



EUROPA-FACHBUCHREIHE  
für elektrotechnische Berufe

# SPS

# Theorie und Praxis

mit Übungsaufgaben

**5. 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 978-3-8085-3533-2

5. Auflage 2017  
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.

© 2017 by Verlag Europa-Lehrmittel, Nourney, Vollmer GmbH & Co. KG, 42781 Haan-Gruiten  
<http://www.europa-lehrmittel.de>

Umschlaggestaltung: Media-Creativ, 40723 Hilden  
Satz und Layout: rkt, 42799 Leichlingen, [www.rktypo.com](http://www.rktypo.com)  
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)

### Download-Hinweis:

Die im Buch verwendete Software kann aus dem Internet geladen werden. Bei den angegebenen Links handelt es sich um kostenlose Test- bzw. Demoversionen. Die Vollversionen können bei den Herstellern erworben werden.

**1. TIA-PORTAL** (SPS-Programmiersoftware)  
SIMATIC STEP 7 PROFESSIONAL V13 TRIAL (21 Tage Testversion)  
<http://support.automation.siemens.com/WW/view/de/78793685>

**2. SPS-VISU S5/S7**, SPS-Visualisierungssoftware  
<http://www.mhj-online.de/de/download>

**3. Siemens Ausbildungsunterlagen**  
Für weiterführende Informationen, die über den Rahmen dieses Buches hinausgehen stellt Siemens Ausbildungsunterlagen zur Verfügung:  
<http://w3.siemens.com/mcms/sce/de/fortbildungen/kurse/Seiten/dDefault.aspx>

# 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 SPS-VISU durch animierte Visualisierungen simuliert werden.

Die theoretischen Erläuterungen, die Beispiele und Übungen basieren auf dem Automatisierungssystem SIMATIC und der Software Step7 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 Bussystemen. Zudem wird auch auf die SPS-Hardware und auf die Fehlersuche eingegangen. Für noch weiterreichende Informationen ist auf der Buch-CD eine umfangreiche Bibliothek mit Ausbildungsunterlagen der Fa. Siemens hinterlegt.

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 5. Auflage

Bei der Programmiersoftware vollzieht sich der Wandel zum TIA-Portal. Dementsprechend sind die Beispielprogramme und die Informationsseiten an die neue Software angepasst worden.

Das Kapitel *Bausteine* ist komplett überarbeitet worden, um den Anforderungen der Norm für die Speicherprogrammierbaren Steuerungen DIN EN 61131-3 zu entsprechen. Neben ausführlichen Beispielen zu bibliotheksfähigen Bausteinen mit Funktionen und Funktionsbausteinen, wird auch das Thema Multiinstanzen behandelt.

Das Kapitel Analogwertverarbeitung ist um zusätzliche Übungen erweitert worden.

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

<b>1. Einleitung</b>	7
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
<b>2. SPS-Hardware</b>	12
2.1 SPS-Aufbau	12
2.2 SPS-Produktspektrum	13
2.3 Darstellung von SPSen in Stromlaufplänen	17
2.4 Wiederholungsfragen	18
<b>3. Step7 – Erstellen eines SPS-Programms</b>	21
3.1 Vorgehensweise bei der Projektbearbeitung	21
3.2 TIA-Portal: Erstellen eines Projektes	22
3.3 Step7 V5.x: Erstellen eines Projektes	25
<b>4. Simulation von Programmen</b>	31
4.1 Simulation mit PLCSIM (für S7-300/S7-400)	31
4.2 Simulation mit PLCSIM V1x (für S7-1200/S7-1500)	33
4.3 Simulation mit SPS-VISU	34
<b>5. Grundverknüpfungen</b>	36
5.1 Programmiersprachen/Darstellungsarten	36
5.2 Grundlagen der Grundfunktionen	37
5.3 Übersicht über Grundfunktionen	38
5.4 Grundverknüpfungen in verschiedenen Programmiersprachen	39
5.5 Adressierung	40
5.6 Merker und Klammerbefehle	40
5.6.1 Merker	40
5.6.2 Klammerbefehle	41
5.7 Verknüpfungsergebnis VKE	42
5.8 Beispielaufgabe: Kühlhaus	43
5.9 Wiederholungsfragen	45
5.10 Übung: Sicherheitscode	47
5.11 Übung: Folgeschaltung von Montagebändern	47
5.12 Übung: Funktionsgleichung	47
5.13 Übung: Rauchmeldeanlage	48
5.14 Übung: Alarmanlage	49
5.15 Übung: Förderbandanlage	50
<b>6. Flipflops (Speicherfunktionen)</b>	52
6.1 SR-Flipflop und RS-Flipflop	52
6.2 Beispielaufgabe: Ansteuerung eines Drehstrommotors	54
6.3 Wiederholungsfragen	56
6.4 Übung: Doppelt wirkender Zylinder	58
6.5 Übung: Wendeschützschaltung	58
6.6 Übung: Förderbandanlage (Folgeschaltung)	59
6.7 Übung: Toranlage	60
6.8 Übung: Sortieranlage	61
<b>7. Strukturierte Programmierung</b>	63
7.1 Lineare Programmierung	63
7.2 Strukturierte Programmierung	63
7.3 Bausteinarten	64
7.4 Wiederholungsfragen	64
<b>8. Zeitfunktionen</b>	65
8.1 SIMATIC-Zeiten	65
8.2 Taktmerker	67
8.3 Beispielaufgabe: Pneumatische Abfülleinrichtung	68

8.4	IEC-Zeiten	70
8.5	Wiederholungsfragen	71
8.6	Übung: Störungslampe (Taktmerker)	72
8.7	Übung: Industrieofen	72
8.8	Übung: Automatische Stern-Dreieck-Schaltung	72
8.9	Übung: Zeitgesteuerte Toranlage	73
8.10	Übung: Zeitgesteuerte Förderbandanlage	74
<b>9.</b>	<b>Bit, Byte, Wort, Doppelwort</b>	<b>75</b>
9.1	Zahlensysteme	75
9.1.1	Das Dezimalsystem	75
9.1.2	Das duale Zahlensystem (Binärsystem)	75
9.1.3	Das BCD-Zahlensystem	75
9.1.4	Das Hexadezimalsystem	76
9.2	Definitionen	76
9.2.1	Bit	76
9.2.2	Byte	76
9.2.3	Wort	76
9.2.4	Doppelwort	77
9.3	Lade- und Transferoperationen	77
9.3.1	Lade- und Transferoperationen in AWL	78
9.3.2	Lade- und Transferoperationen in FUP und KOP	79
9.4	Wiederholungsfragen	79
9.5	Übung: Wortverarbeitung	81
<b>10.</b>	<b>Zähler und Vergleicher</b>	<b>82</b>
10.1	SIMATIC-Zähler	82
10.2	Vergleicher	83
10.3	SIMATIC-Vorwärts-/Rückwärtzzähler mit Vergleicher in AWL, FUP und KOP	84
10.4	IEC-Zähler	86
10.5	Wiederholungsfragen	87
10.6	Übung: Parkplatzampel	89
10.7	Übung: Stanze	90
<b>11.</b>	<b>Verschiedene Programmfunctionen und Befehle</b>	<b>91</b>
11.1	Urlöschen	91
11.2	Remanenzverhalten	91
11.3	Archivieren/Dearchivieren	91
11.4	Flankenauswertung	91
11.5	Sprungoperationen	92
11.6	Wiederholungsfragen	92
<b>12.</b>	<b>Bausteine</b>	<b>94</b>
12.1	Bausteinarten	94
12.1.1	Organisationsbausteine (OB)	94
12.1.2	Funktionen (FC)	94
12.1.3	Funktionsbaustein (FB)	94
12.1.4	Systemfunktionen (SFC) und Systemfunktionsbausteine (SFB)	94
12.1.5	Datenbausteine (DB)	94
12.2	Bibliotheksfähige Bausteine	95
12.3	Anlegen einer eigenen Bibliothek	99
12.4	Datenbausteine	99
12.5	Wiederholungsfragen	101
12.6	Übung: Motorsteuerung mit bibliotheksfähigen Bausteinen	102
<b>13.</b>	<b>Ablaufsteuerungen</b>	<b>104</b>
13.1	Grundlagen zu Ablaufsteuerungen	104
13.2	GRAFCET und DIN EN 61131-3	106
13.3	S7-Graph	110
13.4	Betriebsarten	111
13.5	Wiederholungsfragen	112
13.6	Übung: Leuchtreklame	115
13.7	Übung: Schwimmbad	115

## Inhaltsverzeichnis

---

13.8 Übung: Bohranlage	116
13.9 Übung: Ampelsteuerung	118

## 14. Fehlersuche

---

14.1 Fehlerarten	120
14.2 Fehlersuche bei Hardware-Fehlern	120
14.3 Fehlersuche bei Software-Fehlern	120
14.3.1 Diagnosepuffer	120
14.3.2 Variablen beobachten und steuern	121
14.3.3 Belegungsplan	121
14.3.4 BEA – Bausteinende absolut	122
14.3.5 // - Kommentar	122
14.3.6 Gehe zu => Verwendungsstelle	122
14.3.7 Querverweise	122
14.3.8 Übersetzen	122
14.4 Fehler-Operationsbausteine	123
14.5 Wiederholungsfragen	123
14.6 Übung: Förderbandanlage (Fehlersuche)	125
14.7 Übung: Verpackungsanlage (Fehlersuche)	127

## 15. Mathematische Funktionen

---

15.1 Datentypen	129
15.2 Umwandlungsfunktionen	130
15.3 Rechnen mit Ganzzahlen (INT und DINT)	131
15.4 Rechnen mit Gleitpunktzahlen (REAL)	131
15.5 Übung: Umwandlungsfunktionen	132
15.6 Übung: Mathematische Operation	133

## 16. Verarbeitung von Analogwerten

---

16.1 Analoge Signale	134
16.2 Analogwerte einlesen und ausgeben	135
16.3 Analogwerte einlesen und normieren	135
16.4 Analogwerte auslesen und normieren	137
16.5 Wiederholungsfragen	138
16.6 Übung: Temperaturanzeige	137
16.7 Übung: Temperaturüberwachung	139
16.8 Pegelmessung Wasserkraftwerk 1	141
16.9 Pegelmessung Wasserkraftwerk 2	142

## 17. Bussysteme

---

17.1 Hierarchischer Aufbau	143
17.2 Topologien	144
17.3 Übertragungsmedien	145
17.4 Störgrößen bei leitungsbundener Datenübertragung	145
17.5 Buszugriffsverfahren	146
17.6 Industrielle Bussysteme	147
17.6.1 Ethernet TCP/IP	147
17.6.2 Industrial Ethernet	147
17.6.3 PROFINET	147
17.6.4 PROFIBUS DP	148
17.6.5 Aktor-Sensor-Interface (AS-I)	148
17.7 Wiederholungsfragen	149

## 18. Projektaufgaben

---

18.1 Übung: Motorsteuerung mit bibliotheksfähigen Bausteinen	152
18.2 Übung: Ampelanlage	154
18.3 Übung: Lackierstraße	158
18.4 Übung: Autowaschanlage	160

## 19. Übersicht Befehle unter Step7

---

20. Sachwortverzeichnis	165
-------------------------	-----

# 1 SPS-Grundlagen

## 1.1 Einleitung

Im Alltag begegnet uns eine Vielzahl von Steuerungen, die wir aber nicht bewusst wahrnehmen. Auf dem Weg zur Arbeit steuert die Autoelektronik die technischen Prozesse der Autos, wie Einspritzpumpe, Antiblockiersystem und elektrische Scheibenwischer. Auf der weiteren Fahrt begegnet man vielleicht einer Ampelsteuerung, einer automatisch gesteuerten Straßenbeleuchtung, der gesteuerten Lichtreklame, einer automatischen Parkplatzschranke und vielen anderen Steuerungen.

Die Steuerungstechnik ist aus unserer heutigen Zeit nicht mehr wegzudenken. Sie nimmt uns viele Aufgaben ab und ermöglicht es, Prozesse automatisch ablaufen zu lassen.

Der Mensch verlässt sich ganz auf die Hard- und Software der Steuerungstechnik, z.B. bei einer Ampelsteuerung oder bei einem Fahrstuhl. Die Aufgabe eines Entwicklers von Steuerungseinheiten ist es, die Steuerung so zuverlässig und sicher zu gestalten, dass sich die Anlage oder Maschine jederzeit so verhält, wie es von ihr erwartet wird.

## 1.2 Arten von Steuerungen

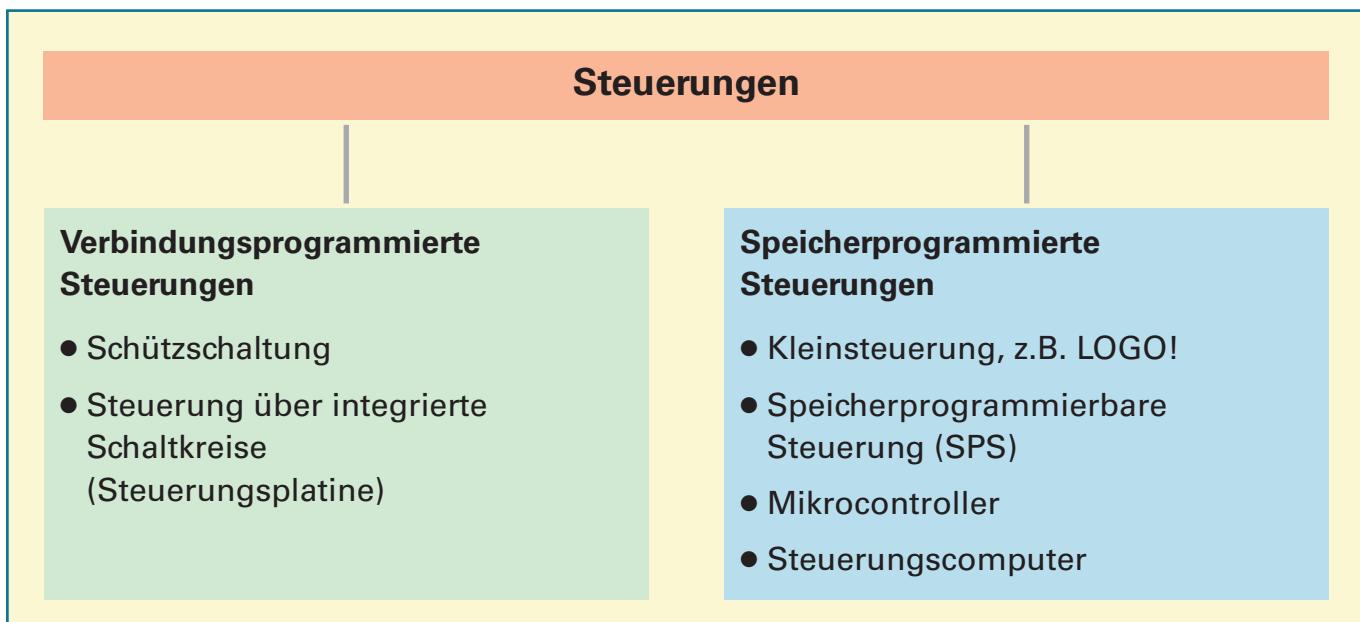


Bild 1: Möglichkeiten von Steuerungen

Um Steuerungen zu realisieren, gibt es verschiedene Möglichkeiten. Sie reichen von der einfachen Schützsteuerung bis zur speicherprogrammierbaren Steuerung mit Busanbindung und der Möglichkeit des weltweiten Fernzugriffs.

Grundsätzlich sind zwei Arten von Steuerungen zu unterscheiden, die *Verbindungsprogrammierten Steuerungen* (VPS), wie sie z.B. in Schützschaltungen zu finden sind, und die *Speicherprogrammierten Steuerungen*.

Speicherprogrammierte Steuerungen können allerdings nur den Steuerstromkreis einer Schützschaltung ersetzen. Zum Schalten von großen Leistungen, z.B. das Einschalten eines Motors, werden nach wie vor Leistungsschütze benötigt.

Der Vorteil einer speicherprogrammierten Steuerung liegt in der wesentlich flexibleren Handhabung. Änderungen oder Ergänzungen sind im Gegensatz zur VPS mit wenig Aufwand vorzunehmen.

### Vorteile einer speicherprogrammierten Steuerung

- Anpassungsfähigkeit
- Wartungsarmut
- Zeitsparende Projektierung
- Platzersparnis
- Automatische Programmdokumentation
- Visualisierung ist möglich
- Kommunikationsfähigkeit (Bussysteme)
- Fernwartung ist möglich

### Nachteile einer speicherprogrammierten Steuerung

- Fachkenntnisse erforderlich
- Kosten für Hard- und Software

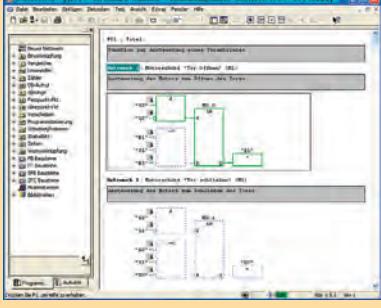
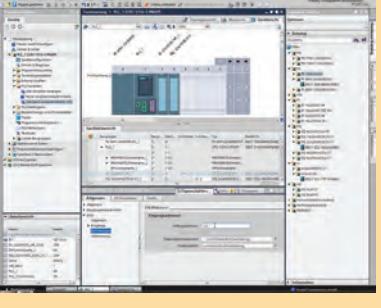
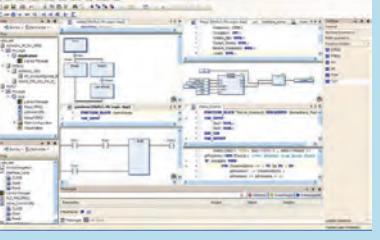
## 1.3 SPS-Bezeichnung

<b>international</b> <b>PLC</b> Programmable Logic Controller		<b>deutsch</b> <b>SPS</b> 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	<b>Step 7 V5.x</b> 	<b>TIA-Portal ab V11</b> 	<b>CoDeSys</b> 
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			

## 1.5 Aufbau und Wirkungsweise einer SPS

Das EVA-Prinzip (Eingabe – Verarbeitung – Ausgabe) stellt die generelle Gliederung einer elektronischen Steuerung dar. Die *Eingabe* kann durch eine Vielzahl verschiedener Sensoren erfolgen, die sowohl digitale als auch analoge Signale an die Steuerung weitergeben. Die Sensoren werden an die Eingabebaugruppen angeschlossen.

Die *Verarbeitung* erfolgt durch das Steuerungsprogramm der SPS, das zyklisch immer wieder durchlaufen wird, um Änderungen der Eingänge zu verarbeiten. Das Steuerungsprogramm wird über die Bediensoftware (bei Siemens: Step7) am Computer erstellt und dann in die SPS übertragen. In der CPU (Central Prozessor Unit) findet die Verarbeitung statt.

Dort befinden sich Speicher für:

- Betriebssystem
- Arbeitsspeicher
- Prozessabbild der Ausgänge
- Zeitglieder
- Merker
- Anwenderprogramm
- Prozessabbild der Eingänge
- Akkumulatoren
- Zähler

Außerdem ist eine CPU mit einer Schnittstelle für den Anschluss des Programmiergerätes ausgestattet. Optional können zusätzliche Bus-Schnittstellen vorhanden sein.

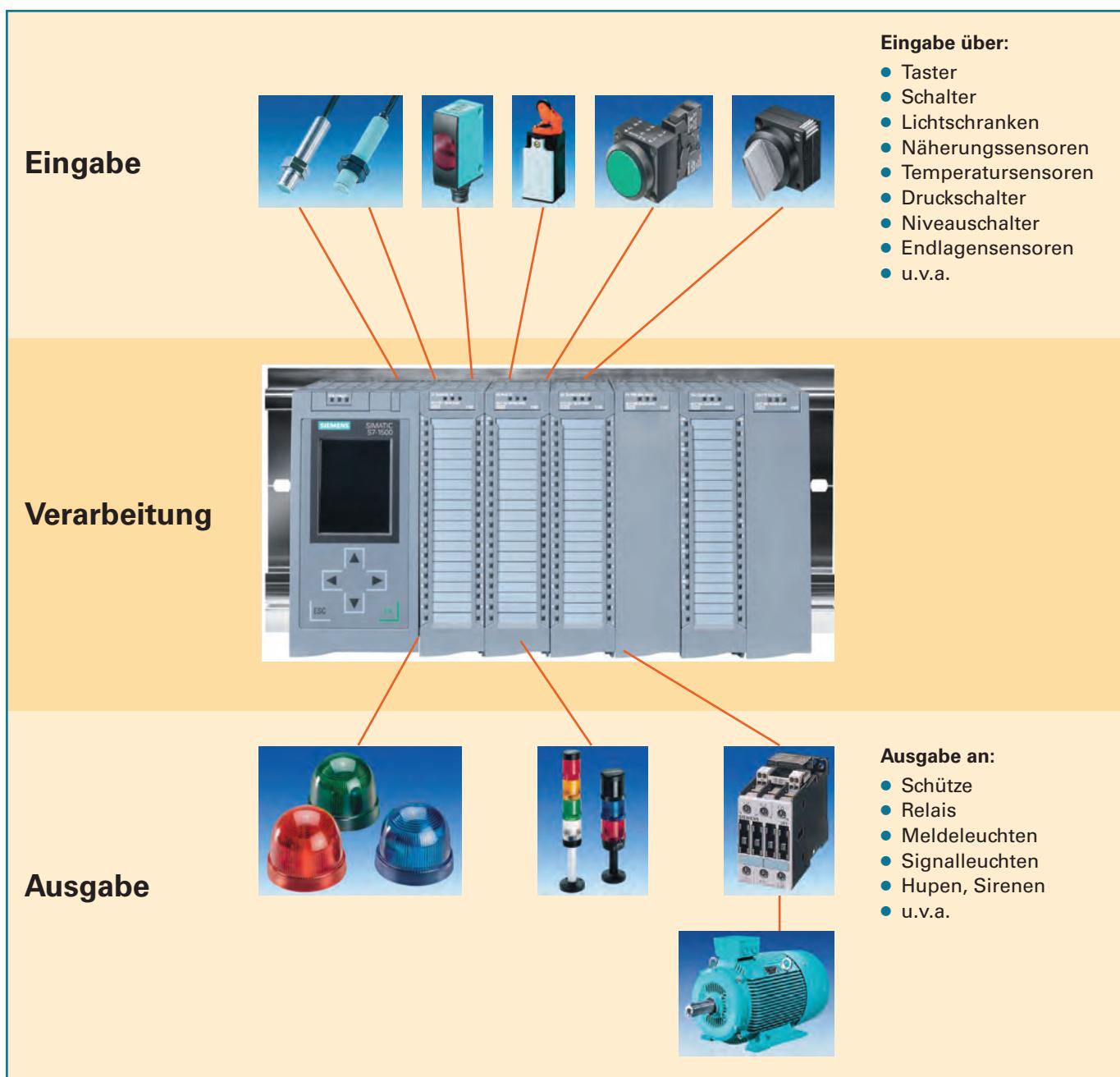


Bild 1: EVA-Prinzip

## 2.2 SPS-Produktspektrum

In der Automatisierungstechnik gibt es eine Vielzahl von Anwendungsmöglichkeiten mit sehr unterschiedlichen Anforderungen an die Hardware. Das Spektrum reicht von sehr kleinen lokalen Steuerungslösungen bis hin zu komplexen Automatisierungslösungen mit einer Vielzahl von Teilnehmern, die über verschiedene Bussysteme miteinander kommunizieren.

Genauso breit wie das Anforderungsspektrum ist auch die Produktvielfalt der angebotenen SPS-Komponenten. Im Folgenden wird ein Überblick über die Vielfalt der Hardware am Beispiel der SIMATIC-Hardware der Fa. Siemens gegeben. Ergänzend werden auch Produkte anderer Hersteller beispielhaft angegeben.

### SPS-Hardware

#### SPS-Controller

SPS-Controller sind einschaltfertige Geräte, die für Steuerungsaufgaben optimiert sind. Je nach Anbieter werden sie als Kompaktgerät oder als modulare SPS angeboten. Diese lassen sich, je nach Bedarf, mit den benötigten Ein-/Ausgangskarten, Funktions- und Kommunikationsbaugruppen aufbauen. Das Spektrum reicht von Kleinsteuerungen bis hin zu hochverfügbaren und fehlersicheren Steuerungen mit dezentraler Peripherie.

<b>LOGO!</b> 	<ul style="list-style-type: none"> <li>• Kleinsteuerung für einfache Steuerungsaufgaben in Industrie und Handwerk</li> <li>• Programmierung über LOGO!-Soft Comfort</li> </ul>
<b>SIMATIC S7-200 / SIMATIC S7-1200</b> 	<ul style="list-style-type: none"> <li>• für kleine und mittlere Steuerungsaufgaben</li> <li>• Programmierung über Step7 Micro bzw. SIMATIC Step7 Basic</li> <li>• S7-1200 ist das Nachfolgeprodukt von S7-200</li> </ul>
<b>SIMATIC S7-300</b> 	<ul style="list-style-type: none"> <li>• für mittlere und größere Steuerungsaufgaben</li> <li>• umfangreiche Funktionalität</li> <li>• viele modulare Bauteile für die verschiedenen Aufgaben verfügbar</li> <li>• Programmierung über Step7</li> </ul>
<b>SIMATIC S7-400</b> 	<ul style="list-style-type: none"> <li>• für umfangreiche Steuerungsaufgaben</li> <li>• umfangreiche Funktionalität</li> <li>• viele modulare Bauteile für die verschiedenen Aufgaben verfügbar</li> <li>• Programmierung über Step7</li> </ul>
<b>SIMATIC S7-1500</b> 	<ul style="list-style-type: none"> <li>• für mittlere bis umfangreiche Steuerungsaufgaben</li> <li>• umfangreiche Funktionalität</li> <li>• viele modulare Bauteile für verschiedene Bauteile verfügbar</li> <li>• Nachfolgeprodukt von S7-300 und S7-400</li> <li>• Programmierung über TIA-Portal</li> </ul>

### SPS-Controller

**Siemens**  
Dezentrale Peripherie SIMATIC ET200



- modulare dezentrale Erweiterung der Steuerung
- dezentrale Ein-/Ausgänge und weitere Module
- Anbindung über Bussysteme
- Schutzart IP65/67 (ohne Schaltschrank) und IP20 (im Schaltschrank)

**SPS**

**Eaton u.  
Omron**



Eaton XC 200 Werksbild Eaton



Beispiele für speicher-programmierbare Steuerungen

### Embedded Automation

Embedded Systeme verbinden einen Controller mit einem PC-System. Steuerung und PC-Applikationen laufen auf einer gemeinsamen robusten Plattform. So kann ein SPS-Programm gemeinsam mit einer Visualisierung auf einem Gerät laufen.

**Siemens**  
Embedded Controller



- Kombination aus Controller und PC-basiertem System
- keine Verwendung von drehenden Teilen wie z.B. Festplatten oder Lüfter
- steuern, bedienen u. beobachten
- Datenverarbeitung
- Kommunikation

**Beckhoff**  
Embedded Controller



Embedded-PC  
Serie CX1020, CX1030

### PC-basierte Steuerungen

Bei PC-basierten Steuerungen wird auf einem PC eine Software-SPS verwendet.

Neben Steuerungs- und Visualisierungsfunktionen können Aufgaben mit hohem Datenaufkommen und schnelle technologische Funktionen auf einer PC-Plattform gelöst werden.

**Siemens**  
PC-basierte Steuerung



- steuern, bedienen und beobachten
- flexibel einsetzbar
- offene Hardware- und Software-Konfiguration
- Industrie-PC

**Phoenix Contact**  
PC-basierte Steuerung



Industrie-PC der Firma Phoenix Contact,  
Typ: S-MAX 400 CE PN II

## 3.2 TIA-Portal: Erstellen eines Projektes

### 1. S7-Projekt anlegen

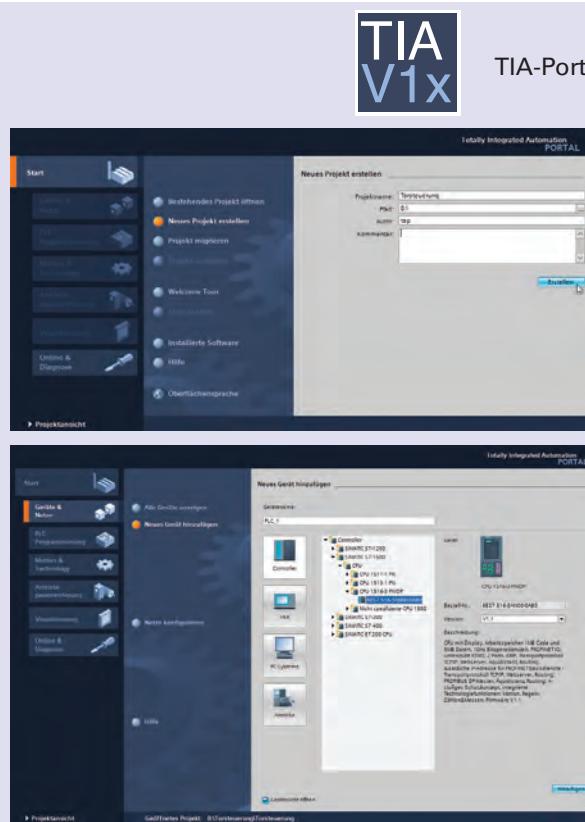
#### Programm öffnen

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 **Projektnamen** 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**.

**Projektansicht beim TIA-Portal**

Projektstruktur	Arbeitsbereich	Taskcards (abhängig vom Editor)

## 1. S7-Projekt anlegen

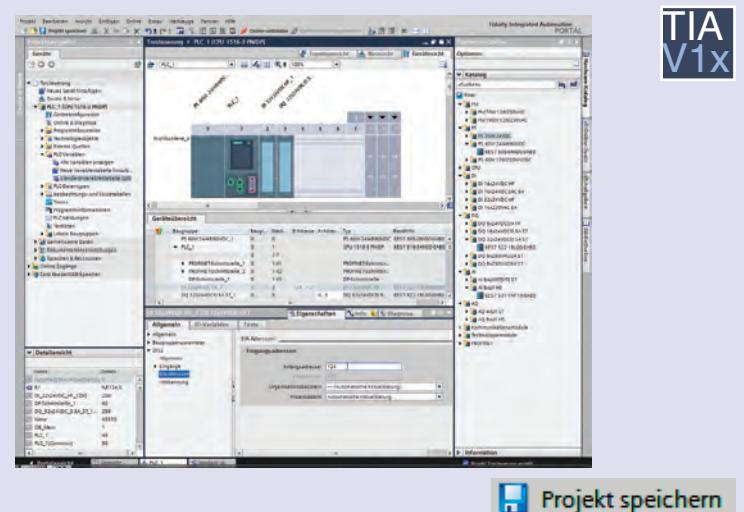
### Hardware auswählen

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 Inspektorenfenster die Eigenschaften angesehen und geändert werden.

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

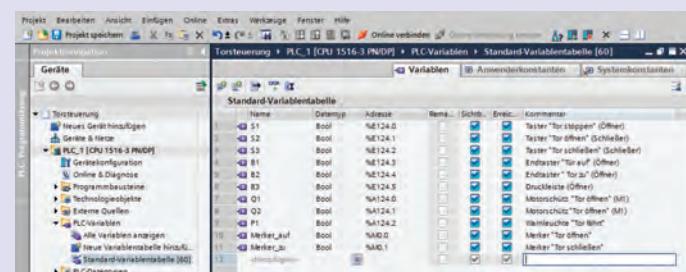
Speichern Sie die Gerätekonfiguration über **Projekt speichern**.



## 3. Variabellabelle erstellen

Wählen sie in der **Projektstruktur** die **Standard-Variabellabelle** 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.

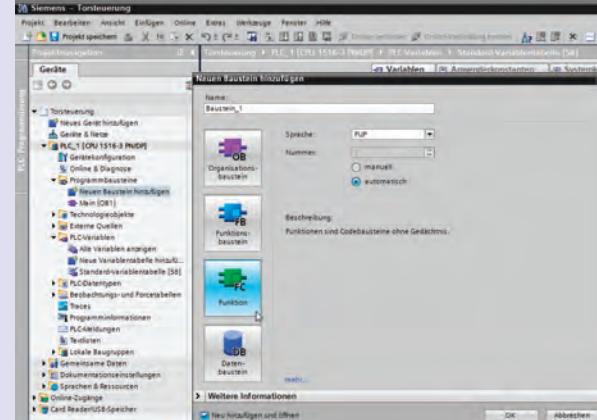


## 4. Bausteine hinzufügen

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

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

Bestätigen Sie mit **OK**.



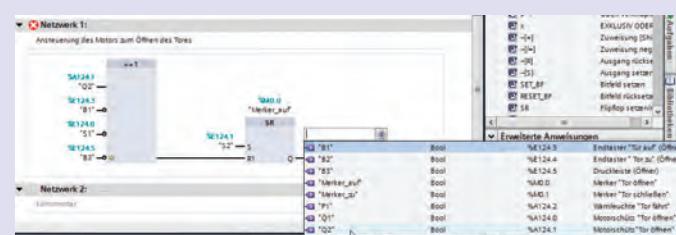
## 5. Programm erstellen

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

Die **Programmiersprache** kann im Inspektorenfenster 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 angewählt werden.

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



# 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

## 8.4 IEC-Zeiten

Es gibt drei verschiedene 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.

	FUP	KOP	AWL
<b>Zeit als Impuls</b>	<p>%DB1 TP Time IN --- #Start PT --- #Zeitwert Q --- ET --- #aktueller Zeitwert = --- Q --- #binärer Ausgang</p>	<p>%DB1 TP Time IN --- #Start Q --- ET --- #aktueller Zeitwert = --- PT --- #Zeitwert</p>	CALL TP, "DB1" Time IN :=# Start PT :=#Zeitwert Q :=# "binärer Ausgang" ET :=# "aktueller Zeitwert"
<b>Einschaltverzögerung</b>	<p>%DB2 TON Time IN --- #Start PT --- #Zeitwert Q --- ET --- #aktueller Zeitwert = --- Q --- #binärer Ausgang</p>	<p>%DB2 TON Time IN --- #Start Q --- ET --- #aktueller Zeitwert = --- PT --- #Zeitwert</p>	CALL TON, "DB2" Time IN :=# Start PT :=#Zeitwert Q :=# "binärer Ausgang" ET :=# "aktueller Zeitwert"
<b>Ausschaltverzögerung</b>	<p>%DB3 TOF Time IN --- #Start PT --- #Zeitwert Q --- ET --- #aktueller Zeitwert = --- Q --- #binärer Ausgang</p>	<p>%DB3 TOF Time IN --- #Start Q --- ET --- #aktueller Zeitwert = --- PT --- #Zeitwert</p>	CALL TOF, "DB3" Time IN :=# Start PT :=#Zeitwert Q :=# "binärer Ausgang" ET :=# "aktueller Zeitwert"

Bei der *Zeit als Impuls* gibt der Ausgang nach einer positiven Flanke an IN für die an PT eingestellte Zeit ein 1-Signal am 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.

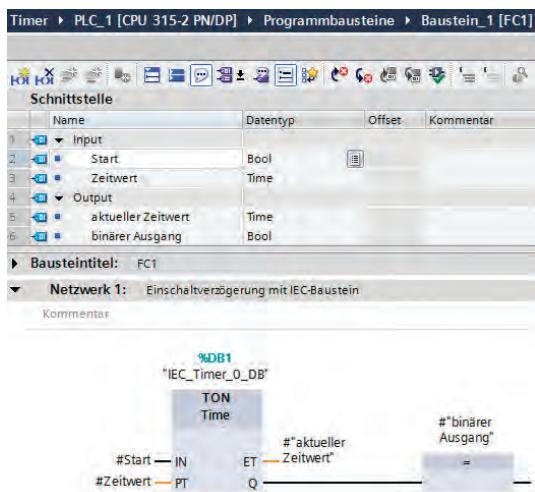


Abb. 1: Programmierung des Rückwärtszählers im FC1

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.

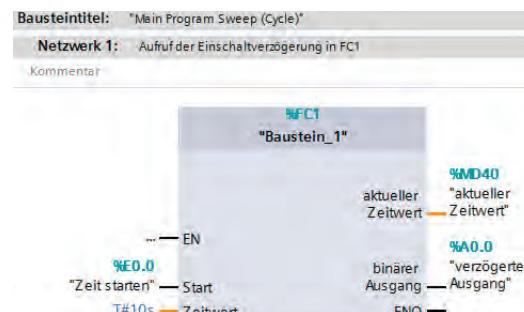


Abb. 2: Aufruf des FC1 im OB1

## 8.9 Übung: Zeitgesteuerte Toranlage

### Problemstellung

Das Tor einer Hofeinfahrt wird über den Taster **Auf** geöffnet. Nach 10 s schließt das Tor automatisch. Über den Taster **Stopp** kann das Tor angehalten werden. Zudem kann das Tor jederzeit mit dem Taster **Schließen** geschlossen werden. Ein direktes Umschalten von Öffnen in Schließen soll möglich sein.

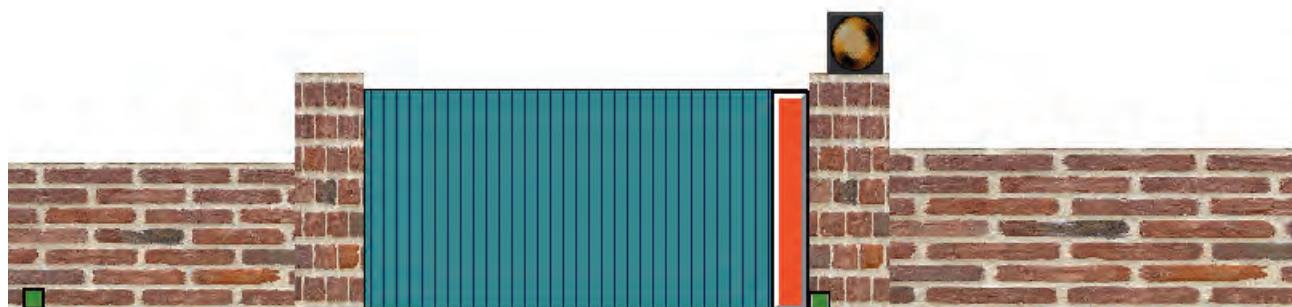
Wird beim Öffnen oder Schließen die rote Druckleiste betätigt, bleibt das Tor stehen.

Eine Leuchte zeigt den Betrieb der Anlage an, sie soll im Sekundentakt blinken.

### Technologieschema:



Bitte auch die rote Druckleiste testen!



Visualisierungsdatei: *08\_Toranlage\_zeitgesteuert.VIS* auf der Buch-CD

### Zuordnungsliste:

Symbol	Operand	Kommentar	Schaltverhalten
Stopp	E0.0	Taster Stopp	Öffner
Auf	E0.1	Taster Tor öffnen	Schließer
Schließen	E0.2	Taster Tor schließen	Schließer
Druckleiste	E0.3	Druckleiste am Tor	Öffner
End_auf	E0.4	Endschalter Tor auf	Öffner
End_zu	E0.5	Endschalter Tor zu	Öffner
M1 (Q1)	A0.0	Motorschütz Tor öffnen	-
M1 (Q2)	A0.1	Motorschütz Tor schließen	-
P1	A0.2	Meldeleuchte Anlage in Betrieb	-

### Aufgabe:

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

Zum Test ist auch die Hardwarekonfiguration in SPS-VISU zu laden.

## 10.5 Wiederholungsfragen

- 1 Welchen minimalen und maximalen Zählstand kann ein Zähler unter Step7 anzeigen?

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

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

<b>Netzwerk 1</b> 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	<b>Netzwerk 2</b> L MW6 L 200 ==I = A124.0
--	--

- 4 Welche Bedeutung haben die Programmzeilen NOP 0 in Aufgabe 3?

## 10.6 Übung: Parkplatzampel

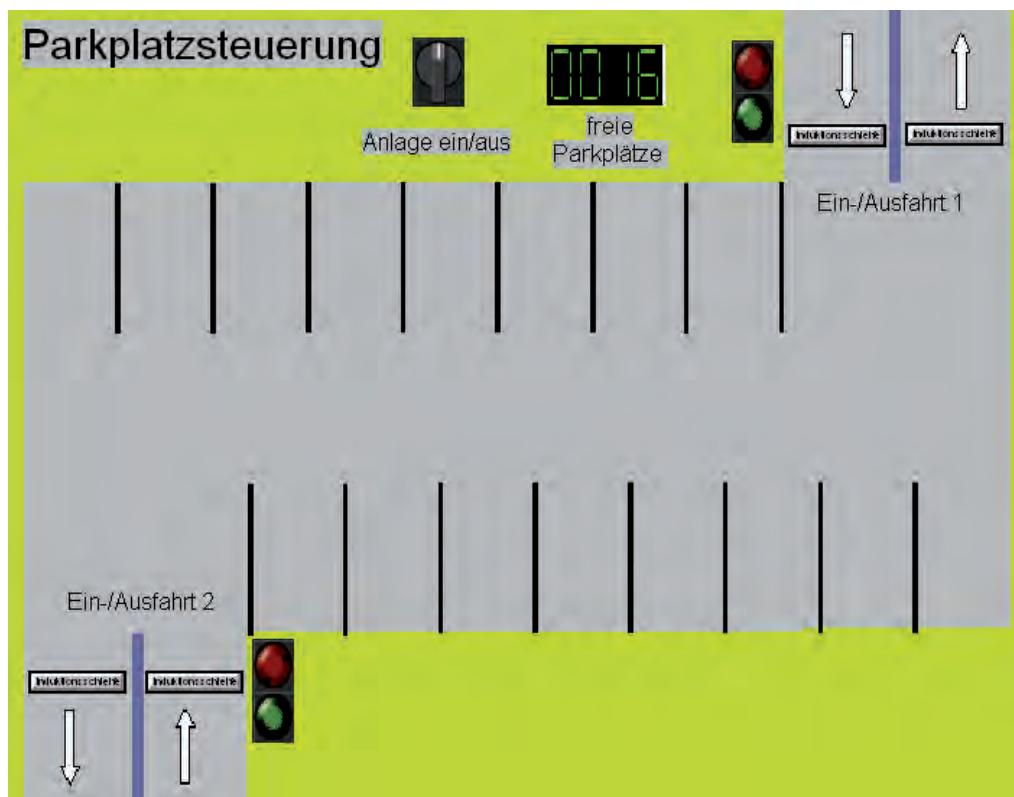
### Problemstellung

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 BCD-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 BCD-Anzeige soll die genaue Anzahl der noch freien Plätze anzeigen.

### Technologieschema:



Visualisierungsdatei: 10\_Parkplatz.VIS auf der Buch-CD

### Zuordnungsliste:

Symbol	Operand	Kommentar	Schaltverhalten
B1	E0.0	Induktionsschleife Einfahrt 1	Schließer
B2	E0.1	Induktionsschleife Ausfahrt 1	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	AW 32	BCD-Anzeige „Freie Parkplätze“	-

### Aufgabe:

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

Der Anfangsschritt (Schritt 1) muss beim Einschalten der Steuerung automatisch aktiv sein. Um dieses zu gewährleisten, wird ein Richtimpuls erzeugt. Im ersten Bearbeitungszyklus hat M80.1 den Zustand „1“, danach wird er automatisch zurückgesetzt. Alle anderen Merker haben beim Einschalten den Zustand „0“. Bei der Steuerung des Rührbehälters gelangt man außerdem in den Schritt 1, wenn man die Ablaufkette durchlaufen hat (Füllstandsmelder B3) oder wenn der Austaster betätigt wurde.

Je nach Anlagentyp kann es sinnvoll sein, beim Betätigen des Austasters nicht in den ersten Schritt zurückzuspringen, sondern im aktuellen Schritt zu bleiben und nur die Ausgänge zu deaktivieren.

Nachdem alle Schritte programmiert worden sind, werden die Zeiten festgelegt. Anschließend können die Ausgänge programmiert werden, wobei ggf. auf erforderliche Verriegelungen zu achten ist.

### Arten von Ablaufsteuerungen:

- Zeitgesteuert
- Prozessgesteuert
- Zeit- und prozessgesteuert

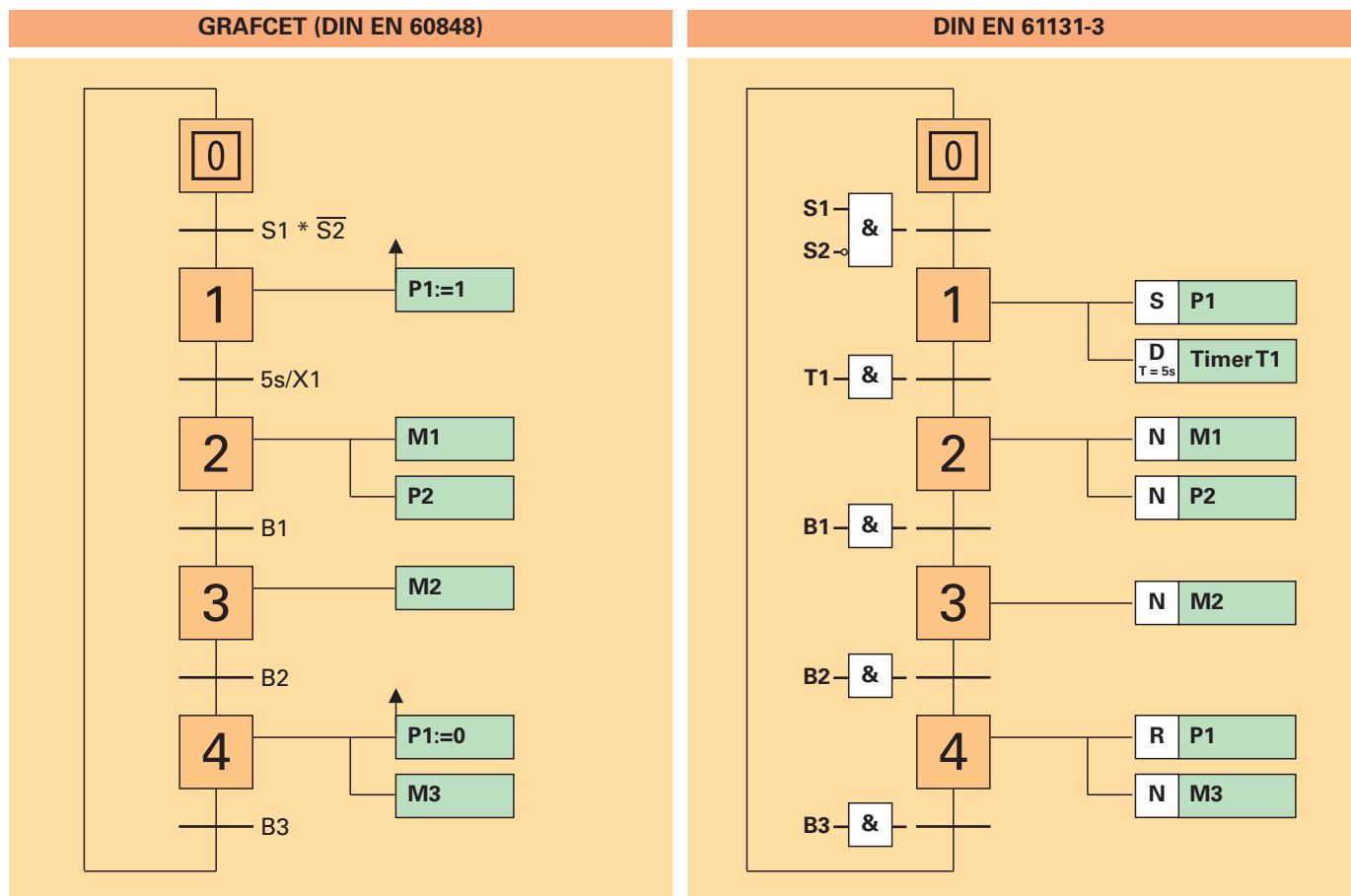
### Betriebsarten:

- Handbetrieb
- Automatikbetrieb
- Einzelschrittbetrieb

Bei Step7 steht mit S7-Graph ein Programmierwerkzeug zur Verfügung, um vereinfacht Ablaufsteuerungen zu erstellen.

## 13.2 GRAFCET und DIN EN 61131-3

Für die Darstellung von Ablaufsteuerungen gibt es zwei Darstellungsweisen. In der **DIN EN 60848 GRAFCET** wird die Ablaufkette unabhängig von der Umsetzung (elektromechanische, pneumatisch, elektronisch oder gemischt) beschrieben. Die DIN EN 60848 hat die DIN 40719-6 ersetzt. Daneben beschreibt die **DIN EN 61131-3** die Umsetzung der Ablaufkette mittels Ablaufsprachen. Bei Siemens wird dies mit der Programmiersprache Graph umgesetzt. Beide Normen haben nebeneinander bestand.

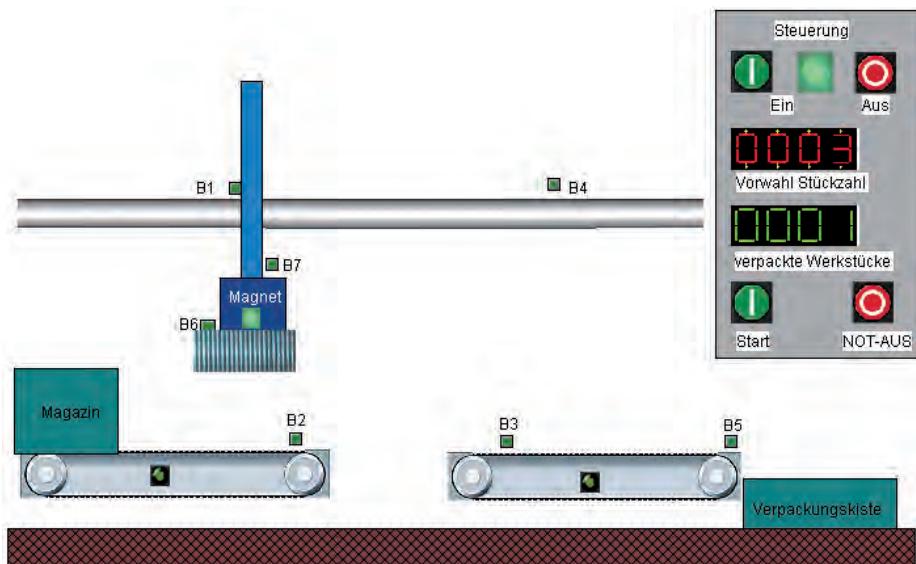


## 14.7 Verpackungsanlage (Fehlersuche)

### 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 Elektromagnet zur Verfügung. Die Anzahl der zu verpackenden Kisten kann vorgewählt werden.

### Technologieschema



Visualisierungsdatei:  
14\_Verpackungsanlage.VIS  
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_AUS	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	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 Wasserwerk 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:



### Zuordnungsliste:

Symbol	Operand	Kommentar
B1	EW256	Ultraschallsensor 0-500 cm (0-10V)
S1	E0.0	Messung ein / aus
S3	E0.1	Taster maximalen Pegel zurücksetzen
S4	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
P6	AW136	Zeit bis zur nächsten Messung

### Aufgabe:

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