



EUROPA-FACHBUCHREIHE
für elektrotechnische Berufe

SPS

Theorie und Praxis

mit Übungsaufgaben

7. Auflage

VERLAG EUROPA-LEHRMITTEL · Nourney, Vollmer GmbH & Co. KG
Düsselberger Straße 23 · 42781 Haan-Gruiten

Europa-Nr.: 30009

Autor

Herbert Tapken Dipl.-Ing (FH), Dipl. Berufspädagoge 26203 Wardenburg

ISBN 978-3-8085-3112-9

7. Auflage 2022
Druck 5 4 3 2 1

Alle Drucke derselben Auflage sind parallel einsetzbar, da sie bis auf die Korrektur von Druckfehlern identisch sind.

Alle Rechte vorbehalten. Das Werk ist urheberrechtlich geschützt. Jede Verwertung außerhalb der gesetzlich geregelten Fälle muss vom Verlag schriftlich genehmigt werden.

© 2022 by Verlag Europa-Lehrmittel, Nourney, Vollmer GmbH & Co. KG, 42781 Haan-Gruiten
www.europa-lehrmittel.de

Umschlag: Media-Creativ, 40723 Hilden
Satz und Layout: rkt, 51379 Leverkusen
Druck: mediaprint solutions GmbH, 33100 Paderborn

Hinweis zur Software:

Die Aufgaben können mit jeder beliebigen SPS bearbeitet werden.

TIA-Portal: Im Buch wird die Software TIA-Portal, inklusive PLCSIM, von Siemens vorgestellt. Alle Lösungen sind ebenfalls mit dem TIA-Portal erstellt worden. Die Software kann bei Siemens und vielen anderen Anbietern erworben werden.

PLC-LAB: Viele Aufgaben im Buch können mit der Simulations- und Visualisierungssoftware PLC-LAB getestet werden. Neben der Vollversion kann bei der Fa. MHJ-Software auch eine speziell für die Aufgaben dieses Buches erstellte **PLC-Lab-Runtime** (19,- €) oder die **PLC-Lab 30-Tage Demo** (kostenlos) bezogen werden. Um die Praxis-Beispiele im Buch mit PLC-Lab durchführen zu können, benötigen Sie das TIA PORTAL (Basic oder Professional).

Download-Link für PLC-Lab-Runtime: https://www.mhj-download.de/plclab/plc_lab_runtime_stp.zip

Vorwort

In Industrie und Handwerk sind automatisierte Prozesse nicht mehr wegzudenken. Über Speicherprogrammierbare Steuerungen (SPS) werden Maschinen und Anlagen gesteuert. Die Automatisierungstechnik ist ein fester Bestandteil der Technik geworden.

Das vorliegende Buch ist ein Lehr- und Arbeitsbuch. Es soll Grund- und Aufbaukenntnisse im Bereich der Speicherprogrammierbaren Steuerungen vermitteln. Die einzelnen Themen werden zunächst fachlich erklärt und dann durch Wiederholungsfragen gefestigt. Anhand von Übungsaufgaben mit verschiedenen Schwierigkeitsgraden kann das Gelernte angewendet werden. Eine Vielzahl von Aufgaben kann mit der Software PLCLAB durch animierte Visualisierungen simuliert werden.

Die theoretischen Erläuterungen, die Beispiele und Übungen basieren auf dem Automatisierungssystem SIMATIC und der Software TIA-Portal der Fa. Siemens. Die Aufgaben können jedoch mit jeder beliebigen SPS-Software bearbeitet werden.

Das Buch richtet sich an alle Berufe aus dem Bereich **Elektrotechnik, Metalltechnik** und **Mechatronik** sowie an **alle beruflichen Vollzeitschulen**, die sich mit der Thematik der Steuerungs- und Automatisierungstechnik beschäftigen. Es kann sowohl als Lehr- und Arbeitsbuch für die **schulische oder betriebliche Aus- und Weiterbildung** als auch für das **Selbststudium** genutzt werden.

Der fachliche Teil des Buches reicht von einfachen Digitalverknüpfungen bis zu vernetzten Automatisierungssystemen. Zudem wird auch auf die SPS-Hardware und auf die Fehlersuche eingegangen.

Die Aufgaben im Buch haben eine Bandbreite von einfachen Programmierübungen bis hin zu komplexen Projekten. Daher ist das Buch sowohl für die **Berufsausbildung** als auch für die **Meister- oder Technikerschule** bis hin zum **Studium** geeignet.

Zu dem Buch ist ein Lösungsbuch mit den Lösungen aller Aufgaben erhältlich.

Bei der Erstellung des Buches, der Aufgaben und der Lösungen wurde mit großer Sorgfalt vorgegangen. Da Fehler aber nie ganz auszuschließen sind, können Verlag und Autor für fehlerhafte Angaben oder Lösungen keine Haftung oder juristische Verantwortung übernehmen.

Bei der Bearbeitung des Buches wünsche ich viel Spaß und Erfolg bei der Lösung der Aufgaben.

Vorwort zur 7. Auflage

Die Automatisierungstechnik entwickelt sich rasant weiter. Eine immer stärkere Vernetzung bis zu Industrie 4.0 Anlagen hat in den Betrieben Einzug genommen. Zudem ist ein Trend zu mehr Programmierung mit neueren Programmiersprachen zu erkennen. Auf die wachsenden Anforderungen wird in der neuen Auflage eingegangen.

Neben den bisher behandelten Programmiersprachen FUP, KOP, AWL, SCL und S7-Graph wird die neue Programmiersprache **Cause-Effect-Matrix** behandelt.

Die meisten Aufgaben können mit der Software PLC-Lab simuliert werden. Neben verbesserter Grafik bietet die Software auch mehr Funktionalitäten. Wer zum Simulieren die Vollversion nicht besitzt, kann auch eine kostengünstige Runtime-Lizenz bei der Fa. MHJ-Software erwerben. Für Benutzer, die weiterhin mit SPS-VISU arbeiten möchten, stehen alle Dateien digital zur Verfügung (siehe Umschlagseite vorne innen). Die Aufgaben können mit beiden Systemen simuliert werden.

Bei der Lektüre des Buches sowie beim Bearbeiten der Aufgaben wünsche ich Neugier, Spaß und viel Erfolg.

Autor und Verlag sind allen Nutzern des Buches für kritisch-konstruktive Hinweise und Verbesserungsvorschläge dankbar. Bitte senden Sie diese an lektorat@europa-lehrmittel.de

Wardenburg, im Sommer 2022

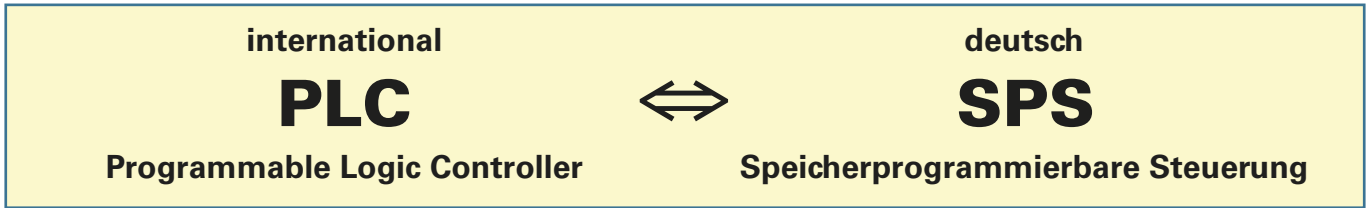
Herbert Tapken (Autor)

1	SPS-Grundlagen	7
1.1	Einleitung	7
1.2	Arten von Steuerungen	7
1.3	SPS-Bezeichnung	8
1.4	SPS – Systemvergleich	8
1.5	Aufbau und Wirkungsweise einer SPS	9
1.6	Wiederholungsfragen	10
2	Hardware	12
2.1	SPS-Aufbau	12
2.2	SPS-Produktspektrum	13
2.3	Darstellung von SPSen in Stromlaufplänen	17
2.4	Wiederholungsfragen	18
3	Erstellen eines SPS-Programms	19
3.1	Vorgehensweise bei der Projektbearbeitung	19
3.2	TIA-Portal: Erstellen eines Projektes	20
4	Simulation von Programmen	23
4.1	Simulation mit PLCSIM	23
4.2	Simulation mit PLC-Lab	24
5	Grundverknüpfungen	25
5.1	Programmiersprachen/Darstellungsarten	25
5.2	Grundlagen der Grundfunktionen	26
5.3	Übersicht der Grundfunktionen	27
5.4	Grundverknüpfungen in verschiedenen Programmiersprachen	28
5.5	Addressierung	29
5.6	Merker	29
5.7	Verknüpfungsergebnis VKE	30
5.8	Beispielaufgabe: Kühlhaus	31
5.9	Wiederholungsfragen	33
5.10	Übung: Sicherheitscode	35
5.11	Übung: Folgeschaltung von Montagebändern	35
5.12	Übung: Funktionsgleichung	35
5.13	Übung: Rauchmeldeanlage	36
5.14	Übung: Alarmanlage	37
5.15	Übung: Förderbandanlage	38
6	Flipflops (Speicherfunktionen)	40
6.1	SR-Flipflop und RS-Flipflop	40
6.2	Beispielaufgabe: Ansteuerung eines Drehstrommotors	42
6.3	Wiederholungsfragen	44
6.4	Übung: Doppelt wirkender Zylinder	46
6.5	Übung: Wendeschützschtaltung	46
6.6	Übung: Förderbandanlage (Folgeschaltung)	47
6.7	Übung: Toranlage	48
6.8	Übung: Sortieranlage	49
7	Strukturierte Programmierung	51
7.1	Lineare Programmierung	51
7.2	Strukturierte Programmierung	51
7.3	Bausteinarten	52
7.4	Wiederholungsfragen	52
8	Zeitfunktionen	53
8.1	SIMATIC-Zeiten	53
8.2	IEC-Zeiten	55
8.3	Taktmerker	56
8.4	Beispielaufgabe: Pneumatische Abfülleinrichtung	57
8.5	Wiederholungsfragen	59
8.6	Übung: Störungslampe (Taktmerker)	60
8.7	Übung: Industrieofen	60

8.8	Übung: Automatische Stern-Dreieck-Schaltung	60
8.9	Übung: Zeitgesteuerte Toranlage	61
8.10	Übung: Zeitgesteuerte Förderbandanlage	62
9	Bit, Byte, Wort, Doppelwort	63
9.1	Zahlensysteme	63
9.2	Definitionen	64
9.2.1	Bit	64
9.2.2	Byte	64
9.2.3	Wort	64
9.2.4	Doppelwort	65
9.3	Lade- und Transferoperationen	65
9.3.1	Lade- und Transferoperationen in AWL	66
9.3.2	Lade- und Transferoperationen in FUP, KOP und AWL	67
9.4	Wiederholungsfragen	67
9.5	Übung: Wortverarbeitung	69
10	Zähler und Vergleicher	70
10.1	SIMATIC-Zähler	70
10.2	Vergleicher	71
10.3	SIMATIC-Vorwärts-/Rückwärtszähler mit Vergleicher in AWL, FUP und KOP	72
10.4	IEC-Zähler	74
10.5	Wiederholungsfragen	75
10.6	Übung: Parkplatzampel	77
10.7	Übung: Stanze	78
11	Verschiedene Programmfunktionen und Befehle	79
11.1	Urlöschen	79
11.2	Systemmerker	79
11.3	Archivieren/Deaktivieren	79
11.4	Flankenauswertung	79
11.5	Sprungoperationen	80
11.6	Wiederholungsfragen	81
12	Bausteine	82
12.1	Bausteinarten	82
12.1.1	Organisationsbausteine (OB)	82
12.1.2	Funktionen (FC)	82
12.1.3	Funktionsbaustein (FB)	82
12.1.4	Systemfunktionen und Systemfunktionsbausteine	82
12.1.5	Datenbausteine (DB)	82
12.2	Bibliotheksfähige Bausteine	83
12.3	Anlegen einer eigenen Bibliothek	86
12.4	Datenbausteine	87
12.5	Wiederholungsfragen	89
12.6	Übung: Motorsteuerung mit bibliotheksfähigen Bausteinen	90
13	Ablaufsteuerungen	92
13.1	Grundlagen zu Ablaufsteuerungen	92
13.2	Darstellung von Schrittketten	94
13.2.1	GRAFCET und DIN EN 61131-3	94
13.2.2	Strukturierung von GRAFCETs	98
13.3	S7-Graph	99
13.4	Betriebsarten	100
13.5	Wiederholungsfragen	101
13.6	Übung: Leuchtreklame	104
13.7	Übung: Schwimmbad	104
13.8	Übung: Bohranlage	105
13.9	Übung: Ampelsteuerung	107
14	Fehlersuche	109
14.1	Fehlerarten	109
14.2	Fehlersuche bei Hardwarefehlern	109
14.3	Fehlersuche bei Softwarefehlern	109

14.4	Fehler-Operationsbausteine	112
14.5	Wiederholungsfragen	112
14.6	Übung: Förderbandanlage (Fehlersuche)	114
14.7	Verpackungsanlage (Fehlersuche)	116
15	Mathematische Funktionen	118
15.1	Datentypen	118
15.2	Umwandlungsfunktionen	119
15.3	Rechnen mit Ganzzahlen (INT und DINT)	120
15.5	Übung: Umwandlungsfunktionen	121
15.6	Übung: Mathematische Operationen	122
16	Verarbeitung von Analogwerten	123
16.1	Analoge Signale	123
16.2	Analogwerte einlesen und ausgeben	124
16.3	Analogwerte einlesen und normieren	124
16.4	Analogwerte normieren und ausgeben	126
16.5	Wiederholungsfragen	127
16.6	Übung: Temperaturanzeige	128
16.7	Übung: Temperaturüberwachung	129
16.8	Pegelmessung an einem Wasserkraftwerk 1	130
16.9	Pegelmessung an einem Wasserkraftwerk 2	131
17	Cause-Effect-Matrix (CEM) und Structed Control Language (SCL)	132
17.1	Cause-Effect-Matrix	132
17.2	Structed Control Language	134
17.2.1	SCL-Befehle	134
17.2.2	Wiederholungsfragen	137
17.2.3	Übung: Zeitgesteuerte Toranlage in SCL	138
18	Vernetzte Automatisierungssysteme	139
18.1	Hierarchischer Aufbau von Automatisierungssystemen	139
18.2	Grundlagen Netzwerktechnik	139
18.2.1	Aufbau eines kleinen Automatisierungsnetzwerk	139
18.2.2	Netzwerkarchitekturen	140
18.2.3	Konfiguration eines Netzwerkes	140
18.3	Topologien	142
18.4	Übertragungsmedien	143
18.5	Störgrößen bei leitungsgebundener Datenübertragung	143
18.6	Buszugriffsverfahren	144
18.7	Industrielle Bussysteme	145
18.7.1	Ethernet TCP/IP	145
18.7.2	Industrial Ethernet	145
18.7.3	PROFINET	145
18.7.4	Profibus DP	146
18.7.5	Aktor-Sensor-Interface (AS-I)	146
18.8	Industrie 4.0	147
18.8.1	Was ist Industrie 4.0	147
18.8.2	Aufbau einer Industrie 4.0-Anlage	148
18.9	Wiederholungsfragen	149
19	Projektaufgaben	151
19.1	Übung: Motorsteuerung mit bibliotheksfähigen Bausteinen	151
19.2	Übung: Ampelanlage	153
19.3	Übung: Lackierstraße	157
19.4	Übung: Autowaschanlage	159
20	Übersicht Befehle unter Step7	162
21	Sachwortverzeichnis	164
Bild und Textquellen		166
Literatur und Downloads		166

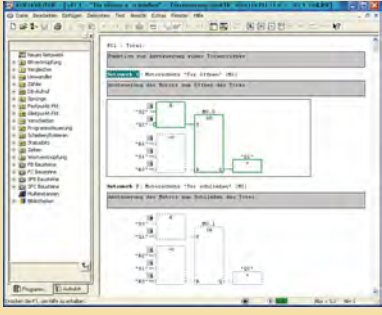
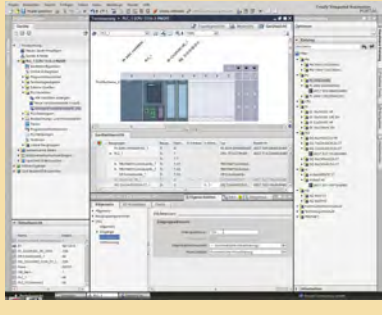
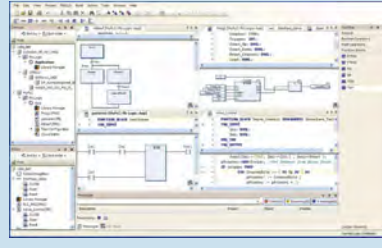


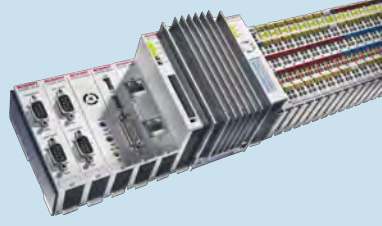
1.3 SPS-Bezeichnung



1.4 SPS – Systemvergleich

Es gibt verschiedene SPS-Grundsysteme. Zum einen gibt es die Siemens-Produkte, wie S7-300, S7-1200 und S7-1500, die mit der Software Step 7 bzw. mit dem TIA-Portal programmiert werden. Auf der anderen Seite gibt es eine Vielzahl anderer Hersteller, die in der Regel über die Programmiersoftware CoDeSys (nach IEC61131-3) programmiert werden. Zusätzlich zu der Grundsoftware benötigt man eine firmenspezifische Target-Software, um das in CoDeSys erstellte Programm an die Steuerung anzupassen.

Die DIN EN 61131-3 ist die deutsche Fassung der internationalen Norm IEC 61131-3.

SYSTEMVERGLEICH			
	Siemens		Andere Hersteller
Norm	Siemens spezifische Bausteine und IEC 61131-3	Siemens spezifische Bausteine und IEC 61131-3	IEC 61131-3
Software	<p style="text-align: center;">Step 7 V5.x</p> 	<p style="text-align: center;">TIA-Portal ab V11</p> 	<p style="text-align: center;">CoDeSys</p> 
Eingänge/Ausgänge	<p style="text-align: center;">E 0.0 A 0.0</p>	<p style="text-align: center;">% E 0.0 % A 0.0</p>	<p style="text-align: center;">% IX 0.0 % QX 0.0</p>
Hardware	<p style="text-align: center;">S7-200 S7-300 S7-400</p> 	<p style="text-align: center;">S7-300 S7-400 S7-1200 S7-1500</p> 	<p style="text-align: center;">z. B.: Beckhoff WAGO Festo</p> 

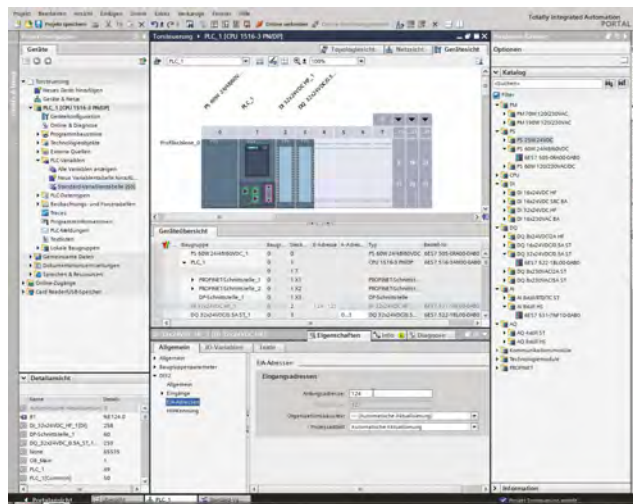
3. Hardware auswählen

Die einzelnen Hardwarekomponenten können aus dem Katalog im rechten Fenster ausgewählt und auf den entsprechenden Steckplatz gezogen werden.

Wenn im Arbeitsbereich ein Element ausgewählt wird, können im Inspektorfenster die Eigenschaften angesehen und geändert werden.

Ändern Sie (zur Übung) die Adressen der Ein- und Ausgänge jeweils auf die Anfangsadressen 124.

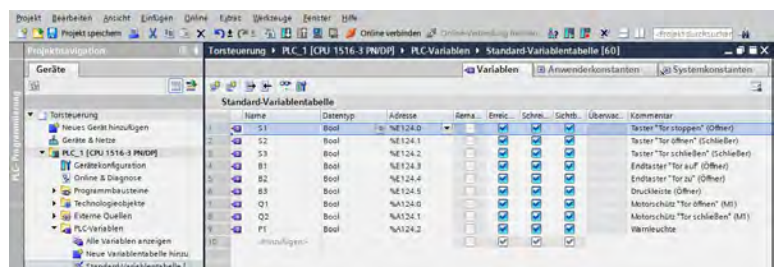
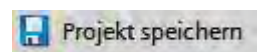
Speichern Sie die Gerätekonfiguration über **Projekt speichern**.



4. Variablen-tabelle erstellen

Wählen Sie in der **Projektstruktur** die **Standard-Variablen-tabelle** an. Geben Sie die Eingänge, Ausgänge, Merker usw., die Sie verwenden möchten, ein. Unter **Name** kann ein *symbolischer Name* frei vergeben werden. In dem Feld **Adresse** wird die *absolute Adresse*, entsprechend der Hardwarekonfiguration, zugeordnet. Außerdem kann ein **Kommentar** eingetragen werden.

Die internationale Schreibweise der E-/A-Adressen (%I0.0, %Q0.0) lässt sich auf die deutsche (%E0.0, %A0.0) unter **Extras** => **Einstellungen** => **Mnemonik** => **deutsch** umstellen.

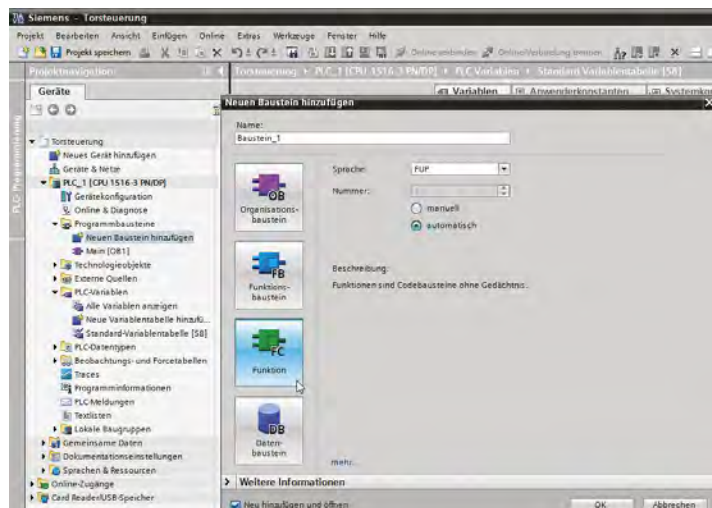


5. Bausteine hinzufügen

Über **Neuen Baustein hinzufügen** können die verschiedenen Bausteintypen hinzugefügt werden.

Erstellen Sie eine **Funktion FC1** in der Programmiersprache **FUP**.

Bestätigen Sie mit **OK**.



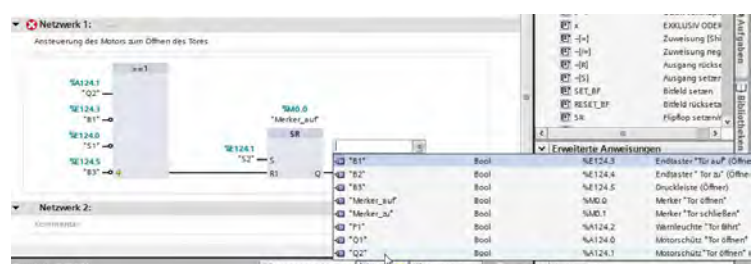
6. Programm erstellen

In der Funktion (hier: FC1) wird jetzt das Anwenderprogramm erstellt.

Die **Programmiersprache** kann im Inspektorfenster des FC bei Bedarf geändert werden.

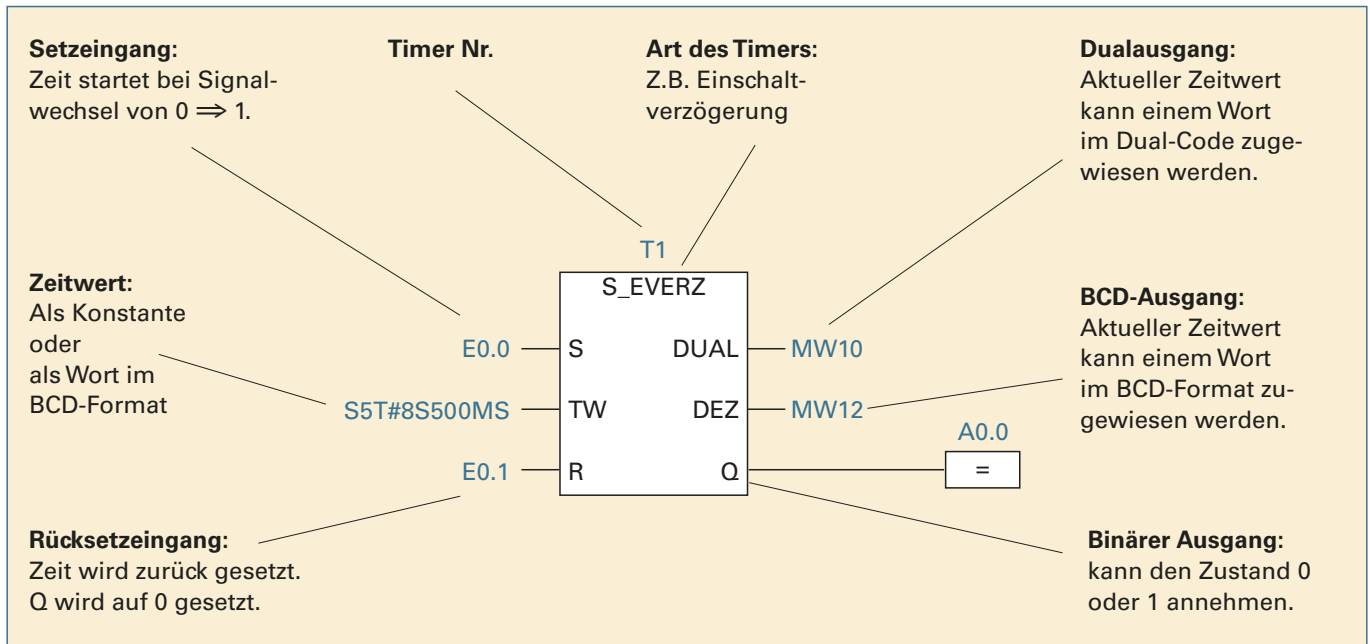
Beim Programmieren kann über ein Auswahlfenster, das bei der Eingabe der Ein- und Ausgänge am rechten Rand erscheint, auf die Symboltabelle zugegriffen werden. Hier können die einzelnen Operanden ausgewählt werden.

Es ist auch möglich, die Adressen oder den symbolischen Namen direkt einzugeben.



Zeitfunktionen sind grundlegende Elemente in der Steuerungstechnik. Mit ihrer Hilfe können zeitliche Verzögerungen, Zeitakte oder Wartezeiten realisiert werden.

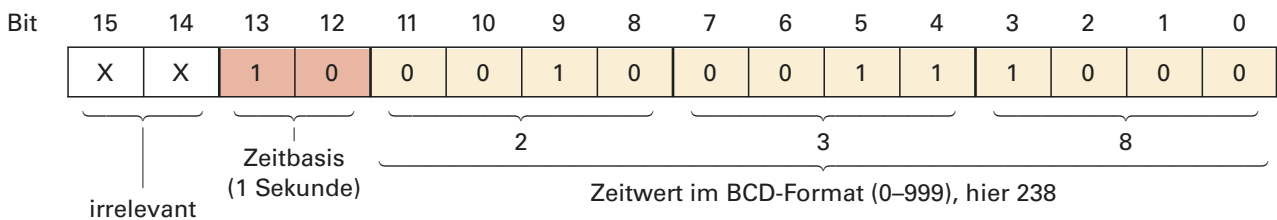
Unter Step7 stehen fünf verschiedene Zeitfunktionen (Timer) zur Verfügung, die gewählt werden können.



Die Vorgabe der Zeit am Zeitbaustein kann über Angabe einer Konstanten, z.B. S5T#10s, oder in Form eines Eingangs-, Ausgangs- oder Merkerwortes erfolgen.

Die Befehlszeile NOP 0 (Nulloperation) in AWL wird als Platzhalter eingefügt. Sie hat keine Befehlsbedeutung.

Intern wird eine Zeit durch eine Zeitbasis und einen Zeitwert realisiert. Eine Zeitwertvorgabe besteht aus 16 Bit, wobei die Bits 0 bis 11 den Zeitwert im BCD-Format (siehe Kapitel 9.1.3) und die Bits 12 und 13 die Zeitbasis enthalten. Bit 14 und 15 sind irrelevant und werden nicht beachtet, wenn die Zeit gestartet wird.



Zeitbasis			Auflösung	
Bit 13	Bit 12	Zeitraster	kleinster Wert	größter Wert
0	0	10 ms	10 ms	9990 ms (9 s 990 ms)
0	1	0,1 s	100 ms	99900 ms (1 min 39 s 900ms)
1	0	1 s	1 s	999 s (16 min 39 s)
1	1	10 s	10 s	9990 s (2 h 46 min 30 s)

Die Zeitangaben können in Stunden, Minuten, Sekunden und Millisekunden erfolgen.

Beispiele für Zeitangaben:

400 ms ⇒ S5T#400MS

17,8 s ⇒ S5T#17S800MS

2 min, 20 s ⇒ S5T#2M20S

1 h, 10 min ⇒ S5T#1H10M

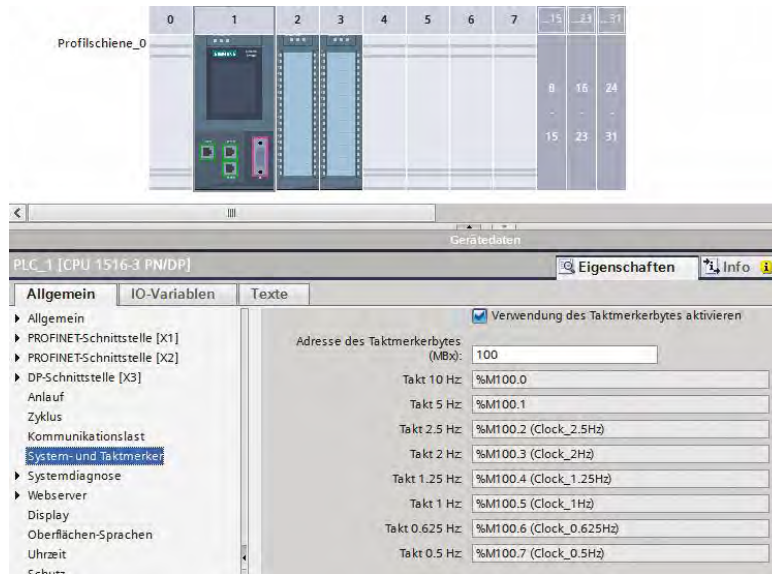
8.3 Taktmerker

Um ein Taktsignal, z.B. zum Blinken einer Warnleuchte, in der SPS verwenden zu können, stellt die CPU einen Taktmerker zur Verfügung.

Dieser muss in der Gerätekonfiguration festgelegt werden.

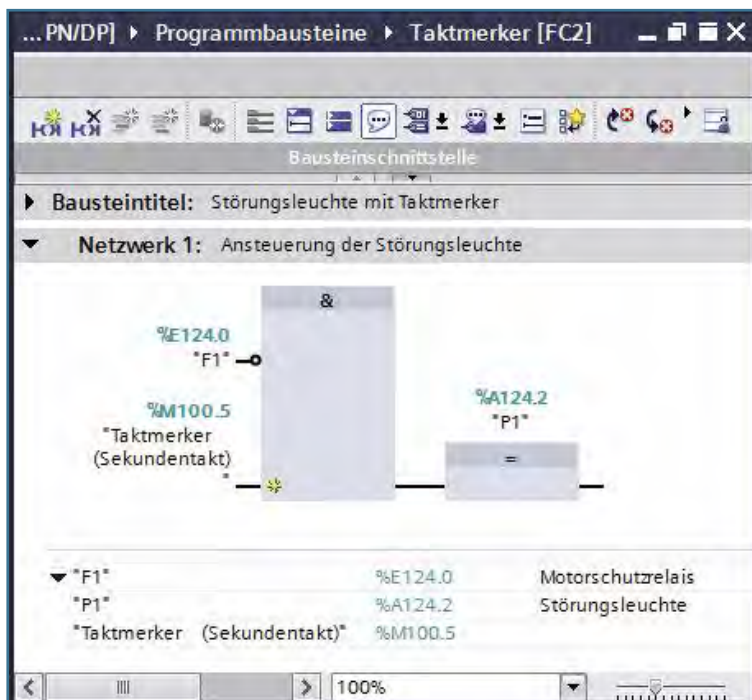
Geräte-/Hardwarekonfiguration ⇒ CPU anwählen ⇒ **System- und Taktmerker** markieren ⇒ **Merkerbyte** wählen.

Das Merkerbyte, das als Taktmerker arbeiten soll, kann frei gewählt werden. Es ist aber darauf zu achten, dass das Merkerbyte nicht anderweitig im Programm benutzt wird.



Jedem Bit des Taktmerkerbytes ist eine Periodendauer/Frequenz zugeordnet:

Bit	7	6	5	4	3	2	1	0
Periodendauer (s):	2	1,6	1	0,8	0,5	0,4	0,2	0,1
Frequenz (Hz):	0,5	0,625	1	1,25	2	2,5	5	10



Beispiel:

Beim Auslösen eines Motorschutzrelais (E 124.0, Öffner) soll eine Störungsleuchte (A 124.2) im Sekundentakt blinken. Als Taktmerker soll das Merkerbyte 100 verwendet werden.

Beim Auslösen des Motorschutzrelais wird der Eingang E124.0 auf „0“ gesetzt.

Da das Merkerbit 100.5 im Sekundentakt ein- und ausgeschaltet wird, entsteht durch die UND-Verknüpfung ein Blinksignal am Ausgang A124.2.

Hinweis:

Wenn der Taktmerker im Visualisierungsprogramm SPS-Visu verwendet werden soll, muss auch die Hardwarekonfiguration von Step7 in die Visualisierung geladen werden.

9 Bit, Byte, Wort, Doppelwort

9.1 Zahlensysteme

Im Folgenden werden verschiedene Zahlensysteme vorgestellt, da die Kenntnis darüber in der SPS-Technik unverzichtbar ist.

9.1.1 Das Dezimalsystem

Wenn im täglichen Leben etwas durch eine Zahl zum Ausdruck gebracht werden soll, z.B. ein Längenmaß, so verwendet man eine Zahl des dezimalen Zahlensystems.

Das dezimale Zahlensystem hat als Basiszahl die „10“. Das bedeutet, dass jede Zahl als Vielfaches einer Zehnerpotenz ausgedrückt wird.

Die **dezimale Zahl 2456** setzt sich folgendermaßen zusammen:

2456			
$= 2 \cdot 10^3 + 4 \cdot 10^2 + 5 \cdot 10^1 + 6 \cdot 10^0$			
10^3	10^2	10^1	10^0
2	4	5	6

9.1.2 Das duale Zahlensystem (Binärsystem)

Die Digitaltechnik ist nur in der Lage, eine „0“ oder eine „1“ zu erkennen.

Deshalb wird in der Digitaltechnik das duale Zahlensystem verwendet. Bei diesem System stellt die „2“ die Basis dar. Jede Zahl wird, ähnlich wie im Dezimalsystem, als Vielfaches einer Potenz von „2“ ausgedrückt.

Beispiel: Es soll die **dezimale Zahl 239** durch eine Dualzahl dargestellt werden.

Potenz	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0							
Wert dezimal	128	64	32	16	8	4	2	1							
Duale Darstellung	1	1	1	0	1	1	1	1							
Berechnung	128	+	64	+	32	+	0	+	8	+	4	+	2	+	1

9.1.3 Das BCD-Zahlensystem

Bei dieser Art der Darstellung wird jede Dezimalstelle durch vier Bit des Dual-Codes ausgedrückt. Der BCD-Code wird oft bei Vorwahlschaltern und Ziffernanzeigen verwendet.

Beispiel: Darstellung der **dezimalen Zahl 239** im BCD-Code.

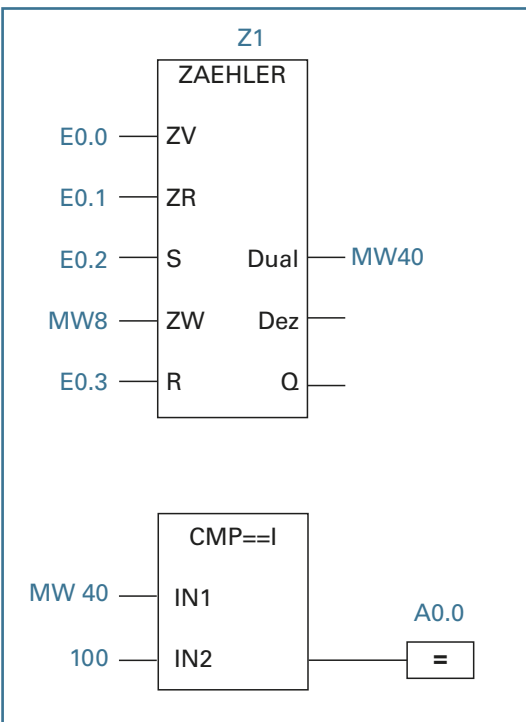
Potenz	2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
Wert	100 er				10 er				1 er			
Wert dezimal	8	4	2	1	8	4	2	1	8	4	2	1
BCD-Code	0	0	1	0	0	0	1	1	1	0	0	1

Das Bitmuster 1111 kann nicht als Ziffer angezeigt werden, da es die 9 übersteigt. Eine BCD-Anzeige würde auf „dunkel“ schalten.

5 Wann wird am Ausgang Q eines SIMATIC-Zählers eine 1 bzw. eine 0 ausgegeben?

- Welche verschiedenen Datentypen können bei Vergleichen verwendet werden?

& Erklären Sie die Funktion der Schaltung.



(Wie lauten die AWL-Befehle, um den aktuellen Zählwert des SIMATIC-Zählers Z1 dualcodiert in das Merkerwort MW10 zu schreiben?

10.7 Übung: Stanze

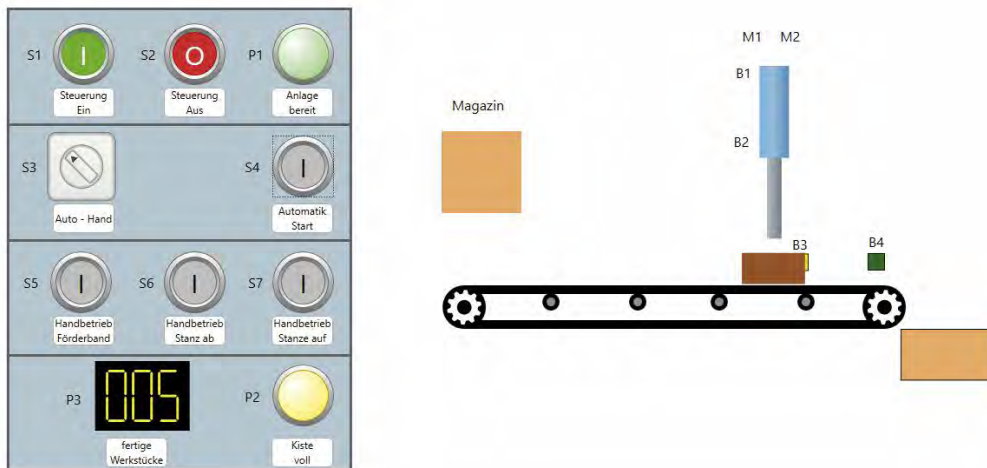


Problemstellung

Short-Link: vel.plus/SPS14

Über ein Förderband werden Werkstücke einer Stanze zugeführt und anschließend in eine Kiste transportiert. Dort können maximal 5 Werkstücke aufgenommen werden. Eine Anzeige gibt die Anzahl der Werkstücke an. Im Automatikbetrieb wird das Förderband über den Taster Automatik Start eingeschaltet. Wenn die Endlage B3 erreicht ist, beginnt der Stanzvorgang. B1 und B2 sind die Endlagenschalter der Stanze. B4 ist der Endlagensensor des Förderbandes. Wenn die Kiste voll ist, stoppt die Anlage. Nach dem manuellen Entleeren der Kiste wird die Anlage mit *Automatik Start* wieder neu gestartet. Die Anlage kann auch im Handbetrieb gesteuert werden.

Technologieschema:



Visualisierungsdatei: *10_07_Stanze.plc1ab* im digitalen Zusatzmaterial

Zuordnungsliste:

Symbol	Operand	Kommentar	Schaltverhalten
S1	E0.0	Taster Steuerung ein	Schließer
S2	E0.1	Taster Steuerung aus	Öffner
S3	E0.2	Schalter Automatik/Handbetrieb	0 = Auto, 1 = Hand
S4	E0.3	Automatikbetrieb Start	Schließer
S5	E0.4	Taster Band Handbetrieb/Tippbetrieb	Schließer
S6	E0.5	Taster Stanze abwärts/Handbetrieb	Schließer
S7	E0.6	Taster Stanze aufwärts/Handbetrieb	Schließer
B1	E0.7	Endschalter Stanze oben	Öffner
B2	E1.0	Endschalter Stanze unten	Öffner
B3	E1.1	Endschalter Werkstück unter Stanze	Schließer
B4	E1.2	Endschalter Band Ende	Schließer
P1	A0.0	Meldeleuchte Anlage bereit	-
P2	A0.1	Meldeleuchte Kiste voll	-
Q1	A0.2	Motorschütz Band	-
M1	A0.3	Stanze abwärts	-
M2	A0.4	Stanze aufwärts	-
P3	AD32	Anzeige fertige Werkstücke	-

Aufgabe:

- Legen Sie ein SPS-Projekt an und erstellen Sie das Steuerungsprogramm.
- Testen Sie das Steuerungsprogramm.

Beispiel für bibliotheksfähige Bausteine

Aufgabe:

Eine Firma hat zwei Lagerhallen mit jeweils einem Lüfter. Diese sollen eine einheitliche automatisierte Steuerung bekommen. Jedes Lagerhaus hat zwei Temperatursensoren.

Symbol	Kommentar	Schaltverhalten
B1-B4	Temperatursensoren	1-Signal, wenn Temperatur zu hoch
Q1, Q2	Motorschütze Ventilatoren	–

a) Erstellen einer bibliotheksfähigen Funktion (FC) nach DIN EN 61131-3

Aufgabe:

Wenn beide Sensoren der Lagerhalle eine zu hohe Temperatur melden, soll der Lüfter eingeschaltet sein.

1. Schritt PLC-Variablen eintragen

In der Projektnavigation wird unter PLC-Variablen eine neue *Variablentabelle Lüftersteuerung* erstellt. Die globalen Variablen mit den absoluten Adressen werden eingetragen.

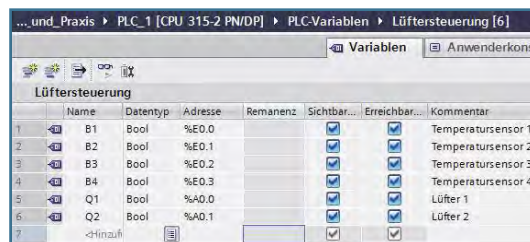


Bild 1: PLC-Variablen (globale Variablen)

2. Funktion erstellen und Variablen definieren

Unter Programmbausteine wird eine neue *Funktion Lüftersteuerung (FC100)* erstellt. Um die Funktion mehrfach aufrufen zu können und um den Anforderungen der DIN EN 61131-3 zu entsprechen, wird diese als bibliotheksfähige Funktion erstellt.

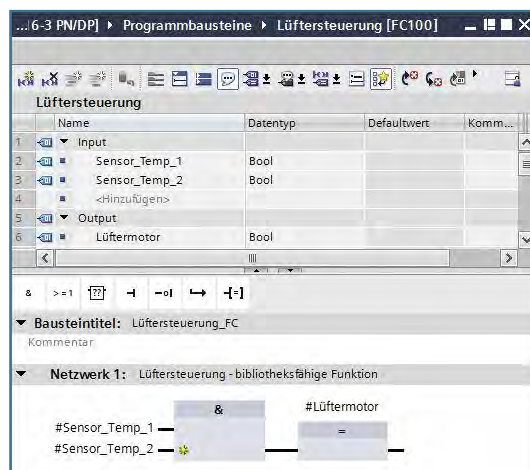


Bild 2: Bibliotheksfähige Funktion (FC100)

Dafür werden im oberen Bereich des Bausteins Input- und Output-Variablen (lokale Variablen) festgelegt. Da es sich um binäre Sensoren und Aktoren handelt, wird der Datentyp BOOL verwendet.

3. Funktion programmieren

Die bibliotheksfähige Funktion besteht nur aus einer UND-Verknüpfung. Wenn beide Sensoren ein 1-Signal haben, wird der Lüfter angesteuert.

4. Funktion aufrufen und parametrieren

Im Organisationsbaustein OB1 wird die Funktion Lüftersteuerung (FC100) zwei Mal aufgerufen. Sie wird jeweils mit den absoluten Adressen der Ein- und Ausgänge (globale Variablen) parametrieren.

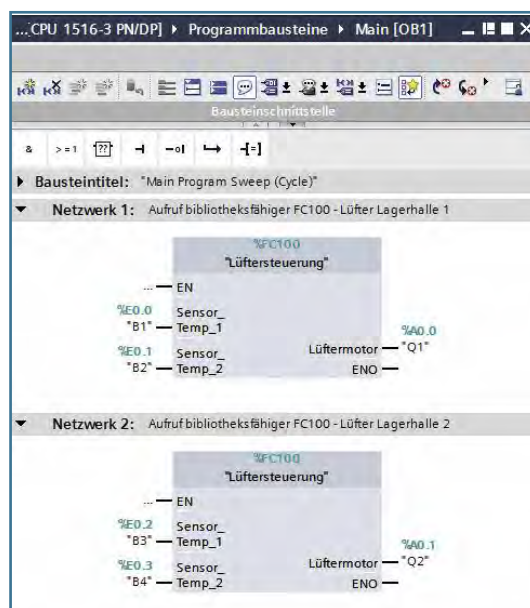
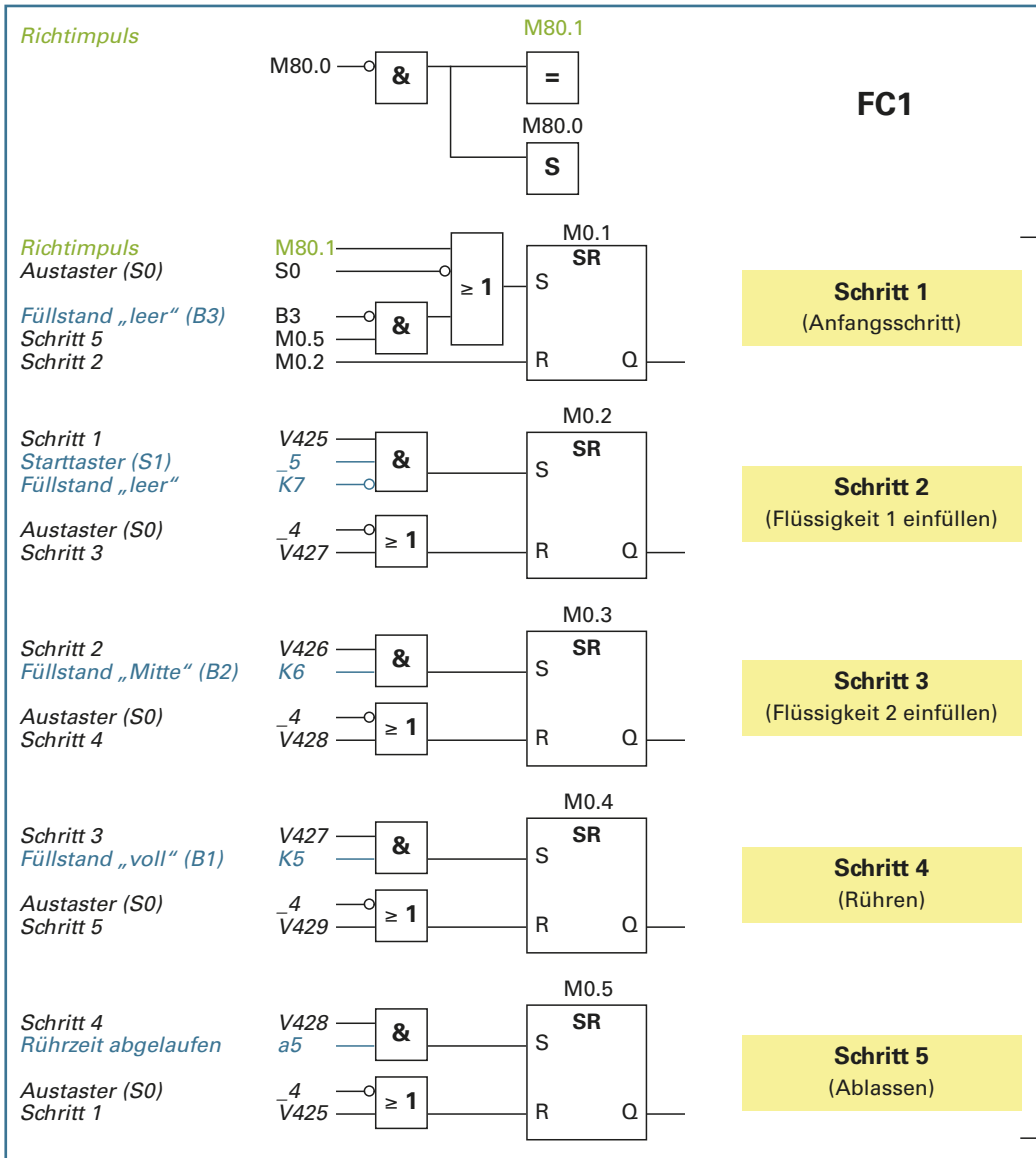
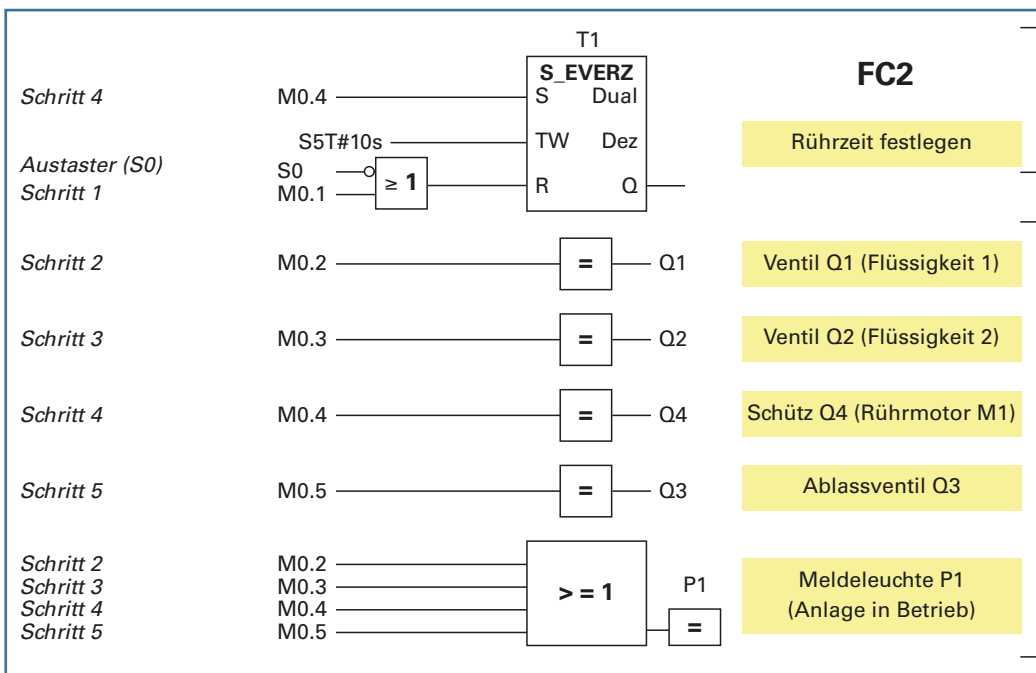


Bild 3: Aufruf des FC100 und Parametrierung mit den globalen Variablen im OB1

Ablaufsteuerung zum Beispiel Rührbehälter



Programmierung der Programmschritte



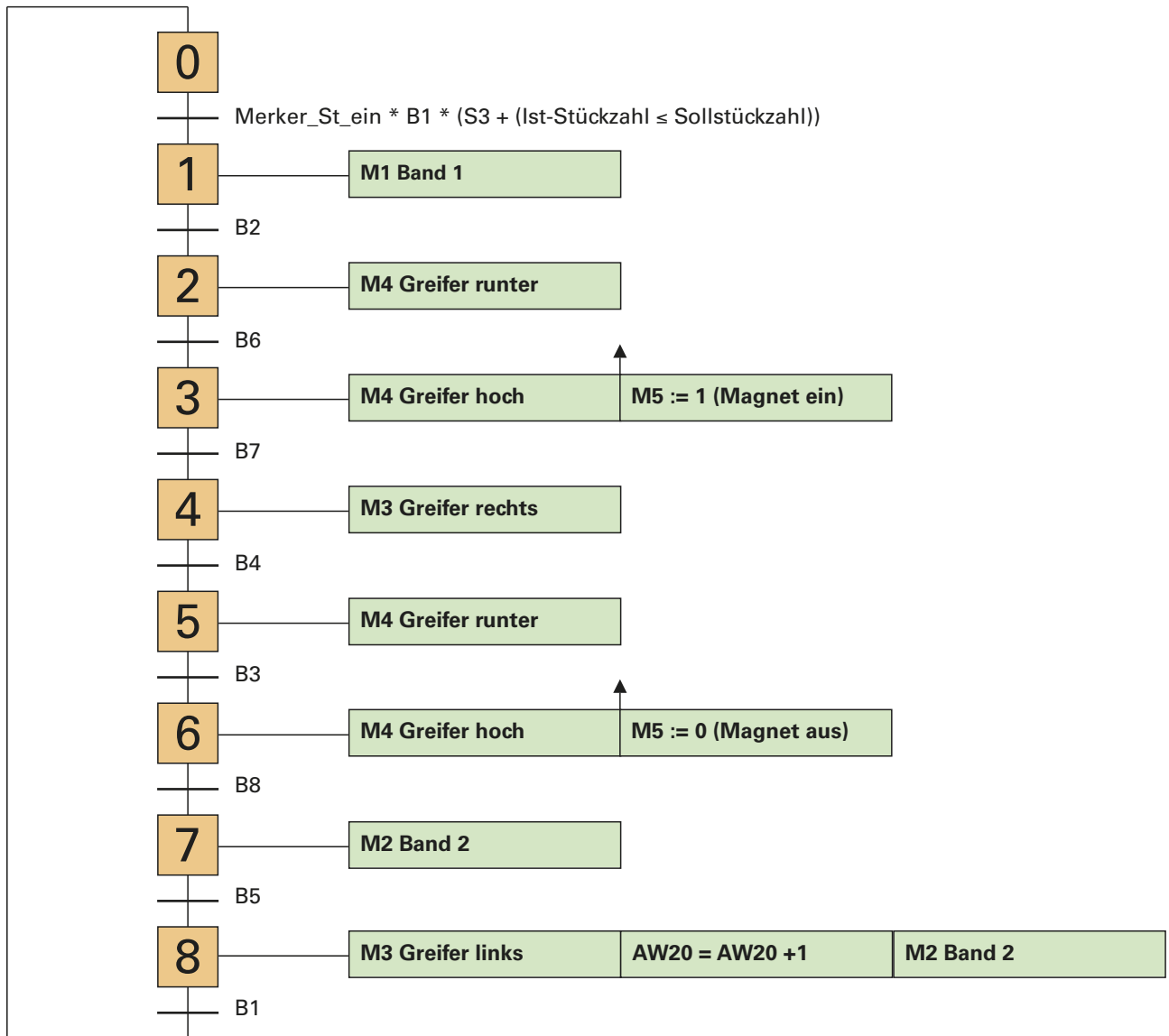
Definition von Zeiten

Programmierung der Ausgänge

Aufgaben:

1. Laden Sie das fehlerhafte Step7-Projekt **14_Verpa** aus dem digitalen Zusatzmaterial.
2. Suchen Sie die Fehler.
3. Notieren Sie sie in der Fehlerliste.
4. Korrigieren Sie das Programm.

Ablaufkette (Funktionsplan):



Fehlerliste:

Baustein, z.B. FC1	Netzwerk Nr.	Fehlerbeschreibung

17 Cause-Effect-Matrix (CEM) und Structed Control Language (SCL)

17.1 Cause-Effect-Matrix

Die Programmiersprache Cause-Effect-Matrix (CEM) bietet eine grafische Programmierumgebung. Sie ist geeignet, wenn Verriegelungs-, Freigabe- und Überwachungsaufgaben realisiert werden sollen. In einer Matrix werden Ursachen (**Causes**) und Wirkungen (**Effects**) dargestellt. Die einzelnen Felder der Matrix werden **Intersections** genannt. Darin wird festgelegt, ob und wie ein Effect auf ein Cause erfolgt.

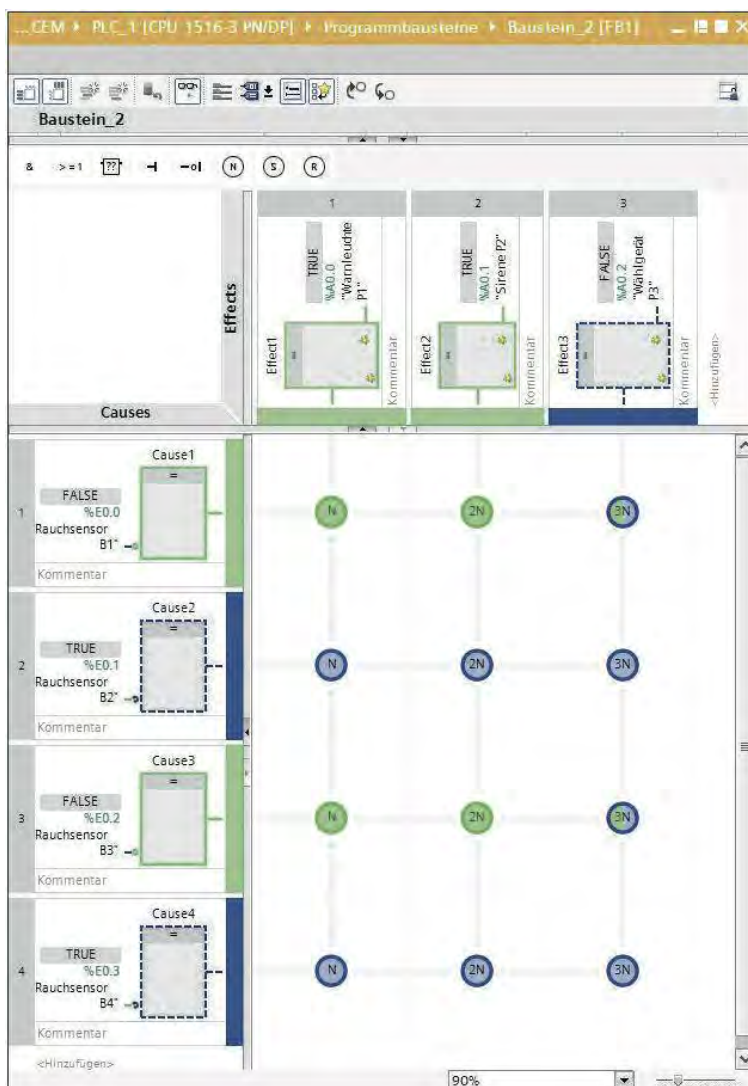
Die Cause-Effect-Matrix steht ab TIA-Portal V17 für S7-1500-Steuerungen ab Firmware V2.6 und in S7-1200-Steuerungen ab Firmware V4.2 zur Verfügung.

Beispiel 1:

Anhand der Beispielaufgabe 5.13 Rauchmeldeanlage auf Seite 36 wird eine Cause-Effect-Matrix erklärt.

Problemstellung

Eine Lagerhalle soll durch vier Rauchsensoren (B1 – B4) überwacht werden. Bei Rauch geben sie ein 0-Signal, sonst ein 1-Signal. Wenn ein Sensor Rauch erkennt, soll eine Warnleuchte (P1) leuchten. Bei mindestens zwei gleichzeitigen Meldungen soll die Sirene (P2) ertönen. Geben mindestens drei Rauchmelder einen Alarm, soll ein automatisches Wählgerät eingeschaltet werden, das die Feuerwehr alarmiert. Das Wählgerät (P3) wird über einen Ausgang der SPS angesteuert (Ausgang auf „1“).



Causes (Ursachen):

Causes sind Ursachen für ein Ereignis. Das können einzelne Eingänge oder aber auch Verknüpfungen von mehreren Bedingungen sein. Zudem sind auch Vergleiche und Timer möglich. Causes werden von oben nach unten abgearbeitet.

Effects (Wirkungen):

Als Wirkungen können Ausgänge, Merker oder Variablen zugewiesen, gesetzt oder zurückgesetzt werden.

Intersections:

Die Abhängigkeit des Effects von den Causes wird in einer Intersection beschrieben. Darin kann festgelegt werden:

N: Setzen, solange Cause aktiv (nicht speichern)

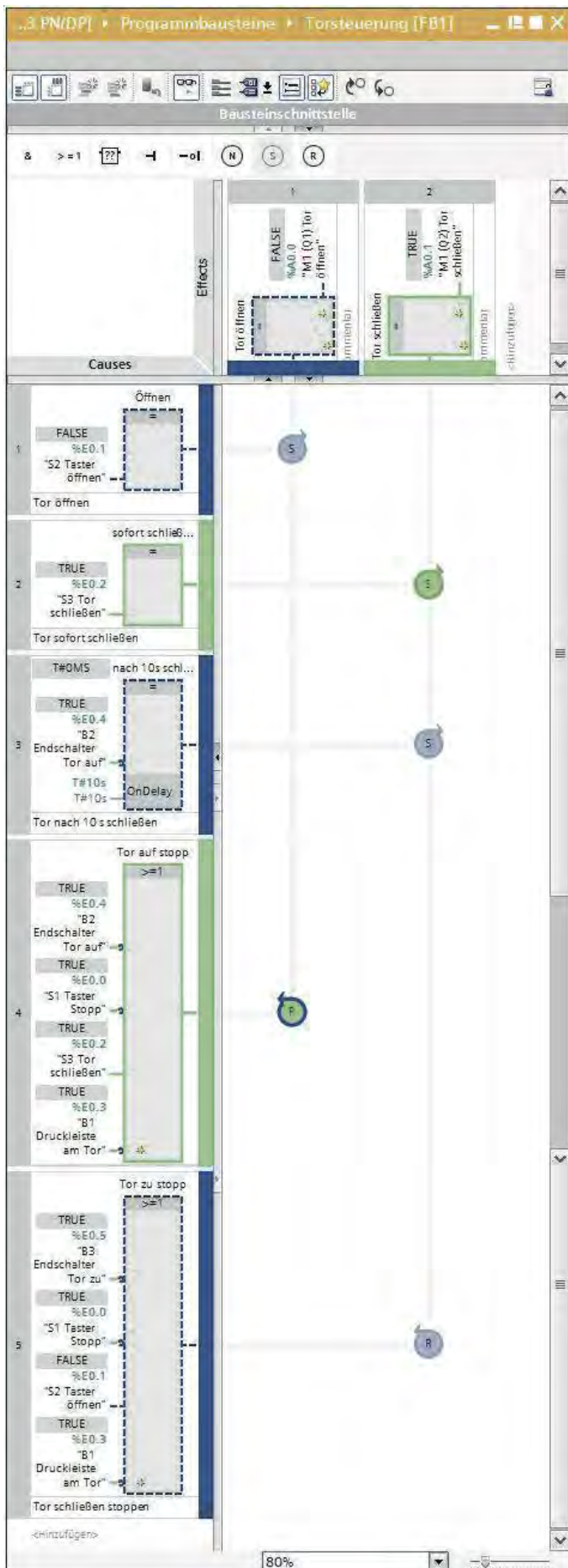
S: Speichernd setzen auf 1

R: Speichernd rücksetzen auf 0

Aktionsgruppe: z. B. 2 aus 3 oder 2 aus 4

Intersections werden von links nach rechts abgearbeitet.

Im Beispiel wird die Meldeleuchte P1 (Effect1) über die Bedingungen N so lange angesteuert, wie mindestens einer der vier Causes ein 1-Signal hat. Die Sirene P2 wird über die Aktionsgruppe 2N nicht speichernd angesteuert. Hier gilt: 2 aus 4. Wenn mindestens zwei der vier Causes ein 1-Signal haben, wird der Effect 2 angesteuert. Beim Wählgerät gilt die Bedingung 3N (3 aus 4).

**Beispiel 2:**

Anhand der Beispielaufgabe 8.9 *Zeitgesteuerte Toranlage* auf Seite 61 werden die Setz- und Rücksetzbe-
fehle sowie die Zeitfunktionen bei der Cause-Effect-
Matrix veranschaulicht.

Problemstellung

Das Tor einer Hofeinfahrt wird über den Taster „Tor öffnen“ geöffnet. Nach 10 s schließt das Tor automa-
tisch. Über den Taster „Stopp“ kann das Tor ange-
halten werden. Zudem kann das Tor jederzeit mit
dem Taster „Schließen“ geschlossen werden. Ein di-
rektes Umschalten von *Öffnen* in *Schließen* soll
möglich sein. Wird beim Öffnen oder Schließen die
Druckleiste betätigt, bleibt das Tor stehen.

Bei der Beispielaufgabe werden die verschiedenen
Möglichkeiten der Cause-Effect-Matrix verdeutlicht:

Causes:

Cause 1 und Cause 2: Abfrage eines Einganges

Cause 3: Eingang mit Negation und Einschaltverzöge-
rung

Cause 4 und Cause 5: ODER-Verknüpfungen

Intersections:

Setz- und Rücksetzbe-
fehle für die einzelnen Effects

Die Cause-Effect-Matrix bietet die Möglichkeit einfache
Überwachungs-, Verriegelungs- und Steuerungsauf-
gaben zu realisieren. Bei komplexeren Aufgabenstel-
lungen kommt die CEM aber sicherlich an ihre Gren-
zen.

Zudem müssen unter anderem folgende Einschrän-
kungen (in TIA V17) hingenommen werden:

- der Aufruf anderer Bausteine ist aus dem Programm
heraus nicht möglich
- CEM-Bausteine sind nicht multiinstanzfähig
- eine implizierte Datenkonvertierung ist nicht möglich
- in fehlersicheren Steuerungen können CEM-Bausteine
nicht verwendet werden
- CEM-Bausteine können nicht ausgedruckt werden

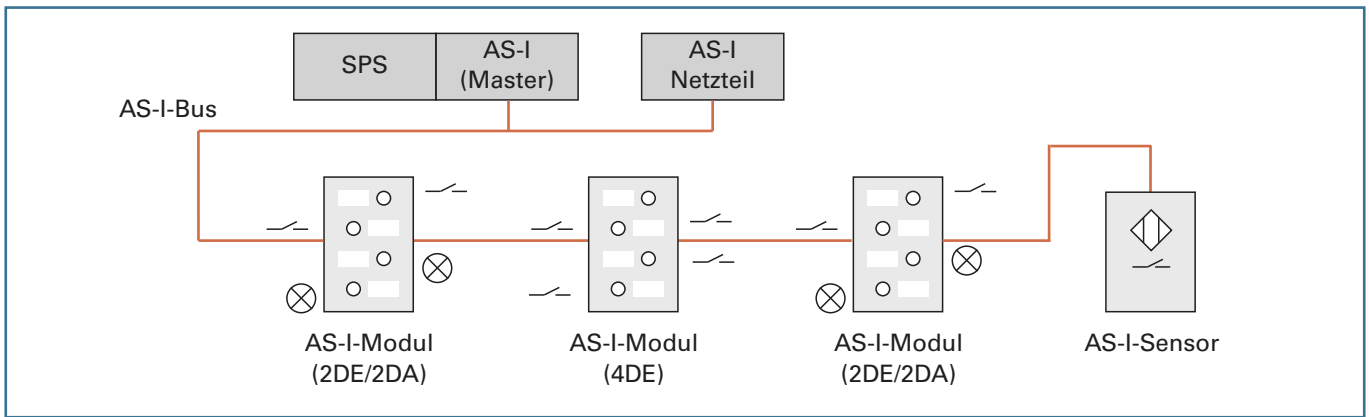


Bild 1: AS-Interface-System

18.8 Industrie 4.0

18.8.1 Was ist Industrie 4.0

Es gibt viele verschiedene Beschreibungen des Begriffs Industrie 4.0. Im Folgenden soll ein Überblick über das Thema und seine Bedeutung gegeben werden.

1. Industrielle Revolution	Ende 18. Jahrhundert	Einführung Dampf- und Wasserkraft als Antrieb für mechanische Produktionsanlagen	Erste mechanische Webstühle
2. Industrielle Revolution	Beginn des 20. Jahrhundert	Elektrische Energie als Antrieb Beginn der Massenproduktion von Gütern	Erste Fließbandproduktionen
3. Industrielle Revolution	Ende der 1970er Jahre	Einführung von Speicherprogrammierbaren Steuerungen	Automatisierung von Produktionsanlagen
4. Industrielle Revolution	Heute	Komplexe vernetzte Steuerung von cyber-physikalischen Systemen	Cloudbasierte vernetzte Steuerungen, Internet of Things (IOT)

Komplexität

Merkmale von Industrie 4.0-Anlagen:

- Intelligente vernetzte Fabriken (Smart Factory)
- Cyber-physikalische Systeme (CPS)
- Digitalisierung der Produktion
- Flexibel und wandlungsfähig
- Jedes Produkt ist individuell (Losgröße 1)
- Vertikale Vernetzung der Prozesssteuerung
- Anbindung an ein MES- System (Produktionsplanung) und ein ERP-System (Warenwirtschaftssystem)
- Sich selbst steuernde Produktion und Logistik
- Alle Teilnehmer sind vernetzt und tauschen selbstständig Informationen aus
- Integration von Zulieferern und Kunden in die Wertschöpfungskette
- IT-Sicherheit
- Digitaler Zwilling
- Verzahnung von Automatisierungstechnik mit IT und kaufmännischen Prozessen

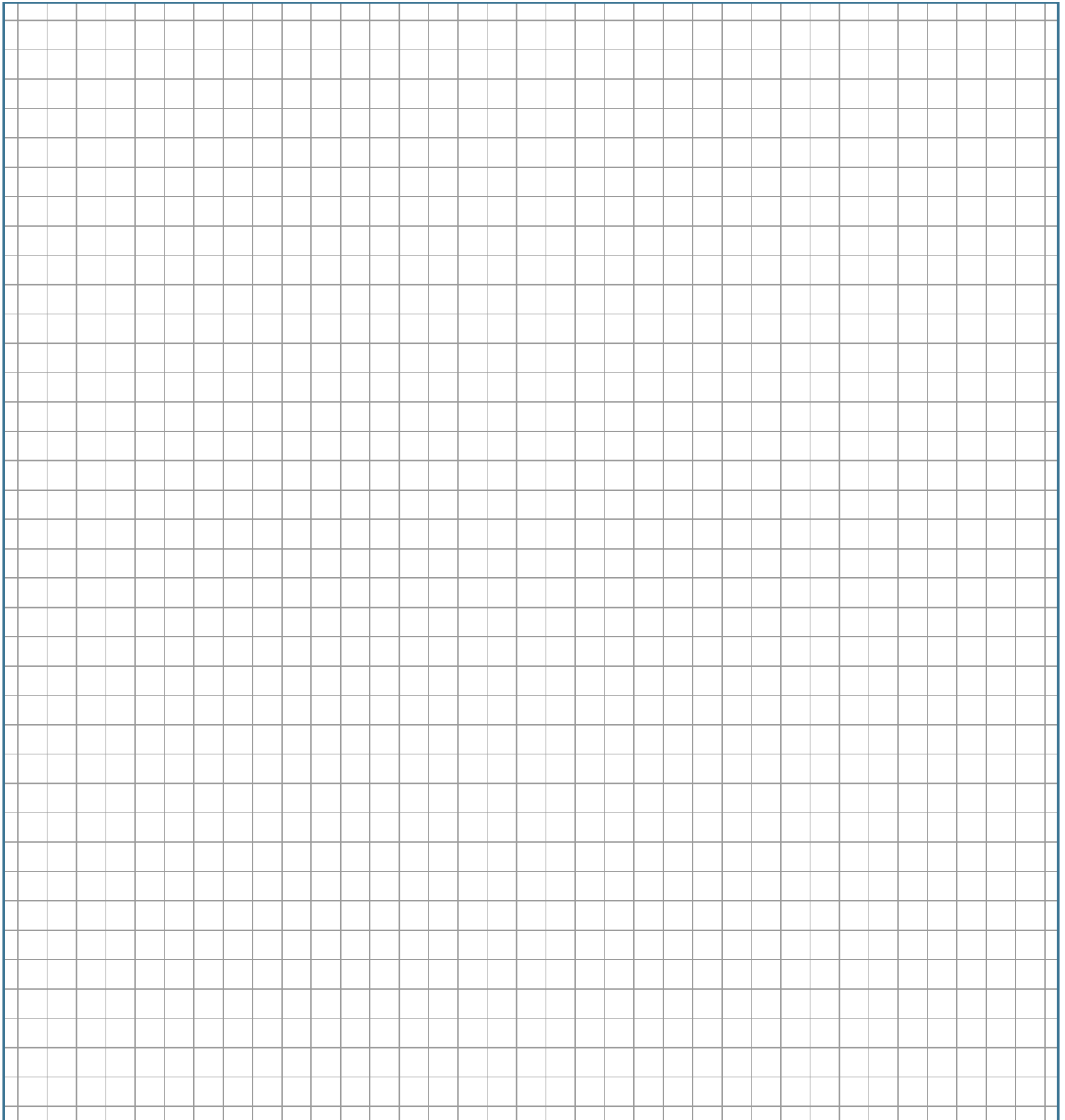
Ein gutes Beispiel für eine Industrie 4.0-Produktion ist die Automobilindustrie. Zu Beginn der industriellen Autoproduktion wurden gleiche Autos am Fließband produziert, die der Kunde dann kaufen konnte.

Heute bestellt der Kunde bei seinem Autohändler ein individuelles nach seinen Wünschen ausgestattetes Auto. Die Bestellung wird online an das ERP-System des Autoherstellers weitergegeben. Das System ordert dann bei den Zulieferern die entsprechenden Teile, die just-in-time vom MES-System gesteuert an das Produktionsband geliefert werden. Dort werden nacheinander Autos von verschiedenen Modellen und komplett unterschiedlicher Ausstattung produziert (Losgröße 1). Die Produktion ist durch SPSen und Robotertechnik stark automatisiert. Die IT-Infrastruktur sorgt für eine Verzahnung und Dokumentation der Prozesse. Durch IOT-Gateways können Prozessdaten an übergeordnete Systeme weitergeleitet und in einem digitalen Zwilling angezeigt werden.

Bei den cloudbasierten global verteilten Prozessteilnehmern ist ein Höchstmaß an IT-Sicherheit zu gewährleisten.

Ablaufkette:

Aufgabe 4 Skizzieren Sie die Ablaufkette nach GRAFCET.



Programmerstellung:

Aufgabe 5 Erstellung Sie das Steuerungsprogramm auf Grundlage der Ablaufkette.

Simulation:

Aufgabe 6 Erstellen Sie eine Simulation der Kreuzung mit SPS-VISU und testen Sie die Steuerung. Alternativ können Sie das Steuerungsprogramm mit PLCSIM testen.