



EUROPA-FACHBUCHREIHE
für elektrotechnische Berufe

SPS

Theorie und Praxis

mit Übungsaufgaben

6. 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 978-3-8085-3817-3

6. Auflage 2020
Druck 5 4 3 2 1

Alle Drucke derselben Auflage sind parallel einsetzbar, da sie bis auf die Behebung von Druckfehlern untereinander unverändert sind.

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

© 2020 by Verlag Europa-Lehrmittel, Nourney, Vollmer GmbH & Co. KG, 42781 Haan-Gruiten
<http://www.europa-lehrmittel.de>

Umschlag: Media-Creativ, 40723 Hilden
Satz und Layout: rkt, 51379 Leverkusen
Druck: Himmer GmbH, 86167 Augsburg

Inhalt der CD zum Buch:

CD: 1. Visualisierungsdateien zu den Aufgaben des Buches

2. SPS-Projekte zur Fehlersuche
(Siehe Kapitel 14 Fehlersuche)

3. Videos zu den Aufgaben und Tutorials

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 Technikschule** 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 6. 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 Programmiersprachen wie Structured Control Language und S7-Graph zu erkennen. Auf die wachsenden Anforderungen wird in der neuen Auflage eingegangen.

Es ist ein neues Kapitel **Vernetzte Automatisierungssysteme** entstanden. Neben den Grundlagen der Netzwerkechnik wird auf Merkmale und Typen von Bussystemen eingegangen. Beim Themenbereich **Industrie 4.0** werden der Aufbau und die Komponenten dieser Anlagen beschrieben. Zudem findet eine Klärung der Begrifflichkeiten im Bereich Vernetzung und Industrie 4.0 statt.

Das neue Kapitel **Structured Control Language (SCL)** gibt eine Übersicht über die SPS-Programmierung mit dieser Hochsprache. Im Bereich der Ablaufsteuerungen ist der Bereich **GRAFCET** um Strukturierungsbefehle, wie Makros usw. ergänzt worden. Die Kapitel Programmiersprache **S7-GRAPH** und **Analogwertverarbeitung** sind neu überarbeitet worden.

Bislang konnten ein Großteil der Aufgaben mit der Visualisierungssoftware SPS-VISU simuliert werden. Wegen anhaltender Windowsprobleme war ein Umstieg auf eine **neue Visualisierungs- und Simulationssoftware** notwendig. **Die Aufgaben können jetzt mit der neuen Software PLC-Lab simuliert werden.** Neben besserer Grafik bietet die Software auch mehr Funktionalitäten. Wer zum Simulieren die Vollversion nicht besitzt, kann auch eine kostengünstige Runtime-Lizenz oder eine kostenlose 30-Tage-Demoversion bei der Fa. MHJ-Software erwerben. Für Benutzer, die weiterhin mit SPS-VISU arbeiten möchten, stehen natürlich alle Dateien nach wie vor auf der dem Buch beiliegenden CD zur Verfügung. Die Aufgaben können mit beiden Systemen simuliert werden.

Neu ist die Möglichkeit, sich **Videos zu den Anleitungen und den Aufgabensimulationen anzusehen**. Dazu stehen **QR-Codes** im Buch zur Verfügung.

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

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	20
3.1 Vorgehensweise bei der Projektbearbeitung	20
3.2 TIA-Portal: Erstellen eines Projektes	21
4 Simulation von Programmen	24
4.1 Simulation mit PLCSIM	24
4.2 Simulation mit PLC-Lab	25
5 Grundverknüpfungen	26
5.1 Programmiersprachen/Darstellungsarten	26
5.2 Grundlagen der Grundfunktionen	27
5.3 Übersicht der Grundfunktionen	28
5.4 Grundverknüpfungen in verschiedenen Programmiersprachen	29
5.5 Addressierung	30
5.6 Merker	30
5.7 Verknüpfungsergebnis VKE	31
5.8 Beispielaufgabe: Kühlhaus	32
5.9 Wiederholungsfragen	34
5.10 Übung: Sicherheitscode	36
5.11 Übung: Folgeschaltung von Montagebändern	36
5.12 Übung: Funktionsgleichung	36
5.13 Übung: Rauchmeldeanlage	37
5.14 Übung: Alarmanlage	38
5.15 Übung: Förderbandanlage	39
6 Flipflops (Speicherfunktionen)	41
6.1 SR-Flipflop und RS-Flipflop	41
6.2 Beispielaufgabe: Ansteuerung eines Drehstrommotors	43
6.3 Wiederholungsfragen	45
6.4 Übung: Doppelt wirkender Zylinder	47
6.5 Übung: Wendeschützschaltung	47
6.6 Übung: Förderbandanlage (Folgeschaltung)	48
6.7 Übung: Toranlage	49
6.8 Übung: Sortieranlage	50
7 Strukturierte Programmierung	52
7.1 Lineare Programmierung	52
7.2 Strukturierte Programmierung	52
7.3 Bausteinarten	53
7.4 Wiederholungsfragen	53
8 Zeitfunktionen	54
8.1 SIMATIC-Zeiten	54
8.2 IEC-Zeiten	56
8.3 Taktmerker	57
8.4 Beispielaufgabe: Pneumatische Abfülleinrichtung	58
8.5 Wiederholungsfragen	60
8.6 Übung: Störungslampe (Taktmerker)	61
8.7 Übung: Industrieofen	61

8.8	Übung: Automatische Stern-Dreieck-Schaltung	61
8.9	Übung: Zeitgesteuerte Toranlage	62
8.10	Übung: Zeitgesteuerte Förderbandanlage	63
9	Bit, Byte, Wort, Doppelwort	64
9.1	Zahlensysteme	64
9.2	Definitionen	65
9.2.1	Bit	65
9.2.2	Byte	65
9.2.3	Wort	65
9.2.4	Doppelwort	66
9.3	Lade- und Transferoperationen	66
9.4	Wiederholungsfragen	68
9.5	Übung: Wortverarbeitung	70
10	Zähler und Vergleicher	71
10.1	SIMATIC-Zähler	71
10.2	Vergleicher	72
10.3	SIMATIC-Vorwärts-/Rückwärtzähler mit Vergleicher in AWL, FUP und KOP	73
10.4	IEC-Zähler	75
10.5	Wiederholungsfragen	76
10.6	Übung: Parkplatzampel	78
10.7	Übung: Stanze	79
11	Verschiedene Programmfunctionen und Befehle	80
11.1	Urlöschen	80
11.2	Systemmerker	80
11.3	Archivieren/Dearchivieren	80
11.4	Flankenauswertung	80
11.5	Sprungoperationen	81
11.6	Wiederholungsfragen	82
12	Bausteine	83
12.1	Bausteinarten	83
12.1.1	Organisationsbausteine (OB)	83
12.1.2	Funktionen (FC)	83
12.1.3	Funktionsbaustein (FB)	83
12.1.4	Systemfunktionen und Systemfunktionsbausteine	83
12.1.5	Datenbausteine (DB)	83
12.2	Bibliotheksfähige Bausteine	84
12.3	Anlegen einer eigenen Bibliothek	87
12.4	Datenbausteine	88
12.5	Wiederholungsfragen	90
12.6	Übung: Motorsteuerung mit bibliotheksfähigen Bausteinen	91
13	Ablaufsteuerungen	93
13.1	Grundlagen zu Ablaufsteuerungen	93
13.2	Darstellung von Schnittkanten	95
13.2.1	Grafset und Din En 61131-3	95
13.2.2	Strukturierung von GRAFCETs	99
13.3	S7-Graph	100
13.4	Betriebsarten	101
13.5	Wiederholungsfragen	102
13.6	Übung: Leuchtreklame	105
13.7	Übung: Schwimmbad	105
13.8	Übung: Bohranlage	106
13.9	Übung: Ampelsteuerung	108
14	Fehlersuche	110
14.1	Fehlerarten	110
14.2	Fehlersuche bei Hardwarefehlern	110
14.3	Fehlersuche bei Softwarefehlern	110
14.4	Fehler-Operationsbausteine	113
14.5	Wiederholungsfragen	113

14.6	Übung: Förderbandanlage (Fehlersuche)	115
14.7	Verpackungsanlage (Fehlersuche)	117
15	Mathematische Funktionen 119	
15.1	Datentypen	119
15.2	Umwandlungsfunktionen	120
15.3	Rechnen mit Ganzzahlen (INT und DINT)	121
15.5	Übung: Umwandlungsfunktionen	122
15.6	Übung: Mathematische Operationen	123
16	Verarbeitung von Analogwerten 124	
16.1	Analoge Signale	124
16.2	Analogwerte einlesen und ausgeben	125
16.3	Analogwerte einlesen und normieren	125
16.4	Analogwerte normieren und ausgeben	127
16.5	Wiederholungsfragen	128
16.6	Übung: Temperaturanzeige	129
16.7	Übung: Temperaturüberwachung	130
16.8	Pegelmessung an einem Wasserkraftwerk 1	131
16.9	Pegelmessung an einem Wasserkraftwerk 2	132
17	Structured Control Language (SCL) 133	
17.1	SCL-Befehle	133
17.2	Wiederholungsfragen	136
17.3	Übung: Zeitgesteuerte Toranlage in SCL	137
18	Vernetzte Automatisierungssysteme 138	
18.1	Hierarchischer Aufbau von Automatisierungssystemen	138
18.2	Grundlagen Netzwerktechnik	138
18.2.1	Aufbau eines kleinen Automatisierungsnetzwerk	138
18.2.2	Netzwerkarchitekturen	139
18.2.3	Konfiguration eines Netzwerkes	139
18.3	Topologien	141
18.4	Übertragungsmedien	142
18.5	Störgrößen bei leitungsgebundener Datenübertragung	142
18.6	Buszugriffsverfahren	143
18.7	Industrielle Bussysteme	144
18.7.1	Ethernet TCP/IP	144
18.7.2	Industrial Ethernet	144
18.7.3	PROFINET	144
18.7.4	Profibus DP	145
18.7.5	Aktor-Sensor-Interface (AS-I)	145
18.8	Industrie 4.0	146
18.8.1	Was ist Industrie 4.0	146
18.8.2	Aufbau einer Industrie 4.0-Anlage	147
18.9	Wiederholungsfragen	148
19	Projektaufgaben 150	
19.1	Übung: Motorsteuerung mit bibliotheksfähigen Bausteinen	150
19.2	Übung: Ampelanlage	152
19.3	Übung: Lackierstraße	156
19.4	Übung: Autowaschanlage	158
20	Übersicht Befehle unter Step7 161	
22	Sachwortverzeichnis 163	
Bild und Textquellen		165
Literatur und Downloads		165

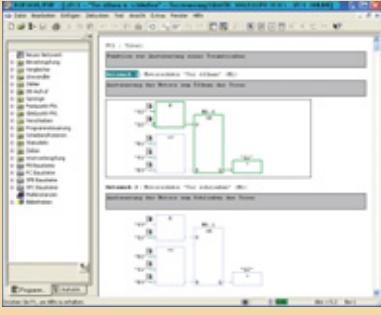
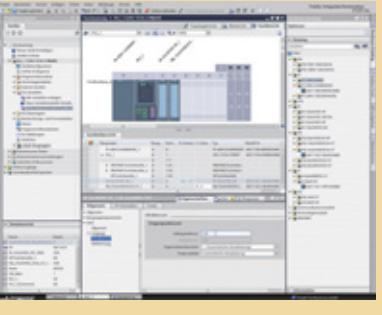
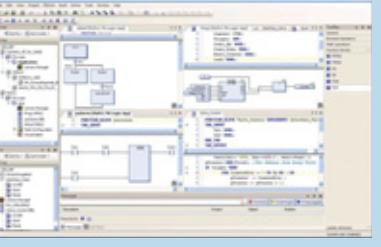
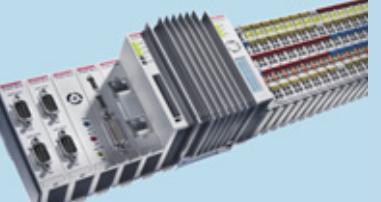
1.3 SPS-Bezeichnung

international PLC Programmable Logic Controller	↔	deutsch SPS Speicherprogrammierbare Steuerung
--	----------	--

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	Step 7 V5.x 	TIA-Portal ab V11 	CoDeSys 
Eingänge/ Ausgänge	E 0.0 A 0.0	% E 0.0 % A 0.0	% IX 0.0 % QX 0.0
	S7-200 S7-300 S7-400	S7-300 S7-400 S7-1200 S7-1500	z.B.: Beckhoff WAGO Festo
Hardware			

3.2 TIA-Portal: Erstellen eines Projektes

1. S7-Projekt anlegen

Programm öffnen

Das TIA-Portal wird über einen Doppelklick auf das Symbol „TIA-Portal“ geöffnet.

Dabei erscheint die **Portalansicht** des Projektes.

Es kann zwischen **Bestehendes Projekt öffnen**, **Neues Projekt anlegen** und **Projekt migrieren** gewählt werden. Beim Migrieren wird ein Projekt, das mit Step7 V5.x erstellt worden ist, in ein TIA-Projekt umgewandelt.

Wählen Sie **Neues Projekt anlegen** und geben Sie den **Projektname** und den **Ablageort** an. Bestätigen Sie mit **Erstellen**.

2. Gerätekonfiguration erstellen

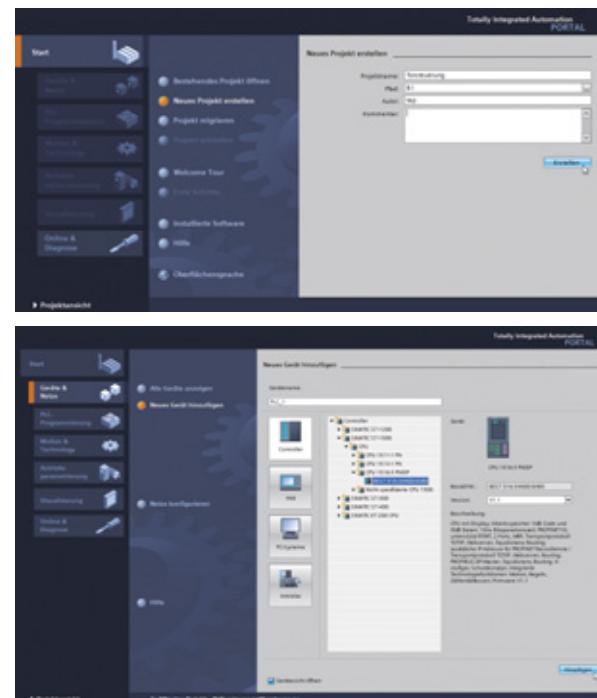
Hardware auswählen

Unter **Neues Gerät hinzufügen** ⇒ **Controller** kann die verwendete CPU, hier eine S7-1500 CPU 1516-3 PN/DP, ausgewählt werden.

Nach dem Betätigen des Button **Hinzufügen** wechselt das Programm in die **Projektansicht**.



TIA-Portal



Projektansicht beim TIA-Portal

Projektstruktur

Arbeitsbereich

Taskcards (abhängig vom Editor)

Detailansicht

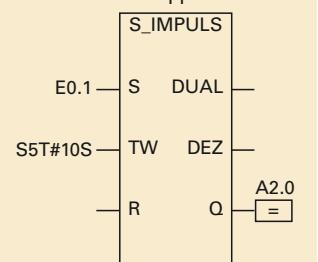
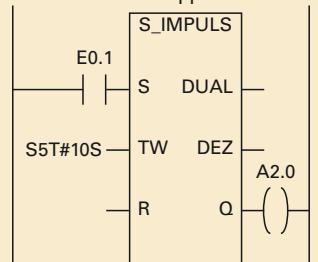
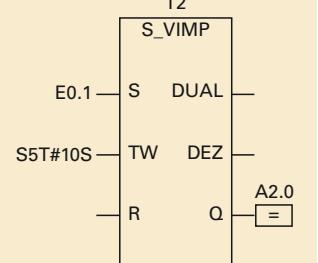
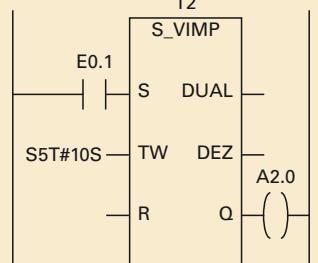
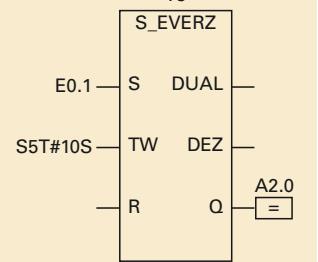
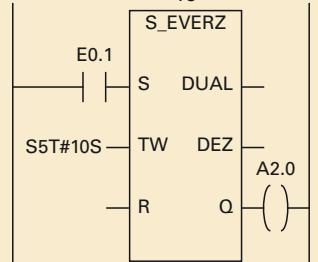
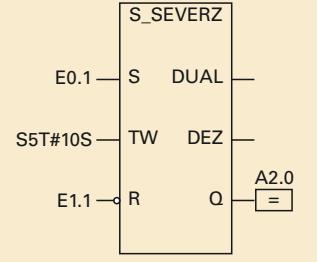
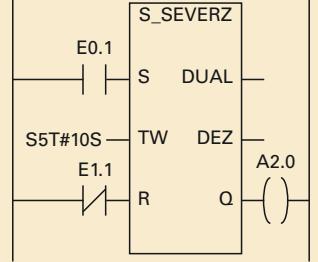
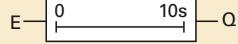
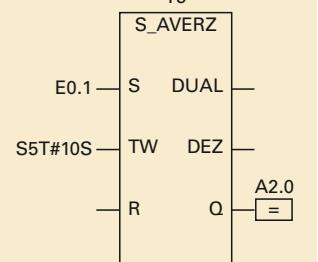
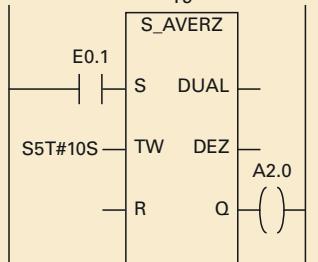
Inspektorenfenster (Eigenschaften des ausgewählten Objekts)

The screenshot displays the TIA-Portal interface with several windows open:

- Projektstruktur (Project Structure):** Shows the project navigation tree with nodes like "Tonsteuerung", "Neues Gerät hinzufügen", "Controller & Netz", and "PLC_1 [CPU 1516-3 PN/DP]".
- Arbeitsbereich (Work Area):** Shows the "Geräte" (Devices) tab selected. It displays a rack diagram with slots 0-7 populated by various modules, and a "Geräteübersicht" (Device Overview) table below it.
- Taskcards (abhängig vom Editor):** Shows a catalog of hardware components, including a list of modules under "Katalog" (Catalog).
- Detailansicht (Detail View):** Shows a detailed view of the selected module "DI 32x24VDC HF_1" (DI 32x24VDC HF_1 [DI 32x24VDC HF_1]). It includes tabs for "Allgemein", "IO-Variablen", and "Texte".
- Inspektorenfenster (Eigenschaften des ausgewählten Objekts) (Inspector Window):** Shows the properties of the selected module, specifically the "Eingangsadressen" (Input Addresses) section.

8 Zeitfunktionen

8.1 SIMATIC-Zeiten

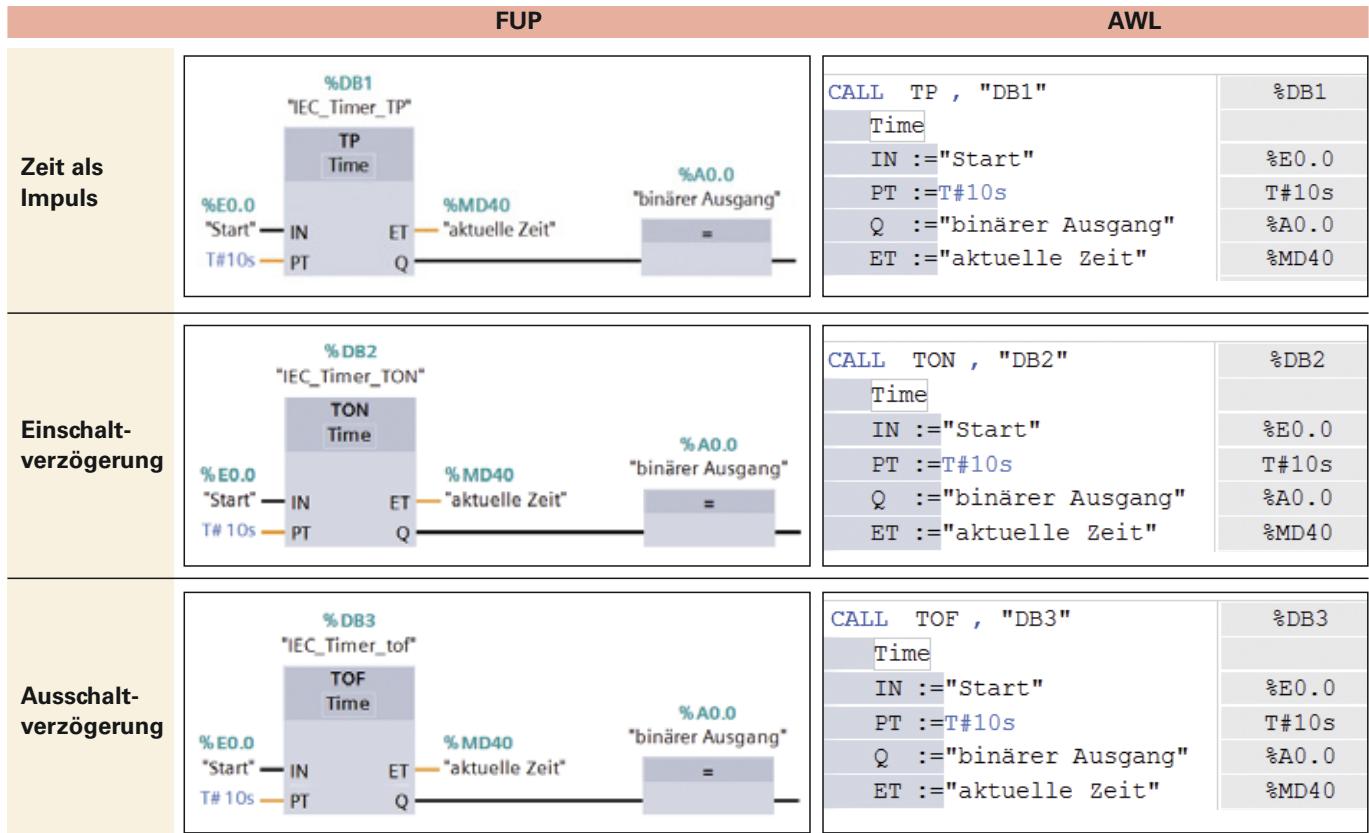
	Symbol allgemein	FUP	Darstellung in Step7 KOP	AWL
Zeit als Impuls				U E0.1 L S5T#10S SI T1 NOP 0 NOP 0 NOP 0 U T1 = A2.0
Zeit als verlängerter Impuls				U E0.1 L S5T#10S SV T2 NOP 0 NOP 0 NOP 0 U T2 = A2.0
Einschalt- verzögerung				U E0.1 L S5T#10S SE T3 NOP 0 NOP 0 NOP 0 U T3 = A2.0
Speichernde Einschalt- verzögerung				U E0.1 L S5T#10S SS T4 UN E1.1 R T4 NOP 0 NOP 0 U T4 = A2.0
Ausschalt- verzögerung				U E0.1 L S5T#10S SA T5 NOP 0 NOP 0 NOP 0 U T5 = A2.0

Die Programmiersprache Structured Control Language (SCL) wird im Kapitel 17 beschrieben.

8.2 IEC-Zeiten

Es gibt drei verschiedene Standard-Zeiten nach IEC 61131-3. Bei einem Wechsel von 0- zum 1-Signal am Starteingang IN werden die Bausteine jeweils aktiviert. An PT wird der Zeitwert im Format Time, z.B. T#10s, vorgegeben. An ET kann der aktuelle Zeitwert im Format Time abgefragt werden.

Die IEC-Zeit-Bausteine sind Systemfunktionsbausteine SFB (siehe Seite 95). Zu jedem SFB gehört ein Instanz-Datenbaustein, in den seine Daten abgelegt werden. Beim Einfügen einer IEC-Zeit öffnet sich ein Dialogfenster „Aufrufoptionen“. Dort kann gewählt werden, ob die IEC-Zeit in einem eigenen Datenbaustein (Einzel-Instanz) oder als lokale Variable (Multi-Instanz) abgelegt wird. Der zu der Zeitfunktion gehörige Datenbaustein wird über dem Baustein eingetragen.



Bei der *Zeit als Impuls* gibt der Ausgang nach einer positiven Flanke an IN für die an PT eingestellte Zeit ein 1-Signal an Q aus. Die *Einschaltverzögerung* schaltet, ausgelöst durch einen positiven Signalwechsel an IN, nach der eingestellten Zeit den Ausgang Q auf ein 1-Signal. Der Ausgang Q der Ausschaltverzögerung wird bei einer positiven Flanke an IN auf eine 1-Signal gesetzt. Bei einer negativen Flanke an IN schaltet er nach der eingestellten Zeit verzögert aus.

Der Zeitwert im Format Time kann Angaben für Tage (d), Stunden (h), Minuten (m), Sekunden (s) und Millisekunden (ms) enthalten. Beispiel: T#11d19h20m30s420ms. Die maximale Zeit beträgt: T#24d_20h_31m_23s_630ms.

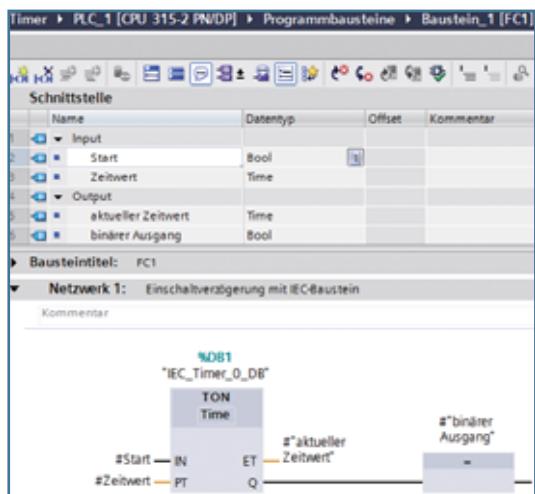


Abb. 1: Programmierung des Rückwärtszählers im FC1

Im nebenstehenden Beispiel sind im FC 1 die Variablen deklariert und eine Einschaltverzögerung programmiert worden. Im OB1 wird die Funktion FC1 mit der Zeitfunktion aufgerufen. Den Variablen werden dabei die absoluten Adressen zugewiesen. Nach einer positiven Flanke von E0.0 wird eine Einschaltverzögerung gestartet. A0.0 schaltet nach 10s auf ein 1-Signal.

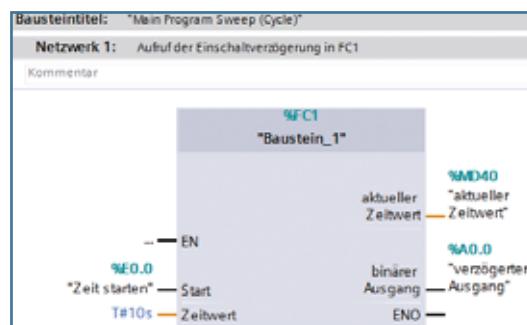


Abb. 2: Aufruf des FC1 im OB1

8.10 Übung: Zeitgesteuerte Förderbandanlage



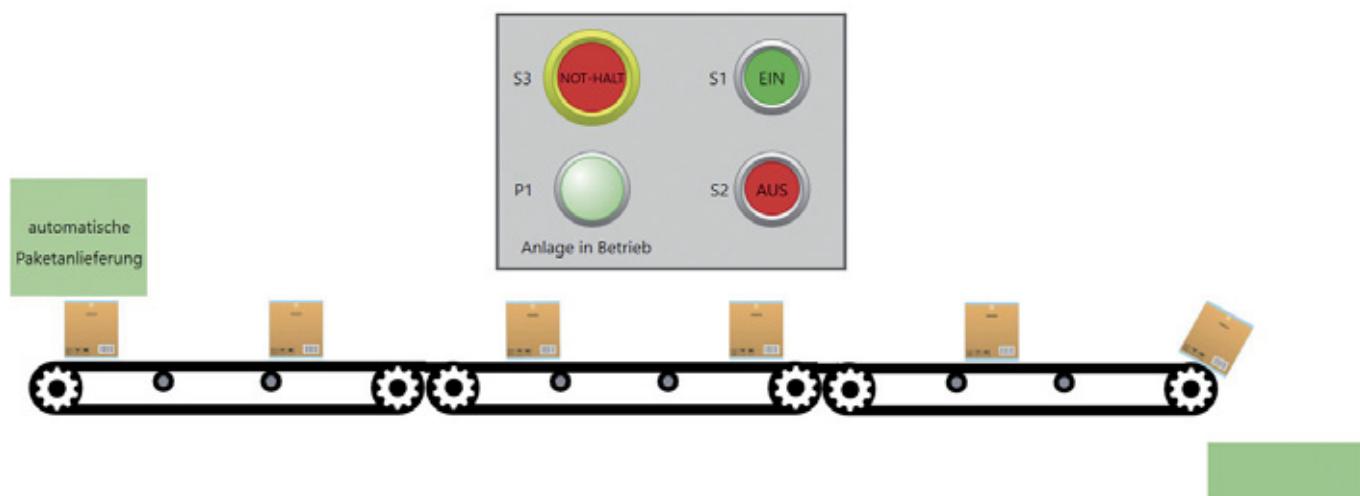
Problemstellung

Über drei Förderbänder sollen Kisten transportiert werden. Um Staus auf den Bändern zu vermeiden, soll nach dem Betätigen des Start-Tasters zuerst Band 3 anlaufen. 5 Sekunden später startet Band 2, weitere 5 Sekunden später Band 1.

Das Ausschalten erfolgt in umgekehrter Reihenfolge. Zuerst stoppt Band 1, 10 Sekunden später Band 2, weitere 10 Sekunden später Band 3.

Bei Betätigung des Not-Aus-Tasters müssen alle Antriebe sofort stillgesetzt werden. Die Meldeleuchte *Anlage in Betrieb* leuchtet, sobald eines der Förderbänder läuft.

Technologieschema:



Visualisierungsdatei: 08_10_Förderbandanlage auf der Buch-CD

Zuordnungsliste:

Symbol	Operand	Kommentar	Schaltverhalten
S1	E0.0	Taster Start	Schließer
S2	E0.1	Taster Stopp	Öffner
S3	E0.2	Taster NOT-HALT	Öffner
M1 (Q1)	A0.0	Motorschütz Band 1	–
M2 (Q2)	A0.1	Motorschütz Band 2	–
M3 (Q3)	A0.2	Motorschütz Band 3	–
P1	A0.3	Meldeleuchte Anlage in Betrieb	

Aufgabe:

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

10.5 Wiederholungsfragen

- 1 Wann schaltet der Ausgang eines IEC-Zählers von einem 0-Signal auf ein 1-Signal um

- 2 An welchem Ausgang kann der Zählstand BCD-codiert abgefragt werden?

- 3 Stellen Sie folgende Schaltung als Funktionsplan (FUP) dar.

Netzwerk 1
U E124.0
ZV Z12
U E124.1
L C#100
S Z12
U E124.2
R Z12
L Z12
T MW6
NOP 0
NOP 0

Netzwerk 2
L MW6
L 200
==|
= A124.0

- 4 Welchen minimalen und maximalen Zählerwert haben SIMATIC- und IEC-Zähler?

10.6 Übung: Parkplatzampel



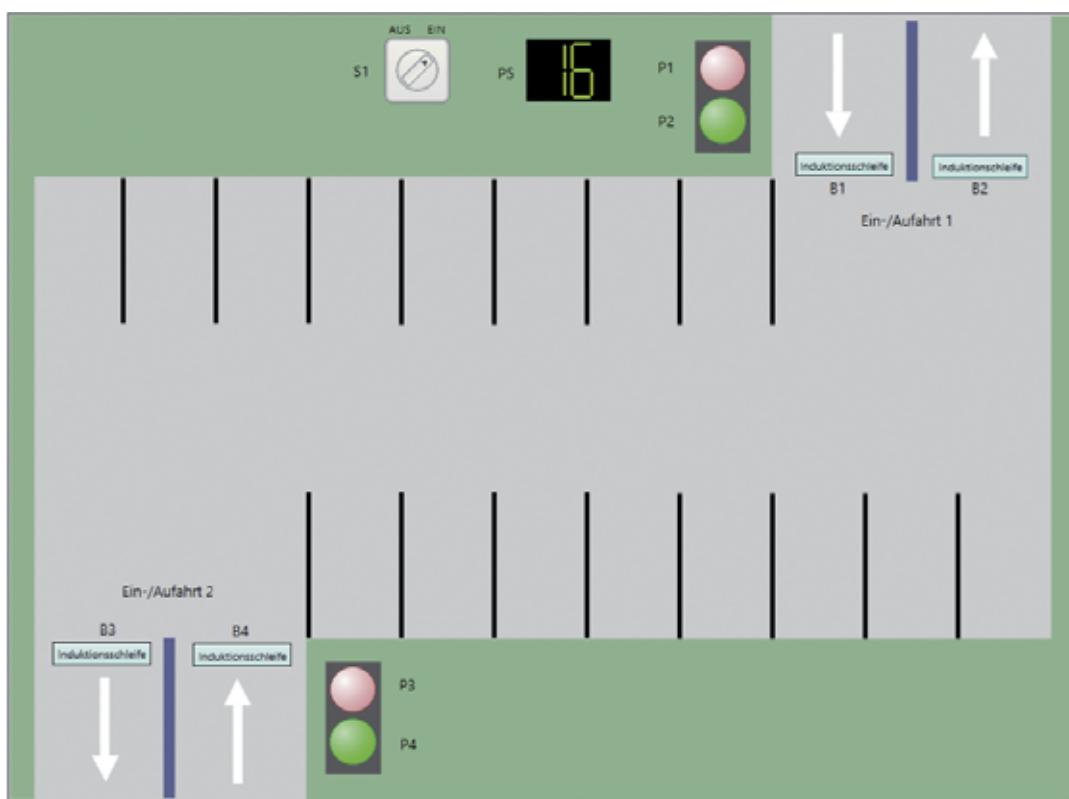
Problemstellung

Auf einem Parkplatz stehen 16 Parkplätze zur Verfügung. Der Parkplatz verfügt über zwei Zufahrten. Jede Zufahrt ist mit einer Ampelanlage ausgestattet. Durch den Wahlschalter „Anlage ein/aus“ wird die Ampelanlage aktiviert. Induktionsschleifen erfassen ein- und ausfahrende Fahrzeuge.

Wenn die Anlage eingeschaltet ist, gibt eine Anzeige an, dass 16 freie Plätze zur Verfügung stehen.

Die Ampeln zeigen an, ob noch Parkplätze frei sind (grün) oder ob der Parkplatz belegt ist (rot). Die Anzeige soll die genaue Anzahl der noch freien Plätze angeben.

Technologieschema:



Visualisierungsdatei: 10_06_Parkplatz.pclab auf der Buch-CD

Zuordnungsliste:

Symbol	Operand	Kommentar	Schaltverhalten
B1	E0.0	Induktionsschleife Einfahrt 1	Schließer
B2	E0.1	Induktionsschleife Ausfahrt1	Schließer
B3	E0.2	Induktionsschleife Ausfahrt 2	Schließer
B4	E0.3	Induktionsschleife Einfahrt 2	Schließer
S1	E0.4	Ein-/Ausschalter	1 = eingeschaltet
P1	A0.0	Ampel 1 rot	–
P2	A0.1	Ampel 1 grün	–
P3	A0.2	Ampel 2 rot	–
P4	A0.3	Ampel 2 grün	–
P5	AD 32	Anzeige „Freie Parkplätze“ (Datentyp: DINT)	–

Aufgabe:

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

12.2 Bibliotheksfähige Bausteine

Die DIN EN 61131-3 macht Vorgaben für die Programmierung von Speicherprogrammierbaren Steuerungen. Um entsprechend der DIN-Vorschrift zu programmieren, sollten folgende Forderungen eingehalten werden:

Anforderungen an bibliotheksfähige Bausteine:

- Der Baustein kann im gleichen oder in einem anderen Programm mehrfach wiederverwendet werden.
- Statt globaler Variablen (E0.0, A4.6, Start, Störung, T1, Z1 usw.) werden im Baustein lokale Variablen verwendet.
- Im aufrufenden Baustein (z.B. OB1) werden den lokalen Variablen absolute Adressen (globale Variablen) zugewiesen.
- Allen Variablen wird ein Datentyp (BOOL, INT, TIME, usw.) zugeordnet.

Die größte Umstellung von der Programmierung mit globalen Variablen zur Programmierung nach DIN EN 61131-3 ist die Verwendung von lokalen Variablen, die in der jeweiligen Funktion (FC) oder dem Funktionsbaustein (FB) frei festgelegt werden können.

Bezeichnung von Variablen

Lokale Variablen	Globale Variablen
#Variablenname	%Variablenname (ab TIA-Portal mit %)
Beispiel: #Start; #Motor1	Beispiel: %S1; %A0.0

Der Vorteil der Programmierung nach DIN EN 61131-3 ist, dass die Bausteine mehrfach aufgerufen werden können. Wenn eine Motorsteuerung mit zwei Drehrichtungen in einer Anlage z.B. dreißig Mal vorkommt, so muss nur ein bibliotheksfähiger Funktionsbaustein nach Norm erstellt werden. Dieser wird dann einfach dreißig Mal aufgerufen und mit den entsprechenden Ein- und Ausgängen parametriert. Dieses vereinfacht die Programmierung und reduziert die Fehlerwahrscheinlichkeit.

In der Variablen-Deklarationstabelle im oberen Teil eines Bausteins können lokale Variablen für einen Baustein definiert werden.

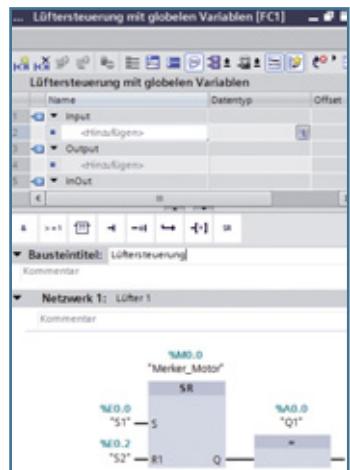


Bild 1: Programmierung mit globalen Variablen (absoluten Adressen)

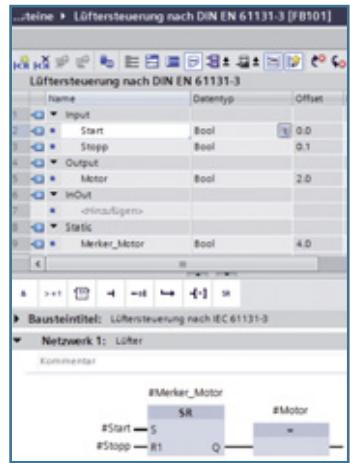


Bild 2: Bibliotheksfähiger Funktionsbaustein mit lokalen Variablen

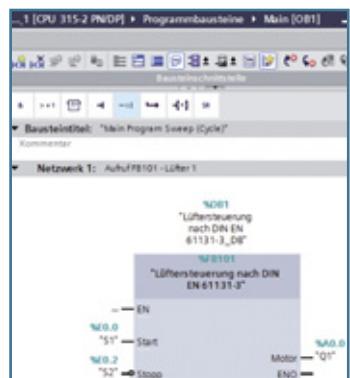


Bild 3: Aufruf des bibliotheksfähigen Funktionsbausteins im OB1

13 Ablaufsteuerungen

13.1 Grundlagen zu Ablaufsteuerungen

Viele Steuerungsprozesse bestehen aus einem festen Ablauf. Eine Ampelsteuerung arbeitet z.B. immer in der gleichen Reihenfolge die einzelnen Programmschritte ab. Steuerungen mit einem wiederkehrenden definierten Ablauf nennt man Ablaufsteuerungen.

Ablaufsteuerungen können nach einem festen Schema programmiert werden (Ablaufkette). Dadurch wird die Programmierung vereinfacht und Programmierfehler werden vermieden.

Ablaufsteuerungen werden in einzelne Programmschritte unterteilt. Bei einer Ampelsteuerung wären solche Programmschritte z.B. Rotphase, Grünpause usw. Der Übergang von einem Schritt zum nächsten erfolgt durch Weiterschaltbedingungen (Transitionen). Sie können zeitgesteuert oder prozessgesteuert sein.

Regeln für Ablaufketten

- Eine Ablaufkette besteht aus Schritten und Weiterschaltbedingungen (Transitionen).
- Zwischen zwei Schritten steht immer eine Transition.
- Der Anfangsschritt ist zu Beginn einer Ablaufkette ohne Bedingung aktiv.
- In linearen Ablaufketten ist immer nur ein Schritt aktiv.
- Man gelangt von einem Schritt in den nächsten, wenn der vorherige Schritt aktiv ist und die Transition erfüllt ist.
- Der nachfolgende Schritt setzt den vorherigen Schritt zurück.
- Den Schritten sind Aktionen zugeordnet, die vom jeweiligen Schritt ausgelöst werden.

Beispiel Rührbehälter:

In einem Behälter werden nacheinander Flüssigkeiten eingefüllt. Anschließend werden die Flüssigkeiten gerührt und dann aus dem Behälter abgepumpt. B1, B2 und B3 geben eine „1“ aus, wenn sie mit Wasser bedeckt sind.

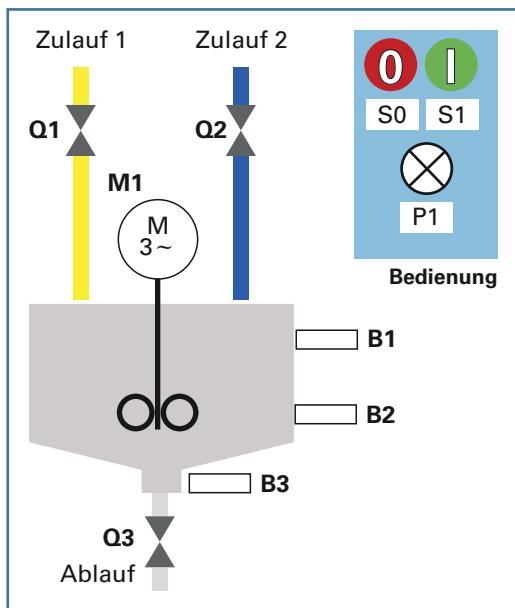


Bild 1: Rührbehälter

Zeichenerklärung:

Anfangsschritt:



Schritt:



Aktionsfeld:
(Befehlsfeld)

Ablassen

Transition:

Rührzeit abgelaufen

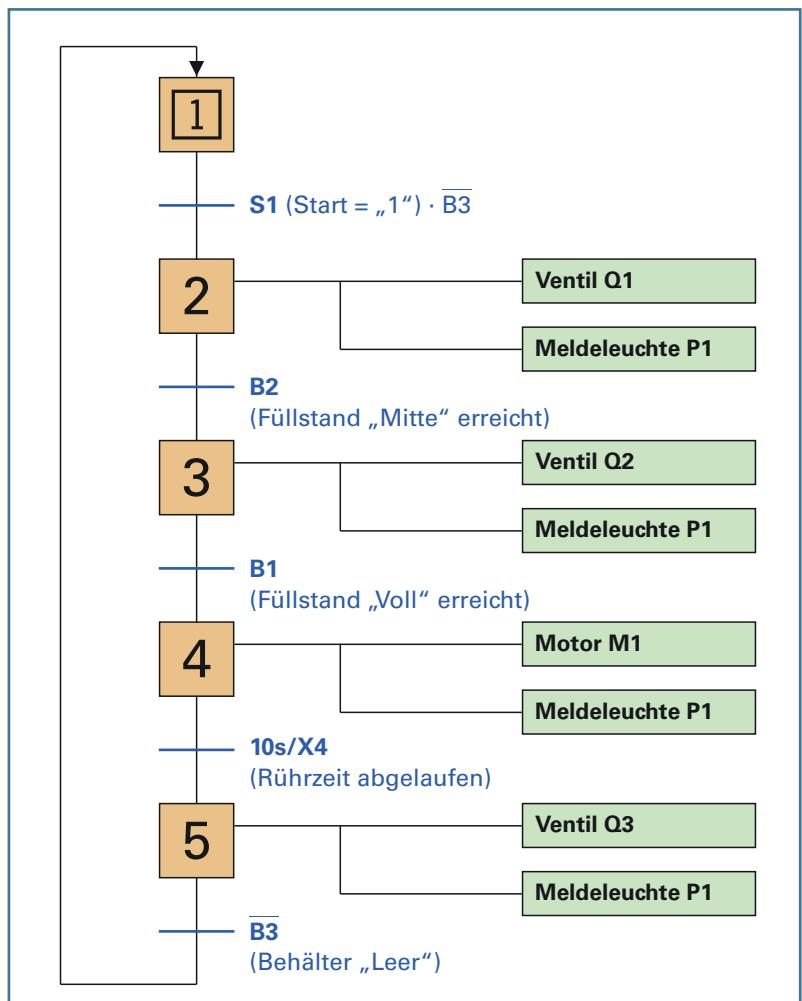


Bild 2: Ablaufkette

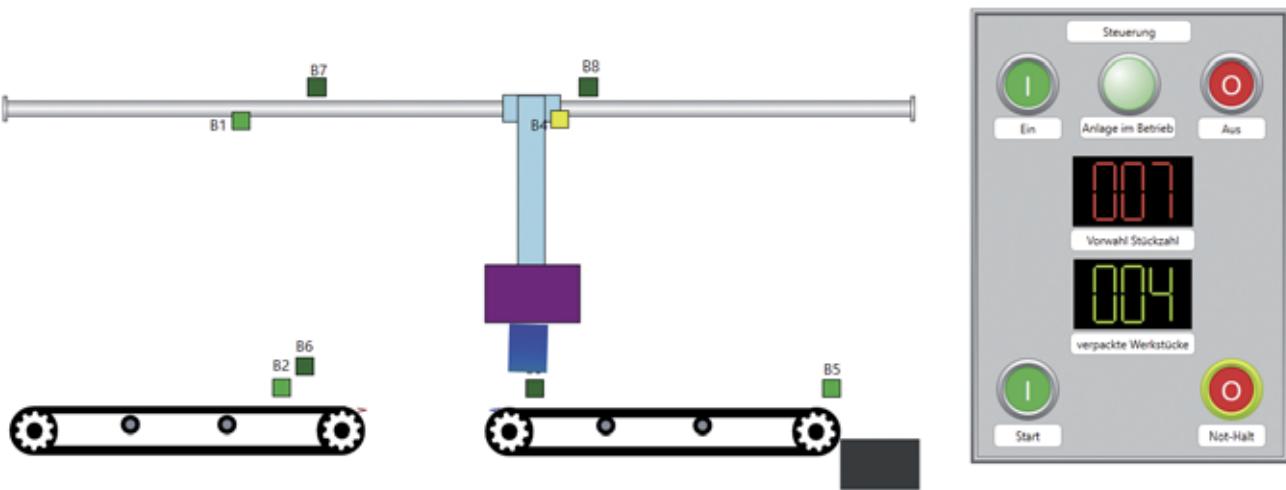
14.7 Verpackungsanlage (Fehlersuche)

Problemstellung

Ein Unternehmen produziert Metallbehälter. Sie sollen in Kisten verpackt werden. Dafür müssen sie von einem Förderband auf ein anderes übergesetzt werden. Dafür steht ein Greifer mit einem Elektromagneten zur Verfügung. Die Anzahl der zu verpackenden Kisten kann vorgewählt werden.



Technologieschema



Visualisierungsdatei: 14_01_Verpackungsanlage.plclab auf der Buch-CD

Zuordnungsliste:

Symbol	Operand	Kommentar	Schaltverhalten
S1	E 0.0	Taster Steuerung ein	Schließer
S2	E 0.1	Taster Steuerung aus	Öffner
S3	E 0.2	Taster Start	Schließer
S4	E 0.3	Taster NOT_HALT	Öffner
B1	E 0.4	Endschalter Position Band 1	Schließer
B2	E 0.5	Endschalter Band 1 Ende	Schließer
B3	E 0.6	Endschalter Band 2 Anfang	Schließer
B4	E 0.7	Endschalter Position Band 2	Schließer
B5	E 1.0	Endschalter Band 2 Ende	Schließer
B6	E 1.1	Endschalter Greifer unten	Schließer
B7	E1.2	Endschalter Greifer oben Pos. 1	Schließer
B8	E1.3	Endschalter oben Pos. 2	Schließer
P1	A 0.0	Meldeleuchte Anlage eingeschaltet	-
M1 (Q1)	A 0.1	Motorschütz Förderband 1	-
M2 (Q2)	A 0.2	Motorschütz Förderband 2	-
M3 (Q3)	A 0.3	Antrieb Greifer nach rechts	-
M3 (Q4)	A 0.4	Antrieb Greifer nach links	-
M4 (Q5)	A 0.5	Antrieb Greifer nach oben	-
M4 (Q6)	A 0.6	Antrieb Greifer nach unten	-
M5 (Q7)	A 0.7	Elektromagnet	1 = ein, 0 = aus
S5	EW20	Soll-Stückzahl	BCD-Format
P2	AW20	Ist-Stückzahl	BCD-Format

16.9 Pegelmessung an einem Wasserkraftwerk 2



Problemstellung

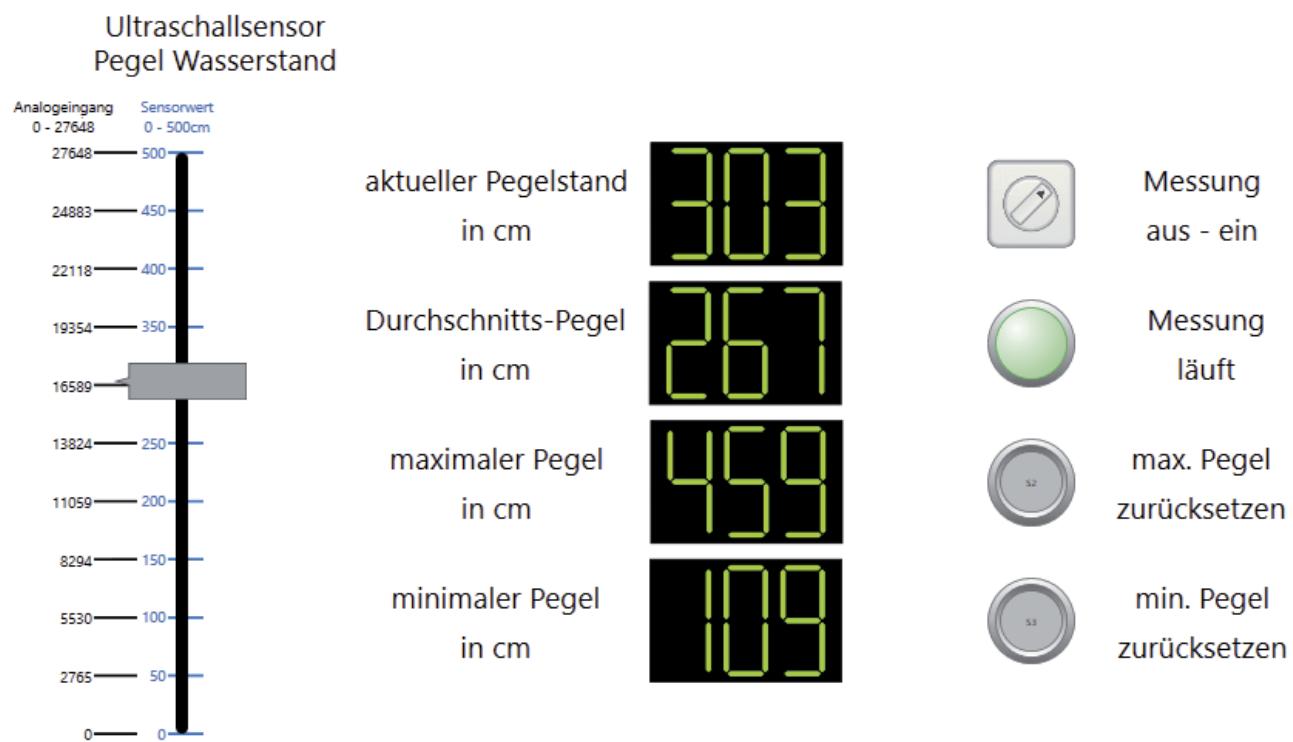
Am Wasserkraftwerk in Oldenburg wird das Anzeigenpanel erweitert.

Ein Ultraschallsensor misst den Wasserstand der Hunte. Der Fluss ist bis zum Wasserkraftwerk noch abhängig von Ebbe und Flut. Ein Anzeigenpanel zeigt alle relevanten Daten an.

Die Messung des Wasserstandes soll alle 20 min (zum Test alle fünf Sekunden) erfolgen. Nach zehn Messungen werden die alten Werte wieder überschrieben. Die Min-/Max-Werte bleiben erhalten.

Die Messwerte sollen in einem Datenbaustein gespeichert werden.

Technologieschema:



Visualisierungsdatei: 16_09_Pegelmessung_Wasserkraftwerk_2.plclab auf der Buch-CD

Zuordnungsliste:

Symbol	Operand	Kommentar
B1	EW256	Ultraschallsensor 0-500 cm (0-10V)
S1	E0.0	Messung ein / aus
S2	E0.1	Taster maximalen Pegel zurücksetzen
S3	E0.2	Taster minimalen Pegel zurücksetzen
P1	A0.0	Meldeleuchte Messung läuft
P2	AW128	aktueller Pegelstand
P3	AW130	Durchschnittspegel
P4	AW132	Maximaler Pegel
P5	AW134	Minimaler Pegel

Aufgabe:

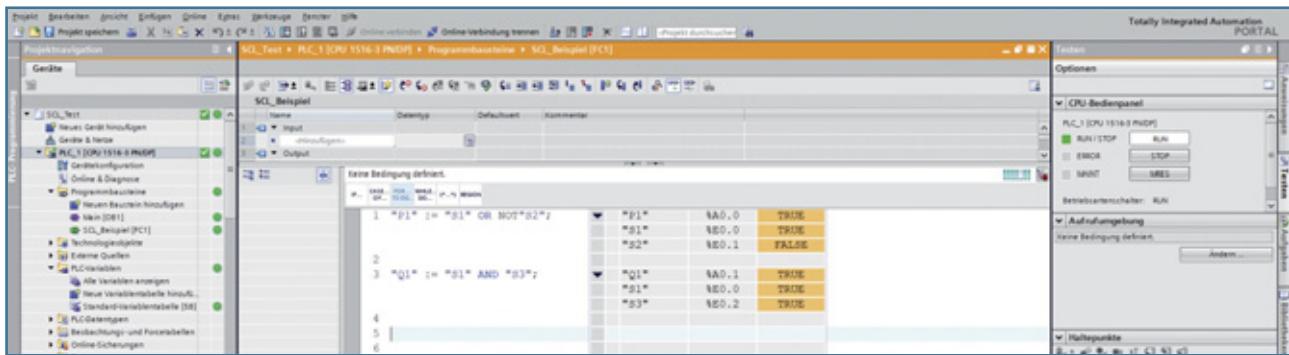
1. Legen Sie ein Projekt an und erstellen Sie das Steuerungsprogramm.
2. Testen Sie das Steuerungsprogramm.

17 Structured Control Language (SCL)

Neben den klassischen Programmiersprachen wie FUP, KOP und AWL bietet das TIA-Portal die Möglichkeit, in der Hochsprache Structured Control Language (SCL) zu programmieren. Bei immer komplexeren Automatisierungsprogrammen bietet SCL eine größere Bandbreite an Programmier- und Strukturierungsmöglichkeiten. Neben den klassischen Grundfunktionen, Zeiten, Zählern bis hin zur Analogwertverarbeitung gibt es bei SCL die Möglichkeit, Schleifen (FOR ...) und Kontrollanweisungen (IF...; CASE...) uvm. zu verwenden.

Alle Anweisungen müssen in SCL mit einem Semikolon (;) abschließen.

Die Programme können über *Beobachten* auch kontrolliert werden. Kommentarzeilen werden mit // gekennzeichnet.



17.1 SCL-Befehle

Zuweisung

Programmanweisung	Erklärung der Programmierzeilen						
<pre>1 "P1" := "S1"; 2 "Q1" := false;</pre> <table border="1"> <tr> <td>"P1"</td> <td>%A0.0</td> </tr> <tr> <td>"S1"</td> <td>%E0.0</td> </tr> <tr> <td>"Q1"</td> <td>%A0.1</td> </tr> </table>	"P1"	%A0.0	"S1"	%E0.0	"Q1"	%A0.1	<p>„P1“ hat ein 1-Signal, wenn „S1“ ein 1-Signal hat.</p> <p>„Q1“ wird der Zustand <i>false</i> (0-Signal) zugewiesen.</p>
"P1"	%A0.0						
"S1"	%E0.0						
"Q1"	%A0.1						

Grundverknüpfungen

Programmanweisung	Erklärung der Programmierzeilen						
UND-Verknüpfung							
<pre>1 "P1" := "S1" AND "S2";</pre> <table border="1"> <tr> <td>"P1"</td> <td>%A0.0</td> </tr> <tr> <td>"S1"</td> <td>%E0.0</td> </tr> <tr> <td>"S2"</td> <td>%E0.1</td> </tr> </table>	"P1"	%A0.0	"S1"	%E0.0	"S2"	%E0.1	„P1“ hat ein 1-Signal, wenn „S1“ und „S2“ ein 1-Signal haben.
"P1"	%A0.0						
"S1"	%E0.0						
"S2"	%E0.1						
ODER-Verknüpfung							
<pre>1 "P1" := "S1" OR "S2";</pre> <table border="1"> <tr> <td>"P1"</td> <td>%A0.0</td> </tr> <tr> <td>"S1"</td> <td>%E0.0</td> </tr> <tr> <td>"S2"</td> <td>%E0.1</td> </tr> </table>	"P1"	%A0.0	"S1"	%E0.0	"S2"	%E0.1	„P1“ hat ein 1-Signal, wenn „S1“ oder „S2“ ein 1-Signal hat.
"P1"	%A0.0						
"S1"	%E0.0						
"S2"	%E0.1						

ODER-Verknüpfung mit negiertem Eingang

Programmanweisung	Erklärung der Programmierzeilen						
<pre>1 "P1" := "S1" OR NOT "S2";</pre> <table border="1"> <tr> <td>"P1"</td> <td>%A0.0</td> </tr> <tr> <td>"S1"</td> <td>%E0.0</td> </tr> <tr> <td>"S2"</td> <td>%E0.1</td> </tr> </table>	"P1"	%A0.0	"S1"	%E0.0	"S2"	%E0.1	„P1“ hat ein 1-Signal, wenn „S1“ und nicht „S2“ ein 1-Signal hat.
"P1"	%A0.0						
"S1"	%E0.0						
"S2"	%E0.1						

NAND-Verknüpfung

Programmanweisung	Erklärung der Programmierzeilen						
<pre>1 "P1" := NOT ("S1" AND "S2");</pre> <table border="1"> <tr> <td>"P1"</td> <td>%A0.0</td> </tr> <tr> <td>"S1"</td> <td>%E0.0</td> </tr> <tr> <td>"S2"</td> <td>%E0.1</td> </tr> </table>	"P1"	%A0.0	"S1"	%E0.0	"S2"	%E0.1	„P1“ verhält sich wie eine UND-Verknüpfung mit negiertem Ausgang.
"P1"	%A0.0						
"S1"	%E0.0						
"S2"	%E0.1						

NOR-Verknüpfung

Programmanweisung	Erklärung der Programmierzeilen						
<pre>1 "P1" := NOT ("S1" OR "S2");</pre> <table border="1"> <tr> <td>"P1"</td> <td>%A0.0</td> </tr> <tr> <td>"S1"</td> <td>%E0.0</td> </tr> <tr> <td>"S2"</td> <td>%E0.1</td> </tr> </table>	"P1"	%A0.0	"S1"	%E0.0	"S2"	%E0.1	„P1“ verhält sich wie eine ODER-Verknüpfung mit negiertem Ausgang.
"P1"	%A0.0						
"S1"	%E0.0						
"S2"	%E0.1						

Exklusiv-ODER-Verknüpfung (XOR)

Programmanweisung	Erklärung der Programmierzeilen						
<pre>1 "P1" := "S1" XOR "S2";</pre> <table border="1"> <tr> <td>"P1"</td> <td>%A0.0</td> </tr> <tr> <td>"S1"</td> <td>%E0.0</td> </tr> <tr> <td>"S2"</td> <td>%E0.1</td> </tr> </table>	"P1"	%A0.0	"S1"	%E0.0	"S2"	%E0.1	P1 hat ein 1-Signal, wenn nur einer der beiden Eingänge ein 1-Signal führt.
"P1"	%A0.0						
"S1"	%E0.0						
"S2"	%E0.1						

18.8.2 Aufbau einer Industrie 4.0-Anlage

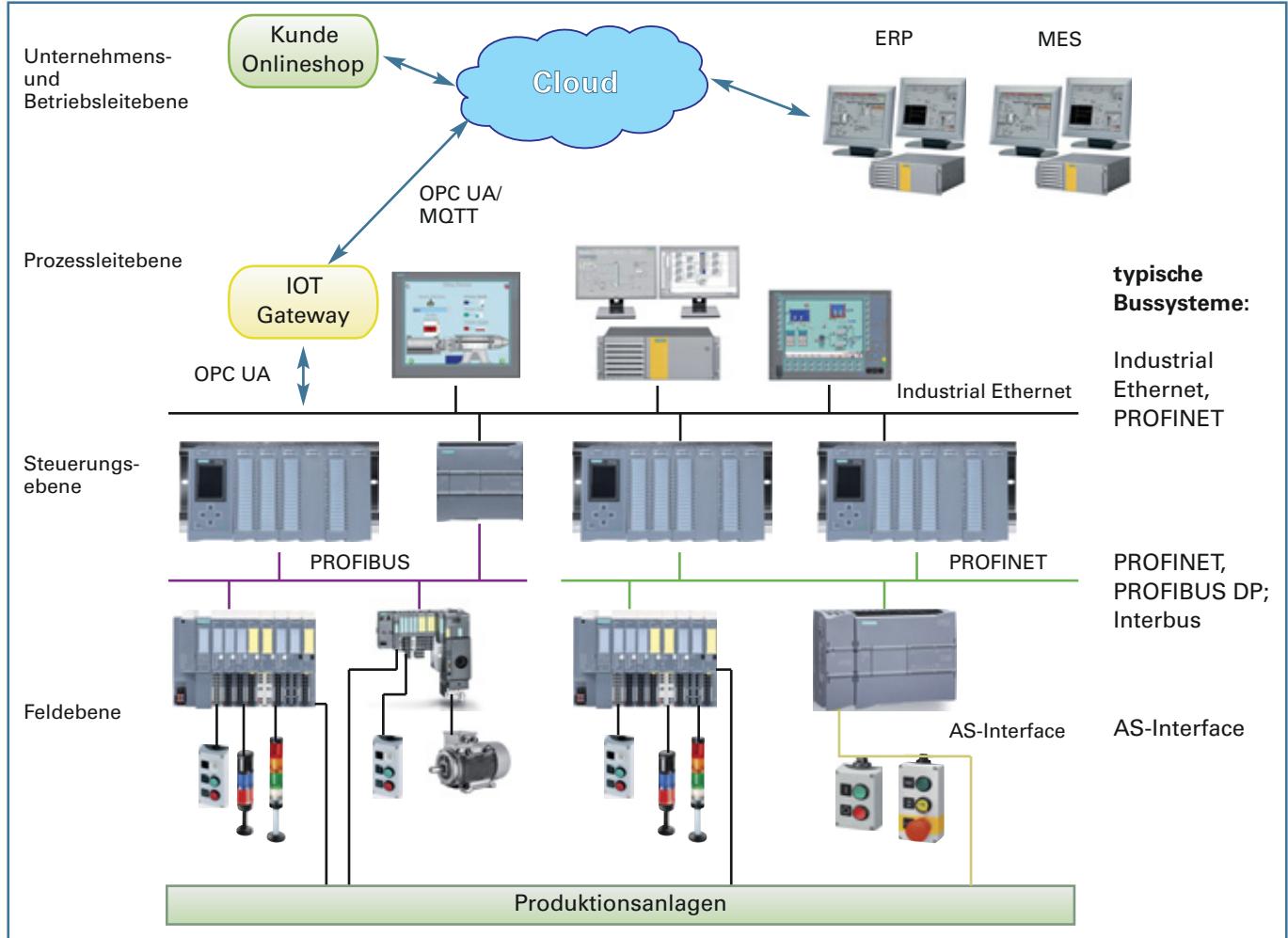


Bild 1: Möglicher Aufbau einer Industrie 4.0-Anlage

Der Aufbau von Industrieanlagen mit einer Industrie 4.0-Ausrichtung kann sehr unterschiedlich sein. Je nach Anlagentyp ist der Grad der Digitalisierung und Vernetzung anders. Die Anbindung an das ERP- und MES-System kann ebenfalls sehr verschieden sein. Die Systeme können direkt über ein Bussystem angeschlossen sein. Sie können aber auch räumlich getrennt von der Produktionsanlage und über internet-/cloudbasierte Zugänge verbunden sein. Je nach Unternehmenstyp ist auch eine entsprechende Anbindung eines Onlineshops denkbar. Beim Datenaustausch von Automatisierungssystemen zur Cloud ist OPC UA (Open Platform Communications Unified Architecture) ein verbreitetes Übertragungsprotokoll.

Begriffe aus dem Bereich Netzwerktechnik / vernetzte Automatisierungssysteme:

Switch:	intelligenter Netzwerkverteiler
Router:	Netzwerkmodul, das Daten zwischen zwei Netzen weiterleiten kann
Client:	Netzwerkteilnehmer, z.B. ein PC, der vom Server Dienste anfordert
Server:	Netzwerkteilnehmer, der Dienste zu Verfügung stellt, z.B. Printserver, Mailserver, Webserver oder Fileserver
Master:	dominanter Teilnehmer in einem Automatisierungssystem, z.B. Profibus-Master, AS-I-Master
Slave:	untergeordneter Teilnehmer in einem Automatisierungssystem, z.B. Profibus-Slave, AS-I-Slave
IP-Adresse:	Adresse eines Netzwerkteilnehmers, z.B. 192.168.0.1
Subnetzmaske:	Unterteilt die IP-Adresse in einen Netz- und einen Hostanteil. Dient zur Strukturierung von Netzwerken; Beispiel: 255.255.255.0
Cloud:	IT-Anwendungen, die online, z.B. über das Internet, zur Verfügung stehen. Sie können Speicherplatz oder auch Anwendungssoftware beinhalten.
OPC UA:	Kommunikationsprotokoll für Industrie 4.0-Anwendungen

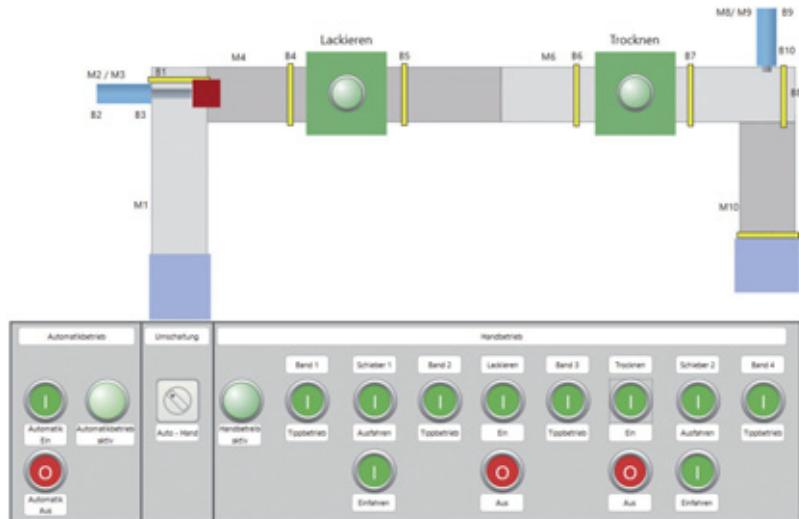
19.3 Übung: Lackierstraße

Problemstellung

In einer automatischen Lackierstraße werden Kisten lackiert und anschließend getrocknet. Die Anlage besteht aus vier Förderbändern für den Transport, zwei pneumatischen Schiebern für die Umsetzung von einem Förderband auf das nächste und aus sieben Lichtschranken (gelb). Die Anlage kann im Automatikbetrieb und im Handbetrieb gefahren werden.



Technologieschema:



Visualisierungsdatei:
19_03_Lackieranlage.plclab
auf der Buch-CD

Zuordnungsliste:

Symbol	Operand	Kommentar	Schaltverhalten
B1	E0.0	Lichtschranke bei Zylinder 1	Schließer
B2	E0.1	Endlage Zylinder 1 eingefahren	Schließer
B3	E0.2	Endlage Zylinder 1 ausgefahren	Schließer
B4	E0.3	Lichtschranke vor Lackierung	Schließer
B5	E0.4	Lichtschranke nach Lackierung	Schließer
B6	E0.5	Lichtschranke vor Trocknung	Schließer
B7	E0.6	Lichtschranke nach Trocknung	Schließer
B8	E0.7	Lichtschranke bei Zylinder 2	Schließer
B9	E1.0	Endlage Zylinder 2 eingefahren	Schließer
B10	E1.1	Endlage Zylinder 2 ausgefahren	Schließer
B11	E1.2	Lichtschranke Band 4 Ende	Schließer
S1	E1.3	Automatikbetrieb ein	Schließer
S2	E1.4	Automatikbetrieb aus	Öffner
S3	E1.5	Band 1 tippen	Schließer
S4	E1.6	Schieber 1 ausfahren	Schließer
S5	E1.7	Schieber 1 einfahren	Schließer
S6	E2.0	Band 2 tippen	Schließer
S7	E2.1	Lackierung ein	Schließer
S8	E2.2	Lackierung aus	Öffner
S9	E2.3	Band 3 tippen	Schließer
S10	E2.4	Trocknung ein	Schließer
S11	E2.5	Trocknung aus	Öffner