM) kopiert, wo die CPU schnellen Zugriff hat. Außerdem werden im RAM die von den Programmen benötigten Variablenwerte gespeichert. Während ihrer Abarbeitung lesen und schreiben die Programme Variablen aus dem RAM. Der Speicherinhalt des RAMs geht aber bei Spannungsausfall verloren.



Die Hersteller bieten CPUs mit unterschiedlicher Leistungsfähigkeit an. Die Auswahl der CPU muss gemäß der Größe und Anforderungen der zu automatisierenden Anlage erfolgen. 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 angeschlossen. Für die Ankopplung busfähiger Sensoren und Aktoren verfügt die SPS über Baugruppen mit Felddbuschnittstellen.

Ein interner *Daten- und Adressbus* verbindet die Baugruppen der SPS und ermöglicht den schnellen 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.

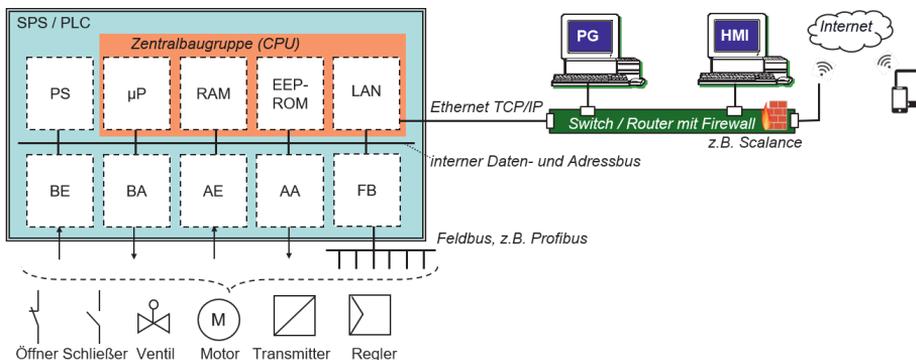


Bild 2.1 Hardwareaufbau einer SPS mit Stromversorgung (PS), µP, RAM, EEPROM, Ethernet-Schnittstelle (LAN) auf einer Zentralbaugruppe und Baugruppen mit binären Ein- (BE) und Ausgangskanälen (BA) sowie analogen Ein- (AE) und Ausgangskanälen (AA) und Felddbuschnittstellen (FB)

2.1.3 Programmiergerät

Das Programmiergerät (PG) 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 Programmiersoftware des Herstellers, z. B. Codesys oder das TIA-Portal von Siemens.

Die vom Anwender erstellten Programme werden in SPS-spezifischen Code übersetzt und können zunächst auf dem Programmiergerät simuliert und ggf. korrigiert werden, bevor sie in die SPS geladen werden und eine eventuell empfindliche Anlage steuern. Beim Laden wird der übersetzte SPS-Code an die Steuerung übertragen. Dabei werden die laufenden Programme gestoppt und die Variablen neu initialisiert.

Einige SPSen erlauben wahlweise auch einen Online-Change, bei dem nur Änderungen übersetzt und ohne Anhalten der Steuerung oder Verlust der Variablenwerte geladen werden [149, 179]. In jedem Fall ist das Programmiersystem aber mit einem Passwort zu sichern, damit der Zugriff auf die SPS nur für autorisierte Bediener möglich ist. Um zu verhindern, dass z. B. während einer Reparatur unerwünscht von der SPS ein Programm abgefahren wird, kann die SPS

in die Betriebsart STOP geschaltet werden. Dann werden alle laufenden Programme angehalten und die SPS-Ausgänge stromlos geschaltet.

2.1.4 Human Machine Interface

Außer dem Programmiersystem laufen auf dem PC auch Programme, die es Menschen erlauben, die 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. Motoren, Ventile oder Regler, von Hand zu aktivieren und somit manuell in den Prozess einzugreifen (s. Abschnitt 2.7).

Auch Tablets oder Smartphone können drahtlos über das Internet mit einer SPS verbunden werden. Die SPS muss dann aber u. a. durch eine *Firewall*, die im Router bzw. Netzwerkschwitch läuft, vor Angriffen aus dem Internet geschützt werden (s. Abschnitt 9.3).

2.1.5 Vernetzte Automatisierungssysteme

In großen Fabriken werden Hunderte von SPSen eingesetzt, die miteinander, mit HMIs und mit anderen Netzwerken Daten austauschen. Dadurch entstehen vernetzte Automatisierungssysteme wie in Bild 2.2.

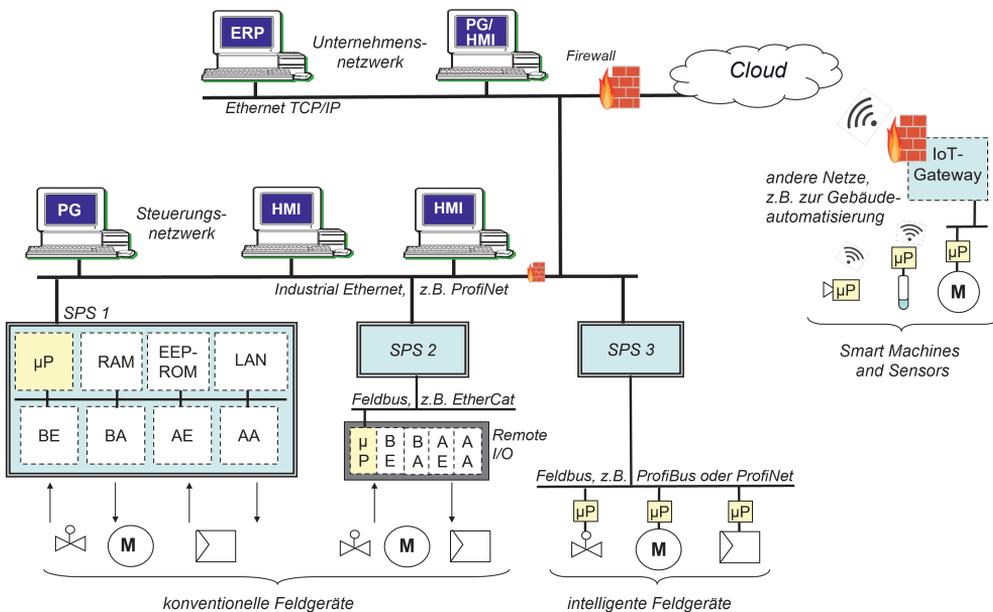


Bild 2.2 Vernetzte Automatisierungssysteme bestehen aus zahlreichen miteinander gekoppelten Feldgeräten, SPSen, HMIs und PCs, die z. B. zum Enterprise Resource Planning (ERP) auch Daten über die Cloud austauschen

Die Feldgeräte können dabei wie in den folgenden Abschnitten erläutert auf unterschiedliche Arten an die SPS angekoppelt werden:

- *konventionell* durch Kupferdrahtleitungen,
- über *Feldbus*, was jedoch intelligente (busfähige) Feldgeräte erfordert,
- über eine *Remote-I/O*, die einerseits über Feldbus mit der SPS verbunden ist und andererseits nicht-busfähige Feldgeräte über dezentrale E/A-Baugruppen anbindet,
- *drahtlos* per Funk (z. B. 5G) oder WLAN, dem im industriellen Umfeld wegen der Störsicherheit aber eine kabelgebundene Ankopplung vorgezogen wird.

■ 2.2 SPS-Arten und IoT-Geräte

Grundsätzlich unterscheidet man drei verschiedene Aufbauarten bei SPSen, nämlich als Hardware-, Slot- oder Soft-SPS. Während die Slot-SPS heutzutage kaum noch eingesetzt wird, sind PC-basierte Steuerungen als Soft-SPS auf dem Vormarsch. PC-basierte Steuerungen können als Edge Controller oder IoT-Gerät eingesetzt werden [51].

2.2.1 Hardware-SPS

Die klassische Aufbauform einer SPS ist eine Hardware-SPS wie in Bild 1.1. Sie erfüllt industrielle Anforderungen hinsichtlich Staub- und Spritzwasserschutz, Durchschlagssicherheit, Elektromagnetischer Verträglichkeit sowie Robustheit gegen Vibrationen und Erschütterungen. Ihre Komponenten sind als Baugruppen über einen Rückwandbus miteinander verbunden. Bei solch modularen Hardware-SPSen können die Baugruppen für CPU, Feldbusankopplung oder Eingangs- und Ausgangs-Kanäle individuell zusammengesteckt werden je nach Größe und Anforderungen der Anlage.

Kleinere Anlagen können auch mit einer sog. Kompakt-SPS automatisiert werden, bei der CPU und E/A-Kanäle in einem Gehäuse integriert sind. Eine Hardware-SPS bedarf immer eines externen PCs als Programmiergerät (vgl. Bild 2.1).

2.2.2 Slot-SPS

Eine Slot-SPS ist eine Einsteckkarte für einen Desktop-PC, die alle Module einer SPS enthält. Anstatt einer CPU besitzt sie einen Co-Prozessor, auf dem ein eigenes multitaskingfähiges Betriebssystem mit einem gemeinsamen Arbeitsspeicher für PC und SPS läuft. Im Grunde nutzt also die Slot-SPS lediglich die Stromversorgung des PCs.

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 Industrie-PC, ein Notebook oder ein Mini-PC wie der RaspberryPi