

Günter Wellenreuther
Dieter Zastrow

Automatisieren mit SPS

Übersichten und Übungsaufgaben

3. Auflage

Von Grundverknüpfungen bis Ablaufsteuerungen:
STEP 7-Programmierung, Lösungsmethoden,
Lernaufgaben, Kontrollaufgaben, Lösungen,
Beispiele zur Anlagensimulation



Viewegs Fachbücher der Technik

Günter Wellenreuther
Dieter Zastrow

**Automatisieren mit SPS
Übersichten und
Übungsaufgaben**

Aus dem Programm
Automatisierungstechnik

Automatisieren mit SPS

Theorie und Praxis

von G. Wellenreuther und D. Zastrow

Speicherprogrammierbare Steuerungen

von W. Braun

Mechatronik

herausgegeben von B. Heinrich

Messtechnik

von R. Parthier

Regelungstechnik für Ingenieure

von M. Reuter und S. Zacher

Regelungstechnik II

**Zustandsregelungen, digitale und nichtlineare
Regelsysteme**

von H. Unbehauen

Übungsbuch Regelungstechnik

von S. Zacher

Günter Wellenreuther
Dieter Zastrow

Automatisieren mit SPS Übersichten und Übungsaufgaben

**Von Grundverknüpfungen bis Ablaufsteuerungen:
STEP 7-Programmierung, Lösungsmethoden,
Lernaufgaben, Kontrollaufgaben, Lösungen,
Beispiele zur Anlagensimulation**

3., überarbeitete und ergänzte Auflage

Mit 10 Einführungsbeispielen und 52 projekthaften
Lernaufgaben, 46 prüfenden Kontrollaufgaben
mit allen Lösungen und vielen Abbildungen

Viewegs Fachbücher der Technik



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

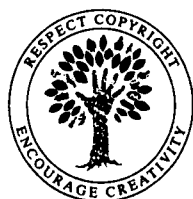
1. Auflage Dezember 2003
- 2., überarbeitete und ergänzte Auflage Juni 2005
- 3., überarbeitete und ergänzte Auflage Februar 2007

Alle Rechte vorbehalten

© Friedr. Vieweg & Sohn Verlag | GWV Fachverlage GmbH, Wiesbaden 2007

Lektorat: Reinhard Dapper

Der Vieweg Verlag ist ein Unternehmen von Springer Science+Business Media.
www.vieweg.de



Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlags unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Umschlaggestaltung: Ulrike Weigel, www.CorporateDesignGroup.de
Technische Redaktion: Hartmut Kühn von Burgsdorff, Wiesbaden
Druck und buchbinderische Verarbeitung: Wilhelm & Adam, Heusenstamm
Gedruckt auf säurefreiem und chlorfrei gebleichtem Papier.
Printed in Germany

ISBN 978-3-8348-0266-8

Vorwort: Buchkonzeption und Lernmöglichkeiten

In einigen elektrotechnischen und maschinenbautechnischen Berufen und in entsprechenden Studiengängen ist das Fachgebiet Automatisierungstechnik eng verknüpft mit Speicherprogrammierbaren Steuerungen in der Ausführung als Hardware- oder Software-SPS. Für Schüler/Studenten besteht über die Laborübungen in der Schule/Hochschule hinaus oftmals ein Bedarf an zusätzlichen *Übungsaufgaben* und einer darauf abgestimmten *informativen Arbeitshilfe*, die es ihnen erleichtert, in das selbstständige und systematische Bearbeiten von Automatisierungsaufgaben hineinzufinden. Das vorliegende Buch mit seiner speziellen Konzeption, bestehend aus Übersichten, Übungsaufgaben, Lösungsvorschlägen zur Selbstkontrolle und Simulationsmöglichkeiten, kann helfen, diese Lücke zu schließen.

Nach dem schnellen Anfangserfolg liegt dieses Übungsbuch nun in der 3. Auflage vor. Änderungen gegenüber der zweiten Auflage hat es nur in Kapitel 7 "Lineare Ablaufsteuerungen mit Betriebsartenteil" gegeben. Das bisherige Bedienfeld wurde um die Betriebsart „Einrichten“ erweitert und neu gestaltet. Der zugehörige Funktionsbaustein FB24 wurde an das neue Bedienfeld angepasst.

Übersichten

Jedes der neun Programmierkapitel beginnt mit einer *tabellenbuchartig gestalteten Übersicht* zu SPS-Grundlagen. Dazu zählen beispielsweise Regeln für die Bausteinauswahl (Funktion FC oder Funktionsbaustein FB), Anleitungen zur Variablendeklaration für ein an der SPS-Norm DIN EN 61131-3 orientiertes Programmieren, anwendbare Lösungsmethoden zur Umsetzung einer Aufgabenstellung in ein Programm und ein auf die im Kapitel angebotenen Übungsaufgaben abgestimmter STEP 7-Befehlsvorrat, der sich in Schritten bis zur Programmierung von Ablaufsteuerungen, auch unter Verwendung vorgefertigter Bausteine für Ablaufketten und Betriebsartenwahl, erweitert. Die Übersichten enthalten nur Fakten aber keine Erklärungen, für solche sei auf den Unterricht bzw. die Vorlesungen oder auf das weit verbreitete Lehrbuch *Automatisieren mit SPS, Theorie und Praxis*, erschienen im Vieweg Verlag, verwiesen. Jede Übersicht schließt ab mit einem Beispiel und seiner vollständigen Lösung zum Nachmachen und zur Vorbereitung auf die Übungsaufgaben.

Übungsaufgaben = Lernaufgaben + Kontrollaufgaben

Die Übungsaufgaben umfassen einfache und komplexe Problemstellungen, die hier in *projektorientierte Lernaufgaben* und *prüfende Kontrollaufgaben* unterschieden werden und die durch eine stets angegebene Lösungsleitlinie zu selbstständigem Lernen anleiten soll. Die Übungsaufgaben sind neutral, d.h. *unabhängig von SPS- und Programmiersystemen* gestellt.

Die Beschäftigung mit den Lernaufgaben hat die Durchdringung und Aneignung der in den Übersichten angebotenen Grundlagen zum Ziel und erfordert einen entsprechenden Zeitaufwand. Zur Eigenkontrolle selbst erarbeiteter Lösungen ist eine Hardware-SPS nicht erforderlich, wohl aber ein Programmiersystem mit Simulationsprogramm. Dies kann z.B. STEP 7 mit PLCSIM (Siemens AG) oder CoDeSys (3S-Smart Software Solutions GmbH) sein.

Das Bearbeiten der Kontrollaufgaben dient der Selbstkontrolle in Hinblick auf eine schnelle Verfügbarkeit des Gelernten in Prüfungssituationen. Im Unterschied zu den Lernaufgaben, die mit einem SPS-Programm abschließen, verlangen die Kontrollaufgaben nur die grundsätzlichen Lösungsschritte ohne Programmausführung.

Umfang der Lösungsvorschläge:

Allgemeingültige Darstellungen, die je nach Aufgabenstellung eine Zuordnungstabelle für SPS-Eingänge/-Ausgänge, einen Funktionsplan, einen Ablauf-Funktionsplan mit Transitionstabelle sowie ggf. ein Funktionsdiagramm umfassen können. Ort: Buch, Kap. 10

Programmdarstellungen für STEP 7 bestehend aus Deklarationstabellen für Variablen, einer Begründung für die Auswahl des verwendeten Bausteintyps (Funktion FC oder Funktionsbaustein FB) und die Bausteinaufrufe im Hauptprogramm OB1. Ort: Buch, Kap. 10

STEP 7-Programme der Beispiele und Lernaufgaben (gepackt, entpacken in STEP 7 über Dearchivieren) zur Ausführung auf einer S7-SPS oder unter PLCSIM. Ort: CD in Buchrückseite und über Internetadressen: www.vieweg.de oder www.automatisieren-mit-sps.de

CoDeSys-Programme nach IEC 61131-3-Standard. Die Autoren sind dem Wunsch vieler Lehrkräfte nach Siemens-unabhängigen Lösungen nachgekommen und haben für alle Lernaufgaben eine CoDeSys-Lösung ins Netz gestellt. Ort: www.automatisieren-mit-sps.de

Pdf-Datei mit Lösungen für die Kontrollaufgaben. Ort: CD in Buchrückseite und über beide Webadressen

Programmierung und Simulation, Softwarevoraussetzungen

Aufgabenlösungen können nicht rein theoretisch bleiben, sondern verlangen nach einer Ausführungskontrolle. Im Laborbetrieb der Schule/Hochschule steht dafür eine entsprechende Ausrüstung zur Verfügung stehen. Für den eigenen Computer zu Hause kann eine Schüler-Studenten-Version der STEP 7-Software (Vollversion mit PLCSIM) kostengünstig über die Schule/-Hochschule bezogen werden. Zur zeitlichen Überbrückung kann auch die im Buch vorne beiliegende 14-Tage-Version von STEP 7 benutzt werden. Dem Wunsch vieler Leser nach einer eingeschränkten STEP 7 Version mit unbegrenzter Laufzeit kann leider nicht nachgekommen werden, obwohl dies auch der Wunsch der Autoren ist.

Das SPS-Programmiersystem CoDeSys lässt sich von der Homepage der Firma 3S unter www.3s-software.com nach Registrierung und Passwortmitteilung kostenlos herunterladen.

Eine neue Qualität erhalten Simulationen durch dynamisierte Anlagenmodelle passend zu den Aufgabenstellungen. Die in der Buchrückseite beiliegende CD enthält zwei funktionsfähige Anlagen-Simulationen, die in eine SIMIT Runtime-Version integriert sind und alle Funktions- und Kopplungsmöglichkeiten der Vollversion des Programms erlaubt. Die Simulationsbeispiele sind zum Ausprobieren gedacht (Anleitung siehe beiliegende CD in Buchrückseite).

Um den Benutzern dieses Buches den Arbeitsaufwand zum Erstellen der Anlagenmodelle zu ersparen, stehen Anlagenmodelle für alle Beispiele und viele Lernaufgabe kostenfrei zur Verfügung. Ort: CD in Buchrückseite und über die beiden genannten Internetadressen.

Zur Ausführung der Anlagensimulationen braucht man im Gegensatz zu den Demo-Beispielen zusätzlich zur STEP7-Software das Simulationsprogramm SIMIT, das koppelbar ist mit einer echten S7-SPS oder mit S7-PLCSIM. Im Anhang des Buches findet man eine kurzgefasste Übersicht zu SIMIT.

Die Autoren des Buches bedanken sich sehr herzlich für die Unterstützung im Vieweg Verlag und bei Herrn Dipl. Phys. Franz Deutel von der Firma Siemens AG. Verbesserungsvorschläge aus dem Leserkreis sind uns immer willkommen.

Mannheim, Ellerstadt, im Januar 2007

Günter Wellenreuther
Dieter Zastrow

Inhaltsverzeichnis

| | |
|--|----|
| 1. Einführung: Übersicht | 2 |
| 1.1 Einfache Programme im Baustein PROGRAMM (OB 1) | 2 |
| 1.1.1 Beispiel: Anlegen eines Projekts mit dem Namen "Beispiel" | 3 |
| 1.2 Programmaufbau und Variablendeklaration | 5 |
| 1.2.1 Beispiel: Pressensteuerung | 6 |
| 2. Zusammengesetzte Grundverknüpfungen, SPS-Programm aus Funktionstabellen: Übersicht | 8 |
| 2.1 Beispiel: Generatorüberwachung | 10 |
| 2.2 Lernaufgaben | 12 |
| Lernaufgabe 2.1: Wahlfreie Schaltstellen | 12 |
| Lernaufgabe 2.2: Siloentleerung | 12 |
| Lernaufgabe 2.3: Lüfterüberwachung | 13 |
| Lernaufgabe 2.4: Füllung zweier Vorratsbehälter | 13 |
| Lernaufgabe 2.5: 7-Segment-Anzeige | 14 |
| Lernaufgabe 2.6: Durchlauferhitzer | 14 |
| Lernaufgabe 2.7: Code-Wandler | 15 |
| 2.3 Kontrollaufgaben 1 - 5 | 16 |
| 3. Speicherfunktionen, Flankenbewertung, Umwandlung von Schütz- und elektropneumatischen Steuerungen: Übersicht | 18 |
| 3.1 Beispiel: Schützschaltung mit gegenseitiger Verriegelung | 22 |
| 3.2 Lernaufgaben | 24 |
| Lernaufgabe 3.1: Torsteuerung | 24 |
| Lernaufgabe 3.2: Vier Vorratsbehälter | 25 |
| Lernaufgabe 3.3: Poliermaschine | 25 |
| Lernaufgabe 3.4: Biegewerkzeug | 26 |
| Lernaufgabe 3.5: Verpackungsrollenbahn | 28 |
| Lernaufgabe 3.6: Belüftungsanlage | 29 |
| Lernaufgabe 3.7: Reklamebeleuchtung | 30 |
| Lernaufgabe 3.8: Verstift-Einrichtung | 31 |
| Lernaufgabe 3.9: Drei Zylinder-Steuerung | 32 |
| 3.3 Kontrollaufgaben 1 - 9 | 33 |

| | |
|---|-----|
| 4. Zeitfunktionen, Taktsignale, Zählfunktionen, freigrafischer Funktionsplan: | |
| Übersicht | 38 |
| 4.1 Beispiel: Überwachung eines Mengenverhältnisses | 44 |
| 4.2 Lernaufgaben | 46 |
| Lernaufgabe 4.1: Stern-Dreieck-Anlauf der Antriebe einer Kompressoranlage... | 46 |
| Lernaufgabe 4.2: Anlassersteuerung | 46 |
| Lernaufgabe 4.3: Automatische Stern-Dreieck-Wendeschtung | 47 |
| Lernaufgabe 4.4: Füllmengenkontrolle | 48 |
| Lernaufgabe 4.5: Zerkleinerungsanlage | 49 |
| Lernaufgabe 4.6: Zwei Zylinder-Ansteuerung mit Zeitfunktionen | 50 |
| Lernaufgabe 4.7: Verkehrs-Lauflichtanlage | 51 |
| Lernaufgabe 4.8: Palettierungs-Anlage | 52 |
| Lernaufgabe 4.9: Parkhauseinfahrt | 52 |
| 4.3 Kontrollaufgaben 1 - 8 | 54 |
| 5. Übertragungsfunktionen, Vergleichsfunktionen, Sprünge: Übersicht | 60 |
| 5.1 Beispiel: Drehzahl- und Stillstandsüberwachung einer mit Schutzgitter gesicherten Maschine | 63 |
| 5.2 Lernaufgaben | 66 |
| Lernaufgabe 5.1: Auswahl-Standard-Funktion FC 36 SEL | 66 |
| Lernaufgabe 5.2: Wählbare Öffnungszeit für eine Klebedüse | 66 |
| Lernaufgabe 5.3: Auswahl-Standard-Funktion FC 25 MAX | 67 |
| Lernaufgabe 5.4: AUF-AB-Zähler mit parametrierbaren Grenzen | 67 |
| Lernaufgabe 5.5: Anzeige der Durchlaufgeschwindigkeit | 68 |
| Lernaufgabe 5.6: Begrenzer mit Indikator | 69 |
| Lernaufgabe 5.7: Überwachung der Walzgeschwindigkeit | 70 |
| 5.3 Kontrollaufgaben 1 - 6 | 71 |
| 6. Lineare Aaufsteuerungen ohne Betriebsartenteil: Übersicht | 74 |
| 6.1 Beispiel: Bohrmaschine | 79 |
| 6.2 Lernaufgaben | 84 |
| Lernaufgabe 6.1: Bar-Code-Stempeleinrichtung | 84 |
| Lernaufgabe 6.2: Reaktionsprozess | 85 |
| Lernaufgabe 6.3: Bördelvorrichtung | 86 |
| Lernaufgabe 6.4: Funktionsdiagramm | 87 |
| Lernaufgabe 6.5: Reinigungsbad bei der Galvanisierung | 88 |
| 6.3 Kontrollaufgaben 1 - 5 | 90 |
| 7. Lineare Ablaufsteuerung mit Betriebsartenteil: Übersicht | 94 |
| 7.1 Beispiel: Biegewerkzeug | 101 |
| 7.2 Lernaufgaben | 104 |
| Lernaufgabe 7.1: Abfüllanlage | 104 |
| Lernaufgabe 7.2: Zubringeinrichtung für Verpackungen | 105 |
| Lernaufgabe 7.3: Tablettenabfülleinrichtung | 106 |
| Lernaufgabe 7.4: Schotterwerk | 107 |
| Lernaufgabe 7.5: Los-Verpackungsanlage | 108 |
| Lernaufgabe 7.6: Rührkessel | 110 |
| 7.3 Kontrollaufgaben 1 - 6 | 111 |

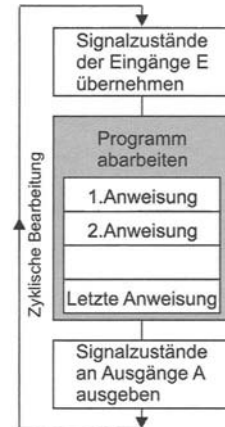
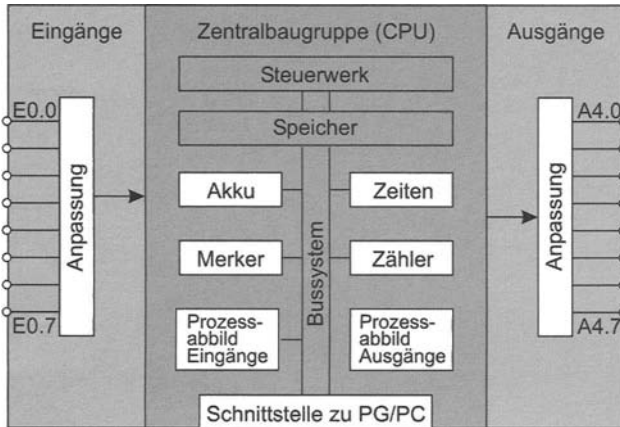
| | |
|---|------------|
| 8. Ablaufsteuerungen mit Verzweigungen: Übersicht | 114 |
| 8.1 Beispiel: Autowaschanlage | 117 |
| 8.2 Lernaufgaben | 122 |
| Lernaufgabe 8.1: Sortieranlage | 122 |
| Lernaufgabe 8.2: Bedarfsampelanlage | 123 |
| Lernaufgabe 8.3: Türschleuse | 124 |
| Lernaufgabe 8.4: Speiseaufzug | 125 |
| Lernaufgabe 8.5: Chargenprozess | 126 |
| 8.3 Kontrollaufgaben 1 - 5 | 129 |
| 9. Bausteinauswahl: FB oder FC? Übersicht | 133 |
| 9.1 Beispiel: Motorgruppenüberwachung | 134 |
| 9.2 Lernaufgaben | 136 |
| Lernaufgaben 9.1: Belüftungsanlage | 137 |
| Lernaufgaben 9.2: Schlossschaltung | 137 |
| Lernaufgaben 9.3: Torsteuerung | 137 |
| Lernaufgaben 9.4: Prägevorrichtung | 138 |
| 9.3 Kontrollaufgaben 1 - 3 | 140 |
| 10. Lösungsvorschläge Lernaufgaben | 142 |
| Hinweise zu den Lösungen | 142 |
| 10.1 Lösungen der Lernaufgaben Kapitel 1: Keine Lernaufgaben in Kapitel 1 | 142 |
| 10.2 Lösungen der Lernaufgaben Kapitel 2 | 143 |
| 10.3 Lösungen der Lernaufgaben Kapitel 3 | 156 |
| 10.4 Lösungen der Lernaufgaben Kapitel 4 | 173 |
| 10.5 Lösungen der Lernaufgaben Kapitel 5 | 193 |
| 10.6 Lösungen der Lernaufgaben Kapitel 6 | 202 |
| 10.7 Lösungen der Lernaufgaben Kapitel 7 | 223 |
| 10.8 Lösungen der Lernaufgaben Kapitel 8 | 242 |
| 10.9 Lösungen der Lernaufgaben Kapitel 9 | 253 |
| Anhang: Anlagensimulation in der Automatisierungstechnik | 257 |
| Sachwortverzeichnis | 261 |
| Was bietet die beiliegende CD? | 262 |

1 Einführung

1.1 Einfache Programme im Baustein PROGRAMM (OB 1)

- SPS-System, Grundverknüpfungen, Programmierung (STEP 7)
- Programmtest durch Simulation (PLCSIM)

SPS-Aufbau und zyklische Programmbearbeitung



Logische Grundverknüpfungen in verschiedenen Darstellungen

| Funktion | Zeitdiagramm | Funktionsplan (FUP) | Kontaktplan (KOP) | AWL |
|---|--------------|---------------------|-------------------|--|
| UND $A = E1 \wedge E2$ $A = E1 \& E2$ $A = E1 E2$ | | | | $U E 0.1$ $U E 0.2$ $= A 4.0$ |
| ODER $A = E1 \vee E2$ | | | | $O E 0.1$ $O E 0.2$ $= A 4.0$ |
| NICHT $A = \bar{E}$ | | | | $UNE 0.0$ $= A 4.0$ |
| Ausgangs-NEGATION $A = \overline{E1 \wedge E2}$ | | | | $U E 0.1$ $U E 0.2$ NOT $= A 4.0$ |

Programmierung

SPS-Programme werden mit Hilfe einer Programmiersoftware auf einem PC erstellt und danach in die SPS übertragen. Weit verbreitete Programmiersysteme sind STEP 7 (speziell für Siemens-SPS) und CoDeSys (allg. für IEC 1131-Norm, Firma 3S-Smart Software Solutions). SPS-Programme werden als Projekte angelegt, unter einem Dateinamen gespeichert und in Bausteine programmiert. Der Organisationsbaustein OB 1 für zyklische Programmbearbeitung in STEP7 entspricht etwa dem Norm-Bausteintyp PROGRAMM. In diesen Bausteinen ist der Gebrauch von SPS-Operanden wie Eingänge (E), Ausgänge (A) zulässig und notwendig.

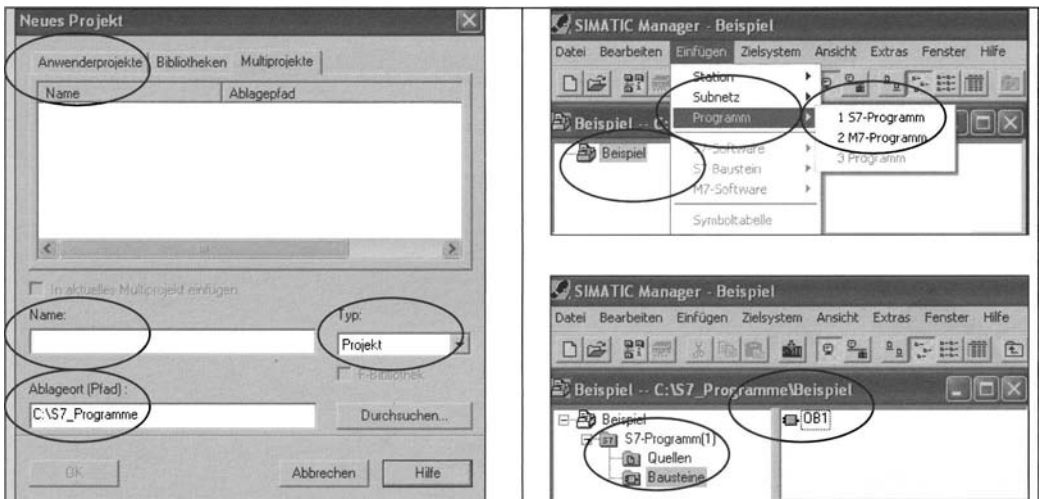
1.1.1

Beispiel

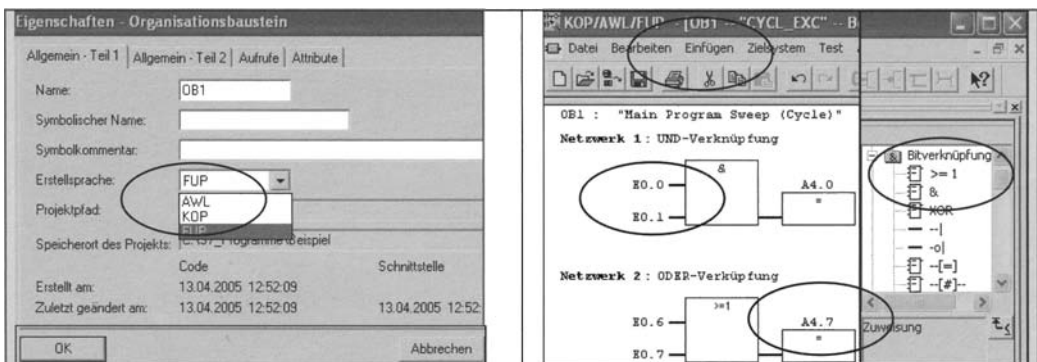
- SPS-Programm und Simulation

Anlegen eines Projekts mit dem Namen "Beispiel"

1. Start des SIMATIC Managers.
2. *Datei > Neu...*
3. Im Fenster Neues Projekt ein Anwenderprojekt anlegen unter dem Namen "Beispiel", OK.
4. *Einfügen > Programm > S7-Programm* auswählen.
5. Objekt-Hierarchie öffnen durch Doppelklick auf "Beispiel", danach auf "S7-Programm(1)".
6. Objekt "Bausteine" markieren, im rechten Feld erscheint OB 1, durch Doppelklick öffnen.

**Programmieren von Grundverknüpfungen im OB 1**

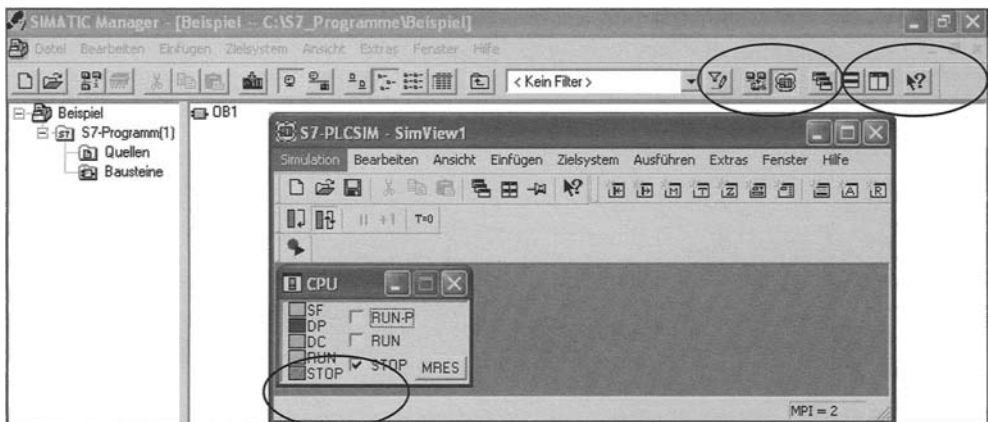
1. Geschlossener OB 1 durch Anklicken zum Programmieren öffnen.
2. Auswahl der Programmiersprache, z.B. FUP, OK.
3. Im Fenster KOP/AWL/FUP Menü Ansicht > Anzeigen mit > Kommentar deaktivieren.
4. In Feld Programmelemente Verzeichnis „Bitverknüpfung“ öffnen.
5. Doppelklick auf UND-Gatter, Ausgang des UND-Gatter markieren, Doppelklick auf Zuweisung
6. Fragezeichen durch SPS-Operanden ersetzen. Neues Netzwerk unter Menü Einfügen.
7. ODER-Gatter einfügen und beschriften.



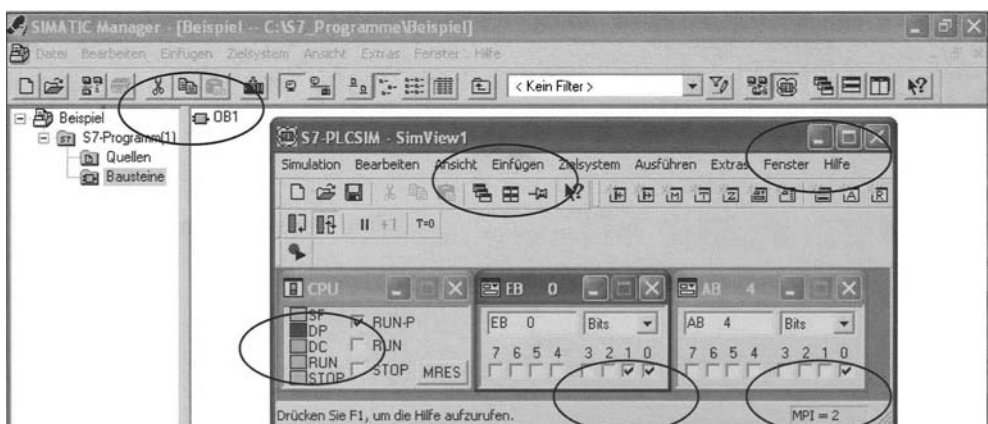
Programmtest durch Simulation (PLCSIM)

Zum Testen eines SPS-Programms kann eine Hardware-SPS durch Simulation ersetzt werden:

1. Speichern des im KOP/AWL/FUP-Editor erstellten *Projekts* > *Klick* auf Diskettensymbol.
2. KOP/AWL/FUP-Editor beenden: *Datei* > *Schließen*, *Datei* > *Beenden*.
3. Im SIMATIC Manager das Icon "Simulation ein/aus" im Menübereich suchen. ?-Zeichen markieren und auf das Icon ziehen, Hilfetext lesen: PLCSIM simuliert eine CPU.
4. Anklicken des Icons "Simulation ein/aus", es öffnet sich S7-PLCSIM und zeigt eine CPU im Betriebszustand STOP. S7-PLCSIM Fenster verkleinern.



5. Signaleingänge einfügen: Menü *Einfügen* > *Eingang*, es erscheint EB 0 (E 0.0 bis E 0.7).
6. Signalausgänge einfügen: Menü *Einfügen* > *Ausgang*, es erscheint AB 0, in AB 4 ändern.
7. S7-PLCSIM-Fenster minimieren, Baustein OB 1 markieren und in Simulations-CPU durch Anklicken des Icons "Laden" übertragen.
8. S7-PLCSIM-Fenster aus der Taskleiste zurückholen und CPU durch Anklicken von "RUN-P" starten. Anzeige schaltet auf "RUN".
9. Programmtest durch Anklicken der Eingänge E 0.0, E 0.1, der Ausgang A 4.0 wird aktiv.



Eigene Übungsbeispiele

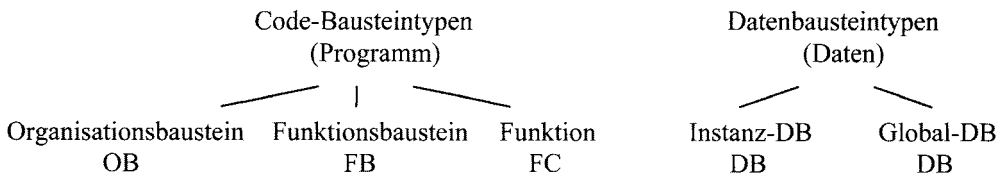
Grundverknüpfungen mit erweiterten Bausteineingängen und Negationen als Funktionsplan programmieren und durch Simulation testen. AWL- und KOP-Darstellung des Programms durch Umschalten im KOP/AWL/FUP-Editor im Menü Ansicht.

1.2 Programmaufbau und Variablendeklaration

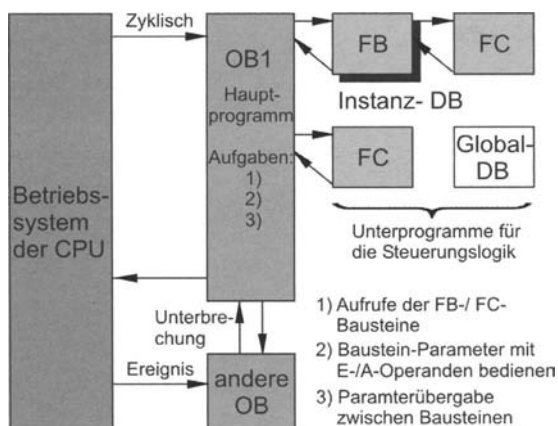
Die im Kapitel 1.1 angewendete Programmierweise entspricht der klassischen SPS-Programmierung aus der Anfangszeit dieser Technologie, die auch heute noch funktioniert aber den modernen Programmiersystemen nicht mehr gerecht wird. Ihr typisches Kennzeichen ist die Verwendung der direkten SPS-Operanden (Eingänge, Ausgänge) und Merker in Steuerungsbefehlen (Anweisungen). Die SPS-Norm IEC 1131-3 (Erscheinungsjahr 1993) verlangt aus guten Gründen, die Deklaration von Variablen mit zugehörigen Datentypen in den eigentlichen Steuerungsprogrammen, also dort, wo die Befehlslogik vorherrscht. Natürlich müssen auch weiterhin SPS-Eingänge und –Ausgänge konkret angegeben werden, aber nur noch zum Zweck der eingangsseitigen Signaleinführung in das Programm und zur ausgangsseitigen Entgegennahme von Programmresultaten. In der Steuerungslogik von Anwenderprogrammen sollen die SPS-Operanden nicht mehr vorkommen (siehe Bild unten).

Zur Unterbringung der Steuerungslogik (Programmcode) stehen Code-Bausteintypen mit verschiedenen Eigenschaften zur Verfügung. Diese Bausteine bestehen aus einem Deklarationsteil zur Bestimmung der bausteinlokalen Variablen und ihrer Datentypen und einem Anweisungsteil für die Programmlogik, die realisiert werden soll.

Übersicht Bausteine in STEP 7:



Aufgabenverteilung für Bausteine:



Kriterien der Bausteinauswahl:

Funktion FC

FCs sind parametrierbare Programmbausteine ohne eigenen Datenbereich. FCs genügen, wenn keine interne Speicherfunktion nötig ist oder die Speicherung einer Variablen nach außen verlagert werden kann.

Funktionsbaustein FB

FBs sind parametrierbare Programmbausteine, denen beim Aufruf ein eigener Speicherbereich (Instanz-DB) zugewiesen wird. FBs sind notwendig, wenn ein speicherndes Verhalten einer bausteininternen Variablen nötig ist.

Zu jeder Regel gibt es eine Ausnahme: Einfachst-Programme, die beispielsweise nur die Funktion von SPS-Befehlen zu Lehrzwecken zeigen sollen, können auch weiterhin direkt im OB 1 unter Verwendung von SPS-Operanden programmiert werden. Der Baustein OB 1 ist das Hauptprogramm, in dem E-/A-Adressen zulässig und erforderlich sind.

Variablendeklaration in FC- und FB-Bausteinen

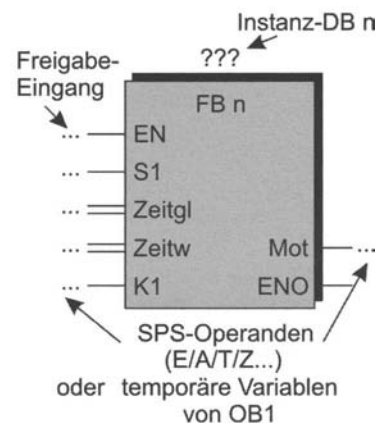
Deklarationstypen für lokale (nur im betreffenden Baustein geltende) Variablen:

- Deklaration **IN**: Der Eingangsparameter kann innerhalb des Code-Bausteins nur abgefragt werden.
- Deklaration **OUT**: Der Ausgangsparameter soll innerhalb des Code-Bausteins nur beschrieben werden.
- Deklaration **IN_OUT**: Der Durchgangsparameter kann innerhalb des Code-Bausteins abgefragt und beschrieben werden.
- Deklaration **STAT**: Interne Zustandsvariable zum Abspeichern von Daten (Gedächtnisfunktion) über den Zyklus einer Bausteinbearbeitung hinaus. Nur bei Funktionsbausteinen FB mit Instanz-Datenbausteinen DB.
- Deklaration **TEMP**: Interne temporäre Variable zum Zwischenspeichern von Ergebnissen innerhalb eines Zyklus der Bausteinbearbeitung und speziell im OB 1 zur Datenübergabe zwischen aufgerufenen Bausteinen.

Deklarationstabelle: FB n (innere Sicht der Deklaration)

| Name | Datentyp | Anfangswert |
|---------------|----------|-------------|
| IN | | |
| S1 | Bool | FALSE |
| Zeitgl | Timer | |
| Zeitw | S5Time | S5T#0MS |
| IN_OUT | | |
| K1 | Bool | FALSE |
| OUT | | |
| Mot | Bool | FALSE |
| STAT | | |
| SP | Int | 0 |
| TEMP | | |
| HO | Word | |

Aufruf des FB im OB1: (äußere Sicht der Deklaration)



Parameterbeschaltung bei Aufruf des Funktionsbausteins FB

Bei der Programmabarbeitung werden die den Variablen zugeordneten E-/A-Adressen der SPS verwendet. Der Baustein kann mehrmals mit verschiedenen Operandensätzen aufgerufen werden. Bei jedem Funktionsbausteinaufruf FB ist ein Instanz-Datenbaustein DB zu generieren.

EN/ENO

Bei EN = 1 oder unbeschaltet wird der Baustein bearbeitet. Tritt dabei kein Fehler auf, wird ENO = 1 gesetzt. Bei EN = 0 wird der Baustein nicht bearbeitet und ENO = 0 gesetzt.

Symboltabelle : Deklaration für globale (im gesamten Programm geltende) Variablen

| Symbol | Adresse | Datentyp |
|--------------|---------|----------|
| Endschalter | E 1.0 | BOOL |
| Schütz | A 4.0 | BOOL |
| Mengenzähler | Z1 | COUNTER |
| Zeitglied | T1 | TIMER |

In STEP7 können für SPS-Operanden wie Eingänge E, Ausgänge A, Merker M, Timer T, Zähler Z in der Symboltabelle symbolische Namen vergeben werden, die immer fest mit einer SPS-Operandenadresse und einem entsprechenden Datentyp verbunden sind.

1.2.1 Beispiel
 • Aufgabenteilung zwischen Hauptprogramm OB 1 und Funktion FC

Pressensteuerung

Eine Excenterpresse führt den Arbeitshub nur aus, wenn das Schutzgitter geschlossen ist und der Start-Taster S1 betätigt wird. Die Überwachung des Schutzgitters erfolgt durch die Endschalter S2 mit Öffnerkontakt und S3 mit Schließerkontakt. Nur wenn die beiden entfernten liegenden Kontakte S2 und S3 betätigt sind, gilt das Schutzgitter als geschlossen. Es sind zwei baugleiche Pressen (Presse A und Presse B) anzusteuern.

Zuordnungstabelle der Eingänge und Ausgänge:

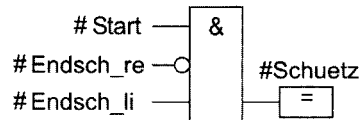
| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | | Adresse |
|------------------------------|--------|----------|--------------------|--------|---------|
| Start-Taster, Presse A | S1 | BOOL | betätigt | S1 = 1 | E 0.1 |
| Endschalter rechts, Presse A | S2 | BOOL | betätigt | S2 = 0 | E 0.2 |
| Endschalter links, Presse A | S3 | BOOL | betätigt | S3 = 1 | E 0.3 |
| Start-Taster, Presse B | S4 | BOOL | betätigt | S4 = 1 | E 0.4 |
| Endschalter rechts, Presse B | S5 | BOOL | betätigt | S5 = 0 | E 0.5 |
| Endschalter links, Presse B | S6 | BOOL | betätigt | S6 = 1 | E 0.6 |
| Ausgangsvariable | | | | | |
| Schütz, Presse A | K1 | BOOL | Arbeitshub | K1 = 1 | A 4.0 |
| Schütz, Presse B | K2 | BOOL | Arbeitshub | K2 = 1 | A 4.1 |

Deklarationstabelle: FC10

| Name | Datentyp | Kommentar |
|------------|----------|-----------|
| IN | | |
| Start | Bool | |
| Endsch_re | Bool | |
| Endsch_li | Bool | |
| OUT | | |
| Schuetz | Bool | |

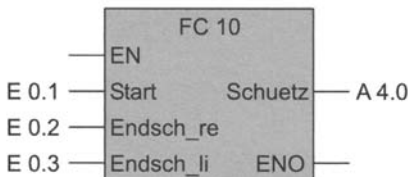
Programm: FC 10

Beim Programmieren des FC10 werden die deklarierten Variablen automatisch mit dem Vorsatzzeichen # zur Kennzeichnung ihres lokalen Geltungsbereichs im FC10 versehen.

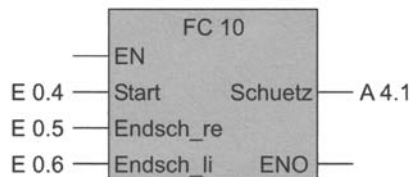


OB 1: Zweimaliger Aufruf des FC 10 mit verschiedenen Operandenadressen. Der OB 1 übernimmt die zyklische Programmaufrufe mit Versorgung der Bausteinparameter des FC 10 und damit die Ansteuerung von zwei Maschinen. Das Steuerungsprogramm selbst steht im FC 10.

Presse A



Presse B



Übungen: Ein entsprechendes Beispiel für einen Funktionsbaustein FB kann erst nach Einführung weiterer Programmierfunktionen gezeigt werden. In den folgenden Kapiteln wird die Verwendung von FC- und FB-Bausteinen schrittweise eingeführt. Wer SPS-Vorkenntnisse hat und es schon jetzt genau wissen will, wann FBs und nicht FCs einzusetzen sind bzw. wann FCs entgegen dem ersten Anschein doch ausreichen, gehe nach Kap 9 zu Aufgaben mit entsprechender Trennschärfe. Sonst gehe man weiter zu Kapitel 2.

2 Zusammengesetzte Grundverknüpfungen, SPS-Programm aus Funktionstabellen

Zusammengesetzte logische Grundverknüpfungen

| | | |
|--|--|---|
| <p>UND-vor-ODER-Verknüpfung</p> <p>1) Allgemeiner Fall siehe nachfolgend bei DNF</p> $A = E1E2\bar{E3} \vee E1\bar{E2}E3$ <p>2) Spezieller Fall: Antivalenz (Exklusiv-ODER)</p> $A = E1 \neq E2$ | | <p>U E0.1 U E0.2 UN E0.3 O U E0.1 UN E0.2 O E0.3 = A4.0</p> <p>X E0.1 X E0.2 = A5.0</p> |
| <p>ODER-vor-UND-Verknüpfung</p> $A = (E1 \vee E2) \wedge (\bar{E1} \vee \bar{E2}) \wedge E3$ | | <p>U(O E0.1 O E0.2) U(ON E0.1 ON E0.2) U E0.3 = A4.0</p> |

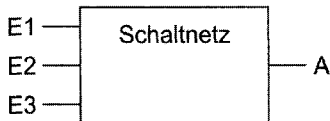
Zulässige Operanden und Datentypen bei Grundverknüpfungen

Operandenbereiche: E(Eingang), A(Ausgang), M(Merker), D(Datenbit), T(Zeitglied), Z(Zähler)

Zugehörige Datentypen: BOOL TIMER COUNTER

Beispiel: U E 0 U A 4.0 U M 5.0 U DBX 8.0 U T 1 U Z 1

Funktionstabelle als Hilfsmittel für den Entwurf von Verknüpfungssteuerungen



Mit n Eingangsvariablen ergeben sich 2ⁿ verschiedene Eingangskombinationen, denen gemäß Aufgabenstellung logische Ausgangszustände zugeordnet werden müssen.

| Zeile | E3 | E2 | E1 | A |
|-------|----|----|----|---|
| 00 | 0 | 0 | 0 | 0 |
| 01 | 0 | 0 | 1 | 0 |
| 02 | 0 | 1 | 0 | 0 |
| 03 | 0 | 1 | 1 | 1 |
| 04 | 1 | 0 | 0 | 0 |
| 05 | 1 | 0 | 1 | 1 |
| 06 | 1 | 1 | 0 | 1 |
| 07 | 1 | 1 | 1 | 1 |

Disjunktive Normalform DNF

Aus der Funktionstabelle kann die DNF-Schaltfunktion für den Ausgang A gebildet werden. Für jede Zeile der Tabelle, bei der die Ausgangsvariable A den Signalzustand "1" führt, wird ein UND-Term mit allen Eingangsvariablen E3 E2 E1 gebildet. Bei Signalzustand "0" wird die Variable E negiert, bei Signalzustand "1" nicht negiert notiert. Die komplette Schaltfunktion erhält man durch ODER-Verknüpfung der UND-Terme: UND-vor-ODER-Verknüpfung.

DNF für Ausgang A aus Funktionstabelle: $A = \overline{E3} E2 E1 \vee E3 \overline{E2} E1 \vee E3 E2 \overline{E1} \vee E3 E2 E1$

Vereinfachung von Schaltfunktionen mittels KVS-Diagramm

Das Ziel ist die minimierte Schaltfunktion für Ausgang A (kürzere Lösung der Aufgabe).

| <table border="1"> <thead> <tr> <th>Zeile</th> <th>E4</th> <th>E3</th> <th>E2</th> <th>E1</th> <th>A</th> </tr> </thead> <tbody> <tr><td>00₈</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>01₈</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>02₈</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>03₈</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>04₈</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>05₈</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>06₈</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>07₈</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> | Zeile | E4 | E3 | E2 | E1 | A | 00 ₈ | 0 | 0 | 0 | 0 | 0 | 01 ₈ | 0 | 0 | 0 | 1 | 0 | 02 ₈ | 0 | 0 | 1 | 0 | 0 | 03 ₈ | 0 | 0 | 1 | 1 | 1 | 04 ₈ | 0 | 1 | 0 | 0 | 0 | 05 ₈ | 0 | 1 | 0 | 1 | 1 | 06 ₈ | 0 | 1 | 1 | 0 | 1 | 07 ₈ | 0 | 1 | 1 | 1 | 1 | | <p>Minimierte DNF für den Ausgang A:</p> $A = E2 E1 \vee E3 E1 \vee E3 E2$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|-------|----|----|----|----|---|-----------------|---|---|---|---|---|-----------------|---|---|---|---|---|-----------------|---|---|---|---|---|-----------------|---|---|---|---|---|-----------------|---|---|---|---|---|-----------------|---|---|---|---|---|-----------------|---|---|---|---|---|-----------------|---|---|---|---|---|-----------------|--|---|---|---|---|-----------------|---|---|---|---|---|-----------------|---|---|---|---|---|-----------------|---|---|---|---|---|-----------------|---|---|---|---|---|-----------------|---|---|---|---|---|-----------------|---|---|---|---|---|-----------------|---|---|---|---|---|--|---|
| Zeile | E4 | E3 | E2 | E1 | A | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 00 ₈ | 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 01 ₈ | 0 | 0 | 0 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 02 ₈ | 0 | 0 | 1 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 03 ₈ | 0 | 0 | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 04 ₈ | 0 | 1 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 05 ₈ | 0 | 1 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 06 ₈ | 0 | 1 | 1 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 07 ₈ | 0 | 1 | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1"> <thead> <tr> <th>Zeile</th> <th>E4</th> <th>E3</th> <th>E2</th> <th>E1</th> <th>A</th> </tr> </thead> <tbody> <tr><td>00₈</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>01₈</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>02₈</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>03₈</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>04₈</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>05₈</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>06₈</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>07₈</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>10₈</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>11₈</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>12₈</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>13₈</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>14₈</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>15₈</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>16₈</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>17₈</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> | Zeile | E4 | E3 | E2 | E1 | A | 00 ₈ | 0 | 0 | 0 | 0 | 0 | 01 ₈ | 0 | 0 | 0 | 1 | 0 | 02 ₈ | 0 | 0 | 1 | 0 | 0 | 03 ₈ | 0 | 0 | 1 | 1 | 0 | 04 ₈ | 0 | 1 | 0 | 0 | 0 | 05 ₈ | 0 | 1 | 0 | 1 | 1 | 06 ₈ | 0 | 1 | 1 | 0 | 1 | 07 ₈ | 0 | 1 | 1 | 1 | 1 | 10 ₈ | 1 | 0 | 0 | 0 | 1 | 11 ₈ | 1 | 0 | 0 | 1 | 1 | 12 ₈ | 1 | 0 | 1 | 0 | 1 | 13 ₈ | 1 | 0 | 1 | 1 | 1 | 14 ₈ | 1 | 1 | 0 | 0 | 0 | 15 ₈ | 1 | 1 | 0 | 1 | 1 | 16 ₈ | 1 | 1 | 1 | 0 | 1 | 17 ₈ | 1 | 1 | 1 | 1 | 1 | | <p>Minimierte DNF für den Ausgang A:</p> $A = E3 E1 \vee E3 E2 \vee E4 \overline{E3}$ |
| Zeile | E4 | E3 | E2 | E1 | A | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 00 ₈ | 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 01 ₈ | 0 | 0 | 0 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 02 ₈ | 0 | 0 | 1 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 03 ₈ | 0 | 0 | 1 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 04 ₈ | 0 | 1 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 05 ₈ | 0 | 1 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 06 ₈ | 0 | 1 | 1 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 07 ₈ | 0 | 1 | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10 ₈ | 1 | 0 | 0 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 11 ₈ | 1 | 0 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 12 ₈ | 1 | 0 | 1 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 13 ₈ | 1 | 0 | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 14 ₈ | 1 | 1 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 ₈ | 1 | 1 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 16 ₈ | 1 | 1 | 1 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 17 ₈ | 1 | 1 | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Regeln des Vereinfachungsverfahrens:

1. In die nummerierten Felder des KVS-Diagramms sind die Signalwerten von A einzutragen.
2. Einkreisungen symmetrisch liegender 1-Felder zu 2er, 4er, 8er-Blöcken einzeichnen.
3. Jedes 1-Feld ist mindestens einmal einzukreisen. Möglichst große Einkreisungen finden.
4. Bei 2er oder 4er oder 8er-Einkreisungen entfallen genau 1 oder 2 oder 3 Variablen.
5. Es entfallen die Variablen, die in Einkreisungen negiert und auch nicht negiert vorkommen.

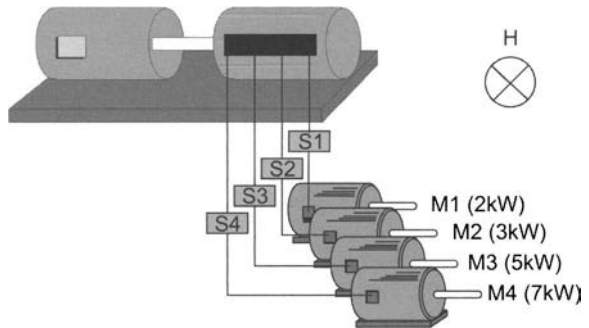
2.1

Beispiel

Generatorüberwachung

Ein Generator ist mit maximal 10 kW belastbar. Anschaltbar sind vier Motoren mit den Leistungen 2 kW, 3 kW, 5 kW und 7 kW. Die Motoren sind mit einem Drehzahlwächter ausgerüstet (S1, S2, S3, S4), die bei laufendem Motor ein 0-Signal an die SPS-Eingänge (E 0.1, E 0.2, E 0.3 und E 0.4) melden. Bei allen zulässigen Betriebskombinationen soll eine Meldeleuchte H durch den SPS-Ausgang A 4.0 eingeschaltet sein.

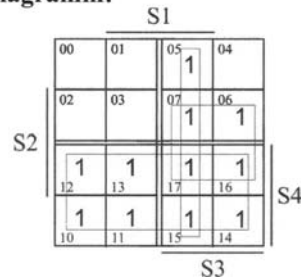
1. Es ist eine Funktionstabelle aufzustellen und aus dieser die Schaltfunktion für die Meldeleuchte H in disjunktiver Normalform anzugeben.
2. Mit Hilfe des KVS-Diagramms ist nach einer einfacheren Lösung für die Schaltfunktion zu suchen.
3. Das Meldeprogramm ist im Baustein FC 10 mit symbolischen Variablen zu programmieren und im OB1 mit den E-/A-Operanden zu beschalten.

**Systematischer Programmwurf****Funktionstabelle**

| Okta-Nr. | S4 S3 S2 S1 | H | Leistung |
|----------|-------------|---|----------|
| 00 | 0 0 0 0 | 0 | 17 kW |
| 01 | 0 0 0 1 | 0 | 15 kW |
| 02 | 0 0 1 0 | 0 | 14 kW |
| 03 | 0 0 1 1 | 0 | 12 kW |
| 04 | 0 1 0 0 | 0 | 12 kW |
| 05 | 0 1 0 1 | 1 | 10 kW |
| 06 | 0 1 1 0 | 1 | 9 kW |
| 07 | 0 1 1 1 | 1 | 7 kW |
| 10 | 1 0 0 0 | 1 | 10 kW |
| 11 | 1 0 0 1 | 1 | 8 kW |
| 12 | 1 0 1 0 | 1 | 7 kW |
| 13 | 1 0 1 1 | 1 | 5 kW |
| 14 | 1 1 0 0 | 1 | 5 kW |
| 15 | 1 1 0 1 | 1 | 3 kW |
| 16 | 1 1 1 0 | 1 | 2 kW |
| 17 | 1 1 1 1 | 1 | 0 kW |

DNF-Schaltfunktion aus Tabelle

$$H = \overline{S4} \overline{S3} \overline{S2} \overline{S1} \vee \overline{S4} \overline{S3} \overline{S2} S1 \vee \overline{S4} \overline{S3} S2 \overline{S1} \vee \overline{S4} \overline{S3} S2 S1 \vee \overline{S4} S3 \overline{S2} \overline{S1} \vee \overline{S4} S3 \overline{S2} S1 \vee \overline{S4} S3 S2 \overline{S1} \vee \overline{S4} S3 S2 S1 \vee S4 \overline{S3} \overline{S2} \overline{S1} \vee S4 \overline{S3} \overline{S2} S1 \vee S4 \overline{S3} S2 \overline{S1} \vee S4 \overline{S3} S2 S1 \vee S4 S3 \overline{S2} \overline{S1} \vee S4 S3 \overline{S2} S1 \vee S4 S3 S2 \overline{S1} \vee S4 S3 S2 S1$$

KVS-Diagramm:**Vereinfachte DNF-Schaltfunktion:**

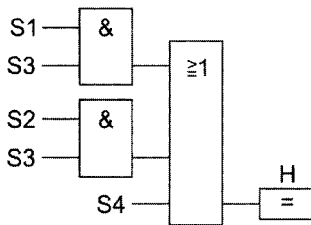
$$H = S4 \vee S3 S2 \vee S3 S1$$

Hinweis: Vergleiche mit der Lösung zu Beispiel 4.9 „Generatorüberwachung“ im Lehrbuch „Automatisieren mit SPS“. Dem dortigen Programm ist die ODER-vor-UND-Verknüpfung zu Grunde gelegt.

Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | | Adresse |
|--------------------------|--------|----------|--------------------|--------|---------|
| Drehzahlwächter 1 (2 kW) | S1 | BOOL | Motor 1 läuft | S1 = 0 | E 0.1 |
| Drehzahlwächter 2 (3 kW) | S2 | BOOL | Motor 2 läuft | S2 = 0 | E 0.2 |
| Drehzahlwächter 3 (5 kW) | S3 | BOOL | Motor 3 läuft | S3 = 0 | E 0.3 |
| Drehzahlwächter 4 (7 kW) | S4 | BOOL | Motor 4 läuft | S4 = 0 | E 0.4 |
| Ausgangsvariable | | | | | |
| Meldeleuchte | H | BOOL | Meldeleuchte an | H = 1 | A 4.0 |

Funktionsplan:

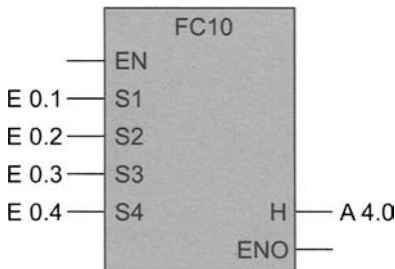


Deklarationstabelle FC 10:

| Name | Datentyp |
|------------|----------|
| IN | |
| S1 | Bool |
| S2 | Bool |
| S3 | Bool |
| S4 | Bool |
| OUT | |
| H | Bool |

Aufruf im OB1:

Funktionsplan-Darstellung:



AWL-Darstellung:

```

U E 0.1
= L 20.0
BLD 103
U E 0.2
= L 20.1
BLD 103
U E 0.3
= L 20.2
BLD 103
U E 0.4
= L 20.3
BLD 103
CALL FC 10
    S1 := L20.0
    S2 := L20.1
    S3 := L20.2
    S4 := L20.3
    H := A4.0
    
```

BLD 103 = Bildaufbauanweisung

Hinweis: Beim Aufruf von Bausteinen in der AWL-Darstellung können die SPS-Operanden auch direkt an die Eingänge des Bausteins geschrieben werden. Die Darstellung im Funktionsplan FUP ist dann allerdings nicht mehr möglich.

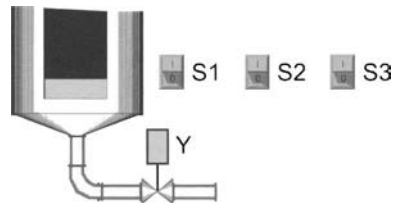
2.2

Lernaufgaben

Lernaufgabe 2.1: Wahlfreie Schaltstellen

Lös. S. 143

Das Ablassventil eines Silos soll von drei Schaltstellen aus (S1, S2 u. S3) über ein 24V-Elektromagnetventil Y wahlweise geöffnet bzw. geschlossen werden können (Wechselschaltungsverhalten von drei Schaltstellen aus). An den Schaltstellen werden einpolige Schalter verwendet.

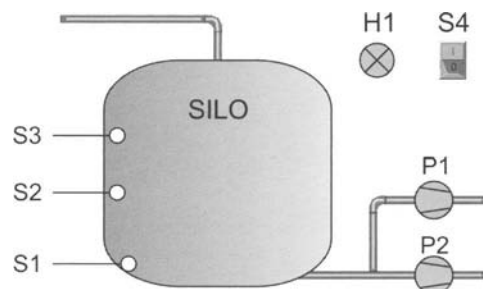
Technologieschema:

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Zeichnen Sie den Anschlussplan der Sensoren und Aktoren an die SPS.
3. Bestimmen Sie mit einer Funktionstabelle den Zusammenhang zwischen den Eingängen S1, S2, S3 und dem Ausgang Y.
4. Bestimmen Sie aus der Funktionstabelle die Schaltfunktion und den Funktionsplan.
5. Das Steuerungsprogramm soll in einem bibliotheksfähigen Baustein realisiert werden. Bestimmen Sie die Bausteinart und geben Sie die Deklarationstabelle an.
6. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.

Lernaufgabe 2.2: Siloentleerung

Lös. S. 144

Der Inhalt eines Silos kann über die Pumpen P1 und P2 entleert werden. Welche der beiden Pumpen bei der Entleerung des Silos eingeschaltet sind, ist abhängig vom Silofüllstand. Befindet sich der Füllstand unterhalb von Sensor S2, ist Pumpe P1 einzuschalten. Liegt der Füllstand zwischen Sensor S2 und Sensor S3, wird die Pumpe P2 eingeschaltet. Bei Füllstand oberhalb von S3 laufen beide Pumpen. Die Entleerung des Silos wird mit dem Schalter S4 ein- und ausgeschaltet.

Technologieschema:

Beim Auftreten eines Sensorfehlers (z.B. S3 meldet und S2 meldet nicht) werden beide Pumpen und eine Störungsanzeige H1 eingeschaltet.

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Bestimmen Sie mit einer Funktionstabelle den Zusammenhang zwischen den Eingängen S1, S2, S3 und den Ausgängen P1 und P2.
3. Ermitteln Sie aus der Funktionstabelle die Schaltfunktionen für P1, P2 bzw. H1 und zeichnen Sie den dazugehörigen Funktionsplan unter Berücksichtigung des Schalters S4.
4. Das Steuerungsprogramm soll in einem bibliotheksfähigen Baustein realisiert werden. Bestimmen Sie die Bausteinart und geben Sie die Deklarationstabelle an.

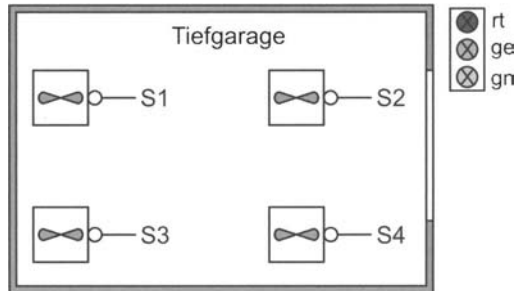
5. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.

Lernaufgabe 2.3: Lüfterüberwachung

Lös. S. 145

In einer Tiefgarage sind vier Lüfter installiert. Die Funktionsüberwachung der Lüfter erfolgt durch je einen Luftströmungswächter. An der Einfahrt der Tiefgarage ist eine Ampel angebracht. Sind alle vier Lüfter oder drei Lüfter in Betrieb, ist für eine ausreichende Belüftung gesorgt und die Ampel zeigt Grün. Bei Betrieb von nur zwei Lüftern schaltet die Ampel auf Gelb. Es dürfen nur Fahrzeuge ausfahren. Sind weniger als zwei Lüfter in Betrieb, muss die Ampel Rot anzeigen.

Technologieschema:



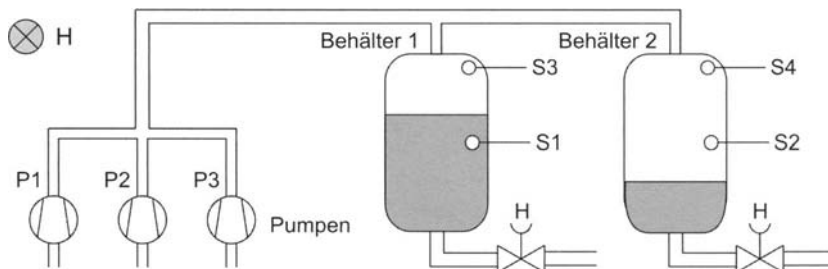
- Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge. Achten Sie dabei auf die richtige logische Zuordnung der Signalegeber.
- Bestimmen Sie mit einer Funktionstabelle den Zusammenhang zwischen den Eingängen S1 bis S4 und den Ausgängen rt, ge und gn.
- Ermitteln Sie aus der Funktionstabelle die Schaltfunktion für zwei der drei Ausgänge. Wählen Sie den Ausgang, der am günstigsten aus dem Signalzustand der beiden anderen Ausgänge gebildet werden kann.
- Zeichnen Sie den Funktionsplan zur Ansteuerung der drei Ausgänge.
- Das Steuerungsprogramm soll in einem bibliotheksfähigen Baustein realisiert werden. Bestimmen Sie die Bausteinart und geben Sie die Deklarationstabelle an.
- Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.

Lernaufgabe 2.4: Füllung zweier Vorratsbehälter

Lös. S. 147

Zwei Vorratsbehälter mit den Signalgebern S3 und S4 für die Vollmeldung und S1 und S2 für die Meldung halbvoll werden in beliebiger Reihenfolge entleert.

Technologieschema:



Die Füllung der Behälter erfolgt abhängig vom Füllstand durch die drei Pumpen P1, P2 und P3. Meldet entweder kein Signalgeber oder nur ein Signalgeber einen erreichten Füllstand, so sollen alle drei Pumpen P1 bis P3 laufen. Melden zwei Signalgeber einen entsprechenden Füllstand, so sollen zwei Pumpen die Füllung übernehmen. Wenn drei Signalgeber melden,

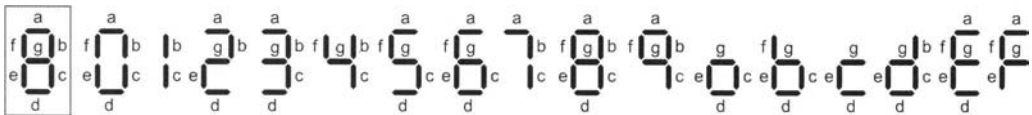
genügt es, wenn nur eine Pumpe die Füllung übernimmt. Melden alle vier Signalgeber, so sind die beiden Vorratsbehälter gefüllt und alle Pumpen bleiben ausgeschaltet. Tritt ein Fehler auf, der von einer widersprüchlichen Meldung der Signalgeber herrührt, so soll dies eine Meldeleuchte H anzeigen und keine Pumpe laufen. Bei der Zuordnung der eingeschalteten Pumpen ist auf eine möglichst gleiche Einschalthäufigkeit zu achten.

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Ermitteln Sie die Funktionstabelle für die Steuerungsaufgabe.
3. Minimieren Sie die Ansteuerfunktionen der Ausgänge mit einem KVS-Diagramm.
4. Zeichnen Sie den Funktionsplan der minimierten Ansteuerfunktionen.
5. Das Steuerungsprogramm soll in einem bibliotheksfähigen Baustein realisiert werden. Bestimmen Sie die Bausteinart und geben Sie die Deklarationstabelle an.
6. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.

Lernaufgabe 2.5: 7-Segment-Anzeige

Lös. S. 149

Mit einer 7-Segment-Anzeige sind die Ziffern 0 .. 9 sowie die Zeichen a, b, c, d, e, und f darzustellen. Für jedes Zeichen müssen die entsprechenden Segmente a bis g angesteuert werden. Die Zeichen werden im 8-4-2-1-Code (BCD-Code) mit den Schaltern S3, S2, S1 und S0 eingestellt.



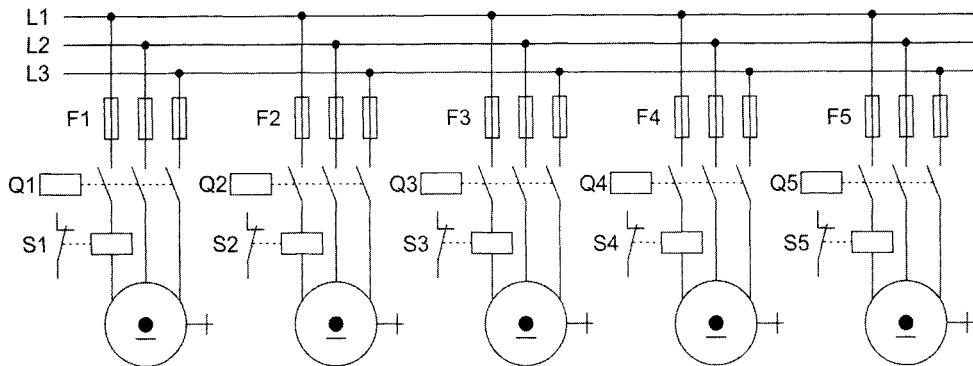
1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Bestimmen Sie mit einer Funktionstabelle den Zusammenhang zwischen den Eingängen S3 bis S0 und den Ausgängen a bis g.
3. Minimieren Sie die Ansteuerfunktionen der Ausgänge mit einem KVS-Diagramm.
4. Zeichnen Sie den Funktionsplan der minimierten Ansteuerfunktionen.
5. Das Steuerungsprogramm soll in einem bibliotheksfähigen Baustein realisiert werden. Bestimmen Sie die Bausteinart und geben Sie die Deklarationstabelle an.
6. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.

Lernaufgabe 2.6: Durchlauferhitzer

Lös. S. 154

In einem größeren Einfamilienhaus mit dezentraler Warmwasserversorgung sind fünf Durchlauferhitzer installiert. Wegen des hohen Anschlusswertes der Durchlauferhitzer erlaubt das Energieversorgungsunternehmen nur den gleichzeitigen Betrieb von zwei Durchlauferhitzern. Mit Lastabwurfrelais wird der Betriebszustand der Durchlauferhitzer erkannt.

Im Ausgangszustand (keine Wasserentnahme) sind die Freigabe-Schütze Q1...Q5 angezogen und die Lastabwurfrelais noch stromlos sowie ihre zugehörigen Kontakte S1...S5 geschlossen ($S = 1$). Das Einschalten eines Durchlauferhitzers wird über die Wasserentnahme gesteuert, die vom eingebauten Druckdifferenz-Schaltssystem erkannt wird. Das Lastabwurfrelais zieht an und öffnet seinen Kontakt ($S = 0$), der die Schaltlogik der Freigabe-Schütze beeinflusst.

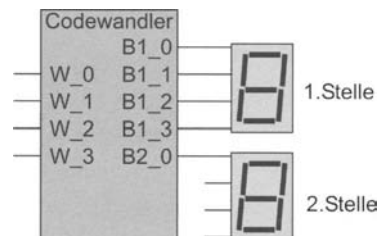
Schaltplan Hauptstromkreis:

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Bestimmen die Funktionstabelle für die Steuerungsaufgabe.
3. Minimieren Sie die Ansteuerfunktionen für einen Ausgang mit einem KVS-Diagramm. Schließen Sie aus dem Ergebnis auf die Ansteuerfunktionen der weiteren Ausgänge.
4. Zeichnen Sie den Funktionsplan der Ansteuerfunktion für einen Ausgang.
5. Das Steuerungsprogramm soll in einem bibliotheksfähigen Baustein realisiert werden. Bestimmen Sie die Bausteinart und geben Sie die Deklarationstabelle an.
6. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.

Lernaufgabe 2.7: Code-Wandler

Lös. S. 153

Es ist ein Code-Wandler zu entwerfen, mit dessen Hilfe der Zahlenwert einer Dualzahl an zwei BCD-codierten Ziffernanzeigen dargestellt werden kann. Der Wert der Dualzahl ist durch die vier Bits W_0 bis W_3 bestimmt.

Funktionsschema:

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Bestimmen Sie mit einer Funktionstabelle den Zusammenhang zwischen den Eingängen W_0 bis W_3 und den Ausgängen $B1_0$ bis $B2_0$ der Codewandlerfunktion.
3. Ermitteln Sie aus der Funktionstabelle die minimierten Schaltfunktionen und zeichnen Sie die dazugehörigen Funktionspläne. Hinweis: Überlegen Sie, ob Sie für die Ansteuerung einer Ausgangsvariablen die XOR-Funktion unter Minimierungsgesichtspunkten günstig einsetzen können.
4. Das Steuerungsprogramm soll in einem bibliotheksfähigen Baustein realisiert werden. Bestimmen Sie die Bausteinart und geben Sie die Deklarationstabelle an.
5. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.

2.3 Kontrollaufgaben

Kontrollaufgabe 2.1:

Gegeben ist eine Schaltfunktion $H1 = f(S3, S2, S1)$ durch die nebenstehende Funktionstabelle:

| Zeile | S3 | S2 | S1 | H1 |
|-------|----|----|----|----|
| 00 | 0 | 0 | 0 | 0 |
| 01 | 0 | 0 | 1 | 1 |
| 02 | 0 | 1 | 0 | 0 |
| 03 | 0 | 1 | 1 | 1 |
| 04 | 1 | 0 | 0 | 1 |
| 05 | 1 | 0 | 1 | 1 |
| 06 | 1 | 1 | 0 | 0 |
| 07 | 1 | 1 | 1 | 0 |

1. Bestimmen Sie aus der Funktionstabelle den schaltalgebraischen Ausdruck in der disjunktiven Normalform.
2. Zeichnen den Funktionsplan zu dem schaltalgebraischen Ausdruck.
3. Bestimmen Sie für die Funktionstabelle das KVS-Diagramm und minimieren Sie damit die Schaltfunktion.
4. Geben Sie für die minimierte Schaltfunktion die Anweisungsliste AWL an.

Kontrollaufgabe 2.2:

Die angegebene Anweisungsliste AWL ist zu analysieren.

1. Zeichnen Sie für die Anweisungsliste den zugehörigen Funktionsplan.
2. Bestimmen Sie mit einer Funktionstabelle den Zusammenhang zwischen den Eingangsvariablen und der Ausgangsvariable.
3. Ermitteln Sie aus der Funktionstabelle die disjunktive Normalform.
4. Geben Sie die Anweisungsliste der disjunktiven Normalform an.
5. Bestimmen Sie unter Verwendung der Exklusiv-ODER-Funktion einen einfacheren und gleichwertigen schaltalgebraischen Ausdruck.

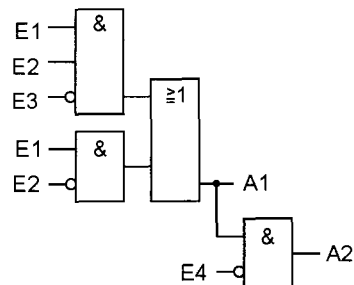
```

U (
  O E 0.1
  O E 0.2
)
U (
  ON E 0.1
  ON E 0.2
)
U E 0.3
= A 4.0

```

Kontrollaufgabe 2.3:

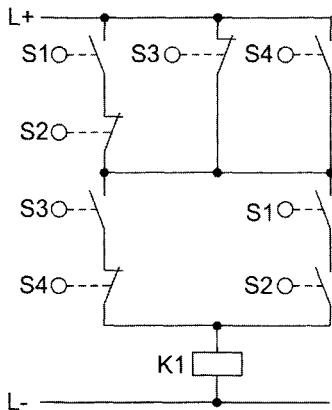
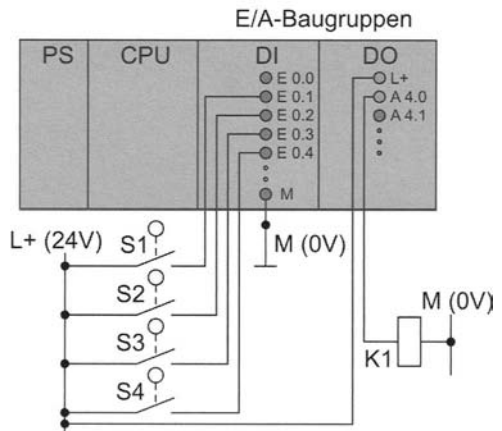
Der nebenstehende Funktionsplan zeigt die Abhängigkeit der beiden Ausgangsvariablen A1 und A2 von den Eingangsvariablen E1 bis E4.



1. Geben Sie für den Funktionsplan die zugehörige Anweisungsliste an.
2. Bestimmen Sie den Zusammenhang zwischen den Eingangsvariablen und der Ausgangsvariablen mit einer Funktionstabelle.
3. Ermitteln Sie mit Hilfe eines KVS-Diagramms einen einfacheren schaltalgebraischen Ausdruck für den Ausgang A1.
4. Geben Sie den Funktionsplan des schaltalgebraischen Ausdrucks an.
5. Der minimierte Ausdruck soll in einem Bibliotheksbaustein realisiert werden. Geben Sie die Deklarationstabelle und die Anweisungsliste an.

Kontrollaufgabe 2.4:

In einer verfahrenstechnischen Anlage wird ein Stellglied K1 durch vier Sensoren nach dem gegebenen Stromlaufplan angesteuert. Die Ansteuerung des Stellgliedes soll künftig mit einer SPS erfolgen, wobei die vier Sensoren an vier Eingänge des Automatisierungssystems angeschlossen werden.

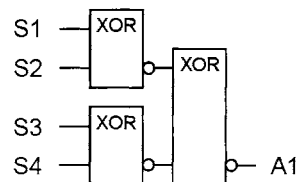
Schaltplan:**Verdrahtungsplan:**

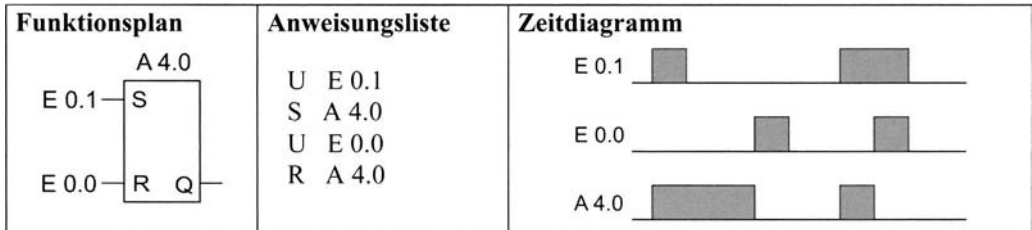
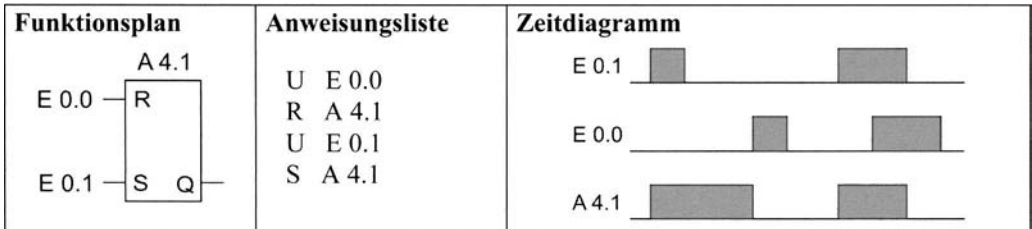
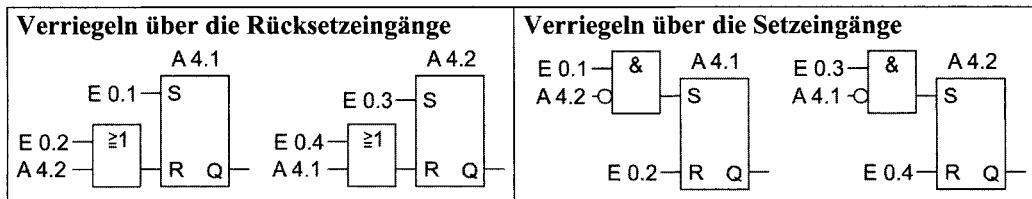
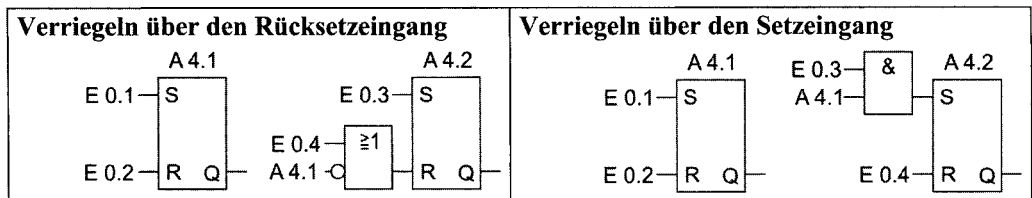
1. Bestimmen Sie die Zuordnungstabelle.
2. Bestimmen Sie aus dem Stromlaufplan einen Funktionsplan, der der logischen Struktur des Stromlaufplanes entspricht.
3. Ermitteln Sie aus dem Stromlaufplan oder Funktionsplan die Ansteuerfunktion für das Stellglied K1.
4. Bestimmen Sie die Funktionstabelle für den Ausgang K1.
5. Minimieren Sie die Ansteuerfunktion für den Ausgang K1 mit einem KVS-Diagramm.
6. Zeichnen Sie den Funktionsplan der minimierten Ansteuerfunktion.
7. Geben Sie die Deklarationstabelle für den Baustein an, mit dem Sie das Steuerungsprogramm realisieren und schreiben Sie die Anweisungsliste.

Kontrollaufgabe 2.5:

Der nebenstehende Funktionsplan ist zu analysieren.

1. Bestimmen Sie mit Hilfe einer Funktionstabelle den Zusammenhang zwischen den Eingängen S1 bis S4 und dem Ausgang A1.
2. Beschreiben Sie den Zusammenhang verbal.
3. Bestimmen Sie einen einfacheren schaltalgebraischen Ausdruck für den Ausgang A1.
4. Zeichnen Sie den Funktionsplan des einfachen schaltalgebraischen Ausdrucks.
5. Die gegebene Funktion soll in einem Bibliotheksbaustein realisiert werden. Geben Sie die Deklarationstabelle und die Anweisungsliste an.



RS-Speicherfunktionen
Speichern mit vorrangigem Rücksetzen

Speichern mit vorrangigem Setzen

Verriegelungen von Speichern
Gegenseitiges Verriegeln

Reihenfolgeverriegelung


Legende:

| | |
|---------------------|----------------|
| S = Setzeingang | Datentyp: Bool |
| R = Rücksetzeingang | Datentyp: Bool |
| Q = Ausgang | Datentyp: Bool |

Flankenauswertung

Zur Flankenauswertung eines Signals gehören ein Flankenoperand FO, der den veränderten Signalwert speichert und ein Impulsoperand IO, der bei Auftreten der Flanke für die Dauer eines Programmzyklus den Signalwert „1“ führt. Der Flankenoperand FO muss als lokale statische Variable deklariert sein und erfordert die Verwendung eines Funktionsbausteins FB, wenn nicht für FO ein Merkerbit oder Datenbit eines Globaldatenbausteins verwendet wird. Im OB1 kann für FO eine temporäre Variable verwendet werden.

Deklarationstabelle:

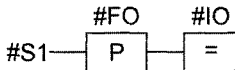
| Name | Datentyp | Anfangswert |
|------|----------|-------------|
| IN | | |
| S1 | Bool | FALSE |
| STAT | | |
| FO | Bool | FALSE |

| Name | Datentyp | Anfangswert |
|------|----------|-------------|
| TEMP | | |
| IO | Bool | |

Beispiele:

FUP

Positive (steigende) Flanke 0 → 1



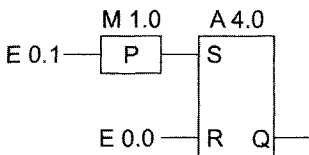
AWL

```
U #S1
FP #FO
= #IO
```

Negative (fallende) Flanke 1 → 0

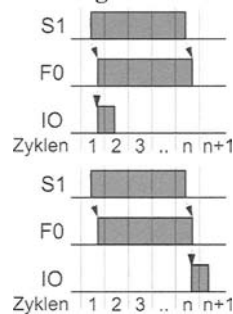


```
U #S1
FN #FO
= #IO
```



```
U E 0.1
FP M 1.0
S A 4.0
U E 0.0
R A 4.0
```

Zeitdiagramm



Auf den Impulsoperanden IO kann verzichtet werden, wenn die Flankenauswertung nur an einer Stelle des Programms benötigt wird. In diesem Fall wird der Ausgang einer Flankenauswertungsoperation direkt z.B. zum Setzen eines SR-Speichers verwendet.

Hilfsmittel für den Programmwurf

1. Schaltfolgetabelle für den Programmwurf mit mehreren S-, R-Bitoperationen

Für den Entwurf von Steuerungsprogramme mit schrittweisen Ablauf ist eine tabellarische Übersicht der Schaltbedingungen für das Setzen und Rücksetzen von Ausgangsvariablen mit S-, R-Bitoperationen hilfreich. Die Änderung der Eingangsvariablen, die zu einem Schrittwechsel führt, wird mit einer Flanke ausgewertet.

Beispiel:

| Schritt | Bedingung | Setzen | Rücksetzen |
|--------------|--|--------|------------|
| 1 | S1 (0 → 1) & 1S1 & 2S1 | 1Y1 | |
| 2 weitere | 1S2 (0 → 1) & 2S1 & 3S1 ↑ Flankenauswertung | 2Y1 | 1Y1 |

2. RS-Tabelle für den Entwurf von Programmen mit mehreren RS-Speichern

Bei Steuerungsprogrammen mit komplexen Speicherbedingungen in Form von Verriegelungen kann eine tabellarische Übersicht der vorzusehenden Setz- und Rücksetzbedingungen zur Klärung der Aufgabenstellung hilfreich sein. Die Anwendung einer RS-Tabelle erfolgt in zwei Schritten.

Schritt 1: Ermittlung der Anzahl notwendiger RS-Speicherglieder auf Grund der unterscheidbaren Steuerungszustände, eventuell gegliedert in Speicher Y und Hilfsspeicher HS.

Beispiel:

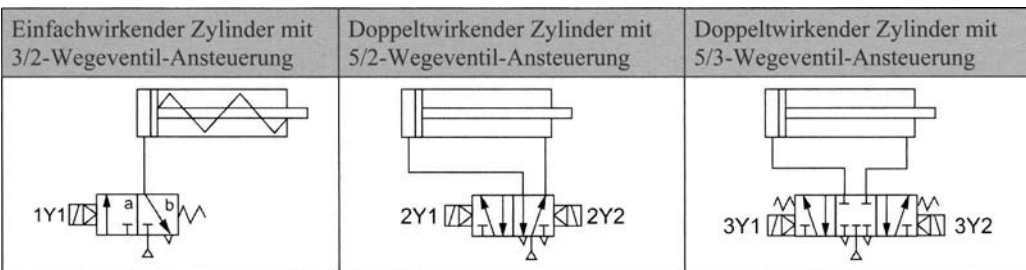
| Ermittlung der Speicher Y und Hilfsspeicher HS | Ermittlung der Variablen für das Setzen | Ermittlung der Variablen für das Rücksetzen |
|--|---|---|
| Y1 für Anlagenteil I | HS1, ... | S1, Y2, ... |
| HS1 für Anlagenteil I | S2, ... | Y1, HS2, ... |

Schritt 2: Eintragen der Setzbedingungen und Rücksetzbedingungen für alle Speicher Y und Hilfsspeicher HS.

Beispiel:

| Zu betätigende Speicherglieder | Bedingungen für das Setzen | Bedingungen für das Rücksetzen |
|--------------------------------|----------------------------|--------------------------------|
| HS1 | S2 | Y1 ∨ HS2 |
| Y1 | HS1 | S1 ∨ Y2 |

3. Darstellung und Eigenschaften elektropneumatischer Stellglieder



3/2-Wegeventil: Elektromagnetisches Ventil mit Rückstellfeder hat nur einen elektrischen Steuereingang Y1 und kann durch ein Stellsignal aus der Schaltstellung b in die Schaltstellung a geschaltet werden. Nach Beendigung des Stellsignals erfolgt eine federmechanische Rückstellung des Ventils, dessen Vorteil die definierte Schaltstellung im unbetätigten Zustand ist.

5/2-Wegeventil: Elektromagnetische Impulsventile haben zwei elektrische Steuereingänge Y1 und Y2, sie können durch kurze Ansteuerimpulse aus einer Schaltstellung in die andere umgeschaltet werden. Die Ventile übernehmen die RS-Speicherfunktion der Steuerung. Nachteil des Speicherverhaltens ist die nicht definierte Schaltstellung im unbetätigten Zustand, daher erfolgt meistens Ansteuerung von Y1 und Y2 mit inversen Signalen.

5/3-Wegeventil: Bei elektromagnetischen Impulsventilen mit Federzentrierung geht das Ventil im unbetätigten Zustand in die Mittelstellung. Daher kann neben der Vorwärts- und Rückwärtsbewegung des Zylinderkolbens auch eine Halteposition veranlasst werden.

Funktionsdiagramme zeigen den Bewegungsverlauf von Zylindern. In der Ordinate wird der zurückgelegte Weg und in der Abszisse werden Schritte oder Zeiten aufgetragen. Zusätzlich können die Zustände von Magnetspulen der Ventile dargestellt werden.

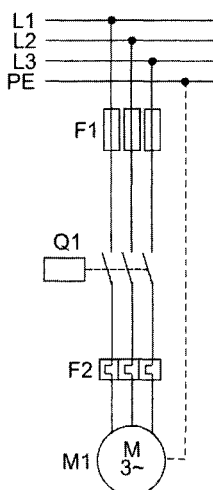
4. Regeln für das Umsetzen von Schützsaltungen in SPS-Programme

Eine gegebene Schützsteuerung, die z.B. zur Ansteuerung von Elektromotoren oder Elektropneumatik eingesetzt wurde, kann durch eine SPS-Steuerung unter sinngemäßer Anwendung der nachfolgenden Umsetzungsregeln für Stromlaufpläne ersetzt werden.

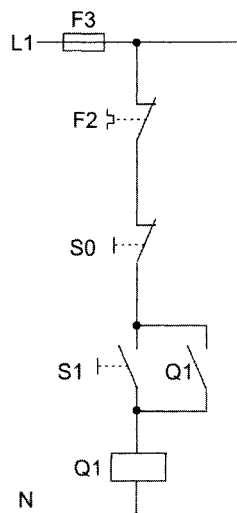
1. Der Hauptstromkreis wird unverändert außerhalb der SPS beibehalten.
2. Hauptschütze werden von den SPS-Ausgängen angesteuert und werden als OUT- oder IN-OUT-Variable deklariert. Der Schaltzustand des Hauptschützes wird somit außerhalb des Bausteins in einem SPS-Ausgang gespeichert. Wird ein Kontakt Q1 des Hauptschützes Q1 am Bausteineingang abgefragt, ist Q1 als Durchgangvariable (IN_OUT) zu deklarieren.
3. Hilfsschütze werden durch temporäre oder statische Variablen innerhalb des Bausteins ersetzt. Wenn Hilfsschütze interne Speicherfunktionen ausüben, muss für jedes Hilfsschütz eine statische Speichervariable (STAT) deklariert werden.
4. Parallelschaltungen von Schützkontakten werden durch ODER-Verknüpfungen und Reihenschaltungen durch UND-Verknüpfungen der entsprechenden Variablen ersetzt.
5. Öffner von Schützkontakten werden negiert und Schließer bejaht im Programm abgefragt.
6. Öffner- und Schließerkontakte von Signalgebern wie Taster, Schalter, Kontakte von Übersstromschutzeinrichtungen etc. werden im Programm stets bejaht abgefragt und als Eingangsvariablen deklariert, wenn derselbe Kontaktyp beibehalten wird.
7. Die bejahte Abfrage von Signalgebern in Regel 6 gilt nicht bei Verwendung der Speicherfunktion anstelle der Selbsthaltung.
8. Die Umsetzungsregeln 1 bis 6 verändern nicht die vorgegebene Steuerungsstruktur, wenn die Schütze keine Zeitverzögerungen oder Wischerkontakte enthalten. Einschalt- und Ausschaltverzögerungen müssen mit Zeitgliedern und Wischerkontakten mit Flankenauswertung nachgebildet werden. Impulse von Wischerkontakten beim Einschalten (Ausschalten) entsprechen steigenden (fallenden) Flanken.

Beispiel:

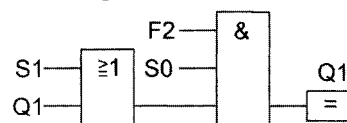
Hauptstromkreis bleibt erhalten



Steuerstromkreis wird ersetzt

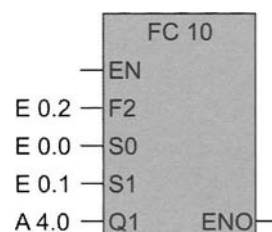


Ersatz-Funktionsplan:



Aufruf im OB1:

Die Signalspeicherung für das Hauptschütz Q1 erfolgt im SPS-Ausgang A 4.0, daher genügt der Bausteintyp FC. Q1 ist als Durchgangvariable (IN_OUT) deklariert, auf die SPS-Ausgangsadresse A 4.0 kann lesend und schreibend zugegriffen werden.



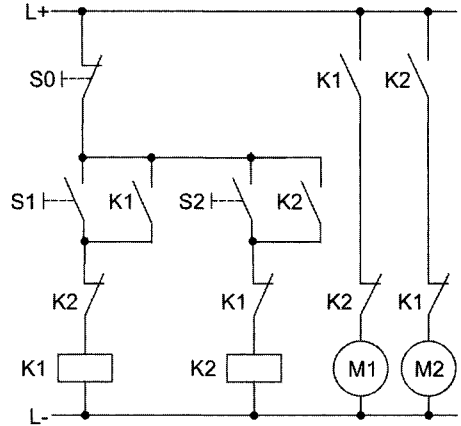
| | |
|------------|--|
| 3.1 | <p>Beispiel</p> <ul style="list-style-type: none"> • Stromlaufplan in SPS-Programm umsetzen • Verriegeln von RS-Speichern, RS-Tabelle |
|------------|--|

Schützschtaltung mit gegenseitiger Verriegelung

Mit einer sicherheitsgerichteten Schaltung können die Kleinspannungs-Motoren M1 und M2 durch den EIN-Taster S1 und S2 wahlweise einzeln eingeschaltet werden und bleiben nach Loslassen der Taster an. Ein Umschalten zwischen M1 und M2 ist nur nach Betätigen des AUS-Taster S0 möglich.

1. Der gegebene Stromlaufplan ist ohne Änderung der Steuerungsstruktur in ein SPS-Programm umzusetzen.
2. Es ist ein funktionsgleiches Programm mit SR-Speichern und Verriegelungen an den Rücksetzeingänge zu suchen.
3. Die Motoren M1, M2 sollen sich nur in abwechselnder Reihenfolge einschalten lassen. Programmentwurf über RS-Tabelle.

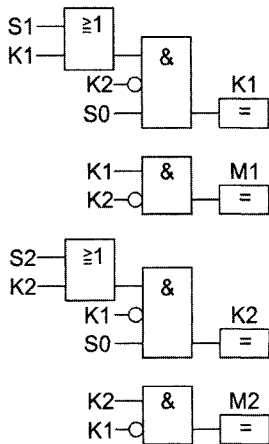
Stromlaufplan:



Zuordnungstabelle der Eingänge und Ausgänge:

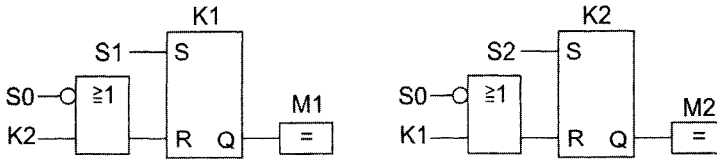
| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | | Adresse |
|------------------|--------|----------|--------------------|--------|---------|
| AUS-Taster | S0 | BOOL | Betätigt | S0 = 0 | E 0.0 |
| EIN-Taster 1 | S1 | BOOL | Betätigt | S1 = 1 | E 0.1 |
| EIN-Taster 2 | S2 | BOOL | Betätigt | S2 = 1 | E 0.2 |
| Ausgangsvariable | | | | | |
| Motor 1 | M1 | BOOL | Motor 1 läuft | M1 = 1 | A 4.0 |
| Motor 2 | M2 | BOOL | Motor 2 läuft | M2 = 1 | A 4.1 |

1. Direktes Umsetzen des Stromlaufplans in ein SPS-Programm: FB10

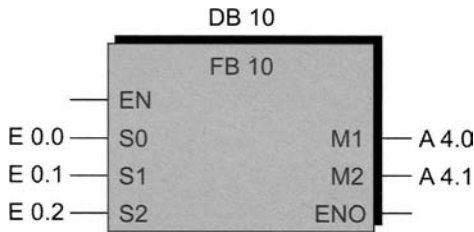


| Name | Datentyp | Anfangswert |
|------|----------|-------------|
| IN | | |
| S0 | Bool | FALSE |
| S1 | Bool | FALSE |
| S2 | Bool | FALSE |
| OUT | | |
| M1 | Bool | FALSE |
| M2 | Bool | FALSE |
| STAT | | |
| K1 | Bool | FALSE |
| K2 | Bool | FALSE |

2. Funktionsgleiche Lösung mit RS-Speicher:

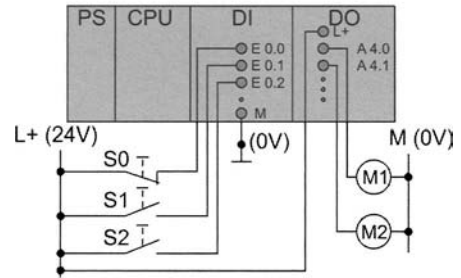


Aufruf im OB1 (für beide Lösungen)



(FB10 mit gleicher Deklarationstabelle)

Anschlussschema

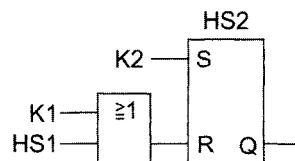
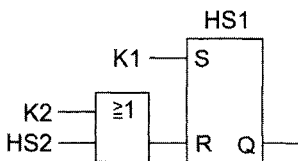
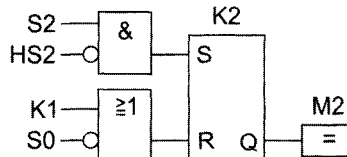
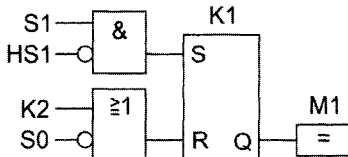


3. RS-Tabelle für das Schalten in wechselnder Reihenfolge

Es sind zwei zusätzliche Hilfsspeicher HS1, HS2 erforderlich, um sich merken zu können, welcher der beiden K-Speicher vor dem Löschen gesetzt war. K1, K2, HS1, HS2 sind als statische Variablen zu deklarieren und somit ist ein FB-Baustein erforderlich. Beim Einschalten der Steuerung zufällig gesetzte Speicher müssen automatisch zurückgesetzt werden.

| Zu betätigende Speicherglieder | Bedingungen für das Setzen | Bedingungen für das Rücksetzen |
|--------------------------------|----------------------------|--------------------------------|
| K1 | $S1 \wedge \overline{HS1}$ | $\overline{S0} \vee K2$ |
| HS1 | K1 | $K2 \vee HS2$ |
| K2 | $S2 \wedge \overline{HS2}$ | $\overline{S0} \vee K1$ |
| HS2 | K2 | $K1 \vee HS1$ |

Funktionsplan: FB20



Aufruf im OB1 (AWL)

```

U E 0.0
= L 20.0
U E 0.1
= L 20.1
U E 0.2
= L 20.2
CALL FB10, DB10
S0 :=L 20.0
S1 :=L 20.1
S2 :=L 20.2
M1:=A 5.0
M2:=A 5.1
    
```

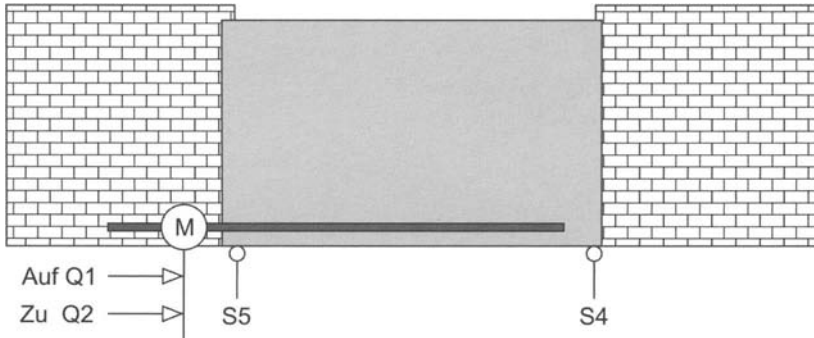

3.2 Lernaufgaben

Lernaufgabe 3.1: Torsteuerung

Lös. S. 156

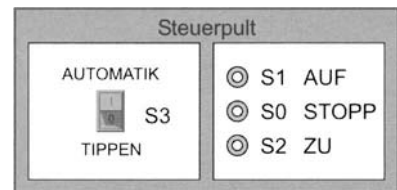
Ein Werkstor wird mit einem Elektromotor auf und zu gesteuert. Die Ansteuerung des Elektromotors erfolgt mit den Leistungsschützen Q1 (Rechtslauf Tor auf) und Q2 (Linkslauf Tor zu). Die Endlagen des Schiebetors werden mit den Initiatoren S4 (Tor zu) und S5 (Tor auf) gemeldet.

Technologieschema:



Zur Bedienung des Schiebetors ist an der Pforte ein Bedienpult angebracht. Steht der Wahlschalter S3 in Stellung Automatik, kann durch kurzzeitiges Drücken des S1- bzw. S2-Tasters das Tor auf- bzw. zugesteuert werden. Ein Umschalten der Drehrichtung des Motors ist nur über S0 (STOPP) möglich.

Bei Betätigung des Tasters S0 bleibt das Tor sofort stehen. Zum Öffnen oder Schließen ist dann eine erneute Betätigung der Taster S1 oder S2 erforderlich. Steht der Wahlschalter S3 in Stellung Tippen, so wird das Werkstor mit den Tastern S1 und S2 nur solange geöffnet bzw. geschlossen, wie die entsprechende Taste betätigt wird.



Die Initiatoren S4 und S5 beenden bei Betätigung jeweils sofort das Öffnen bzw. Schließen des Tores.

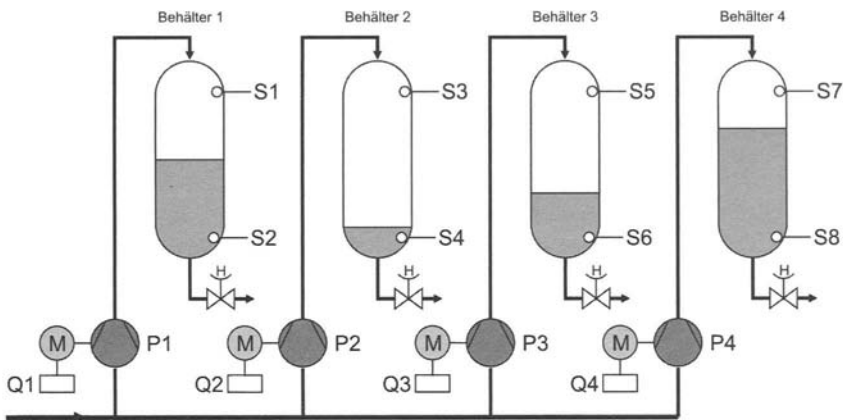
1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge. Legen Sie dabei die richtige Zuordnung der Signalgeber (Öffner oder Schließer) fest.
2. Bestimmen Sie mit einer RS-Tabelle die Bedingungen für die Ansteuerung der Setz- und Rücksetz-Eingänge der beiden Speicherglieder für Q1 und Q2.
3. Ermitteln Sie aus der RS-Tabelle den Funktionsplan für die beiden Ausgänge Q1 und Q2.
4. Da das Steuerungsprogramm zur Ansteuerung weiterer Schiebetore auf dem Werksgelände genutzt werden soll, ist dafür ein bibliotheksfähiger Baustein zu verwenden. Überlegen Sie, ob Sie dafür eine Funktion verwenden können.
5. Geben Sie die Deklarationstabelle für den Baustein an, mit dem Sie das Steuerungsprogramm realisieren.
6. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.

Lernaufgabe 3.2: Vier Vorratsbehälter**Lös. S. 157**

Vier Vorratsbehälter, die von Hand entleert werden können, werden mit Pumpen aus einem gemeinsamen Versorgungsnetz gefüllt. Jeder Behälter hat einen Signalgeber für die Vollmeldung und für die Leermeldung. Die Motoren der Pumpen haben unterschiedliche Anschlussleistungen:

$P1 = 3 \text{ kW}$; $P2 = 4 \text{ kW}$, $P3 = 7 \text{ kW}$ und $P4 = 5 \text{ kW}$.

Eine Steuerschaltung soll bewirken, dass bei Leermeldung eines Behälters dieser wieder gefüllt wird, wobei jedoch eine Gesamtleistung der laufenden Pumpen von 10 kW nicht überschritten werden darf.

Technologieschema:

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Stellen Sie eine Tabelle auf, aus der ersichtlich wird, welche Pumpen gleichzeitig laufen dürfen und welche Pumpe dann nicht zugeschaltet werden darf.
3. Bestimmen Sie mit einer RS-Tabelle unter zu Hilfenahme der in Punkt 2 aufgestellten Tabelle die Bedingungen für die Ansteuerung der Setz- und Rücksetz-Eingänge der vier Speicherglieder für Pumpen.
4. Ermitteln Sie aus der RS-Tabelle den Funktionsplan für die Ansteuerung der Motorschütze Q1 bis Q4.
5. Das Steuerungsprogramm soll in einem bibliotheksfähigen Baustein realisiert werden. Überlegen Sie, ob Sie dafür eine Funktion verwenden können.
6. Geben Sie die Deklarationstabelle für den Baustein an, mit dem Sie das Steuerungsprogramm realisieren.
7. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.

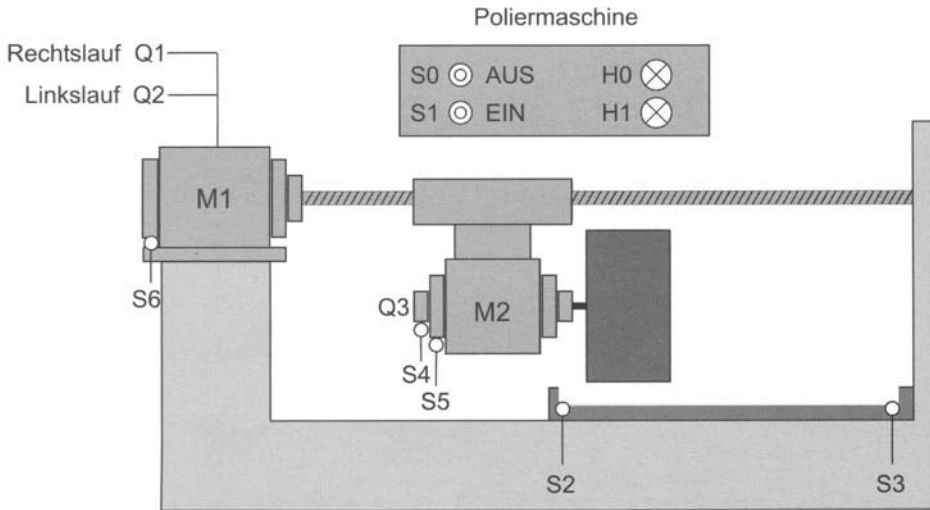
Lernaufgabe 3.3: Poliermaschine**Lös. S. 159**

Eine Poliermaschine wird durch Betätigung des Tasters S1 eingeschaltet. Der Betriebszustand „EIN“ wird mit der Meldeleuchte H1 angezeigt. In diesem Betriebszustand sind der Antriebsmotor M2 für die Poliereinrichtung und der Schlittenantrieb M1 in Betrieb. Der Schlittenantrieb M1 läuft aber nur, wenn der Drehgeber S4 meldet, dass der Motor M2 der Poliereinrichtung

tung die erforderliche Drehzahl hat. Beim Erreichen der linken Endposition S2 bzw. der rechten Endposition S3 wird die Drehrichtung des Schlittenmotors umgeschaltet.

Mit Taster S0 wird die Poliermaschine abgeschaltet. Der Schlittenantrieb und der Antriebsmotor für die Poliereinrichtung laufen jedoch noch bis zum Erreichen einer der beiden Endpositionen weiter. In dieser Zeit wird die Meldeleuchte H0 eingeschaltet. Das Auftreten einer thermischen Auslösung S5 oder S6 bei einem der beiden Motoren führt sofort zum ausgeschalteten Betriebszustand.

Technologieschema:



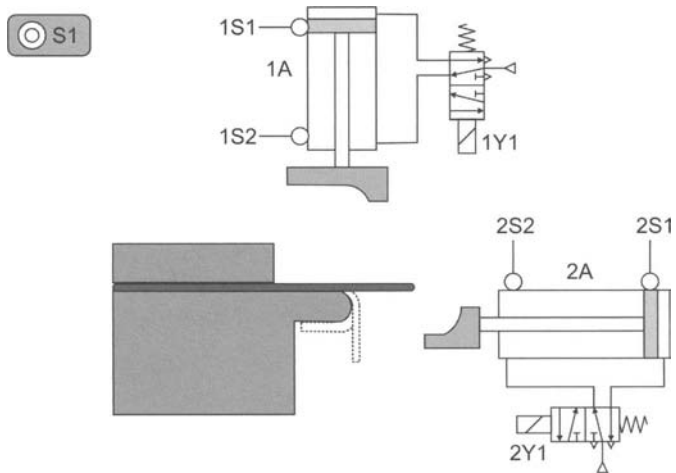
1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge. Legen Sie dabei die richtige Zuordnung der Signalgeber (Öffner oder Schließer) fest.
2. Ermitteln Sie mit einer RS-Tabelle die Bedingungen für die Ansteuerung der Setz- und Rücksetz-Eingänge der Speicherglieder für K1' und K2 sowie H0 und H1.
3. Ermitteln Sie aus der RS-Tabelle den Funktionsplan für die Schütze Q1, Q2 und Q3 sowie die Anzeigeleuchten H0 und H1.
4. Der gefundene Funktionsplan ist in einen bibliotheksfähigen Baustein umzusetzen. Überlegen Sie, welche Bausteinart Sie für die Umsetzung verwenden können.
5. Geben Sie die Deklarationstabelle für den Baustein an, mit dem Sie das Steuerungsprogramm realisieren.
6. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.

Lernaufgabe 3.4: Biegewerkzeug

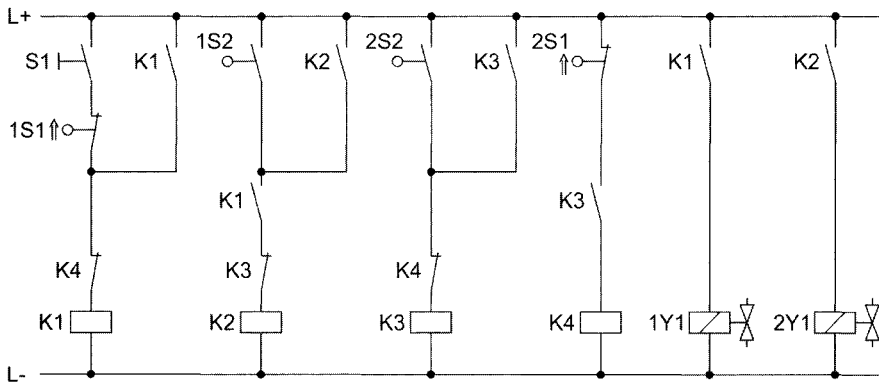
Lös. S. 160

Mit einem Biegewerkzeug werden Bleche in eine bestimmte Form gebogen. Der Biegevorgang läuft wie folgt ab: Nach Betätigung von Taster S1 fährt Zylinder 1A aus. Dabei wird das Blech gespannt und vorgebogen. Ist die vordere Endlage des Zylinders 1A erreicht, fährt Zylinder 2A aus, biegt dabei das Blech fertig und fährt nach Erreichen der Endlage wieder ein. Ist Zylinder 2A in der hinteren Endlage angekommen, fährt auch Zylinder 1A wieder ein. Das folgende Technologieschema zeigt die Anordnung der Zylinder.

Technologieschema:



Die Ansteuerung der Elektromagnetventile erfolgte bisher durch folgende Schützsteuerung:



Das Biegewerkzeug soll in eine komplexe Automatisierungsanlage integriert werden und deshalb die Ansteuerung der Ventile über ein Automatisierungssystem erfolgen. Dazu ist die Schützsteuerung durch ein SPS-Programm zu ersetzen.

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Zeichnen Sie das zur Aufgabenbeschreibung zugehörige Funktionsdiagramm, welches die beiden Zylinder 1A und 2A sowie die Schütze K1 bis K4 enthält.
3. Ermitteln Sie aus dem gegebenen Stromlaufplan nach den Umsetzungsregeln zum Ersetzen von Schützschaltungen den zugehörigen Funktionsplan.
4. Welche lokalen Variablen müssen Sie bei der Realisierung des Funktionsplanes in einem wieder verwendbaren Baustein für die Schütze K1 bis K4 deklarieren und welche Bausteinart folgt daraus? Geben Sie die Deklarationstabelle an.
5. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.
6. Es soll ein neuer Funktionsplan unter Verwendung von drei RS-Speichergliedern entworfen werden. Zwei RS-Speicherglieder dienen zur Ansteuerung der Magnetspulen 1Y1 und 2Y1. Das dritte Speicherglied wird als Hilfsvariable HV1 benötigt. Bestimmen Sie mit ei-

ner RS-Tabelle die Bedingungen für die Ansteuerung der Setz- und Rücksetz-Eingänge der drei RS-Speicherglieder.

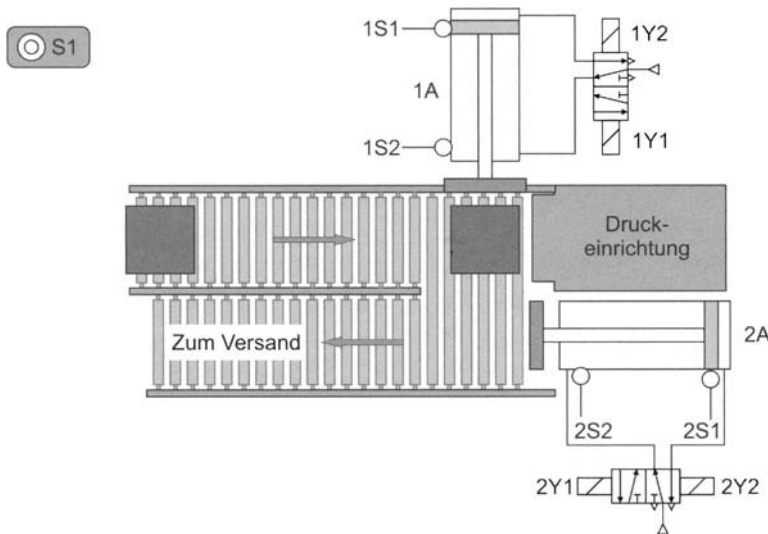
7. Ermitteln Sie aus der RS-Tabelle den Funktionsplan für die Ausgänge 1Y1 und 2Y1 sowie für die Hilfsvariable HV1.
8. Geben Sie die Deklarationstabelle für den Baustein an, mit dem Sie das Steuerungsprogramm realisieren.
9. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.
10. Es soll ein neuer Funktionsplan für das Biegewerkzeug mit Hilfe der Schaltfolgetabelle bestimmt werden. Erstellen Sie die Schaltfolgetabelle.
11. Bestimmen Sie den sich aus der Schaltfolgetabelle ergebenden Funktionsplan.
12. Das Steuerungsprogramm soll in einem Baustein bibliotheksfähig realisiert werden. Bestimmen Sie die Bausteinart und geben Sie die Deklarationstabelle für den Baustein an.
13. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.

Lernaufgabe 3.5: Verpackungsrollenbahn

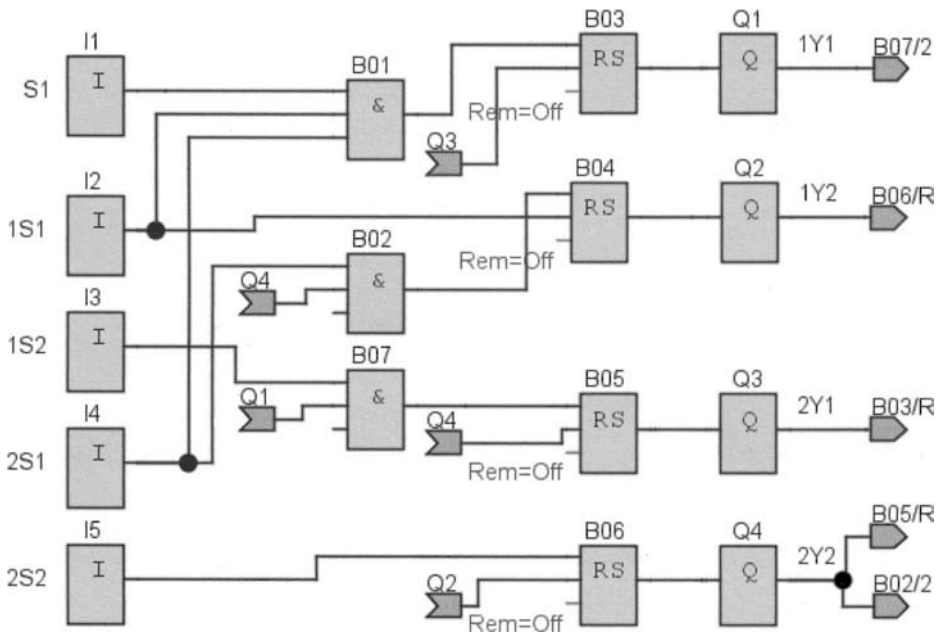
Lös. S. 164

Am Ende einer Verpackungsrollenbahn erhalten die Verpackungen einen Aufdruck. Die Pakete rollen über die Rollenrutsche vor die Druckeinrichtung. Nach Betätigung des Tasters S1 schiebt Zylinder 1A das Paket auf die entgegengesetzte Rollenbahn. Zylinder 2A schiebt dann das Paket auf eine Rollenrutsche zum Versand. Der Druckvorgang ist nicht Gegenstand der Aufgabe.

Technologieschema:



Die Ansteuerung der Magnetspulen der beiden 5/2-Wegeventile erfolgte bisher durch eine LOGO-Kleinsteuerung. Der nachfolgende Funktionsplan gibt das Steuerungsprogramm wieder.

Logo-Funktionsplan:

Die bisherige steuerungstechnische Inselfösung soll in ein komplexes Automatisierungssystem integriert werden. Dazu ist das LOGO-Steuerungsprogramm in ein SPS-Programm umzuwandeln.

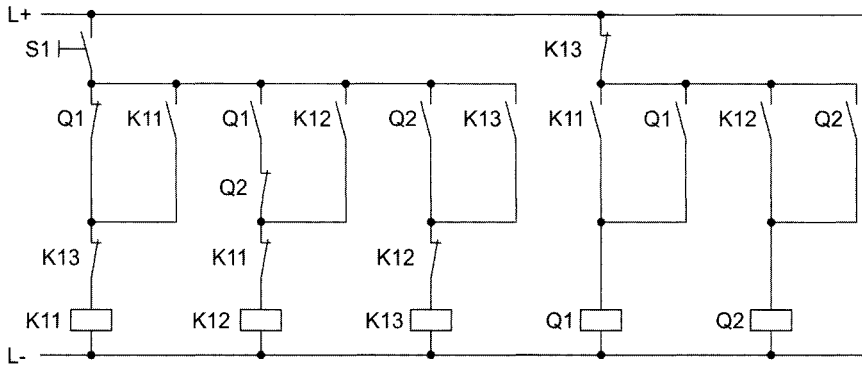
1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Entnehmen Sie dem LOGO-Programm die Bedingungen für die Ansteuerung von Setz- und Rücksetz-Eingang der vier RS-Speicherglieder für den Eintrag in eine RS-Tabelle.
3. Ermitteln Sie aus der RS-Tabelle den Funktionsplan für das SPS-Programm.
4. Bestimmen Sie aus dem Funktionsplan das Funktionsdiagramm, welches die beiden Zylinder 1A bzw. 2A und die vier Magnetspulen 1Y1, 1Y2, 2Y1 und 2Y2 enthält.
5. Das Steuerungsprogramm soll in einem Baustein bibliotheksfähig realisiert werden. Überlegen Sie, welche Art von Baustein Sie für das Steuerungsprogramm verwenden und geben Sie die Deklarationstabelle für den Baustein an.
6. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.

Lernaufgabe 3.6: Belüftungsanlage

Lös. S. 166

Zwei Ventilatoren einer Belüftungsanlage werden über einen Taster S1 wie folgt gesteuert: Nach der ersten Betätigung des Tasters S1 wird Ventilator 1 über das Schütz K1 eingeschaltet. Bei der nächsten Betätigung von S1 wird Ventilator 2 über das Schütz K2 dazu geschaltet. Die dritte Betätigung führt zur Abschaltung der beiden Ventilatoren. Bisher wurde die Steuerungsaufgabe der Belüftungsanlage von einer Schützsteuerung übernommen.

Stromlaufplan der Steuerung:



Die Ansteuerung der beiden Leistungsschütze Q1 und Q2 soll künftig über eine SPS erfolgen.

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Ermitteln Sie aus dem gegebenen Stromlaufplan nach den Umsetzungsregeln zum Ersetzen von Schützsaltungen den zugehörigen Funktionsplan.
3. Welche lokalen Variablen müssen Sie bei der Realisierung des Funktionsplanes in einem wieder verwendbaren Baustein für die Schütze K11 bis K13 deklarieren und welche Bausteinart folgt daraus? Geben Sie die Deklarationstabelle an.
4. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.
5. Entwerfen Sie einen neuen Funktionsplan unter Verwendung der Flankenauswertung sowie Setz- und Rücksetzfunktionen für das SPS-Programm. Dabei kann die im Lehrbuch vorgestellte Methode des Binäruntersetzers verwendet werden.
6. Bestimmen Sie die Bausteinart zur Realisierung des neuen Funktionsplanes in einem wieder verwendbaren Baustein und geben Sie die Deklarationstabelle für den Baustein an.
7. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.

Lernaufgabe 3.7: Reklamebeleuchtung

Lös. S. 167

Drei Beleuchtungskörper H1, H2 und H3 einer Reklametafel sollen jeweils bei Betätigung des Tasters S1 nach nebenstehendem Muster ein- bzw. ausgeschaltet werden:

| | Beleuchtungsgruppe H1 | Beleuchtungsgruppe H2 | Beleuchtungsgruppe H3 |
|----------------|-----------------------|-----------------------|-----------------------|
| Reklametafel | | | |
| 1. Betätigung: | | | |
| 2. Betätigung: | | | |
| 3. Betätigung: | | | |
| 4. Betätigung: | | | |
| 5. Betätigung: | | | |
| 6. Betätigung: | | | |

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.

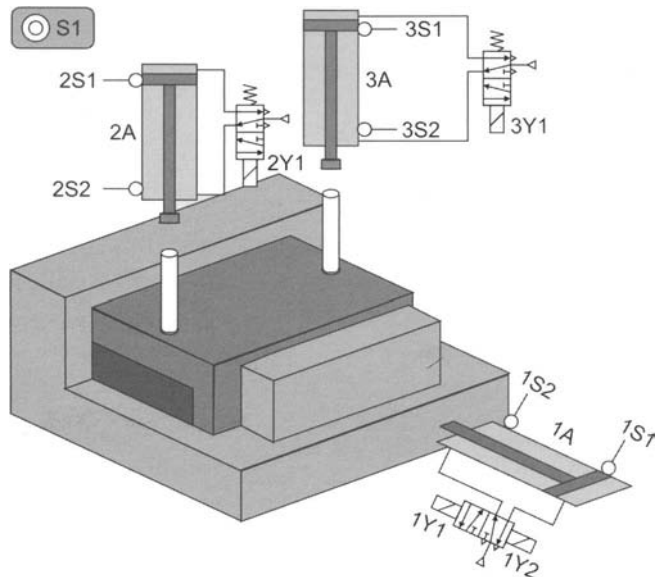
2. Zeichnen Sie den Funktionsplan für das SPS-Programm. Dabei kann die im Lehrbuch vorgestellte Methode des Binäruntersetzers verwendet werden. Es ist allerdings ein Hilfsspeicher HV1 einzuführen, der mit der letzten einzuschaltenden Beleuchtungsgruppe gesetzt und mit der letzten auszuschaltenden Beleuchtungsgruppe rückgesetzt wird.
3. Das Steuerungsprogramm soll in einem Baustein bibliotheksfähig realisiert werden. Bestimmen Sie die Bausteinart und geben Sie die Deklarationstabelle für den Baustein an.
4. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.

Lernaufgabe 3.8: Verstift-Einrichtung

Lös. S. 168

In einer Verstift-Einrichtung sollen zwei Werkstücke durch Verstiften in ihrer Lage gesichert werden. Nach Betätigung des Tasters S1 spannt Zylinder 1A die Werkstücke. Danach fährt Zylinder 2A aus, presst den Zylinderstift ein und fährt sofort wieder zurück. Befindet sich der Zylinder 2A wieder in seiner oberen Endlage, erfolgt der gleiche Einpressvorgang mit Zylinder 3A. Der Spannzylinder 1A fährt ein, wenn sich Zylinder 3A wieder in der hinteren Endlage befindet.

Technologieschema:



Die Ansteuerung der Elektromagnetventile soll durch eine SPS erfolgen. Dazu ist in einem bibliotheksfähigen Baustein ein Steuerungsprogramm zu entwerfen, welches auf der Auswertung der Schaltfolgetabelle basiert.

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Zeichnen Sie das zur Aufgabenbeschreibung zugehörige Funktionsdiagramm, welches die Zylinder 1A, 2A und 3A enthält.
3. Stellen Sie die Schaltfolgetabelle auf.
4. Zeichnen Sie den sich aus der Schaltfolgetabelle ergebenden Funktionsplan.
5. Das Steuerungsprogramm soll in einem Baustein bibliotheksfähig realisiert werden. Bestimmen Sie für das Steuerungsprogramm den erforderlichen Bausteintyp und geben

Sie die Deklarationstabelle für den Baustein an.

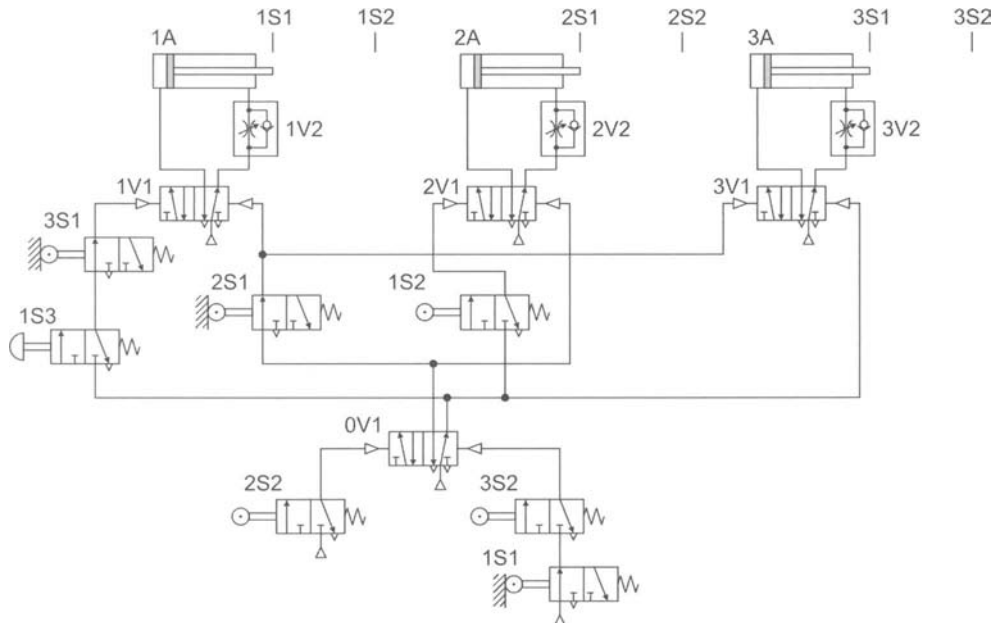
6. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.

Lernaufgabe 3.9: Drei Zylinder-Steuerung

Lös. S. 170

Der gegebene Pneumatikschaltplan zeigt die Ansteuerung von drei Zylindern mittels pneumatischer Ventile.

Pneumatikschaltplan:



Die Ansteuerung der drei Zylinder soll künftig mit einer SPS über 5/2-Wegeventile mit beidseitig elektromagnetischer Betätigung erfolgen. Die pneumatischen Endlagegeber werden durch elektrische an den Zylindern angebrachten Sensoren ersetzt.

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Zeichnen Sie das zum pneumatischen Schaltplan zugehörige Funktionsdiagramm, welches die drei Zylinder 1A, 2A und 3A enthält.
3. Zeichnen Sie den neuen pneumatischen Schaltplan.
4. Bestimmen Sie mit einer Schaltfolgetabelle die Bedingungen für das Setzen und das Rücksetzen der Magnetventile.
5. Zeichnen Sie den sich aus der Schaltfolgetabelle ergebenden Funktionsplan.
6. Das Steuerungsprogramm soll in einem Baustein bibliotheksfähig realisiert werden. Bestimmen Sie den erforderlichen Bausteintyp für das Steuerungsprogramm und geben Sie die Deklarationstabelle für den Baustein an.
7. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.

3.3 Kontrollaufgaben

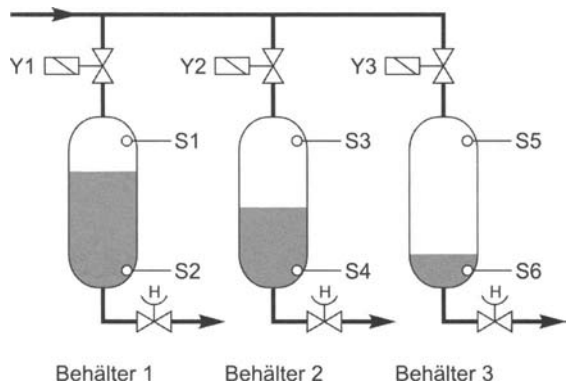
Kontrollaufgabe 3.1

Ein elektrisch angetriebenes Tor wird durch die Taster S1 (AUF), S2 (ZU) und S0 (STOPP) gesteuert. Die Endschalter S3 und S4 melden, ob das Tor geöffnet oder geschlossen ist. Der Antriebsmotor ist mit einem Motorschutzrelais zu schützen. Die Umschaltung der Drehrichtung kann nur über S0 (STOPP) erfolgen.

1. Zeichnen Sie den Hauptstromkreis der Torsteuerung.
2. Die Ansteuerung der beiden Leistungsschütze Q1 und Q2 soll durch eine SPS erfolgen. Zeichnen Sie den Anschlussplan der SPS.
3. Bestimmen Sie mit einer RS-Tabelle die Bedingungen für die Ansteuerung von Setz- und Rücksetz-Eingang der beiden Ausgänge Q1 und Q2.
4. Ermitteln Sie aus der RS-Tabelle den Funktionsplan zur Ansteuerung der Ausgänge Q1 und Q2.
5. Der gefundene Funktionsplan soll mit einem bibliotheksfähigen Baustein realisiert werden. Geben Sie die Deklarationstabelle für den Baustein an.
6. Bestimmen Sie die Anweisungsliste AWL des Bausteins.

Kontrollaufgabe 3.2

Drei Vorratsbehälter mit den Signalgebern S1, S3 und S5 für die Vollmeldung und S2, S4 und S6 für die Leermeldung können von Hand in beliebiger Reihenfolge entleert werden. Eine Steuerung soll bewirken, dass stets nur zwei Behälter gleichzeitig nach erfolgter Leermeldung gefüllt werden können. Das Füllen eines Behälters dauert solange an, bis die entsprechende Vollmeldung erfüllt ist.



1. Geben Sie die Zuordnungstabelle mit allen Eingängen und Ausgängen an.
2. Bestimmen Sie mit einer RS-Tabelle die Bedingungen für die Ansteuerung der Setz- und Rücksetz-Eingänge der Speicherglieder Y1, Y2 und Y3.
3. Ermitteln Sie aus der RS-Tabelle den Funktionsplan für die Ausgänge Y1, Y2 und Y3.
4. Die Steuerungsfunktion soll mit einem Baustein bibliotheksfähig realisiert werden. Geben Sie die Deklarationstabelle des Bausteins an.
5. Schreiben Sie die Anweisungsliste AWL des Steuerungsprogramms.

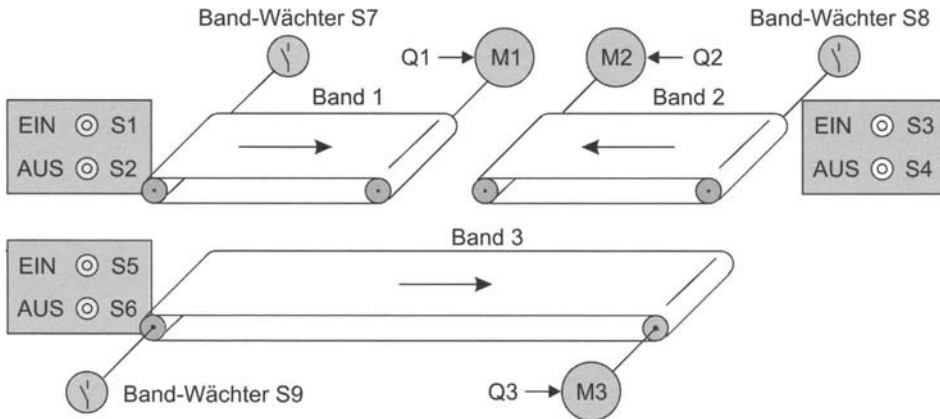
Kontrollaufgabe 3.3

Mit dem Förderband 1 und dem Förderband 2 werden unterschiedliche Rohmaterialien dem Förderband 3 zugeführt. Die Förderbänder 1 und 2 dürfen niemals gleichzeitig laufen. Außerdem dürfen die beiden Bänder nur fördern, wenn das Förderband 3 eingeschaltet ist.

Ein Abschalten von Band 3 ist nur möglich, wenn zuvor Band 1 oder Band 2 abgeschaltet worden ist.

Alle Bänder haben jeweils einen EIN- und einen AUS-Taster. Der Bandlauf wird an allen Bändern mit Bandwächtern überwacht.

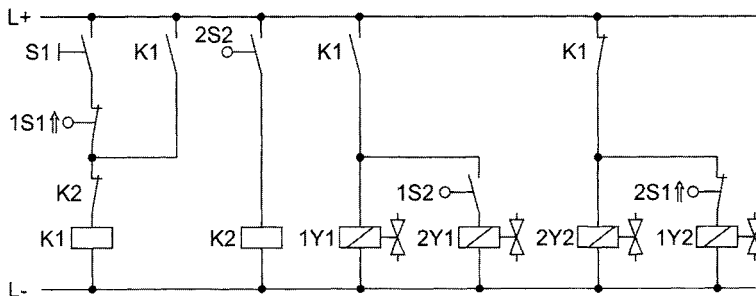
Technologieschema:



1. Geben Sie die Zuordnungstabelle mit allen Eingängen und Ausgängen an.
2. Bestimmen Sie mit einer RS-Tabelle die Bedingungen für die Ansteuerung der Setz- und Rücksetz-Eingänge der Speicherglieder für die Ansteuerschütze Q1, Q2 und Q3.
3. Ermitteln Sie aus der RS-Tabelle den Funktionsplan.
4. Die Steuerungsfunktion soll mit einem Baustein bibliotheksfähig realisiert werden. Geben Sie die Deklarationstabelle des Bausteins an.
5. Schreiben Sie die Anweisungsliste AWL des Steuerungsprogramms.

Kontrollaufgabe 3.4

Der nachfolgend angegebene Stromlaufplan dient der Ansteuerung von zwei doppelt wirkenden Zylindern 1A und 2A über 5/2-Wegeventile mit jeweils zwei Magnetspulen 1Y1 und 1Y2 bzw. 2Y1 und 2Y2.

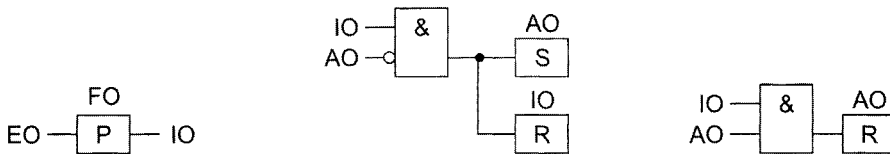


1. Zeichnen Sie den pneumatischen Schaltplan mit den beiden Zylindern und den Ventilen. Das Ausfahren der Zylinder soll über Drosselrückschlagventile verzögert werden.
2. Bestimmen Sie das Funktionsdiagramm, das beide Schütze K1 und K2, die Magnetventile 1Y1, 1Y2, 2Y1 und 2Y2 sowie die beiden Zylinder 1A und 2A enthält.

- Die Ansteuerung der vier Magnetspulen soll künftig von einer SPS aus erfolgen. Bestimmen Sie aus dem gegebenen Stromlaufplan einen Funktionsplan gleicher Struktur.
- Der in 3. gefundene Funktionsplan soll durch Einführung eines Speichergliedes und den dadurch möglichen Verzicht auf Hilfsschütze vereinfacht werden. Zeichnen Sie den vereinfachten Funktionsplan.
- Der vereinfachte Funktionsplan soll mit einem bibliotheksfähigen Baustein realisiert werden. Bestimmen Sie die Deklarationstabelle des Bausteins.

Kontrollaufgabe 3.5

Gegeben ist der nachfolgende Funktionsplan bestehend aus drei Netzwerken.



- Erklären Sie die Funktion, die mit dem Netzwerk 1 ausgeführt wird.
- Am Eingang EO findet ein Signalwechsel von „0“ nach „1“ statt. Erklären Sie, welche Auswirkungen dieser Signalwechsel auf die Variablen FO, IO und AO hat.
- Das Steuerungsprogramm ist in einem Funktionsbaustein FB mit dem Eingang EO und dem Ausgang AO realisiert. Geben Sie die vollständige Deklarationstabelle für den Funktionsbaustein an.
- Geben Sie die Anweisungsliste des Funktionsbausteins an.

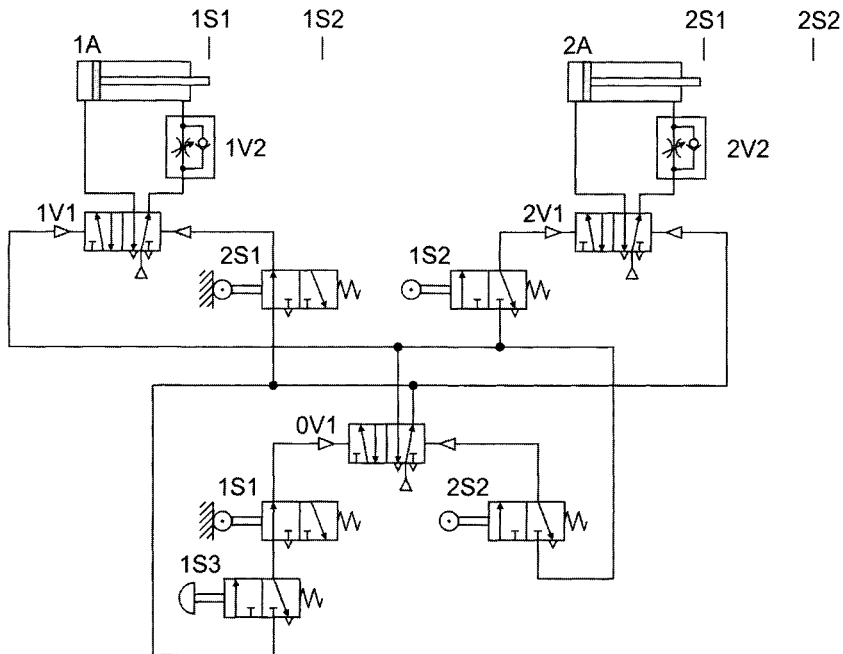
Kontrollaufgabe 3.6

Durch einmaliges Betätigen eines Tasters S1 wird die Meldeleuchte H1 eingeschaltet. Wird der Taster S1 nochmals betätigt, wird die Meldeleuchte H2 eingeschaltet. Bei der dritten Betätigung des Tasters S1 wird die Meldeleuchte H2 und bei der nachfolgenden Betätigung auch die Meldeleuchte H1 ausgeschaltet. Der Vorgang kann dann wiederholt werden.

- Zeichnen Sie ein Funktionsdiagramm mit S1, H1 und H2, aus dem der Ablauf der Steuerungsaufgabe deutlich wird.
- Bestimmen Sie in einer Schaltfolgetabelle für jede Betätigung die Bedingungen für das Setzen und das Rücksetzen der Meldeleuchten.
- Zeichnen Sie den sich aus der Schaltfolgetabelle ergebenden Funktionsplan für die Steuerung.
- Der Funktionsplan soll mit einem bibliotheksfähigen Baustein realisiert werden. Bestimmen Sie die Deklarationstabelle des Bausteins.
- Geben Sie die Anweisungsliste des Bausteins an.

Kontrollaufgabe 3.7

Der gegebene Pneumatikplan zeigt die Ansteuerung von zwei Zylindern. Die Ansteuerung der Zylinder soll künftig über 5/2-Wegeventile mit beidseitig elektromagnetischer Betätigung durch eine SPS erfolgen. Die Endlagengeber werden durch induktive Sensoren an den Zylindern und das handbetätigte 3/2-Wegeventil durch einen Taster S1 ersetzt.



1. Ermitteln Sie mit einem Funktionsdiagramm in welcher Reihenfolge nach Betätigung des Tastventils 1S3 die Zylinder aus- und einfahren.
2. Zeichnen Sie den Pneumatikplan für die Ansteuerung mit einer SPS.
3. Bestimmen Sie mit einer Schaltfolgetabelle die Bedingungen für das Setzen und Rücksetzen der Magnetventile 1Y1, 1Y2, 2Y1 und 2Y2.
4. Zeichnen Sie den sich aus der Schaltfolgetabelle ergebenden Funktionsplan.
5. Der Funktionsplan soll mit einem bibliotheksfähigen Baustein realisiert werden. Bestimmen Sie die Deklarationstabelle des Bausteins.
6. Geben Sie die Anweisungsliste des Bausteins an.

Kontrollaufgabe 3.8

Das gegebene Funktionsdiagramm zeigt in welcher Reihenfolge die Zylinder 1A und 2A aus- und einfahren, nachdem das Tastventil 1S3 betätigt wurde.

| Bauglieder | | | Zeit | | | | |
|-------------|--------|-------------|-----------|---|---|---|---|
| Benennung | Kennz. | Zustand | Schritt 1 | 2 | 3 | 4 | 5 |
| Taster | 1S3 | EIN | | | | | |
| DW-Zylinder | 1A | ausgefahren | | | | | |
| | | eingefahren | | | | | |
| DW-Zylinder | 2A | ausgefahren | | | | | |
| | | eingefahren | | | | | |

Die Ansteuerung der Zylinder erfolgte bisher über folgende pneumatische Ventile: zwei 5/2-Wegeventile durch Druckbeaufschlagung direkt betätigt, zwei Drosselrückschlagventile zum Einstellen der Ausfahrgeschwindigkeit, ein 3/2-Wegeventil durch Druckknopf betätigt, zwei 3/2-Wegeventile mit Rollenbetätigung durch die vordere Endlage der Zylinder und ein 3/2-Wegeventil mit Rollenbetätigung in nur einer Richtung bei Zylinder 1A.

1. Zeichnen Sie den Pneumatikschaltplan mit den angegebenen Ventilen.
2. Die beiden Zylinder sollen künftig von einer SPS angesteuert werden. Dazu werden im neuen Pneumatikplan zwei 5/2-Wegeventile beidseitig elektromagnetisch betätigt, zwei Drosselrückschlagventile und vier induktive Endlagengeber eingesetzt. Zeichnen Sie den neuen Pneumatikplan.
3. Bestimmen Sie mit einer Schaltfolgetabelle die Bedingungen für das Setzen und Rücksetzen der Magnetventile 1Y1, 1Y2, 2Y1 und 2Y2.
4. Zeichnen Sie den sich aus der Schaltfolgetabelle ergebenden Funktionsplan.
5. Der Funktionsplan soll mit einem bibliotheksfähigen Baustein realisiert werden. Bestimmen Sie die Deklarationstabelle des Bausteins.
6. Schreiben Sie die Anweisungsliste des Bausteins.

Kontrollaufgabe 3.9

Gegeben ist das nachfolgende Steuerungsprogramm eines Funktionsbausteins FB 10.

FUNCTION_BLOCK **FB 10**

```

VAR_INPUT           VAR_OUTPUT           H1 : BOOL           VAR
LI1 : BOOL;         VZI : BOOL;          H2 : BOOL           FO1 : BOOL;
LI2 : BOOL;         RZI : BOOL;          END_VAR             FO2 : BOOL;
END_VAR                                                     END_VAR

BEGIN

Netzwerk1           Netzwerk2           Netzwerk3           Netzwerk4
U #LI1;             U #LI1;             U #VZI;             U #RZI;
U(;                 U(;                 S #H1;              S #H2;
U #LI2;             U #LI2;             U #RZI;             U #VZI;
FP #FO1;            FN #FO2;            R #H1;              R #H2;
);                  );
= #VZI;             = #RZI;
END_FUNCTION_BLOCK

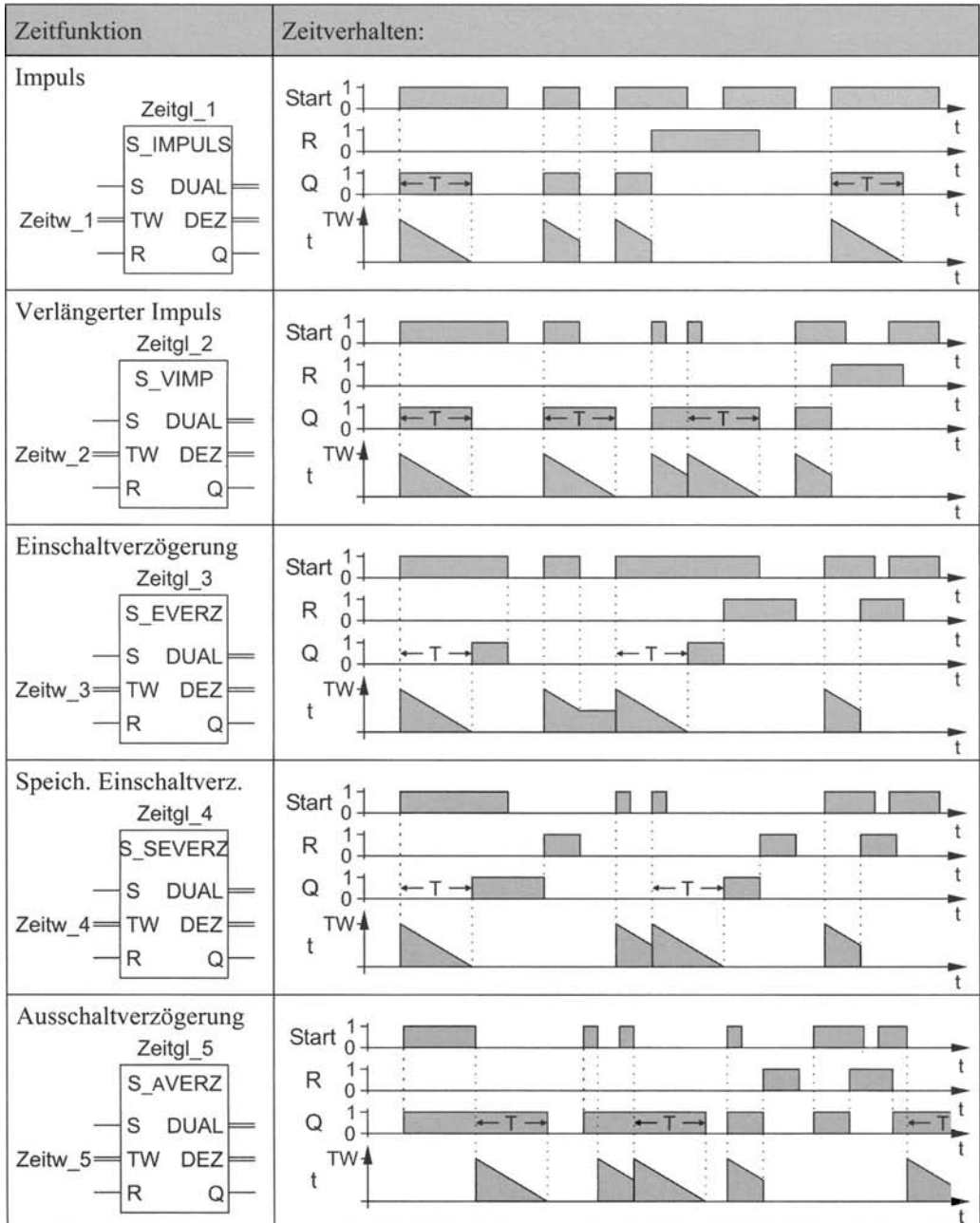
```

1. Welche Bedeutung hat die Angabe "BOOL" im Deklarationsteil des Funktionsbausteins.
2. Geben Sie für das Netzwerk1 und das Netzwerk 3 den Funktionsplan an.
3. Der Funktionsbaustein FB 10 wird im Organisationsbaustein OB1 in Funktionsplandarstellung aufgerufen. Bestimmen Sie den Funktionsplan des Bausteinaufrufs.
4. Bei welcher Signalkombination von LI1 und LI2 erhält die Variable VZI „I“-Signal.

4

Zeitfunktionen, Taktsignale, Zählerfunktionen, freigrafischer Funktionsplan

Zeitfunktionen SI, SV, SE, SS, SA



Programmieren von Zeitfunktionen

Operanden und Datentyp der Übergabeparameter

| Übergabeparameter | Beschreibung | Operand (Auswahl) | Datentyp |
|-------------------|--|-------------------|----------|
| | Tx: Zeitoperand T0...T15 (CPU abhängig) | T | TIMER |
| | T_Fkt. Zeitfunktion (SI, SV, SE, SS, SA) | --- | --- |
| | S: Starteingang | E, A, T, Z | BOOL |
| | TW: Vorgabewert der Zeitdauer | Konst, EW | S5TIME |
| | R: Rücksetzeingang | E, A, T, Z | BOOL |
| | DUAL: Restwert der Zeit dual-codiert | AW, DBW | WORD |
| | DEZ: Restwert der Zeit BCD-codiert | AW, DBW | WORD |
| | Q: Status des Zeitoperanden | A, DBX | BOOL |

Starten und Rücksetzen einer Zeitfunktion

Das Starten einer Zeit SI (Impuls), SV (verzögerter Impuls), SE (Einschaltverzögerung), SS (speichernde Einschaltverzögerung) mit steigenden Flanke (0 → 1) am Starteingang S.

Starten einer Zeit SA (Ausschaltverzögerung) mit fallender Flanke (1 → 0) am Starteingang S.

Rücksetzen einer Zeitfunktion mit einem 1-Signal am Rücksetzeingang (R) des Zeitgliedes, dabei wird die Restzeit gelöscht.

Eingabe der Zeitdauer

| als Konstante mit der Syntax S5T#aHbbMccSdddMS | als deklarierte Variable Name : Zeitw Datentyp: S5TIME | als Wortoperand z.B. für BCD-Zahleneinsteller, der an EW 8 angeschlossen ist. |
|--|--|---|
| AWL: L S5T#1S | AWL: L #Zeitwert | AWL: L EW 8 |
| FUP: | FUP: | Bit-Belegung der Zeitdauer im Akku |

Abfragen einer Zeitfunktion

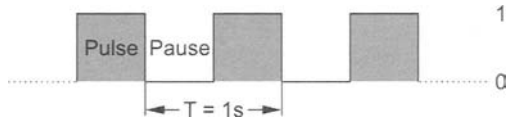
| | Zeitstatus (binär) abfragen | Restzeitwert (digital) abfragen |
|-----|--|--|
| AWL | U #Zeitgl; O #Zeitgl UN #Zeitgl; ON #Zeitgl | L #Rest_dual / in Akku LC #Rest_bcd / in Akku |
| FUP | | |

Erzeugen von Taktsignalen

Taktmerker eines in der CPU projektierten Merkerbytes (z.B. MB 0):

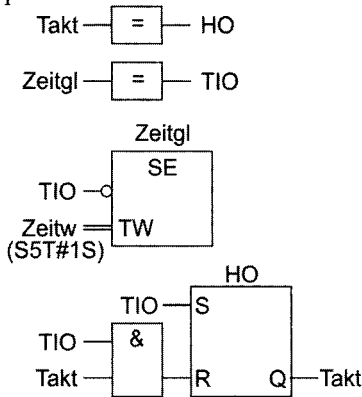
| Bit | M 0.7 | M 0.6 | M 0.5 | M 0.4 | M 0.3 | M 0.2 | M 0.1 | M 0.0 |
|-------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Frequenz (Hz) | 0,5 | 0,625 | 1 | 1,25 | 2 | 2,5 | 5 | 10 |
| Periodendauer (s) | 2 | 1,6 | 1 | 0,8 | 0,5 | 0,4 | 0,2 | 0,1 |

Abfrage von M 0.5 liefert die Impulsfolge:

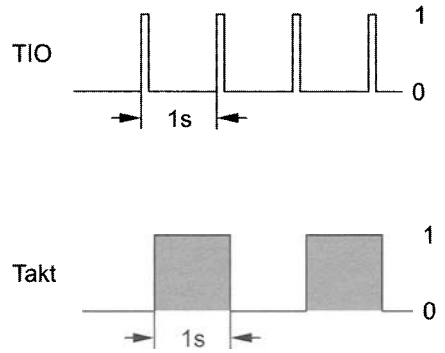


Taktgeber

Funktionsplan:



Zeitdiagramme:



Deklarationstabelle für FC-Taktgeber:

| Name | Datentyp | Kommentar |
|-------------|----------|---------------------|
| IN | | |
| Zeitgl | Timer | |
| Zeitw | S5Time | |
| OUT | | |
| Takt | Bool | Taktsignal |
| TEMP | | |
| HO | Bool | Hilfsoperand |
| TIO | Bool | Timer-Impulsoperand |

Aufruf im OBI:



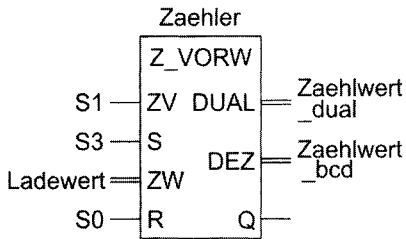
Hinweis. Die Variable Takt müsste eigentlich als IN_OUT-Variable deklariert werden. Zur besseren Darstellung der Ausgangsvariablen im Funktionsplan, erlaubt STEP 7 in diesem Fall auch die Deklaration der Variablen als OUT-Variable.

Weckalarmer

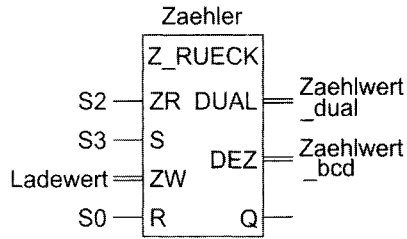
| | |
|---|--|
| In der CPU einer S7-300 gibt es den Weckalarm OB35 mit fester Bearbeitungspriorität 12; einstellbar ist das Zeitintervall von 1 ms bis 1 min in 1 ms-Schritten. | Weckalarmprogramm im OP 35: UN A 4.0 = A 4.0 |
|---|--|

Zählerfunktionen ZV, ZR

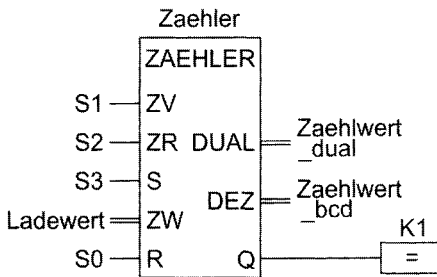
Vorwärtszähler



Rückwärtszähler



Vorwärts-Rückwärtszähler



AWL:

```

U #S1      L #Zaehler
ZV#Zaehler T #Zaehlwert_dual
U #S2      LC#Zaehler
ZR#Zaehler T #Zaehlwert_bcd
U #S3      U #Zaehler
L #Ladewert = #K1
S #Zaehler
U #S0
R #Zaehler
  
```

Einzelne Zählerfunktionen



Programmieren von Zählfunktionen

Operanden und Datentyp der Übergabeparameter

| Übergabeparameter | Beschreibung | Operand (Auswahl) | Datentyp |
|-------------------|--------------------------------------|-------------------|----------|
| | Zx: Zähleroperand Z0..Z63 (CPU abh.) | Z | COUNTER |
| | T_Fkt: Zeitfunktion (ZV, ZR, Z) | --- | --- |
| | ZV: Vorwärtszähleingang | E, A, DBX | BOOL |
| | ZR: Rückwärtszähleingang | E, A, DBX | BOOL |
| | S: Zählfunktion auf Ladewert setzen | E, A, T, Z | BOOL |
| | ZW: Ladewert | Konst, EW | WORD |
| | R: Rücksetzeingang | E, A, T, Z | BOOL |
| | DUAL: Zählwert dual codiert | AW, DBW | WORD |
| | DEZ: Zählwert BCD-codiert | AW, DBW | WORD |
| | Q: Status des Zähleroperanden | A, DBX | BOOL |

Vorwärtszählen, Rückwärtszählen

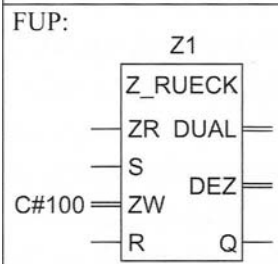
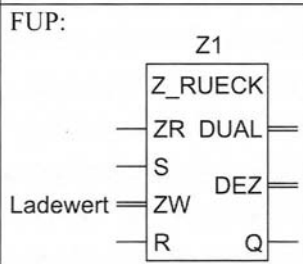
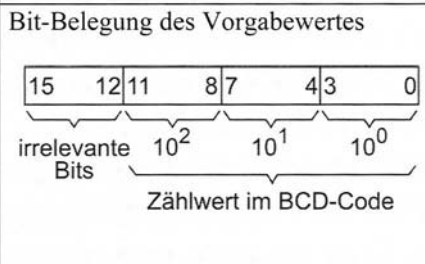
Bei steigender Flanke ($0 \rightarrow 1$) am Vorwärtszähleingang ZV wird der Zählerstand um eins erhöht. Bei Erreichen der oberen Zählgrenze von 999 haben weitere Zählimpulse keine Auswirkung mehr.

Bei steigender Flanke ($0 \rightarrow 1$) am Rückwärtszähleingang ZR wird der Zählerstand um eins verringert. Bei Erreichen der unteren Zählgrenze 0 haben weitere Zählimpulse keine Auswirkung mehr.

Zähleranfangswert setzen

Mit einer steigenden Flanke ($0 \rightarrow 1$) am Setzeingang S wird die Zählerfunktion auf den im Akkumulator stehenden Zahlenwert gesetzt.

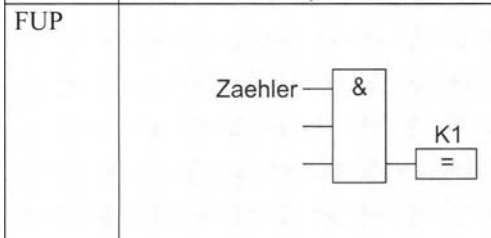
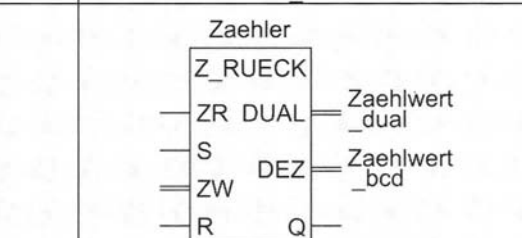
Eingabe eines Zähleranfangswertes

| als Konstante | als deklarierte Variable Name : Ladewert Datentyp: WORD | als Wortoperand z.B. für BCD-Zahleneinsteller, der an EW 8 angeschlossen ist. |
|--|--|---|
| AWL: L C#100 L W#16#100 | AWL: L #Ladewert | AWL: L EW 8 |
| FUP:  | FUP:  | Bit-Belegung des Vorgabewertes  |

Rücksetzen der Zählerfunktion

Mit einem 1-Signal am Rücksetzeingang R des Zählers wird der Zählerwert auf null zurückgesetzt. Bei erfüllter Rücksetzbedingung kann weder ein Vorgabewert gesetzt noch gezählt werden.

Abfragen einer Zählerfunktion

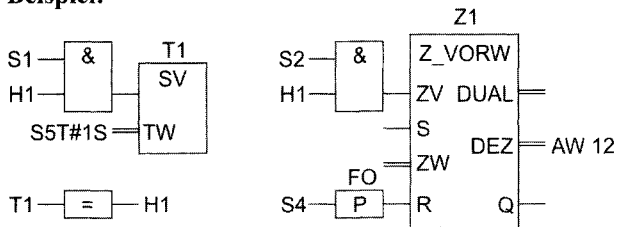
| | Zählerstatus (binär) abfragen | Zählerwert (digital) abfragen |
|-----|---|--|
| AWL | U #Zaehler; O #Zaehler; UN #Zaehler; ON #Zaehler | L #Zaehlwert_dual / in Akku LC #Zaehlwert_bcd / in Akku |
| FUP |  |  |

Freigrafische Funktionsplandarstellung

Nutzen: Erweiterte Darstellungsmöglichkeiten gegenüber STEP 7-FUP

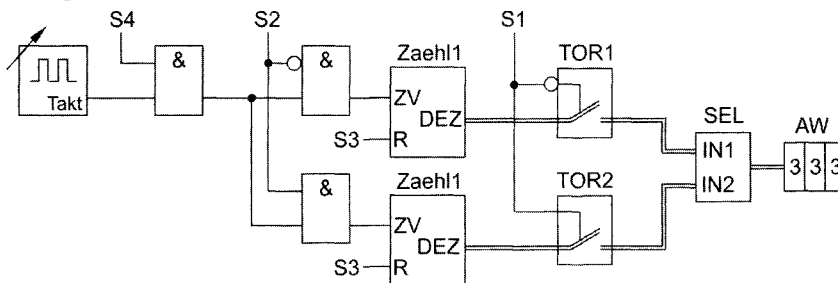
1. Didaktisch sinnvolle grafische Unterscheidung von Bitleitungen und Wortleitungen. Einfachlinien für Bitleitungen bei booleschen Operanden (Variablen) und Doppellinien für Wortleitungen bei Wortoperanden (Variablen) mit den Datentypen BYTE, WORD, DWORD, INT, DINT, REAL, S5TIME und den Paramertypen TIMER, COUNTER.

Beispiel:



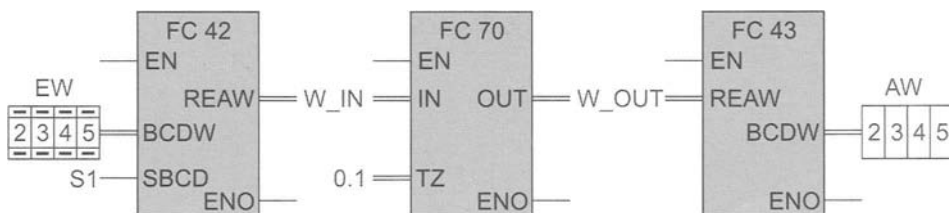
2. Übersichtliche Funktionsplandarstellung ohne Aufteilung in Netzwerke, die in STEP 7 nur bei booleschen, jedoch nicht bei digitalen Operationen möglich ist.

Beispiel:



3. Darstellung von Bauelementstrukturen, die einen Programmaufbau übersichtlicher machen.

Beispiel:



Programmierung:

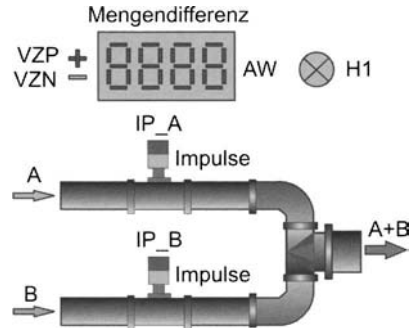
Freigrafische Funktionsplandarstellung – auch Continuous Function Chart (CFC) genannt – ermöglicht die vollgrafische Programmierung unter Verwendung von digitalen und regelungstechnischen Symbolen für komplexere Funktionen (z.B. PID-Regler) ergänzt mit abstrakten CFC-Objekten und Anzeigefeldern zur Darstellung von Messwertverläufen. Es bestehen Umschaltmöglichkeiten zwischen Übersichts- und Detaildarstellungen. Die freigrafische Funktionsplandarstellung ist in der Automatisierungstechnik verfügbar (z.B. SIMATIC CFC).

Bei der Programmeingabe eines Funktionsplans in STEP 7 müssen die Einschränkungen des FUP-Editors beachtet werden, d.h. einige Möglichkeiten der freigrafischen Darstellung sind nicht verfügbar.

4.1 Beispiel
Zeitfunktionen, Taktgeber, Zähler, freigraferischer Funktionsplan

Überwachung eines Mengenverhältnisses im kontinuierlichen Verarbeitungsprozess

In einem kontinuierlichen Prozess müssen zwei Stoffmengen A und B in einem konstanten Mengenverhältnis verarbeitet werden. Die Zuführung der Stoffmengen wird durch Sensoren erfasst, welche die Impulse IP_A und IP_B liefern. Ein Überwachungsprogramm soll die Abweichungen vom vorgegebenen Mengenverhältnis als vorzeichenbehafteten Differenzwert $DIFF = IP_A - IP_B$ erfassen, mit dem ggf. eine Nachregelung auf $DIFF = 0$ erfolgen kann. Die Bildung der Mengendifferenz soll in einem Vorwärts-Rückwärts-Zähler erfolgen.



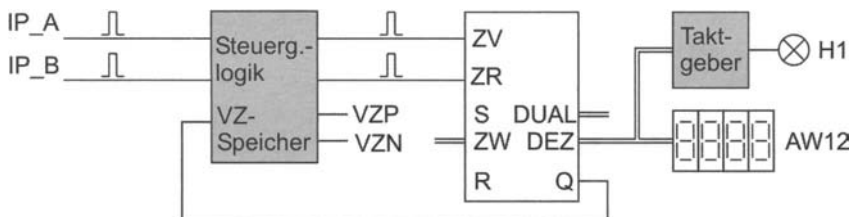
Da ein solcher Zähler nicht in den negativen Zahlenbereich ($DIFF < 0$) zählen kann, ist eine Zusatzschaltung erforderlich, die vom Zählerstand null an alle weiteren Rückwärtszählimpulse in Vorwärtsrichtung und Vorwärtszählimpulse in Rückwärtsrichtung bei gleichzeitiger Ausgabe eines negativen Vorzeichensignals $VZN = 1$ zählt. Entsprechend ist für den positiven Zahlenbereich ($DIFF > 0$) ein positives Vorzeichensignal $VZP = 1$ zu setzen. Eine Ziffernanzeige soll die Mengendifferenz anzeigen. Eine Meldeleuchte H1 blinkt bei auftretenden Differenzwerten (im simulierten Anlagenmodell erst bei $|DIFF| > 80$ erkennbar).

1. Aus dem Aufgabentext ist eine freigraferische Funktionsplandarstellung mit allen Signalen zu entwickeln (Steuerungsprinzip).
2. Der freigraferische Funktionsplan ist in ein Steuerungsprogramm umzusetzen. Neben der Steuerungslogik ist noch ein Taktgeber für das Blinklicht zu entwerfen. Die Einschaltzeit des Blinktaktes von H1 soll sich mit zunehmendem Differenzwert DIFF verlängern.

Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|---------------------|--------|----------|--------------------|-------------------|
| Sensor A | IP_A | BOOL | Impulse | IP_A = 1 E 0.0 |
| Sensor B | IP_B | BOOL | Impulse | IP_B = 1 E 0.1 |
| Ausgangsvariable | | | | |
| BCD-Anzeige | DIFF | WORD | BCD-Code | AW12 |
| Vorzeichenanzeige + | VZP | BOOL | Leuchtet | VZP = 1 A 4.0 |
| Vorzeichenanzeige - | VZN | BOOL | Leuchtet | VZN = 1 A 4.1 |
| Meldeleuchte | H1 | BOOL | Blinklicht | A 4.2 |

Freigraferischer Funktionsplan:



Steuerungslogik des Zweiquadrantenzählers ZAE:

Vorwärtszähleingang:

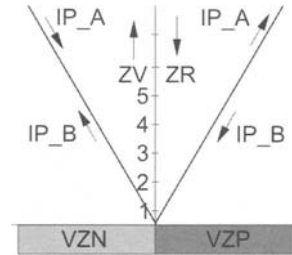
$$ZV = (VZP \& IP_A \uparrow) \vee (VZN \& IP_B \uparrow) \vee ZAE (IP_A \uparrow \vee IP_B \uparrow)$$

Rückwärtszähleingang

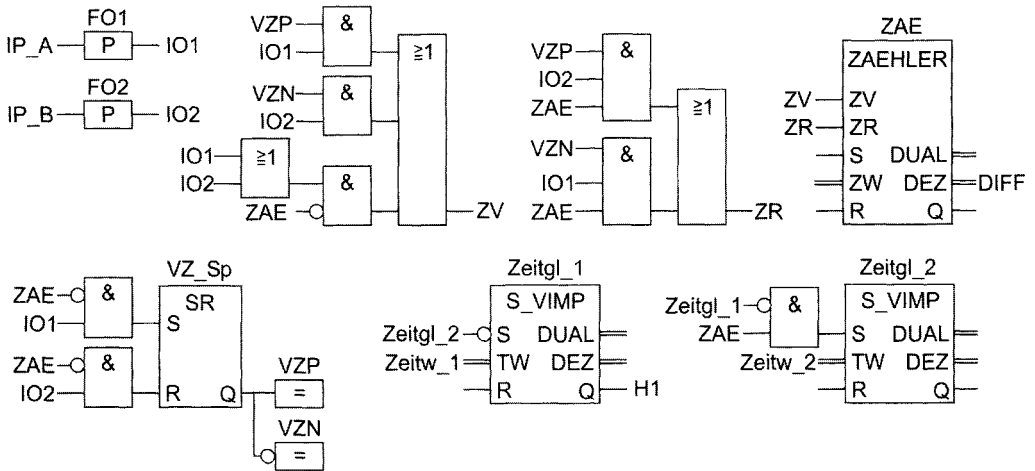
$$ZR = (VZP \& IP_B \uparrow \& ZAE) \vee (VZN \& IP_A \uparrow \& ZAE)$$

Vorzeichenspeicher VZ_Sp:

$$S = \overline{ZAE} \& IP_A \uparrow; \quad R = ZAE \& IP_B \uparrow$$



Funktionsplan:

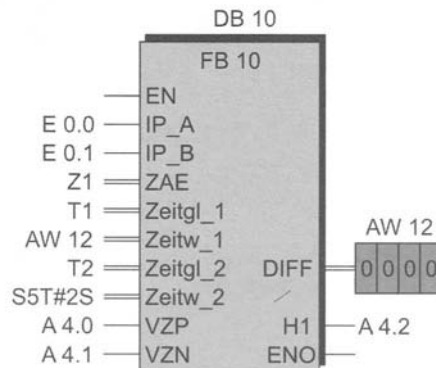


Deklarationstabelle: FB10

| Name | Datentyp | Anfangswert |
|----------|----------|-------------|
| IN | | |
| IP_A | Bool | FALSE |
| IP_B | Bool | FALSE |
| Zaehler | Counter | |
| Zeitgl_1 | Timer | |
| Zeitw_1 | S5Time | S5T#0MS |
| Zeitgl_2 | Timer | |
| Zeitw_2 | S5Time | S5T#0MS |
| OUT | | |
| DIFF | Word | W#16#0 |
| H1 | Bool | FALSE |
| IN_OUT | | |
| VZP | Bool | FALSE |
| VZN | Bool | FALSE |
| STAT | | |
| VZ_Sp | Bool | FALSE |
| FO1 | Bool | FALSE |
| FO2 | Bool | FALSE |

| Name | Datentyp | Anfangswert |
|------|----------|-------------|
| TEMP | | |
| ZV | Bool | |
| ZR | Bool | |
| IO1 | Bool | |
| IO2 | Bool | |

Aufruf im OB1:



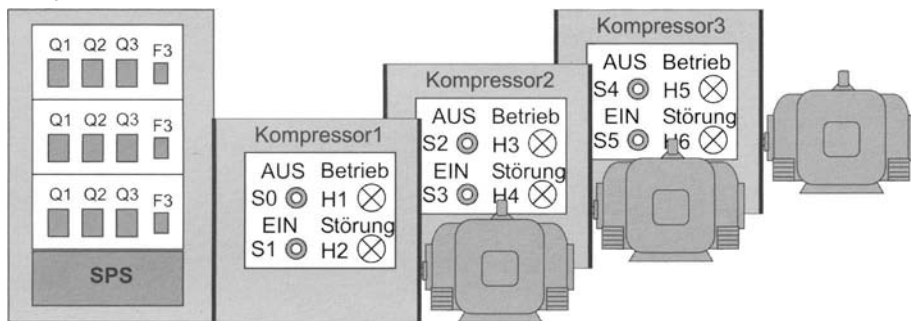
4.2

Lernaufgaben

Lernaufgabe 4.1: Stern-Dreieck-Anlauf einer Kompressoranlage

Lös. S. 173

Die Drehstrom-Asynchronmotoren der drei Kompressoren werden jeweils mit einer selbsttätigen Stern-Dreieck-Umschaltung angefahren. Die Umschaltung von Stern auf Dreieck erfolgt automatisch nach 5 Sekunden. Zur Vermeidung von eventuellen Schäden, falls einer der Schütze defekt ist, geben die Schütze eine Rückmeldung über den tatsächlichen Schaltzustand. Ein Überlastrelais schützt den Motor vor thermischer Überbeanspruchung.

Technologieschema:

Zur Bedienung eines Kompressors ist ein EIN-Taster S1 und ein AUS-Taster S0 vorgesehen. Die Meldeleuchte H1 zeigt den Betrieb des Antriebsmotors an. Hat ein Schütz 1s nach der Ansteuerung nicht angezogen bzw. ist 1s nach Abschaltung des Schützes der Rückmeldekontakt noch nicht abgefallen oder löst das Überstromrelais aus, zeigt dies die Störungsleuchte H2 an. Der Antrieb wird dann sofort abgeschaltet. Die Störungsanzeige H2 wird durch Betätigung des AUS-Taster gelöscht.

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge für einen Kompressorantrieb. Legen Sie dabei die erforderliche Zuordnung der Signalgeber fest.
2. Bestimmen Sie mit einer RS-Tabelle die Bedingungen für die Ansteuerung der Setz- und Rücksetz-Eingänge der Speicherglieder für Q1, Q2 und Q3. Erweitern Sie die Tabelle durch die erforderlichen Zeitglieder und tragen Sie die Bedingungen für den Start (Setzen) und das Rücksetzen der Zeitglieder dort ein.
3. Ermitteln Sie aus der erweiterten RS-Tabellen den Funktionsplan für die Ansteuerung der RS-Speicherglieder und der Zeitglieder.
4. Da das Steuerungsprogramm zur Ansteuerung der noch vorhandenen Kompressoren auf der Anlage genutzt werden soll, ist ein bibliotheksfähiger Baustein zu erzeugen. Überlegen Sie, ob Sie dafür eine Funktion verwenden können und geben Sie die Deklarationstabelle für den Baustein an.
5. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.

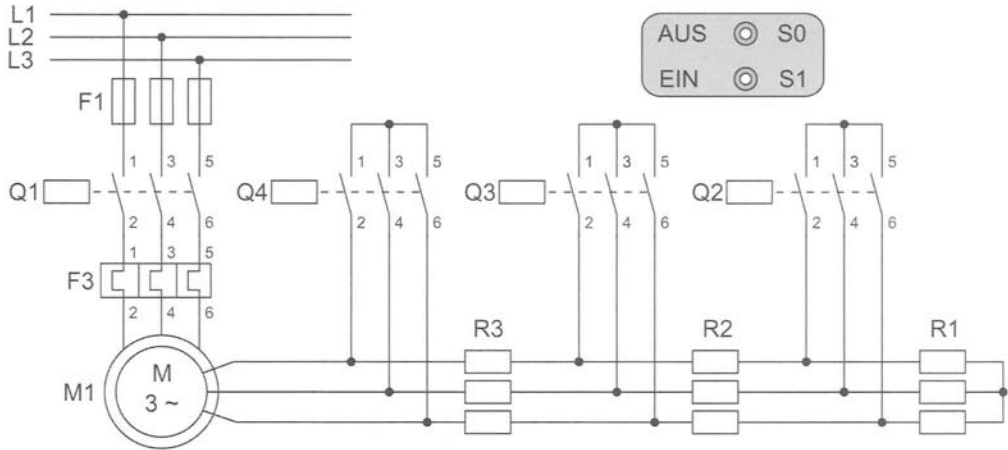
Lernaufgabe 4.2: Anlassersteuerung

Lös. S. 175

Bei Drehstrom-Schleifringläufermotoren werden zur Vermeidung eines hohen Einschaltstromes Widerstandsgruppen in den Läuferkreis geschaltet. Wird der EIN-Taster S1 betätigt,

zieht das Netzschütz Q1 an. Die Schütze Q2, Q3 und Q4 ziehen dann jeweils nach Ablauf einer Verzögerungszeit von 5s in der Reihenfolge Q2, Q3, Q4 an und schließen nacheinander die entsprechenden Widerstandsgruppen R1 bis R3 kurz. Hat das letzte Schütz angezogen, sind die Schleifringe des Läufers kurzgeschlossen und der Motor läuft im Nennbetrieb. Bei Betätigung des AUS-Tasters S0 wird der Motor ausgeschaltet.

Technologieschema:

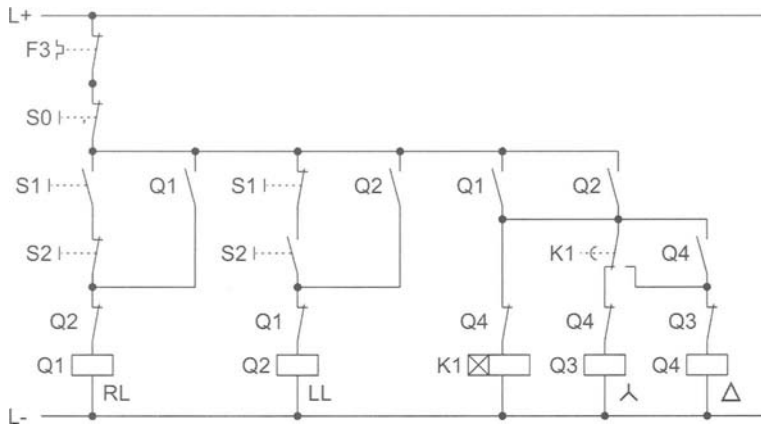


- Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
- Bestimmen Sie mit einer RS-Tabelle die Bedingungen für die Ansteuerung der Setz- und Rücksetz-Eingänge der Speicherglieder für Q1, Q2, Q3 und Q4. Erweitern Sie die Tabelle durch drei Zeitfunktionen und tragen Sie die Bedingungen für den Start (Setzen) der Zeitfunktionen dort ein.
- Ermitteln Sie aus der erweiterten RS-Tabelle den Funktionsplan für die Ansteuerung der Schütze Q1 bis Q4 und der Zeitfunktionen.
- Bestimmen Sie mit einer neuen RS-Tabelle die Bedingungen für die Ansteuerung der Setz- und Rücksetz-Eingänge der Speicherglieder für Q1, Q2, Q3 und Q4, wenn davon ausgegangen wird, dass die drei Verzögerungszeiten von nur einer Zeitfunktion gebildet wird. Erweitern Sie die Tabelle durch die eine Zeitfunktion und tragen Sie die Bedingungen für den Start (Setzen) und das Rücksetzen der Zeitfunktion dort ein.
- Ermitteln Sie aus der RS-Tabelle den Funktionsplan für die Ansteuerung der Motorschütze Q1 bis Q4 und der Zeitfunktion.
- Das Steuerungsprogramm soll in einem bibliotheksfähigen Baustein realisiert werden. Überlegen Sie, ob Sie dafür eine Funktion verwenden können und geben Sie die Deklarationstabelle für den Baustein an.
- Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.

Lernaufgabe 4.3: Automatische Stern-Dreieck-Wendeschtaltung

Lös. S. 176

Der nachfolgende Stromlaufplan zeigt die Ansteuerung der vier erforderlichen Leistungsschütze für eine automatische Stern-Dreieck-Wendeschtaltung.

Stromlaufplan:

Die Leistungsschütze Q1 bis Q4 sollen künftig von einer SPS angesteuert werden.

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Zeichnen Sie den Hauptstromkreis der Stern-Dreieck-Schaltung.
3. Wandeln Sie den gegebenen Stromlaufplan nach den entsprechenden Umsetzungsregeln in einen Funktionsplan um. Überlegen Sie, ob eine Tasterverriegelung von S1 und S2 erforderlich ist.
4. Das Steuerungsprogramm soll in einem bibliotheksfähigen Baustein realisiert werden. Überlegen Sie, ob Sie dafür eine Funktion verwenden können und geben Sie die Deklarationstabelle für den Baustein an.
5. Schreiben Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Übergabeparameter mit den E-/A-Operanden aus Ihrer Zuordnungstabelle.
6. Das Steuerungsprogramm soll als Neuentwurf mit RS-Speicherfunktionen zur Ansteuerung der Leistungsschütze ausgeführt werden. Bestimmen Sie die RS-Tabelle.
7. Ermitteln Sie aus der RS-Tabelle den Funktionsplan.
8. Der gefundene Funktionsplan ist in einem bibliotheksfähigen Baustein umzusetzen. Überlegen Sie, welche Bausteinart Sie für die Umsetzung verwenden können und geben Sie die Deklarationstabelle für den Baustein.
9. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.

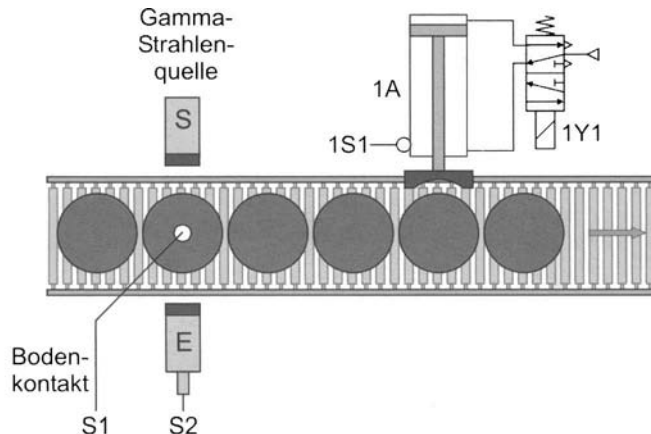
Lernaufgabe 4.4: Füllmengenkontrolle

Lös. S. 179

Am Ende einer Konservendosenabfüllanlage werden die Dosen auf vollständige Füllung kontrolliert. Dazu werden die Dosen auf einem Förderband nacheinander mit einem kleinen Zwischenraum durch eine Gamma-Strahlenquelle transportiert. Der Empfänger des Sensors meldet den Signalzustand „1“, wenn eine ungenügende Füllung der Dose vorliegt. Die Messung wird jedoch nur ausgeführt, wenn eine Dose den Bodenkontakt S1 betätigt (Signalzustand „1“ bei Betätigung). Zum Auswerfen einer nicht korrekt gefüllten Dose muss das elektropneumatische Ventil 1Y1 zwei Sekunden nach der Messung kurz angesteuert werden.

Zu beachten ist, dass sich maximal vier Dosen zwischen der Erfassung und dem Auswerfer befinden können und dass auch mehrere Dosen hintereinander die geforderte Füllmengen unterschreiten können.

Technologieschema:



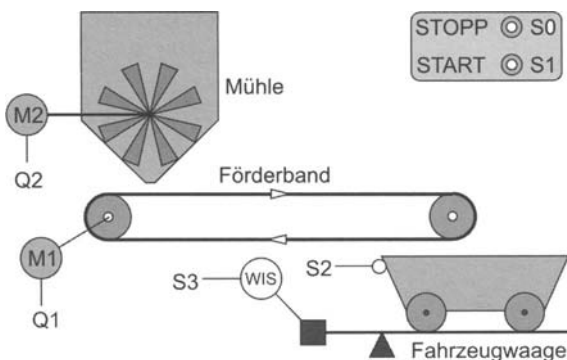
1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Für maximal vier Dosen müssen Zeitglieder gestartet werden. Ermitteln Sie mit einer RS-Tabelle die Bedingungen für die Ansteuerung der vier Zeitglieder sowie für den RS-Speicher zur Ansteuerung des Ventils 1Y1.
3. Ermitteln Sie aus der RS-Tabelle den Funktionsplan.
4. Der gefundene Funktionsplan ist in einem bibliotheksfähigen Baustein umzusetzen. Überlegen Sie, welche Bausteinart Sie für die Umsetzung verwenden können und geben Sie die Deklarationstabelle für den Baustein an.
5. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.

Lernaufgabe 4.5: Zerkleinerungsanlage

Lös. S. 182

In einer Zerkleinerungsanlage für Steingut wird das zerkleinerte Material aus einer Mühle über ein Transportband in einen Wagen verladen.

Technologieschema:



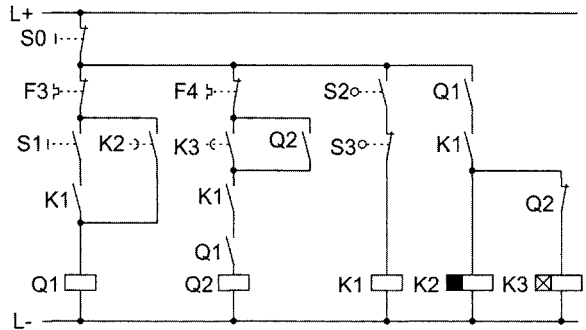
Der Abfüllvorgang für den Wagen kann durch Betätigen des Tasters S1 gestartet werden, wenn ein Wagen an der Rampe steht (S2 = „1“). Um Stauungen des Fördergutes auf dem Transportband zu vermeiden, muss das Band zwei Sekunden laufen, bevor die Mühle mit Q2 eingeschaltet wird.

Meldet die Waage mit S3 = 0, dass der Wagen gefüllt ist, wird die Mühle ausgeschaltet. Das Förderband läuft noch drei Sekunden nach, um das Steingut vollständig vom Band zu entfernen.

Durch Betätigung des Stopp-Tasters S0 wird der Abfüllvorgang sofort unterbrochen und das Förderband abgeschaltet. Die beiden Motoren sind jeweils mit einem Thermorelais F3, F4 gegen Überlastung geschützt.

Die Ansteuerung der Motoren M1 und M2 über die Hauptschütze Q1 und Q2 erfolgte bisher durch nebenstehende Schützsteuerung:

Eine Revision der Zerkleinerungsanlage wird zum Anlass genommen, die Ansteuerung der Leistungsschütze Q1 und Q2 künftig mit einem Automatisierungssystem auszuführen. Dazu ist die gegebene Schützsteuerung durch ein SPS-Programm zu ersetzen.

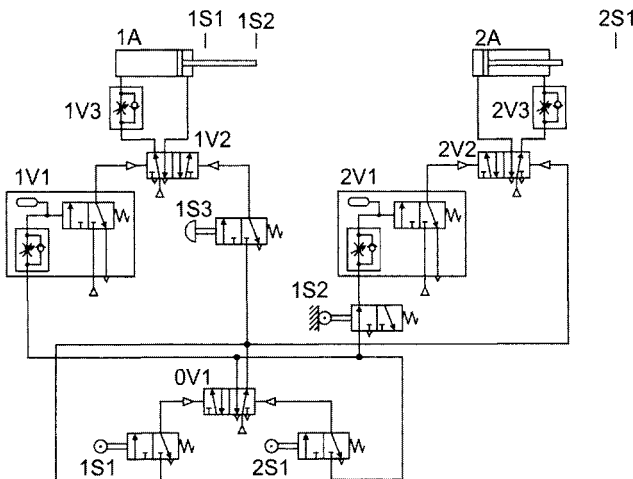


1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Geben Sie einen Befüllungsdurchlauf von der Betätigung der Start-Taste S1 bis zur Vollmeldung des Wagens in einem Funktionsdiagramm wieder, das die Leistungsschütze Q1 und Q2 sowie die Hilfsschütze K1, K2 und K3 enthält.
3. Ermitteln Sie aus dem gegebenen Stromlaufplan nach den Umsetzungsregeln zum Ersetzen von Schützschaltungen den zugehörigen Funktionsplan.
4. Das Steuerungsprogramm soll in einem wieder verwendbaren Baustein realisiert werden. Zeichnen Sie den Bausteinaufruf im Funktionsplan und geben Sie die Deklarationstabelle an.
5. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.

Lernaufgabe 4.6: Zwei Zylinder-Ansteuerung mit Zeitfunktionen

Lös. S. 183

Der gegebene Pneumatikplan zeigt die Ansteuerung zweier Zylinder mit pneumatischer Ventilen.



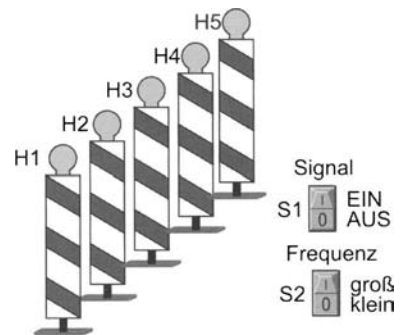
Die Betätigung der beiden 5/2-Wegeventile 1V2 und 2V2 soll künftig beidseitig durch Elektromagnete erfolgen, damit ein Automatisierungsgerät die Steuerungsfunktion übernehmen kann. Das druckknopfbetätigte 3/2-Wegeventil 1S3 wird durch einen elektrischen Taster S1 ersetzt. Das Zeitventil 1V1 ist auf fünf Sekunden und das Zeitventil 2V1 auf sieben Sekunden eingestellt.

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge für das Automatisierungsgerät.
2. Ermitteln Sie aus dem Pneumatikplan das Funktionsdiagramm, welches die beiden Zylinder 1A bzw. 2A enthält.
3. Bestimmen Sie mit einer Schaltfolgetabelle die Bedingungen für das Setzen und Rücksetzen für die im Steuerungsprogramm erforderlichen zwei Zeitglieder und vier Speicherglieder.
4. Zeichnen Sie den sich aus der Schaltfolgetabelle ergebenden Funktionsplan.
5. Das Steuerungsprogramm soll in einem Baustein bibliotheksfähig realisiert werden. Überlegen Sie, welche Art von Baustein Sie für das Steuerungsprogramm verwenden und geben Sie die Deklarationstabelle für den Baustein an.
6. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.

Lernaufgabe 4.7: Verkehrs-Lauflichtanlage

Lös. S. 185

Verkehrs-Lauflichtanlagen dienen zur Absicherung von gefährlichen Engstellen oder Baustellen. Das führende Licht weist den Verkehrsteilnehmer rechtzeitig auf die zu erwartende Richtungsänderung hin und entschärft den gefährlichen Verschwenkungs- oder Überleitungsbereich. Dazu leuchten die Warnlampen H1 bis H5 in der angegebenen Reihenfolge jeweils eine Taktzeit lange auf. Danach wiederholt sich der Vorgang solange die Anlage eingeschaltet ist.



Mit Schalter S1 wird das Lauflicht eingeschaltet. Die Stellung des Schalters S2 bestimmt die Frequenz des Lauflichts. Liefert S2 „0“-Signal, beträgt die Frequenz $f = 2$ Hz. Bei „1“-Signal beträgt die Frequenz $f = 0,5$ Hz. Die Ansteuerung der Lampen H1 bis H5 soll mit einer SPS erfolgen. Dazu ist das zugehörige Steuerungsprogramm zu entwerfen.

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Zeigen Sie, wie mit dem zweimaligen Aufruf des Taktbausteins FC 100 aus der Übersicht 4 bzw. dem Lehrbuch im Organisationsbaustein OB1, die beiden unterschiedlichen Frequenzen für das Lauflicht bzw. den entsprechende Baustein erzeugt werden können.
3. Bestimmen Sie mit einer Schaltfolgetabelle die Bedingungen für das Setzen und Rücksetzen der fünf Speicherglieder zur Ansteuerung der Lampen H1 bis H5. Verwenden Sie dabei die Methode des Binäruntersetzers.
4. Zeichnen Sie den sich aus der Schaltfolgetabelle ergebenden Funktionsplan.
5. Das Programm zur Ansteuerung der Lampen soll in einem Baustein bibliotheksfähig realisiert werden. Überlegen Sie, welche Art von Baustein Sie für das Steuerungsprogramm verwenden und geben Sie die Deklarationstabelle für den Baustein an.
6. Stellen Sie in einem freigrafischen Funktionsplan den Aufruf der Bausteine dar und geben Sie die erforderlichen Übergabeparameter an.

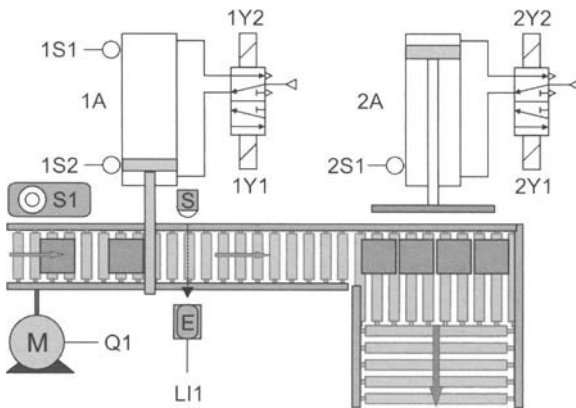
7. Programmieren Sie die Bausteine, rufen Sie diese vom OB1 entsprechend dem freigrafi-schen Funktionsplan auf und versehen Sie die Eingänge und Ausgänge der Bausteine mit SPS-Operanden aus der Zuordnungstabelle.

Lernaufgabe 4.8: Palettierungs-Anlage

Lös. S. 188

Am Ende einer Verpackungsstraße werden zur Palettierung jeweils vier Kartons zusammengefasst.

Technologieschema:



Nach Betätigung des Tasters S1 öffnet der Zylinder 1A die Sperre. Befindet sich der Zylinder in der hinteren Endlage, wird der Antriebsmotor M der Rollenbahn eingeschaltet. Die einzeln vorbeierollenden Kartons werden von der Lichtschranke LI1 erfasst. Haben 4 Kartons die Lichtschranke passiert, wird die Sperre mit Zylinder 1A wieder geschlossen und der Antriebsmotor M ausgeschaltet. Fünf Sekunden nachdem Zylinder 1A sich wieder in der vorderen Endlage befindet, schiebt Zylinder 2A die Kartons auf eine bereitstehende Palette.

Befindet sich der Zylinder 2A wieder in seiner hinteren Endlage, kann der Vorgang durch Betätigung der Taste S1 wiederholt werden.

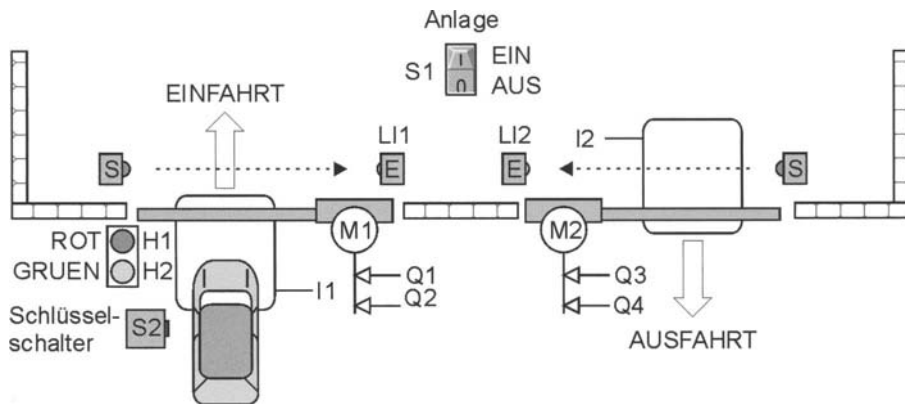
Die Ansteuerung der Elektromagnetventile und des Schützes Q1 für den Antriebsmotors M soll durch eine SPS erfolgen. Dazu ist in einem bibliotheksfähigen Baustein ein Steuerungsprogramm zu entwerfen.

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Zeichnen Sie das zur Aufgabenbeschreibung zugehörige Funktionsdiagramm, welches die beiden Zylinder 1A, 2A und den Schütz Q1 enthält.
3. Bestimmen Sie mit einer Schaltfolgetabelle die Bedingungen für das Setzen und Rücksetzen für die im Steuerungsprogramm erforderlichen Speicherglieder und das Zeitglied.
4. Zeichnen Sie den sich aus der Schaltfolgetabelle ergebenden Funktionsplan.
5. Das Steuerungsprogramm soll in einem Baustein bibliotheksfähig realisiert werden. Überlegen Sie, welche Art von Baustein Sie für das Steuerungsprogramm verwenden und geben Sie die Deklarationstabelle für den Baustein an.
6. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.

Lernaufgabe 4.9 Parkhauseinfahrt

Lös. S. 190

Die Ein- und Ausfahrt in ein Parkhaus mit 12 Stellplätzen wird durch Schranken freigegeben. Eine Ampel zeigt an, ob noch Stellplätze im Parkhaus frei sind und eine Einfahrt möglich ist. Vor den Schranken sind jeweils Induktionsschleifen angebracht, welche melden, ob ein Ein- oder Ausfahrwunsch besteht. An der Einfahrt ist zusätzlich ein Schlüsselschalter angebracht.

Technologieschema:**Funktionsbeschreibung:**

Einfahrt: Steht ein PKW auf der Einfahrtsinduktionsschleife I1, sind noch Stellplätze frei und wird der Schlüsselschalter S2 betätigt, öffnet sich die Einfahrtschranke (Schranke 1). Zum Öffnen der Schranke wird ein Motor M1 über Q1 im Rechtslauf für 5 Sekunden angesteuert. Nach Freiwerden der Induktionsschleife I1 schließt die Schranke 1 wieder. Dazu wird der Motor M1 über das Schütz Q2 im Linkslauf für 5 Sekunden angesteuert. Die Mechanik der Schranke beinhaltet eingebaute Endschalter, welche die Abschaltung des Antriebsmotors jeweils beim Erreichen der Endlagen steuern. Unterbricht ein einfahrendes Fahrzeug die Lichtschranke LI1, gibt diese einen Rückwärtszählimpuls an den Zähler für die Anzahl der freien Plätze.

Ausfahrt: Steht ein PKW auf der Ausfahrtlichtschranke I2 öffnet sich die Schranke 2. Das Öffnen und Schließen der Schranke 2 erfolgt durch den Motor M2 mit der Ansteuerung über Q3 (auf) und Q4 (zu) entsprechend der Schranke 1. Die Lichtschranke LI2 an der Ausfahrt gibt bei Unterbrechung einen Vorwärtszählimpuls an den Zähler.

Ampelsteuerung: Die Ampel wird vom Zähler für die freien Parkplätze angesteuert. Ist der Zählerstand 0, zeigt die Ampel „ROT“, ansonsten zeigt die Ampel „GRÜN“.

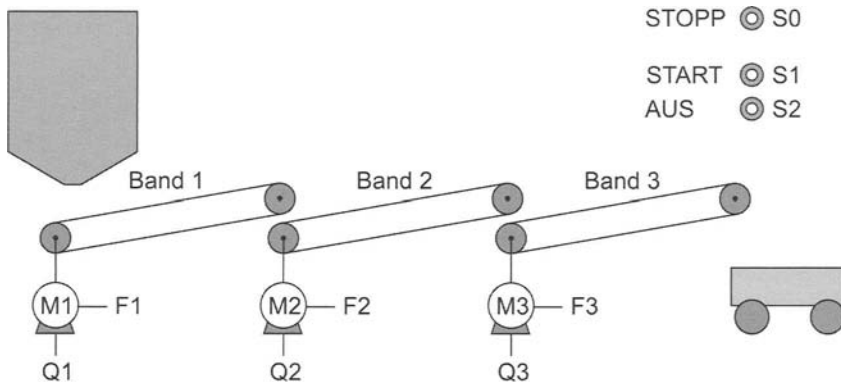
Anlagensteuerung: Mit dem Schalter S1 wird die Anlage eingeschaltet, der Zähler auf 12 (Anzahl der Parkplätze) gesetzt und die beiden Schranken geschlossen. Beim Ausschalten werden beide Schranken geöffnet und die Ampeln abgeschaltet.

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Ermitteln Sie mit einer RS-Tabelle die Bedingungen für das Setzen und Rücksetzen der im Steuerungsprogramm erforderlichen Speicher- und Zeitglieder.
3. Zeichnen Sie den sich aus der Schaltfolgetabelle ergebenden Funktionsplan. Ergänzen Sie den Funktionsplan durch den erforderlichen Zähler.
4. Das Steuerungsprogramm soll in einem Baustein bibliotheksfähig realisiert werden. Überlegen Sie, welche Art von Baustein Sie für das Steuerungsprogramm verwenden und geben Sie die Deklarationstabelle für den Baustein an.
5. Programmieren Sie den Baustein, rufen Sie diesen vom OBI aus auf und versehen Sie die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.

4.3 Kontrollaufgaben

Kontrollaufgabe 4.1

Drei Förderbänder einer Kiesanlage sollen mit einer SPS so angesteuert werden, dass nach Betätigung des EIN-Tasters S1 die Antriebsmotoren im Abstand von 5s in der Reihenfolge M3, M2 und M1 eingeschaltet werden. Bei Betätigung des AUS-Tasters S2 werden die Bänder in der umgekehrten Reihenfolge mit einem Abstand von 10s ausgeschaltet. Wird jedoch der STOPP-Taster S0 betätigt oder löst eines der Überstromrelais F1, F2 oder F3 aus, werden die Antriebsmotoren der Bänder sofort ausgeschaltet.



1. Geben Sie die Zuordnungstabelle mit allen Ein- und Ausgängen an.
2. Bestimmen Sie mit einer um die Zeitfunktionen erweiterten RS-Tabelle die Bedingungen für die Ansteuerung der Setz- und Rücksetz-Eingänge der erforderlichen Speicher- und Zeitglieder.
3. Ermitteln Sie aus der erweiterten RS-Tabelle den Funktionsplan.
4. Die Steuerungsfunktion soll mit einem Baustein bibliotheksfähig realisiert werden. Geben Sie die Deklarationstabelle des Bausteins an.
5. Zeichnen Sie den Funktionsplan des Bausteinaufrufs im OB1.

Kontrollaufgabe 4.2

Die Ansteuerung der nebenstehenden Reklametafel soll mit einer SPS ausgeführt werden. Nach dem Einschalten der Anlage mit dem Schalter S1, soll sofort die erste Reklameleuchte H1 leuchten. Im Takt von jeweils 3 Sekunden sollen dann die Leuchten H2 und danach die Leuchte H3 angesteuert werden.



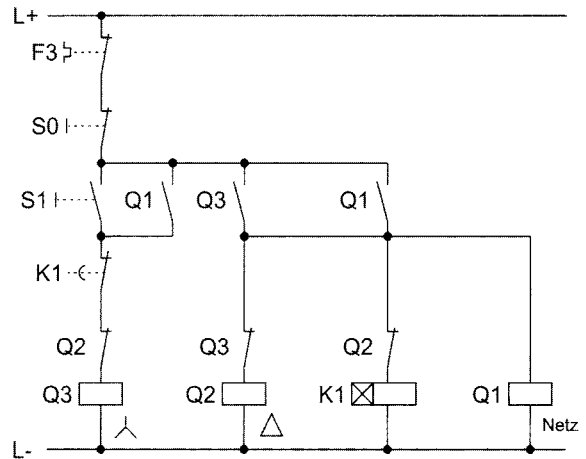
Nachdem alle drei Leuchten einen weiteren Takt leuchten, werden diese für wiederum eine Taktzeit ausgeschaltet. Danach wiederholt sich der Vorgang ständig. Wird der Anlagenbetrieb mit S1 beendet, werden alle Reklameleuchten sofort ausgeschaltet.

1. Geben Sie die Zuordnungstabelle mit den erforderlichen Ein- und Ausgängen an.
2. Bestimmen Sie mit einer Schaltfolgetabelle die Bedingungen für die Ansteuerung der Setz- und Rücksetz-Eingänge der benötigten Speicher- und Zeitglieder.

3. Ermitteln Sie aus der erweiterten Schaltfolgetabelle den Funktionsplan.
4. Die Steuerungsfunktion soll mit einem Baustein bibliotheksfähig realisiert werden. Geben Sie die Deklarationstabelle des Bausteins an.
5. Geben Sie den Funktionsplan des Bausteinaufrufs im OB1 an.

Kontrollaufgabe 4.3

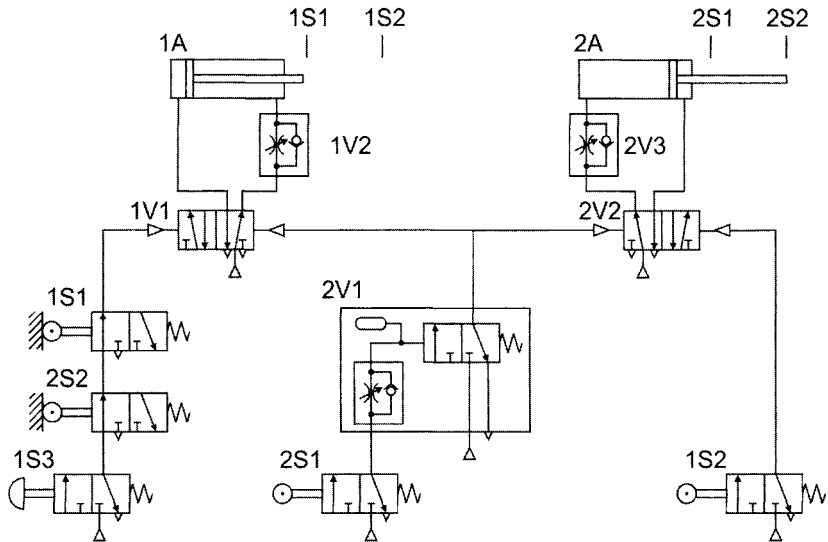
Der nebenstehende Stromlaufplan zeigt die Ansteuerung der drei Leistungsschütze Q1, Q2 und Q3 für einen automatischen Stern-Dreieck-Anlauf.



1. Zeichnen Sie den Hauptstromkreis der Stern-Dreieck-Schaltung.
2. Die Ansteuerung der Leistungsschütze Q1, Q2 und Q3 soll durch eine SPS erfolgen. Zeichnen Sie den Anschlussplan der SPS.
3. Wandeln Sie den gegebenen Stromlaufplan nach den entsprechenden Umsetzungsregeln in einen Funktionsplan um.
4. Der gefundene Funktionsplan soll mit einem bibliotheksfähigen Baustein realisiert werden. Geben Sie die Deklarationstabelle für den Baustein an.
5. Bestimmen Sie die Anweisungsliste AWL des Bausteins.
6. Es ist ein Neuentwurf des Programms mit RS-Speicherfunktionen durchzuführen. Ermitteln Sie mit einer RS-Tabelle die Bedingungen für die Ansteuerung von Setz- und Rücksetz-Eingang der erforderlichen Speicher- und Zeitfunktionen.
7. Zeichnen Sie aus der RS-Tabelle den Funktionsplan.
8. Der gefundene Funktionsplan soll mit einem bibliotheksfähigen Baustein realisiert werden. Geben Sie die Deklarationstabelle für den Baustein an.
9. Bestimmen Sie die Anweisungsliste AWL des Bausteins.

Kontrollaufgabe 4.4

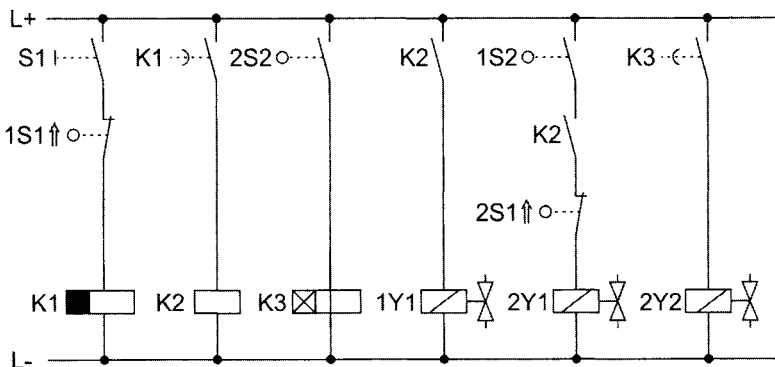
Der gegebene Pneumatikplan zeigt die Ansteuerung von zwei Zylindern. Die Ansteuerung der Zylinder soll künftig über 5/2-Wegeventile mit beidseitig elektromagnetischer Betätigung durch eine SPS erfolgen. Die Endlagengeber werden durch induktive Sensoren an den Zylindern und das handbetätigte 3/2-Wegeventil 1S3 durch einen Taster S1 ersetzt.



1. Ermitteln Sie mit einem Funktionsdiagramm in welcher Reihenfolge nach Betätigung des Tastventils 1S3 die Zylinder aus- und einfahren.
2. Zeichnen Sie den Pneumatikplan für die Ansteuerung mit einer SPS.
3. Bestimmen Sie mit einer Schaltfolgetabelle die Bedingungen für das Setzen und Rücksetzen der Magnetventile und des erforderlichen Zeitglieds.
4. Zeichnen Sie den sich aus der Schaltfolgetabelle ergebenden Funktionsplan.
5. Der Funktionsplan soll mit einem bibliotheksfähigen Baustein realisiert werden. Bestimmen Sie die Deklarationstabelle des Bausteins.
6. Geben Sie die Anweisungsliste des Bausteins an.

Kontrollaufgabe 4.5

Die nachfolgende Schützsteuerung dient zur Ansteuerung zweier Zylinder 1A und 2A über die 5/2-Wegeventile 1V1 und 2V1. Beide Zylinder fahren gedrosselt aus. Die eingestellte Zeit beträgt für das abfallverzögerte Relais 8 s und für das anzugsverzögerte Relais 5 s.



Die Ansteuerung der drei Elektromagnete 1Y1, 2Y1 und 2Y2 soll künftig über eine SPS erfolgen.

1. Zeichnen Sie den Pneumatikschaltplan für die Ansteuerung der Ventile mit einer SPS.

2. Bestimmen Sie aus dem Stromlaufplan das Funktionsdiagramm mit den beiden Zylindern.
3. Ermitteln Sie mit einer Schaltfolgetabelle die Bedingungen für die erforderlichen Zeiten und das Setzen und Rücksetzen der Magnetventile 1Y1, 2Y1 und 2Y2.
4. Zeichnen Sie den sich aus der Schaltfolgetabelle ergebenden Funktionsplan.
5. Der Funktionsplan soll mit einem bibliotheksfähigen Baustein realisiert werden. Bestimmen Sie die Deklarationstabelle des Bausteins.
6. Geben Sie die Anweisungsliste des Bausteins an.

Kontrollaufgabe 4.6

Das nachfolgend als AWL-Quelle angegebene Steuerungsprogramm eines Funktionsbausteins FB16 zur Behandlung einer Störungsmeldung ist zu analysieren.

AWL-Quelle:

```

FUNCTION_BLOCK FB 16
VAR_INPUT          VAR_IN_OUT          VAR          VAR_TEMP
STOE : BOOL;      HU : BOOL;           FO1 : BOOL;   Z_IMP : BOOL;
QUITT : BOOL;    HA : BOOL;           STOE_SP: BOOL; END_VAR
Zeitgl: TIMER;   END_VAR             T_IMP : BOOL;
Zeitw : S5TIME;  WERT0 : BOOL;
END_VAR          WERT1 : BOOL;
                 WERT2 : BOOL;
                 END_VAR

BEGIN
Störungsspeicher und Alarmhupe
U( ;
U #STOE;
S #STOE_SP;
U #QUITT;
R #STOE_SP;
U #STOE_SP;
) ;
FP #FO1;
S #HU;

Alarmquittierung
O #HA;
O #QUITT;
R #HU;

Taktgenerator
U #STOE_SP;
UN #T_IMP;
UN #QUITT;
L #Zeitw;
SE #Zeitgl;

Taktimpulse
U #Zeitgl;
= #T_IMP;
= #Z_IMP;

Taktimpulsauswertung Teil 1
U #Z_IMP;
UN #WERT1;
U #HU;
S #WERT1;
R #WERT0;
R #Z_IMP;

Taktimpulsauswertung Teil 2
U #Z_IMP;
UN #WERT2;
U #HU;
S #WERT2;
R #WERT0;
R #WERT1;
R #Z_IMP;

Taktimpulsauswertung Teil 3
U #Z_IMP;
UN #WERT0;
U #HU;
S #WERT0;
R #Z_IMP;

Taktimpulsauswertung Ende
U #Z_IMP;
U #WERT2;
U #HU;
O #QUITT;
R #WERT2;
R #WERT0;
R #WERT1;
R #Z_IMP;
S #HA;

Alarmleuchte AUS
U #Z_IMP;
U #HA;
O #QUITT;
R #HA;
R #Z_IMP;

Alarmleuchte EIN
U #Z_IMP;
UN #HA;
S #HA;

END_FUNCTION_BLOCK

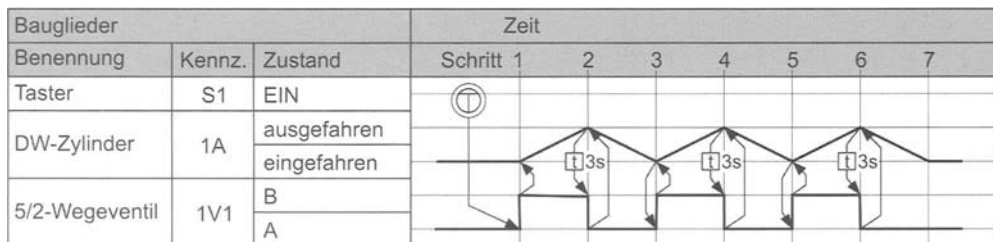
```

Beschreibung des Funktionsbausteins: Ein „1“-Signal am Eingang „STOE“ des Funktionsbausteins meldet eine neu aufgetretene Störung, die in den Störungsspeicher STOE_SP übernommen wird. Der Ausgang HU für eine Alarmhupe führt sofort „1“-Signal. Nach einer bestimmten Anzahl von Takten mit der am Eingang Zeit_W eingestellten Taktzeit wird der Ausgang HU ausgeschaltet und eine am Ausgang HA angeschlossene Alarmleuchte beginnt mit der Taktzeit zu blinken. Eine aufgetretene Störung muss in jedem Fall durch ein „1“-Signal am Eingang „QUITT“ quittiert werden. Alarmhupe und Alarmanzeige gehen bei der Quittierung sofort aus. Nach Betätigen der Quittierungstaste und noch nicht beseitigter Störung beginnt der Vorgang wieder mit der Ansteuerung der Alarmhupe.

1. Bestimmen Sie den Aufruf des Funktionsbausteins FB16 im OB1 in der Funktionsplandarstellung mit entsprechenden Globalparametern.
2. Übersetzen Sie das gegebene AWL-Programm in die Funktionsplandarstellung.
3. Zeichnen Sie den zeitlichen Verlauf der Taktimpulse T_IMP.
4. Nach wie vielen Takten wird von der Alarmhupe auf die Alarmleuchte umgeschaltet? Begründen Sie Ihre Antwort.
5. Die Zeit des Umschaltens von der Hupe auf die Leuchte soll verdoppelt werden. Welche Änderungen sind im Funktionsplan durchzuführen, wenn die Taktfrequenz unverändert bleiben soll?
6. Wie ändert sich das Verhalten des Funktionsbausteins, wenn im 1. Netzwerk die Programmzeilen 2+3 mit 3+4 in der Reihenfolge vertauscht werden.

Kontrollaufgabe 4.7

Das gegebene Funktionsdiagramm zeigt, dass der Zylinder 1A nach Betätigung des Tasters S1 dreimal verzögert ausfährt und für 3 Sekunden jeweils in der vorderen Endlage bleibt.



Die Ansteuerung des 5/2-Wegeventiles für den Zylinder 1A erfolgt beidseitig mit Elektromagneten. Zwei induktive Sensoren bestimmen die jeweilige Endlage des Zylinders.

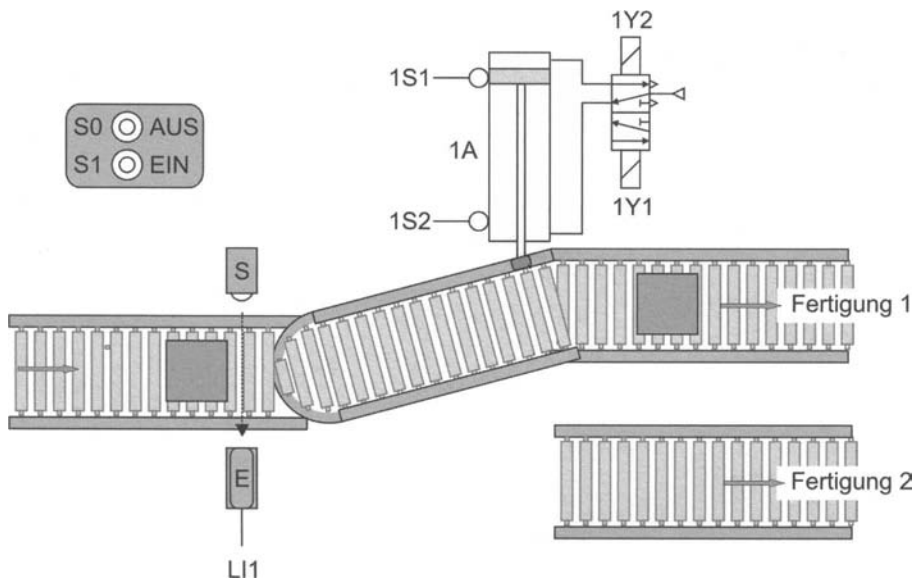
1. Zeichnen Sie den Pneumatikschaltplan.
2. Ermitteln Sie mit einer Schaltfolgetabelle die Bedingungen für das Setzen und Rücksetzen der Magnetventile 1Y1 und 1Y2 sowie für das Zeitglied und den Zähler.
3. Zeichnen Sie den sich aus der Schaltfolgetabelle ergebenden Funktionsplan.
4. Der Funktionsplan soll mit einem bibliotheksfähigen Baustein realisiert werden. Bestimmen Sie die Deklarationstabelle des Bausteins.
5. Geben Sie die Anweisungsliste des Bausteins an.

Kontrollaufgabe 4.8

In einer Produktionshalle werden Zurichtteile in Kisten zu zwei Fertigungsstationen auf einem Rollenband transportiert. Mit einer Bandweiche werden die Kisten auf die Stationen verteilt. Ist die Bandweiche nach Betätigung des Tasters S1 eingeschaltet, steuert die Weiche des Rollenbandes nach 7 Paketen die Kisten auf die jeweils andere Fertigungsinsel. Die Kisten werden durch die Lichtschranke LI1 erkannt. Durchläuft eine Kiste während des Umsteuerns der Weiche die Lichtschranke, so ist ihre Zählung der angesteuerten Fertigungsinsel zuzuordnen.

Beim Ausschalten der Bandweiche durch Taster S0 fährt Zylinder 1A ein und der Zähler wird auf den Wert 0 gesetzt. Die Ansteuerung des Zylinders erfolgt mit einem 5/2-Wegeventil, das beidseitig elektromagnetisch betätigt wird. Die Endlagen des Zylinders und somit die Stellung der Bandweiche werden mit den Endlagengebern 1S1 und 1S2 erfasst.

Technologieschema:

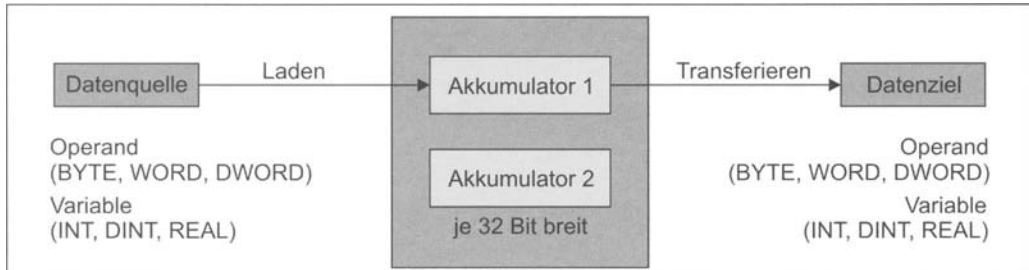


1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Zeichnen Sie den Pneumatikschaltplan, wenn sowohl die Einfahr- wie auch die Ausfahr- geschwindigkeit einstellbar sein soll.
3. Bestimmen Sie mit einer Schaltfolgetabelle die Bedingungen für das Setzen und Rücksetzen der erforderlichen Speicherglieder und den Zähler.
4. Zeichnen Sie den sich aus der Schaltfolgetabelle ergebenden Funktionsplan.
5. Das Steuerungsprogramm soll in einem Baustein bibliotheksfähig realisiert werden. Überlegen Sie, welche Art von Baustein Sie für das Steuerungsprogramm verwenden und geben Sie die Deklarationstabelle für den Baustein an.
6. Schreiben Sie die AWL des Bausteins.

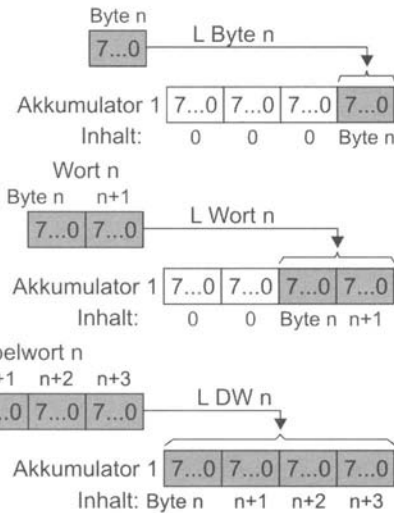
5 Übertragungsfunktionen, Vergleichsfunktionen, Sprünge

Übertragungsfunktionen in der AWL-Sprache

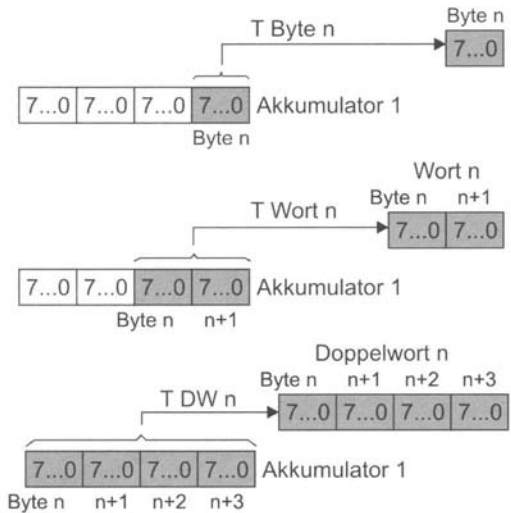
Übertragungsfunktionen ermöglichen das *Kopieren* von Digitalvariablen und Digitaloperanden. Nach der Richtung des Datenflusses bezüglich des beteiligten Akku 1 unterscheidet man *Laden* und *Transferieren*. Lade- und Transferbefehle sind unbedingte Befehle, die unabhängig vom Verknüpfungsergebnis (VKE) ausgeführt werden und dieses auch nicht beeinflussen.



Rechtsbündige Datenablage in Akku 1:



Rechtsbündige Datenentnahme aus Akku 1:



| Laden von | Anweisung | Kommentar | Transfer zu | Anweisung | Kommentar |
|-------------|-------------|-----------------|-------------|--------------|-----------------------------|
| Variablen | L #Sollwert | Mit Datentyp | Variablen | T #Stellwert | Mit Datentyp |
| Konstanten | L B#16#FF | Hex-Zahl | Globaldaten | T DB2.DBB0 | Datenbyte |
| | L W#16#1234 | Hex-BCD-Zahl | | T DB2.DBW4 | Datenwort |
| | L -500 | Integerzahl | | T DB2.DBD8 | Doppelwort |
| Eingängen | L EW 8 | Wort (Byte, ..) | Eingängen | T EB 8 | Wirkt nur auf Prozessabbild |
| Ausgängen | L AW 12 | Wort (Byte, ..) | | | |
| Globaldaten | L DB5.DBW0 | Datenwort, .. | Ausgängen | T AW 12 | Wort (Byte, ..) |
| Merkern | L MW 10 | Byte, Wort, .. | Merkern | T MW 100 | Wort (Byte, ..) |
| Peripherie | L PW 320 | Wort | Peripherie | T PW 336 | Wort |

Weitere Operandenbereiche: Zeiten und Zähler

Übertragungsfunktionen in der Programmiersprache FUP

Die MOVE-Box enthält die beiden Übertragungswege *Laden* und *Transferieren*. Sie kopiert die am Eingang IN anstehende Information in den Akku 1 und von dort zum Ausgang OUT.

| Box-Darstellung | Parameter | Datenbereiche | Datentypen |
|-----------------|-----------|---|---|
| | EN | Variablen, E, A, M, DBX, T, Z | BOOL |
| | IN | Variablen, Konstanten E, A, M, D, L, P | Alle außer BOOL 1, 2 oder 4 Byte-Operand |
| | OUT | Variablen E, A, M, D, L | Alle außer BOOL 1, 2 oder 4 Byte-Operand |
| | ENO | Variablen, E, A, M, DBX, EN | BOOL |

Regeln für den Gebrauch der MOVE-Box:

- Am Eingang IN und Ausgang OUT können Digitalvariablen mit elementaren Datentypen außer BOOL eingegeben werden. Die Variablen am Eingang IN und Ausgang OUT können unterschiedliche Datentypen aufweisen.
- An Eingang IN und Ausgang OUT können Digitaloperanden unterschiedlicher Operandenbreite 1, 2 oder 4 Byte eingegeben werden. Ist der Eingangsoperand breiter als der Ausgangsoperand, wird nur der rechtsliegende Teil soweit möglich übertragen.
- Mit Eingang EN kann die Abarbeitung der MOVE-BOX bedingt durchgeführt werden. Bei EN = 1 wird die an Eingang IN anliegende Bitkette zum Ausgang OUT übertragen. Bei EN = 0 wird der Ausgang OUT nicht bearbeitet.
- Eingang EN und Ausgang ENO müssen jedoch nicht beschaltet werden. Bei unbeschaltetem Eingang EN wird die Übertragungsfunktion der MOVE-Box ausgeführt.

Datentypenumwandlung mittels Übertragungsfunktionen

Die Lade- und Transferfunktionen ermöglichen zusätzlich zum Übertragen von Daten in einigen Fällen eine Datentypumwandlung ohne Verwendung echter Umwandlungsfunktionen.

Elementare Datentypen, die mit der MOVE-Funktion umwandelbar sind

- für Digitaloperanden: BYTE (Byte), WORD (Wort), DWORD (Doppelwort).
- für Digitalvariablen: INT (Integer).

Jeder Digitaldatentyp ist gekennzeichnet durch seine

- Datenlänge (Länge der Bitkette).
- Interpretation (Deutung der Bitkette).

Der Normalfall ist das Kopieren von Daten ohne Änderung des Datentyps

- Beispiel für Digitaloperanden:
 - L EW 8 //Datentyp: WORD
 - T AW 12 //Datentyp: WORD
- Beispiel für Digitalvariablen:
 - L #Sollwert //Datentyp: INT
 - T #Zwischenspeicher //Datentyp: INT

Der Spezialfall ist das Kopieren von Daten mit Änderung des Datentyps

- Beispiel für Digitaloperanden:
 - L EB 8 //Datentyp: BYTE
 - T AW 12 //Datentyp: WORD
- Beispiel für Digitalvariablen:
 - L #Zaehlwert_dual //Datentyp: WORD
 - T #Zaehlwert_int //Datentyp: INT

Die möglichen unechten Datentypumwandlungen sind im Prinzip Tricks des Programmierers:

| Beispiele | Erklärung |
|--------------|--|
| BYTE_TO_WORD | Gleiche Interpretation der Bitketten. Die Zahl im WORD-Format ist nur mit führenden Nullen aufgefüllt. |
| WORD_TO_BYTE | Gleiche Interpretation der Bitketten, jedoch mit Informationsverlust (1 Byte), der beabsichtigt sein kann. |
| WORD_TO_INT | Verschiedene Interpretation der Bitkette, aber richtiges Ergebnis: 1) Die Digitalvariable mit dem Datentyp WORD beinhaltet einen dual-codierten Zählerstand, der immer das Vorzeichenbit Bit 15 = 0 hat. Die Interpretation als Integerzahl bewirkt die Umwandlung einer vorzeichenlosen in eine positive Ganzzahl, z.B.: 31 in +31. 2) Ein Digitaloperand mit dem Datentyp WORD (z.B. PW 320) beinhaltet eine echte Integerzahl, die von einer Analogbaugruppe kommt. Die Interpretation als Integerzahl ist daher immer richtig. |

Vergleichsfunktionen für Integerzahlen

Mit Vergleichsfunktionen werden die Werte zweier Operanden des gleichen Datentyps verglichen. Das Ergebnis des Vergleichs beeinflusst das Verknüpfungsergebnis. Trifft die Vergleichsaussage zu, ist das VKE = 1 und kann mit Zuweisungen oder Sprungfunktionen ausgewertet werden.

| Vergleichsfunktion | AWL | FUP |
|------------------------------------|---|-----|
| Größer: > (Größer gleich: >=) | L Zahl_1 L Zahl_2 >I = Ergeb_1 | |
| Gleich: == (Ungleich: <>) | L Zahl_3 L Zahl_4 ==I = SPB M002 | |
| Kleiner: < (Kleiner gleich: <=) | L Zahl_5 L Zahl_6 <I = SPBN M003 | |

Sprungfunktionen (Auswahl)

| Operation | Operand | Bedeutung | FUP-Symbole | |
|-----------|---------|--|-------------|-------------------|
| SPA | Marke | Springe unbedingt zur Marke M001 | | |
| SPB | Marke | Springe bedingt bei VKE = „1“ zur Marke M002 | | Einsprung M003 |
| SPBN | Marke | Springe bedingt bei VKE = „0“ zur Marke M003 | | |

Bausteinfunktionen

BEA = Bausteinende absolut

BEB = Bausteinende bedingt

CALL FC (FB) = Funktion aufrufen

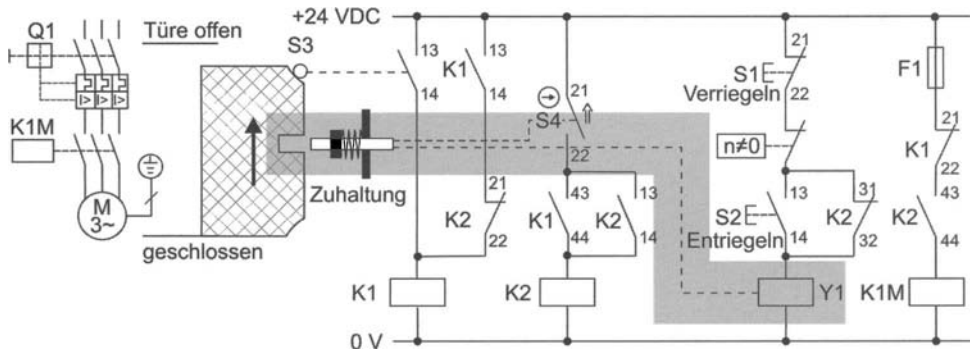
AUF DB = Global-Datenbaustein aufrufen

5.1 Beispiel
 • Lade- und Transferfunktionen, Vergleichsfunktionen, Sprünge

Drehzahl- und Stillstandsüberwachung einer mit Schutzgitter gesicherten Maschine

Der Zugang zu einer Maschine ist durch eine Schutzgittertüre zu sichern, die den Zutritt des Bedienpersonals vor Beendigung der gefährlichen Maschinenbewegung verhindern soll. Die Schutzgittertüre ist mit einer elektromechanischen Zuhaltung mit Meldekontakt S4 und einem weiteren Türkontakt S3 ausgerüstet. Ein Drehzahlwächter erkennt den Stillstand des Motors. Während des Betriebs soll eine Drehzahlüberwachung melden, ob die Drehzahl im zulässigen Bereich ist.

Technologieschema:



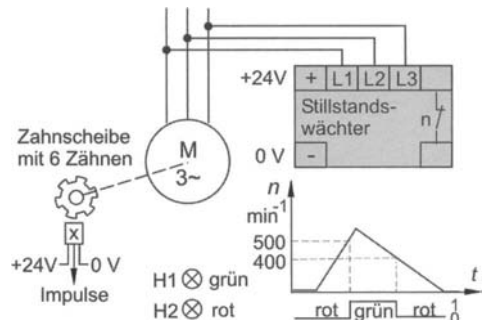
Funktionsweise der Schutztür-Zuhaltung

Die Verriegelung der Schutztüre ist mit einer Testung der Sicherheitsfunktion verbunden. Ausgehend von der Grundstellung (Motorschütz K1M ist stromlos, Türe ist zu, Betätigungsmagnet Y1 ist stromführend, Zuhaltung ist entriegelt, S4-Kontakt ist geöffnet) ist die Schutztüre einmal zu öffnen (Überprüfung des Türkontaktes S3, indem K1 anzieht) und wieder zu schließen (K1 hält sich selbst und bereitet K2 vor). Motorschütz K1M bleibt stromlos, da Kontakte K1 und K2 geöffnet sind. Durch Betätigen des Verriegelungstaster S1 wird Betätigungsmagnet Y1 stromlos, Zuhaltung rastet federmechanisch ein, Zuhaltekontakt S4 schließt (Überprüfung des S4-Kontaktes, indem K2 anzieht und K1 abwirft). Motorschütz K1M zieht über K1- und K2-Kontakt an. Nur in dieser Reihenfolge der Betätigung und bei funktionierender Schutzschaltung zieht K1M an! Zum Entriegeln des Schutzgitters muss der Motorstillstand erkannt sein (Stillstandskontakt n hat geschlossen) und der Entriegelungstaster S2 betätigt werden (Betätigungsmagnet Y1 zieht an, die Zuhaltung wird entriegelt, S4-Kontakt öffnet, K2 fällt ab, Motorschütz K1M wird stromlos) und die Schutztüre lässt sich öffnen.

Drehzahlüberwachung

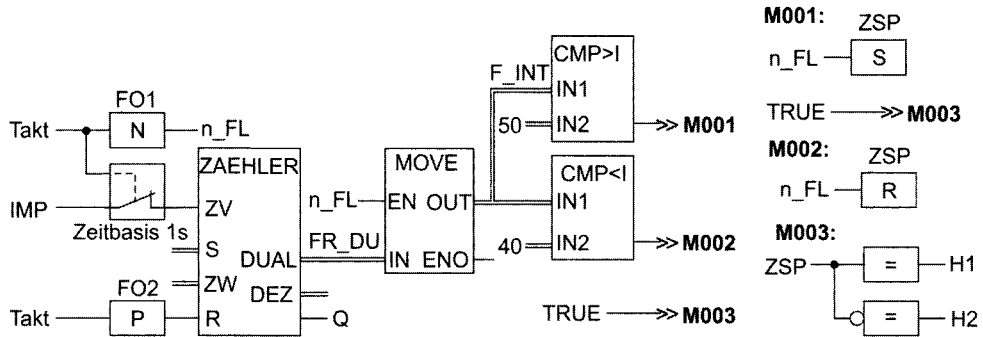
Die Motordrehzahl wird über eine angeflanschte Nockenscheibe mit sechs Zähnen durch einen induktiven Näherungsschalter erfasst, der Impulse der Frequenz $f = n/60 \cdot 6$ liefert, wobei n die Drehzahl in Umdrehungen pro Minute angibt (für die Simulation sind die Werte zu verringern).

Das Diagramm zeigt die Zuordnung der zu bildenden Drehzahlkontrollsignale Grün (H1) und Rot (H2) zum Drehzahlbereich n.



Aufgaben:

1. Für die Schutztürsteuerung ist ein Steuerungsprogramm aus der gegebenen Schutzsteuerung zu entwickeln.
2. Der gegebene freigrafische Funktionsplan für die Drehzahlüberwachung ist in ein Steuerungsprogramm umzusetzen.



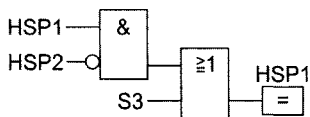
3. Die Bausteine sind zu programmieren und im OB1 aufzurufen. Die Eingänge und Ausgänge der Bausteine sind mit den SPS-Operanden zu versehen.

Zuordnungstabelle der Eingänge und Ausgänge:

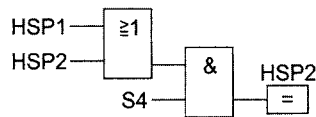
| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | | Adresse |
|---------------------|--------|----------|-------------------------|---------|---------|
| Verriegelungstaster | S1 | BOOL | Betätigt | S1 = 0 | E 0.1 |
| Entriegelungstaster | S2 | BOOL | Betätigt | S2 = 1 | E 0.2 |
| Türkontakt | S3 | BOOL | Betätigt | S3 = 1 | E 0.3 |
| Zuhaltekontakt | S4 | BOOL | Betätigt | S4 = 1 | E 0.4 |
| Stillstandswächter | n | BOOL | Motorstillstand | n = 1 | E 0.5 |
| Drehzahlimpulse | IMP | BOOL | 6 Impulse pro Umdrehung | | E 0.6 |
| Ausgangsvariable | | | | | |
| Betätigungsmagnet | Y1 | BOOL | Angezogen | Y1 = 1 | A 4.0 |
| Motorschütz | K1M | BOOL | Angezogen | K1M = 1 | A 4.1 |
| Meldeleuchte Grün | H1 | BOOL | Meldeleuchte an | H1 = 1 | A 4.2 |
| Meldeleuchte Rot | H2 | BOOL | Meldeleuchte an | H2 = 1 | A 4.3 |

1.a. Funktionsplan Schutztürsteuerung

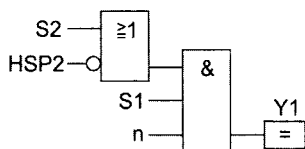
Programmiersatz für Hilfsschütz K1



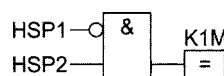
Programmiersatz für Hilfsschütz K2



Programmiersatz für Betätigungsmagnet Y1



Programmiersatz für Motorschütz

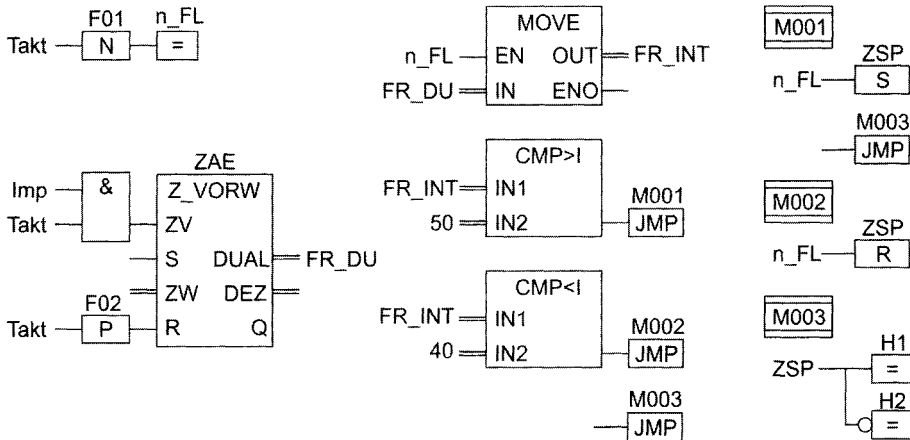


1.b. Deklarationstabelle FB 10:

| Name | Datentyp | Anfangswert |
|-----------|----------|-------------|
| IN | | |
| S1 | Bool | FALSE |
| S2 | Bool | FALSE |
| S3 | Bool | FALSE |
| S4 | Bool | FALSE |
| n | Bool | FALSE |

| Name | Datentyp | Anfangswert |
|-------------|----------|-------------|
| OUT | | |
| Y1 | Bool | FALSE |
| K1M | Bool | FALSE |
| STAT | | |
| HSP1 | Bool | FALSE |
| HSP2 | Bool | FALSE |

2.a. Funktionsplan Drehzahlüberwachung

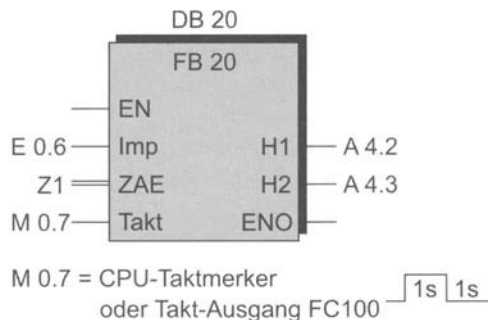
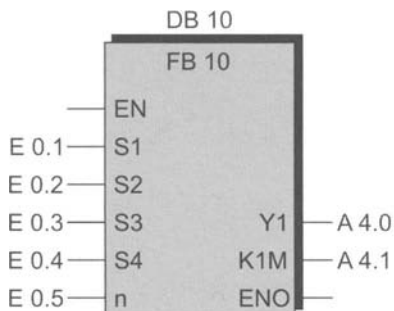


2.b. Deklarationstabelle FB 20:

| Name | Datentyp | Anfangswert |
|------------|----------|-------------|
| IN | | |
| Imp | Bool | FALSE |
| ZAE | Counter | |
| Takt | Bool | FALSE |
| OUT | | |
| H1 | Bool | FALSE |
| H1 | Bool | FALSE |

| Name | Datentyp | Anfangswert |
|-------------|----------|-------------|
| STAT | | |
| FR_INT | Int | 0 |
| ZSP | Bool | FALSE |
| F01 | Bool | FALSE |
| F02 | Bool | FALSE |
| TEMP | | |
| FR_DU | Word | |
| n_FL | Bool | |

3. Aufruf im OBI:



5.2

Lernaufgaben

Lernaufgabe 5.1 Auswahl-Standard-Funktion FC 36 SEL

Lös. S. 193

Es ist eine Funktion FC 36 für die eigene Programmbibliothek zu entwickeln, welche aus acht Eingangsvariablen IN0 bis IN7 mit dem Datenformat REAL, abhängig vom Eingangsparameters G, den Wert einer Variablen auswählt und der Ausgangsvariablen OUT zuweist.

Übergabevariablen:

G: (INTEGER) //Wert zwischen 0 und 7

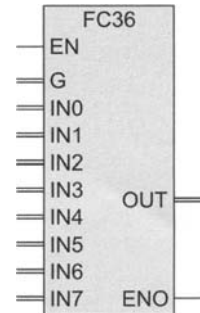
IN0: (REAL) //Eingabewert 0 IN1: (REAL) //Eingabewert 1

IN2: (REAL) //Eingabewert 2 IN3: (REAL) //Eingabewert 3

IN4: (REAL) //Eingabewert 4 IN5: (REAL) //Eingabewert 5

IN6: (REAL) //Eingabewert 6 IN7: (REAL) //Eingabewert 7

OUT: (REAL) //Ausgewählter Wert



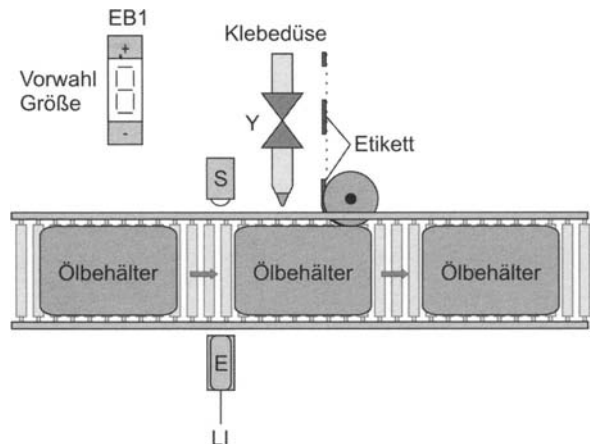
Zum Test der Funktion ist diese im OB1 aufzurufen und mit beliebigen Eingangswerten zu versehen. An den Eingang G kann ein einstelliger Zifferneinsteller gelegt werden.

1. Bestimmen Sie mit einem freigrafischen Funktionsplan die Programmstruktur der Funktion FC 36.
2. Geben Sie die Deklarationstabelle der Funktion an.
3. Ermitteln Sie aus dem freigrafischen Funktionsplan die Anweisungsliste AWL.
4. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus mit SPS-Operanden nach Ihrer Wahl auf und testen Sie die Funktion.

Lernaufgabe 5.2 Wählbare Öffnungszeit für eine Klebedüse

Lös. S. 194

Bei einer Abfüllanlage für Schmieröle werden je nach Inhalt unterschiedliche Etiketten auf die Ölbehälter geklebt. Es sind insgesamt 4 verschiedene Größen von Etiketten möglich. Zum Aufbringen des Klebstoffes wird eine Klebedüse mit Ventil Y für eine von der Größe des Etiketts abhängigen Zeit geöffnet. An einem Bedienpult wird mit einem einstelligen Zifferneinsteller EB1 durch Angabe einer Zahl von 1 bis 4 die Größe des Aufklebers voreingestellt. Gestartet wird der Klebevorgang, wenn ein Ölbehälter eine Lichtschranke LI passiert.

Technologieschema:

Die Öffnungszeiten der Klebedüse sind wie folgt vorgegeben:

Wird versehentlich am Zifferneinsteller die Ziffer 0 eingestellt, ist die Klebedüse mit der kürzesten Zeit zu öffnen. Bei einer Einstellung größer 4, wird die Klebedüse mit der längsten Zeit geöffnet.

| Größe | Zeit |
|-------|------|
| 1 | 5s |
| 2 | 7s |
| 3 | 9s |
| 4 | 11s |

Für die Ansteuerung des Ventils Y der Klebedüse ist ein bibliotheksfähiger Baustein zu entwerfen.

1. Geben Sie die Zuordnungstabelle der Eingänge und Ausgänge an.
2. Bestimmen Sie mit einem freigrafischen Funktionsplan die Programmstruktur des Bausteins.
3. Geben Sie die Deklarationstabelle für den Baustein an.
4. Ermitteln Sie aus dem freigrafischen Funktionsplan die Anweisungsliste AWL.
5. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.

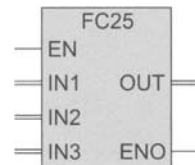
Lernaufgabe 5.3 Auswahl-Standard-Funktion FC 25 MAX

Lös. S. 195

Es ist eine Funktion FC 25 für die eigene Programmbibliothek zu entwerfen, welche aus drei Eingangsvariablen IN1 bis IN3 mit dem Datenformat REAL die größte auswählt und der Ausgangsvariablen OUT zuweist.

Übergabevariablen:

- IN1: (REAL) //Eingabewert 1
 IN2: (REAL) //Eingabewert 2
 IN3: (REAL) //Eingabewert 3
 OUT: (REAL) //Maximalwert



Zum Test der Funktion ist diese im OB1 aufzurufen und mit beliebigen Eingangswerten zu versehen.

1. Bestimmen Sie mit einem freigrafischen Funktionsplan die Programmstruktur der Funktion FC 25.
2. Geben Sie die Deklarationstabelle der Funktion an.
3. Ermitteln Sie aus dem freigrafischen Funktionsplan die Anweisungsliste AWL.
4. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus mit SPS-Operanden nach Ihrer Wahl auf und testen Sie die Funktion.

Lernaufgabe 5.4: AUF-AB-Zähler mit parametrierbaren Grenzen

Lös. S. 196

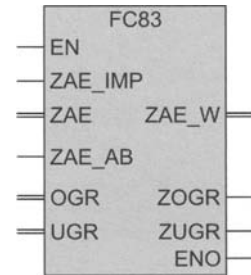
Zur Zählung und Kontrolle unterschiedlicher Mengen ist eine Standard-Funktion FC 83 zu entwerfen, die einen AUF-AB-Zähler mit parametrierbarer Ober- und Untergrenze abbildet.

Der Eingang ZAE_AB bestimmt die Zählrichtung. Liegt an diesem Eingang der Wert „TRUE“, wird abwärts gezählt. Die Grenzen des Zählwertes werden als Integer-Werte an den Eingänge OGR und UGR des Bausteins angegeben. Der Maximalwert für die Obergrenze beträgt OGRmax = 999 und der Minimalwert für die Untergrenze beträgt UGRmin = 0. Liegt beim Anlegen einer neuen Ober- oder Untergrenze der aktuelle Zählwert außerhalb des neuen

Bereichs, wird der entsprechende Grenzwert als Zählerstand übernommen. Die Ausgänge ZOGR bzw. ZUGR führen bei Erreichen der Ober- bzw. Untergrenze den Signalwert „TRUE“.

Übergabeparameter der Funktion FC 83:

ZAE_IMP: (BOOL) // Eingang für die Zählimpulse
 ZAE: (COUNTER) // S7-Zähler
 ZAE_AB: (BOOL) // Auswahl der Zählrichtung
 OGR: (INTEGER) // Obergrenze des Zählwertes
 UGR: (INTEGER) // Untergrenze des Zählwertes
 ZAE_W: (WORD) // Zählerstand
 ZOGR: (BOOL) // Zählerwert an der Obergrenze
 ZUGR: (BOOL) // Zählerwert an der Untergrenze



Zum Test der Funktion FC 83 werden Zählimpulse von einem Taktgeber mit der Frequenz 5 Hz an den Eingang ZAE_IMP gelegt. Mit dem binären Eingang ZAE_AB kann die Zählrichtung vorgegeben werden. Der aktuelle Zählerstand wird an einer BCD-Ziffernanzeige angezeigt und an die Indikatoren ZOGR bzw. ZUGR für das Erreichen der Ober- bzw. Untergrenze sind die Anzeigeleuchten H1 bzw. H2 zu legen.

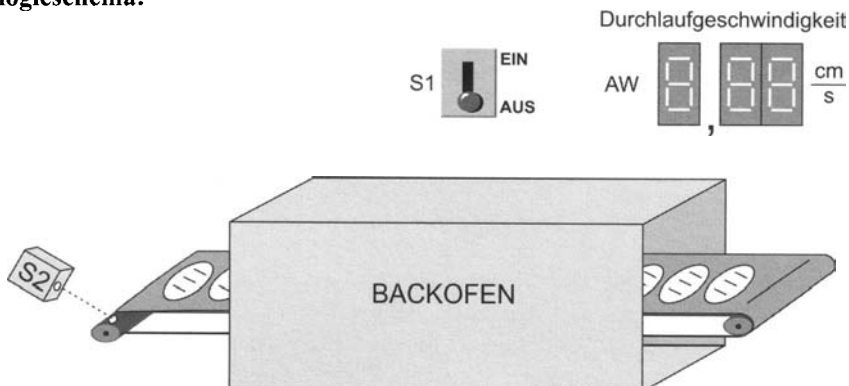
1. Geben Sie die Zuordnungstabelle der Eingänge und Ausgänge für den Test an.
2. Bestimmen Sie mit dem freigrafischen Funktionsplan die Programmstruktur der Funktion.
3. Geben Sie die Deklarationstabelle der Funktion an.
4. Ermitteln Sie aus dem freigrafischen Funktionsplan die Anweisungsliste AWL.
5. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie zum Test der Funktion die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.

Lernaufgabe 5.5: Anzeige der Durchlaufgeschwindigkeit

Lös. S. 197

In einer Großbäckerei wird das Backgut auf einem Förderband durch den Ofen geführt. An einer dreistelligen Siebensegment-Anzeige soll die Durchlaufgeschwindigkeit in cm/s angezeigt werden. Zur Ermittlung des Wertes werden die Drehimpulse des Bandantriebs, die Sensor S2 abgibt, gezählt.

Technologieschema:



Ein Taktgenerator mit der Frequenz 0,1 Hz und Puls-Pause-Verhältnis 1:1 gibt die Zeitbasis vor. Während der Pulszeit des Taktgenerators werden die Drehimpulse gezählt. Die gezählten Impulse werden nach Ende der Zählung jeweils an einer BCD-Anzeige angezeigt.

Pro Impuls legt das Backgut 0,1 cm auf dem Band zurück. Werden beispielsweise 150 Impulse angezeigt, entspricht dies einer Durchlaufgeschwindigkeit von $v = \frac{150 \cdot 0,1 \text{ cm}}{10 \text{ s}} = 1,5 \frac{\text{cm}}{\text{s}}$. Mit

dem Schalter S1 wird der Zählvorgang gestartet und die Anzeige aktiviert.

Für das Programm zur Ansteuerung der Anzeige der Durchlaufgeschwindigkeit soll ein bibliotheksfähiger Baustein entwickelt werden. Als Taktgenerator kann die Bibliotheksfunktion FC 100 aus dem Lehrbuch verwendet werden.

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Ermitteln Sie mit einem freigrafischen Funktionsplan die Programmstruktur des Bausteins für die Anzeige der Durchlaufgeschwindigkeit.
3. Geben Sie die Deklarationstabelle für den Baustein an.
4. Bestimmen Sie aus dem freigrafischen Funktionsplan die Anweisungsliste AWL.
5. Zeichnen Sie den Aufruf des Bausteins und der Bibliotheksfunktion FC 100 in der Funktionsplandarstellung mit den erforderlichen Übergabeparametern.
6. Programmieren Sie den Baustein, rufen Sie diesen und die Taktgenerator-Funktion FC 100 vom OB1 aus auf und versehen Sie die Eingänge und Ausgänge der Bausteine mit SPS-Operanden aus der Zuordnungstabelle bzw. temporären Variablen.

Lernaufgabe 5.6 Begrenzer mit Indikator

Lös. S. 198

Es ist ein bibliotheksfähiger Baustein LIMIT_IND (Begrenzer mit Indikator) zu entwerfen, der bei einer steigenden Flanke am Eingang CLK einen Integer-Eingangswert (IN) an den Ausgang (OUT) übergibt, wenn der Eingangswert den Minimalwert (MN) nicht unterschreitet und den Maximalwert (MX) nicht überschreitet. Unterschreitet der Eingangswert (IN) den Minimalwert (MN), wird der Minimalwert an den Ausgang OUT übergeben. Der Ausgang MN_IND zeigt dies mit einem „1“-Signal an. Überschreitet der Eingangswert den Maximalwert, wird der Maximalwert an den Ausgang übergeben. Der Ausgang MX_IND zeigt dies mit einem „1“-Signal an. Bei der erstmaligen Abarbeitung des Bausteins LIMIT_IND wird bis zur ersten auftretenden Flanke am Eingang CLK der Startwert PV an den Ausgang gegeben.

Zum Test des Bausteins ist dieser im OB1 aufzurufen und mit Eingangs- und Ausgangswerten zu versehen.

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge zum Test der Funktion.
2. Ermitteln Sie mit einem freigrafischen Funktionsplan die Programmstruktur des Bausteins.
3. Bestimmen Sie aus dem freigrafischen Funktionsplan die Anweisungsliste AWL.
4. Geben Sie die Deklarationstabelle für den Baustein an.
5. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.

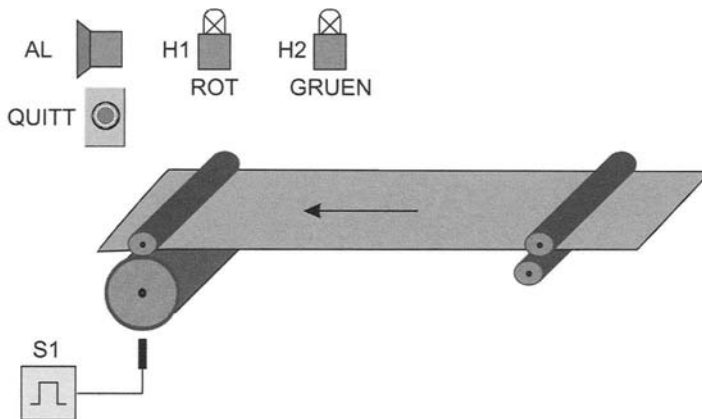
Lernaufgabe 5.7 Überwachung der Walzgutgeschwindigkeit

Lös. S. 200

Um den Qualitätsanforderungen der Kunden gerecht zu werden, wird in einer Walzerei die Walzprozessüberwachung unter anderem durch Messung der Geschwindigkeit des Walzgutes durchgeführt.

Dazu ist in einem komplexen Steuerungsprogramm für die gesamte Walzanlage ein Baustein zu schreiben, der überprüft, ob die Geschwindigkeit des Walzgutes in einem bestimmten Bereich liegt. Ist dies der Fall, wird die grüne Meldeleuchte H1 angesteuert. Liegt die Geschwindigkeit außerhalb des Bereichs, wird die rote Meldeleuchte H2 angesteuert. Bleibt die Geschwindigkeit über drei aufeinanderfolgenden Messperioden außerhalb des Bereichs, wird zusätzlich eine Alarmsirene AL aktiviert. Diese kann nur durch den Quittierungstaster QUITT gelöscht werden.

Zur Messung der Geschwindigkeit werden Impulse mit einem induktiven Näherungsschalter S1 erfasst. Pro Umdrehung entsteht ein Impuls. Die Messung wird mit einem Taktgenerator gesteuert. Der Taktgenerator liefert sechs Sekunden ein „1“-Signal und eine Sekunde ein „0“-Signal. Während des „1“-Signals wird gemessen. Die Geschwindigkeit des Bandes ist richtig, wenn die Anzahl der Impulse innerhalb der Messzeit von 6s zwischen 50 und 60 liegt.

Technologieschema:

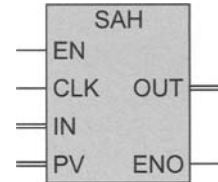
1. Geben Sie die Zuordnungstabelle der Eingänge und Ausgänge für den Test an.
2. Bestimmen Sie mit einem freigrafischen Funktionsplan die Programmstruktur des Codebausteins an.
3. Geben Sie die Deklarationstabelle der Funktion an.
4. Ermitteln Sie aus dem freigrafischen Funktionsplan die Anweisungsliste AWL.
5. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie zum Test der Funktion die Eingänge und Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.

Hinweis: Zum Test des Bausteins können die Impulse mit Bibliotheksbaustein „Takt“ (FC100) erzeugt werden.

5.3 Kontrollaufgaben

Kontrollaufgabe 5.1

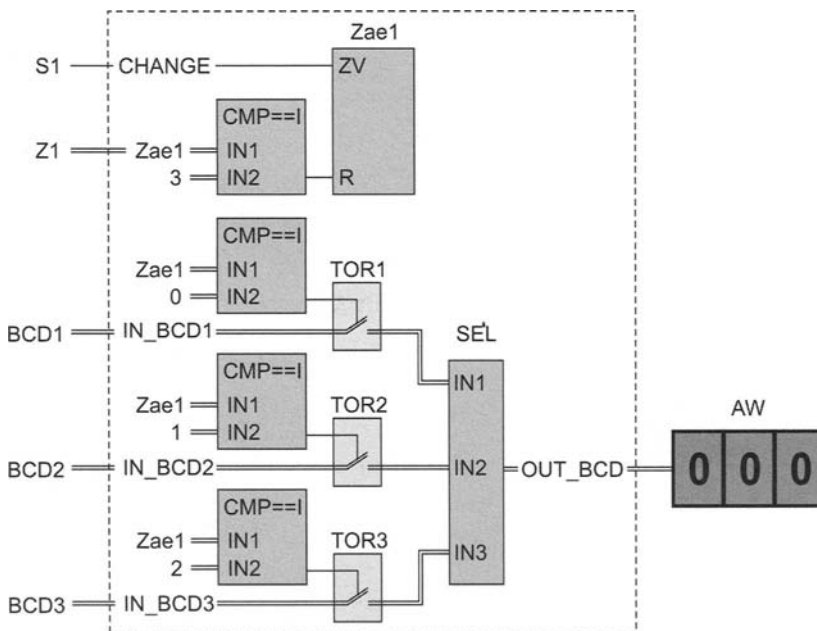
Der nebenstehende Baustein SAH (Select And Hold) übergibt bei einem 0→1-Übergang am Eingang CLK der Wert (Integer) von Eingang IN an den Ausgang OUT. Dieser Wert bleibt am Ausgang solange bestehen, bis mit dem nächsten 0→1-Übergang an CLK ein neuer Wert von IN nach OUT geladen wird. Beim Erststart des Bausteins, wenn noch kein 0→1-Übergang an CLK aufgetreten ist, wird der Startwert PV an den Ausgang OUT gelegt.



- Überlegen Sie, welche Bausteinart Sie zur Realisierung verwenden können und geben Sie die Deklarationstabelle des Bausteins an.
- Bestimmen Sie mit einem freigrafischen Funktionsplan die Struktur des Steuerungsprogramms.
- Schreiben Sie die zu dem Baustein gehörende Anweisungsliste AWL.

Kontrollaufgabe 5.2

Der gegebene freigrafische Funktionsplan zeigt, wie bei Betätigung des Tasters S1 jeweils ein bestimmter BCD-Wert an die Anzeige gelegt wird.



Das innerhalb der gestrichelten Linie dargestellte Steuerungsprogramm soll mit einem bibliotheksfähigen Baustein realisiert werden.

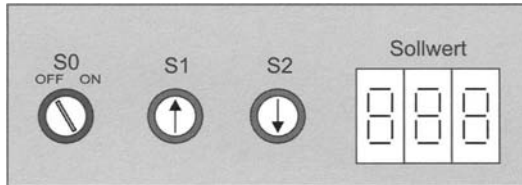
- Überlegen Sie, welche Bausteinart Sie zur Realisierung verwenden können und geben Sie die Deklarationstabelle des Bausteins an.
- Zeichnen Sie den Bausteinaufruf in Funktionsplandarstellung.

3. Ermitteln Sie die Anweisungsliste AWL des Bausteins.
4. Beschreiben Sie die Aufgabe des Bausteins.

Kontrollaufgabe 5.3

Zu einer stoßfreien Sollwertvorgabe sind zwei Taster und eine BCD-Anzeige auf einem Bedienfeld angebracht. Bei Betätigung des Tasters S1 (\uparrow) wird der Sollwert pro Sekunde um vier Werte erhöht. Bei Betätigung des Tasters S2 (\downarrow) entsprechend um 4 Werte pro Sekunde verkleinert. Der einstellbare Zahlenbereich des Sollwertes soll zwischen 0 und 999 liegen. Die Verstellung und der aktuelle Sollwert können an der dreistelligen Ziffernanzeige beobachtet werden. Mit dem Schlüsselschalter S0 wird die Freigabe der Verstellung des Sollwertes erteilt. Bei gleichzeitiger Betätigung von S1 und S2 bleibt der Sollwert stehen.

Bedienfeld:

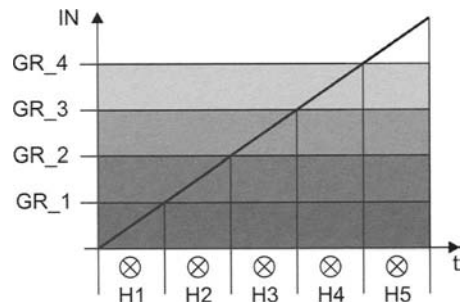


Für das Steuerungsprogramm des Software-Sollwertgebers ist ein bibliotheksfähiger Baustein zu entwerfen.

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Geben Sie mit einem freigrafischen Funktionsplan die Programmstruktur des Bausteins an.
3. Überlegen Sie, welche Bausteinart Sie zur Realisierung verwenden können und geben Sie die Deklarationstabelle des Bausteins an.
4. Ermitteln Sie aus dem freigrafischen Funktionsplan die Anweisungsliste AWL.
5. Zeichnen Sie den Bausteinaufruf in Funktionsplandarstellung.

Kontrollaufgabe 5.4

Es ist ein Programmbaustein zu entwerfen, der anzeigt, in welchem Bereich ein Integer-Wert IN liegt. Zur Anzeige werden fünf Meldeleuchten H1 bis H5 verwendet. Die vier Grenzen GR_1 bis GR_4 können am Baustein vorgegeben werden.



1. Bestimmen Sie mit einem freigrafischen Funktionsplan die Programmstruktur des Bausteins.
2. Überlegen Sie, welche Bausteinart Sie zur Realisierung verwenden können und geben Sie die Deklarationstabelle des Bausteins an.
3. Zeichnen Sie den Bausteinaufruf in Funktionsplandarstellung.
4. Ermitteln Sie die Anweisungsliste des Bausteins.

Kontrollaufgabe 5.5

Das nachfolgend als AWL-Quelle angegebene Steuerungsprogramm der Funktion FC 27 ist zu analysieren.

AWL-Quelle:

```

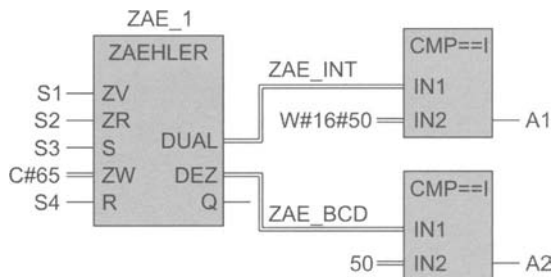
FUNCTION FC 27 : VOID
VAR_INPUT          VAR_IN_OUT
  IN1 : REAL ;      OUT : REAL ;
  IN2 : REAL ;      END_VAR
  IN3 : REAL ;
END_VAR
BEGIN
  L #IN1;            L #IN3;            L #IN3;
  L #IN2;            T #OUT;            T #OUT;
  <R ;              BEA ;                BEA ;
  SPB M001;          M002: L #IN2;        M003: L #IN1;
                               T #OUT;        T #OUT;
  L #IN2;            BEA ;
  L #IN3;            M001: L #IN1;
  <R ;              L #IN3;            END_FUNCTION
  SPB M002;          <R ;
                               SPB M003;

```

1. Bestimmen Sie den Aufruf der Funktion FC 27 im OB1 in der Funktionsplandarstellung. Geben Sie dabei sinnvolle SPS-Operanden nach Ihrer Wahl an.
2. Zeichnen Sie den zu der Anweisungsliste AWL gehörenden freigrafischen Funktionsplan.
3. Folgende Werte sind an die Eingänge gelegt: IN1: 2.4; IN2: 1.3; IN3: -5.1. Welcher Wert liegt nach Bearbeitung der Funktion am Ausgang OUT?
4. Beschreiben Sie die Funktion, die durch den Baustein FC 27 ausgeführt wird.

Kontrollaufgabe 5.6

Der nebenstehende freigrafische Funktionsplan gibt die Programmstruktur einer Funktion FC 16 wieder.



1. Bestimmen Sie den Aufruf der Funktion FC 16 im OB1 in der Funktionsplandarstellung.
2. Geben Sie die Deklarationstabelle des Bausteins an.
3. Ermitteln Sie die zum freigrafischen Funktionsplan gehörende Anweisungsliste AWL.
4. Bestimmen Sie den Eingang S1 bzw. S2 und die Anzahl der Rechteckimpulse an S1 bzw. S2, die nach einer positiven Flanke am Eingang S3 erforderlich sind, damit a) der Ausgang A1 und b) der Ausgang A2 „1“-Signal erhält.
5. Die Funktion FC 16 ist durch den Eingang OGR (INT) zu erweitern, der den maximal erreichbaren Zählerstand vorgibt. Ist dieser erreicht, kann nur noch abwärts gezählt werden. Zeichnen Sie den freigrafischen Funktionsplan für die neue Programmstruktur.

6 Lineare Ablaufsteuerungen ohne Betriebsartenteil

Steuerungsprogramme, die einen schrittweisen Prozessablauf in Anlagen nach den Vorgaben von *Ablauf-Funktionsplänen* ausführen, werden *Ablaufsteuerungen* genannt.

Funktionsplandarstellung der linearen Ablaufsteuerung

Ablaufkette und Übungsgangsbedingung:

Ein Ablauf-Funktionsplan beschreibt unabhängig von der technischen Realisierung den Steuerungsablauf in prozessspezifischen Einzelschritten. Für jeden Einzelschritt des Prozessablaufs ist ein Ablaufschritt im Funktionsplan vorzusehen.

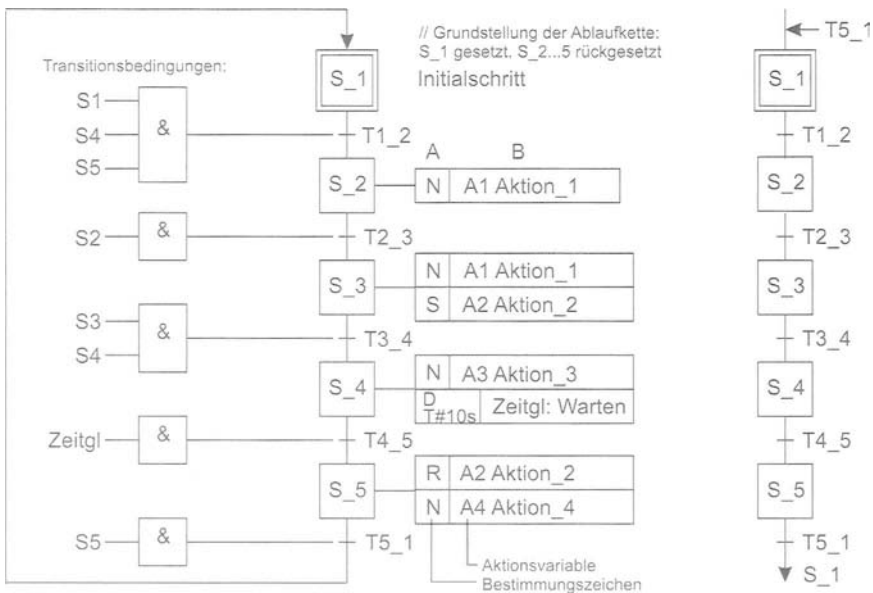
Schritte und Aktionen:

Schritte werden grafisch durch Blöcke dargestellt, die einen Schrittnamen enthalten. Der Anfangsschritt wird durch eine doppelte Umrahmung gekennzeichnet. Seitlich vom Schrittssymbol werden die Aktionsblöcke angebracht, die im Feld A das Bestimmungszeichen für die Aktionsart und im Feld B die Aktionsvariable mit kurzer Aktionsbeschreibung nennt.

Wirkungslinien und Transitionen:

Die Schrittssymbole sind durch Wirkungslinien zu verbinden; Übergänge (Transitionen) werden durch kurze Querstriche in den Wirkungslinien gekennzeichnet und mit Übergangsbedingungen (Transitionsbedingungen) für die Schrittweitschaltung versehen.

Beispiel: Funktionsplandarstellung einer linearen Ablaufsteuerung:

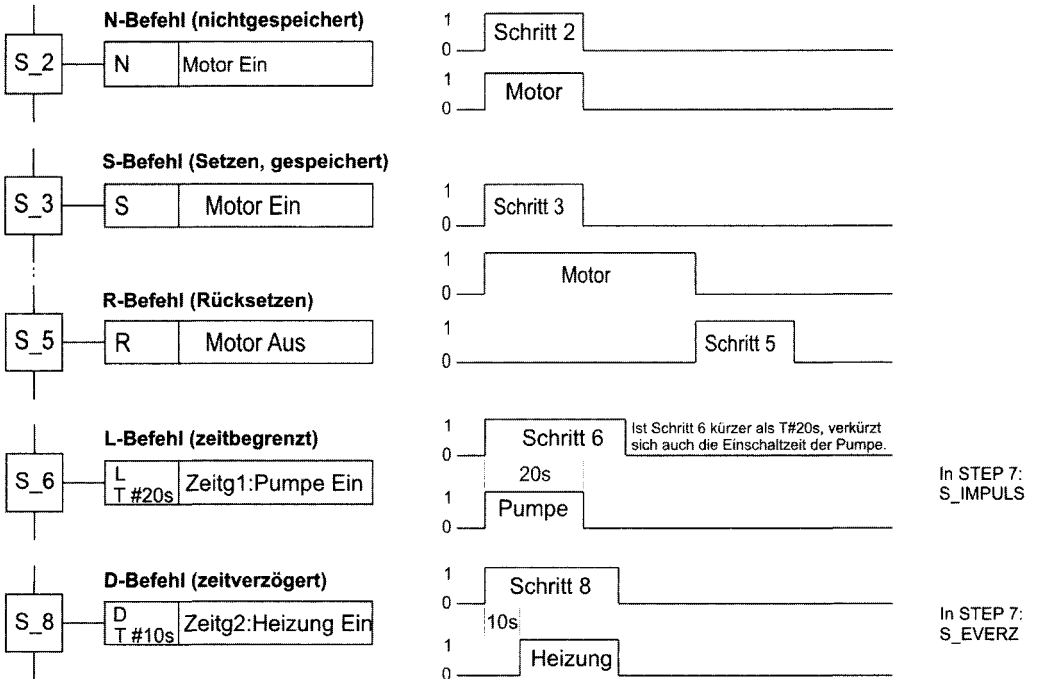


Transitionstabelle

| Transition | Transitionsbedingung |
|------------|----------------------|
| T-1 | T5_1 = S5 |
| T-2 | T1_2 = S1 & S4 & S5 |
| T-3 | T2_3 = S2 |
| | usw. |

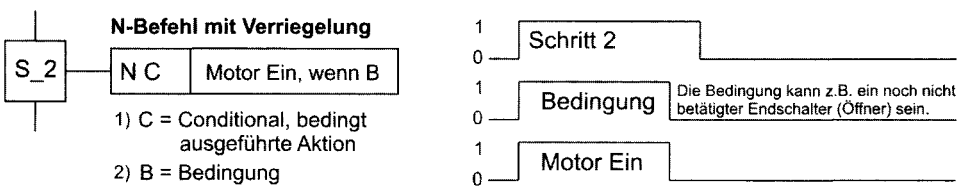
Hinweis: Bedeutung T-n
 Alle Transitionsbedingungen nach Schritt n.
 Bei verzweigten Ablaufketten können dies mehrere sein.

Bestimmungszeichen für Aktionen (DIN EN 61131-3)

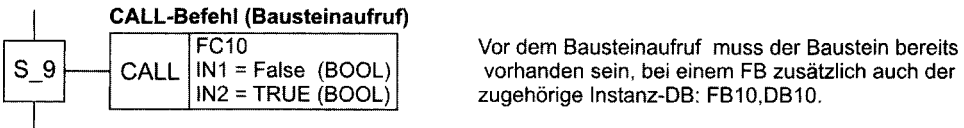


Weitere Bestimmungszeichen für Aktionen speziell in STEP 7

Die Verriegelungen einer Aktion innerhalb eines Ablaufschrittes wird durch Anhängen des Bestimmungszeichens C für bedingte Freigabe (C = Conditional) an das führende Bestimmungszeichen gekennzeichnet. Die Aktion wird im betreffenden Schritt nur ausgeführt, wenn die vereinbarte Verriegelungsbedingung B erfüllt ist. Das Bestimmungszeichen C ist in der DIN EN 61131-3 nicht vorgesehen, es ist aus der DIN 40719 Teil 6 entnommen und wird bei S7-GRAPH ergänzend zu den Bestimmungszeichen N, S, R, D, L verwendet. Das Umschalten in den nächsten Schritt erfolgt unabhängig von der Verriegelungsbedingung.



Ein Bausteinaufruf innerhalb eines Ablaufschrittes wird mit dem Bestimmungszeichen CALL gekennzeichnet. CALL ruft den angegebenen Baustein auf, dessen Programm wird ausgeführt und danach wird zum Baustein der Ablaufkette in den aktiven Schritt zurückgekehrt. Der Bausteinaufruf kann mit Parameterversorgung (bei FC- und FB-Bausteinen) erfolgen. CALL entspricht dem Aufruf eines Unterprogramms.

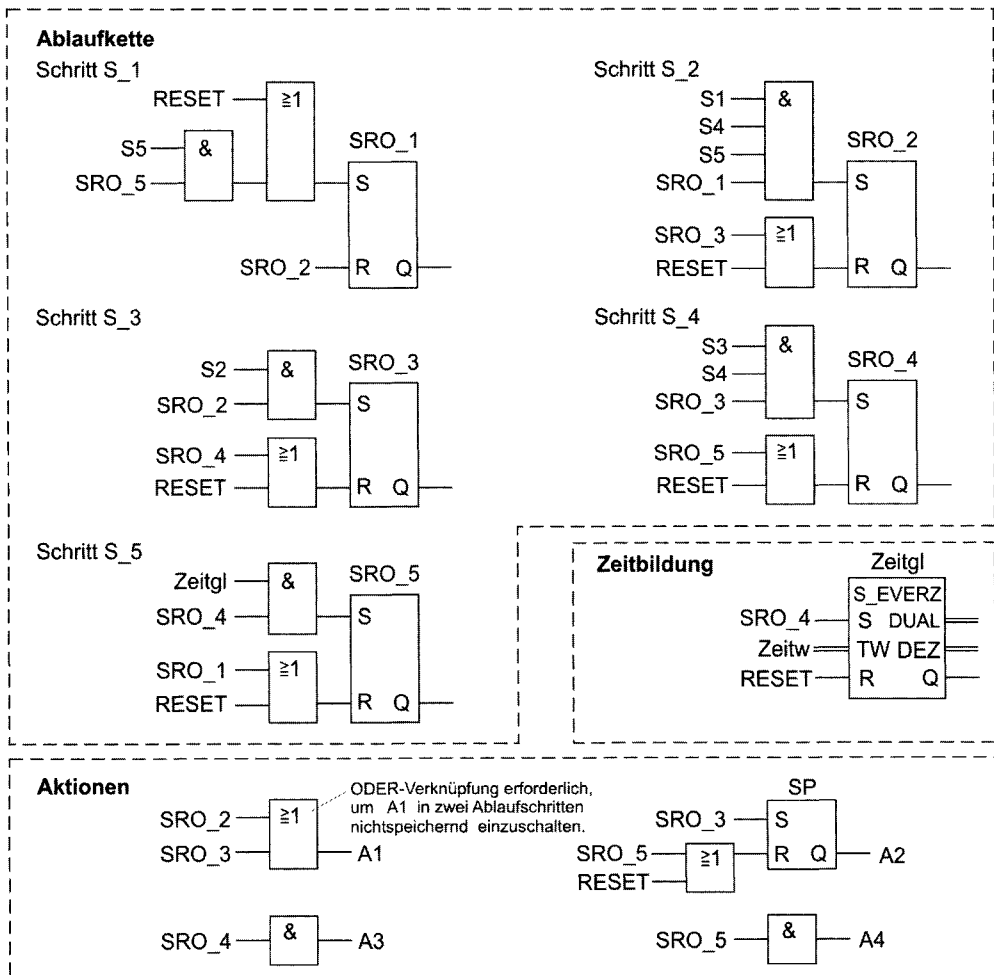


Programmierung von Ablaufsteuerungen

Realisierung mit einem anlagenspezifischen Funktionsbaustein:

- Für jeden Ablaufschritt ist ein RS-Speicherglied mit der Logik für die Transitionsbedingungen zum Setzen des Nachfolgespeichers vorzusehen. Über den Rücksetzeingang wird der Vorgängerspeicher von seinem Nachfolgerspeicher gelöscht.
- Die Grundstellung der Ablaufkette wird entweder durch den betriebsmäßigen Ablauf oder durch Ansteuerung mit einem RESET-Signal erreicht. RESET setzt den Speicher des Initialschrittes und setzt alle anderen Schrittspeicher zurück.
- Die Schrittspeicher steuern die Ausgänge (Aktoren) direkt an. Ist ein Ausgang von mehreren Schrittspeichern anzusteuern, so sind diese durch ODER zu verknüpfen.
- Erfordert der Ablauf Zeitbedingungen, sind entsprechende Zeitglieder vorzusehen.
- Der FB10 ist im OB1 aufzurufen und mit den E-/A-Operanden zu versehen.

Beispiel: Baustein mit Ablaufkette, Zeitbildung und Aktionen für eine lineare Ablaufsteuerung



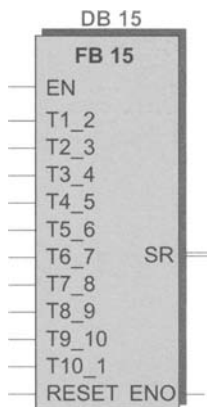
Realisierung mit Standard-Ablaufkettenbaustein FB 15 und Aktionsbaustein FC 16:

- Für lineare Ablaufketten kann wegen des gleichartigen Schrittaufbaus eine standardisierte Lösung verwendet werden. Der nachfolgend dargestellte Funktionsbaustein FB 15 enthält eine lineare Ablaufkette für zehn Schritte. Bei Aufruf des Bausteins im OB 1 müssen nur noch die Transitionsbedingungen angegeben werden.
- Bei Ansteuerung des RESET-Eingangs mit 1-Signal wird die Ablaufkette in die Grundstellung gesetzt.
- Der standardisierte Baustein kann keine Befehlsausgabe enthalten, da diese aufgabenabhängig ist. Der Bausteinausgang SR gibt jedoch die aktuelle Schrittnummer aus.
- Ein aufgabenabhängig programmierter Befehlsausgabebaustein FC 16 muss die aktuelle Schrittnummer des FB 15 auswerten und die Befehlsausgabe ausführen können.

1. Standardisierter Baustein mit linearer Ablaufkette und Schrittnummern-Ausgabe:

Schnittstellen des Funktionsbausteins:

Beschreibung der Übergabeparameter:



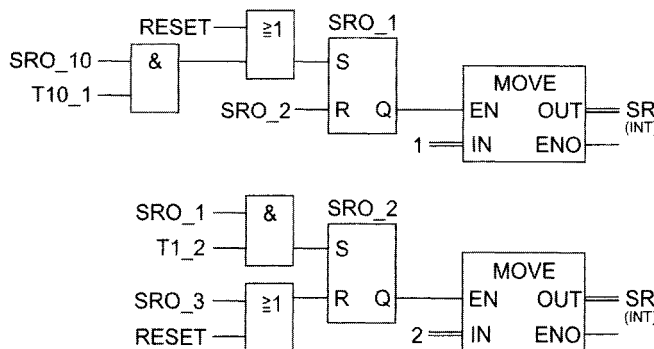
T1_2: Variable für die Annahme der Transitionsbedingung von Schritt 1 nach 2. In dieser Transitionsbedingung ist das Signal „Start“ mit auszuwerten.

T10_1: Variable für die Annahme der Transitionsbedingung von Schritt 10 nach 1.

RESET: Variable für die Annahme eines Signals, um die Ablaufkette in die Grundstellung zu setzen.

SR: Variable für die Ausgabe der aktuellen Schrittnummer (Integer)

Programmausschnitt FB 15: Schritt 1 und Schritt 2:

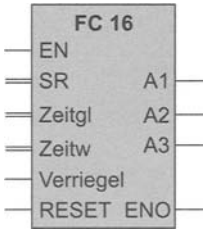


- a) SRO_1 = Schrittoperand für Schritt 1 (Speicher, lokale statische Variable)
- b) Die Programmteile für die Schritte 3 bis 10 sind entsprechend Schritt 2 aufgebaut.
- c) Die Eingangsparameter T1_2 bis T10_1 sind mit Anfangswert TRUE deklariert, um bei ungeschalteten Transitionseingängen ein automatisches Weiterschalten der Kette zu erzielen.

2. Aktionsbaustein FC 16 zur Ergänzung der standardisierten Ablaufkette:

Wird die Ablaufkette mit dem standardisierten Ablaufkettenbaustein FB 15 realisiert, müssen die Ausgabeaktionen außerhalb des FB 15 gebildet werden. Es fördert die Übersichtlichkeit, wenn alle Ausgabeaktionen in einem Befehlsausgabebaustein zusammengefasst werden.

Schnittstellen des FC-Bausteins:

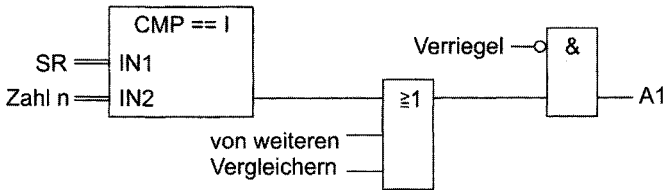


Beschreibung der Übergabeparameter:

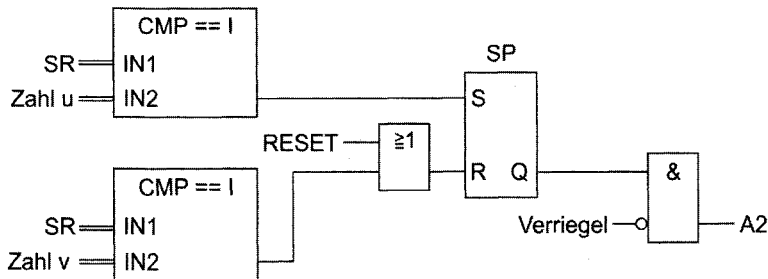
- SR: Variable für die Schrittnummer
- Zeitgl: Variable für einen Zeitgliedoperanden
- Zeitw: Variable für einen Zeitwert
- Verriegel: Variable für eine Ausgangsverriegelung
- A1...A3: Variablen für die Ausgangsoperanden
- RESET: Variable zum Löschen eines Speichers, eines Zeitgliedes oder Zählers

Grundregeln für das Erstellen des Befehlsausgabeprogramms (FC/FB 16):

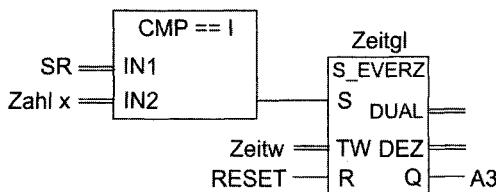
1. Jeder Teil des Befehlsausgabeprogramms beginnt mit der Auswertung der Schrittnummer durch einen Vergleich.
2. Ist das Bestimmungszeichen der Aktion ein "N", muss eine einfache Zuweisung programmiert werden. Wird ein Aktor von mehreren Ablaufschritten nichtspeichernd angesteuert, müssen die Einzelwertzuweisungen ODER-verknüpft werden.



3. Ist das Bestimmungszeichen der Aktion ein "S", muss eine SR-Speicherfunktion verwendet werden, die durch eine nachfolgende Aktion „R“ beendet wird.



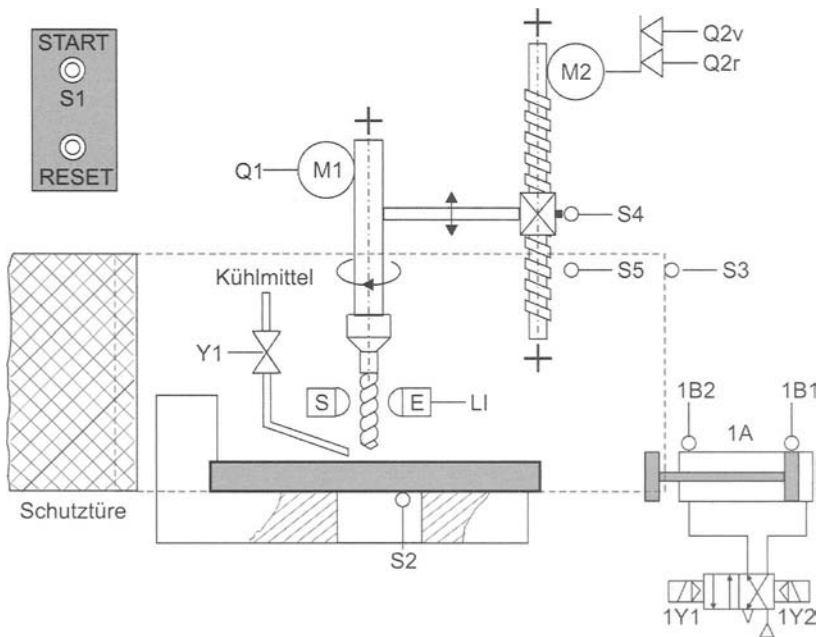
4. Ist das Bestimmungszeichen der Aktion ein "D", muss eine Einschaltverzögerung verwendet werden (IEC 61131-3: TON; STEP 7: S_EVERZ).



6.1

Beispiel**RS-Speicherglieder und Ablaufketten-Funktionsbaustein****Bohrmaschine**

Das Werkstück wird von Hand eingelegt und von Sensor S2 erkannt. Der Bearbeitungsvorgang lässt sich mit Taster S1 bei geschlossener Schutzgittertüre, die durch den Kontakt S3 überwacht wird, starten. Zuerst wird durch den pneumatischen Zylinder, der von einem Impulsventil angesteuert wird, das Werkstück gespannt und gleichzeitig der Bohrmotor M1 über Schütz Q1 eingeschaltet. Nach einer Wartezeit von 5 Sekunden für das Hochlaufen des Bohrmotors auf Nenndrehzahl wird über Schütz Q2v die Bohrspindel abgesenkt. Während des Bohrens wird ein Kühlmittel zugeführt und die Lichtschranke zur Bohrerbruchkontrolle ausgewertet. Der Vorschub des Bohrers wird beendet, wenn der untere Endpunkt durch Sensor S5 erkannt wird. Anschließend wird Schütz Q2r eingeschaltet und der Rückzug eingeleitet, bis Sensor S4 das Erreichen des oberen Endpunktes anzeigt. Danach wird der Bohrmotor ausgeschaltet. Bei Bruch des Bohrers oder Öffnen des Schutzgitters werden der Bohrmotor M1 und die Kühlmittelzufuhr abgeschaltet sowie der Bohrervorschub beendet.

Technologieschema:**Aufgaben:**

1. Entwerfen eines Ablauf-Funktionsplans mit zugehöriger Transitionstabelle.

Realisieren mit einem anlagenspezifischen Funktionsbaustein:

2. Zeichnen des Funktionsplans zur Umsetzung der Ablaufkette, der Zeitbildung und der Aktionsausgabe in einem Steuerungsprogramm.
3. Erforderliche Deklarationen für den anlagenspezifischen Funktionsbaustein ausführen.
4. Programmieren des Funktionsbausteins und Aufruf im OB 1. Bausteinein-/ausgänge mit den entsprechenden SPS-Operanden versehen.

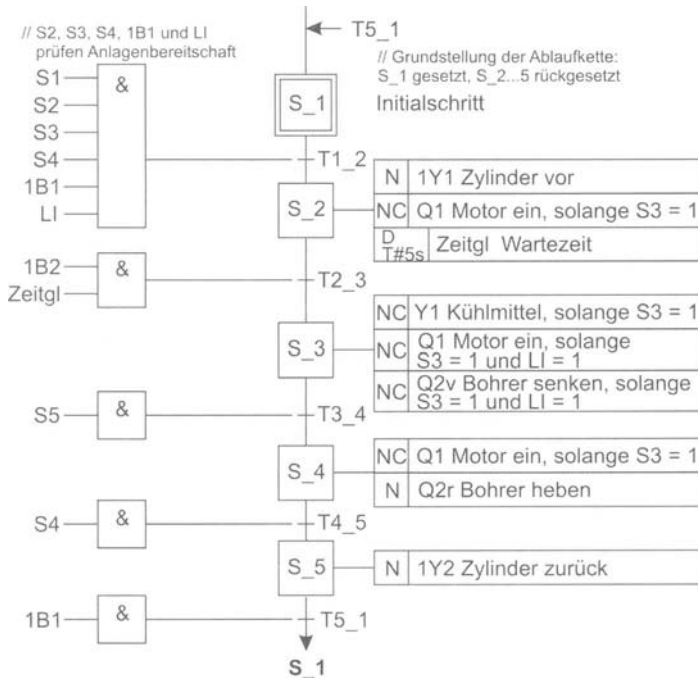
Realisieren mit dem Standard-Funktionsbaustein FB 15 und dem Aktionsbaustein FC 16:

5. Zeichnen des Funktionsplans für die Aktionsausgabe.
6. Angabe der erforderlichen Deklarationen für den Aktionsbaustein FC 16.
7. Programmieren der Bausteine FB 15 und FC 16 und Aufruf im OB 1. Bausteinein- und -ausgänge mit den SPS-Operanden und Übergabevariablen versehen.

Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | | Adresse |
|-------------------------------|--------|----------|--------------------|-----------|---------|
| Starttaste | S1 | BOOL | Betätigt | S1 = 1 | E 0.1 |
| Werkstück eingelegt | S2 | BOOL | Werkstück da | S2 = 1 | E 0.2 |
| Endschalter Schutztüre | S3 | BOOL | Türe geschlossen | S3 = 1 | E 0.3 |
| Endschalter Bohrer oben | S4 | BOOL | Endlage erreicht | S4 = 1 | E 0.4 |
| Endschalter Bohrer unter | S5 | BOOL | Endlage erreicht | S5 = 1 | E 0.5 |
| Hintere Endlage Zylinder | 1B1 | BOOL | Endlage erreicht | 1B1 = 1 | E 0.6 |
| Vordere Endlage Zylinder | 1B2 | BOOL | Endlage erreicht | 1B2 = 1 | E 0.7 |
| Lichtschranke | LI | BOOL | Bohrerbruch (frei) | LI = 0 | E 1.0 |
| Taster Ablaufk. Grundstellung | RESET | BOOL | Betätigt | RESET = 1 | E 0.0 |
| Ausgangsvariable | | | | | |
| Schütz Bohrmotor | Q1 | BOOL | Motor läuft | Q1 = 1 | A 4.0 |
| Schütz Vorschubmotor vor | Q2v | BOOL | Motor läuft | Q2v = 1 | A 4.1 |
| Schütz Vorschubmotor rück | Q2r | BOOL | Motor läuft | Q2r = 1 | A 4.2 |
| Magnetspule Zylinder 1A vor | 1Y1 | BOOL | Zylinder vor | 1Y1 = 1 | A 4.3 |
| Magnetspule Zylinder 1A rück | 1Y2 | BOOL | Zylinder zurück | 1Y2 = 1 | A 4.4 |
| Kühlmittelventil | Y1 | BOOL | Ventil auf | Y1 = 1 | A 4.5 |

1.a. Ablauf-Funktionsplan



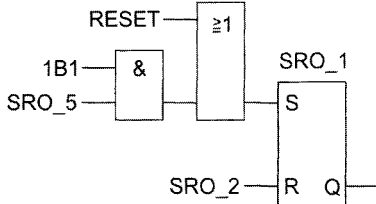
1.b. Transitionstabelle

| Transition | Transitionsbedingung |
|------------|--|
| T-1 | $T5_1 = 1B1$ |
| T-2 | $T1-2 = S1 \ \& \ S2 \ \& \ S3 \ \& \ S4 \ \& \ 1B1 \ \& \ LI$ |
| T-3 | $T2-3 = 1B2 \ \& \ Zeitgl$ |
| T-4 | $T3_4 = S5$ |
| T-5 | $T4_5 = S4$ |

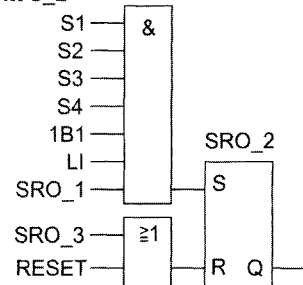
Realisieren mit einem anlagenspezifischen Funktionsbaustein

2. Funktionsplan des anlagenspezifischen Bausteins

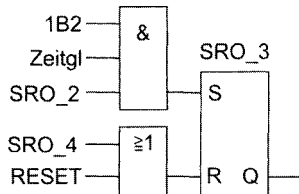
Schritt S_1



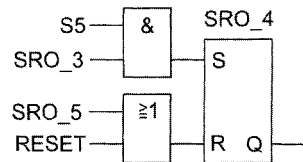
Schritt S_2



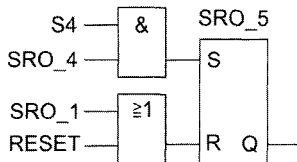
Schritt S_3



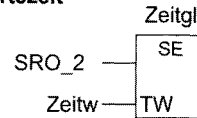
Schritt S_4



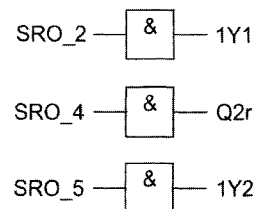
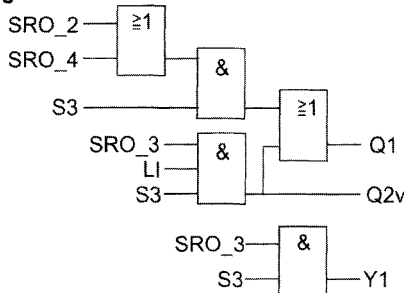
Schritt S_5



Wartezeit



Befehlsausgabe

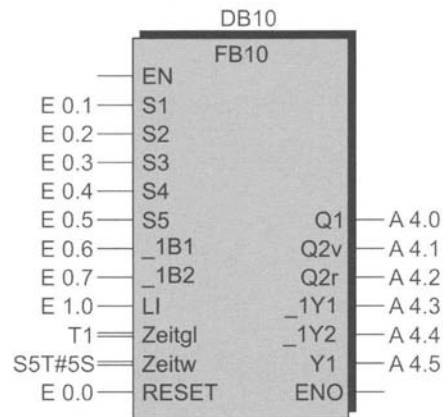


3. Deklarationstabelle für anlagenspezifischen Funktionsbaustein FB10

| Name | Datentyp | Anfangswert |
|------------|----------|-------------|
| IN | | |
| S1 | Bool | FALSE |
| S2 | Bool | FALSE |
| S3 | Bool | FALSE |
| S4 | Bool | FALSE |
| S5 | Bool | FALSE |
| _1B1 | Bool | FALSE |
| _1B2 | Bool | FALSE |
| LI | Bool | FALSE |
| Zeitgl | Timer | |
| Zeitw | S5Time | S5T#0MS |
| RESET | Bool | FALSE |
| OUT | | |
| Q1 | Bool | FALSE |
| Q2v | Bool | FALSE |
| Q2r | Bool | FALSE |
| _1Y1 | Bool | FALSE |
| _1Y2 | Bool | FALSE |
| Y1 | Bool | FALSE |

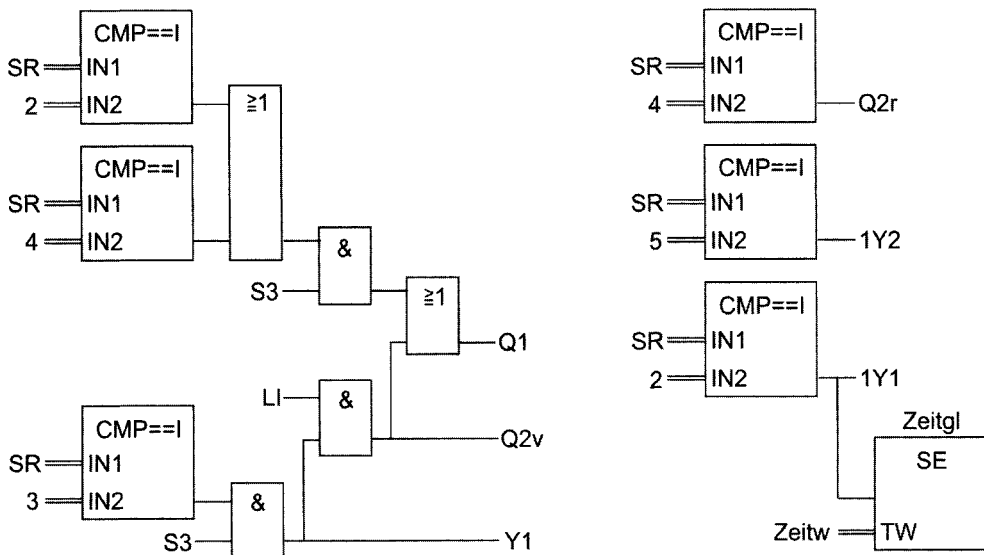
| Name | Datentyp | Anfangswert |
|-------------|----------|-------------|
| STAT | | |
| SRO_1 | Bool | TRUE |
| SRO_2 | Bool | FALSE |
| SRO_3 | Bool | FALSE |
| SRO_4 | Bool | FALSE |
| SRO_5 | Bool | FALSE |

4. Aufruf des FB10 im OBI:



Realisieren mit dem Standard-Funktionsbaustein FB15 und dem Aktionsbaustein FC16

5. Funktionsplan für Aktionsausgabe:



6. Deklarationstabelle für FC16:

| Name | Datentyp |
|-----------|----------|
| IN | |
| SR | Bool |
| Zeitgl | Timer |
| Zeitw | S5Time |
| S3 | Bool |
| LI | Bool |

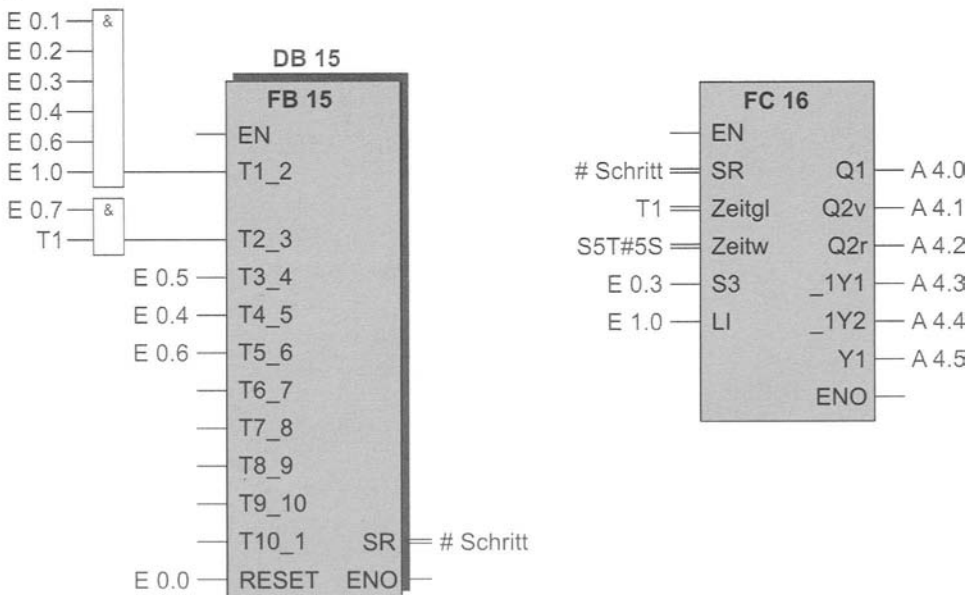
| Name | Datentyp |
|------------|----------|
| OUT | |
| Q1 | Bool |
| Q2v | Bool |
| Q2r | Bool |
| _1Y1 | Bool |
| _1Y2 | Bool |
| Y1 | Bool |

7. Bausteinaufrufe im OB1

Deklarationstabelle:

| Name | Datentyp | Anfangswert | Kommentar |
|------------------|----------|-------------|------------------|
| TEMP | | | |
| Standardeinträge | | | |
| Schritt | Int | | Übergabevariable |

Programm:



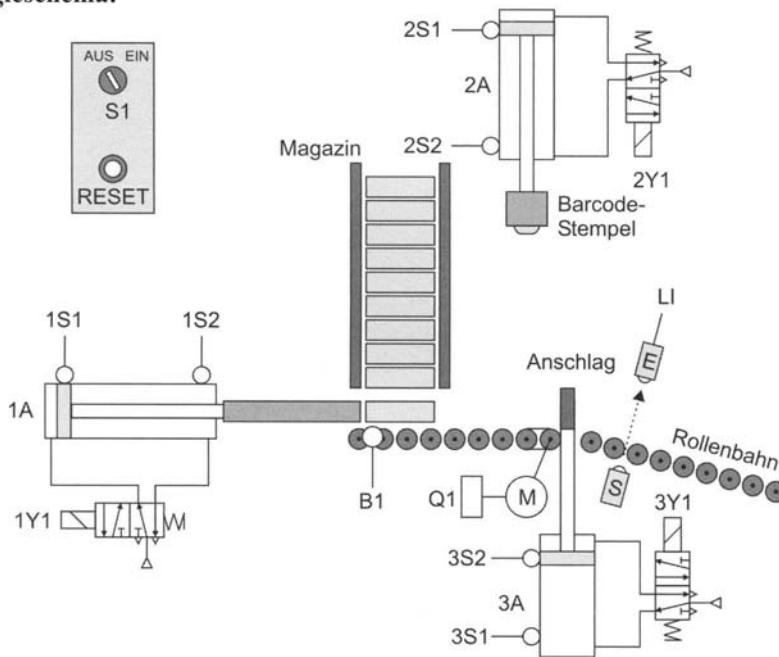
6.2 Lernaufgaben

Lernaufgabe 6.1: Bar-Code-Stempeleinrichtung

Lös. S. 202

Am Anfang der Fertigungsstrasse einer Konservenproduktion werden Schraubdeckel aus einem Magazin geschoben, mit einem Bar-Code versehen und über eine Rollenbahn dem weiteren Prozessablauf zugeführt.

Technologieschema:



Fertigungsablauf:

Mit dem Schlüsselschalter S1 wird der Ablauf freigegeben. Befinden sich alle Zylinder in der gezeichneten Lage und erkennt Sensor B1 einen Schraubdeckel aus dem Magazin und meldet die Lichtschranke LI freien Durchgang, dann schiebt Zylinder 1A den Schraubdeckel bis zum Anschlag vor. Danach fährt Zylinder 2A aus und bleibt zwei Sekunden in der vorderen Endlage. Dabei wird der Bar-Code aufgedruckt und danach Zylinder 2A wieder eingezogen. Meldet der Geber 2S1, dass Zylinder 2A die hintere Endlage erreicht hat, wird Motor M über das Schütz Q1 eingeschaltet und der Schraubdeckel zur schrägen Rollenbahn befördert. Erkennt die Lichtschranke das Passieren des Schraubdeckels, fährt der Anschlag mit Zylinder 3A wieder aus. Der Steuerungsablauf wiederholt sich dann ständig, bis mit dem Schlüsselschalter S1 wieder ausgeschaltet wird. Tritt während des Betriebs eine Störung auf, kann mit der RESET-Taste die Ablaufkette in die Grundstellung gebracht werden.

Da sich in der Produktionsanlage mehrere solcher Bar-Code-Stempeleinrichtungen befinden, soll das Steuerungsprogramm für eine Anlage in einen bibliotheksfähigen Baustein geschrieben werden, der alle Ein- und Ausgänge der Stempeleinrichtung besitzt.

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Stellen Sie den Prozessablauf mit einem Ablauf-Funktionsplan dar.

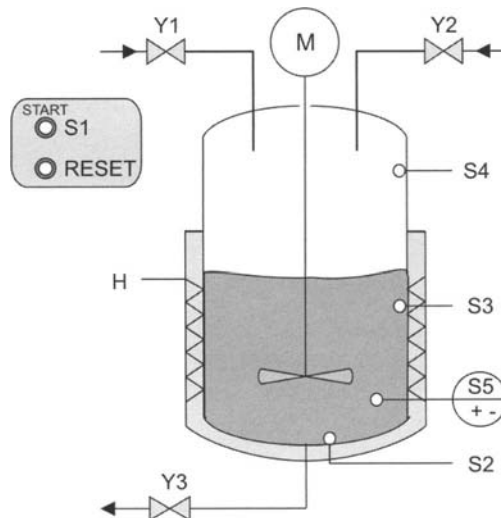
3. Ermitteln Sie aus dem Ablauf-Funktionsplan die Transitionsbedingungen T-1 bis T-n.
4. Zeichnen Sie den Funktionsplan für die Umsetzung der Ablaufkette, der Zeitbildung und der Aktionsausgabe in einem Funktionsbaustein (z.B. FB10).
5. Geben Sie die zur Programmierung des anlagenspezifischen Funktionsbausteins erforderlichen Deklarationen an.
6. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Ein-/ Ausgänge des Bausteins mit den SPS-Operanden der Zuordnungstabelle.

Lernaufgabe 6.2: Reaktionsprozess

Lös. S. 205

In einem Reaktionsbehälter werden zwei unterschiedliche chemische Ausgangsstoffe zusammengeführt, bis zu einer vorgegebenen Temperatur erwärmt und danach noch eine bestimmte Zeit gerührt.

Technologieschema:



Prozessablaufbeschreibung:

Nach Betätigung des Tasters S1 wird, sofern der Behälter leer und das Ventil Y3 geschlossen ist, das Vorlaufventil Y1 geöffnet, bis der Niveauschalters S3 („1“) anspricht. Danach schaltet das Rührwerk ein und das Ventil Y2 wird geöffnet. Spricht der Niveauschalter S4 („1“) an, schließt das Ventil Y2 wieder und die Heizung H schaltet ein. Meldet der Temperatursensor S5 („1“) das Erreichen der vorgegebenen Temperatur, wird die Heizung H abgeschaltet und die Mischzeit von zehn Sekunden gestartet. Nach Ablauf der Mischzeit schaltet das Rührwerk ab und das Ventil Y3 öffnet. Meldet der Niveauschalter S2 („1“), dass der Behälter leer ist, wird das Ventil Y3 wieder geschlossen und der Prozessablauf kann wiederholt werden.

Mit der RESET-Taste kann die Ablaufkette in die Grundstellung gebracht werden.

Das Steuerungsprogramm soll mit dem anlagenspezifischen Funktionsbaustein FB10 und mit dem Bibliotheks-Funktionsbaustein FB 15 sowie dem Aktionsbaustein FC 16 realisiert werden.

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Stellen Sie den Prozessablauf mit einem Ablauf-Funktionsplan dar.

3. Ermitteln Sie aus dem Ablauf-Funktionsplan die Transitionsbedingungen T-1 bis T-n.

Realisierung mit einem anlagenspezifischen Funktionsbaustein:

4. Zeichnen Sie den Funktionsplan für die Umsetzung der Ablaufkette, der Zeitbildung und der Aktionsausgabe.
5. Geben Sie die zur Programmierung des anlagenspezifischen Funktionsbausteins erforderlichen Deklarationen an.
6. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Ein-/ Ausgänge des Bausteins mit SPS-Operanden der Zuordnungstabelle.

Realisierung mit Standard-Funktionsbaustein FB15 und Aktionsbaustein FC16:

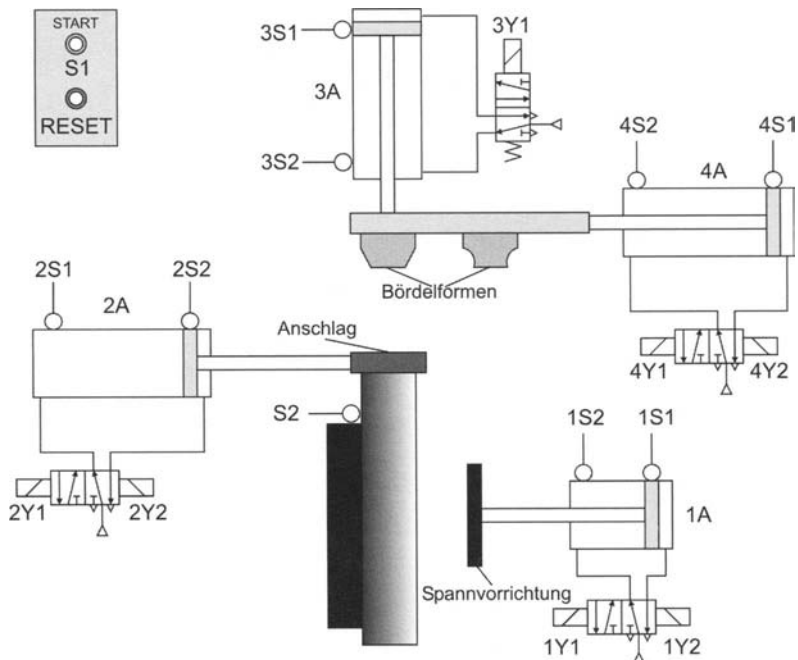
7. Zeichnen Sie den Funktionsplan für die Aktionsausgabe.
8. Geben Sie die zur Programmierung des Aktionsbausteins erforderlichen Deklarationen an.
9. Programmieren Sie den Aktionsbaustein FC 16, rufen Sie die Bausteine FB 15 und FC16 vom OB1 aus auf und versehen Sie die Ein-/ Ausgänge der Bausteine mit einer Übergangsvariablen und den SPS-Operanden der Zuordnungstabelle.

Lernaufgabe 6.3: Bördelvorrichtung

Lös. S. 208

In einer Vorrichtung wird ein Rohr in zwei Arbeitsgängen gebördelt. Die Anordnung der vier doppelwirkenden Zylinder der Bördelvorrichtung zeigt das nachfolgende Technologieschema.

Technologieschema:



Fertigungsablauf:

Befinden sich alle Zylinder in der gezeichneten Lage und liegt ein Rohr in der Vorrichtung (S2 = „1“) kann der Bördelvorgang durch Betätigung der Taste S1 gestartet werden. Zylinder 1A

fährt aus und spannt das Rohr in der Vorrichtung. Danach fährt Zylinder 2A mit dem Anschlag ein. Zylinder 3A fährt dann aus und bleibt drei Sekunden in der vorderen Endlage. Nachdem Zylinder 3A wieder eingefahren ist, wird durch das Ausfahren von Zylinder 4A die Bördelform gewechselt. Zylinder 3A fährt dann nochmals aus und bleibt nun 5 Sekunden in der vorderen Endlage. Ist Zylinder 3A wieder eingefahren, fahren Zylinder 4A und Zylinder 1A ebenfalls wieder ein. Nach Entnahme des gebördelten Rohrs, fährt der Zylinder 2A mit dem Anschlag aus und der Bearbeitungsvorgang kann mit dem nächsten Rohr wiederholt werden.

Mit der RESET-Taste kann die Ablaufkette jederzeit in die Grundstellung gebracht werden.

Das Steuerungsprogramm soll mit einem Funktionsbaustein und mit dem Bibliotheks-Funktionsbaustein FB 15 für lineare Ablaufketten ohne Betriebsartenteil sowie dem anlagen-spezifischen Ausgabebaustein FC 16 realisiert werden.

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Stellen Sie den Prozessablauf mit einem Ablauf-Funktionsplan dar.
3. Ermitteln Sie aus dem Ablauf-Funktionsplan die Transitionsbedingungen T-1 bis T-n.

Realisierung mit einem anlagenspezifischen Funktionsbaustein:

4. Zeichnen Sie den Funktionsplan für die Umsetzung der Ablaufkette, der Zeitbildung und der Aktionsausgabe.
5. Geben Sie die zur Programmierung erforderlichen Deklarationen an.
6. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Ein-/ Ausgänge des Bausteins mit SPS-Operanden der Zuordnungstabelle.

Realisierung mit Standard-Funktionsbaustein FB15 und Aktionsbaustein FC16:

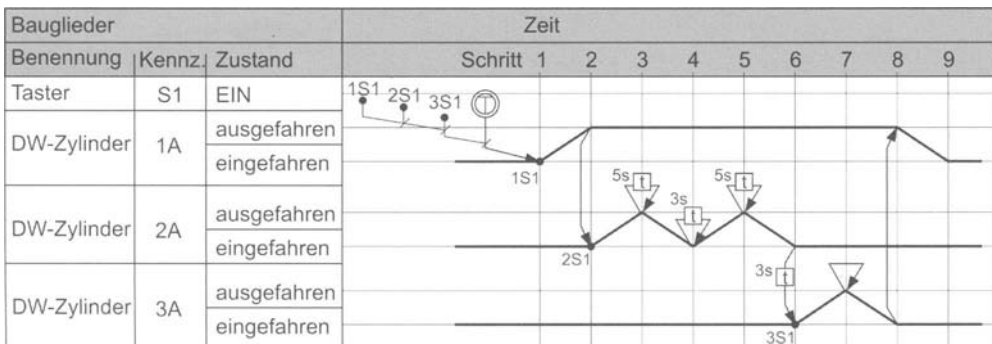
7. Zeichnen Sie den Funktionsplan für die Aktionsausgabe.
8. Geben Sie die zur Programmierung des Aktionsbausteins erforderlichen Deklarationen an.
9. Programmieren Sie den Aktionsbaustein FC 16, rufen Sie die Bausteine FB 15 und FC16 vom OB1 aus auf und versehen Sie die Ein-/ Ausgänge der Bausteine mit einer Übergabevariablen und den SPS-Operanden der Zuordnungstabelle.

Lernaufgabe 6.4: Funktionsdiagramm

Lös. S. 215

Das gegeben Funktionsdiagramm gibt den Funktionsablauf dreier doppelwirkender Zylinder nach Betätigung des Tasters S1 wieder.

Funktionsdiagramm:



Alle drei Zylinder werden mit 5/2-Wegeventilen angesteuert und haben zur Endlagenmeldung jeweils zwei induktive Sensoren. Für Zylinder 1A wird ein beidseitig elektromagnetisch betätigtes 5/2-Wegeventil verwendet. Die 5/2-Wegeventile der Zylinder 2A und 3A sind einseitig elektromagnetisch betätigt. Alle Zylinder sind in der Ausfahrgeschwindigkeit einstellbar.

Das Steuerungsprogramm soll in einem ersten Lösungsansatz mit einem Ablauf-Funktionsplan realisiert werden, der mit dem Standard-Funktionsbaustein FB 15 und Aktionsbaustein FC 16 umgesetzt wird. In einem zweiten Lösungsansatz sollen zwei korrespondierende Ablaufketten den Funktionsablauf beschreiben. Die zweite Ablaufkette übernimmt dabei nur die Funktion des Aus- und Einfahrens von Zylinder 2A und wird deshalb zweimal durchlaufen.

Während beim ersten Lösungsansatz für jedes Aus- und Einfahren des Zylinders 2A ein Schritt benötigt wird, ist beim zweiten Lösungsansatz jeweils nur ein Schritt für das Aus- und das Einfahren des Zylinders erforderlich. Mit einem Zähler wird die Anzahl der Durchläufe der zweiten Ablaufkette gezählt. Ein wesentlicher Vorteil der zweiten Lösungsmethode ergibt sich, wenn Zylinder 2A mehr als zweimal oder wahlfrei ausfährt.

Mit einer RESET-Taste können die Ablaufketten in die Grundstellung gebracht werden.

1. Zeichnen sie den Pneumatikplan für die drei Zylinder.

2. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.

1. Lösungsansatz:

3. Stellen Sie den Funktionsablauf mit einem Ablauf-Funktionsplan dar.

4. Geben Sie in einer Tabelle die Transitionsbedingungen T-1 bis T-n an.

5. Zeichnen Sie den Funktionsplan für die Aktionsausgabe.

6. Geben Sie die zur Programmierung des Aktionsbausteins erforderlichen Deklarationen an.

7. Programmieren Sie den Aktionsbaustein FC 16, rufen Sie die Bausteine FB 15 und FC 16 vom OB1 aus auf und versehen Sie die Ein-/ Ausgänge der Bausteine mit einer Übergangsvariablen und den SPS-Operanden aus der Zuordnungstabelle.

2. Lösungsansatz:

8. Stellen Sie den Funktionsablauf mit zwei korrespondierenden Ablaufketten dar.

9. Ermitteln Sie die Transitionsbedingungen T-1 bis T-n der beiden Ablaufketten.

10. Zeichnen Sie den Funktionsplan für die Aktionsausgabe.

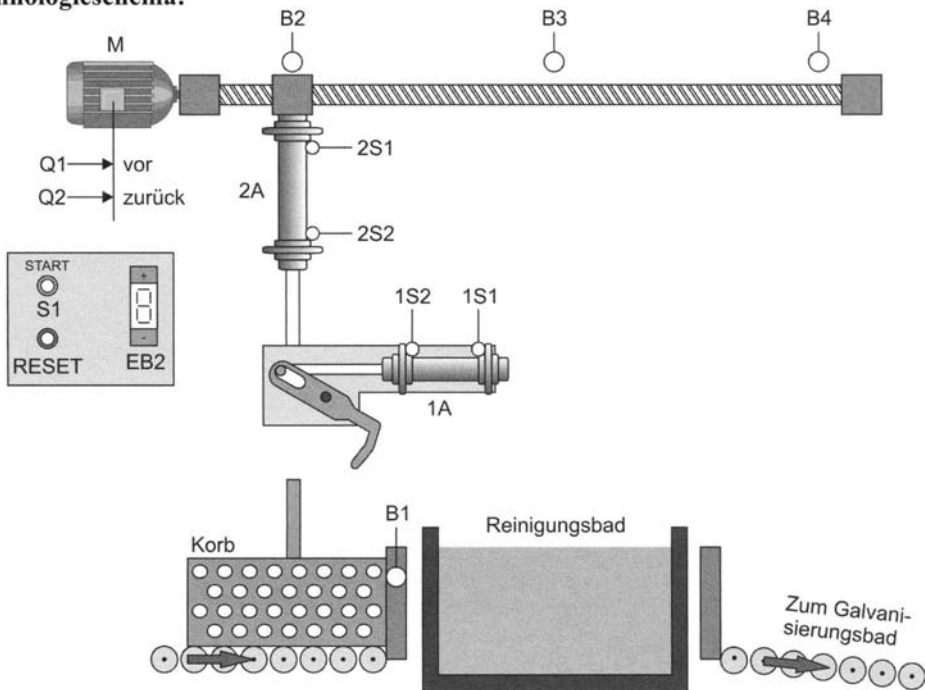
11. Geben Sie die zur Programmierung des Aktionsbausteins erforderlichen Deklarationen an.

12. Programmieren Sie den Aktionsbaustein FC 16, rufen Sie die Bausteine FB 15 (2x) und FC16 vom OB1 aus auf und versehen Sie die Ein-/ Ausgänge der Bausteine mit Übergangsvariablen und SPS-Operanden aus der Zuordnungstabelle.

Lernaufgabe 6.5: Reinigungsbad bei der Galvanisierung

Lös. S. 219

In einer rechnergesteuerten Galvanikanlage wird eine Mehrfachsicht auf Metallteile aufgebracht. Zuvor werden die Metallteile in einem Reinigungsbad von Öl und Schmutzstoffen befreit. Abhängig vom Verschmutzungsgrad wird der Korb mit den Metallteilen mehrmals in das Reinigungsbad eingetaucht. Der einstellige Zifferneinsteller gibt dabei die Anzahl der Tauchvorgänge an. Das nachfolgende Technologieschema zeigt die Anordnung der Sensoren und Aktoren des Reinigungsbades. Die beiden Zylinder werden durch 5/2-Wegeventile mit beidseitig elektromagnetischer Betätigung angesteuert.

Technologieschema:**Fertigungsablauf:**

Befinden sich alle Zylinder in der gezeichneten Endlage und ist ein Korb am Anschlag (Meldung mit B1), kann mit Taster S1 der Reinigungsvorgang gestartet werden. Zylinder 1A fährt aus und hängt den Korb ein. Zylinder 2A fährt ein und hebt den Korb an. Ist der Zylinder 2A ganz eingefahren, wird der Spindelmotor M über das Schütz Q1 angesteuert, bis sich der Korb über dem Reinigungsbad befindet (Meldung mit B3). Der Korb wird nun sooft in das Reinigungsbad getaucht, wie der Zifferneinsteller vorgibt. Bei einem Tauchvorgang wird dabei der Korb durch Ausfahren des Zylinders 2A fünf Sekunden in das Reinigungsbad getaucht. Danach fährt Zylinder 2A wieder ein. Der Korb bleibt dann für neun Sekunden über dem Reinigungsbad stehen. Sind alle Tauchvorgänge beendet, wird der Korb mit dem Spindelmotor zum zweiten Rollenband bewegt (Meldung mit B4). Der Korb wird auf dem Rollenband abgestellt und der Greifer mit dem Spindelmotor in die Ausgangsposition zurückgebracht.

Mit der RESET-Taste können die Ablaufketten in die Grundstellung gebracht werden.

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge für das Reinigungsbad.
2. Stellen Sie den Funktionsablauf mit zwei korrespondierenden Ablaufketten dar.
3. Ermitteln Sie die Transitionsbedingungen T-1 bis T-n der beiden Ablaufketten.
4. Zeichnen Sie den Funktionsplan für die Aktionsausgabe.
5. Geben Sie die zur Programmierung des Aktionsbausteins erforderlichen Deklarationen an.
6. Programmieren Sie den Aktionsbaustein FC 16, rufen Sie die Bausteine FB 15 (2x) und FC16 vom OB1 aus auf und versehen Sie die Ein-/ Ausgänge der Bausteine mit Übergangsvariablen und SPS-Operanden aus der Zuordnungstabelle.

6.3 Kontrollaufgaben

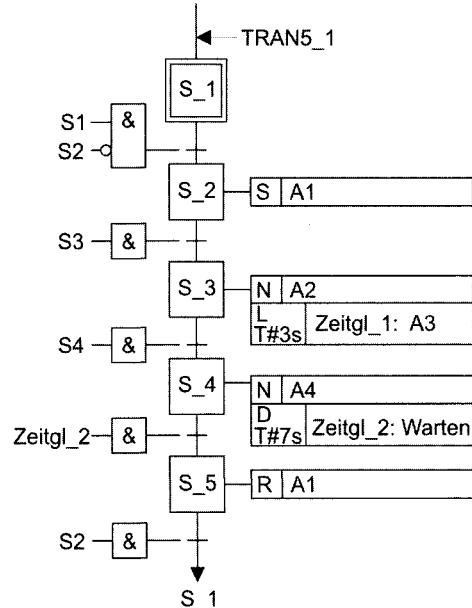
Kontrollaufgabe 6.1

Die nebenstehende lineare Ablaufkette beschreibt die Funktionsweise einer Steuerungsaufgabe mit vier Eingängen S1 bis S4 und vier Ausgängen A1 bis A4. Mit einem weiteren Eingang S0 kann die Ablaufkette jederzeit in die Grundstellung gebracht werden.

- Bestimmen Sie für jeden Ablaufschritt ein RS-Speicherglieder mit den erforderlichen Bedingungen für das Setzen und Rücksetzen.
- Ermitteln Sie den Funktionsplan für die Ansteuerung der Ausgänge A1 bis A4 und der beiden Zeitglieder.

Die Ablaufkette soll mit dem Standardfunktionsbaustein FB 15 und dem Aktionsbaustein FC16 realisiert werden.

- Bestimmen Sie den Funktionsplan für den Aktionsausgabebaustein.
- Geben Sie die zur Programmierung des anlagenspezifischen Funktionsbausteins erforderlichen Deklarationen an.
- Zeichnen Sie den Aufruf des Funktionsbausteins FB 15 und der Funktion FC 16 im OB1. Versehen Sie die Ein-/Ausgänge der Bausteine mit angenommenen SPS-Operanden.



Kontrollaufgabe 6.2

Das nachfolgend als AWL-Quelle angegebene Steuerungsprogramm des Funktionsbausteins FB10 stellt die Realisierung einer linearen Ablaufsteuerung dar. Es ist eine Analyse dieser AWL durchzuführen.

AWL-Quelle:

```

FUNCTION FB 10
VAR_INPUT
    S1      : BOOL;
    S2      : BOOL;
    S3      : BOOL;
    Zeitgl1 : TIMER;
    Zeitw   : S5Time
    RESET   : BOOL;
END VAR

VAR_OUTPUT
    A1 : BOOL;
    A2 : BOOL;
END VAR

VAR
    SRO_1 : BOOL := TRUE;
    SRO_2 : BOOL;
    SRO_3 : BOOL;
    SRO_4 : BOOL;
END VAR

```

```

BEGIN
U( ;           U( ;           U( ;           U #SRO_3;
O #RESET;     O #SRO_3;     O #SRO_4;     L Zeitw;
O ;           O #RESET;     O #RESET;     SE #Zeitgl;
UN #S3;       ) ;           ) ;
U #SRO_4;     R #SRO_2;     R #SRO_3;     U #SRO_2;
) ;           R #SRO_2;     R #SRO_3;     S #A1;
S #SRO_1;     U( ;           U #Zeitgl;    U( ;
U #SRO_2;     O #S2;        U #S3;        O #SRO_4;
R #SRO_1;     O #S3;        U #SRO_3;    O #RESET;
              ) ;           S #SRO_4;    ) ;
U #S1;        U #SRO_2;    U( ;         R #A1;
UN #S2;       S #SRO_3;    O #SRO_1;
U #SRO_1;    O #RESET;    U #SRO_3;
S #SRO_2;    ) ;         = #A2;
              R #SRO_4;    END_FUNCTION_BLOCK

```

1. Zeichnen Sie den Aufruf des Funktionsbausteins FB10 im OB1 in der Funktionsplandarstellung und versehen Sie die Ein- Ausgänge des Bausteins mit den erforderlichen SPS-Operanden.
2. Bestimmen Sie den zu der AWL-Quelle gehörenden Funktionsplan.
3. Ermitteln Sie aus der Anweisungsliste AWL die Transitionsbedingungen T-1 bis T-n und tragen Sie diese in eine Tabelle ein.
4. Zeichnen Sie den zu der Anweisungsliste gehörenden Ablauf-Funktionsplan.

Kontrollaufgabe 6.3

Die nachfolgende Transitionstabelle beschreibt die Weiterschaltbedingungen einer linearen Ablaufkette. In jedem Ablaufschritt wird dabei als Aktion einem Ausgang A1 bis A5 nicht-speichernd ein „1“-Signal zugewiesen.

Transitionstabelle:

| Transition | Weiterschaltbedingungen |
|------------|-------------------------|
| T-1 | T5_1 = 1S1 & 2S1 |
| T-2 | T1_2 = S1 & 1S1 & 2S1 |
| T-3 | T2_3 = 1S1 & 2S2 |
| T-4 | T3_4 = Zeitgl & 2S1 |
| T-5 | T4_5 = 1S2 & 2S1 |

1. Zeichnen Sie den zu der Transitionstabelle und der Beschreibung der Aktionen gehörenden Ablauf-Funktionsplan.

Realisierung der Ablaufkette mit dem anlagenspezifischen Funktionsbaustein FB10:

2. Bestimmen Sie für jeden Ablaufschritt ein RS-Speicherglieder mit den erforderlichen Bedingungen für das Setzen und Rücksetzen.
3. Ermitteln Sie den Funktionsplan für die Ansteuerung der Ausgänge A1 bis A7 und des Zeitgliedes.

Die Ablaufkette soll mit dem Standard-Funktionsbaustein FB 15 und dem Aktionsbaustein FC16 realisiert werden:

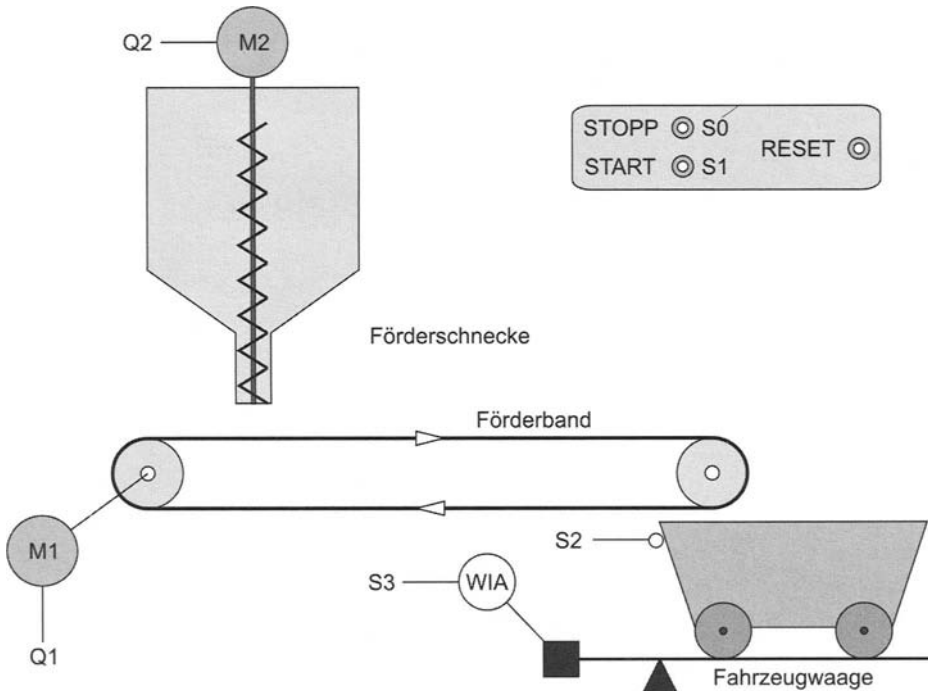
4. Bestimmen Sie den Funktionsplan für die Aktionsausgabe.
5. Geben Sie die zur Programmierung des Aktionsbausteins erforderlichen Deklarationen an.
6. Zeichnen Sie den Aufruf der Bausteine FB 15 und FC16 vom OB1 aus auf und versehen Sie die Ein-/Ausgänge der Bausteine mit entsprechenden SPS-Operanden und Übergabevariablen.

Kontrollaufgabe 6.4

Bei einer Beschickungsanlage wird aus einem Silo durch eine Förderschnecke mit dem Antriebsmotor M2 ein rieselfähiges Gut auf ein Transportband mit dem Antriebsmotor M1 gebracht und in einen Wagen geladen. Der Wagen steht auf einer Waage, die über einen Sensor S3 meldet, wenn der Wagen gefüllt ist.

Der Bestückungsvorgang wird durch Betätigen des Start-Tasters S1 ausgelöst, sofern der Wagen an der Verlade rampe steht (Meldung mit S2). Damit sich kein Fördergut auf dem Transportband staut, muss zunächst das Transportband 3 s laufen, bevor die Förderschnecke in Betrieb gesetzt wird. Meldet der Sensor S3 an der Waage, dass der Wagen gefüllt ist, oder meldet der Endschalter S2, dass sich der Wagen nicht mehr in der Endposition der Verlade rampe befindet, oder wird der Stopp-Taster S0 betätigt, wird die Förderschnecke sofort abgeschaltet. Das Förderband läuft jedoch noch 5 s weiter, um das Band völlig zu entleeren. Ein weiterer Beschickungsvorgang kann dann wieder durch die Betätigung des Start-Tasters S1 eingeleitet werden.

Technologieschema:



1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Stellen Sie den Prozessablauf mit einem Ablauf-Funktionsplan dar.
3. Ermitteln Sie aus dem Ablauf-Funktionsplan die Transitionsbedingungen T-1 bis T-n und tragen Sie diese in eine Tabelle ein.

Realisierung mit dem anlagenspezifischen Funktionsbaustein FB10:

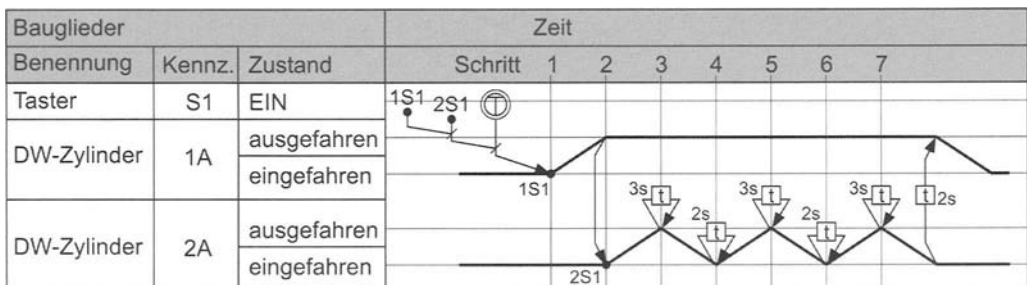
4. Zeichnen Sie den Funktionsplan für die Umsetzung der Ablaufkette, der Zeitbildung und der Aktionsausgabe.
5. Geben Sie die zur Programmierung des anlagenspezifischen Funktionsbausteins erforderlichen Deklarationen an.

Realisierung mit dem Standard-Funktionsbaustein FB15 und dem Ausgabebaustein FC16:

6. Bestimmen Sie den Funktionsplan für die Aktionsausgabe.
7. Geben Sie die zur Programmierung des Aktionsbausteins erforderlichen Deklarationen an.
8. Zeichnen Sie den Aufruf der Bausteine FB15 und FC16 vom OB1 aus auf und versehen Sie die Ein-/Ausgänge der Bausteine mit entsprechenden SPS-Operanden und Übergabevariablen.

Kontrollaufgabe 6.5

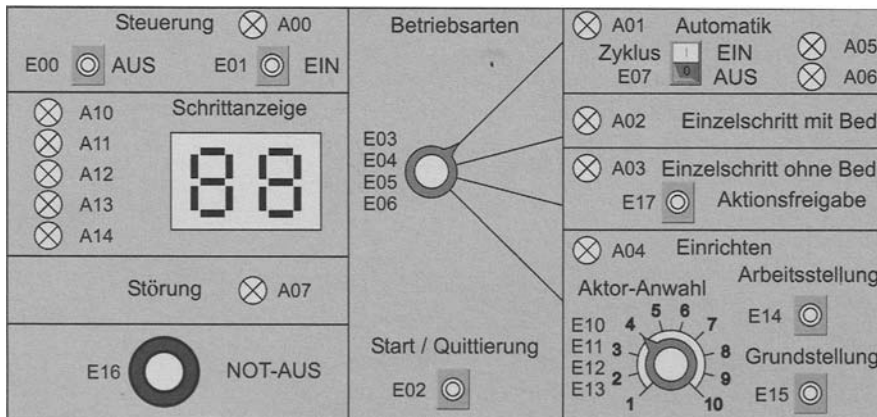
Das gegebene Funktionsdiagramm gibt den Funktionsablauf von zwei doppelwirkenden Zylindern nach Betätigung des Tasters S1 wieder. Die beiden Zylinder werden von 5/2-Wegeventilen mit beidseitig elektromagnetischer Betätigung angesteuert und haben zur Endlagenmeldung jeweils zwei induktive Sensoren.



1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Stellen Sie den Funktionsablauf mit zwei korrespondierenden Ablaufketten dar.
3. Ermitteln Sie die Transitionsbedingungen T-1 bis T-n der beiden Ablaufketten und tragen Sie diese in entsprechende Tabellen ein.
4. Bestimmen Sie den Funktionsplan für die Aktionsausgabe.
5. Geben Sie die zur Programmierung des Aktionsbausteins erforderlichen Deklarationen an.
6. Zeichnen Sie den Aufruf der Bausteine FB15 (2x) und FC16 vom OB1 aus auf und versehen Sie die Ein-/Ausgänge der Bausteine mit entsprechenden SPS-Operanden und Übergabevariablen.

Funktionen eines Bedienfeldes

Eingriffe in Steuerungen sollen von einem Bedienfeld aus erfolgen, dessen Bedienoberfläche nicht auf eine spezielle Steuerungsaufgabe zugeschnitten, sondern allgemein gehalten ist.



Beabsichtigte Funktionen der Befehlsgeber und Anzeigen des Bedienfeldes:

Taster E00: Steuerung: AUS Hinweis: Statt der beiden Taster kann auch ein EIN-AUS-

Taster E01: Steuerung: EIN Schalter bzw. ein Schlüsselschalter verwendet werden.

Taster E02: Start/Quittierung

Automatikbetrieb: Ist die Betriebsart mit E03 vorgewählt, wird beim Betätigen der Taste der Automatikbetrieb gestartet. *Einzelschrittbetrieb:* Bei Betätigung erfolgt eine Einzelschritt-Weiterschaltung. Bei ausgeschalteter Steuerung wird in besonderen Betriebsituationen durch Betätigen der Start/Quitt-Taste das RESET Betriebsartensignal B0 ausgelöst.

Wahlschalter E03, E04, E05 und E06: Start / Einzelschritt

Wahl der Betriebsart: E03 = 1 für Automatik, E04 = 1 für Einzelschrittbetrieb mit Bedingungen, E05 = 1 für Einzelschrittbetrieb ohne Bedingungen und E06 = 1 für Einrichten. Ein Betriebsartenwechsel ist über den Wahlschalter jederzeit ohne Bearbeitungsabbruch möglich. Die Umschaltung in die Betriebsart Weiterschalten ohne Bedingungen und Einrichten stoppt die Ansteuerung der Aktoren.

Taster E07: Zyklus EIN / AUS

Steht der Schalter auf „EIN“ wird ein Bearbeitungszyklus ständig wiederholt. Steht der Wahlschalter auf „AUS“ wird der Bearbeitungszyklus nur einmal durchlaufen und der Automatikbetrieb dann beendet. Wird während eines Bearbeitungszyklus von „EIN“ auf „AUS“ umgeschaltet, wird der Automatikbetrieb nach Ablauf des Zyklus beendet.

Wahlschalter E10 ... E13: Aktor Anwahl

In der Betriebsart Einrichten können 10 verschiedene Aktoren angewählt werden. An den Eingängen E10 ... E13 liegt der Wert dualcodiert vor. Die Auswahl bezieht sich dabei auf Zylinder oder Motoren.

Taster E14: Aktor - Arbeitsstellung

Die Bewegung wird nur bei gedrückter Taste ausgeführt. Der gewählte Zylinder fährt in die Arbeitsstellung bzw. der gewählte Motor dreht „rechts“.

Taster E15: Aktor - Grundstellung

Die Bewegung wird nur bei gedrückter Taste ausgeführt. Der gewählte Zylinder fährt in die Grundstellung bzw. der gewählte Motor dreht „links“.

Taster E16: NOT-AUS-Taster

Eine Betätigung dieses oder weiterer NOT-AUS-Taster schaltet die Steuerung aus. Alle Bewegungen werden angehalten und die Antriebe stillgesetzt.

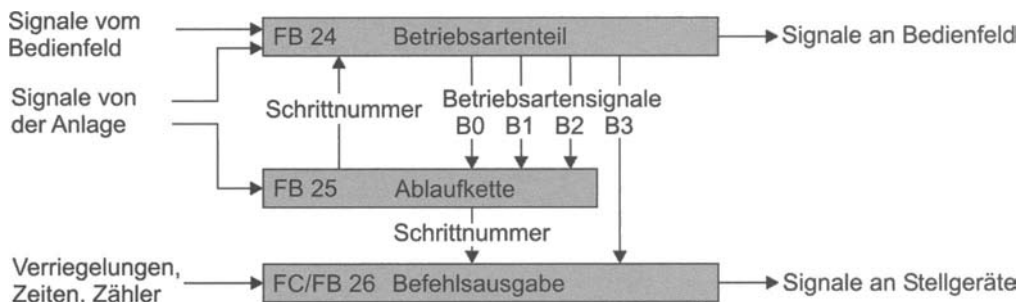
Taster E17: Aktionsfreigabe

In der Betriebsart Einzelschrittbetrieb ohne Bedingung wird bei gedrückter Taste die Aktion des aktiven Schrittes ausgeführt.

Anzeigen:

Mit den Leuchtmeldern A00 bis A07 wird der jeweils zugehörige Betriebszustand angezeigt. A10...A14 sind für die dualcodierte Anzeige des jeweiligen Schrittes vorgesehen.

Struktur einer Ablaufsteuerung



Ablaufsteuerungen in der Praxis bestehen aus Ablaufketten und verfügen über einen übergeordneten Betriebsartenteil für die Inbetriebnahme, Anlagenbetrieb und Störungsbeseitigung. Ablaufsteuerungen bestehen im Prinzip immer aus drei Bausteinen:

- FB 24: Standard - Betriebsartenbaustein
- FB 25: Standard - Ablaufkettenbaustein für lineare Ablaufsteuerungen
- FC /FB 26: Anlagenspezifischer Befehlsausgabebaustein

Betriebsartensignale

Betriebsartensignale sind die vom Betriebsartenprogramm erzeugten Steuersignale für das Weiterschalten und Rücksetzen der Ablaufkette sowie zur Freigabe von Aktionen. In Klammern angegeben sind deren zugehörige Übergabevariablen zwischen den Bausteinen. An dieser Stelle wird nicht die Erzeugung sondern nur die Verwendung der Betriebsartensignale beschrieben.

Rücksetzen RESET (B0)

RESET wirkt durch die Übergabevariable B0 auf den Eingangsparameter RESET des Ablaufkettenbausteins FB25 und versetzt die Schrittkette in die Grundstellung oder wirkt über den RESET-Eingang des Aktionsbausteins und setzt dort Speicher, Zeitglieder und Zähler zurück.

Freigabe der Kette mit Bedingungen FR_K_mB (B1)

FR_K_mB wirkt durch die Übergabevariable B1 auf den Eingangsparameter WEITER_mB im Ablaufkettenbaustein. Nur wenn B1 = 1 ist, wird bei erfüllten Weiterschaltbedingungen der nächste Schritt gesetzt. Das B1-Signal kann ein Impuls bei Hand-Einzelschrittbetrieb oder ein Dauersignal bei Automatikbetrieb sein.

Freigabe der Kette ohne Bedingungen FR_K_oB (B2)

FREI_K_oB wirkt durch die Übergabevariable B2 auf den Eingangsparameter WEITER_oB im Ablaufkettenbaustein. Nur wenn B2 = 1 ist, wird im Hand-Einzelschrittbetrieb der nächste Schrittspeicher gesetzt, ohne dass die Weiterschaltbedingung erfüllt sein muss. Das B2-Signal muss immer ein Impulssignal mit der Länge von einer Zykluszeit sein.

Freigabe Aktion FR_AKTION (B3)

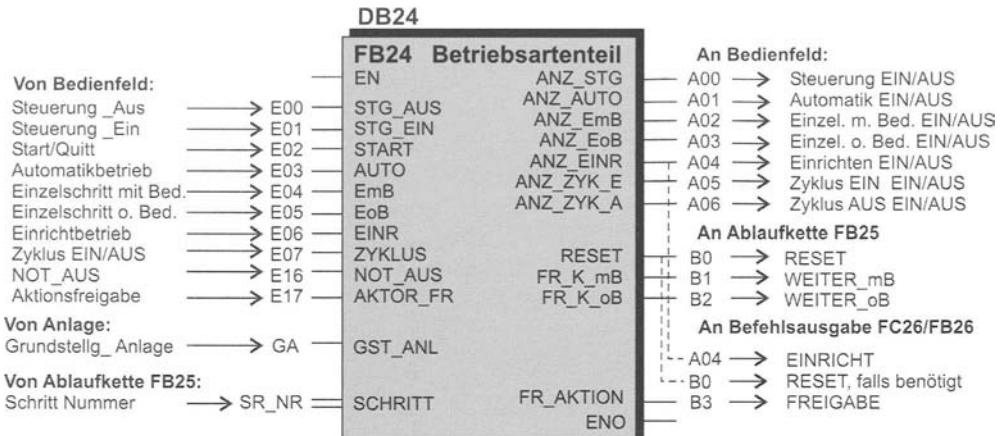
FR_AKTION wirkt durch die Übergabevariable B3 auf den Eingangsparameter FREIGABE des Aktionsbausteins und kann dort einen Aktionsausgang freischalten bzw. sperren.

Beschreibung der Standard Funktionsbausteine des Ablaufsteuerungskonzepts

• **Betriebsartenteil-Baustein FB 24**

Das Betriebsartenprogramm des FB 24 setzt die Eingaben des Bedienfeldes und Rückmeldungen aus der Anlage in Steuersignale für die Bausteine FB 25 und FC/FB 26 um.

Schnittstellen des Betriebsarten-Funktionsbausteins FB24:

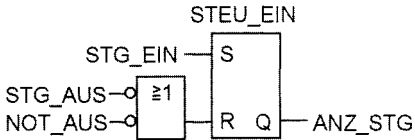


Deklarationstabelle des Betriebsarten-Funktionsbausteins FB 24:

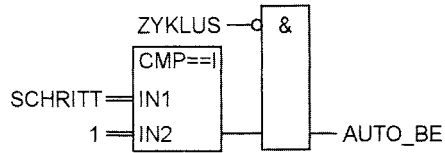
| Name | Datentyp | Name | Datentyp | Name | Datentyp |
|-----------|----------|------------|----------|-------------|----------|
| IN | | OUT | | STAT | |
| STG_AUS | Bool | ANZ_STG | Bool | STEU_EIN | Bool |
| STG_EIN | Bool | ANZ_AUTO | Bool | AUTO_EIN | Bool |
| START | Bool | ANZ_EmB | Bool | AUTO_BE | Bool |
| AUTO | Bool | ANZ_EoB | Bool | START_SP | Bool |
| EmB | Bool | ANZ_EINR | Bool | FO1 | Bool |
| EoB | Bool | ANZ_ZYK_E | Bool | FO2 | Bool |
| EINR | Bool | ANZ_ZYK_A | Bool | IO_WEITER | Bool |
| ZYKLUS | Bool | RESET | Bool | | |
| NOT_AUS | Bool | FR_K_mB | Bool | | |
| AKTOR_FR | Bool | FR_K_oB | Bool | | |
| GST_ANL | Bool | FR_Aktion | Bool | | |
| SCHRITT | Int | | | | |

Funktionsplan des Betriebsartenbausteins FB24:

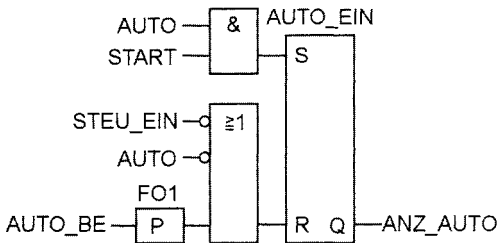
Steuerung EIN-AUS



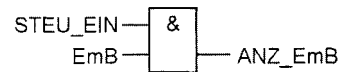
Vorwahl Automatikbetrieb beenden



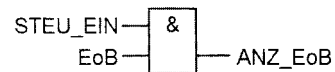
Automatikbetrieb



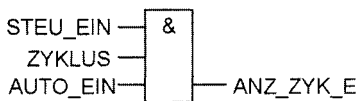
Anzeige Einzelschritt mit Bedingung



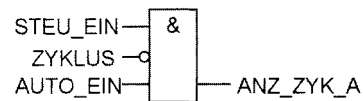
Anzeige Einzelschritt ohne Bedingung



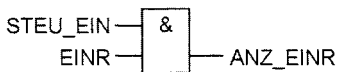
Anzeige Zyklus EIN



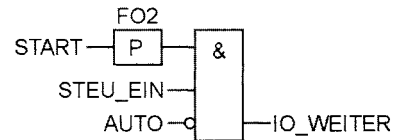
Anzeige Zyklus AUS



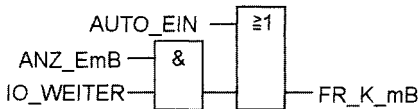
Anzeige Einrichtbetrieb



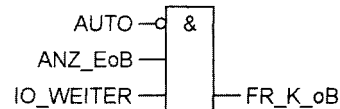
Einzelschritt Weiterschalt-Impuls



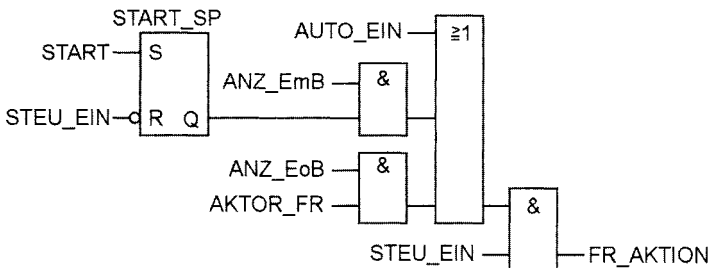
Freigabe Kette mit Bedingungen



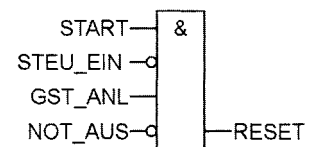
Freigabe Kette ohne Bedingungen



Aktionsfreigabe



Reset

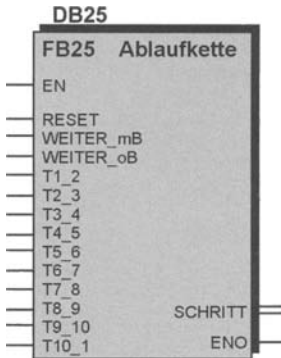


Der Betriebsarten-Funktionsbaustein FB24 wird bei den nachfolgenden Lernaufgaben unverändert übernommen. Auch bei der Verwendung von Bedienfeldern mit nur drei wählbaren Betriebsarten (siehe Lernaufgabe 7.6) kann der Funktionsbaustein FB24 eingesetzt werden. Da das Programm für den Baustein FB24 empirisch entwickelt wurde, macht es wenig Sinn, sich ausführlich mit diesem zu beschäftigen. Sowohl in der Ausbildung wie in der Praxisanwendung genügt es, die funktionale Beschreibung des Bausteins zu kennen.

Funktion und Programm des Ablaufkettenbausteins (FB 25)

Der Standard-Funktionsbaustein FB 25 ist für zehn lineare Ablaufschritte ausgelegt und unterscheidet sich vom FB 15 aus Kapitel 6 durch seine Verwendungsmöglichkeit im Zusammenhang mit dem Betriebsartenbaustein FB 24. Neu ist seine Fähigkeit, sich im Einzelschrittmodus ohne Weiterschaltbedingungen steuern zu lassen.

Schnittstellen des Funktionsbausteins:



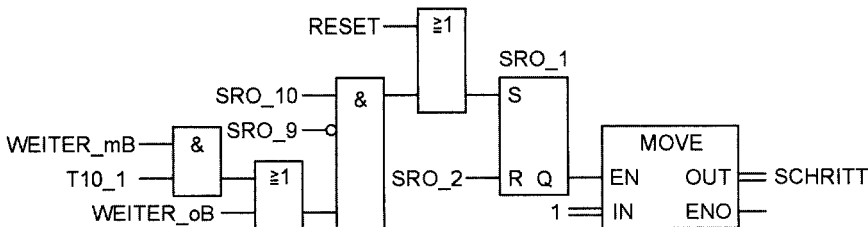
Beschreibung der Übergabeparameter:

- RESET: Variable für das Signal B0.
- WEITER_mB: Variable für das Signal B1.
- WEITER_oB: Variable für das Signal B2.
- T1_2: Variable für die Transition (Weiterschaltbedingung) von Schritt 1 nach Schritt 2.
- T2_3 bis T10_1 Variablen für die Transition (Weiterschaltbedingungen) der restlichen Schritte.
- SCHRITT: Gibt die Nummer des aktuellen Schrittes im Datenformat Integer an.

Der Standardfunktionsbaustein FB 25 unterscheidet sich vom Schrittkettenbaustein FB15 durch die drei weiteren Eingangsvariablen: RESET, WEITER_mB und WEITER_oB.

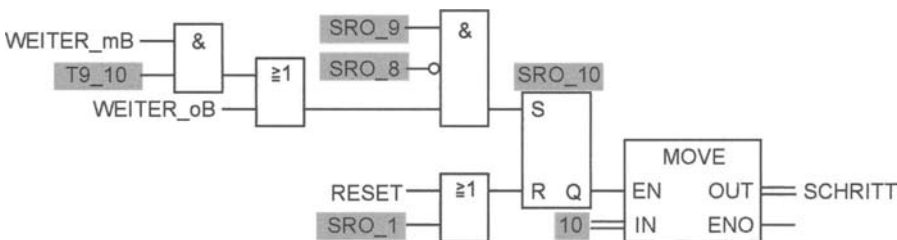
Steuerungsprogramm des Ablaufketten-Funktionsbausteins FB25:

Schritt 1: Initialschritt



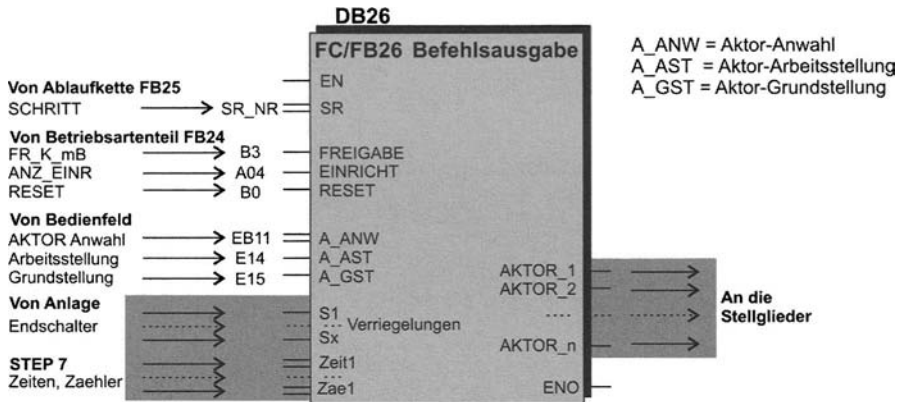
Die Funktionspläne der Schritte 2 bis Schritt 10 haben alle die gleiche Struktur. Im nachfolgend dargestellten Funktionsplan für Schritt 10 sind die Variablen grau hinterlegt, welche bei dem jeweiligen Schritt angepasst werden müssen.

Schritt 10:



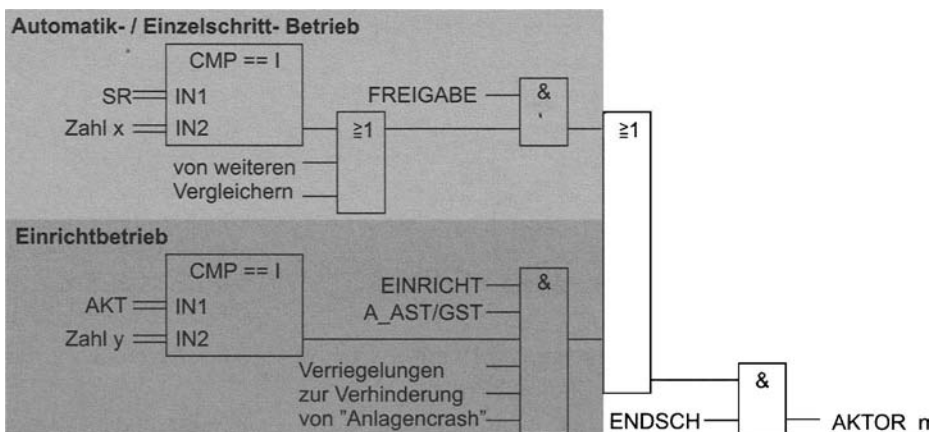
Die Eingangsparameter RESET, WEITER_mB und WEITER_oB. werden von der gewählten Betriebsart beeinflusst. Bei Automatikbetrieb und im Modus Weiterschalten mit Bedingungen erhält der Eingang WEITER_mB ein „1“-Signal.

Funktion und Programm des Befehlsausgabebausteins FC/FB26



Für alle Ausgänge des Befehlsausgabe-Bausteins gilt die gleiche, im nachfolgenden Funktionsplan gezeigte Steuerungsstruktur. Die dort hellgrau hinterlegten Verknüpfungen ergeben sich aus der Ansteuerung des Stellgliedes in den Betriebsarten Automatik und Einzelschritt. Die in dunkelgrau hinterlegte Fläche zeigt die Verknüpfung für die Ansteuerung des Stellgliedes im Einrichtbetrieb.

Struktur der Ansteuerung eines Stellgliedes (Aktor_n) im Befehlsausgabebaustein:



In den Betriebsarten Automatik, Einzelschritt und Einrichten ist bei der Ansteuerung von Stellgliedern, die eine Bewegung zur Folge haben und bestimmte Endschalter nicht überfahren dürfen, zusätzlich eine Endschalter-Verriegelung erforderlich.

Der im Funktionsplan dunkelgrau hinterlegte Teil für den Einrichtbetrieb muss anlagenabhängig realisiert werden. Folgende Regeln sind beim Entwurf zu berücksichtigen:

1. Am Eingang A_ANW (Datenformat BYTE) des Ausgabebausteins wird im Einrichtbetrieb der anzusteuernde Aktor angegeben. Damit die übrigen 4 Binärstellen des Bytes noch verwendet werden können, sind in der nebenstehenden AWL die niederwertigen 4 Bits der Eingangsvariablen A_ANW mit W#16#000F maskiert und der lokalen Variablen AKT zugewiesen.

```

AWL:
L #A_ANW
L W#16#F
UW
T #AKT
  
```

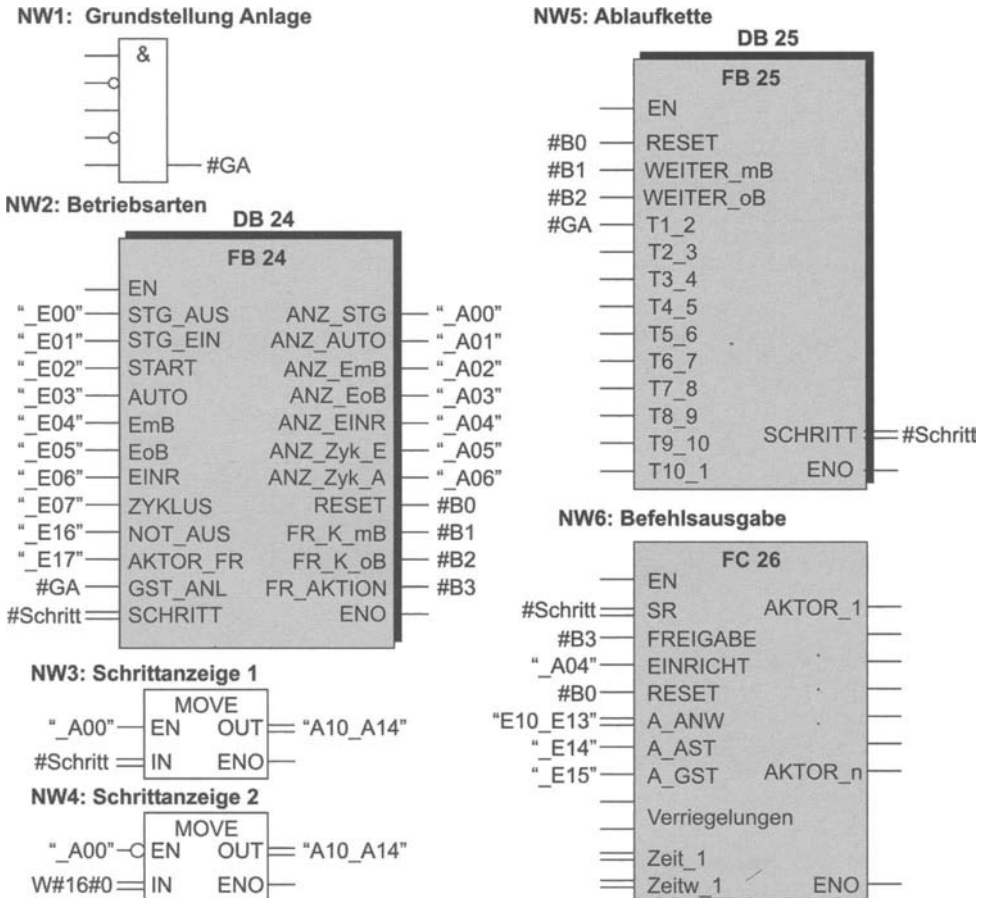
- Mit einem Vergleich wird bestimmt, welcher Aktor ausgewählt wurde. Die Variable A_AST (Aktor Arbeitsstellung) bzw. A_GST (Aktor Grundstellung) am nachfolgenden UND-Glied veranlasst dann die Befehlsausgabe.

Bauelementstruktur mit Übergabevariablen im OB 1

Deklarationstabelle:

| Name | Datentyp | Kommentar |
|------------------|----------|---------------------------------|
| TEMP | | |
| Standardeinträge | | |
| Schritt | Int | Übergabe Schrittnummer |
| GA | Bool | Grundstellung Anlage |
| B0 | Bool | RESET |
| B1 | Bool | Weiterschalten mit Bedingungen |
| B2 | Bool | Weiterschalten ohne Bedingungen |
| B3 | Bool | Aktionsfreigabe |

Programm:



Hinweis: "_E00" u. a. sind in der Symboltabelle für einen SPS-Operanden deklariert.

7.1

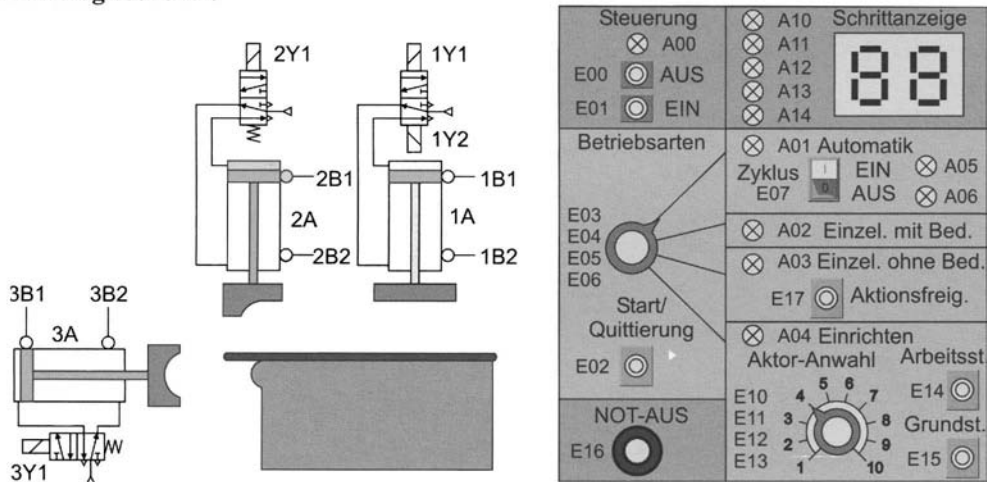
Beispiel

Anwendung der Baueinstruktur

Biegewerkzeug

Das zu biegende Blech wird von Hand in die Biegevorrichtung eingelegt und der Biegevorgang vom Bedienfeld aus eingeleitet. Der Zylinder 1 fährt aus und hält das Blech fest. Der Zylinder 2 biegt das Blech zunächst um 90° bevor Zylinder 3 das Blech in die endgültige Form bringt.

Technologieschema:

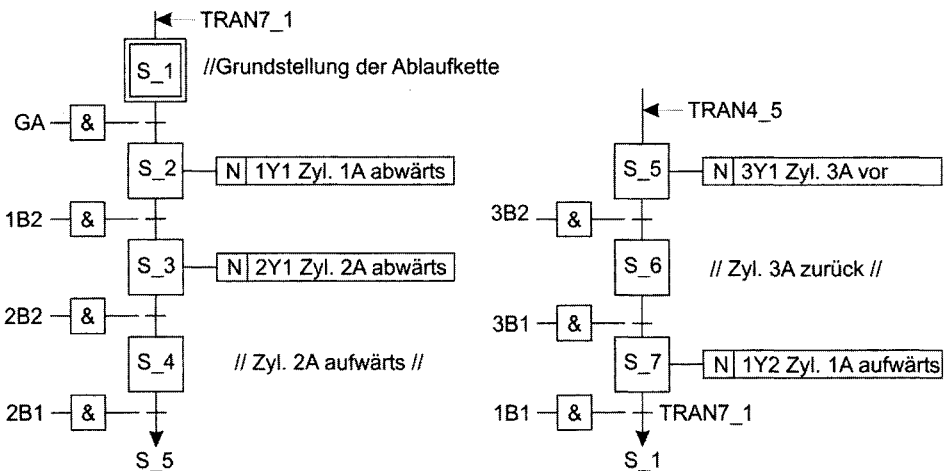


Zuordnungstabelle der Eingänge und Ausgänge von Bedienfeld und Anlage

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|--------------------------|---------|----------|--------------------------|------------|
| Steuerung EIN | _E00 | BOOL | Betätigt _E00 = 0 | E 10.0 |
| Steuerung AUS | _E01 | BOOL | Betätigt _E01 = 1 | E 10.1 |
| Taster Start/Quittierung | _E02 | BOOL | Betätigt _E02 = 1 | E 10.2 |
| Auswahl Automatik | _E03 | BOOL | Ausgewählt _E03 = 1 | E 10.3 |
| Auswahl Einzelschr. m.B. | _E04 | BOOL | Ausgewählt _E04 = 1 | E 10.4 |
| Auswahl Einzelschr. o.B. | _E05 | BOOL | Ausgewählt _E05 = 1 | E 10.5 |
| Auswahl Einrichten | _E06 | BOOL | Ausgewählt _E06 = 1 | E 10.6 |
| Wahlschalter Zyklus | _E07 | BOOL | Zyklus ein _E07 = 1 | E 10.7 |
| Aktor-Anwahl | E10_E13 | ½ BYTE | Dualzahl von 1 bis 10 | E11.0–11.3 |
| Taster Arbeitsstellung | _E14 | BOOL | Betätigt _E14 = 1 | E 11.4 |
| Taster Grundstellung | _E15 | BOOL | Betätigt _E15 = 1 | E 11.5 |
| NOT-AUS | _E16 | BOOL | Betätigt _E16 = 0 | E 11.6 |
| Taster Aktionsfreigabe | _E17 | BOOL | Betätigt _E17 = 1 | E 11.7 |
| Hintere Endl. Zylinder 1 | 1B1 | BOOL | Endlage erreicht 1B1 = 1 | E 0.1 |
| Vordere Endl. Zylinder 1 | 1B2 | BOOL | Endlage erreicht 1B2 = 1 | E 0.2 |
| Hintere Endl. Zylinder 2 | 2B1 | BOOL | Endlage erreicht 2B1 = 1 | E 0.3 |
| Vordere Endl. Zylinder 2 | 2B2 | BOOL | Endlage erreicht 2B2 = 1 | E 0.4 |
| Hintere Endl. Zylinder 3 | 3B1 | BOOL | Endlage erreicht 3B1 = 1 | E 0.5 |
| Vordere Endl. Zylinder 3 | 3B2 | BOOL | Endlage erreicht 3B2 = 1 | E 0.6 |

| Ausgangsvariable | | | | | |
|---------------------------|---------|--------|----------------------|----------|------------|
| Anzeige Steuerung EIN | _A00 | BOOL | Leuchtet | _A00 = 1 | A 10.0 |
| Anzeige Automatik | _A01 | BOOL | Leuchtet | _A01 = 1 | A 10.1 |
| Anzeige Einzelschr. m. B. | _A02 | BOOL | Leuchtet | _A02 = 1 | A 10.2 |
| Anzeige Einzelschr. o. B. | _A03 | BOOL | Leuchtet | _A03 = 1 | A 10.3 |
| Anzeige Einrichten | _A04 | BOOL | Leuchtet | _A00 = 1 | A 10.4 |
| Anzeige Zyklus EIN | _A05 | BOOL | Leuchtet | _A01 = 1 | A 10.5 |
| Anzeige Zyklus AUS | _A06 | BOOL | Leuchtet | _A02 = 1 | A 10.6 |
| Schrittanzeige | A10_A14 | 5xBool | Dualzahl von 0 .. 32 | | A11.0–11.4 |
| Magnetspule 1 Zyl. 1A | 1Y1 | BOOL | Zyl.1A vor | 1Y1 = 1 | A 4.0 |
| Magnetspule 2 Zyl. 1A | 1Y2 | BOOL | Zyl.1A zurück | 1Y2 = 1 | A 4.1 |
| Magnetspule 1 Zyl. 2A | 2Y1 | BOOL | Zyl.2A vor | 2Y1 = 1 | A 4.2 |
| Magnetspule 1 Zyl. 3A | 3Y1 | BOOL | Zyl.3A vor | 3Y1 = 1 | A 4.4 |

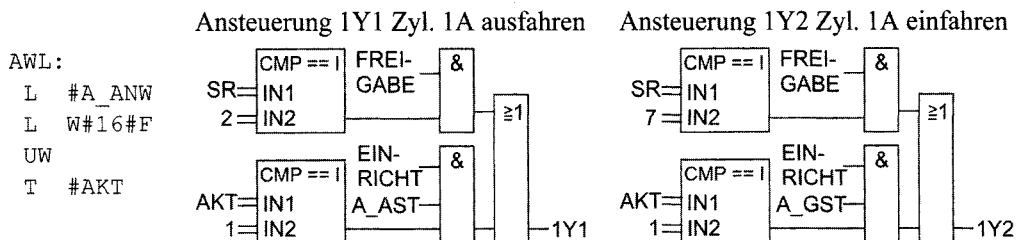
Ablauf-Funktionsplan:



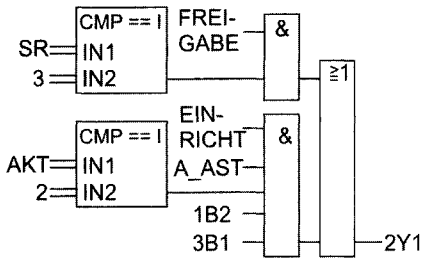
Transitionstabelle

| Transition | Weiterschaltbedingung | Transition | Weiterschaltbedingung |
|------------|-----------------------|------------|-----------------------|
| T-1 | T7_1 = 1B1 | T-5 | T4_5 = 2B1 |
| T-2 | T1_2 = GA | T-6 | T5_6 = 3B2 |
| T-3 | T2_3 = 1B2 | T-7 | T6_7 = 3B1 |
| T-4 | T3_4 = 2B2 | | |

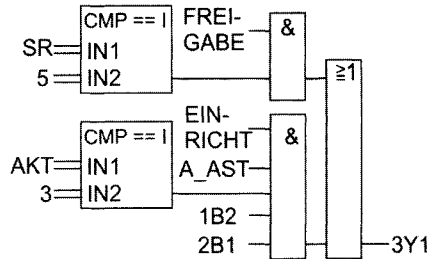
Funktionsplan für den Aktionsausgabebaustein FC 26



Ansteuerung 2Y1 Zyl. 2A ausfahren



Ansteuerung 3Y1 Zyl. 3A ausfahren



Deklarationstabelle für FC 26

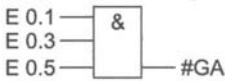
| Name | Datentyp |
|----------|----------|
| IN | |
| SR | Int |
| FREIGABE | Bool |
| EINRICHT | Bool |
| A_ANW | Bool |
| A_AST | Bool |
| A_GST | Bool |

| Name | Datentyp |
|------|----------|
| IN | |
| 1B2 | Bool |
| 2B1 | Bool |
| 3B1 | Bool |
| Name | Datentyp |
| TEMP | |
| AKT | Int |

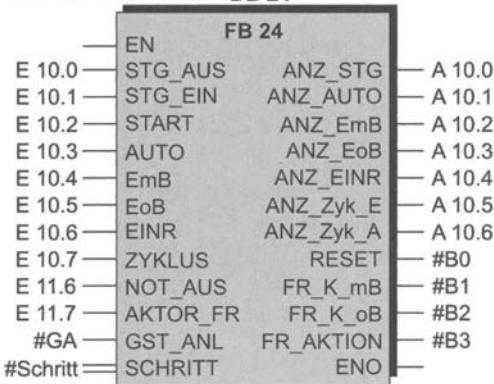
| Name | Datentyp |
|------|----------|
| OUT | |
| _1Y1 | Bool |
| _1Y2 | Bool |
| _2Y1 | Bool |
| _3Y1 | Bool |

Aufruf der Bausteine im OB1:

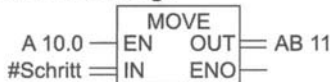
NW1: Grundstellung Anlage



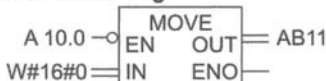
NW2: Betriebsarten DB 24



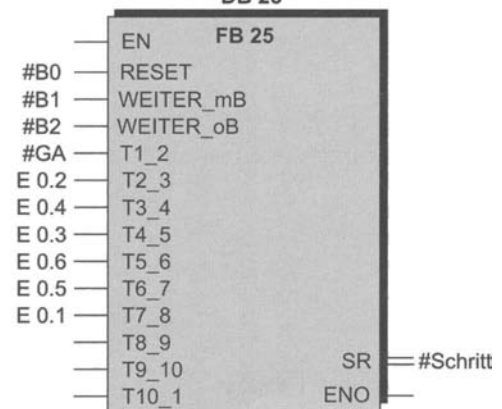
NW3: Schrittanzeige



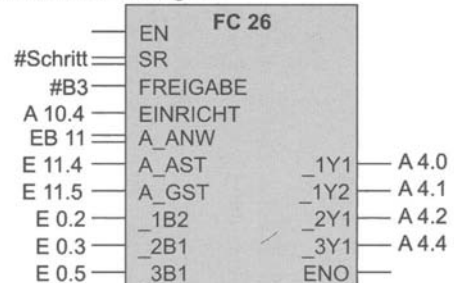
NW4: Schrittanzeige



NW5: Ablaufkette DB 25



NW6: Befehlsausgabe



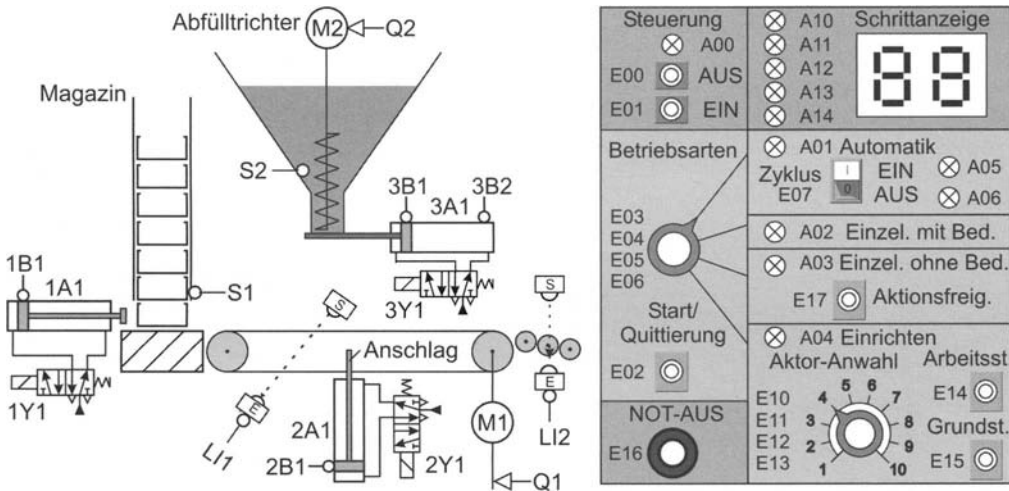
7.2 Lernaufgaben

Lernaufgabe 7.1: Abfüllanlage

Lös. S. 223

In einer Abfüllanlage wird Waschpulver in die zugehörigen Behältnisse gefüllt. Die Öffnungszeit des Abfüllbehälters bestimmt dabei die Abfüllmenge. Zur Inbetriebnahme und beim Auftreten von Störungen soll das Standard-Bedienfeld verwendet werden.

Technologieschema:



Prozessablauf:

Nach Betätigung des Start-Tasters E02 in der eingestellten Betriebsart „Automatik“, schiebt bei Grundstellung der Anlage der Zylinder 1A1 ein Waschmittelbehälter auf das Förderband. Meldet die Lichtschranke LI1, dass der Behälter den vorderen Rand des Abfülltrichters erreicht, fährt ein Anschlag durch Zylinder 2A1 aus. Zwei Sekunden nachdem Zylinder 2A1 ausgefahren ist, wird der Bandmotor M1 abgeschaltet und mit der Füllung begonnen. Dazu öffnet Zylinder 3A1 den Auslass des Abfülltrichters. Der Antrieb der Förderschnecke wird eingeschaltet. Nach der Abfüllzeit von 8 Sekunden wird der Auslass geschlossen und der Förderschneckenantrieb ausgeschaltet. Der Anschlag wird eingefahren und das Förderband läuft wieder an. Die Waschmittelbehälter werden über eine Rollenbahn zur Verpackungsstation transportiert. Die Lichtschranke LI2 meldet, dass ein Behälter auf der Rollenbahn angekommen ist und es kann mit dem nächsten Abfüllvorgang begonnen werden.

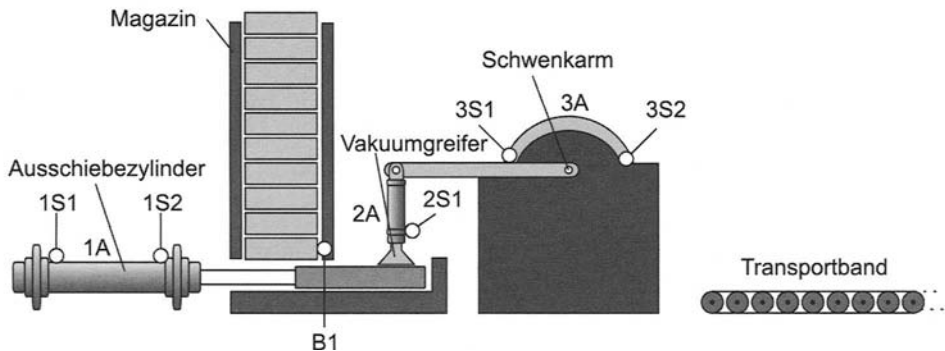
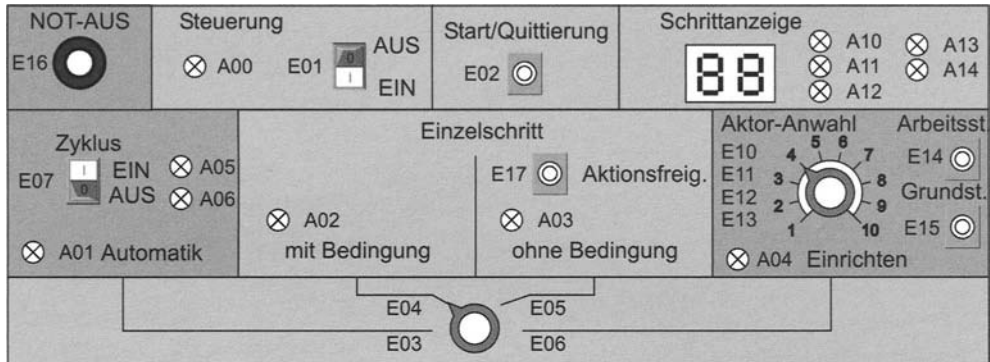
- Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge der Anlage. (Hinweis: Ein- und Ausgänge des Bedienfeldes aus Beispiel 7.1 siehe Seite 101.)
- Stellen Sie den Prozessablauf mit einem Ablauf-Funktionsplan dar.
- Ermitteln Sie aus dem Ablauf-Funktionsplan die Transitionsbedingungen T-1 bis T-n.
- Zeichnen Sie den Funktionsplan für die Aktionsausgabe.
- Geben Sie die zur Programmierung des Aktionsbausteins erforderlichen Deklarationen an.
- Programmieren Sie den Aktionsbaustein FC/FB26, rufen Sie die Bausteine FB24, FB25 und FC/FB26 vom OB1 aus auf und versehen Sie die Ein-/ Ausgänge der Bausteine mit entsprechenden SPS-Operanden der Zuordnungstabelle und Übergabevariablen.

Lernaufgabe 7.2: Zubringeinrichtung für Verpackungen

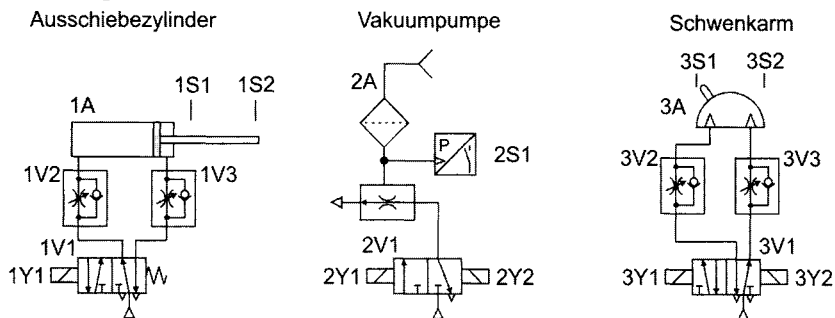
Lös. S. 226

Bei der Produktion von Kern- und Feinseifen werden Verpackungsschachteln aus einem Magazin vereinzelt und mit einem Schwenkantrieb auf ein Rollenband für den weiteren Verpackungsprozess gebracht. Mit einem an der Station angebrachten Bedienfeld kann die Anlage mit verschiedenen Betriebsarten gefahren werden.

Technologieschema:



Pneumatikschaltplan:



Prozessablauf:

Meldet in der eingestellten Betriebsart „Automatik“ der Initiator B1, dass eine Seifenschachtel im Magazin vorhanden ist und befinden sich der Ausschiebezylinder 1A und der Schwenkarm 3A in den gezeichneten Positionen, beginnt der Ablaufzyklus damit, dass der Schwenkarm 3A in die Position Transportband und der Ausschiebezylinder 1A in die hintere Endlage fährt. Erreichen die beiden Zylinder jeweils die Endlagengeber 1S1 bzw. 3S2, fährt der Zylinder 1A

wieder aus und platziert so eine Seifenschachtel auf den Übergabeplatz. Der Schwenkarm 3A fährt dann in die Position Magazin zurück. Durch Ansteuerung der Spule 2Y1 wird im Vakuumgreifer 2A nach dem Venturi-Prinzip ein Unterdruck erzeugt. Meldet der Unterdruckschalter 2S1 einen ausreichenden Unterdruck, fährt der Schwenkarm mit der Schachtel zur Position Transportband. Durch die Spule 2Y2 wird die Unterdruckerzeugung abgeschaltet und die Seifenschachtel auf dem Transportband abgelegt. Der Schwenkarm fährt wieder in seine Ausgangsposition zurück und der Ablauf kann wiederholt werden.

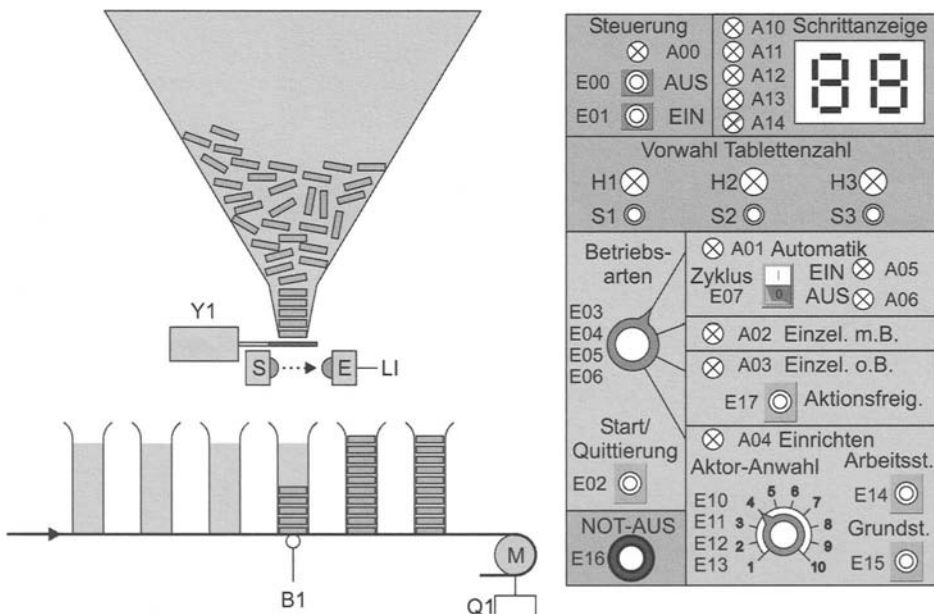
1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge der Anlage. (Hinweis: Ein- und Ausgänge des Bedienfeldes aus Beispiel 7.1 siehe Seite 101.)
2. Stellen Sie den Prozessablauf mit einem Ablauf-Funktionsplan dar.
3. Ermitteln Sie aus dem Ablauf-Funktionsplan die Transitionsbedingungen T-1 bis T-n.
4. Zeichnen Sie den Funktionsplan für die Aktionsausgabe.
5. Geben Sie die zur Programmierung des Aktionsbausteins erforderlichen Deklarationen an.
6. Programmieren Sie den Aktionsbaustein FC/FB26, rufen Sie die Bausteine FB24, FB25 und FC/FB26 vom OB1 aus auf und versehen Sie die Ein-/ Ausgänge der Bausteine mit entsprechenden SPS-Operanden der Zuordnungstabelle und Übergabevariablen.

Lernaufgabe 7.3: Tablettenabfülleinrichtung

Lös. S. 228

Aus einem Vorratstrichter soll eine bestimmte Anzahl von Tabletten in Röhrchen abgefüllt werden. Mit dem angegebenen Anzeige- und Bedienfeld kann die Abfülleinrichtung in verschiedenen Betriebsarten gefahren und die abzufüllende Tablettenzahl mit den Tastern S1 (N1 = 10), S2 (N2 = 20) bzw. S3 (N3 = 100) vorgewählt werden.

Technologieschema:



Der elektromagnetische Schieber Y1 wird im stromlosen Zustand durch Federkraft in seine Ausgangsstellung gezogen. Die Lichtschranke LI dient der Zählung der Tabletten. Die jeweils aktuell gewählte Tablettenzahl wird an den entsprechenden Leuchten H1 bis H3 angezeigt.

Funktionsablauf:

Nach Betätigung des Start-Tasters E02 in der eingestellten Betriebsart „Automatik“ beginnt der Abfüllvorgang, wenn eine bestimmte Tablettenzahl vorgewählt ist. Der Bandmotor M fördert das nächste leere Röhrchen unter die Abfüllstation (Meldung durch Sensor B1). Der elektromagnetische Schieber Y1 öffnet dann den Vorratsbehälter. Ist die eingestellte Tablettenzahl erreicht, schließt der Schieber die Abfüllstation und der Bandmotor wird gestartet, um das nächste Röhrchen unter die Station zu fördern. Dieser Ablauf wiederholt sich, bis der Zyklus-Schalter E07 ausgeschaltet wird.

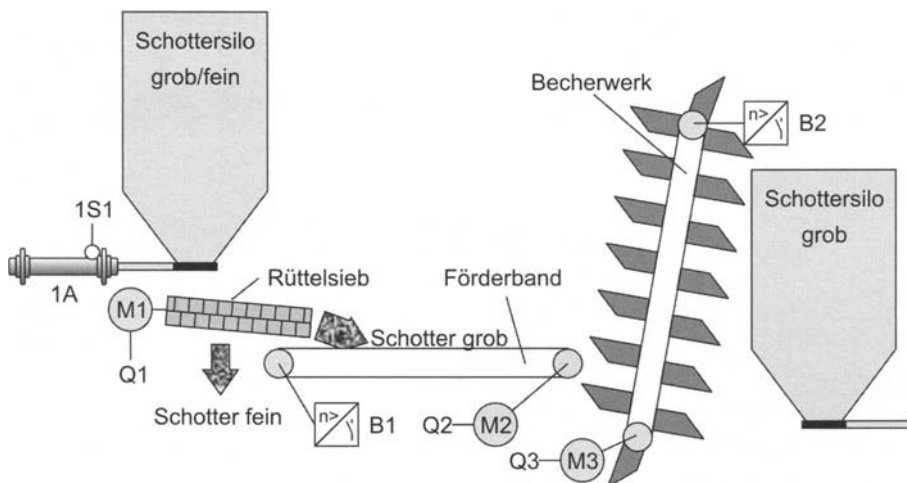
Wird eine andere Tablettenzahl durch Betätigen des entsprechenden Tasters S1, S2 bzw. S3 gewünscht, so ist ein gerade in Betrieb befindlicher Abfüllvorgang noch mit der alten Tablettenzahl zu beenden. Beim Ausschalten (BED_AUS = 1) wird die gespeicherte Vorwahl der Tablettenzahl gelöscht.

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge der Anlage. (Hinweis: Ein- und Ausgänge des Bedienfeldes aus Beispiel 7.1 siehe Seite 101.)
2. Stellen Sie den Prozessablauf mit einem Ablauf-Funktionsplan dar.
3. Ermitteln Sie aus dem Ablauf-Funktionsplan die Transitionsbedingungen T-1 bis T-n.
4. Zeichnen Sie den Funktionsplan für die Aktionsausgabe. Dieser soll auch die Speicherung der gewünschten Tablettenzahl und den Zähler beinhalten.
5. Geben Sie die zur Programmierung des Aktionsbausteins erforderlichen Deklarationen an.
6. Programmieren Sie den Aktionsbaustein FC/FB26, rufen Sie die Bausteine FB24, FB25 und FC/FB26 vom OB1 aus auf und versehen Sie die Ein-/ Ausgänge der Bausteine mit entsprechenden SPS-Operanden der Zuordnungstabelle und Übergabevariablen.

Lernaufgabe 7.4: Schotterwerk

Lös. S. 231

In einer Schotterwerkanlage soll der Schotter nach verschiedenen Größen ausgesiebt werden. Für den im Technologieschema angegebenen Teil der Anlage soll die Steuerung entworfen werden.

Technologieschema:

Mit dem angegebenen Anzeige- und Bedienfeld kann die Anlage in verschiedenen Betriebsarten gefahren werden.



Funktionsablauf:

Nach Betätigung des Start-Tasters E02 in der eingestellten Betriebsart „Automatik“ wird zunächst der Motor M3 des Becherwerks eingeschaltet. Meldet der Drehzahl-Sensor (B2=1), dass das Becherwerk läuft, schaltet der Förderbandmotor M2 ein. Läuft das Band mit der richtigen Geschwindigkeit (B1 = 1) schaltet der Motor M1 des Rüttelsiebs ein und durch Ansteuerung des federrückgestellten Ventils 1Y1 öffnet der hydraulische Schieber 1A das Silo.

Die Anlage verbleibt in diesem Zustand, bis entweder der Zyklus-Schalter E07 ausgeschaltet wird oder die Drehzahlwächter B1 bzw. B2 eine Störung melden. Der Schieber 1A schließt dann sofort das Schottersilo und der Motor für das Rüttelsieb wird ausgeschaltet. Nach Schließen des Silos läuft das Förderband noch fünf Sekunden weiter. Steht das Förderband still, wird nach einer Nachlaufzeit von neun Sekunden auch das Becherwerk abgeschaltet.

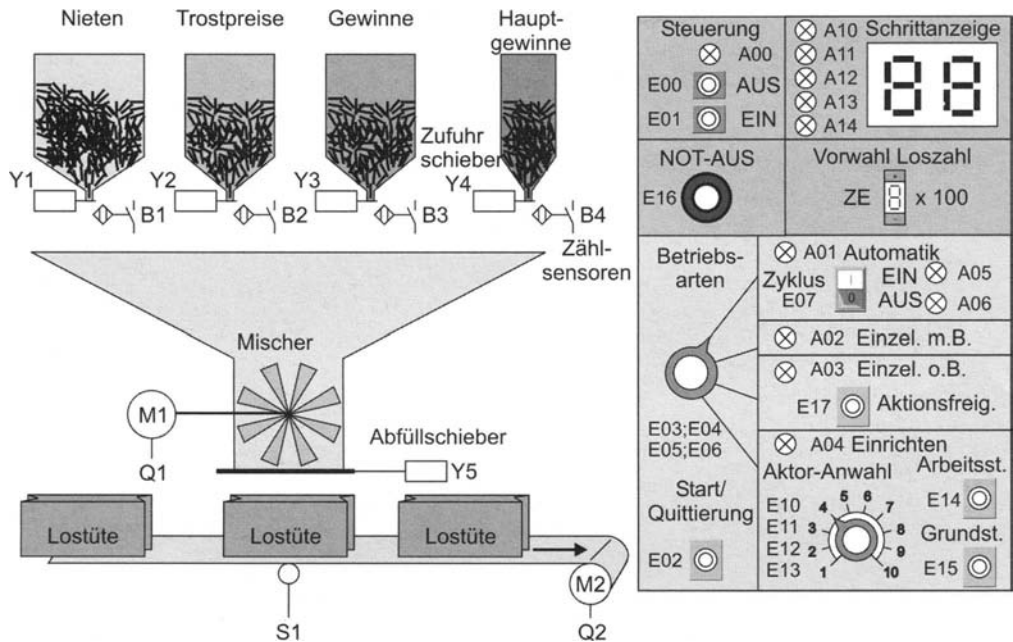
1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge der Anlage. (Hinweis: Ein- und Ausgänge des Bedienfeldes aus Beispiel 7.1 siehe Seite 101.)
2. Stellen Sie den Prozessablauf mit einem Ablauf-Funktionsplan dar.
3. Ermitteln Sie aus dem Ablauf-Funktionsplan die Transitionsbedingungen T-1 bis T-n.
4. Zeichnen Sie den Funktionsplan für die Aktionsausgabe.
5. Geben Sie die zur Programmierung des Aktionsbausteins erforderlichen Deklarationen an.
6. Programmieren Sie den Aktionsbaustein FC/FB26, rufen Sie die Bausteine FB24, FB25 und FC/FB26 vom OB1 aus auf und versehen Sie die Ein-/ Ausgänge der Bausteine mit entsprechenden SPS-Operanden der Zuordnungstabelle und Übergabevariablen.

Lernaufgabe 7.5: Los-Verpackungsanlage

Lös. S. 234

In einer Los-Verpackungsanlage werden Tüten für Losbuden mit der entsprechenden Anzahl von Losen abgefüllt. Vier verschiedene Losarten werden dabei gemischt: Hauptgewinne, Normalgewinne, Trostpreise und Nieten. Die Anteile der Losarten stehen stets in einem festen Verhältnis. Bei einer Gesamtzahl von 100 Losen sind enthalten: 62 Nieten, 25 Trostpreise, 12 Normalgewinne und 1 Hauptgewinn. Die gewünschte Anzahl der Lose pro Tüte kann am Zifferneinsteller als Vielfaches von 100 eingestellt werden. Bei der Einstellung 0 am Zifferneinsteller werden 1000 Lose in eine Tüte abgefüllt.

Mit dem angegebenen Anzeige- und Bedienfeld kann die Anlage in verschiedenen Betriebsarten gefahren und die Loszahl pro Abfüllvorgang eingestellt werden.

Technologieschema:

Alle elektromagnetischen Schieber Y1 bis Y5 werden im stromlosen Zustand durch Federkraft in ihre Ausgangslage gezogen.

Funktionsablauf:

Durch Betätigung des Start-Tasters E02 in der eingestellten Betriebsart „Automatik“ beginnt der Abfüllvorgang. Die am Zifferneinsteller vorgegebene Zahl wird übernommen und die nächste leere Lostüte unter den Abfüllstutzen des Mischbehälters gebracht. Die Zufuhrschieber öffnen die Vorratsbehälter mit den Losen. Melden die Zähler für die Impulse der Zählsensoren B1, B2, B3 bzw. B4 das Erreichen der entsprechenden Anzahl von Losen, wird der jeweilige Schieber geschlossen. Nach Schließen aller Schieber läuft der Mischer neun Sekunden. Danach öffnet der Abfüllschieber für fünf Sekunden und die Lose fallen in die Lostüten. Dieser Ablauf wiederholt sich, bis der Zyklus-Schalter E07 ausgeschaltet wird.

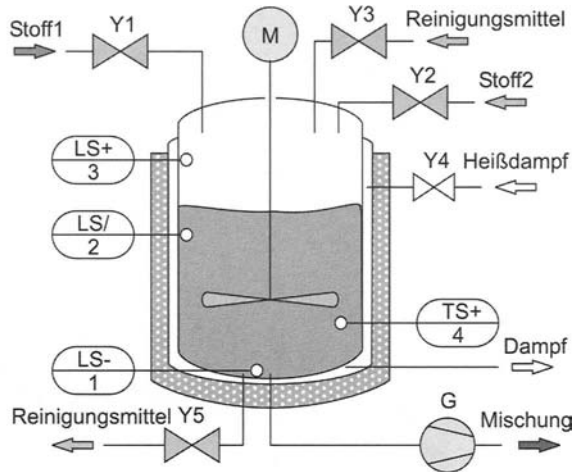
Wird die abzufüllende Loszahl am Zifferneinsteller während eines Abfüllvorgangs verändert, gilt diese erst für den nächsten Abfüllvorgang.

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge. (Hinweis: Ein- und Ausgänge des Bedienfeldes aus Beispiel 7.1 siehe Seite 101.)
2. Stellen Sie den Prozessablauf mit einem Ablauf-Funktionsplan dar.
3. Ermitteln Sie aus dem Ablauf-Funktionsplan die Transitionsbedingungen T-1 bis T-n.
4. Zeichnen Sie den Funktionsplan für die Aktionsausgabe. Die Auswertung des Zifferneinstellers sowie die Füllmengenberechnung der einzelnen Losarten ist ebenfalls in dem Aktionsbaustein auszuführen.
5. Geben Sie die zur Programmierung des Aktionsbausteins erforderlichen Deklarationen an.
6. Programmieren Sie den Aktionsbaustein FC/FB26, rufen Sie die Bausteine FB24, FB25 und FC/FB26 vom OB1 aus auf und versehen Sie die Ein-/ Ausgänge der Bausteine mit entsprechenden SPS-Operanden der Zuordnungstabelle und Übergabevariablen.

Lernaufgabe 7.6: Rührkessel

Lös. S. 238

In einem Rührkessel werden zwei Stoffe dosiert, miteinander vermischt und nach dem Erhitzen abgepumpt. Mit dem angegebenen modifizierten Bedienfeld soll der Prozess in den Betriebsarten: „Automatik“, „Einzelschritt mit Bedingungen“ und „Einzelschritt ohne Bedingungen“ gefahren werden. Auf die Betriebsart „Einrichten“ wird bei diesem Prozess verzichtet.

Technologieschema:**Prozessablauf:**

Nach Betätigung des Start-Tasters E3 in der eingestellten Betriebsart „Automatik“, wird bei Grundstellung der Anlage (Behälter leer) das Vorlaufventil Y1 solange geöffnet, bis der Niveauschalter LS/ anspricht. Danach schaltet das Rührwerk ein und das Ventil Y2 öffnet. Spricht Niveauschalter LS+ an, schließt das Ventil Y2 wieder und das Ventil Y4 für die Heißdampfzufuhr öffnet. Meldet der Temperatursensor TS+ das Erreichen der vorgegebenen Temperatur, schließt das Ventil Y4 und das Rührwerk schaltet ab. Mit der Pumpe G wird das Fertigprodukt aus dem Rührkessel abgepumpt. Zeigt der Niveauschalter LS- an, dass der Kessel leer ist, schaltet die Pumpe ab und ein anschließender Reinigungsprozess wird gestartet. Dazu öffnet das Ventil Y3 für den Zufluss des Reinigungsmittels solange, bis Niveauschalter LS+ anspricht. Bei eingeschaltetem Rührwerk wird der Kessel neun Sekunden gereinigt. Nach Ablauf des Reinigungsvorgangs öffnet das Ablassventil Y5. Ist der Behälter vollständig entleert (Niveauschalter LS-), kann ein weiterer Prozessablauf gestartet werden.

1. Bestimmen Sie die Zuordnungstabellen für das Bedienfeld und den Rührkessel.
2. Stellen Sie den Prozessablauf mit einem Ablauf-Funktionsplan dar.
3. Ermitteln Sie aus dem Ablauf-Funktionsplan die Transitionsbedingungen T-1 bis T-n.
4. Zeichnen Sie den Funktionsplan für die Aktionsausgabe.
5. Geben Sie die zur Programmierung des Aktionsbausteins erforderlichen Deklarationen an.
6. Programmieren Sie den Aktionsbaustein FC/FB26, rufen Sie die Bausteine FB24, FB25 und FC/FB26 vom OB1 aus auf und versehen Sie die Ein-/ Ausgänge der Bausteine mit entsprechenden SPS-Operanden der Zuordnungstabelle und Übergabevariablen.

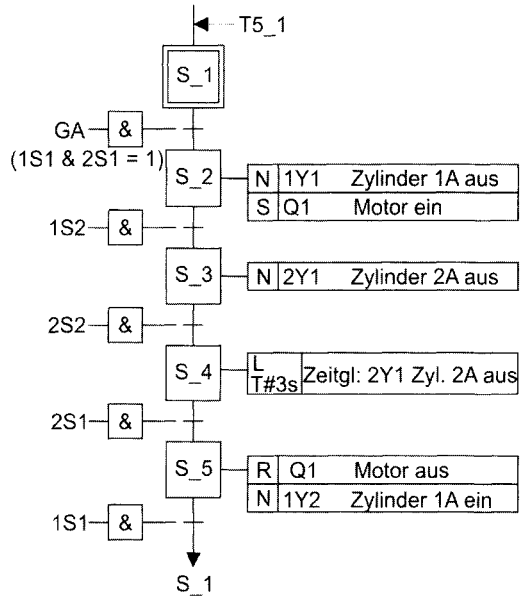
Hinweis: Der in Übersicht (7.1) vorgegebene Funktionsbaustein FB24 ist zu verwenden. Durch entsprechende Beschaltung kann dieser auf das veränderte Bedien- und Anzeigefeld angepasst werden.

7.3 Kontrollaufgaben

Kontrollaufgabe 7.1

Der nebenstehende Ablauf-Funktionsplan beschreibt den Prozessablauf einer Verpackungseinrichtung, welche mit dem Standard-Bedienfeld in verschiedenen Betriebsarten gefahren werden kann.

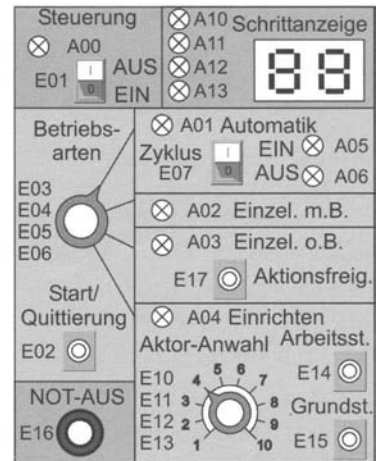
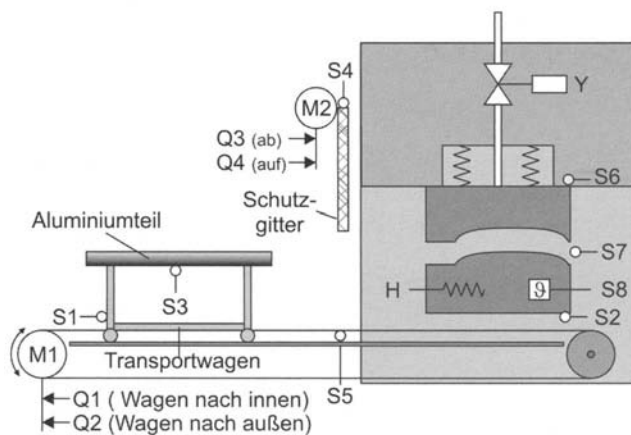
1. Ermitteln Sie aus der Schrittfolge die Transitionsbedingungen.
2. Zeichnen Sie den Funktionsplan des Aktionsbausteins.
3. Bestimmen Sie die Bausteinart für die Aktionsausgabe und geben Sie die für die Programmierung erforderlichen Deklarationen an.
4. Zeichnen Sie den Aufruf der Bausteine FB 25, FB24 und FC/FB 26 im OBI. Versehen Sie die Ein-/Ausgänge der Bausteine mit angenommenen SPS-Operanden.



Kontrollaufgabe 7.2

Mit der im Technologieschema gegebenen Anlage werden Aluminiumteile gebogen. Die Roh-teile müssen von Hand auf den Transportwagen gelegt werden.

Technologieschema:



Funktionsablauf:

Ist der Transportwagen beladen (S3) und in seiner Endlage (S1) sowie das Biegewerkzeug geöffnet (S6) und das Schutzgitter oben (S4), kann der Bediener durch Betätigung der Start-

taste E3 den Biegeprozess ablaufen lassen. Der Wagen wird durch Ansteuerung von Q1 mit Motor M1 in die Biegeeinrichtung bis zum Endschalter (S2) gezogen. Das Schutzgitter (Q3) fährt ab und die Heizung (H) schaltet ein. Erreicht die Temperatur der Biegeform den geforderten Wert (S8) und ist das Schutzgitter noch geschlossen (S5), beginnt das Biegen (Y). Befindet sich das Biegewerkzeug in der unteren Endlage (S7), ist die Heizung auszuschalten. Nach Ablauf der Biegezeit $T = 5$ s wird der Biegevorgang beendet und das Schutzgitter geöffnet (Q4). Sind Biegewerkzeug und Schutzgitter wieder in den oberen Endlagen (S4 und S6), zieht der Motor M1 durch Ansteuerung von Q2 den Transportwagen mit dem fertigen Formteil aus der Biegeform. Erreicht der Wagen die Endposition (S1), kann nach der Entladung der Vorgang wiederholt werden.

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Stellen Sie den Prozessablauf mit einem Ablauf-Funktionsplan dar.
3. Ermitteln Sie aus dem Ablauf-Funktionsplan die Transitionsbedingungen T-1 bis T-n.
4. Zeichnen Sie den Funktionsplan für die Aktionsausgabe.
5. Geben Sie die zur Programmierung des Aktionsbausteins erforderlichen Deklarationen an.
6. Zeichnen Sie den Aufruf der Bausteine FB 25, FB 24 und FC/FB 26 im OB1. Versehen Sie die Ein-/Ausgänge der Bausteine mit den angenommenen SPS-Operanden.

Kontrollaufgabe 7.3

Die in Kontrollaufgabe 7.2 gegebene Anlage zum Biegen von Aluminium-Teilen soll einem Kunden übergeben werden.

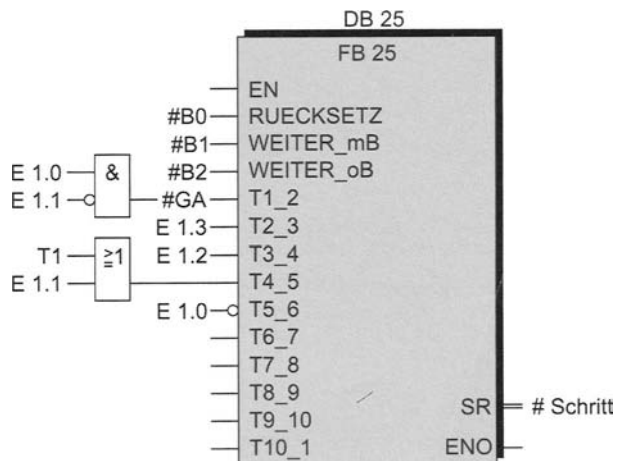
1. Formulieren Sie für die Dokumentation mit der Darstellung eines Ablauf-Funktionsplanes eine für den Kunden verständliche Beschreibung des Prozessablaufs.
2. Erläutern Sie die verschiedenen möglichen Betriebsarten, mit der die Anlage gefahren werden kann.
3. Erklären Sie für eine Bedienungsanleitung die Aufgaben jedes auf dem Anzeige- und Bedienfeld befindlichen Schalters bzw. Tasters.

Kontrollaufgabe 7.4

Aus dem nebenstehenden Aufruf des Standard-Funktionsbausteins FB25 für Ablaufketten mit Betriebsartenteil im OB1 sollen die Transitionsbedingungen einer Anlage ermittelt werden.

Die Anlage besitzt vier Sensoren S0 bis S3.

Mit dem Anzeige- und Bedienfeld kann die Anlage in verschiedenen Betriebsarten gefahren werden.

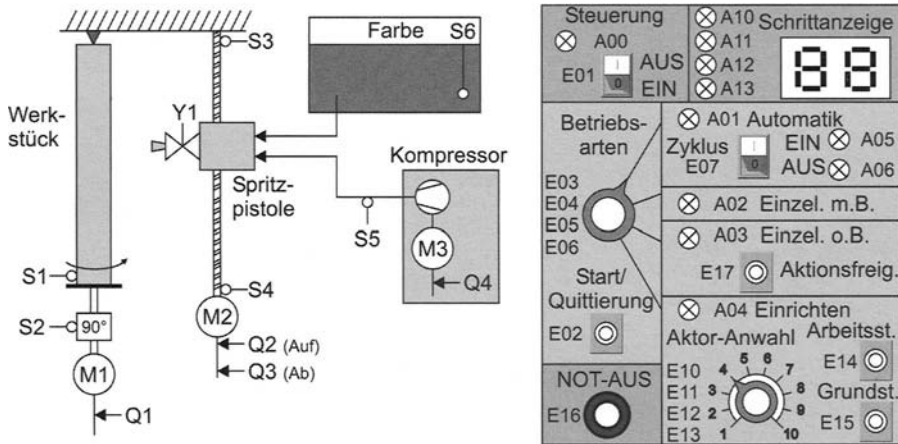


1. Zeichnen Sie für die Umsetzung der Schrittkette im Ablaufketten-Funktionsbaustein FB25 den Funktionsplan für Schritt 1 und Schritt 3
2. Ermitteln Sie aus dem Aufruf des FB 25 die Transitionsbedingungen T-1 bis T-n.
3. Zeichnen Sie den zugehörigen Ablauf-Funktionsplan ohne die Aktionen.

Kontrollaufgabe 7.5

Ein Werkstück soll auf einer Farbspritzmaschine auf vier Seiten mit einem Schutzlack überzogen werden.

Technologieschema:



Funktionsablauf:

Ist ein Werkstück in der Vorrichtung eingespannt (S1), die Spritzpistole in der oberen Endlage (S3) und genügend Farbe vorrätig (S6), kann der Bediener durch Betätigung der Starttaste E02 im Automatikbetrieb den Prozessablauf starten.

Der Kompressor läuft an und meldet mit S5 das Erreichen des erforderlichen Spritzdruckes. Das Ventil Y1 der Spritzpistole wird geöffnet. Der Motor M2 (Q3) fährt über eine Spindel die Pistole von oben nach unten über den gesamten Bereich des Werkstücks. Unten angekommen (S4), wird das Ventil Y1 geschlossen und das Werkstück mit M1 um 90° gedreht (Meldung mit S2). Danach öffnet das Ventil Y1 wieder und die Spritzpistole wird mit Q2 in die obere Endlage gefahren.

Dieser Vorgang wiederholt sich dann für die restlichen beiden Seiten des Werkstücks. Ein Unterschreiten der erforderlichen Farbmenge verhindert ein erneutes Starten des Prozessablaufs.

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Stellen Sie den Prozessablauf mit einem Ablauf-Funktionsplan dar.
3. Ermitteln Sie aus dem Ablauf-Funktionsplan die Transitionsbedingungen T-1 bis T-n.
4. Zeichnen Sie den Funktionsplan für die Aktionsausgabe.
5. Geben Sie die zur Programmierung des Aktionsbausteins erforderlichen Deklarationen an.
6. Zeichnen Sie den Aufruf der Bausteine FB 25, FB 24 und FC/FB 26 im OB1. Versehen Sie die Ein-/Ausgänge der Bausteine mit angenommenen SPS-Operanden.

Strukturen von Ablaufsteuerungen

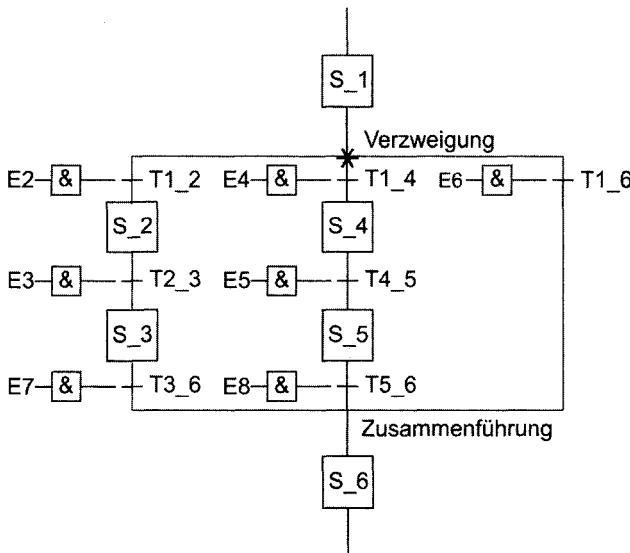
Die lineare Ablaufkette ist die einfachste Ablaufstruktur (s. Kap. 6 und 7). In komplexen Steuerungsaufgaben kommen Ablaufstrukturen mit mehreren Schrittkettensträngen, Kettensprüngen und Kettenschleifen vor.

Alternativverzweigung (1- aus n-Verzweigung)

An der Verzweigung erfolgt die Auswahl und Bearbeitung nur eines Schrittkettenstranges aus mehreren. Die Auswahl zwischen den Schrittkettensträngen wird durch die Transitionen nach der horizontalen Verzweigungslinie entschieden.

Der Stern (*) gibt an, dass die Bearbeitung der Transitionen von links nach rechts erfolgt. Damit hat bei gleichzeitiger Erfüllung mehrerer Transitionsbedingungen der weiter links liegende Schrittkettenstrang Vorrang.

Am Ende eines jeden Schrittkettenstranges muss eine Transitionsbedingung vorhanden sein. Wenn die Transitionsbedingung wahr und der letzte Kettenschritt aktiv ist, erfolgt der Übergang auf den nach der Zusammenführung folgenden Schritt.



Der Übergang von Schritt 1 nach Schritt 2 erfolgt, wenn Schritt 1 aktiv und die Transitionsbedingung T1_2 wahr ist.

Der Übergang von Schritt 3 nach Schritt 6 erfolgt, wenn Schritt 3 aktiv und die Transitionsbedingung T3_6 wahr ist.

Der Übergang von Schritt 1 nach Schritt 6 erfolgt, wenn Schritt 1 aktiv und die Transitionsbedingungen T1_2 und T1_4 nicht wahr sind und T1_6 wahr ist. Dieser Sonderfall eines Schrittkettenstranges ohne eigene Schritte wird als Kettensprung bezeichnet.

Transitionstabelle

Für die Realisierung der Transitionsbedingungen bestehen mehrere Möglichkeiten; Beispiele:

| Transition | Weiterschaltbedingung |
|------------|---|
| T-2 | $T1_2 = E2$ |
| T-4 | $T1_4 = E4 \& \overline{S_2}$ (Verriegelung mit S_2) |
| T-6 | $T1_6 = E6 \& \overline{S_2} \& \overline{S_4}$ (Verriegelung mit S_2 und S_4) $T3_6 = E6$ $T5_6 = E8$ |

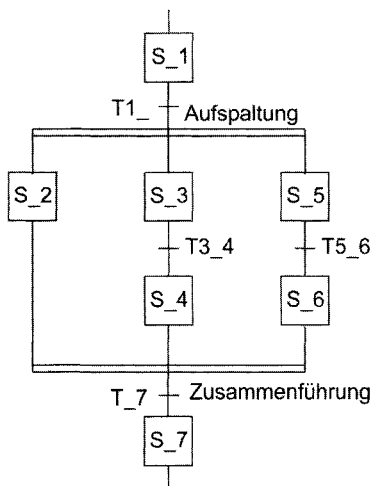
Simultanverzweigung (Parallelverzweigung)

Es erfolgt die gleichzeitige Aktivierung der Anfangsschritte mehrerer Schrittkettenstränge, die dann aber unabhängig voneinander bearbeitet werden.

Anfang und Ende von Simultanverzweigungen werden durch waagerechte Doppellinien dargestellt.

Alle Schrittkettenstränge unterliegen auf der Aufspaltungsseite oberhalb der waagerechten Doppellinie nur einer gemeinsamen Transitionsbedingung.

Auf der Zusammenführungsseite darf unter der waagerechten Doppellinie nur eine Transitionsbedingung für den Übergang auf den folgenden Schritt vorkommen.



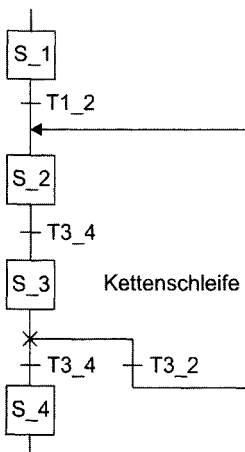
Der gleichzeitige Übergang von Schritt S_1 zu den Schritten S_2, S_3 und S_5 erfolgt, wenn Schritt S_1 aktiv und die Transitionsbedingung T1_ wahr ist.

Der gleichzeitige Übergang zum Schritt S_7 erfolgt, wenn die Schritte S_2 und S_4 und S_6 aktiv sind und die Transitionsbedingung T_7 wahr ist.

Die Aktivierung der Schritte S_4 und S_6 erfolgt unabhängig voneinander durch ihre Transitionsbedingungen T3_4 bzw. T5_6.

Kettenschleife

Eine Kettenschleife ermöglicht die mehrfache Wiederholung einer bestimmten Schrittfolge.



Der Übergang von Schritt S_3 nach Schritt S_2 findet statt, wenn Schritt S_3 aktiv und die Transitionsbedingung T3_2 wahr und die Transitionsbedingung T3_4 nicht wahr ist.

Ist T3_4 wahr und T3_2 nicht wahr, wird der Ablauf mit Schritt S_4 fortgesetzt.

Der Pfeil ist erforderlich, da Wirkverbindungen mit definitionsgemäßen Ablauf sonst von oben nach unten (und von links nach rechts) gerichtet sind.

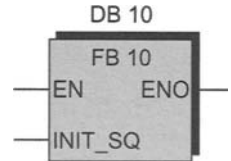
Bei unmittelbaren Schleifen (z.B. S_3 → S_4 → S_3) müssen die beiden Schritte jeweils mit dem Folgezustand und der negierten Setzbedingung des Folgezustandes zurückgesetzt werden.

Programmieren von verzweigten Ablaufsteuerungen

Für verzweigte Ablaufsteuerungen kann es leider keinen standardisierten Ablaufkettenbaustein geben. Professionell wird man komplexe Ablaufsteuerungen mit einem entsprechenden Programmierwerkzeug erstellen (z.B. GRAPH 7 bei STEP7).

Hinweis auf GRAPH 7:

Dieses Werkzeug wandelt den projektierten grafischen Ablauf-Funktionsplan beim Abspeichern automatisch in einen Funktionsbaustein um, der in der S7-CPU ablauffähig ist. Wahlweise steht ein komfortabler Betriebsartenteil zur Verfügung. **OB 1**

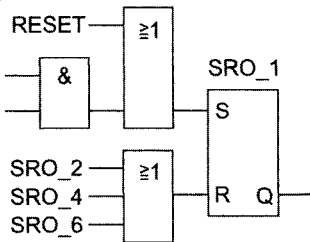


Für komplexe Ablaufketten ohne Betriebsartenteil ist die Einstellung „minimale FB-Parameter“ vorgesehen, bei der nur ein beschaltbarer Eingangsparameter INIT_SQ zur Ein-/Aus-Steuerung der Ablaufkette erzeugt wird.

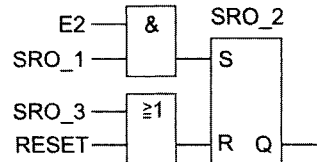
Die Verfügbarkeit eines solchen Programmierwerkzeugs wird hier nicht vorausgesetzt. Es wird vorgeschlagen, mit der elementaren RS-Speichermethode die verzweigten Ablaufketten der Lern- und Kontrollaufgaben in einem Funktionsbaustein FB 10 selbst zu erstellen und den erforderlichen Zeitaufwand hinzunehmen. Auf einen Betriebsartenteil wird deshalb bei diesen Aufgaben auch verzichtet.

Programmausschnitt für die Alternativverzweigung (siehe Seite 114):

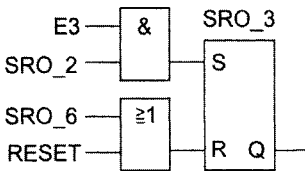
Schritt S_1



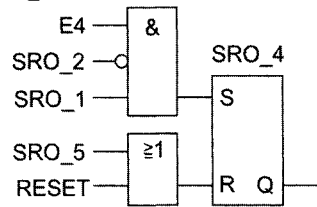
Schritt S_2



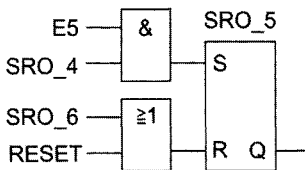
Schritt S_3



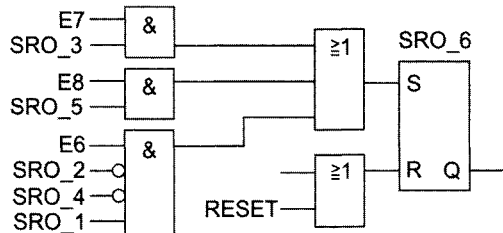
Schritt S_4



Schritt S_5



Schritt S_6



8.1

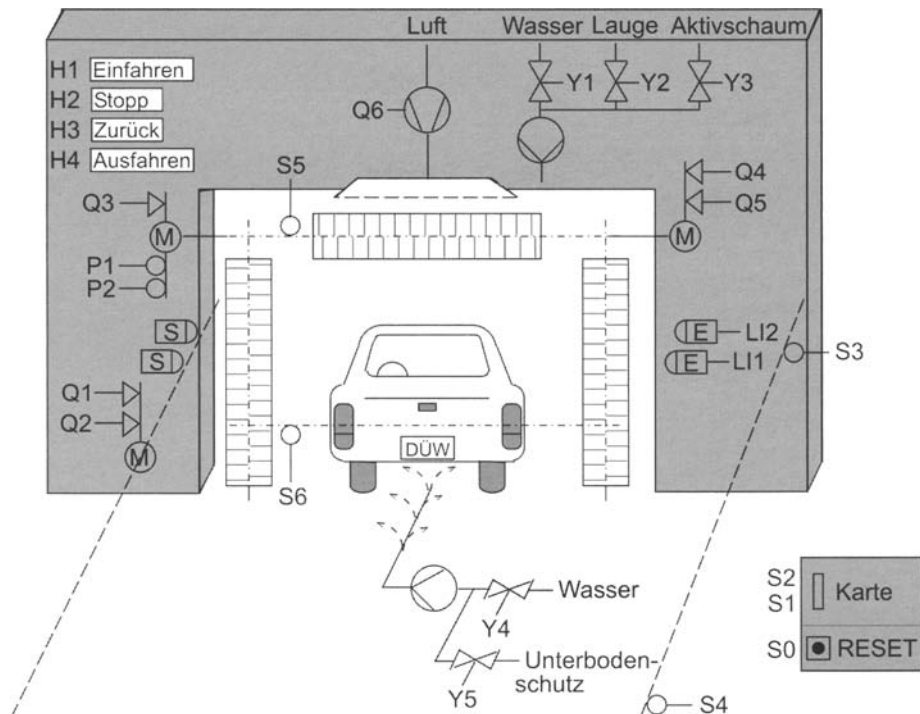
Beispiel**Ablaufkette mit Alternativ- und Simultanverzeigung****Autowaschanlage**

Die Anzeige H1 (Einfahren) zeigt die Betriebsbereitschaft der Waschanlage. Ein Auto kann bis zu der durch Lichtschranken kontrollierten Waschposition vorfahren, die beim Aufleuchten der Anzeige H2 (Stopp) erreicht ist. Wurde das Auto zu weit vorgefahren, leuchtet die Anzeige H3 (Zurück) auf. Durch Einstecken der Waschkarte wird das Programm gestartet. Nach Beendigung der Autowäsche erscheint die Anzeige H4 (Ausfahren).

Die Autowäsche besteht entsprechend der gesteckten Waschkarte aus einer Vorwäsche mit Aktivschaum oder ohne Vorwaschgang und einer Hauptwäsche mit parallel ablaufender Unterbodenwäsche, sowie dem Klarspülen und Trocknen. Bei der Vorwäsche und Hauptwäsche wird die Waschportal je einmal vor- und zurückgefahren. Eine weitere Vorwärtsfahrt wird zum Klarspülen und die anschließende Rückwärtsfahrt zum Trocknen genutzt. Die Vorwäsche besteht aus Einsprühen und Abspülen. Bei der Hauptwäsche mit Waschlauge werden die Waschbürsten entlang der Fahrzeugkontur geführt. Die Ansteuerung der beiden seitlichen Waschbürsten entspricht der Dachwaschbürstensteuerung und ist deshalb mit ihren Aktoren hier nicht ausgeführt. Bei Betätigung des AUS-Tasters wird die Ablaufkette in die Grundstellung gesetzt (RESET).

Aufgaben:

1. Entwurf eines Ablauf-Funktionsplans
2. Ablaufkette mit RS-Speichern und Aktionsausgabe im FB 10

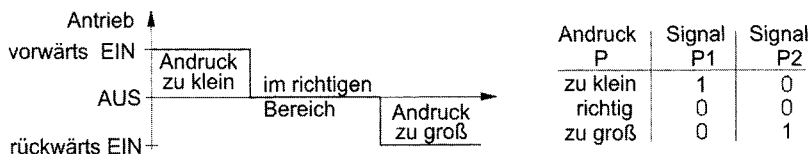
Technologieschema:

Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | | Adresse |
|------------------------------|--------|----------|--------------------|---------|---------|
| RESET | S0 | BOOL | Betätigt | S0 = 0 | E 0.0 |
| Waschkarte: Start | S1 | BOOL | Betätigt | S1 = 1 | E 0.1 |
| Waschkarte: Vorwäschewahl | S2 | BOOL | Mit Vorwäsche | S2 = 1 | E 0.2 |
| Endschalter Portal hinten | S3 | BOOL | Betätigt | S3 = 0 | E 0.3 |
| Endschalter Portal vorne | S4 | BOOL | Betätigt | S4 = 0 | E 0.4 |
| Endschalter Dachbürste oben | S5 | BOOL | Betätigt | S5 = 0 | E 0.5 |
| Endschalter Dachbürste unten | S6 | BOOL | Betätigt | S6 = 0 | E 0.6 |
| Lichtschranke 1 | L11 | BOOL | Unterbrochen | L11 = 1 | E 1.0 |
| Lichtschranke 2 | L12 | BOOL | Unterbrochen | L12 = 1 | E 1.1 |
| Andruck Dachrolle (min) | P1 | BOOL | Druck zu klein | P1 = 1 | E 1.2 |
| Andruck Dachrolle (max) | P2 | BOOL | Druck zu groß | P2 = 1 | E 1.3 |
| Ausgangsvariable | | | | | |
| Portalmotor vorwärts | Q1 | BOOL | Motor ein | Q1 = 1 | A 4.0 |
| Portalmotor rückwärts | Q2 | BOOL | Motor ein | Q2 = 1 | A 4.1 |
| Dachbürste rotieren | Q3 | BOOL | Motor ein | Q3 = 1 | A 4.2 |
| Dachbürste abwärts | Q4 | BOOL | Motor ein | Q4 = 1 | A 4.3 |
| Dachbürste aufwärts | Q5 | BOOL | Motor ein | Q5 = 1 | A 4.4 |
| Gebläsemotor | Q6 | BOOL | Motor ein | Q6 = 1 | A 4.5 |
| Klarwasserventil oben | Y1 | BOOL | Ventil auf | Y1 = 1 | A 4.6 |
| Waschlaugenventil | Y2 | BOOL | Ventil auf | Y2 = 1 | A 4.7 |
| Aktivschaumventil | Y3 | BOOL | Ventil auf | Y3 = 1 | A 5.0 |
| Klarwasserventil unten | Y4 | BOOL | Ventil auf | Y4 = 1 | A 5.1 |
| Unterbodenschutzventil | Y5 | BOOL | Ventil auf | Y5 = 1 | A 5.2 |
| Anzeige „Einfahren“ | H1 | BOOL | Anzeige an | H1 = 1 | A 5.3 |
| Anzeige „Stopp“ | H2 | BOOL | Anzeige an | H2 = 1 | A 5.4 |
| Anzeige „Zurück“ | H3 | BOOL | Anzeige an | H3 = 1 | A 5.5 |
| Anzeige „Ausfahren“ | H4 | BOOL | Anzeige an | H4 = 1 | A 5.6 |

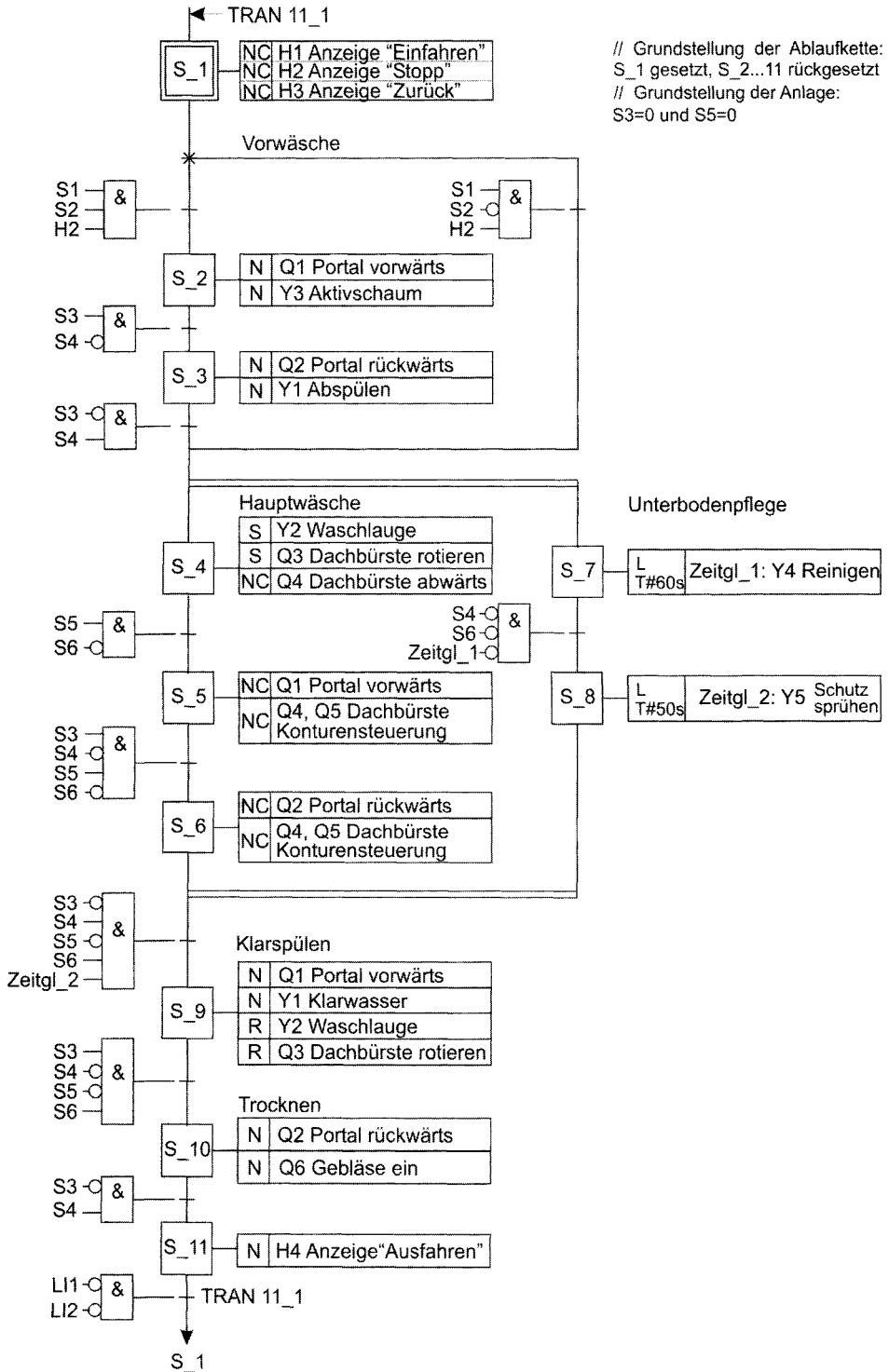
Ergänzung zur Funktionsbeschreibung

Das Heranfahren der Waschbürsten an die Fahrzeugflächen wird in der Praxis über die Auswertung der Wirkleistungsaufnahme der Bürstenmotoren gesteuert. Bei größerem Andruck erhöht sich die Wirkleistungsaufnahme der Motoren für die Rotation der Waschbürsten. Ein Wirkleistungs-Messumformer liefert die Messwerte an eine Elektronik, die daraus zwei binäre Ausgangssignale P1, P2 mit hier vereinfacht angenommener 3-stufiger Schaltlogik bildet.



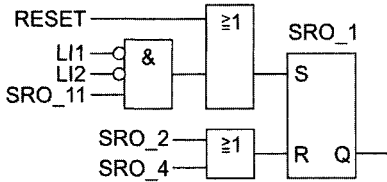
Die Positionssteuerung der Waschbürsten und die Bewegung des Waschportals müssen so ausgeführt werden, dass Kollisionen der Waschbürsten mit dem Auto vermieden werden. Im Funktions-Ablaufplan wird dies durch Eintrag eines zweiten Bestimmungszeichens (C = Conditional) hinter dem führenden Bestimmungszeichen ausgedrückt. So gekennzeichnete Aktionen, z.B. mit dem zusammengesetzten Bestimmungszeichen NC werden so ausgeführt, dass innerhalb des Ablaufschrittes auch bestimmte Verriegelungsbedingungen berücksichtigt werden.

1. Ablauf-Funktionsplan

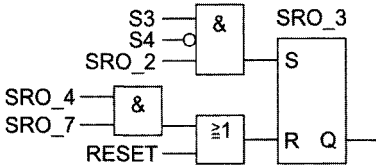


2. Funktionsplan: Ablaufkette

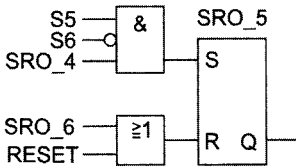
Schritt S_1



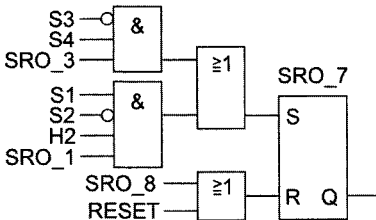
Schritt S_3



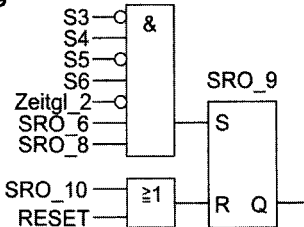
Schritt S_5



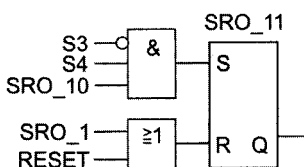
Schritt S_7



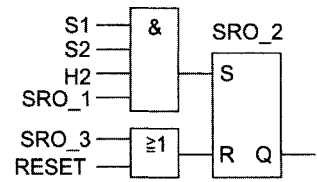
Schritt S_9



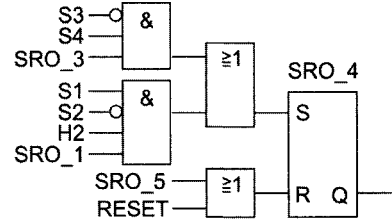
Schritt S_11



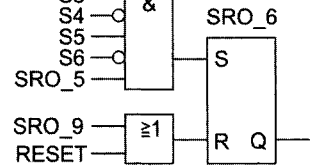
Schritt S_2



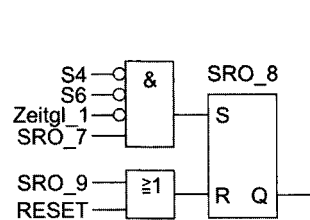
Schritt S_4



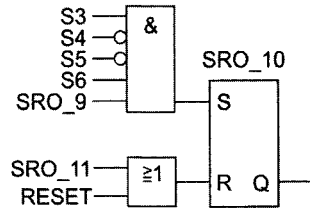
Schritt S_6



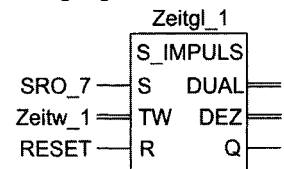
Schritt S_8



Schritt S_10

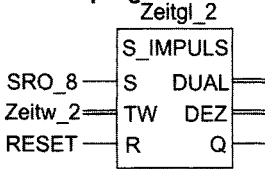


Zeit Unterbodenreinigung

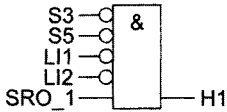


Funktionsplan: Befehlsausgabe

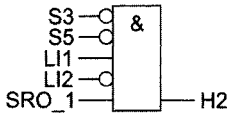
Zeit Unterbodenpflege



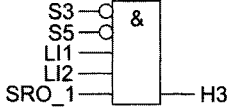
Einfahren



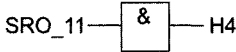
Stopp



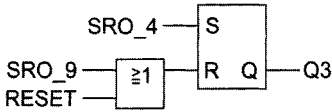
Zurück



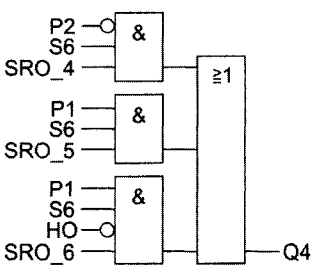
Ausfahren



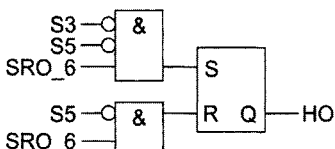
Dachbürsten rotieren



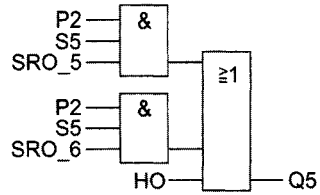
Dachbürste senken



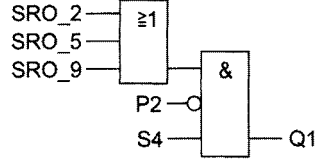
Vorbereitung Dachbürste Rückfahrt



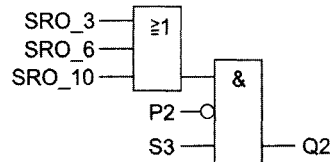
Dachbürste heben



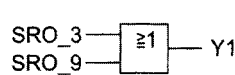
Portal vorwärts



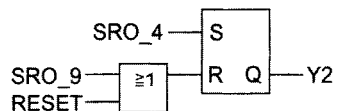
Portal rückwärts



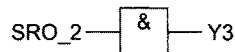
Klarwasserventil oben



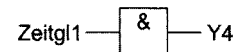
Waschlaugenventil



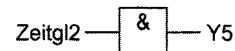
Aktivschaumventil



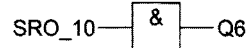
Klarwasserventil unten



Unterbodenschutzventil



Gebälse



Auf die Abbildung der Deklarationstabelle für den Funktionsbaustein sowie den Aufruf des Bausteins im OBI wird aus Umfangsgründen verzichtet und auf die beiliegende CD oder Internetadresse <http://www.vieweg.de> verwiesen. Siehe dort: STEP7-Lösungen → 8_1.

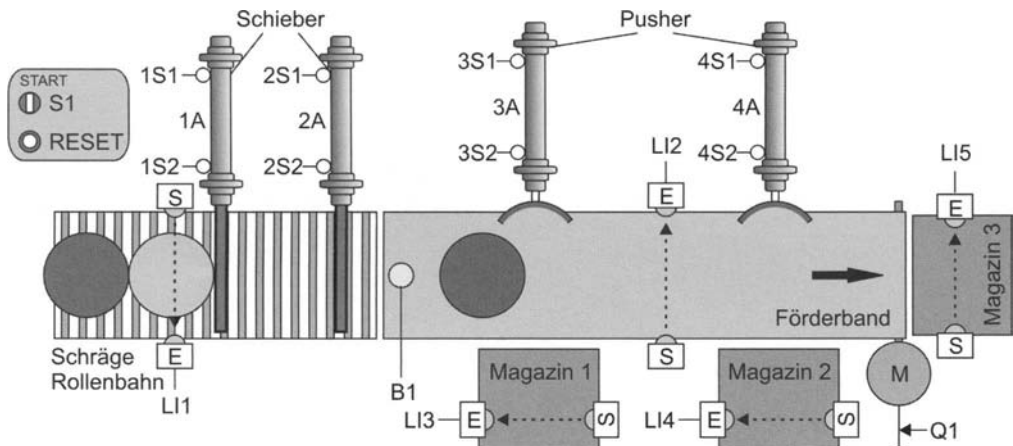
8.2 Lernaufgaben

Lernaufgabe 8.1: Sortieranlage

Lös. S. 242

Eine Sortieranlage soll Fertigungsteile nach Größe und Werkstoffart sortieren. Die Anlage besteht aus einer schrägen Rollenbahn mit zwei Schiebern zur Vereinzelung der Teile, einer Bandförderung für den Transport der Fertigungsteile und zwei Ausstoßvorrichtungen (Pusher). Die pneumatischen Zylinder der Schieber und Pusher werden über 5/2-Wegeventile mit beidseitiger elektromagnetischer Betätigung angesteuert.

Technologieschema:



Funktionsablauf:

Die Freigabe des Ablaufs erfolgt mit dem Schlüsselschalter S1. Befinden sich Teile auf der Rollenbahn (LI1 = 1), wird ein Fertigungsteil durch die beiden Schieber vereinzelnd und rutscht auf das Förderband. Ist der Schieber 2A wieder in der vorderen Endlage, läuft der Bandmotor M an. Meldet der Sensor B1 „1“-Signal, besteht das Fertigungsteil aus einem metallischen Werkstoff. Solche Teile werden durch Zylinder 3A in das Magazin 1 befördert. Dazu wird das Band zwei Sekunden nach Meldung durch Sensor B1 angehalten. Befindet sich der Zylinder 3A wieder in der hinteren Endlage, kann das nächste Fertigungsteil auf der Rollenbahn vereinzelnd werden.

Die Lichtschranke LI2 meldet, wenn ein Fertigungsteil auf dem Band eine bestimmte Größe überschreitet. Diese Teile werden durch Zylinder 4A in das Magazin 2 befördert. Dazu wird das Band drei Sekunden nach Meldung durch Lichtschranke LI2 angehalten. Nachdem sich Zylinder 4A wieder in der hinteren Endlage befindet, kann das nächste Fertigungsteil auf der Rollenbahn vereinzelnd werden.

Ist ein Fertigungsteil weder metallisch noch hat es eine bestimmte Größe, so wird es vom Band in das Magazin 3 transportiert. Meldet die Lichtschranke LI5, dass ein Teil in das Magazin 3 gefallen ist, hält das Band an und das nächste Fertigungsteil kann auf der Rollenbahn vereinzelnd werden.

Mit der RESET-Taste kann die Ablaufkette jederzeit in die Grundstellung versetzt werden.

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge der Sortieranlage.

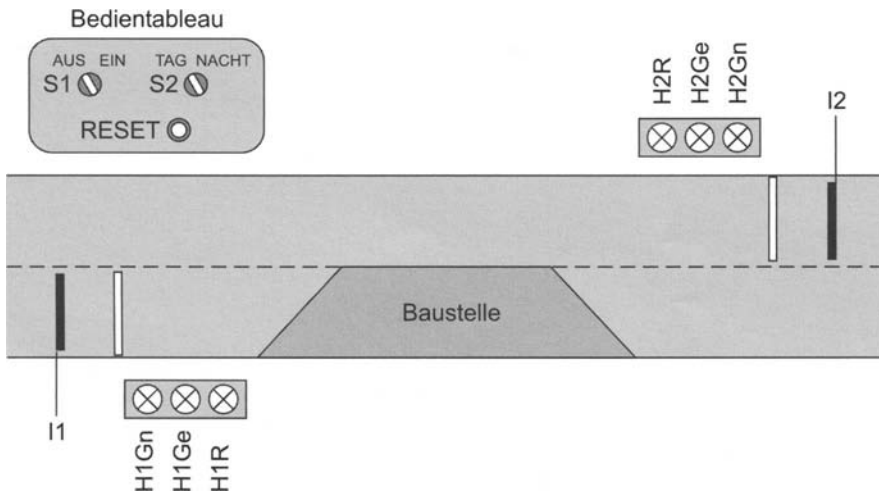
2. Geben Sie den Ablauf-Funktionsplan der Steuerungsaufgabe an.
3. Ermitteln Sie aus dem Ablauf-Funktionsplan alle Transitionsbedingungen T-1 bis T-n.
4. Zeichnen Sie den Funktionsplan für die Umsetzung der Ablaufkette in einem Baustein.
5. Geben Sie die für die Realisierung des Funktionsplans erforderlichen Deklarationen an.
6. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Ein-/Ausgänge des Bausteins mit SPS-Operanden der Zuordnungstabelle.

Lernaufgabe 8.2: Bedarfsampelanlage

Lös. S. 244

Die mobile Bedarfsampelanlage eines Straßenbaubetriebs ist SPS-gesteuert und wird an einer Baustelle bei erforderlicher einspuriger Verkehrsführung aufgestellt. Das nachfolgende Technologieschema zeigt die Anordnung der Geräte und das zugehörige Bedientableau.

Technologieschema:



Mit dem Schalter S1 kann die Anlage eingeschaltet werden. Steht der zweite Schalter S2 auf „Nachtbetrieb“, blinken die beiden gelben Signalleuchten der Ampeln mit einer Frequenz von 1 Hz. Steht der Schalter S2 auf „Tagbetrieb“ oder wird auf „Tagbetrieb“ umgeschaltet, zeigen beide Signalampeln Rot. Meldet einer der Initiatoren I1 bzw. I2 „1“-Signal, schaltet die entsprechende Ampel nach zehn Sekunden die gelbe Signalleuchte zusätzlich ein. Die Rot-Gelb-Phase dauert zwei Sekunden, bevor die Ampel auf Grün umschaltet.

Die Mindesteinschaltzeit der grünen Signalleuchte beträgt zwanzig Sekunden. Erst danach wird bei einer Betätigung des Initiators auf der anderen Seite eine Gelb-Phase von fünf Sekunden gestartet. Nach Ablauf der Gelbphase zeigen beide Ampeln zehn Sekunden wieder Rot bevor die entsprechende Seite mit zunächst Rot-Gelb und dann Grün bedient wird.

Liegt keine Meldung eines entsprechenden Initiators vor, bleibt die Ampelanlage in ihrem jeweiligen Zustand. Das Ausschalten der Anlage sowie der Wechsel vom „Tagbetrieb“ in den „Nachtbetrieb“ erfolgt erst, wenn beiden Ampeln Rot zeigen, wobei ein Übergang von Grün über eine Gelb-Phase führt. Im „Nachtbetrieb“ kann die Anlage direkt ausgeschaltet werden.

Mit der RESET-Taste kann die Ablaufkette in die Grundstellung versetzt werden.

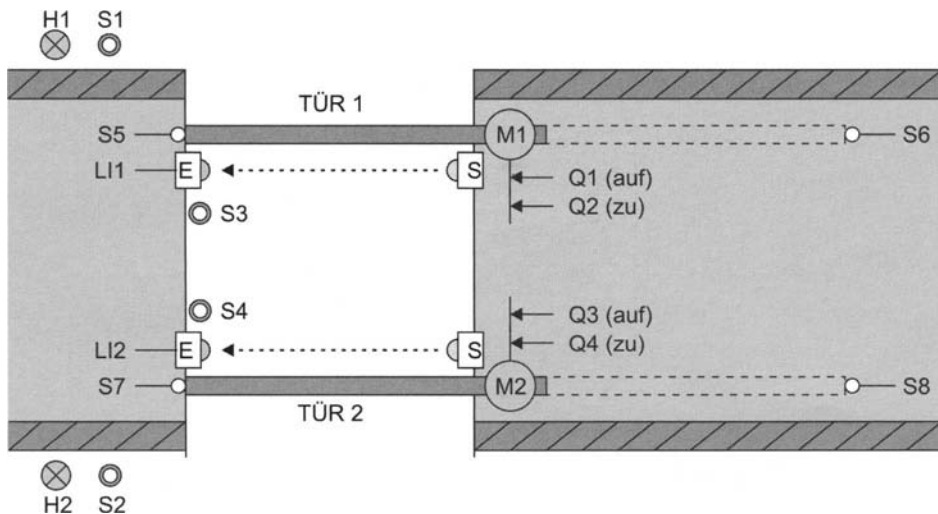
1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Geben Sie den Ablauf-Funktionsplan der Steuerungsaufgabe an.
3. Ermitteln Sie aus dem Ablauf-Funktionsplan alle Transitionsbedingungen T-1 bis T-n.
4. Zeichnen Sie den Funktionsplan für die Umsetzung der Ablaufkette in einem Baustein.
5. Geben Sie die für die Realisierung des Funktionsplans erforderlichen Deklarationen an.
6. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Ein-/Ausgänge des Bausteins mit SPS-Operanden der Zuordnungstabelle.

Lernaufgabe 8.3: Türschleuse

Lös. S. 245

Der Zutritt zu einem staubfreien Labor ist nur über eine Türschleuse möglich, welche aus zwei Schiebetüren besteht. Beide Schiebetüren dürfen unter keinen Umständen gleichzeitig geöffnet sein.

Technologieschema:



Wird beispielsweise Taster S1 zum Betreten des Labors betätigt, öffnet sich die Tür 1 und bleibt drei Sekunden offen, bevor sie wieder automatisch schließt. Danach öffnet sich Tür 2 und bleibt ebenfalls für drei Sekunden offen. Ein entsprechenden Ablauf gilt auch für das Verlassen des Labors.

An jeder Tür sind zwei induktive Endschalter angebracht, die mit einem „I“-Signal melden, wenn die Tür geöffnet bzw. geschlossen ist. Außerdem wird jeder Eingang der Schleuse mit einer Lichtschranke überwacht. Solange die Lichtschranke unterbrochen ist, darf eine geöffnete Tür nicht zugehen bzw. muss eine gerade zugehende Tür wieder geöffnet werden. Ebenso wird das Zugehen der Schleusentüren unterbrochen, wenn die Taster S1 oder S3 bzw. S2 oder S4 betätigt werden.

In der Schleuse sind zur Sicherheit zwei Taster S3 und S4 angebracht, mit denen nur die jeweils zugehörige Tür geöffnet werden kann, wenn beispielsweise jemand die Schleuse betritt, ohne zuvor den entsprechenden Taster S1 oder S2 betätigt zu haben. Dies ist denkbar, wenn der Betreffende eine offene Schleusentür vorfindet, da ein anderer gerade die Schleuse in der Gegenrichtung verlassen hat.

Mit der im Schaltschrank angebrachten RESET-Taste kann die Ablaufkette in die Grundstellung versetzt werden.

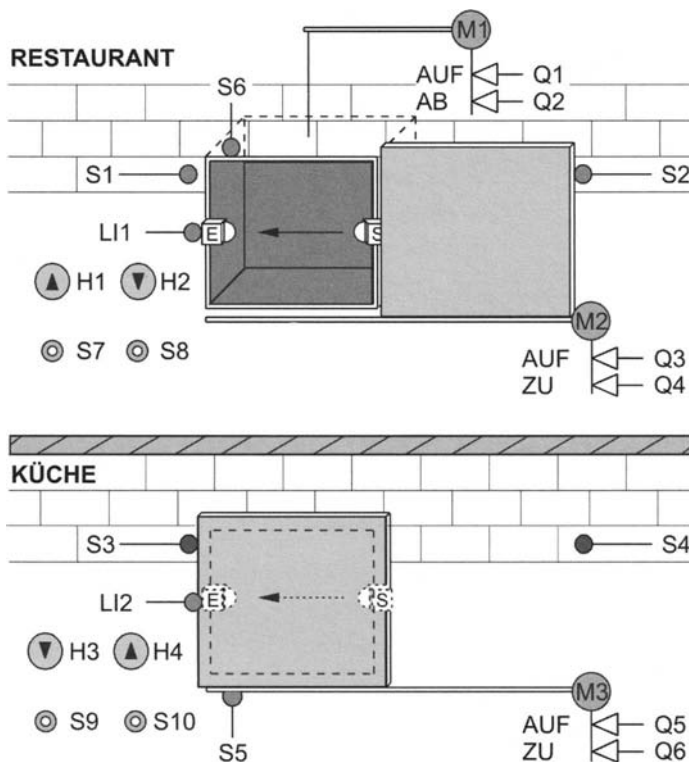
1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Geben Sie den Ablauf-Funktionsplan der Steuerungsaufgabe an.
3. Ermitteln Sie aus dem Ablauf-Funktionsplan alle Transitionsbedingungen T-1 bis T-n.
4. Zeichnen Sie den Funktionsplan für die Umsetzung der Ablaufkette in einem Baustein.
5. Geben Sie die für die Realisierung des Funktionsplans erforderlichen Deklarationen an.
6. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Ein-/Ausgänge des Bausteins mit SPS-Operanden aus der Zuordnungstabelle.

Lernaufgabe 8.4: Speiseaufzug

Lös. S. 247

Ein Speiseaufzug stellt die Verbindung von der im Keller gelegenen Küche zu dem im Erdgeschoss befindlichen Restaurant dar. In der Küche und im Restaurant sind automatische Türen und entsprechende Bedienelemente mit Anzeigen angebracht.

Technologieschema:



Das System Aufzugkorb mit Gegengewicht wird von dem Motor $M1$ mit umschaltbarer Drehrichtung angetrieben. Hierfür sind die beiden Leistungsschütze $Q1$ bzw. $Q2$ anzusteuern. Sowohl in der Küche, wie auch im Restaurant, sind je zwei Ruftaster angebracht. Mit $S7$ oder $S9$ kann der Fahrkorb jeweils geholt werden und mit $S8$ oder $S10$ kann der Korb in das jeweils andere Stockwerk geschickt werden. Die zu den Ruftastern gehörenden Anzeigeleuchten mel-

den, dass die Steuerung den Tastendruck bearbeitet und zeigen die gewünschte Fahrrichtung an.

Die Türen zum Aufzugsschacht werden automatisch geöffnet, wenn der Fahrkorb in dem entsprechenden Stockwerk steht. Hierzu werden die beiden Türmotoren M2 bzw. M3 über die Leistungsschütze Q3, Q4 bzw. Q5, Q6 in zwei Drehrichtungen betrieben. Die Mindestöffnungszeit einer Tür beträgt 3 Sekunden. Liegt keine Bedarfsanforderung vor, bleibt die Tür in dem Stockwerk, in dem sich der Fahrkorb befindet, geöffnet. Die Türöffnungen werden mit den Lichtschranken LI1 und LI2 überwacht. Wird während des Schließens der Tür die Lichtschranke unterbrochen oder einer der entsprechenden Taster S7 bzw. S9 betätigt, öffnet sich die Tür sofort wieder.

Da beim Einschalten der Steuerung der Anlagenzustand nicht bekannt ist, wird der Speiseaufzug durch eine Referenzfahrt in eine definierte Ausgangsstellung gebracht. Dabei werden beide Türen zunächst geschlossen, der Förderkorb in die Küche gefahren und die dortige Tür geöffnet.

Mit der im Schaltschrank angebrachten RESET-Taste kann die Ablaufkette in die Grundstellung versetzt werden.

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Geben Sie den Ablauf-Funktionsplan der Steuerungsaufgabe an.
3. Ermitteln Sie aus dem Ablauf-Funktionsplan alle Transitionsbedingungen T-1 bis T-n.
4. Zeichnen Sie den Funktionsplan für die Umsetzung der Ablaufkette in einem Baustein.
5. Geben Sie die für die Realisierung des Funktionsplans erforderlichen Deklarationen an.
6. Programmieren Sie den Baustein, rufen Sie diesen vom OB1 aus auf und versehen Sie die Ein-/Ausgänge des Bausteins mit SPS-Operanden der Zuordnungstabelle.

Lernaufgabe 8.5: Chargenprozess

Lös. S. 249

Zur Herstellung eines bestimmten Kunststoffes arbeiten zwei Vorlagebehälter für Einsatzstoffe mit einem Mischkessel zusammen. In den Vorlagebehältern erfolgt das Dosieren und Aufheizen der Rohprodukte. Je zwei Chargen von Vorlagebehälter 1 und Vorlagebehälter 2 werden im Mischkessel gesammelt, auf die Reaktionstemperatur gebracht und eine vorgegebene Zeit gemischt. Nach einer Abkühlung auf 20 °C ist das gewünschte Produkt fertig.

Prozessablauf Vorlagebehälter:

Die zwei Vorlagebehälter beginnen ihren Betrieb nach Betätigung der Start-Taste S1. Da die Behälter gleichzeitig betrieben werden, gilt die Beschreibung für beide.

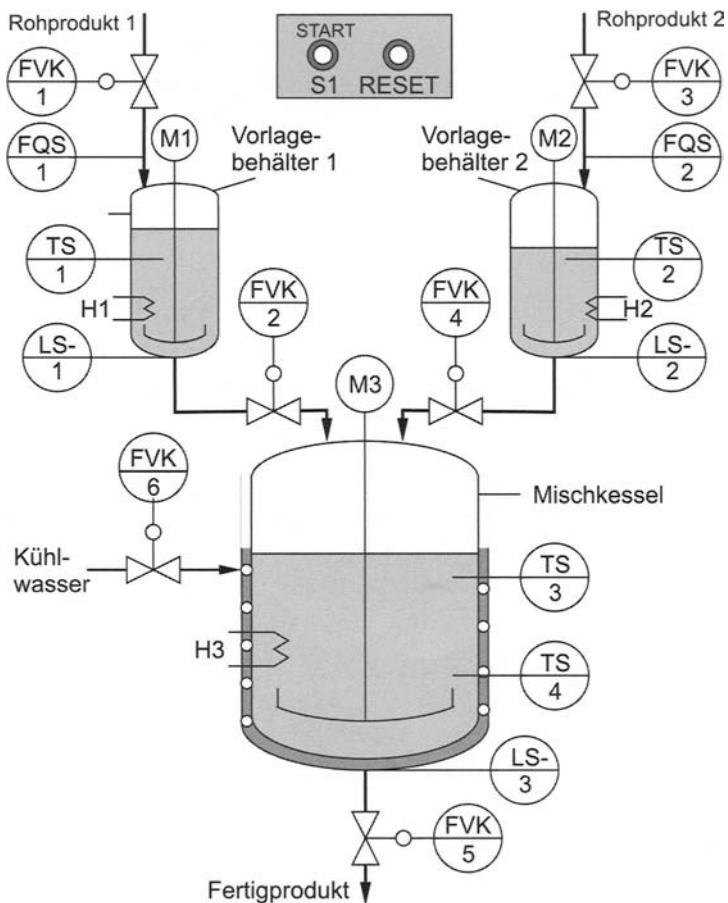
Zunächst wird das Einlassventil FVK1 (FVK3) für die Zufuhr des Einsatzstoffes geöffnet. Meldet der Dosierzähler FQS1 (FQS2) das Erreichen der eingestellten Dosiermenge, wird das Einlassventil wieder geschlossen und die Heizung H1 (H2) eingeschaltet. Erreicht die Temperatur im Vorlagebehälter den eingestellten Wert, gibt der Temperatursensor TS1 (TS2) ein Signal und der Rührwerkmotor M1 (M2) ist einzuschalten. Die Füllungen sind nach Einschalten des Rührwerks und eingeschalteter Heizung bei Vorlagebehälter 1 nach 15 Sekunden und bei Vorlagebehälter 2 nach 20 Sekunden fertig. Bei dem zuerst fertigen Vorlagebehälter wird die Heizung abgeschaltet, während das Rührwerk jedoch weiterläuft. Erst wenn in beiden Vorlagebehältern die Chargen fertig sind, werden die Auslassventile FVK2 bzw. FVK4 geöffnet und das Rührwerk ausgeschaltet. Bei leerem Behälter (LS1 = 1 bzw. LS2 = 1) ist das Aus-

lassventil wieder zu schließen. Sind beide Vorlagebehälter vollständig entleert, kann mit der Füllung der jeweils zweiten Charge begonnen werden.

Prozessablauf Mischkesselbetrieb:

Bei der Befüllung des Mischkessels mit der ersten Charge wird das Rührwerk (M3) eingeschaltet. Ist die Befüllung beendet, schaltet die Heizung H3 ein. Nach Erreichen der erforderlichen Temperatur im Mischkessel ($TS_3 = 1$) soll das Rührwerk noch mindestens 10 Sekunden laufen, bevor die Befüllung mit der zweiten Charge gestartet wird. Ist die zweite Befüllung vollständig durchgeführt, muss die gesamte Mischung bei immer noch eingeschalteter Heizung 25 Sekunden gerührt werden. Danach ist die Heizung H3 auszuschalten und der Kühlkreislauf FVK6 einzuschalten. Erreicht die Mischung dann die Temperatur von 20 °C ($TS_4 = 1$) ist das Produkt fertig. Das Rührwerk M3 und der Kühlkreislauf FVK6 sind abzuschalten und das Auslassventil FVK5 ist zu öffnen. Ist der Mischkessel leer, wird das Auslassventil wieder geschlossen und der gesamte Prozess kann mit der Start-Taste S1 erneut gestartet werden.

Technologieschema:



Legende: FVK = binäres Stellventil (F = Durchfluss, V = Stellgerätfunktion, K = binär); FQS = Durchflussmengenmesser (FQ = Durchflussmenge, S = Frequenz); TS = Temperatursensor (T = Temperatur, S = Schaltung); LS = Standmessung (L = Stand, S = Schaltung).

Lösungshinweise:

Die Umsetzung des Prozessablaufs in einem verzweigten Ablauf-Funktionsplan führt wegen der Vielzahl der Ablaufschritte zu einer unübersichtlichen Darstellung. Ein anderer Lösungsansatz besteht darin, den gesamten Chargenprozess in die drei Funktionseinheiten; Mischkessel, Vorlagebehälter 1 und Vorlagebehälter 2 zu untergliedern.

Für jede dieser Funktionseinheiten ist eine Schrittkette zu entwickeln. Die Schrittkette für den Vorlagebehälter 1 (SK3) und die Schrittkette für den Vorlagebehälter 2 (SK4) unterscheiden sich dabei aufgabenbedingt nur durch die zum jeweiligen Behälter gehörenden Sensoren bzw. Aktoren und den vorgegebenen Rührzeiten.

Die Abstimmung der drei Schrittketten und somit die Steuerung des Gesamtablaufs übernimmt eine Koordinations-Schrittkette (SK1). Diese enthält als Aktionen nur Transitionen für die anderen Schrittketten und die Transitionen werden durch das Erreichen bestimmter Ablaufschritte der Schrittketten SK2 SK3 und SK4 für die Funktionseinheiten gebildet.

Da sich jeweils nur lineare Schrittketten für die Funktionseinheiten und die Koordinations-Schrittkette (SK1) ergeben, kann bei der Realisierung des Steuerungsprogramms für den Chargenbetrieb der Standard-Funktionsbaustein FB15 aus Kapitel 6 verwendet werden.

Mit der RESET-Taste können alle Schrittketten in die Grundstellung versetzt werden.

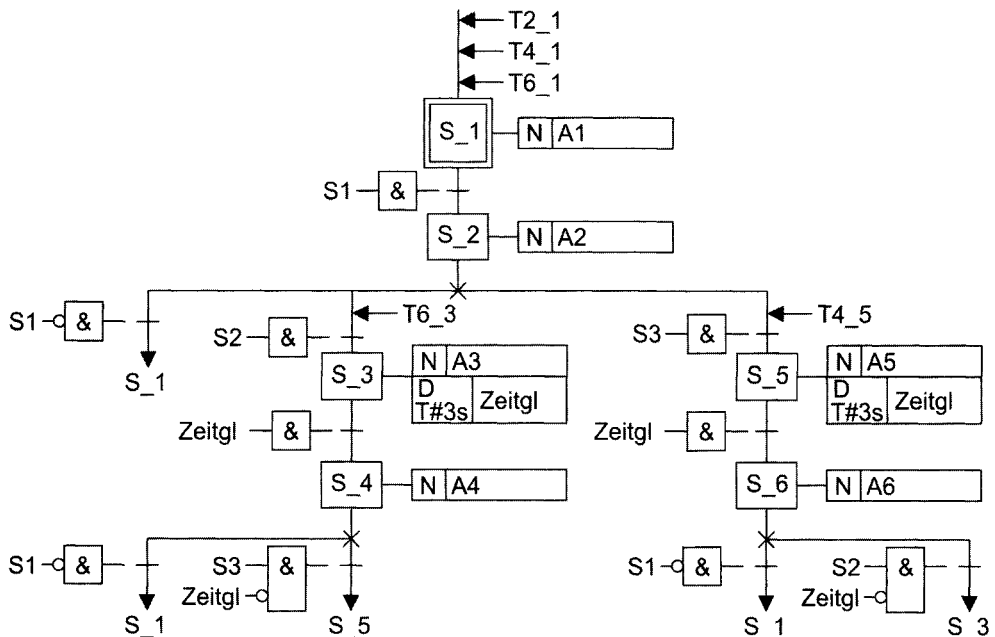
1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Stellen Sie den Prozessablauf mit einer Koordinations-Schrittkette und den Schrittketten für die Funktionseinheiten Mischkessel, Vorlagebehälter 1 und Vorlagebehälter 2 dar.
3. Zeichnen Sie den Funktionsplan für die Aktionsausgabe.
4. Geben Sie die zur Programmierung des Aktionsbausteins erforderlichen Deklarationen an.
5. Programmieren Sie den Aktionsbaustein FC 16, rufen Sie die Bausteine FB 15 (4x) und FC16 vom OB1 aus auf und versehen Sie die Ein-/Ausgänge der Bausteine mit entsprechenden Übergabevariablen und SPS-Operanden der Zuordnungstabelle.

Hinweis: Zum Test des Steuerungsprogramms bei der Erstinbetriebnahme ist es sehr hilfreich, den jeweils aktuellen Schritt jeder Schrittkette zu verfolgen. Dies ist möglich, wenn im OB1 die Übergabevariablen SK1 bis SK4 an Ausgangsvariable AW6 bis AW12 mit der MOVE-Funktion übergeben werden. Nach Feststellung der Richtigkeit des Steuerungsprogramms, können dann die Zuweisungen wieder entfernt werden.

8.3 Kontrollaufgaben

Kontrollaufgabe 8.1

Der gegebene Ablauf-Funktionsplan beschreibt die Funktionsweise einer Steuerungsaufgabe mit drei Eingängen S1 bis S3 und sechs Ausgängen A1 bis A6. Mit einem weiteren Eingang S0 kann die Ablaufkette jederzeit in die Grundstellung gebracht werden.



1. Ermitteln Sie aus dem Ablauf-Funktionsplan alle Transitionsbedingungen T-1 bis T-6.
2. Zeichnen Sie den Funktionsplan für die Umsetzung der Ablaufkette.
3. Geben Sie die für die Realisierung des Funktionsplans erforderlichen Deklarationen an.
4. Zeichnen Sie den Aufruf des Funktionsbausteins FB 10 im OB1 und versehen Sie die Eingänge und Ausgänge des Bausteins mit angenommenen SPS-Operanden.

Kontrollaufgabe 8.2

Das nachfolgend als AWL-Quelle angegebene Steuerungsprogramm des Funktionsbausteins FB10 stellt die Realisierung einer Ablaufsteuerung dar.

AWL-Quelle:

```

FUNCTION FB 10
VAR_INPUT
  S1 : BOOL ;
  S2 : BOOL ;
  S3 : BOOL ;
  S4 : BOOL ;
  RESET : BOOL ;
END_VAR
VAR_OUTPUT
  A1 : BOOL ;
  A2 : BOOL ;
  A3 : BOOL ;
  A4 : BOOL ;
  A5 : BOOL ;
  A6 : BOOL ;
  A7 : BOOL ;
END_VAR
VAR
  SRO_1 : BOOL := TRUE;
  SRO_2 : BOOL ;
  SRO_3 : BOOL ;
  SRO_4 : BOOL ;
  SRO_5 : BOOL ;
  SRO_6 : BOOL ;
  SRO_7 : BOOL ;
END_VAR

```

```

BEGIN          U #S2;          O #SRO_6;      U #SRO_1;
O #RESET;     UN #S4;          O #SRO_1;     = #A1;
O ;           U #SRO_2;      O #RESET;
UN #S1;       S #SRO_3;      R #SRO_5;      U #SRO_2;
U #SRO_4;     O #SRO_4;          O #A2;
O ;           O #RESET;      U #S3;
UN #S1;       R #SRO_3;      U #SRO_5;      U #SRO_3;
U #SRO_6;     S #SRO_6;          = #A3;
O ;           U #S3;          O #SRO_1;
UN #S1;       U #SRO_3;      O #SRO_7;      U #SRO_4;
U #SRO_7;     S #SRO_4;      O #RESET;      = #A4;
S #SRO_1;     O #SRO_1;      R #SRO_6;
U #SRO_2;     O #SRO_5;          U #SRO_5;
R #SRO_1;     O #RESET;      U #S2;          = #A5;
              R #SRO_4;      U #S4;
U #S1;        U #SRO_2;      U #SRO_2;      U #SRO_6;
U #SRO_1;     UN #S2;      O ;            = #A6;
S #SRO_2;     U #S4;          U #S2;
O #SRO_3;     U #SRO_2;      U #SRO_6;      U #SRO_7;
O #SRO_5;     O ;            S #SRO_7;      = #A7;
O #SRO_7;     O #S4;          O #SRO_1;
O #RESET;     U #SRO_4;      O #RESET;
R #SRO_2;     S #SRO_5;      R #SRO_7;

```

1. Zeichnen Sie den Aufruf des Funktionsbausteins FB10 im OB1 in der Funktionsplandarstellung und versehen Sie die Ein- und Ausgänge des Bausteins mit angenommenen SPS-Operanden.
2. Bestimmen Sie den Funktionsplan, der der AWL-Quelle entspricht.
3. Ermitteln Sie die Transitionsbedingungen T-1 bis T-7.
4. Zeichnen Sie den zu der Anweisungsliste gehörenden Ablauf-Funktionsplan.

Kontrollaufgabe 8.3

Die nachfolgende Transitionstabelle beschreibt die Weiterschaltbedingungen eines Ablauf-Funktionsplanes. In jedem Ablaufschritt wird dabei als Aktion einem Ausgang A1 bis A6 nichtspeichernd ein „1“-Signal zugewiesen. Mit der RESET-Taste kann die Ablaufkette in die Grundstellung versetzt werden.

Transitionstabelle:

| Transition | Weiterschaltbedingungen |
|------------|---|
| T-1 | $T4_1 = \overline{S1}$ $T6_1 = S1$ |
| T-2 | $T1_2 = S1$ |
| T-3 | $T2_3 = S2 \& \overline{S3}$ $T5_3 = S2$ $T6_3 = S2$ |

| Transition | Weiterschaltbedingungen |
|------------|---|
| T-4 | $T3_4 = S4$ |
| T-5 | $T2_5 = \overline{S2} \& S3$ $T3_5 = S3$ $T4_5 = S3$ |
| T-6 | $T5_6 = S4$ |

1. Zeichnen Sie den zur Transitionstabelle gehörenden Ablauf-Funktionsplan.
2. Zeichnen Sie den Funktionsplan für die Umsetzung der Ablaufkette.
3. Geben Sie die für die Realisierung des Funktionsplans erforderlichen Deklarationen an.
4. Zeichnen Sie den Aufruf des Funktionsbausteins FB 10 im OB1 und versehen Sie die Eingänge und Ausgänge des Bausteins mit angenommenen SPS-Operanden.

Kontrollaufgabe 8.4

Der Ölbrenner aus einer Heizungsanlage besteht aus einem Motor, der das Gebläse und die Ölpumpe antreibt, einem Magnetventil, welches die Ölzufuhr von der Pumpe zur Düse freigibt und einer Zündeinrichtung, die mit Hilfe eines Hochspannungstransformators einen Lichtbogen unmittelbar vor der Düse erzeugt.

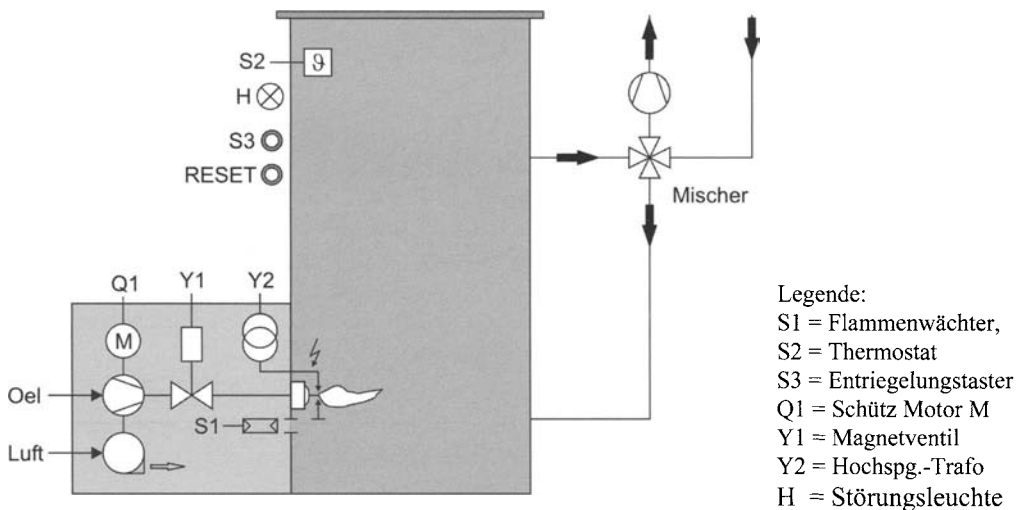
Die Ölbrennersteuerung soll den Motor mit Gebläse und Ölpumpe einschalten, wenn der Thermostat anspricht, weil die eingestellte Kesseltemperatur unterschritten wird.

Nach einer Vorbelüftungszeit von drei Sekunden wird durch das Magnetventil die Ölzufuhr freigegeben und gleichzeitig die Zündung eingeschaltet. Die Zündung wird sofort wieder ausgeschaltet, sobald der Flammenwächter das Entstehen der Flamme meldet. Ist die am Thermostat eingestellte Kesseltemperatur erreicht, müssen Motor und Magnetventil ausgeschaltet werden.

Wenn trotz eingeschalteter Zündung keine Flamme entsteht, liegt eine Störung vor. Nach einer Sicherheitszeit von neun Sekunden muss die Ölzufuhr gesperrt, der Motor abgeschaltet und ein Alarm (Störungsleuchte) ausgelöst werden. Nach einem Alarmzustand kann die Anlage von Hand durch Betätigen eines Entriegelungstasters wieder in Betrieb genommen werden. Erlischt die Flamme während des Brennerbetriebs, so muss die Zündung automatisch eingeschaltet werden.

Mit der RESET-Taste kann die Ablaufkette jederzeit in die Grundstellung versetzt werden.

Technologieschema:



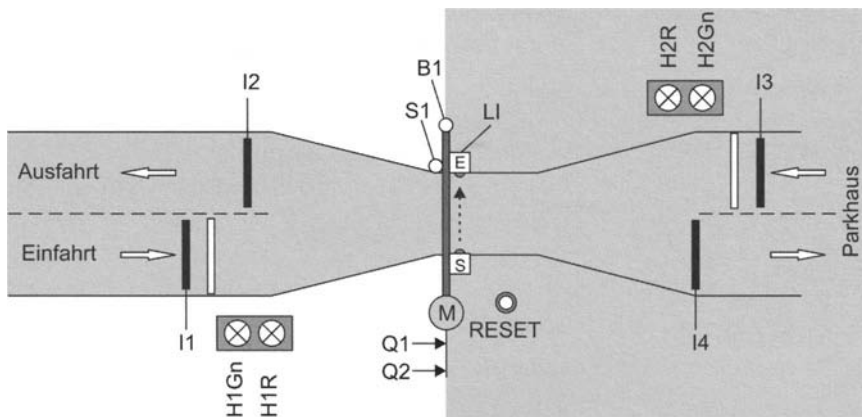
1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Geben Sie den Ablauf-Funktionsplan der Steuerungsaufgabe an.
3. Ermitteln Sie aus dem Ablauf-Funktionsplan alle Transitionsbedingungen T-1 bis T-n.
4. Zeichnen Sie den Funktionsplan für die Umsetzung der Ablaufkette in einem Baustein.
5. Geben Sie die für die Realisierung des Funktionsplans erforderlichen Deklarationen an.
6. Zeichnen Sie den Aufruf des Funktionsbausteins FB 10 im OB1 und versehen Sie die Eingänge und Ausgänge des Bausteins mit angenommenen SPS-Operanden.

Kontrollaufgabe 8.5

Die Ein- und Ausfahrt einer Tiefgarage ist nur einspurig zu befahren und mit einem Rollentor verschlossen. Auf beiden Seiten des Tores befinden sich jeweils zwei Induktionsschleifen. Die Unterkante des Tores ist mit einem luftgefüllten Schlauch abgeschlossen. Wenn die Tür geschlossen ist, erhöht sich der Druck im Schlauch und der Antrieb wird über einen Druckwächter B1 abgeschaltet.

Die Endstellung „offen“ wird durch den Endschalter S1 gemeldet. Die Lichtschranke LI soll ein Schließen des Tores verhindern, wenn sich ein Hindernis im Torbereich befindet. Vor und hinter dem Tor sind Ampeln angebracht, welche die Durchfahrt steuern sollen.

Technologieschema:



Funktionsablaufbeschreibung:

Im Normalfall ist das Tor geschlossen und beide Ampeln zeigen Rot. Fährt ein Fahrzeug auf die Induktionsschleife I1 bzw. I3 öffnet das Tor (Q1). Ist das Tor ganz geöffnet (S1), schaltet die zugehörige Ampel auf Grün. Verlässt das Fahrzeug dann die Induktionsschleife, schaltet die Ampel sofort wieder auf Rot. Erst wenn das ein- oder ausfahrende Fahrzeug die Induktionsschleife (I2 oder I4) auf der anderen Seite passiert hat, wird das Rollentor wieder geschlossen (Q2). Steht allerdings noch ein Fahrzeug auf der Einfahrt- oder Ausfahrtinduktionsschleife, bleibt das Tor oben und die zugehörige Ampel schaltet auf Grün.

Fahrzeuge, welche die Induktionsschleife I1 oder I3 betätigt haben, müssen zwangsweise über eine der Induktionsschleifen I2 oder I4 fahren. Damit in der Tiefgarage eine unnötige Autoabgasbelastung vermieden wird, soll die Ausfahrt Vorrang vor der Einfahrt haben.

Mit der RESET-Taste kann die Ablaufkette in die Grundstellung versetzt werden.

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Geben Sie den Ablauf-Funktionsplan der Steuerungsaufgabe an.
3. Ermitteln Sie aus dem Ablauf-Funktionsplan alle Transitionsbedingungen T-1 bis T-n.
4. Zeichnen Sie den Funktionsplan für die Umsetzung der Ablaufkette in einem Baustein.
5. Geben Sie die für die Realisierung des Funktionsplans erforderlichen Deklarationen an.
6. Zeichnen Sie den Aufruf des Funktionsbausteins FB 10 im OBI und versehen Sie die Eingänge und Ausgänge des Bausteins mit angenommenen SPS-Operanden.

SPS-Programme werden in Bausteine geschrieben. Ausgangspunkt der Auswahlentscheidung zwischen FC und FB sind die Anforderungen der im Baustein zu deklarierenden Variablen.

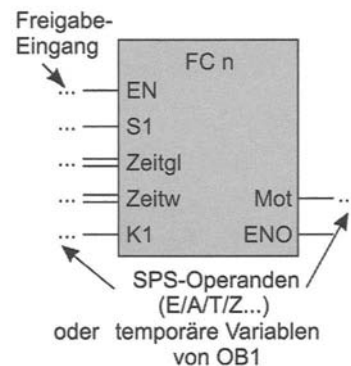
Einsatz und Deklaration einer Funktion FC

FCs sind parametrierbare Programmbausteine ohne eigenen Datenbereich. FCs genügen, wenn keine interne Speicherfunktion nötig ist oder die Speicherung eines Variablenwertes nach außen verlagert werden kann.

Deklarationstabelle: FC n

| Name | Datentyp |
|--------|----------|
| IN | |
| S1 | Bool |
| Zeitgl | Timer |
| Zeitw | S5Time |
| IN_OUT | |
| K1 | Bool |
| OUT | |
| Mot | Bool |
| TEMP | |
| HO | Word |

Aufruf des FC n im OB1:



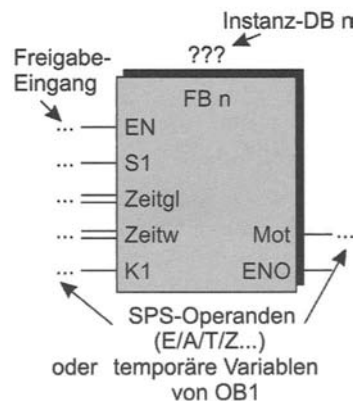
Einsatz und Deklaration eines Funktionsbausteins FB

FBs sind parametrierbare Programmbausteine, denen beim Aufruf ein eigener Speicherbereich (Instanz-DB) zugewiesen wird. FBs sind notwendig, wenn ein speicherndes Verhalten einer bausteininternen Variablen nötig ist.

Deklarationstabelle: FB n

| Name | Datentyp | Anfangswert |
|--------|----------|-------------|
| IN | | |
| S1 | Bool | FALSE |
| Zeitgl | Timer | |
| Zeitw | S5Time | S5T#0MS |
| IN_OUT | | |
| K1 | Bool | FALSE |
| OUT | | |
| Mot | Bool | FALSE |
| STAT | | |
| SP | Int | 0 |
| TEMP | | |
| HO | Word | |

Aufruf des FB n im OB1:



Organisationsbaustein OB1

Der OB 1 übernimmt die zyklische Programmbearbeitung der aufgerufenen FBs und FCs und die Versorgung ihrer Bausteinparameter mit E-/A-Adressen der SPS. Außerdem dient er der Parameterübergabe zwischen aufgerufenen Bausteinen mit Hilfe eigener temporärer Variablen (Datenaustausch zwischen zwei oder mehreren Bausteinen).

9.1 Beispiel

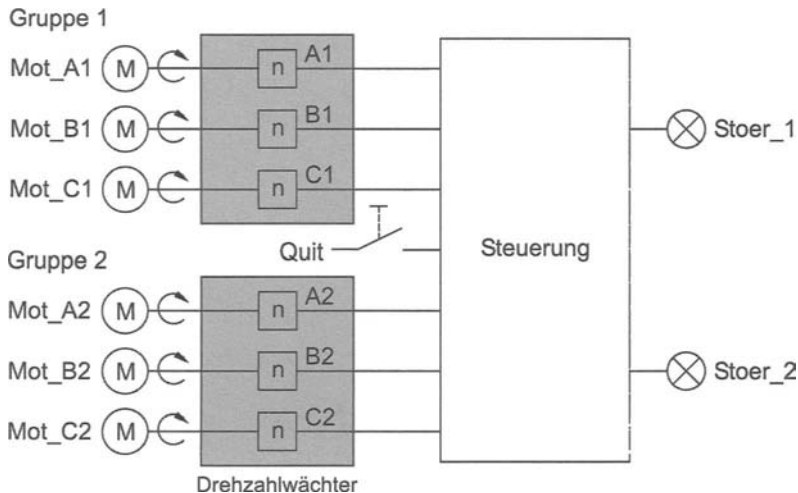
Motorgruppenüberwachung

Zwei Motorgruppen mit den Motoren A, B, C sollen überwacht werden. Jeder Motor ist mit einem Drehzahlwächter ausgerüstet, der mit 1- oder 0-Signal meldet, ob der Motor dreht oder nicht. Die Störungsanzeige jeder Motorengruppe soll in folgenden Fällen aufleuchten:

1. Fall: Wenn zwei der drei Motoren länger als 5 Sekunden ausgefallen sind.
2. Fall: Wenn alle drei Motoren ausgefallen sind (sofortige Meldung).

Die Störungsanzeige soll selbsttätig verlöschen, wenn die Störungsursachen behoben sind, also wenigstens zwei Motoren wieder laufen. Bei Ausfall aller drei Motoren, muss nach der Störungsbeseitigung noch zusätzlich die für beide Motorengruppe geltende Quittierungstaste betätigt werden, um die Störmeldung abzuschalten.

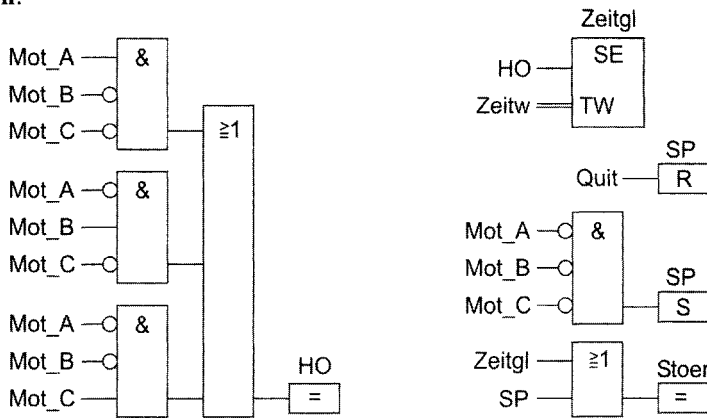
Technologieschema:



Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | | Adresse |
|------------------------|---------|----------|--------------------|-------------|---------|
| Drehzahlwächter Mot_A1 | A1 | BOOL | Motor A1 läuft | A1 = 1 | E 0.0 |
| Drehzahlwächter Mot_B1 | B1 | BOOL | Motor B1 läuft | B1 = 1 | E 0.1 |
| Drehzahlwächter Mot_C1 | C1 | BOOL | Motor C1 läuft | C1 = 1 | E 0.2 |
| Drehzahlwächter Mot_A2 | A2 | BOOL | Motor A2 läuft | A2 = 1 | E 1.0 |
| Drehzahlwächter Mot_B2 | B2 | BOOL | Motor B2 läuft | B2 = 1 | E 1.1 |
| Drehzahlwächter Mot_C2 | C2 | BOOL | Motor C2 läuft | C2 = 1 | E 1.2 |
| Quittierung | Quit | BOOL | Betätigt | Quit = 1 | E 1.7 |
| Ausgangsvariable | | | | | |
| Störungsanzeige 1 | Stoer_1 | BOOL | Leuchtet | Stoer_1 = 1 | A 4.0 |
| Störungsanzeige 2 | Stoer_2 | BOOL | Leuchtet | Stoer_2 = 1 | A 5.0 |

Funktionsplan:



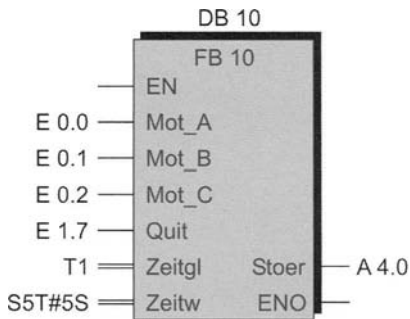
Die Analyse des Programms und seiner Variablen ergibt, dass SP eine interne Variable mit Speicherverhalten (Gedächtnis) sein muss, sodass ein Funktionsbaustein FB auszuwählen ist.

Deklarationstabelle: FB10

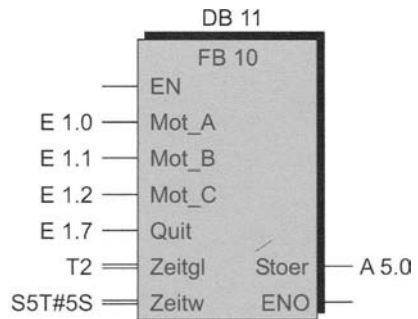
| Name | Datentyp | Anfangswert | Kommentar |
|-------------|----------|-------------|---|
| IN | | | |
| Mot_A | Bool | FALSE | |
| Mot_B | Bool | FALSE | |
| Mot_C | Bool | FALSE | |
| Quit | Bool | FALSE | |
| Zeitgl | Timer | | Zeitglied für Einschaltverzögerung |
| Zeitw | S5Time | S5T#0S | Zeitwert des Zeitgliedes |
| OUT | | | |
| Stoer | Bool | FALSE | |
| STAT | | | |
| SP | Bool | FALSE | Statische Variable erfordert Bausteintyp FB |
| TEMP | | | |
| HO | Bool | | Hilfsoperand (Vermeidung eines Merkers) |

OB 1: Zweimaliger Aufruf des FB 10 erfordert zwei Instanz-DBs

Störungsanzeige Motorgruppe 1:



Störungsanzeige Motorgruppe 2:

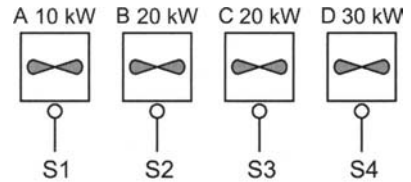


9.2 Lernaufgaben

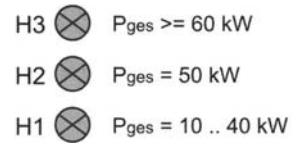
Lernaufgabe 9.1 Belüftungsanlage

Lös. S. 253

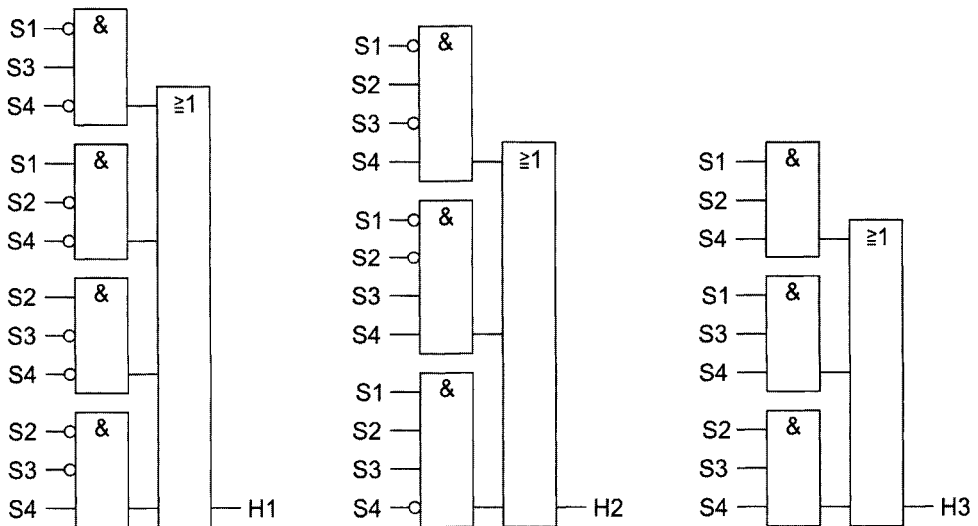
Eine größere Belüftungsanlage besteht aus vier Lüftergruppen. Die Leistungswerte der einzelnen Gruppen sind im nebenstehenden Bild angegeben. Ob eine Lüftergruppe an ist, wird mit den Signalgebern S1 bis S4 einer Steuerung gemeldet



Drei Meldeleuchten H1, H2 und H3 zeigen den geschalteten Leistungszustand optisch an. Die Signallampe H1 leuchtet, wenn die geschaltete Leistung 10 kW ... 40 kW beträgt, die Signallampe H2 leuchtet, wenn die geschaltete Leistung 50 kW beträgt und die Signallampe H3 leuchtet, wenn die geschaltete Leistung 60 kW oder mehr beträgt.



Der folgende Funktionsplan gibt die Ansteuerlogik für die Meldeleuchten wieder.

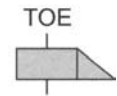
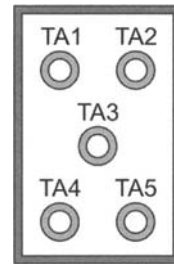


- Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge. Wählen Sie dabei die Zuordnung „1“-Signal für betätigte Signalgeber.
- Das gegebene Steuerungsprogramm soll in einem bibliotheksfähigen Baustein realisiert werden. Überlegen Sie, welche Art von Baustein Sie für das gegebene Steuerungsprogramm verwenden können.
- Geben Sie die Deklarationstabelle für den Baustein an.
- Programmieren Sie den Baustein mit einer Programmiersprache Ihrer Wahl (AWL, FUP oder KOP).
- Rufen Sie den Baustein vom OB1 aus auf und versehen Sie die Ein- und Ausgänge mit den SPS-Operanden aus Ihrer Zuordnungstabelle.

Lernaufgabe 9.2 Schlossschaltung

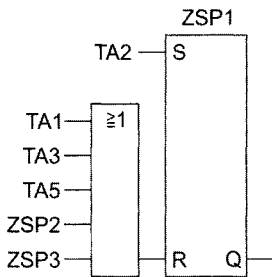
Lös. S. 253

Der Eingang eines Lagerraums ist mit einer Schlossschaltung gesichert. Hierzu sind fünf Taster TA1 bis TA5 angebracht. Nach Eingabe der Tastenfolge TA2-TA4-TA3-TA2-TA5 wird ein Türöffner TOE solange angesteuert, wie der Taster TA5 gedrückt wird. Wird in der Tastenfolge eine falsche Taste gedrückt, so muss die richtige Reihenfolge nochmals von Beginn an eingegeben werden.

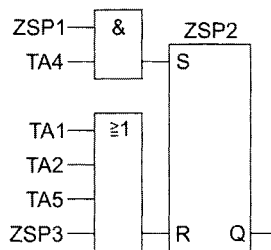


Der nachfolgende Funktionsplan gibt das Steuerungsprogramm für die Schlossschaltung wieder:

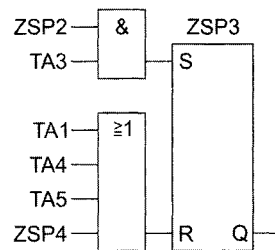
1. Zwischenspeicher ZSP1



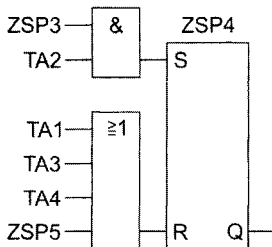
2. Zwischenspeicher ZSP2



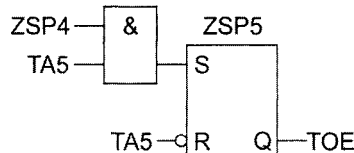
3. Zwischenspeicher ZSP3



4. Zwischenspeicher ZSP4



5. Zwischenspeicher ZSP5 und Ansteuerung des Türöffners

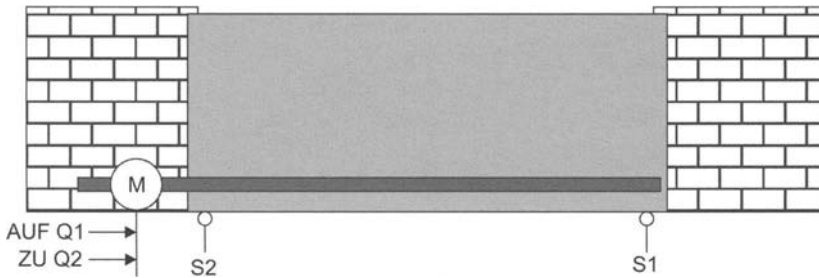


1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Das gegebene Steuerungsprogramm soll in einem bibliotheksfähigen Baustein realisiert werden. Welche Art von Codebaustein muss für das Steuerungsprogramm verwendet werden?
3. Geben Sie die vollständige Deklarationstabelle für den Baustein an.
4. Programmieren Sie den Baustein mit einer Programmiersprache Ihrer Wahl (AWL, FUP oder KOP).
5. Rufen Sie den Baustein vom OBI aus auf und versehen Sie die Ein- und Ausgänge mit den SPS-Operanden aus Ihrer Zuordnungstabelle.

Lernaufgabe 9.3 Torsteuerung

Lös. S. 254

Die Ein- und Ausfahrt in ein Fabrikgelände kann über drei Werktoere erfolgen. Jedes der Werktoere wird mit einem Elektromotor über zwei Leistungsschütze angesteuert. Die Endlagen der Schiebetore werden mit entsprechenden Endschaltern überwacht.

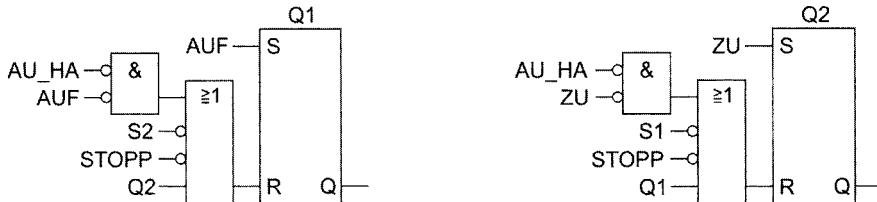
Technologieschema:

Zur Bedienung der drei Tore ist im Pförtnerhaus ein zentrales Bedienpult angebracht. Von diesem kann jedes Tor einzeln gesteuert werden. Mit einem Wahlschalter kann Automatik- oder Handbetrieb gewählt werden.

Steht der Wahlschalter auf AUTOMATIK, wird durch einen einmaligen Tastendruck das jeweilige Tor bis zur Abschaltung durch einen Endschalter auf- oder zugefahren. In der Stellung HAND, muss der jeweilige Taster ständig gedrückt werden.



Ein Betätigung des STOPP-Tasters unterbricht sofort das Auf- oder Zufahren des Tores. Das Steuerungsprogramm für ein Tor ist durch den nachfolgenden Funktionsplan gegeben.



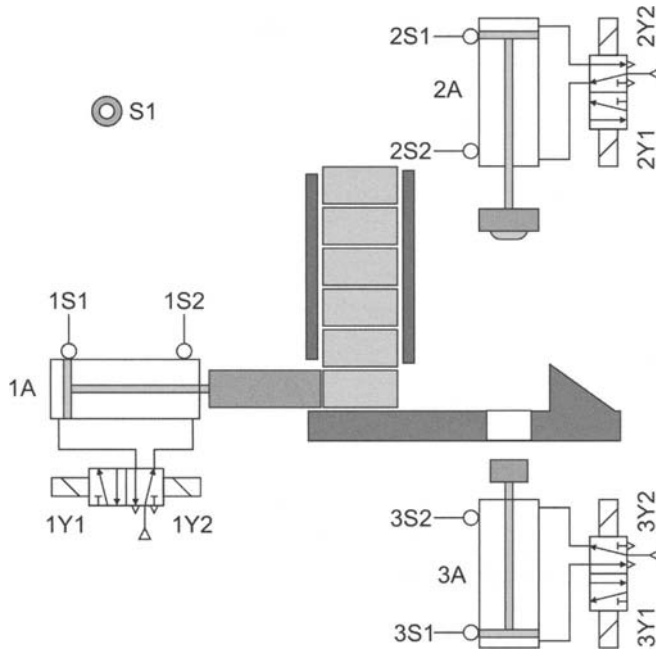
1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge. Achten Sie dabei auf drahtbruchsichere Zuordnung („0“ oder „1“-Signal) für betätigte Endschalter bzw. Taster.
2. Das gegebene Steuerungsprogramm soll mit einem Baustein realisiert werden, der für jedes Tor aufgerufen und mit den entsprechenden SPS-Operanden versehen wird. Überlegen Sie, welcher Bausteintyp verwendet werden kann.
3. Geben Sie die Deklarationstabelle für den Baustein an.
4. Programmieren Sie den Baustein, rufen Sie diesen von OB1 aus dreimal auf und versehen Sie die Eingänge und Ausgänge mit den SPS-Operanden aus Ihrer Zuordnungstabelle.

Lernaufgabe 9.4: Prägevorrichtung

Lös. S. 255

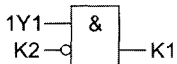
Mit einer Prägevorrichtung wird eine Kennnummer auf Werkstücken angebracht. Der Prägevorgang läuft wie folgt ab: Nach Betätigung des Tasters S1 fährt Zylinder 1A1 aus und spannt das Werkstück. Erreicht Zylinder 1A1 die vordere Endlage, fährt Zylinder 2A1 nach unten zum Anbringen der Kennung. Nachdem Zylinder 2A1 wieder in der oberen Endlage angekommen ist, fährt auch Zylinder 1A1 ein. Zylinder 3A1 fährt dann aus und schiebt das Werkstück zur nächsten Bearbeitungsstation. Das folgende Technologieschema zeigt die Anordnung der Zylinder.

Technologieschema:

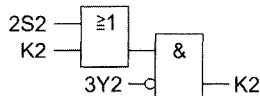


Die Ansteuerung der elektrischen Magnetventile erfolgte bisher durch eine Schützsteuerung. Bei einer Revision der Anlage soll die Schützsteuerung durch eine SPS ersetzt werden. Das Programm wurde aus der Logik der Schützsteuerung entwickelt. Dabei ergab sich folgender Funktionsplan:

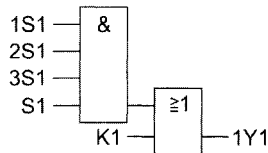
Hilfsvariable K1



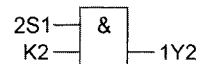
Hilfsvariable K2



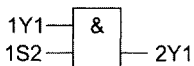
Magnetventil 1Y1



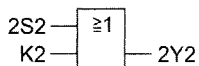
Magnetventil 1Y2



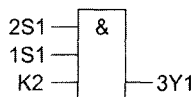
Magnetventil 2Y1



Magnetventil 2Y2



Magnetventil 3Y1



Magnetventil 3Y2



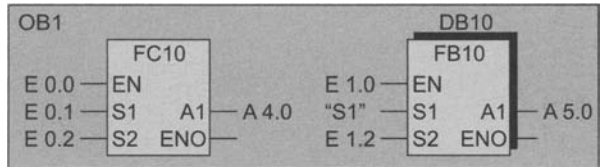
Hinweis. Die Hilfsvariablen K1 und K2 ersetzen die Hilfsschütze der Schützsteuerung.

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Der gegebene Funktionsplan soll mit einem bibliotheksfähigen Baustein realisiert werden. Überlegen Sie, welche Art von Baustein Sie für den gegebenen Funktionsplan verwenden.
3. Geben Sie die Deklarationstabelle für den Baustein an.
4. Schreiben Sie den Baustein, rufen Sie diesen von OB1 aus auf und versehen Sie die Ein- und Ausgänge mit den SPS-Operanden aus Ihrer Zuordnungstabelle.

9.3 Kontrollaufgaben

Kontrollaufgabe 9.1

Das nebenstehende Bild zeigt den Aufruf der Codebausteine FB10 und FC10 im Organisationsbaustein OB1.



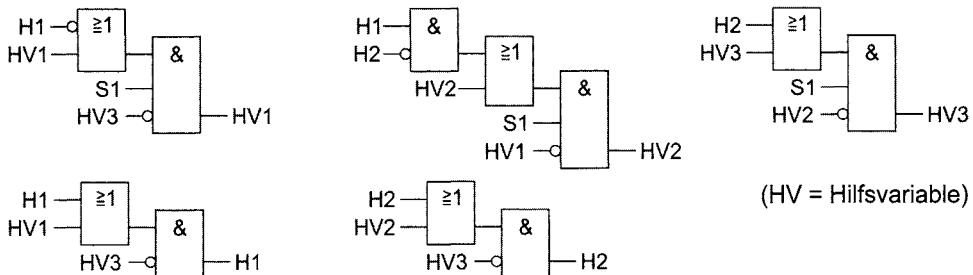
1. Erläutern Sie den Unterschied zwischen dem Baustein FC10 und dem Baustein FB10.
2. Welchen besonderen Namen hat der Datenbaustein DB10 und welche Aufgabe hat dieser?
3. Welche Deklarationen sind im Baustein FC10 und welche im Baustein FB10 möglich?
4. Welchen Deklarationen kann ein Anfangswert zugewiesen werden?
5. Erklären Sie die Besonderheit einer IN_OUT-Variablen.
6. Erläutern Sie den Unterschied zwischen einer globalen und einer lokalen Variablen und geben Sie Beispiele von globalen Variablen an.
7. Welcher Unterschied besteht zwischen der symbolischen Variablen "S1" und der Eingangsvariablen S1 des Codebausteins FB10?
8. Wozu kann der Eingang EN der beiden Codebausteine verwendet werden?

Kontrollaufgabe 9.2

Der gegebene Funktionsplan beschreibt die Ansteuerung der Meldeleuchten H1 und H2 durch den Taster S1.



Funktionsplan:



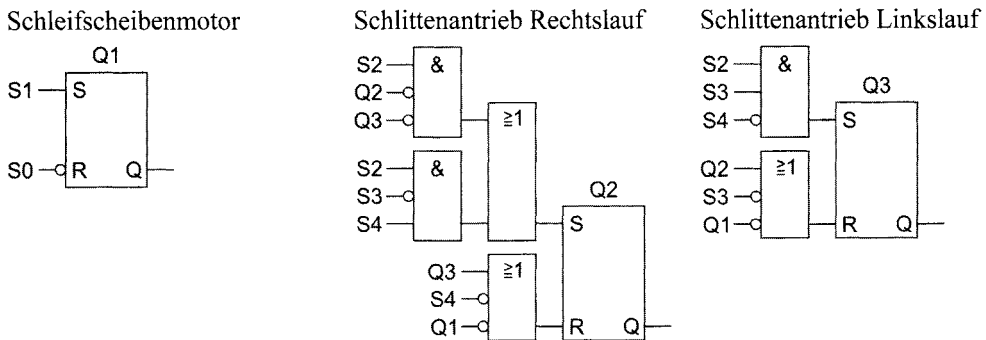
Nach der ersten Betätigung von S1 leuchtet die Meldeleuchte H1. Die zweite Betätigung von S1 schaltet die zweite Meldeleuchte dazu. Bei der dritten Betätigung von S1 erlöschen beide Meldeleuchten wieder.

1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Das gegebene Steuerungsprogramm soll in einem bibliotheksfähigen Baustein realisiert werden. Überlegen Sie, welchen Bausteintyp FC oder FB Sie für das gegebene Steuerungsprogramm verwenden können.
3. Geben Sie die vollständige Deklarationstabelle für den Baustein an.
4. Zeichnen Sie den Aufruf des Bausteins im OB1 und versehen Sie die Ein- und Ausgänge mit den SPS-Operanden aus Ihrer Zuordnungstabelle.

Kontrollaufgabe 9.3

Eine Schleifmaschine wird durch Betätigung des Tasters S1 in Betrieb gesetzt. Der Schleifscheibenmotor M1 wird über Motorschütz Q1 eingeschaltet. Hat der Motor seine Nenndrehzahl erreicht (Meldung mit S2 = 1), startet der Schlittenantrieb M2 durch das Motorschütz Q2 im Rechtslauf. Beim Erreichen der Endposition „RECHTS“ (S3 = 0) wird der Schlittenantrieb M2 durch das Schütz Q3 (Linkslauf) bis zum Erreichen der Endposition „LINKS“ (S4 = 0) umgeschaltet. Dieser Vorgang wiederholt sich, bis mit S0 die Schleifmaschine abgeschaltet wird.

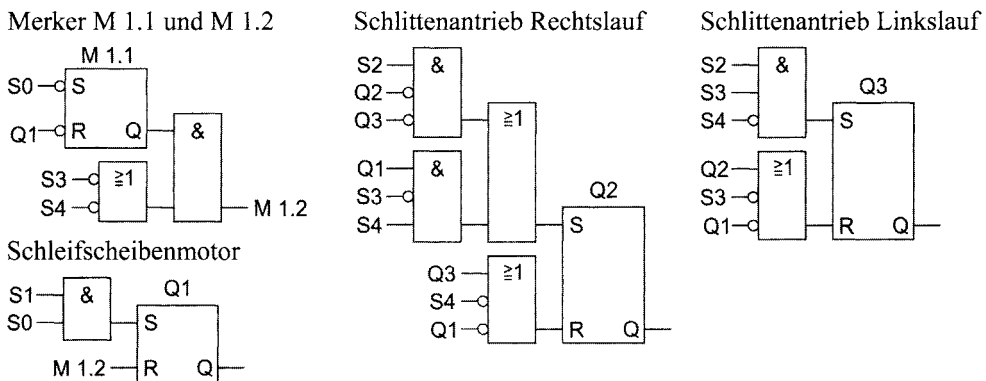
Das nachfolgende Funktionsplan zeigt das Steuerungsprogramm der Schleifmaschine.



1. Bestimmen Sie die Zuordnungstabelle der Eingänge und Ausgänge.
2. Das gegebene Steuerungsprogramm soll in einem bibliotheksfähig Baustein realisiert werden. Welche Art von Codebaustein kann für das Steuerungsprogramm verwendet werden?
3. Geben Sie die vollständige Deklarationstabelle für den Baustein an.

Das Steuerungsprogramm wird durch Hinzufügen zweier Merker so verändert, dass nach Betätigung der Taste S0 der Schlittenantrieb und der Schleifscheibenmotor bis zum Erreichen einer Endposition weiterläuft.

Veränderter Funktionsplan:



4. Welche Art von Codebaustein muss für das Steuerungsprogramm verwendet werden, wenn anstelle der beiden Merker M 1.1 und M 1.2 die beiden lokalen Variablen HV1 und HV2 treten sollen, um die Bibliotheksfähigkeit des Bausteins zu erhalten?
5. Geben Sie die vollständige Deklarationstabelle für den Baustein an.

Lösungen der Lernaufgaben

1. Nachfolgend finden Sie die Zusammenstellung der Lösungen aller Lernaufgaben unter deren fortlaufender Kapitelnummer.
2. Die Lösungen umfassen je nach Aufgabenstellung eine allgemeingültige Funktionsplandarstellung, eine Deklarationstabelle mit den Variablen, eine Zuordnungstabelle für die verwendeten SPS-Eingänge/-Ausgänge, einen Ablauf-Funktionsplan und die Transitionstabelle bei Ablaufsteuerungen, eine Begründung für die Auswahl des verwendeten Bausteintyps (FC oder FB/DB) und den Bausteinaufruf im OB1 sowie ggf. Funktionsdiagramme (Weg-Schritt-Diagramme) für Magnetventile und pneumatische Zylinder.
3. Auf die explizite Darstellung der STEP 7-Programme in ihren Sprachen AWL, FUP und KOP wurde aus Gründen der Umfangsbeschränkung der Buches verzichtet, zumal sich diese nur geringfügig von den allgemeinen Darstellungen unterscheiden und vom Programmiersystem automatisch die genaue STEP 7-Notation erhalten. Die Programme befinden sich jedoch auf der beiliegenden CD oder können aktuell via Internet kostenfrei heruntergeladen werden.
Die Adresse lautet: <http://www.automatisieren-mit-sps.de>
4. Alle Lernaufgaben der Kapitel 2 bis 9 können auch mit dem Programmiersystem **CoDeSys 2.3** nach der in den folgenden Lösungen dargestellten Programmstruktur programmiert werden. Lösungsvorschlägen finden Sie unter der genannten Internetadresse zum kostenfreien download.

Lösungen der Kontrollaufgaben

Die Lösungen der Kontrollaufgaben sind im vorliegenden Buch nicht abgedruckt, stehen jedoch als PDF-Datei auf der beiliegenden CD oder unter obiger Internetadresse kostenfrei zur Verfügung. In der Zusammenstellung der Lösungen aller Kontrollaufgaben finden Sie die Lösungsvorschläge unter den entsprechenden Kapitelnummer. (Die Lösungsvorschläge zu den Kontrollaufgaben von Kapitel n sind somit in Abschnitt 11. n zu finden).

10.1 Lösungen der Lernaufgaben Kapitel 1

Keine Lernaufgaben im Kapitel 1 „Einführung“.

Lernaufgaben beginnen erst im Kapitel 2, deren Lösungen auf der folgenden Seite anfangen.

10.2 Lösungen der Lernaufgaben Kapitel 2

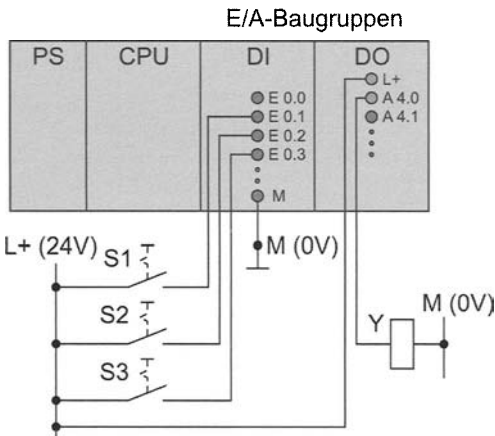
Lösung Lernaufgabe 2.1

Aufg. S. 12

1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|---------------------|--------|----------|-----------------------|---------|
| Schalter 1 | S1 | BOOL | Betätigt S1 = 1 | E 0.1 |
| Schalter 2 | S2 | BOOL | Betätigt S2 = 1 | E 0.2 |
| Schalter 3 | S3 | BOOL | Betätigt S3 = 1 | E 0.3 |
| Ausgangsvariable | | | | |
| Elektromagnetventil | Y | BOOL | Ventil geöffnet Y = 1 | A 4.0 |

2. Anschlussplan des SPS:

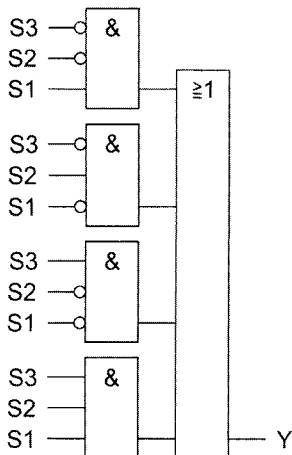


3. Funktionstabelle:

| S3 | S2 | S1 | Y |
|----|----|----|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

4. DNF – Schaltfunktion und Funktionsplan:

$$Y = \bar{S3} \bar{S2} S1 \vee S3 \bar{S2} \bar{S1} \vee S3 S2 \bar{S1} \vee S3 S2 S1$$

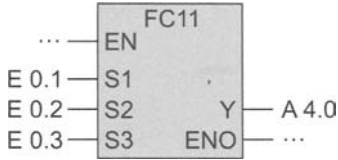


5. Deklarationstabelle FC11:

| Name | Datentyp |
|------------|----------|
| IN | |
| S1 | Bool |
| S2 | Bool |
| S3 | Bool |
| OUT | |
| Y | Bool |

6. Realisierung:

Aufruf im OBI:



Programmierung und Aufruf des Bausteins siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de>

STEP7-Lösungen → 2_2_01

Lösung Lernaufgabe 2.2 Aufg. S. 12

1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | | Adresse |
|------------------|--------|----------|--------------------|--------|---------|
| Sensor unten | S1 | BOOL | Betätigt | S1 = 1 | E 0.1 |
| Sensor halbvoll | S2 | BOOL | Betätigt | S2 = 1 | E 0.2 |
| Sensor oben | S3 | BOOL | Betätigt | S3 = 1 | E 0.3 |
| Schalter | S4 | BOOL | Betätigt | S4 = 1 | E 0.4 |
| Ausgangsvariable | | | | | |
| Pumpe 1 | P1 | BOOL | Pumpe an | P1 = 1 | A 4.1 |
| Pumpe 2 | P2 | BOOL | Pumpe an | P2 = 1 | A 4.2 |
| Störungsanzeige | H1 | BOOL | Störung liegt vor | H1 = 1 | A 4.3 |

2. Funktionstabelle:

| S3 | S2 | S1 | P1 | P2 | H1 | Bemerkungen |
|----|----|----|----|----|----|--------------|
| 0 | 0 | 0 | 1 | 0 | 0 | |
| 0 | 0 | 1 | 1 | 0 | 0 | |
| 0 | 1 | 0 | 1 | 1 | 1 | Sensorfehler |
| 0 | 1 | 1 | 0 | 1 | 0 | |
| 1 | 0 | 0 | 1 | 1 | 1 | Sensorfehler |
| 1 | 0 | 1 | 1 | 1 | 1 | Sensorfehler |
| 1 | 1 | 0 | 1 | 1 | 1 | Sensorfehler |
| 1 | 1 | 1 | 1 | 1 | 0 | |

3. DNF – Schaltfunktion und Funktionsplan:

$$\overline{P1} = \overline{S3} \overline{S2} \overline{S1}$$

$$\Rightarrow P1 = S3 \vee \overline{S2} \vee \overline{S1}$$

$$\overline{P2} = \overline{S3} \overline{S2} \overline{S1} \vee \overline{S3} \overline{S2} S1 = S3 S2 (\overline{S1} \vee S1) = \overline{S3} \overline{S2}$$

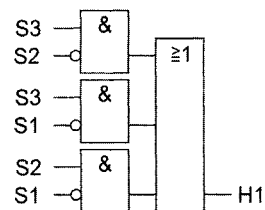
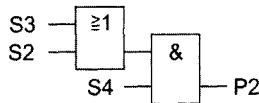
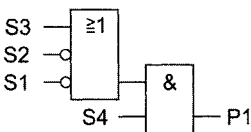
$$\Rightarrow P2 = S3 \vee S2$$

$$H1 = \overline{S3} S2 \overline{S1} \vee S3 \overline{S2} \overline{S1} \vee S3 \overline{S2} S1 \vee S3 S2 \overline{S1} \quad *)$$

$$\Rightarrow H1 = S3 \overline{S2} \vee S3 \overline{S1} \vee S2 \overline{S1}$$

*) Nach Minimierung mit einem KVS-Diagramm

Unter Berücksichtigung von Schalter S4:



4. Deklarationstabelle FC12:

| Name | Datentyp |
|------|----------|
| IN | |
| S1 | Bool |
| S2 | Bool |

| Name | Datentyp |
|------|----------|
| IN | |
| S3 | Bool |
| S4 | Bool |

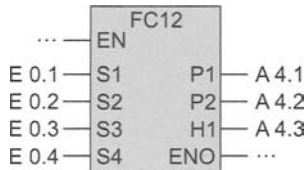
| Name | Datentyp |
|------|----------|
| OUT | |
| P1 | Bool |
| P2 | Bool |
| H1 | Bool |

5. Realisierung:

Aufruf im OB1:

Programmierung und Aufruf des Bausteins siehe beiliegende CD oder <http://automatisieren-mit-sps.de>

STEP7-Lösungen → 2_2_02



| | |
|-------------------------------|-------------|
| Lösung Lernaufgabe 2.3 | Aufg. S. 13 |
|-------------------------------|-------------|

1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | | Adresse |
|-------------------------|--------|----------|--------------------|--------|---------|
| Luftströmungswächter 1 | S1 | BOOL | Lüfter 1 an | S1 = 1 | E 0.1 |
| Luftströmungswächter 2 | S2 | BOOL | Lüfter 2 an | S2 = 1 | E 0.2 |
| Luftströmungswächter 3 | S3 | BOOL | Lüfter 3 an | S3 = 1 | E 0.3 |
| Luftströmungswächter 4 | S4 | BOOL | Lüfter 4 an | S4 = 1 | E 0.4 |
| Ausgangsvariable | | | | | |
| Lampe Rot | rt | BOOL | Leuchtet | rt = 1 | A 4.1 |
| Lampe Gelb | ge | BOOL | Leuchtet | ge = 1 | A 4.2 |
| Lampe Grün | gn | BOOL | Leuchtet | gn = 1 | A 4.3 |

2. Funktionstabelle:

| S4 | S3 | S2 | S1 | gn | ge | rt |
|----|----|----|----|----|----|----|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 |

3. DNF-Schaltfunktion:

Zur Bildung der DNF werden zwei der drei Ausgänge mit einer geringeren Anzahl von „1“-Signalwerten (also gn und rt) gewählt.

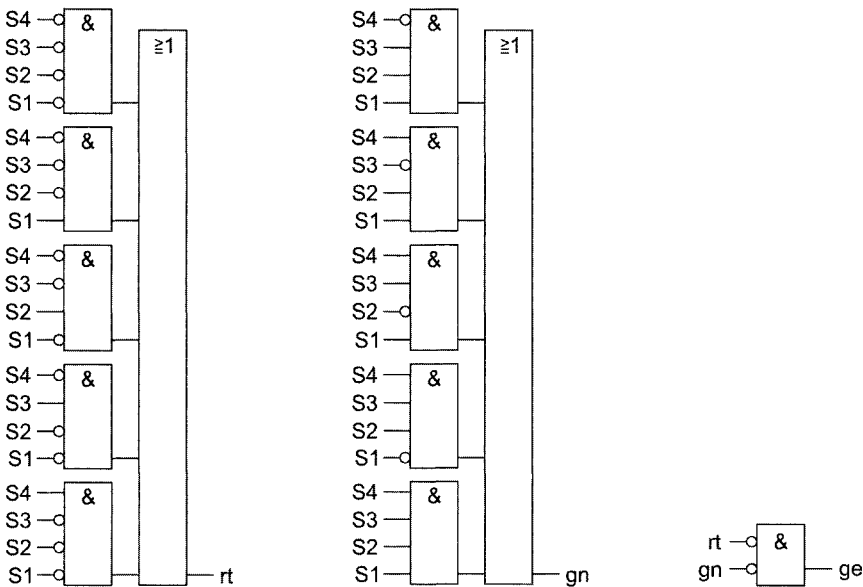
$$rt = \overline{S4} \overline{S3} \overline{S2} \overline{S1} \vee \overline{S4} \overline{S3} \overline{S2} S1 \vee \overline{S4} \overline{S3} S2 \overline{S1} \vee \overline{S4} S3 \overline{S2} \overline{S1} \vee S4 \overline{S3} \overline{S2} \overline{S1}$$

$$gn = \overline{S4} S3 S2 S1 \vee S4 \overline{S3} S2 S1 \vee S4 S3 \overline{S2} S1 \vee S4 S3 S2 \overline{S1} \vee S4 S3 S2 S1$$

Die dritte Ausgangsvariable (ge) wird aus den bereits ermittelten Variablen (rt und gn) wie folgt gebildet:

$$ge = \overline{gn} \& \overline{rt} = \overline{gn \vee rt}, \text{ wobei } rt \text{ und } gn \text{ als IN_OUT-Variablen zu deklarieren sind}$$

4. Funktionsplan:



5. Deklarationstabelle FC13:

| Name | Datentyp |
|-----------|----------|
| IN | |
| S1 | Bool |
| S2 | Bool |

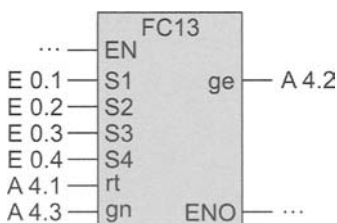
| Name | Datentyp |
|-----------|----------|
| IN | |
| S3 | Bool |
| S4 | Bool |

| Name | Datentyp |
|------------|----------|
| OUT | |
| ge | Bool |

| Name | Datent. |
|---------------|---------|
| IN_OUT | |
| rt | Bool |
| gn | Bool |

6. Realisierung:

Aufruf im OB1:



Programmierung und Aufruf des Bausteins siehe beiliegende CD oder <http://automatisieren-mit-sps.de>
STEP7-Lösungen → 2_2_03

Lösung Lernaufgabe 2.4

Aufg. S. 13

1. Zuordnungstabelle der Eingänge und Ausgänge:

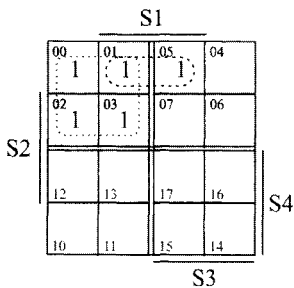
| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|---------------------|--------|----------|---------------------------|---------|
| Geber unten, Beh. 1 | S1 | BOOL | Füllstand erreicht S1 = 1 | E 0.1 |
| Geber unten, Beh. 2 | S2 | BOOL | Füllstand erreicht S2 = 1 | E 0.2 |
| Geber oben, Beh. 1 | S3 | BOOL | Füllstand erreicht S3 = 1 | E 0.3 |
| Geber oben, Beh. 2 | S4 | BOOL | Füllstand erreicht S4 = 1 | E 0.4 |
| Ausgangsvariable | | | | |
| Pumpe 1 | P1 | BOOL | Pumpe ein P1 = 1 | A 4.1 |
| Pumpe 2 | P2 | BOOL | Pumpe ein P2 = 1 | A 4.2 |
| Pumpe 3 | P3 | BOOL | Pumpe ein P3 = 1 | A 4.3 |
| Meldeleuchte | H | BOOL | Leuchtet H = 1 | A 4.4 |

2. Funktionstabelle:

| | S4 | S3 | S2 | S1 | P1 | P2 | P3 | H |
|----|----|----|----|----|----|----|----|---|
| 00 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 01 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| 02 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 03 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 04 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 05 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 06 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 07 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 10 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 11 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 12 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 13 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 14 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 15 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 16 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 17 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

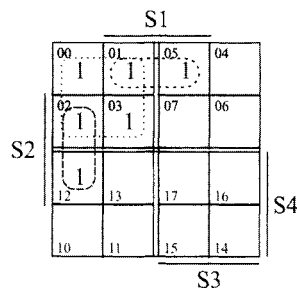
3. KVS-Diagramme:

P1:



$$P1 = \overline{S4}S3 \vee \overline{S4}S2S1$$

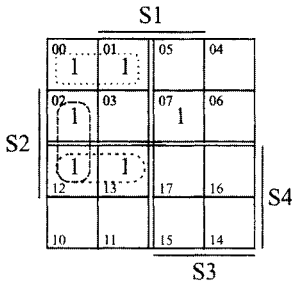
P2:



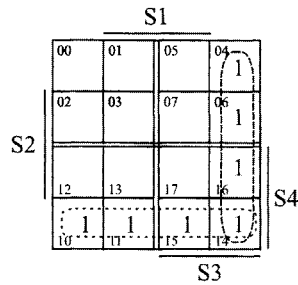
$$P2 = \overline{S4}S3 \vee \overline{S4}S2S1 \vee \overline{S3}S2S1$$

$$P2 = P1 \vee \overline{S3}S2S1$$

P3:



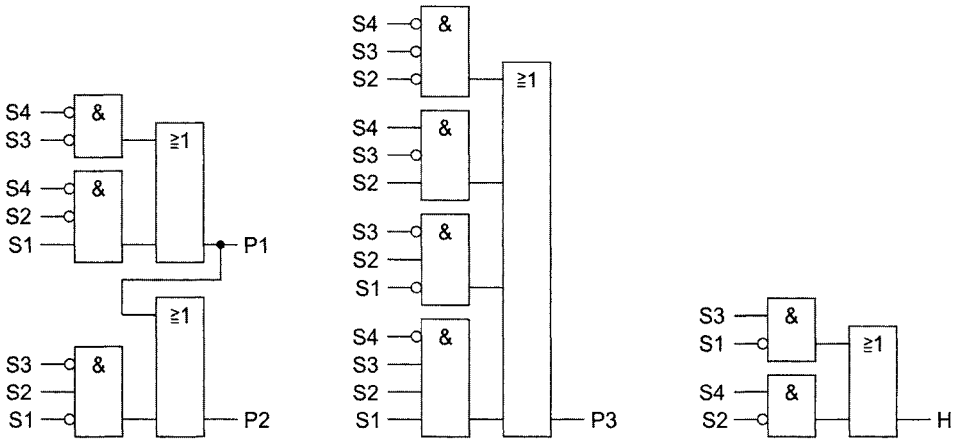
H:



$$P3 = \overline{S4} \overline{S3} \overline{S2} \vee \overline{S4} S3 \overline{S2} \vee \overline{S3} S2 \overline{S1} \vee \overline{S4} S3 S2 S1$$

$$H = S3 \overline{S1} \vee S4 \overline{S2}$$

4. Funktionsplan



5. Deklarationstabelle FC14:

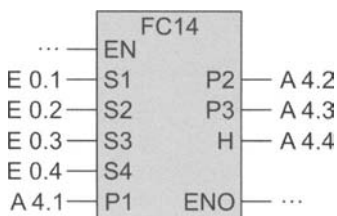
| Name | Datentyp |
|------|----------|
| IN | |
| S1 | Bool |
| S2 | Bool |
| S3 | Bool |
| S4 | Bool |

| Name | Datentyp |
|------|----------|
| OUT | |
| P2 | Bool |
| P2 | Bool |
| H | Bool |

| Name | Datentyp |
|--------|----------|
| IN_OUT | |
| P1 | Bool |

6. Realisierung:

Aufruf im OB1:



Programmierung und Aufruf des Bausteins siehe beiliegende CD oder <http://automatisieren-mit-sps.de>
STEP7-Lösungen → 2_2_04

Lösung Lernaufgabe 2.5

Aufg. S. 14

1. Zuordnungstabelle der Eingänge und Ausgänge:

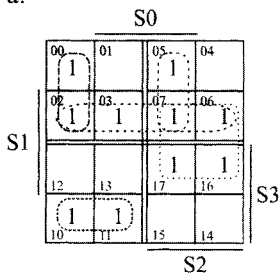
| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|------------------|--------|----------|--------------------------|---------|
| Schalter Wert 1 | S0 | BOOL | Schalter betätigt S0 = 1 | E 0.0 |
| Schalter Wert 2 | S1 | BOOL | Schalter betätigt S1 = 1 | E 0.1 |
| Schalter Wert 4 | S2 | BOOL | Schalter betätigt S2 = 1 | E 0.2 |
| Schalter Wert 8 | S3 | BOOL | Schalter betätigt S3 = 1 | E 0.3 |
| Ausgangsvariable | | | | |
| Segment a | a | BOOL | Segment leuchtet a = 1 | A 4.0 |
| Segment b | b | BOOL | Segment leuchtet b = 1 | A 4.1 |
| Segment c | c | BOOL | Segment leuchtet c = 1 | A 4.2 |
| Segment d | d | BOOL | Segment leuchtet d = 1 | A 4.3 |
| Segment e | e | BOOL | Segment leuchtet e = 1 | A 4.4 |
| Segment f | f | BOOL | Segment leuchtet f = 1 | A 4.5 |
| Segment g | g | BOOL | Segment leuchtet g = 1 | A 4.6 |

2. Funktionstabelle:

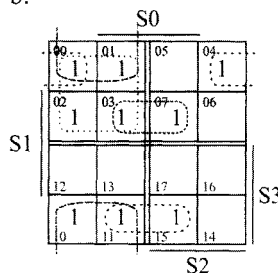
| Ziffer | Okt. | S3 | S2 | S1 | S0 | a | b | c | d | e | f | g |
|--------|------|----|----|----|----|---|---|---|---|---|---|---|
| 0 | 00 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 01 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 2 | 02 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 3 | 03 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 4 | 04 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 5 | 05 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 6 | 06 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 7 | 07 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 8 | 10 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 11 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| a | 12 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| b | 13 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| c | 14 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| d | 15 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| E | 16 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| F | 17 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |

3. KVS-Diagramme

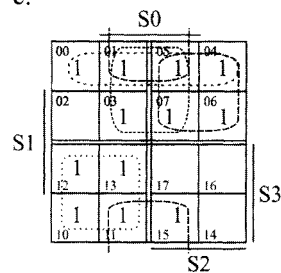
a:



b:



c:

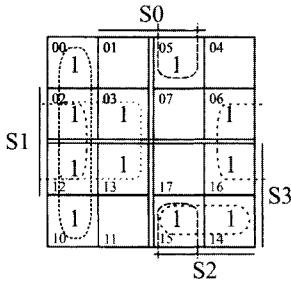


$$a = \overline{S3}S1 \vee S2S1 \vee S3\overline{S2}S1 \vee S3\overline{S2}S0 \vee \overline{S3}S2S0$$

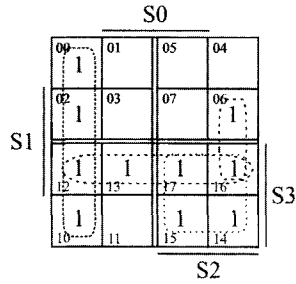
$$b = \overline{S_3}S_2 \vee \overline{S_2}S_1 \vee \overline{S_3}S_1S_0 \vee S_3\overline{S_1}S_0 \vee \overline{S_3}\overline{S_1}S_0$$

$$c = \overline{S_3}S_1 \vee \overline{S_3}S_0 \vee S_3S_2 \vee \overline{S_3}S_2 \vee \overline{S_1}S_0$$

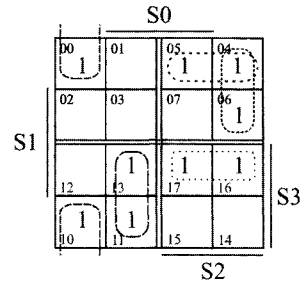
d:



e:



f:

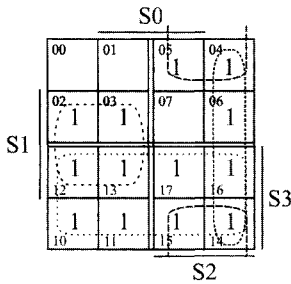


$$d = \overline{S_2}S_1 \vee \overline{S_2}S_0 \vee S_1S_0 \vee S_3S_2\overline{S_1} \vee S_2\overline{S_1}S_0$$

$$e = \overline{S_2}S_0 \vee S_3S_1 \vee S_3S_2 \vee S_2S_1S_0$$

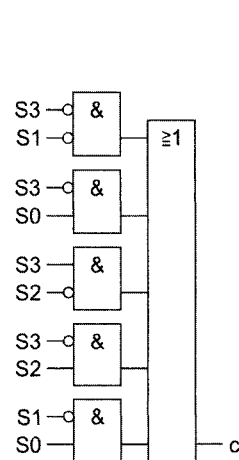
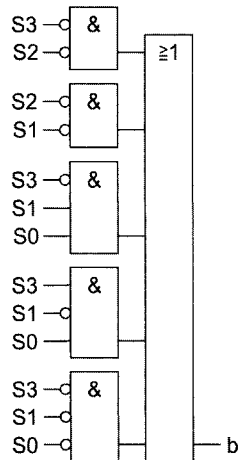
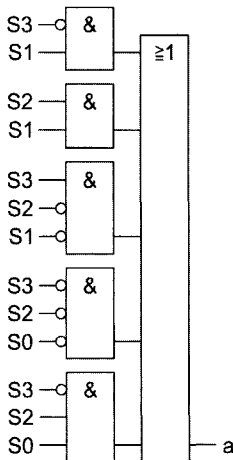
$$f = S_3S_2S_1 \vee \overline{S_3}S_2\overline{S_1} \vee \overline{S_3}S_2S_0 \vee S_3\overline{S_2}S_0 \vee \overline{S_2}\overline{S_1}S_0$$

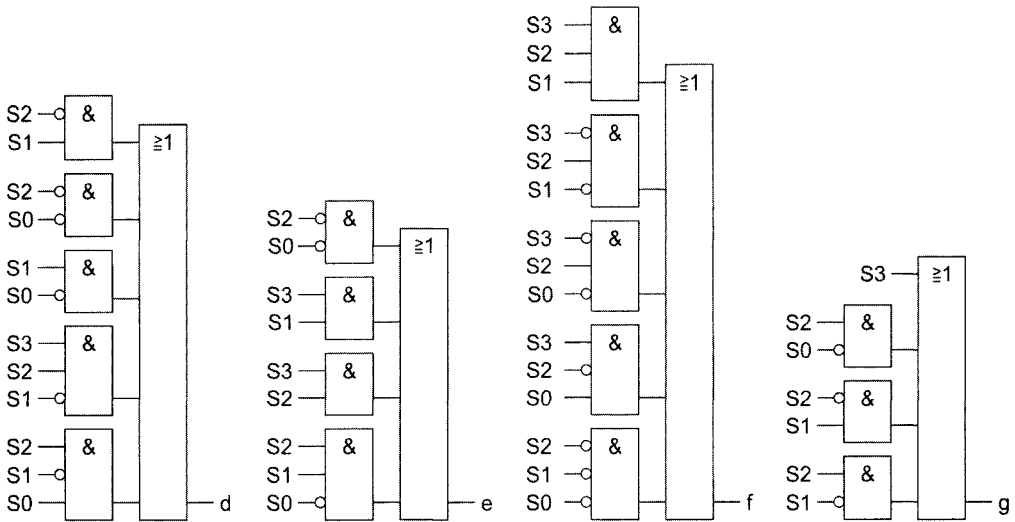
g:



$$g = S_3 \vee S_2\overline{S_0} \vee \overline{S_2}S_1 \vee S_2\overline{S_1}$$

4. Funktionsplan





5. Deklarationstabelle FC15:

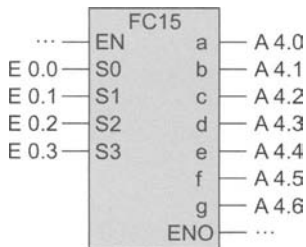
| Name | Datentyp |
|-----------|----------|
| IN | |
| S0 | Bool |
| S1 | Bool |
| S2 | Bool |
| S3 | Bool |

| Name | Datentyp |
|------------|----------|
| OUT | |
| a | Bool |
| b | Bool |
| c | Bool |
| d | Bool |

| Name | Datentyp |
|------------|----------|
| OUT | |
| e | Bool |
| f | Bool |
| g | Bool |

6. Realisierung:

Aufruf im OB1:



Programmierung und Aufruf des Bausteins siehe beliebige CD oder <http://automatisieren-mit-sps.de>
STEP7-Lösungen → 2_2_05

Lösung Lernaufgabe 2.6 Aufg. S. 14

1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|--------------------|--------|----------|-----------------------|---------|
| Lastabwurfrelais 1 | S1 | BOOL | Wasserentnahme S1 = 0 | E 0.1 |
| Lastabwurfrelais 2 | S2 | BOOL | Wasserentnahme S2 = 0 | E 0.2 |
| Lastabwurfrelais 3 | S3 | BOOL | Wasserentnahme S3 = 0 | E 0.3 |
| Lastabwurfrelais 4 | S4 | BOOL | Wasserentnahme S4 = 0 | E 0.4 |
| Lastabwurfrelais 5 | S5 | BOOL | Wasserentnahme S5 = 0 | E 0.5 |

| Ausgangsvariable | | | | | |
|-------------------|----|------|-----------|--------|-------|
| Freigabe-Schütz 1 | Q1 | BOOL | Angezogen | Q1 = 1 | A 4.1 |
| Freigabe-Schütz 2 | Q2 | BOOL | Angezogen | Q2 = 1 | A 4.2 |
| Freigabe-Schütz 3 | Q3 | BOOL | Angezogen | Q3 = 1 | A 4.3 |
| Freigabe-Schütz 4 | Q4 | BOOL | Angezogen | Q4 = 1 | A 4.4 |
| Freigabe-Schütz 5 | Q5 | BOOL | Angezogen | Q5 = 1 | A 4.5 |

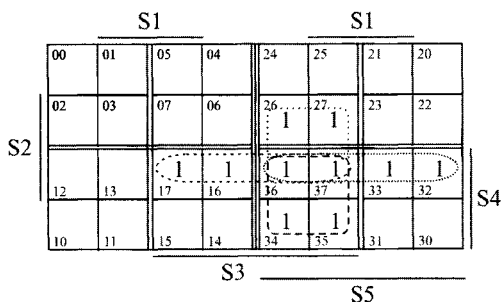
2. Funktionstabelle:

Die Darstellung der vollständigen Funktionstabelle für 5 Variablen, d.h. $2^5 = 32$ Kombinationen ist nicht erforderlich. Auf alle Kombinationen mit drei oder mehr angezogenen Lastabwurfrelais kann verzichtet werden, da diese Kombinationen nicht auftreten können. Ausgewertet werden müssen nur die in der reduzierten Funktionstabelle eingetragenen erlaubten Eingangskombinationen mit maximal zwei Nullen.

| Oktal-Nr | S5 | S4 | S3 | S2 | S1 | Q5 | Q4 | Q3 | Q2 | Q1 |
|----------|----|----|----|----|----|----|----|----|----|----|
| 07 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 13 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 15 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 16 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 17 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 23 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 25 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 26 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 27 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 31 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 32 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 33 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 34 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 35 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 36 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 37 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

3. KVS-Diagramm:

Allgemein handelt es sich bei der Ansteuerung der Freigabeschütze um eine 2 aus 5 Auswahl der Lastabwurfrelais S. Ist für ein Freigabeschütz die Ansteuerfunktion gefunden, kann diese entsprechend auf die anderen Freigabeschütze übertragen werden. Die Ansteuerfunktion wird für das Schütz Q1 aus einem KVS-Diagramm bestimmt, das auf 5 Variablen erweitert wird.



$$Q1 = S4S3S2 \vee S5S3S2 \vee S5S4S2 \vee S5S4S3$$

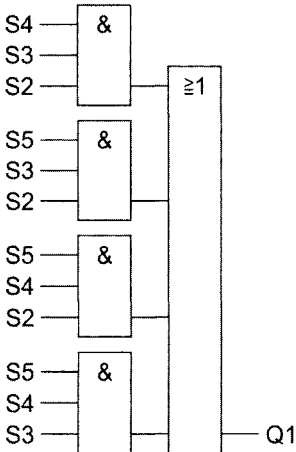
Aus der gefundenen Ansteuerfunktion ist zu entnehmen, dass sich diese aus der Kombination der jeweiligen Lastabwurfrelais S, die nicht zum eigenen Schütz gehören, zusammensetzt.

Die anderen Ansteuerfunktionen lauten somit:

$$Q2 = S4S3S1 \vee S5S3S1 \vee S5S4S1 \vee S5S4S3; Q3 = S4S2S1 \vee S5S2S1 \vee S5S4S1 \vee S5S4S2;$$

$$Q4 = S3S2S1 \vee S5S2S1 \vee S5S3S1 \vee S5S3S2; Q5 = S3S2S1 \vee S4S2S1 \vee S4S3S1 \vee S4S3S2;$$

4. Funktionsplan

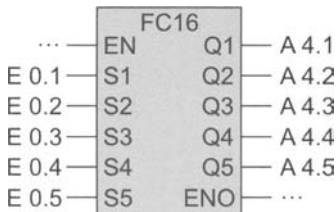


5. Deklarationstabelle FC15:

| Name | Datentyp |
|------------|----------|
| IN | |
| S1 | Bool |
| S2 | Bool |
| S3 | Bool |
| S4 | Bool |
| S5 | Bool |
| OUT | |
| Q1 | Bool |
| Q2 | Bool |
| Q3 | Bool |
| Q4 | Bool |
| Q5 | Bool |

6. Realisierung:

Aufruf im OB1:



Programmierung und Aufruf des Bausteins siehe beiliegende CD oder <http://automatisieren-mit-sps.de>

STEP7-Lösungen → 2_2_06

Lösung Lernaufgabe 2.7

Aufg. S. 15

1. Zuordnungstabelle der Eingänge und Ausgänge:

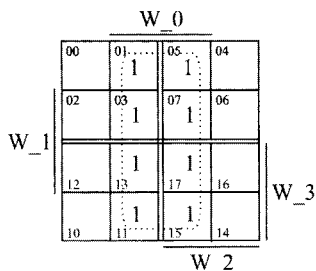
| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|-------------------------|--------|----------|-------------------------|---------|
| Dualwert 0 | W_0 | BOOL | Eingeschaltet W_0 = 1 | E 0.0 |
| Dualwert 1 | W_1 | BOOL | Eingeschaltet W_1 = 1 | E 0.1 |
| Dualwert 2 | W_2 | BOOL | Eingeschaltet W_2 = 1 | E 0.2 |
| Dualwert 3 | W_3 | BOOL | Eingeschaltet W_3 = 1 | E 0.3 |
| Ausgangsvariable | | | | |
| BCD-Wert 1_0 | B1_0 | BOOL | Erforderlich BCD1_0 = 1 | A 4.0 |
| BCD-Wert 1_1 | B1_1 | BOOL | Erforderlich BCD1_1 = 1 | A 4.1 |
| BCD-Wert 1_2 | B1_2 | BOOL | Erforderlich BCD1_2 = 1 | A 4.2 |
| BCD-Wert 1_3 | B1_3 | BOOL | Erforderlich BCD1_3 = 1 | A 4.3 |
| BCD-Wert 2_0 | B2_0 | BOOL | Erforderlich BCD2_0 = 1 | A 4.4 |

2. Funktionstabelle:

| Dez | Okt. | W 3 | W 2 | W 1 | W 0 | B2 0 | B1 3 | B1 2 | B1 1 | B1 0 |
|-----|------|-----|-----|-----|-----|------|------|------|------|------|
| 00 | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 01 | 01 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 02 | 02 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 03 | 03 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 04 | 04 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 05 | 05 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 06 | 06 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 07 | 07 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 08 | 10 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 09 | 11 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 10 | 12 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 11 | 13 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 12 | 14 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 13 | 15 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 14 | 16 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 15 | 17 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |

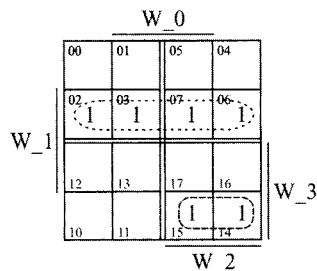
3. KVS-Diagramme:

BCD1_0:



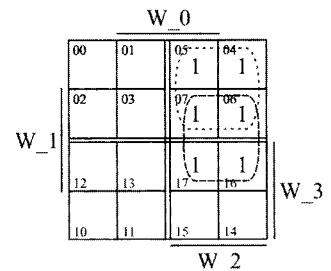
$$BCD1_0 = W_0$$

BCD1_1:



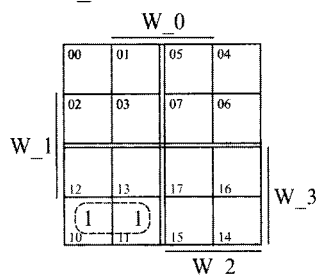
$$BCD1_1 = \overline{W_3} \& W_1 \vee W_3 \& W_2 \& \overline{W_1}$$

BCD1_2:



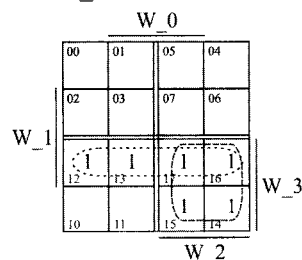
$$BCD1_2 = \overline{W_3} \& W_2 \vee W_2 \& W_1$$

BCD1_3:



$$BCD1_3 = W_3 \& \overline{W_2} \& \overline{W_1};$$

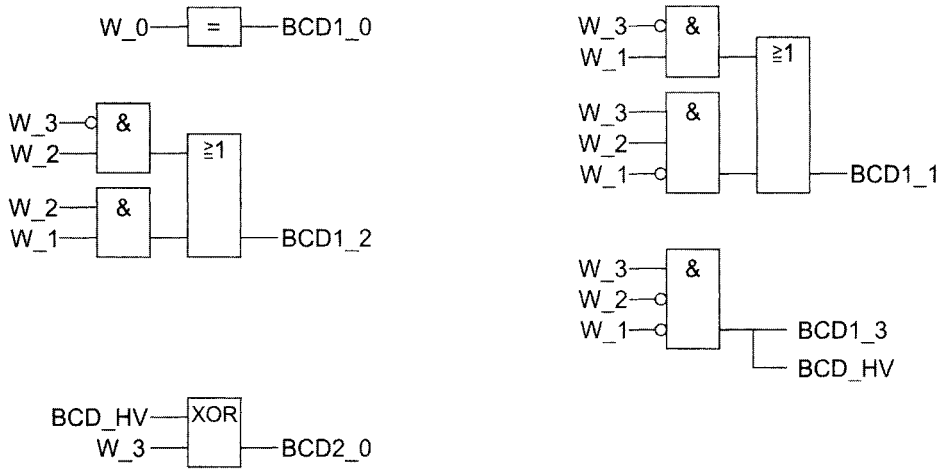
BCD2_0:



$$BCD2_0 = W_3 \& W_1 \vee W_3 \& W_2$$

Aus der Funktionstabelle ergibt sich auch: $BCD2_0 = W_3 \neq BCD1_3$

4. Funktionsplan



Hinweis: Damit bei der Funktion keine IN_OUT-Variablen auftreten, wurde die lokale Hilfsvariable BCD_HV eingeführt.

5. Deklarationstabelle FC17:

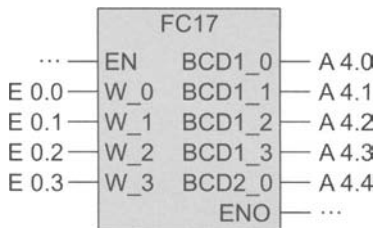
| Name | Datentyp |
|-----------|----------|
| IN | |
| W_0 | Bool |
| W_1 | Bool |
| W_2 | Bool |
| W_3 | Bool |

| Name | Datentyp |
|------------|----------|
| OUT | |
| BCD1_0 | Bool |
| BCD1_1 | Bool |
| BCD1_2 | Bool |
| BCD1_3 | Bool |
| BCD2_0 | Bool |

| Name | Datentyp |
|-------------|----------|
| TEMP | |
| BCD_HV | Bool |

6. Realisierung:

Aufruf im OBI:



Programmierung und Aufruf des Bausteins siehe beiliegende CD oder <http://automatisieren-mit-sps.de> STEP7-Lösungen → 2_2_07

10.3 Lösungen der Lernaufgaben Kapitel 3

Lösung Lernaufgabe 3.1

Aufg. S. 24

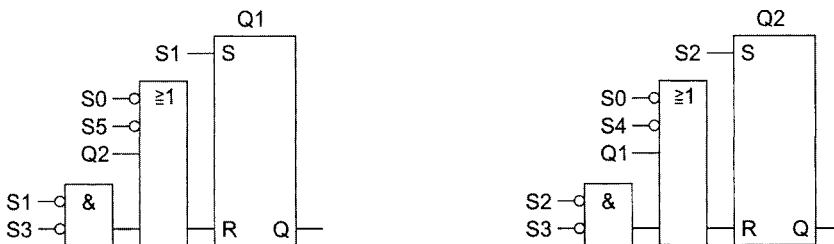
1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | | Adresse |
|----------------------------|--------|----------|--------------------|--------|---------|
| Taster STOPP | S0 | BOOL | Betätigt | S0 = 0 | E 0.0 |
| Taster AUF | S1 | BOOL | Betätigt | S1 = 1 | E 0.1 |
| Taster ZU | S2 | BOOL | Betätigt | S2 = 1 | E 0.2 |
| Wahlschalter Autom./Tippen | S3 | BOOL | Automatik | S3 = 1 | E 0.3 |
| Initiator Tor zu | S4 | BOOL | Betätigt | S4 = 0 | E 0.4 |
| Initiator Tor auf | S5 | BOOL | Betätigt | S5 = 0 | E 0.5 |
| Ausgangsvariable | | | | | |
| Schütz Tor auf | Q1 | BOOL | Angezogen | Q1 = 1 | A 4.1 |
| Schütz Tor zu | Q2 | BOOL | Angezogen | Q2 = 1 | A 4.2 |

2. RS-Tabelle:

| Zu betätigende Speicherglieder | Bedingungen für das Setzen | Bedingungen für das Rücksetzen (Verriegelungen) |
|--------------------------------|----------------------------|--|
| Schütz Tor auf K1 | S1 | $\overline{S0} \vee \overline{S5} \vee Q2 \vee \overline{S3} \& \overline{S1}$ |
| Schütz Tor zu K2 | S2 | $\overline{S0} \vee \overline{S4} \vee Q1 \vee \overline{S3} \& \overline{S2}$ |

3. Funktionsplan:



4. Bausteintyp:

Der Funktionsplan zeigt, dass der zu verwendende Baustein nur Eingabe- und Ausgabevariablen besitzt. Da keine lokalen statischen Variablen benötigt werden, genügt der Bausteintyp Funktion FC. Die Signalzustände der beiden RS-Speicherglieder werden in den zugehörigen SPS-Ausgängen gespeichert. Auch bei mehrmaligem Aufruf des Bibliothekbausteins innerhalb eines Programms zur Steuerung mehrerer Werkstore, werden die Signalzustände für die weiteren Motorschütze jeweils in den zugeordneten Ausgängen gespeichert.

5. Deklarationstabelle FC11:

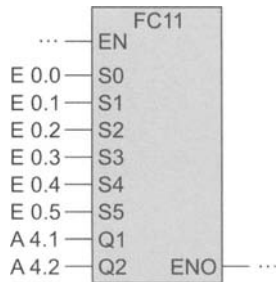
| Name | Datentyp |
|------|----------|
| IN | |
| S0 | Bool |
| S1 | Bool |
| S2 | Bool |

| Name | Datentyp |
|------|----------|
| IN | |
| S3 | Bool |
| S4 | Bool |
| S5 | Bool |

| Name | Datentyp |
|--------|----------|
| IN_OUT | |
| Q1 | Bool |
| Q2 | Bool |

6. Realisierung:

Aufruf im OB1:



Programmierung und Aufruf des Bausteins siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de>
STEP7-Lösungen → 3_2_01

Lösung Lernaufgabe 3.2 Aufg. S. 25

1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | | Adresse |
|------------------------|--------|----------|--------------------|--------|---------|
| Vollmelder Behälter 1 | S1 | BOOL | Behälter 1 voll | S1 = 0 | E 0.0 |
| Vollmelder Behälter 2 | S3 | BOOL | Behälter 2 voll | S3 = 0 | E 0.2 |
| Vollmelder Behälter 3 | S5 | BOOL | Behälter 3 voll | S5 = 0 | E 0.4 |
| Vollmelder Behälter 4 | S7 | BOOL | Behälter 4 voll | S7 = 0 | E 0.6 |
| Leermeldung Behälter 1 | S2 | BOOL | Behälter 1 leer | S2 = 1 | E 0.1 |
| Leermeldung Behälter 2 | S4 | BOOL | Behälter 2 leer | S4 = 1 | E 0.3 |
| Leermeldung Behälter 3 | S6 | BOOL | Behälter 3 leer | S6 = 1 | E 0.5 |
| Leermeldung Behälter 4 | S8 | BOOL | Behälter 4 leer | S8 = 1 | E 0.7 |
| Ausgangsvariable | | | | | |
| Motorschütz Pumpe 1 | Q1 | BOOL | Pumpe läuft | Q1 = 1 | A 4.1 |
| Motorschütz Pumpe 2 | Q2 | BOOL | Pumpe läuft | Q2 = 1 | A 4.2 |
| Motorschütz Pumpe 3 | Q3 | BOOL | Pumpe läuft | Q3 = 1 | A 4.3 |
| Motorschütz Pumpe 4 | Q4 | BOOL | Pumpe läuft | Q4 = 1 | A 4.4 |

2. Tabelle der Zuschaltbedingungen:

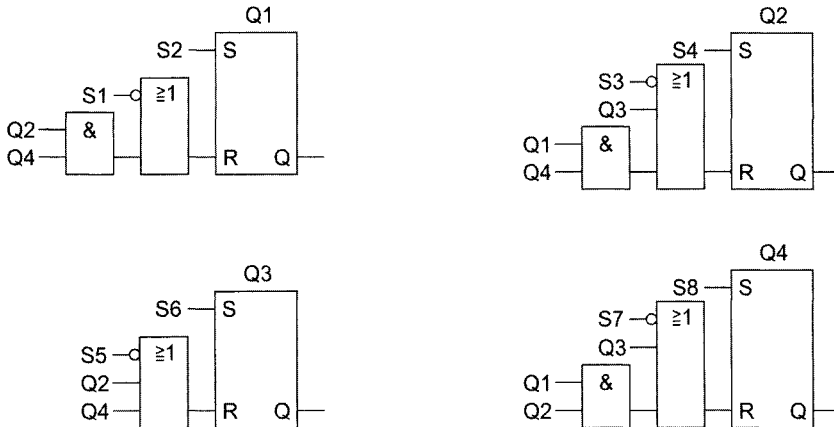
| Nicht zugeschaltet werden darf | Pumpe P1 3kW | Pumpe P2 4kW | Pumpe P3 7kW | Pumpe P4 5kW |
|---|-----------------|--------------------------|---|--------------------------|
| Bei Betrieb der Pumpen - /Kombinationen (Nicht aufgeführt sind Pumpen oder Pumpenkombinationen die nicht in Betrieb sein können.) | P2 & P4 | P3 P1 & P3 P1 & P4 | P2 P4 P1 & P2 P1 & P4 P2 & P4 | P3 P1 & P2 P1 & P3 |
| Vereinfachte Bedingungen | P2 & P4 | P3 P1 & P4 | P2 P4 | P3 P1 & P2 |

3. RS-Tabelle:

| Zu betätigende Speicherglieder | Bedingungen für das Setzen | Bedingungen für das Rücksetzen (Verriegelungen) |
|--------------------------------|----------------------------|---|
| Q1 (Pumpe1) | S2 | $\overline{S1} \vee Q2 \& Q4$ |
| Q2 (Pumpe2) | S4 | $\overline{S3} \vee Q3 \vee Q1 \& Q4$ |

| | | |
|-------------|----|---------------------------------------|
| Q3 (Pumpe3) | S6 | $\overline{S5} \vee Q2 \vee Q4$ |
| Q4 (Pumpe4) | S8 | $\overline{S7} \vee Q3 \vee Q1 \& Q2$ |

4. Funktionsplan:



5. Bausteintyp:

Der Funktionsplan zeigt, dass der zu verwendende Baustein nur Eingabe- und Ausgabevariablen besitzt. Da keine lokalen statischen Variablen benötigt werden, genügt der Bausteintyp Funktion FC. Die Signalzustände der beiden RS-Speicherglieder werden in den zugehörigen SPS-Ausgängen gespeichert.

6. Deklarationstabelle FC12:

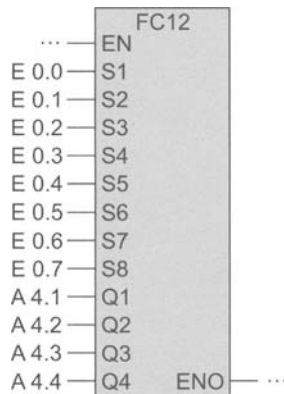
| Name | Datentyp |
|------|----------|
| IN | |
| S1 | Bool |
| S2 | Bool |
| S3 | Bool |
| S4 | Bool |

| Name | Datentyp |
|------|----------|
| IN | |
| S5 | Bool |
| S6 | Bool |
| S7 | Bool |
| S8 | Bool |

| Name | Datentyp |
|--------|----------|
| IN_OUT | |
| Q1 | Bool |
| Q2 | Bool |
| Q3 | Bool |
| Q4 | Bool |

7. Realisierung:

Aufruf im OB1:



Programmierung und Aufruf des Bausteins siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de>

STEP7-Lösungen → 3_2_02

Lösung Lernaufgabe 3.3

Aufg. S. 25

1. Zuordnungstabelle der Eingänge und Ausgänge:

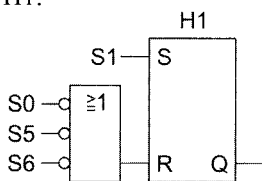
| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | | Adresse |
|--------------------------------|--------|----------|--------------------|--------|---------|
| Taster AUS | S0 | BOOL | Betätigt | S0 = 0 | E 0.0 |
| Taster EIN | S1 | BOOL | Betätigt | S1 = 1 | E 0.1 |
| Initiator Position links | S2 | BOOL | Betätigt | S2 = 0 | E 0.2 |
| Initiator Position rechts | S3 | BOOL | Betätigt | S3 = 0 | E 0.3 |
| Drehgeber | S4 | BOOL | M2 läuft | S4 = 1 | E 0.4 |
| Thermischer Auslöser Motor 2 | S5 | BOOL | Ausgelöst | S5 = 0 | E 0.5 |
| Thermischer Auslöser Motor 1 | S6 | BOOL | Ausgelöst | S6 = 0 | E 0.6 |
| Ausgangsvariable | | | | | |
| Rechtslauf Schlittenantrieb M1 | Q1 | BOOL | Angezogen | Q1 = 1 | A 4.1 |
| Linkslauf Schlittenantrieb M1 | Q2 | BOOL | Angezogen | Q2 = 1 | A 4.2 |
| Antriebsmotor M2 | Q3 | BOOL | Angezogen | Q3 = 1 | A 4.3 |
| Meldeleuchte AUS | H0 | BOOL | Leuchtet | H0 = 1 | A 4.4 |
| Meldeleuchte EIN | H1 | BOOL | Leuchtet | H1 = 1 | A 4.5 |

2. RS-Tabelle:

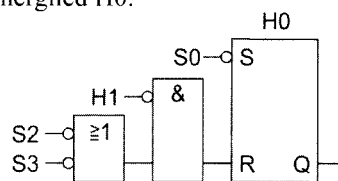
| | Zu betätigende Speicherglieder | Bedingungen für das Setzen | Bedingungen für das Rücksetzen (Verriegelungen) |
|---|--------------------------------|----------------------------|--|
| Reihenfolge der Speicherglieder, nach der die Logik durchdacht werden kann. | H1 | S1 | $\overline{S0} \vee \overline{S5} \vee \overline{S6}$ |
| | H0 | $\overline{S0}$ | $\overline{H1} \& (\overline{S2} \vee \overline{S3})$ |
| | Q3 | H1 | $\overline{S5} \vee \overline{S6} \vee H0(1 \rightarrow 0)$ |
| | Q2 | $\overline{Q1} \& H1$ | $\overline{S2} \vee \overline{S4} \vee \overline{S5} \vee \overline{S6}$ |
| | Q1 | $\overline{Q2} \& H1$ | $\overline{S3} \vee \overline{S4} \vee \overline{S5} \vee \overline{S6}$ |

3. Funktionsplan:

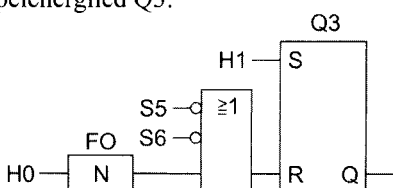
Speicherglied H1:



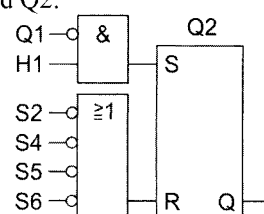
Speicherglied H0:



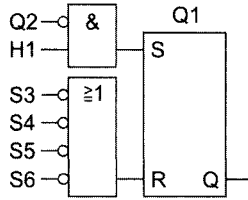
Speicherglied Q3:



Speicherglied Q2:



Speicherglied Q1:



4. Bausteintyp:

Für die Flankenauswertung bei H0 wird ein Flankenoperand FO benötigt, der als statische Variable zu deklarieren ist. Das erfordert die Verwendung des Bausteintyps FB.

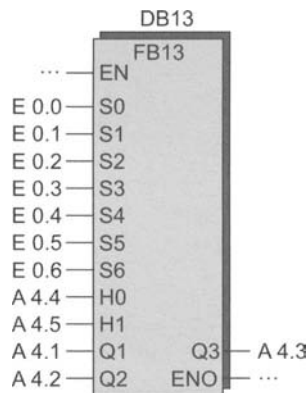
5. Deklarationstabelle FB13:

| Name | Datentyp | Anfangswert |
|------------|----------|-------------|
| IN | | |
| S0 | Bool | FALSE |
| S1 | Bool | FALSE |
| S3 | Bool | FALSE |
| S4 | Bool | FALSE |
| S5 | Bool | FALSE |
| S6 | Bool | FALSE |
| OUT | | |
| Q3 | Bool | FALSE |

| Name | Datentyp | Anfangswert |
|---------------|----------|-------------|
| IN_OUT | | |
| H0 | Bool | FALSE |
| H1 | Bool | FALSE |
| Q1 | Bool | FALSE |
| Q2 | Bool | FALSE |
| STAT | | |
| FO | Bool | FALSE |

6. Realisierung:

Aufruf im OB1:



Programmierung und Aufruf des Bausteins siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de>
STEP7-Lösungen → 3_2_03

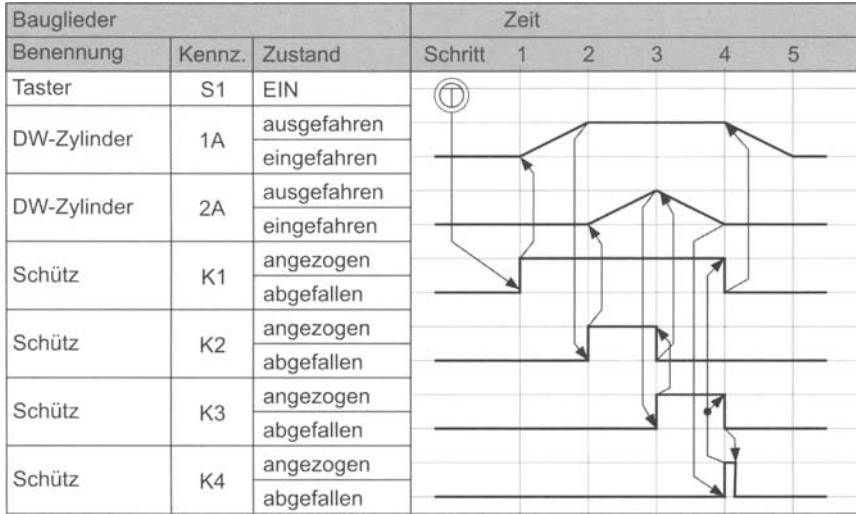
Lösung Lernaufgabe 3.4 Aufg. S. 26

1. Zuordnungstabelle der Eingänge und Ausgänge:

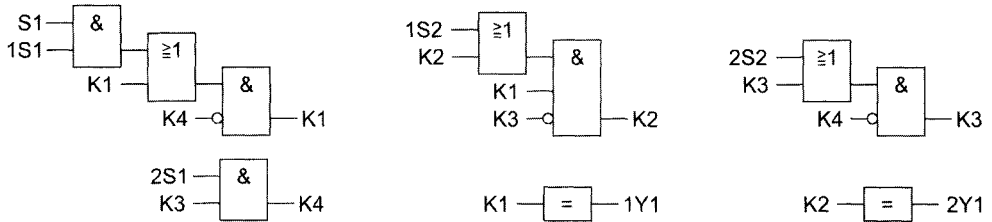
| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|---------------------------------|--------|----------|--------------------|---------|
| Taster EIN | S1 | BOOL | Betätigt S1 = 1 | E 0.1 |
| Initiator hintere Endl. Zyl. 1A | 1S1 | BOOL | Betätigt 1S1 = 1 | E 0.2 |
| Initiator vordere Endl. Zyl. 1A | 1S2 | BOOL | Betätigt 1S2 = 1 | E 0.3 |
| Initiator hintere Endl. Zyl. 2A | 2S1 | BOOL | Betätigt 2S1 = 1 | E 0.4 |
| Initiator vordere Endl. Zyl. 2A | 2S2 | BOOL | Betätigt 2S2 = 1 | E 0.5 |

| Ausgangsvariable | | | | |
|-----------------------------|-----|------|---------------------------|-------|
| Magnetspule Zylinder 1A vor | 1Y1 | BOOL | Zyl. 1A fährt aus 1Y1 = 1 | A 4.1 |
| Magnetspule Zylinder 2A vor | 2Y1 | BOOL | Zyl. 2A fährt aus 2Y1 = 1 | A 4.2 |

2. Funktionsdiagramm:



3. Funktionsplan:



4. Bausteutyp und Deklarationstabelle:

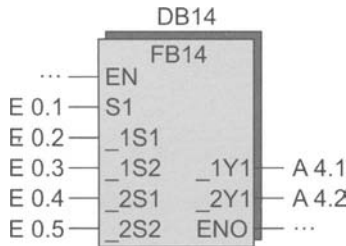
Die lokalen Variablen K1, K2 und K3 besitzen eine Selbsthaltung. Das erfordert die Verwendung des Bausteutyps FB. Die Zuweisung der temporären Variablen K4 muss im ersten Netzwerk erfolgen, da der Signalwert bei der Zuweisung von K1, K2 und K3 abgefragt wird.

Deklarationstabelle FB14:

| Name | Datentyp | Anfangswert |
|-----------|----------|-------------|
| IN | | |
| S1 | Bool | FALSE |
| _1S1 | Bool | FALSE |
| _1S2 | Bool | FALSE |
| _2S1 | Bool | FALSE |
| _2S2 | Bool | FALSE |

| Name | Datentyp | Anfangswert |
|-------------|----------|-------------|
| OUT | | |
| _1Y1 | Bool | FALSE |
| _2Y1 | Bool | FALSE |
| STAT | | |
| K1 | Bool | FALSE |
| K2 | Bool | FALSE |
| K3 | Bool | FALSE |
| TEMP | | |
| K4 | Bool | FALSE |

5. Realisierung: OB1:



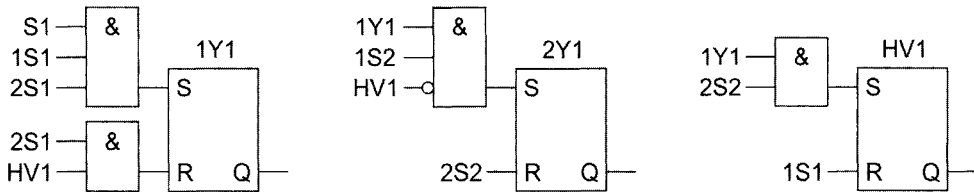
Programmierung und Aufruf des Bausteins siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de>

STEP7-Lösungen → 3_2_04a

6. RS-Tabelle:

| Zu betätigende Speicherglieder | Bedingungen für das Setzen | Bedingungen für das Rücksetzen (Verriegelungen) |
|--------------------------------|----------------------------|---|
| 1Y1 (Magnetspule Zyl. 1A) | S1 & 1S1 & 2S1 | 2S1 & HV1 |
| 2Y1 (Magnetspule Zyl. 2A) | 1Y1 & 1S2 & HV1 | 2S2 |
| HV1 (Hilfsvariable) | 1Y1 & 2S2 | 1S1 |

7. Funktionsplan:



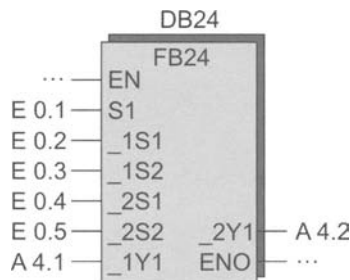
8. Deklarationstabelle FB24:

| Name | Datentyp | Anfangswert |
|------|----------|-------------|
| IN | | |
| S1 | Bool | FALSE |
| _1S1 | Bool | FALSE |
| _1S2 | Bool | FALSE |
| _2S1 | Bool | FALSE |
| _2S2 | Bool | FALSE |

| Name | Datentyp | Anfangswert |
|--------|----------|-------------|
| OUT | | |
| _2Y1 | Bool | FALSE |
| IN_OUT | | |
| _1Y1 | Bool | FALSE |
| STAT | | |
| HV1 | Bool | FALSE |

9. Realisierung:

Aufruf im OB1:



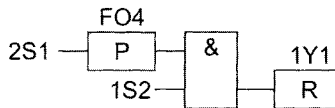
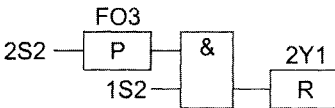
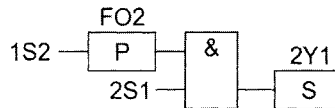
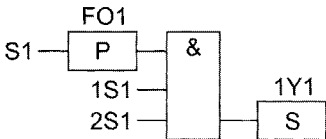
Programmierung und Aufruf des Bausteins siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de>

STEP7-Lösungen → 3_2_04b

10. Schaltfolgetabelle

| Schritt | Bedingung | Setzen | Rücksetzen |
|---------|----------------------|--------|------------|
| 1 | S1 (0→1) & 1S1 & 2S1 | 1Y1 | |
| 2 | 1S2 (0→1) & 2S1 | 2Y1 | |
| 3 | 2S2 (0→1) & 1S2 | | 2Y1 |
| 4 | 2S1 (0→1) & 1S2 | | 1Y1 |

11. Funktionsplan



12. Bausteintyp und Deklarationstabelle:

Die Flankenoperanden FO1 bis FO4 sind als lokale statische Variablen zu deklarieren. Deshalb ist der Bausteintyp Funktionsbaustein FB erforderlich.

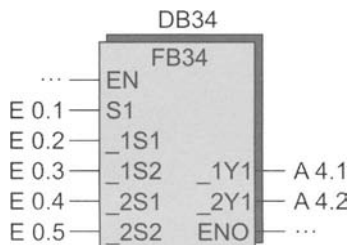
Deklarationstabelle FB34:

| Name | Datentyp | Anfangswert |
|------------|----------|-------------|
| IN | | |
| S1 | Bool | FALSE |
| _1S1 | Bool | FALSE |
| _1S2 | Bool | FALSE |
| _2S1 | Bool | FALSE |
| _2S2 | Bool | FALSE |
| OUT | | |
| _1Y1 | Bool | FALSE |
| _2Y1 | Bool | FALSE |

| Name | Datentyp | Anfangswert |
|-------------|----------|-------------|
| STAT | | |
| FO1 | Bool | FALSE |
| FO2 | Bool | FALSE |
| FO3 | Bool | FALSE |
| FO4 | Bool | FALSE |

13. Realisierung:

Aufruf im OB1:



Programmierung und Aufruf des Bausteins siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de>
STEP7-Lösungen → 3_2_04c

Lösung Lernaufgabe 3.5

Aufg. S. 28

1. Zuordnungstabelle der Eingänge und Ausgänge:

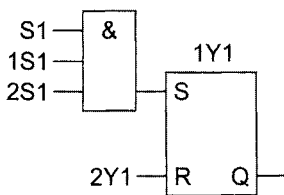
| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|---------------------------------|--------|----------|---------------------------|---------|
| Taster EIN | S1 | BOOL | Betätigt S1 = 1 | E 0.1 |
| Initiator hintere Endl. Zyl. 1A | 1S1 | BOOL | Betätigt 1S1 = 1 | E 0.2 |
| Initiator vordere Endl. Zyl. 1A | 1S2 | BOOL | Betätigt 1S2 = 1 | E 0.3 |
| Initiator hintere Endl. Zyl. 2A | 2S1 | BOOL | Betätigt 2S1 = 1 | E 0.4 |
| Initiator vordere Endl. Zyl. 2A | 2S2 | BOOL | Betätigt 2S2 = 1 | E 0.5 |
| Ausgangsvariable | | | | |
| Magnetspule Zylinder 1A vor | 1Y1 | BOOL | Zyl. 1A fährt aus 1Y1 = 1 | A 4.1 |
| Magnetspule Zylinder 1A zurück | 1Y2 | BOOL | Zyl. 1A fährt ein 1Y2 = 1 | A 4.2 |
| Magnetspule Zylinder 2A vor | 2Y1 | BOOL | Zyl. 2A fährt aus 2Y1 = 1 | A 4.3 |
| Magnetspule Zylinder 2A zurück | 2Y2 | BOOL | Zyl. 2A fährt ein 2Y2 = 1 | A 4.4 |

2. RS-Tabelle:

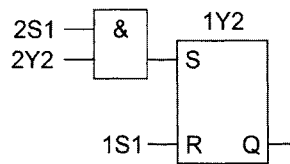
| Zu betätigende Speicherglieder | Bedingungen für das Setzen | Bedingungen für das Rücksetzen (Verriegelungen) |
|----------------------------------|----------------------------|---|
| 1Y1 (Magnetspule Zyl. 1A vor) | S1 & 1S1 & 2S1 | 2Y1 |
| 1Y2 (Magnetspule Zyl. 1A zurück) | 2S1 & 2Y2 | 1S1 |
| 2Y1 (Magnetspule Zyl. 2A vor) | 1S2 & 1Y1 | 2Y2 |
| 2Y2 (Magnetspule Zyl. 2A zurück) | 2S2 | 1Y2 |

3. Funktionsplan:

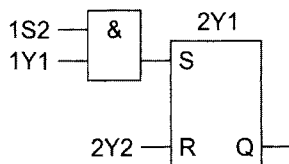
Ansteuerung Magnetspule 1Y1



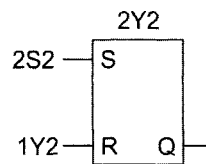
Ansteuerung Magnetspule 1Y2



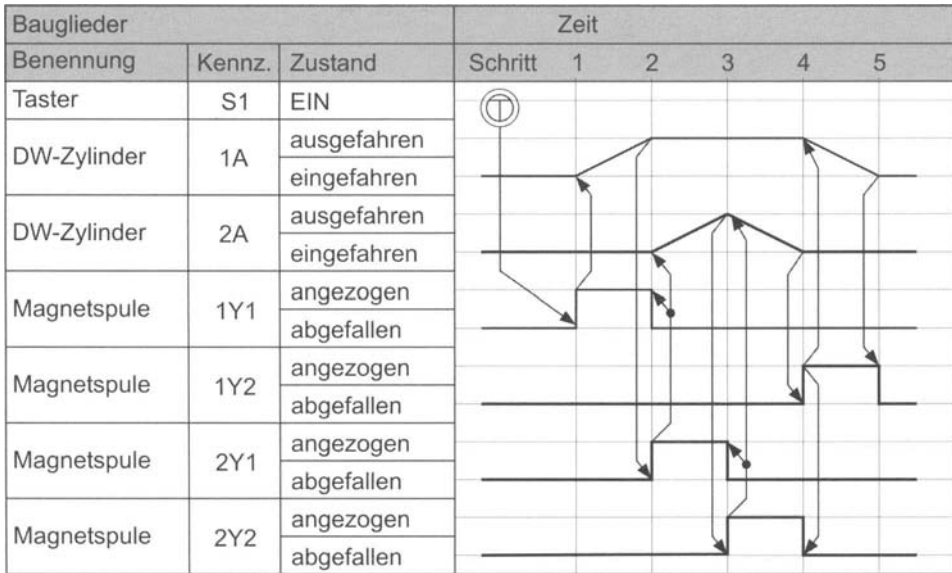
Ansteuerung Magnetspule 2Y1



Ansteuerung Magnetspule 2Y2



4. Funktionsdiagramm:



5. Bausteintyp und Deklarationstabelle:

Es genügt der Bausteintyp Funktion FC, da keine lokalen statischen Variablen erforderlich sind. Die Signalwerte der Elektromagnete Y1 bis Y4 werden in den SPS-Ausgängen gespeichert.

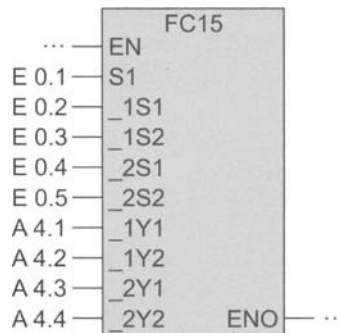
Deklarationstabelle FC15:

| Name | Datentyp |
|------|----------|
| IN | |
| S1 | Bool |
| _1S1 | Bool |
| _1S2 | Bool |
| _2S1 | Bool |
| _2S2 | Bool |

| Name | Datentyp |
|--------|----------|
| IN_OUT | |
| _1Y1 | Bool |
| _1Y2 | Bool |
| _2Y1 | Bool |
| _2Y2 | Bool |

6. Realisierung:

Aufruf im OB1:



Programmierung und Aufruf des Bausteins siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de>

STEP7-Lösungen → 3_2_05

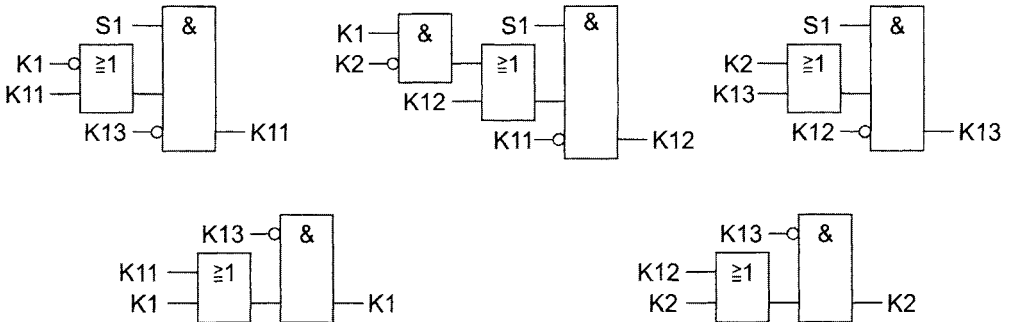
Lösung Lernaufgabe 3.6

Aufg. S. 29

1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|-------------------------|--------|----------|-------------------------|---------|
| Taster | S1 | BOOL | Betätigt S1 = 1 | E 0.1 |
| Ausgangsvariable | | | | |
| Schütz Ventilator 1 | K1 | BOOL | Ventilator läuft K1 = 1 | A 4.1 |
| Schütz Ventilator 2 | K2 | BOOL | Ventilator läuft K2 = 1 | A 4.2 |

2. Funktionsplan:



3. Bausteintyp und Deklarationstabelle:

Da die Hilfsschütze K11, K12 und K13 eine Selbsthaltung besitzen, sind sie als lokale statische Variablen zu deklarieren. Deshalb ist der Bausteintyp Funktionsbaustein FB erforderlich.

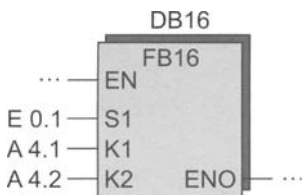
Deklarationstabelle FB16:

| Name | Datentyp | Anfangswert |
|--------|----------|-------------|
| IN | | |
| S1 | Bool | FALSE |
| IN_OUT | | |
| K1 | Bool | FALSE |
| K2 | Bool | FALSE |

| Name | Datentyp | Anfangswert |
|------|----------|-------------|
| STAT | | |
| K11 | Bool | FALSE |
| K12 | Bool | FALSE |
| K13 | Bool | FALSE |

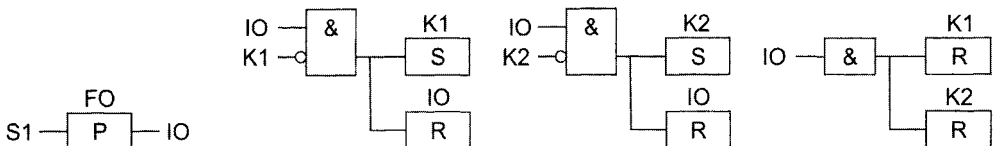
4. Realisierung:

OB1:



Programmierung und Aufruf des Bausteins siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de>
STEP7-Lösungen → 3_2_06a

5. Funktionsplan nach der Binäruntersetzermethode:



6. Bausteintyp und Deklarationstabelle:

Der Flankenoperand FO ist als lokale statische Variablen zu deklarieren. Deshalb ist der Bausteintyp Funktionsbaustein FB erforderlich.

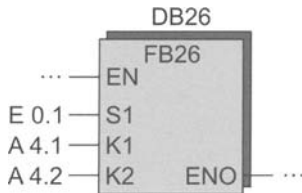
Deklarationstabelle FB26:

| Name | Datentyp | Anfangswert |
|--------|----------|-------------|
| IN | | |
| S1 | Bool | FALSE |
| IN_OUT | | |
| K1 | Bool | FALSE |
| K2 | Bool | FALSE |

| Name | Datentyp | Anfangswert |
|------|----------|-------------|
| STAT | | |
| FO | Bool | FALSE |
| TEMP | | |
| IO | Bool | |

7. Realisierung:

OB1:



Programmierung und Aufruf des Bausteins siehe: <http://www.automatisieren-mit-sps.de>
STEP7-Lösungen → 3_2_06b

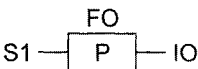
Lernaufgabe 3.7 Aufg. S. 30

1. Zuordnungstabelle der Eingänge und Ausgänge:

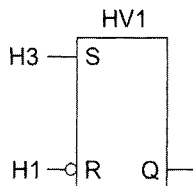
| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|-------------------------|--------|----------|-----------------------|---------|
| Taster | S1 | BOOL | Betätigt S1 = 1 | E 0.1 |
| Ausgangsvariable | | | | |
| Beleuchtungsgruppe 1 | H1 | BOOL | Beleuchtung an H1 = 1 | A 4.1 |
| Beleuchtungsgruppe 2 | H2 | BOOL | Beleuchtung an H2 = 1 | A 4.2 |
| Beleuchtungsgruppe 3 | H3 | BOOL | Beleuchtung an H3 = 1 | A 4.3 |

2. Funktionsplan:

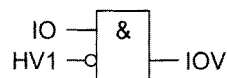
Flankenauswertung



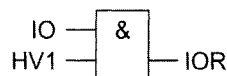
Richtungsumschalter



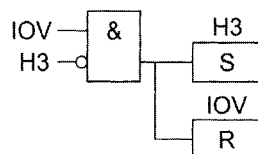
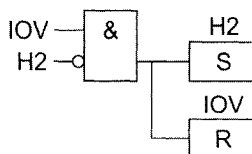
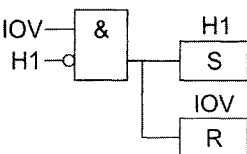
Vorwärtsimpuls



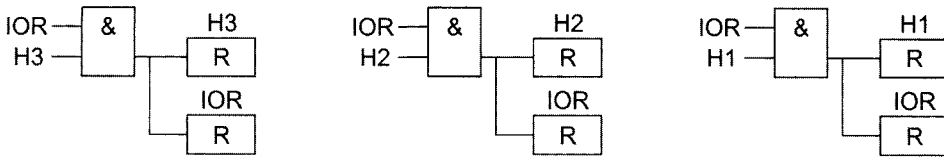
Rückwärtsimpuls



Vorwärts- oder Einschalt-Richtung



Rückwärts- oder Ausschalt-Richtung



3. Bausteintyp und Deklarationstabelle:

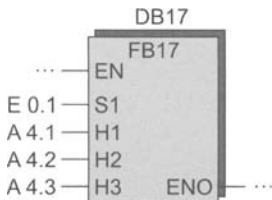
Der Signalzustand des Flankenoperanden FO und der Hilfsvariablen HV1 muss gespeichert werden. Die beiden Variablen sind deshalb als lokale statische Variablen zu deklarieren. Somit ist der Bausteintyp Funktionsbaustein FB zu verwenden.

Deklarationstabelle FB14:

| Name | Datentyp | Anfangswert | Name | Datentyp | Anfangswert |
|--------|----------|-------------|------|----------|-------------|
| IN | | | STAT | | |
| S1 | Bool | FALSE | FO | Bool | FALSE |
| IN_OUT | | | HV1 | Bool | FALSE |
| H1 | Bool | FALSE | TEMP | | |
| H2 | Bool | FALSE | IO | Bool | |
| H3 | Bool | FALSE | IOV | Bool | |
| | | | IOR | Bool | |

4. Realisierung:

OB1:



Programmierung und Aufruf des Bausteins siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de>
STEP7-Lösungen → 3_2_07

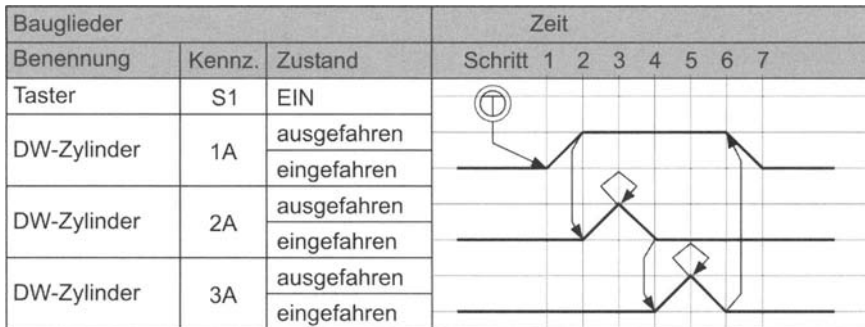
Lösung Lernaufgabe 3.8 Aufg. S. 31

1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|---------------------------------|--------|----------|---------------------------|---------|
| Taster EIN | S1 | BOOL | Betätigt S1 = 1 | E 0.1 |
| Initiator hintere Endl. Zyl. 1A | 1S1 | BOOL | Betätigt 1S1 = 1 | E 0.2 |
| Initiator vordere Endl. Zyl. 1A | 1S2 | BOOL | Betätigt 1S2 = 1 | E 0.3 |
| Initiator hintere Endl. Zyl. 2A | 2S1 | BOOL | Betätigt 2S1 = 1 | E 0.4 |
| Initiator vordere Endl. Zyl. 2A | 2S2 | BOOL | Betätigt 2S2 = 1 | E 0.5 |
| Initiator hintere Endl. Zyl. 3A | 3S1 | BOOL | Betätigt 3S1 = 1 | E 0.6 |
| Initiator vordere Endl. Zyl. 3A | 3S2 | BOOL | Betätigt 3S2 = 1 | E 0.7 |
| Ausgangsvariable | | | | |
| Magnetspule Zylinder 1A vor | 1Y1 | BOOL | Zyl. 1A fährt aus 1Y1 = 1 | A 4.1 |
| Magnetspule Zylinder 1A zurück | 1Y2 | BOOL | Zyl. 1A fährt ein 1Y2 = 1 | A 4.2 |

| | | | | |
|-----------------------------|-----|------|---------------------------|-------|
| Magnetspule Zylinder 2A vor | 2Y1 | BOOL | Zyl. 2A fährt aus 2Y1 = 1 | A 4.3 |
| Magnetspule Zylinder 3A vor | 3Y1 | BOOL | Zyl. 3A fährt aus 3Y2 = 1 | A 4.4 |

2. Funktionsdiagramm:

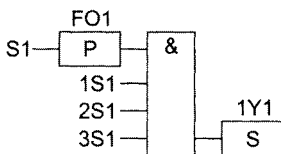


3. Schaltfolgetabelle

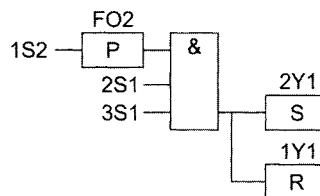
| Schritt | Bedingung | Setzen | Rücksetzen |
|---------|----------------------------|--------|------------|
| 1 | S1 (0→1) & 1S1 & 2S1 & 3S1 | 1Y1 | |
| 2 | 1S2 (0→1) & 2S1 & 3S1 | 2Y1 | 1Y1 |
| 3 | 1S2 & 2S2 (0→1) & 3S1 | | 2Y1 |
| 4 | 1S2 & 2S1 (0→1) & 3S1 | 3Y1 | |
| 5 | 1S2 & 2S1 & 3S2 (0→1) | | 3Y1 |
| 6 | 1S2 & 2S1 & 3S1 (0→1) | 1Y2 | |
| 7 | 1S1 (0→1) & 2S1 & 3S1 | | 1Y2 |

4. Funktionsplan:

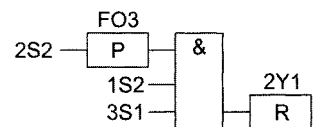
Schritt 1



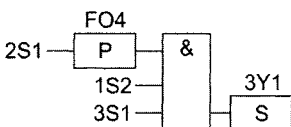
Schritt 2



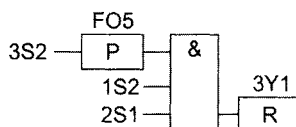
Schritt 3



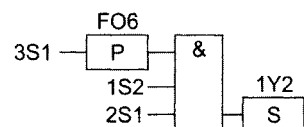
Schritt 4



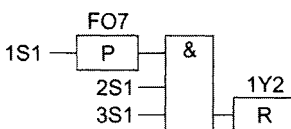
Schritt 5



Schritt 6



Schritt 7



5. Bausteintyp und Deklarationstabelle:

Die Flankenoperanden FO1 bis FO7 sind als lokale statische Variablen zu deklarieren. Deshalb ist der Bausteintyp Funktionsbaustein FB erforderlichlich.

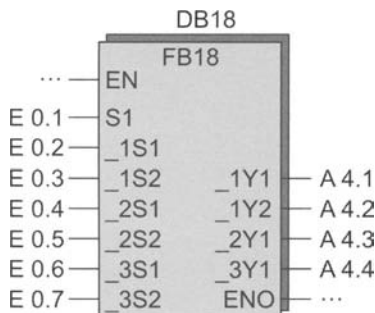
Deklarationstabelle FB18:

| Name | Datentyp | Anfangswert |
|------------|----------|-------------|
| IN | | |
| S1 | Bool | FALSE |
| _1S1 | Bool | FALSE |
| _1S2 | Bool | FALSE |
| _2S1 | Bool | FALSE |
| _2S2 | Bool | FALSE |
| _3S1 | Bool | FALSE |
| _3S2 | Bool | FALSE |
| OUT | | |
| _1Y1 | Bool | FALSE |
| _1Y2 | Bool | FALSE |

| Name | Datentyp | Anfangswert |
|-------------|----------|-------------|
| _2Y1 | Bool | FALSE |
| _3Y1 | Bool | FALSE |
| STAT | | |
| FO1 | Bool | FALSE |
| FO2 | Bool | FALSE |
| FO3 | Bool | FALSE |
| FO4 | Bool | FALSE |
| FO5 | Bool | FALSE |
| FO6 | Bool | FALSE |
| FO7 | Bool | FALSE |

6. Realisierung:

OB1:



Programmierung und Aufruf des Bausteins siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de>
STEP7-Lösungen → 3_2_08

Lösung Lernaufgabe 3.9

Aufg. S. 32

1. Zuordnungstabelle der Eingänge und Ausgänge:

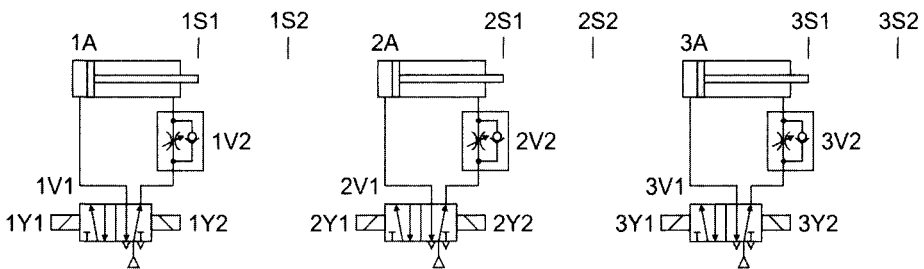
| Eingangsvariable | Symbol | Datent. | Logische Zuordnung | Adresse |
|---------------------------------|--------|---------|---------------------------|---------|
| Taster EIN | S1 | BOOL | Betätigt S1 = 1 | E 0.1 |
| Initiator hintere Endl. Zyl. 1A | 1S1 | BOOL | Betätigt 1S1 = 1 | E 0.2 |
| Initiator vordere Endl. Zyl. 1A | 1S2 | BOOL | Betätigt 1S2 = 1 | E 0.3 |
| Initiator hintere Endl. Zyl. 2A | 2S1 | BOOL | Betätigt 2S1 = 1 | E 0.4 |
| Initiator vordere Endl. Zyl. 2A | 2S2 | BOOL | Betätigt 2S2 = 1 | E 0.5 |
| Initiator hintere Endl. Zyl. 3A | 3S1 | BOOL | Betätigt 3S1 = 1 | E 0.6 |
| Initiator vordere Endl. Zyl. 3A | 3S2 | BOOL | Betätigt 3S2 = 1 | E 0.7 |
| Ausgangsvariable | | | | |
| Magnetspule Zylinder 1A vor | 1Y1 | BOOL | Zyl. 1A fährt aus 1Y1 = 1 | A 4.1 |
| Magnetspule Zylinder 1A zurück | 1Y2 | BOOL | Zyl. 1A fährt ein 1Y2 = 1 | A 4.2 |
| Magnetspule Zylinder 2A vor | 2Y1 | BOOL | Zyl. 2A fährt aus 2Y1 = 1 | A 4.3 |

| | | | | |
|--------------------------------|-----|------|---------------------------|-------|
| Magnetspule Zylinder 2A zurück | 2Y2 | BOOL | Zyl. 2A fährt ein 2Y2 = 1 | A 4.4 |
| Magnetspule Zylinder 3A vor | 3Y1 | BOOL | Zyl. 3A fährt aus 3Y1 = 1 | A 4.5 |
| Magnetspule Zylinder 3A zurück | 3Y2 | BOOL | Zyl. 3A fährt ein 3Y2 = 1 | A 4.6 |

2. Funktionsdiagramm:

| Bauglieder | | | Zeit | | | | | | |
|-------------|--------|----------------------------|---------|---|---|---|---|---|---|
| Benennung | Kennz. | Zustand | Schritt | 1 | 2 | 3 | 4 | 5 | 6 |
| Taster | S1 | EIN | | | | | | | |
| DW-Zylinder | 1A | ausgefahren eingefahren | | | | | | | |
| DW-Zylinder | 2A | ausgefahren eingefahren | | | | | | | |
| DW-Zylinder | 3A | ausgefahren eingefahren | | | | | | | |

3. Pneumatischer Schaltplan

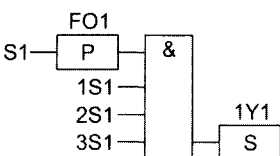


4. Schaltfolgetabelle

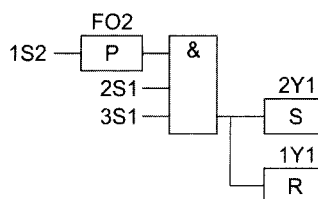
| Schritt | Bedingung | Setzen | Rücksetzen |
|---------|---|----------|------------|
| 1 | S1 (0→1) & 1S1 & 2S1 & 3S1 | 1Y1 | |
| 2 | 1S2 (0→1) & 2S1 & 3S1 | 2Y1 | 1Y1 |
| 3 | 1S2 & 2S2 (0→1) & 3S1 | 2Y2 | 2Y1 |
| 4 | 1S2 & 2S1 (0→1) & 3S1 | 3Y1; 1Y2 | 2Y2 |
| 5 | 1S1 (0→1) & 2S1 & 3S2 v 1S1 & 2S1 & 3S2 (0→1) | 3Y2 | 3Y1; 1Y2 |
| 6 | 1S1 & 2S1 & 3S1 (0→1) | | 3Y2 |

5. Funktionsplan:

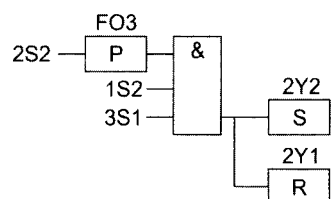
Schritt 1



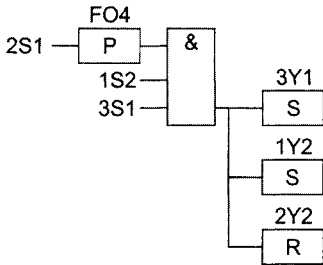
Schritt 2



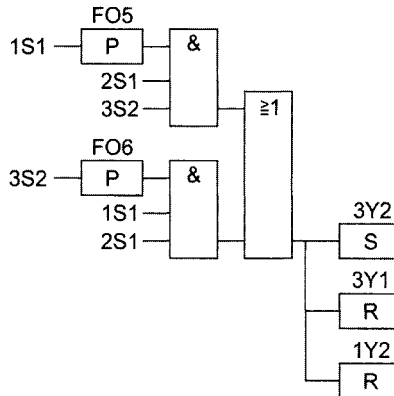
Schritt 3



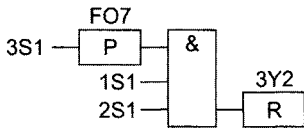
Schritt 4



Schritt 5



Schritt 6



6. Bausteintyp und Deklarationstabelle:

Die Flankenoperanden FO1 bis FO7 sind als lokale statische Variablen zu deklarieren. Deshalb ist der Bausteintyp Funktionsbaustein FB erforderlich.

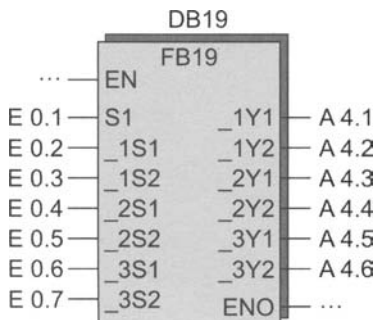
Deklarationstabelle FB19:

| Name | Datentyp | Anfangswert |
|------------|----------|-------------|
| IN | | |
| S1 | Bool | FALSE |
| _1S1 | Bool | FALSE |
| _1S2 | Bool | FALSE |
| _2S1 | Bool | FALSE |
| _2S2 | Bool | FALSE |
| _3S1 | Bool | FALSE |
| _3S2 | Bool | FALSE |
| OUT | | |
| _1Y1 | Bool | FALSE |
| _1Y2 | Bool | FALSE |

| Name | Datentyp | Anfangswert |
|-------------|----------|-------------|
| _2Y1 | Bool | FALSE |
| _2Y2 | Bool | FALSE |
| _3Y1 | Bool | FALSE |
| _3Y2 | Bool | FALSE |
| STAT | | |
| FO1 | Bool | FALSE |
| FO2 | Bool | FALSE |
| FO3 | Bool | FALSE |
| FO4 | Bool | FALSE |
| FO5 | Bool | FALSE |
| FO6 | Bool | FALSE |
| FO7 | Bool | FALSE |

7. Realisierung:

OB1:



Programmierung und Aufruf des Bausteins siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de>

STEP7-Lösungen → 3_2_09

10.4 Lösungen der Lernaufgaben Kapitel 4

Lösung Lernaufgabe 4.1

Aufg. S. 46

1. Zuordnungstabelle der Eingänge und Ausgänge für einen Kompressorantrieb:

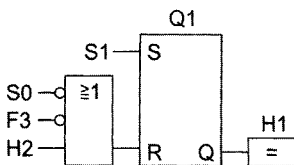
| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|----------------------|--------|----------|--------------------------|---------|
| Taster AUS | S0 | BOOL | Betätigt S0 = 0 | E 0.0 |
| Taster EIN | S1 | BOOL | Betätigt S1 = 1 | E 0.1 |
| Überstromschutz | F3 | BOOL | Motor überlastet F3 = 0 | E 0.2 |
| Kontakt Schütz K1 | Q1R | BOOL | Schütz angezogen Q1R = 1 | E 0.3 |
| Kontakt Schütz K2 | Q2R | BOOL | Schütz angezogen Q2R = 1 | E 0.4 |
| Kontakt Schütz K3 | Q3R | BOOL | Schütz angezogen Q3R = 1 | E 0.5 |
| Ausgangsvariable | | | | |
| Netzschütz | Q1 | BOOL | Angezogen Q1 = 1 | A 4.1 |
| Sternschütz | Q2 | BOOL | Angezogen Q2 = 1 | A 4.2 |
| Dreieckschütz | Q3 | BOOL | Angezogen Q3 = 1 | A 4.3 |
| Meldeleuchte Betrieb | H1 | BOOL | Leuchtet H1 = 1 | A 4.4 |
| Störungsleuchte | H2 | BOOL | Leuchtet H2 = 1 | A 4.5 |

2. Erweiterte RS-Tabelle:

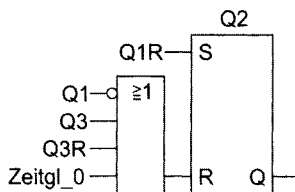
| Zu betätigende Speicher- und Zeitglieder | Bedingungen für das Setzen bzw. Starten | Bedingungen für das Rücksetzen |
|--|---|--|
| Netzschütz, Meldeleuchte (Q1), (H1) | S1 | $\overline{S0} \vee \overline{F3} \vee H2$ |
| Sternschütz (Q2) | Q1R | $\overline{Q1} \vee Q3 \vee Q3R \vee \text{Zeitgl_0}$ |
| Dreieckschütz (Q3) | Q1R & Zeitgl_0 | $\overline{Q1} \vee Q2 \vee Q2R$ |
| SE-Zeitglied St./Dreie. (Zeitgl_0) | Q2R | |
| SE-Zeitglied Überw. Q1 (Zeitgl_1) | $Q1 \& \overline{Q1R} \vee \overline{Q1} \& Q1R$ | |
| SE-Zeitglied Überw. Q2 (Zeitgl_2) | $Q2 \& \overline{Q2R} \vee \overline{Q2} \& Q2R$ | |
| SE-Zeitglied Überw. Q3 (Zeitgl_3) | $Q3 \& \overline{Q3R} \vee \overline{Q3} \& Q3R$ | |
| Störungsleuchte (H2) | $\overline{F3} \vee \text{Zeitgl_1} \vee \text{Zeitgl_2} \vee \text{Zeitgl_3}$ | $\overline{S0}$ |

3. Funktionsplan

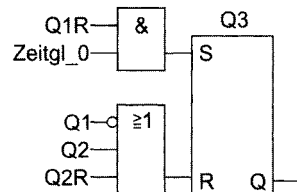
Netzschütz Q1



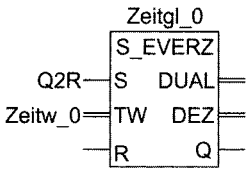
Sternschütz Q2



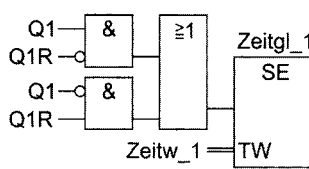
Dreieckschütz Q3



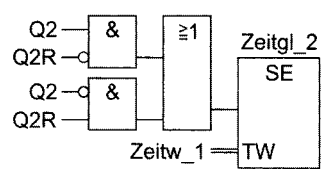
SE-Zeitglied Stern/Dreieck



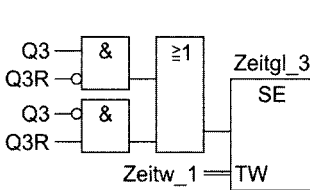
SE-Zeitglied Überw. Q1



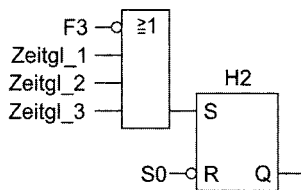
SE-Zeitglied Überw. Q2



SE-Zeitglied Überw. Q3



Störungsleuchte H2



Hinweis: Da die Überwachungszeit für alle drei Leistungsschütze gleich ist, wird für alle drei Zeitglieder der gleiche Zeitwert „Zeitw_1“ verwendet.

4. Bausteintyp:

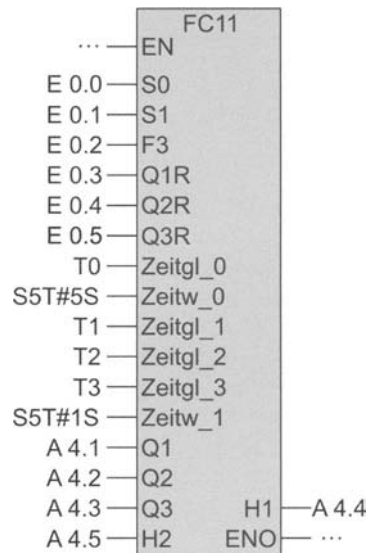
Es genügt der Bausteintyp Funktion FC, da keine lokalen statischen Variablen erforderlich sind. Die Signalwerte der Schütze Q1 bis Q3 werden in den SPS-Ausgängen gespeichert.

5. Deklarationstabelle FC 11:

| Name | Datentyp |
|---------------|----------|
| IN | |
| S0 | Bool |
| S1 | Bool |
| F3 | Bool |
| Q1R | Bool |
| Q2R | Bool |
| Q3R | Bool |
| Zeitgl_0 | Timer |
| Zeitw_0 | S5Time |
| Zeitgl_1 | Timer |
| Zeitgl_2 | Timer |
| Zeitgl_3 | Timer |
| Zeitw_1 | S5Time |
| OUT | |
| H1 | Bool |
| IN_OUT | |
| Q1 | Bool |
| Q2 | Bool |
| Q3 | Bool |
| H2 | Bool |

6. Realisierung:

OB1:



Programmierung und Aufruf des Bausteins siehe bei-liegende CD oder

<http://www.automatisieren-mit-sps.de>

STEP7-Lösungen → 4_2_01

Lösung Lernaufgabe 4.2

Aufg. S. 46

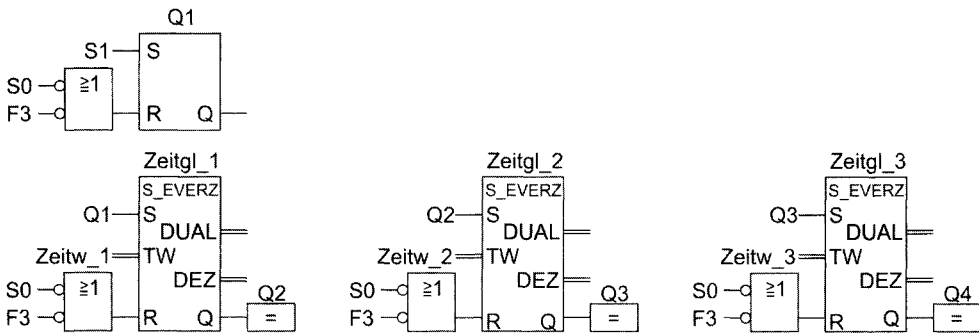
1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | | Adresse |
|----------------------|--------|----------|--------------------|--------|---------|
| Taster AUS | S0 | BOOL | Betätigt | S0 = 0 | E 0.0 |
| Taster EIN | S1 | BOOL | Betätigt | S1 = 1 | E 0.1 |
| Thermische Auslösung | F3 | BOOL | Betätigt | F3 = 0 | E 0.2 |
| Ausgangsvariable | | | | | |
| Netzschütz | Q1 | BOOL | Angezogen | Q1 = 1 | A 4.1 |
| Schütz 2 | Q2 | BOOL | Angezogen | Q2 = 1 | A 4.2 |
| Schütz 3 | Q3 | BOOL | Angezogen | Q3 = 1 | A 4.3 |
| Schütz 4 | Q4 | BOOL | Angezogen | Q4 = 1 | A 4.4 |

2. Erweiterte RS-Tabelle für die Lösung mit drei Zeitgliedern:

| Zu betätigende Speicher- und Zeitglieder | Bedingungen für das Setzen bzw. Starten | Bedingungen für das Rücksetzen |
|--|---|------------------------------------|
| Netzschütz (Q1) | S1 | $\overline{S0} \vee \overline{F3}$ |
| SE-Zeitglied 1 (Zeitgl_1), Q2 | Q1 | $\overline{S0} \vee \overline{F3}$ |
| SE-Zeitglied 2 (Zeitgl_2), Q3 | Q2 | $\overline{S0} \vee \overline{F3}$ |
| SE-Zeitglied 3 (Zeitgl_3), Q4 | Q3 | $\overline{S0} \vee \overline{F3}$ |

3. Funktionsplan für die Lösung mit drei Zeitgliedern:



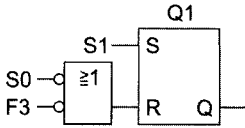
4. Erweiterte RS-Tabelle für die Lösung mit einem Zeitglied:

| Zu betätigende Speicher- und Zeitglieder | Bedingungen für das Setzen bzw. Starten | Bedingungen für das Rücksetzen |
|--|---|------------------------------------|
| Netzschütz (Q1) | S1 | $\overline{S0} \vee \overline{F3}$ |
| Schütz 4 (Q4) | Zeitgl & Q1 & Q2 & Q3 | $\overline{S0} \vee \overline{F3}$ |
| Schütz 3 (Q3) | Zeitgl & Q1 & Q2 | $\overline{S0} \vee \overline{F3}$ |
| Schütz 2 (Q2) | Zeitgl & Q1 | $\overline{S0} \vee \overline{F3}$ |
| SE-Zeitglied Zeitgl | $\overline{\text{Zeitgl}} \& Q1 \& \overline{Q4}$ | |

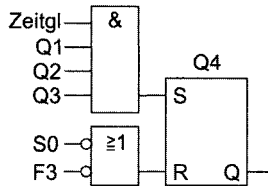
Hinweis: Die Speicherglieder müssen in der Reihenfolge Q1, Q4, Q3 und Q2 programmiert werden, damit das Steuerungsprogramm die Funktion richtig ausführt.

5. Funktionsplan für Lösung mit einem Zeitglied:

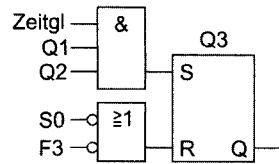
Schütz Q1



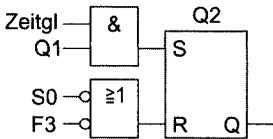
Schütz Q4



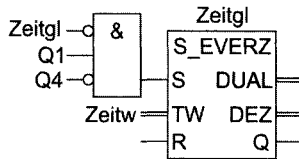
Schütz Q3



Schütz Q2



SE-Zeitglied



6. Bausteintyp und Deklarationstabelle:

Es genügt der Bausteintyp FC, da keine lokalen statischen Variablen erforderlich sind. Die Signalwerte der Schütze Q1 bis Q4 werden in den SPS-Ausgängen gespeichert.

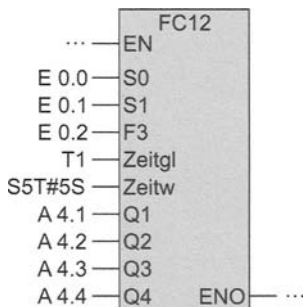
| Name | Datentyp |
|------|----------|
| IN | |
| S0 | Bool |
| S1 | Bool |
| F3 | Bool |

| Name | Datentyp |
|--------|----------|
| Zeitgl | Timer |
| Zeitw | S5Time |
| IN_OUT | |
| Q1 | Bool |

| Name | Datentyp |
|------|----------|
| Q2 | Bool |
| Q3 | Bool |
| Q4 | Bool |

7. Realisierung:

OB1:



Programmierung und Aufruf des Bausteins siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de>
STEP7-Lösungen → 4_2_02

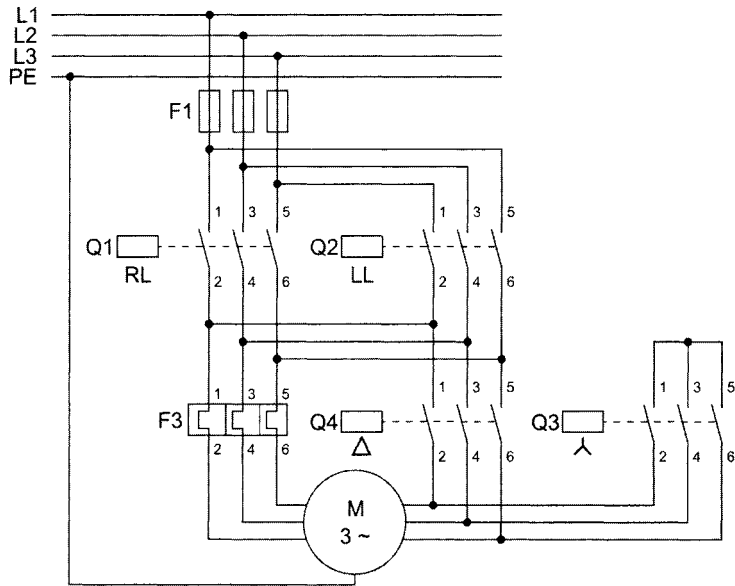
Lösung Lernaufgabe 4.3 Aufg. S. 47

1. Zuordnungstabelle der Eingänge und Ausgänge :

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|-----------------------|--------|----------|--------------------|---------|
| Taster AUS | S0 | BOOL | Betätigt S0 = 0 | E 0.0 |
| Taster EIN Rechtslauf | S1 | BOOL | Betätigt S1 = 1 | E 0.1 |
| Taster EIN Linkslauf | S2 | BOOL | Betätigt S2 = 1 | E 0.2 |

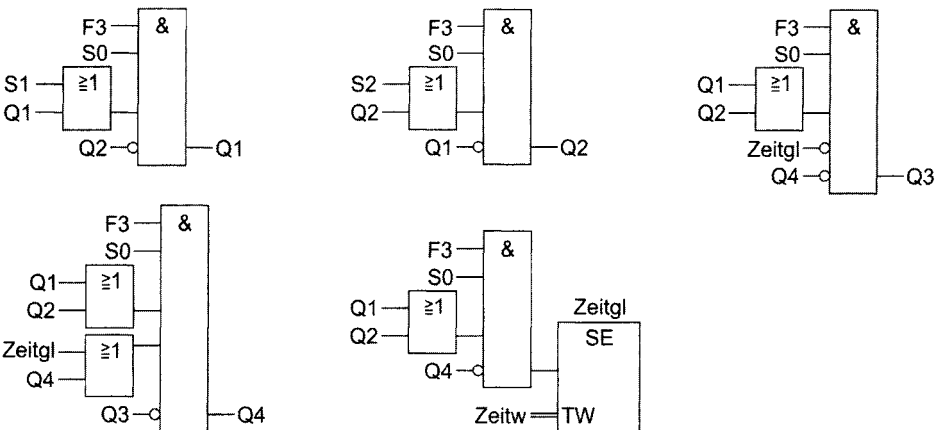
| Überstromschutz | F3 | BOOL | Motor überlastet | F3 = 0 | E 0.3 |
|------------------|----|------|------------------|--------|-------|
| Ausgangsvariable | | | | | |
| Netzschütz RL | Q1 | BOOL | Angezogen | Q1 = 1 | A 4.1 |
| Netzschütz LL | Q2 | BOOL | Angezogen | Q2 = 1 | A 4.2 |
| Sternschütz | Q3 | BOOL | Angezogen | Q3 = 1 | A 4.3 |
| Dreieckschütz | Q4 | BOOL | Angezogen | Q4 = 1 | A 4.4 |

2. Hauptstromkreis:



3. Funktionsplan:

Der Stromlaufplan enthält ein anzugsverzögertes Schütz K1. Dieses muss im Funktionsplan durch ein Zeitglied mit Einschaltverzögerung ersetzt werden. Die Tasterverriegelung von S1 und S2 in der Schützschaltung soll verhindern, dass bei gleichzeitiger Betätigung beider Taster ein Kurzschluss im Hauptstromkreis entsteht, obwohl die Schützverriegelung mit Q1 und Q2 vorhanden ist, die aber nicht schnell genug wirkt. Bei einer SPS-Lösung kann die Tasterverriegelung entfallen, da die SPS die Programmteile für Q1 und Q2 nicht gleichzeitig, sondern nacheinander abarbeitet.



4. Bausteintyp und Deklarationstabelle:

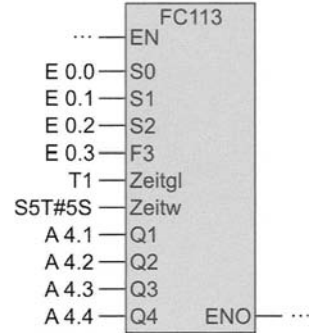
Es genügt der Bausteintyp FC, da keine lokalen statischen Variablen erforderlich sind. Die Signalwerte der Schütze Q1 bis Q4 werden in den SPS-Ausgängen gespeichert.

Deklarationstabelle FC 113:

| Name | Datentyp |
|--------|----------|
| IN | |
| S0 | Bool |
| S1 | Bool |
| S2 | Bool |
| F3 | Bool |
| Zeitgl | Timer |
| Zeitgw | S5Time |

| Name | Datentyp |
|--------|----------|
| IN_OUT | |
| Q1 | Bool |
| Q2 | Bool |
| Q3 | Bool |
| Q4 | Bool |

5. Realisierung 1: OB1:



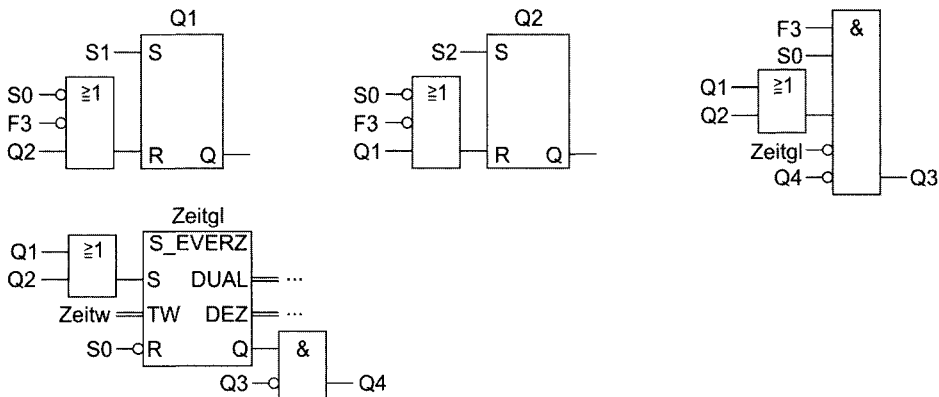
Programm s. beiliegende CD od. <http://automatisieren-mit-sps.de> STEP7-Lösungen → 4_2_03a

6. Erweiterte RS-Tabelle:

| Zu betätigende Speicher- und Zeitglieder | Bedingungen für das Setzen bzw. Starten | Bedingungen für das Rücksetzen |
|--|---|---|
| Netzschütz RL (Q1) | S1 | $\overline{S0} \vee \overline{F3} \vee Q2$ |
| Netzschütz LL (Q2) | S2 | $\overline{S0} \vee \overline{F3} \vee Q1$ |
| Sternschütz (Q3) | $Q1 \vee Q2$ | $\overline{S0} \vee \overline{F3} \vee \text{Zeitgl} \vee (\overline{Q1} \& \overline{Q2})$ |
| SE-Zeitglied Dreieck (Zeitgl) | $Q1 \vee Q2$ | $\overline{S0}$ |
| Dreieckschütz (Q4) | $\text{Zeitgl} \& \overline{Q3}$ | $\overline{\text{Zeitgl}} \vee Q3$ |

Hinweis: Da bei der Ansteuerung von Q3 und Q4 die Setzbedingung negiert auch in der Rücksetzbedingung auftritt, wird bei der Umsetzung keine Speicherfunktion verwendet.

7. Funktionsplan



8. Bausteintyp:

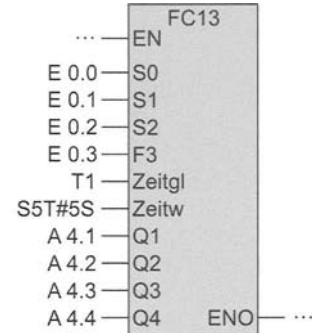
Da keine lokalen statischen Variablen benötigt werden, genügt der Bausteintyp FC. Die Signalwerte der Schütze Q1 bis Q3 werden in den SPS-Ausgängen gespeichert.

Deklarationstabelle FC 13:

| Name | Datentyp |
|--------|----------|
| IN | |
| S0 | Bool |
| S1 | Bool |
| S2 | Bool |
| F3 | Bool |
| Zeitgl | Timer |
| Zeitgw | S5Time |

| Name | Datentyp |
|--------|----------|
| IN_OUT | |
| Q1 | Bool |
| Q2 | Bool |
| Q3 | Bool |
| Q4 | Bool |

9. Realisierung 2: OB1:



Programm s. beiliegende CD od. <http://automatisieren-mit-sps.de> STEP7-Lösungen → 4_2_03b

| | |
|-------------------------------|-------------|
| Lösung Lernaufgabe 4.4 | Aufg. S. 48 |
|-------------------------------|-------------|

1. Zuordnungstabelle der Eingänge und Ausgänge :

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|----------------------------|--------|----------|--------------------|---------|
| Bodenkontakt | S1 | BOOL | Betätigt S1 = 0 | E 0.1 |
| Strahlungsquelle | S2 | BOOL | Dose voll S2 = 0 | E 0.2 |
| Initiator vord. Endlage 1A | IS1 | BOOL | Betätigt IS1 = 1 | E 0.3 |
| Ausgangsvariable | | | | |
| Magnetspule | IY1 | BOOL | Angezogen Q1 = 1 | A 4.1 |

2. Erweiterte RS-Tabelle:

Zur Lösung der Steuerungsaufgabe müssen folgende lokale Hilfsvariablen eingeführt werden:

DSP1: Fehlerspeicher für die 1. Dose, DSP2: Fehlerspeicher für die 2. Dose,

DSP3: Fehlerspeicher für die 3. Dose, DSP4: Fehlerspeicher für die 4. Dose,

Für die Flankenwertung Bodenkontakt betätigt und Dose nicht voll werden die beiden Operanden IO: Impulsoperand (Füllungsfehler) und FO: Flankenoperand eingeführt.

| Zu betätigende Speicher- und Zeitglieder | Bedingungen für das Setzen bzw. Starten | Bedingungen für das Rücksetzen |
|--|---|--------------------------------|
| Fehlerspeicher 4. Dose (DSP4) | IO & DSP1 & DSP2 & DSP3 | IS1 & Zeitgl_4 |
| Fehlerspeicher 3. Dose (DSP3) | IO & DSP1 & DSP2 | IS1 & Zeitgl_3 |
| Fehlerspeicher 2. Dose (DSP2) | IO & DSP1 | IS1 & Zeitgl_2 |
| Fehlerspeicher 1. Dose (DSP1) | IO | IS1 & Zeitgl_1 |
| SE-Zeitglied 4. Dose (Zeitgl_4) | DSP4 | |
| SE-Zeitglied 3. Dose (Zeitgl_3) | DSP3 | |

| | | |
|---------------------------------|---|-----|
| SE-Zeitglied 2. Dose (Zeitgl_2) | DSP2 | |
| SE-Zeitglied 1. Dose (Zeitgl_1) | DSP1 | |
| Magnetspule (1Y1) | Zeitgl_1 v Zeitgl_2 v Zeitgl_3 v Zeitgl_4 | IS1 |

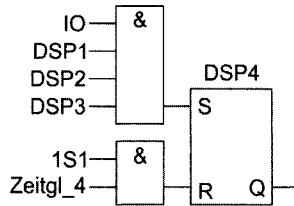
Hinweis: Die Speicherglieder müssen in der angegebenen Reihenfolge programmiert werden.

3. Funktionsplan

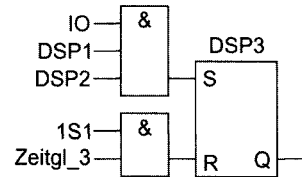
Impuls bei Füllungsfehler



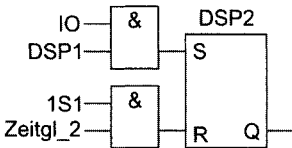
Fehlerspeicher 4. Dose



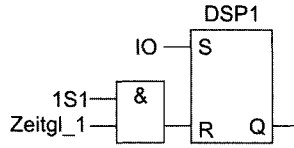
Fehlerspeicher 3. Dose



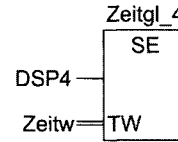
Fehlerspeicher 2. Dose



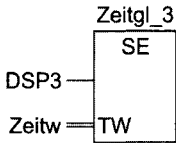
Fehlerspeicher 1. Dose



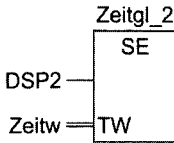
SE-Zeitglied 4. Dose



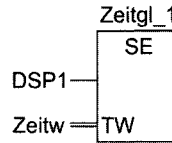
SE-Zeitglied 3. Dose



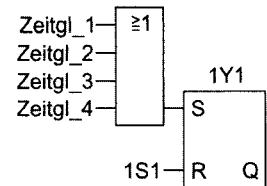
SE-Zeitglied 2. Dose



SE-Zeitglied 1. Dose



Magnetspule



4. Bausteintyp und Deklarationstabelle:

Da der Signalzustand der Dosenspeicher DSP gespeichert werden muss und ein Flankenoperand benötigt wird, sind statische Variablen erforderlich. Deshalb Bausteintyp FB.

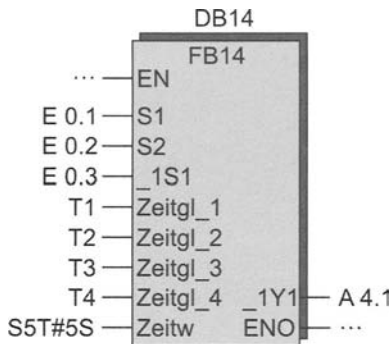
Deklarationstabelle FB 14:

| Name | Datentyp | Anfangswert |
|----------|----------|-------------|
| IN | | |
| S1 | Bool | FALSE |
| S2 | Bool | FALSE |
| _1S1 | Bool | FALSE |
| Zeitgl_1 | Timer | |
| Zeitgl_2 | Timer | |
| Zeitgl_3 | Timer | |
| Zeitgl_4 | Timer | |
| Zeitw | S5Time | S5T#2S |

| Name | Datentyp | Anfangswert |
|------|----------|-------------|
| OUT | | |
| _1Y1 | Bool | FALSE |
| STAT | | |
| DSP1 | Bool | FALSE |
| DSP2 | Bool | FALSE |
| DSP3 | Bool | FALSE |
| DSP4 | Bool | FALSE |
| FO | Bool | |
| TEMP | | |
| IO | Bool | |

5. Realisierung:

OBI:



Programmierung und Aufruf des Bausteins siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de> STEP7-Lösungen → 4_2_04

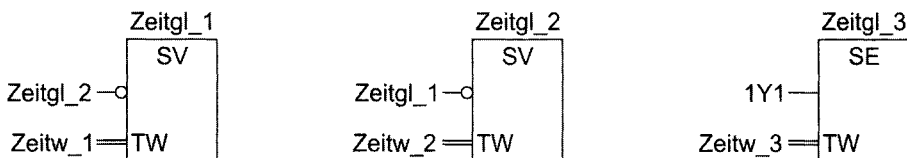
Hinweis:

Zur Überprüfung des Steuerungsprogramms mit PLCSIM ist es ratsam, eine Hilfsfunktion FC14 zu programmieren, die folgende Funktionen erfüllt:

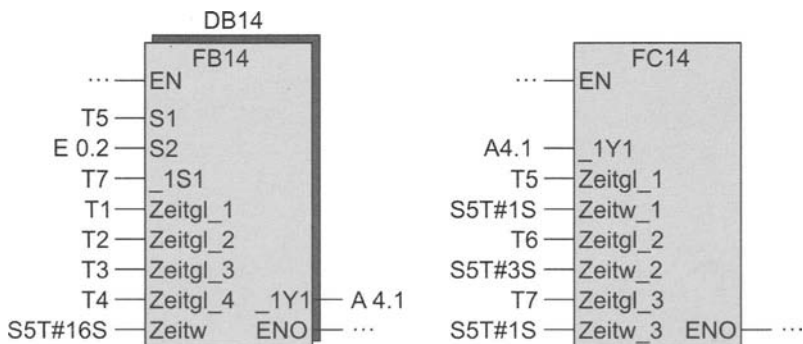
1. Ein Taktgenerator mit 1 Sekunde Puls und 3 Sekunden Pause, der mit dem Puls jeweils den Bodenkontakt S1 bedient.
2. Ein Zeitglied, das nach einer Sekunde nachdem das Magnetventil 1Y1 angesteuert wurde den Endschalter 1S1 bedient.

Legt man an den Eingang Zeitw_1 des Funktionsbausteins FB 14 den Zeitwert S5T#16S, so kann mit E 0.2, eine Fehlerfolge eingegeben und nach Ablauf von 16 Sekunden beobachtet werden, wie der Zylinder angesteuert wird.

Funktionsplan der Hilfsfunktion FC 14:



Aufruf der beiden Bausteine im OBI:



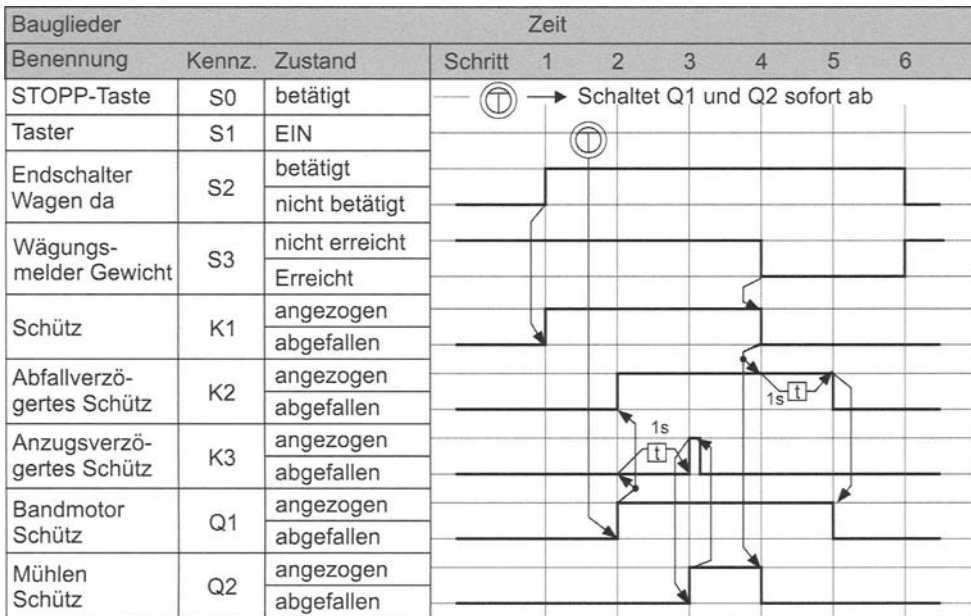
Lösung Lernaufgabe 4.5

Aufg. S. 49

1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse | |
|-----------------------|--------|----------|--------------------|---------|-------|
| Taster STOPP | S0 | BOOL | Betätigt | S0 = 0 | E 0.0 |
| Taster START | S1 | BOOL | Betätigt | S1 = 1 | E 0.1 |
| Endschalter Wagen | S2 | BOOL | Wagen da | S2 = 1 | E 0.2 |
| Wägungsmelder | S3 | BOOL | Gewicht erreicht | S3 = 0 | E 0.3 |
| Überstromschutz M1 | F3 | BOOL | Motor überlastet | F3 = 0 | E 0.4 |
| Überstromschutz M2 | F4 | BOOL | Motor überlastet | F4 = 0 | E 0.5 |
| Ausgangsvariable | | | | | |
| Leistungsschütz Band | Q1 | BOOL | Band läuft | Q1 = 1 | A 4.1 |
| Leistungsschütz Mühle | Q2 | BOOL | Mühle läuft | Q2 = 1 | A 4.2 |

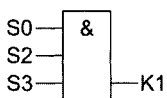
2. Funktionsdiagramm:



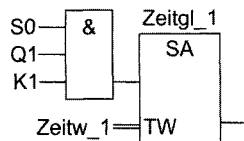
3. Funktionsplan:

Der Stromlaufplan enthält ein anzugsverzögertes Schütz K3 und ein abfallverzögertes Schütz K2. Beide Zeitschütze müssen durch entsprechende Zeitfunktionen ersetzt werden.

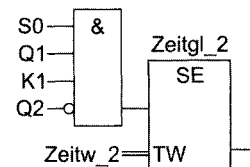
Schütz K1



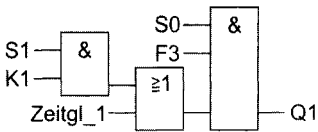
Zeitglied abfallverzögert (K2)



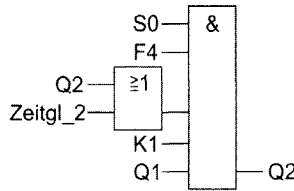
Zeitglied anzugsverzögert (K3)



Leistungsschütz Q1



Leistungsschütz Q2



4. Bausteintyp:

Da keine lokalen statischen Variablen benötigt werden, genügt der Bausteintyp FC.

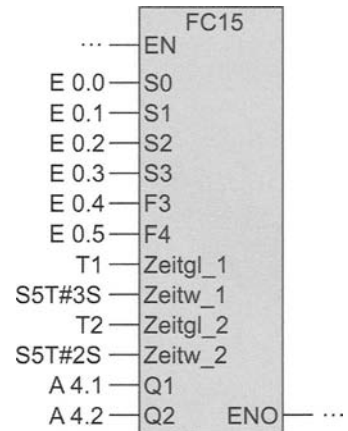
Deklarationstabelle FC 15:

| Name | Datentyp |
|-----------|----------|
| IN | |
| S0 | Bool |
| S1 | Bool |
| S2 | Bool |
| S3 | Bool |
| F3 | Bool |
| F4 | Bool |
| Zeitgl_1 | Timer |
| Zeitw_1 | S5Time |
| Zeitgl_2 | Timer |
| Zeitw_2 | S5Time |

| Name | Datentyp |
|---------------|----------|
| IN_OUT | |
| Q1 | Bool |
| Q2 | Bool |
| TEMP | |
| K1 | Bool |

5. Realisierung:

OB1:



Programm siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de>
STEP7-Lösungen → 4_2_05

Lösung Lernaufgabe 4.6 Aufg. S. 50

1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Sym- bol | Daten- typ | Logische Zuordnung | Adresse |
|---------------------------------|-------------|---------------|---------------------------|---------|
| Starttaster | S1 | BOOL | Betätigt S1 = 1 | E 0.1 |
| Initiator hintere Endl. Zyl. 1A | 1S1 | BOOL | Betätigt 1S1 = 1 | E 0.2 |
| Initiator vordere Endl. Zyl. 1A | 1S2 | BOOL | Betätigt 1S2 = 1 | E 0.3 |
| Initiator vordere Endl. Zyl. 2A | 2S1 | BOOL | Betätigt 2S1 = 1 | E 0.4 |
| Ausgangsvariable | | | | |
| Magnetspule Zylinder 1A vor | 1Y1 | BOOL | Zyl. 1A fährt aus 1Y1 = 1 | A 4.1 |
| Magnetspule Zylinder 1A zurück | 1Y2 | BOOL | Zyl. 1A fährt ein 1Y2 = 1 | A 4.2 |
| Magnetspule Zylinder 2A vor | 2Y1 | BOOL | Zyl. 2A fährt aus 2Y1 = 1 | A 4.3 |
| Magnetspule Zylinder 2A zurück | 2Y2 | BOOL | Zyl. 2A fährt ein 2Y2 = 1 | A 4.4 |

2. Funktionsdiagramm:

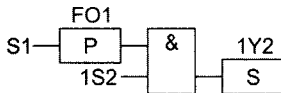
| Bauglieder | | | Zeit | | | | |
|-------------|--------|-------------|-----------|---|---|---|---|
| Benennung | Kennz. | Zustand | Schritt 1 | 2 | 3 | 4 | 5 |
| Taster | S1 | EIN | | | | | |
| DW-Zylinder | 1A | ausgefahren | — | — | — | — | — |
| | | eingefahren | — | — | — | — | — |
| DW-Zylinder | 2A | ausgefahren | — | — | — | — | — |
| | | eingefahren | — | — | — | — | — |

3. Schaltfolgetabelle:

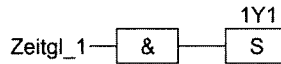
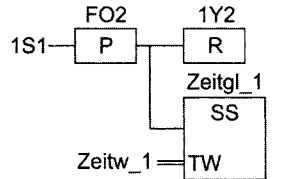
| Schritt | Bedingung | Setzen | Rücksetzen |
|---------|---------------------------|----------------------|----------------|
| 1 | S1 (0→1) & 1S2 | 1Y2 | |
| 2 | 1S1 (0→1) Zeitgl_1 = 1 | Zeitgl_1 (SS) 1Y1 | 1Y2 |
| 3 | 1S2 (0→1) Zeitgl_2 = 1 | Zeitgl_2 (SS) 2Y1 | 1Y1; Zeitgl_1 |
| 4 | 2S1 (0→1) & 1S2 | 2Y2 | 2Y1 ; Zeitgl_2 |
| 5 | 2S1 (1→0) & 1S2 | | 2Y2 |

4. Funktionsplan:

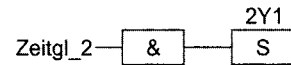
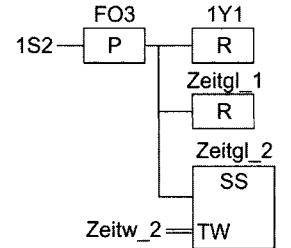
Schritt 1



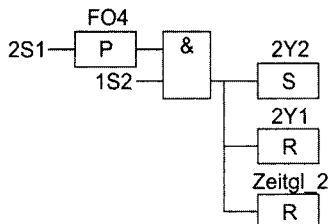
Schritt 2



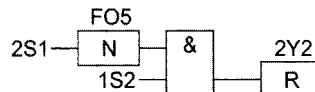
Schritt 3



Schritt 4



Schritt 5



5. Bausteintyp und Deklarationstabelle:

Bei der Flankenbewertung werden für die Flankenoperanden lokale statische Variablen benötigt, deshalb ist der Bausteintyp Funktionsbaustein FB erforderlich.

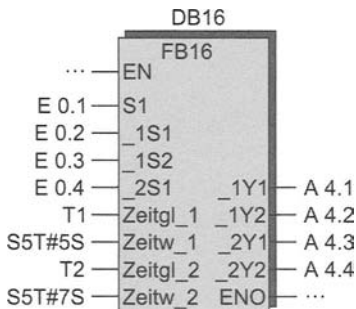
Deklarationstabelle FB 16:

| Name | Datentyp | Anfangswert |
|-----------|----------|-------------|
| IN | | |
| S1 | Bool | FALSE |
| _1S1 | Bool | FALSE |
| _1S2 | Bool | FALSE |
| _2S1 | Bool | FALSE |
| Zeitgl_1 | Timer | |
| Zeitw_1 | S5Time | S5T#5s |
| Zeitgl_2 | Timer | |
| Zeitw_2 | S5Time | S5T#7s |

| Name | Datentyp | Anfangswert |
|-------------|----------|-------------|
| OUT | | |
| _1Y1 | Bool | FALSE |
| _1Y2 | Bool | FALSE |
| _2Y1 | Bool | FALSE |
| _2Y2 | Bool | FALSE |
| STAT | | |
| FO1 | Bool | FALSE |
| FO2 | Bool | FALSE |
| FO3 | Bool | FALSE |
| FO4 | Bool | FALSE |
| FO5 | Bool | FALSE |

6. Realisierung:

OB1:



Programmierung und Aufruf des Bausteins siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de> STEP7-Lösungen → 4_2_06

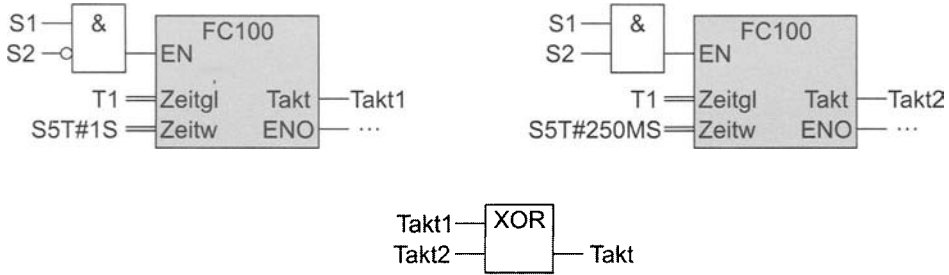
Lösung Lernaufgabe 4.7

Aufg. S. 51

1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|-------------------------|--------|----------|----------------------|--------------|
| Schalter EIN/AUS | S1 | BOOL | Betätigt, EIN | S1 = 1 E 0.1 |
| Wahlschalter Frequenz | S2 | BOOL | Betätigt, f = 0,5 Hz | S2 = 1 E 0.2 |
| Ausgangsvariable | | | | |
| Warnlampe 1 | H1 | BOOL | Leuchtet | H1 = 1 A 4.1 |
| Warnlampe 2 | H2 | BOOL | Leuchtet | H2 = 1 A 4.2 |
| Warnlampe 3 | H3 | BOOL | Leuchtet | H3 = 1 A 4.3 |
| Warnlampe 4 | H4 | BOOL | Leuchtet | H4 = 1 A 4.4 |
| Warnlampe 5 | H5 | BOOL | Leuchtet | H5 = 1 A 4.5 |

2. Zweimaliger Aufruf der Taktfunktion FC 100 zur Erzeugung von zwei unterschiedlichen Frequenzen:



3. Schaltfolgetabelle

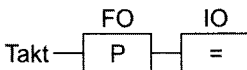
Aus dem Takt wird mit einer Flankenbewertung ein Impulsmerker (IO = Takt (0 → 1)) gebildet.

| Schritt | Bedingung | Setzen | Rücksetzen |
|---------|---|--------|------------|
| 1 | $IO \& \overline{H1} \& \overline{H2} \& \overline{H3} \& \overline{H4} \& \overline{H5}$ | H1 | IO |
| 2 | $IO \& H1 \& \overline{H2} \& \overline{H3} \& \overline{H4} \& \overline{H5}$ | H2 | H1, IO |
| 3 | $IO \& \overline{H1} \& H2 \& \overline{H3} \& \overline{H4} \& \overline{H5}$ | H3 | H2, IO |
| 4 | $IO \& \overline{H1} \& \overline{H2} \& H3 \& \overline{H4} \& \overline{H5}$ | H4 | H3, IO |
| 5 | $IO \& \overline{H1} \& \overline{H2} \& \overline{H3} \& H4 \& \overline{H5}$ | H5 | H4, IO |
| 6 | $IO \& \overline{H1} \& \overline{H2} \& \overline{H3} \& \overline{H4} \& H5$ | | H5, IO |

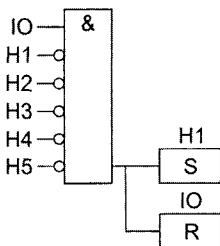
Hinweis: Damit bei ausgeschaltetem Verkehrslicht auch alle Warnlampen ausgeschaltet sind, werden im letzten Netzwerk des Bausteins alle Lampen H1 bis H5 zurückgesetzt, wenn S1 = „0“ – Signal führt.

4. Funktionsplan:

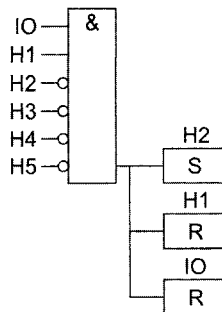
Flankenbewertung des Takteingangs:



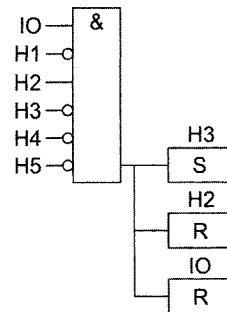
Schritt 1



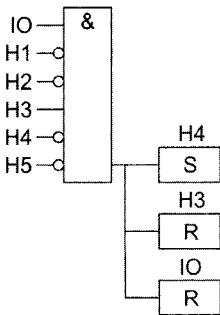
Schritt 2



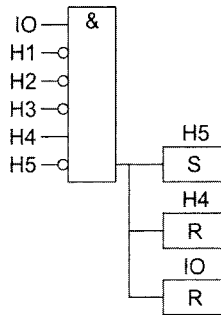
Schritt 3



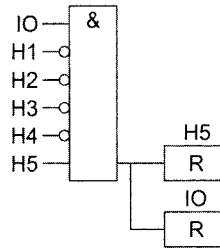
Schritt 4



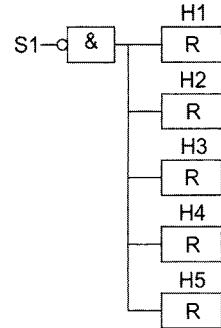
Schritt 5



Schritt 6



Lauflicht AUS



5. Bausteintyp und Deklarationstabelle:

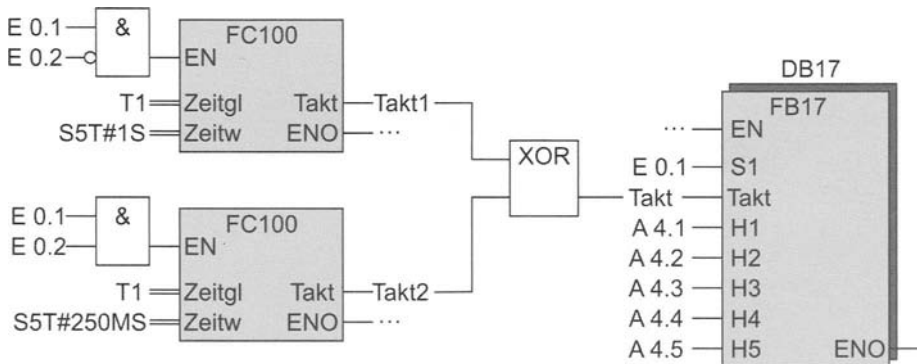
Bei der Flankenbewertung muss der Flankenoperand FO als lokale, statische Variable deklariert werden. Deshalb ist der Bausteintyp Funktionsbaustein FB erforderlich ist.

Deklarationstabelle FB 17:

| Name | Datentyp | Anfangswert |
|--------|----------|-------------|
| IN | | |
| S1 | Bool | FALSE |
| Takt | Bool | FALSE |
| IN OUT | | |
| H1 | Bool | FALSE |
| H2 | Bool | FALSE |
| H3 | Bool | FALSE |
| H4 | Bool | FALSE |
| H5 | Bool | FALSE |

| Name | Datentyp | Anfangswert |
|------|----------|-------------|
| STAT | | |
| FO | Bool | FALSE |
| TEMP | | |
| IO | Bool | |

6. Freigrafischer Funktionsplan:



7. Realisierung:

Der freigrafische Funktionsplan zeigt die Bausteinaufrufe im OB1. Die drei Übergabevariablen Takt1, Takt2 und Takt müssen im OB1 als temporäre Variablen deklariert werden.

Vollständiges Programm siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de> STEP7-Lösungen → 4_2_07

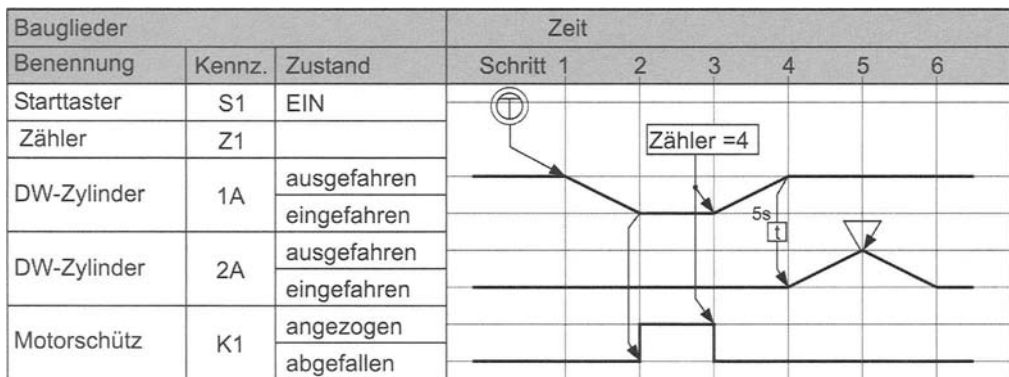
Lösung Lernaufgabe 4.8

Aufg. S. 52

1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|---------------------------------|--------|----------|---------------------------|---------|
| Starttaster | S1 | BOOL | Betätigt S1 = 1 | E 0.1 |
| Initiator hintere Endl. Zyl. 1A | 1S1 | BOOL | Betätigt 1S1 = 1 | E 0.2 |
| Initiator vordere Endl. Zyl. 1A | 1S2 | BOOL | Betätigt 1S2 = 1 | E 0.3 |
| Initiator vordere Endl. Zyl. 2A | 2S1 | BOOL | Betätigt 2S1 = 1 | E 0.4 |
| Lichtschranke | LI | BOOL | Betätigt LI = 1 | E 0.5 |
| Ausgangsvariable | | | | |
| Motorschütz Q1 | Q1 | BOOL | Motor ein Q1 = 1 | A 4.0 |
| Magnetspule Zylinder 1A vor | 1Y1 | BOOL | Zyl. 1A fährt aus 1Y1 = 1 | A 4.1 |
| Magnetspule Zylinder 1A zur. | 1Y2 | BOOL | Zyl. 1A fährt ein 1Y2 = 1 | A 4.2 |
| Magnetspule Zylinder 2A vor | 2Y1 | BOOL | Zyl. 2A fährt aus 2Y1 = 1 | A 4.3 |
| Magnetspule Zylinder 2A zur. | 2Y2 | BOOL | Zyl. 2A fährt ein 2Y2 = 1 | A 4.4 |

2. Funktionsdiagramm:

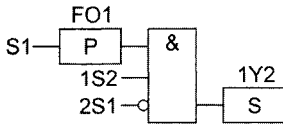


3. Schaltfolgetabelle:

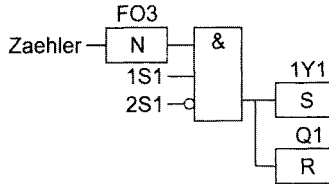
| Schritt | Bedingung | Setzen | Rücksetzen |
|---------|---|-----------------------------|-------------|
| 1 | $S1(0 \rightarrow 1) \& 1S2 \& \overline{2S1}$ | 1Y2 | |
| 2 | $1S1(0 \rightarrow 1) \& \overline{2S1}$ $LI \& 1S1 \& \overline{2S1}$ | Q1, Zaehler=4 ZR Zaehler | 1Y2 |
| 3 | Zaehler(1 \rightarrow 0) & $1S1 \& \overline{2S1}$ | 1Y1 | Q1 |
| 4 | $1S2(0 \rightarrow 1) \& \overline{2S1}$ Zeitgl = 1 | Zeitgl (SS) 2Y1 | 1Y1 |
| 5 | $2S1(0 \rightarrow 1) \& 1S2$ | 2Y2 | 2Y1, Zeitgl |
| 6 | $2S1(1 \rightarrow 0) \& 1S2$ | | 2Y2 |

4. Funktionsplan:

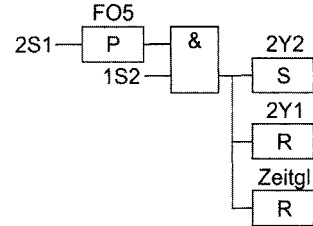
Schritt 1



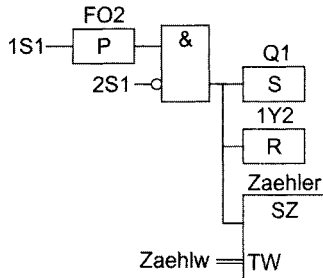
Schritt 3



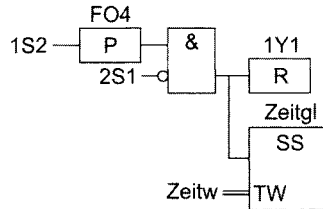
Schritt 5



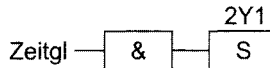
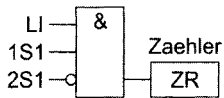
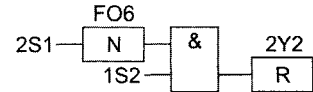
Schritt 2



Schritt 4



Schritt 6



5. Bausteintyp und Deklarationstabelle:

Bei der Flankenbewertung werden für die Flankenoperanden lokale statische Variablen benötigt, so dass der Bausteintyp Funktionsbaustein FB erforderlich ist.

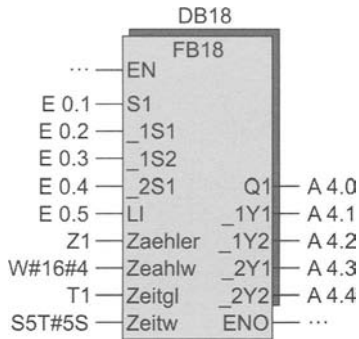
Deklarationstabelle FB 18:

| Name | Datentyp | Anfangswert |
|-----------|----------|-------------|
| IN | | |
| S1 | Bool | FALSE |
| _1S1 | Bool | FALSE |
| _1S2 | Bool | FALSE |
| _2S1 | Bool | FALSE |
| LI | Bool | FALSE |
| Zaehler | Counter | |
| Zaehlw | Word | W#16#4 |
| Zeitgl | Timer | |
| Zeitw | S5Time | S5T#5s |

| Name | Datentyp | Anfangswert |
|-------------|----------|-------------|
| OUT | | |
| Q1 | Bool | FALSE |
| _1Y1 | Bool | FALSE |
| _1Y2 | Bool | FALSE |
| _2Y1 | Bool | FALSE |
| _2Y2 | Bool | FALSE |
| STAT | | |
| FO1 | Bool | FALSE |
| FO2 | Bool | FALSE |
| FO3 | Bool | FALSE |
| FO4 | Bool | FALSE |
| FO5 | Bool | FALSE |
| FO6 | Bool | FALSE |

6. Realisierung:

OB1:



Programmierung und Aufruf des Bausteins
siehe beiliegende CD oder

<http://www.automatisieren-mit-sps.de>

STEP7-Lösungen → 4_2_08

Lösung Lernaufgabe 4.9

Aufg. S. 52

1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse | |
|---------------------|--------|----------|--------------------|---------|-------|
| Anlagenschalter | S1 | BOOL | EIN | S1 = 1 | E 0.1 |
| Schlüsselschalter | S2 | BOOL | EIN | S2 = 1 | E 0.2 |
| Initiator 1 | I1 | BOOL | Aktiviert | I1 = 1 | E 0.3 |
| Lichtschranke 1 | LI1 | BOOL | Unterbrochen | LI1 = 1 | E 0.4 |
| Initiator 2 | I2 | BOOL | Aktiviert | I2 = 1 | E 0.5 |
| Lichtschranke 2 | LI2 | BOOL | Unterbrochen | LI2 = 1 | E 0.6 |
| Ausgangsvariable | | | | | |
| Motor 1 RL Einfahrt | Q1 | BOOL | Schranke öffnet | Q1 = 1 | A 4.1 |
| Motor 1 LL Einfahrt | Q2 | BOOL | Schranke schließt | Q2 = 1 | A 4.2 |
| Motor 2 RL Ausfahrt | Q3 | BOOL | Schranke öffnet | Q3 = 1 | A 4.3 |
| Motor 1 LL Ausfahrt | Q4 | BOOL | Schranke schließt | Q4 = 1 | A 4.4 |
| Ampel Rot | H1 | BOOL | Leuchtet | H1 = 1 | A 4.5 |
| Ampel Grün | H2 | BOOL | Leuchtet | H2 = 1 | A 4.6 |

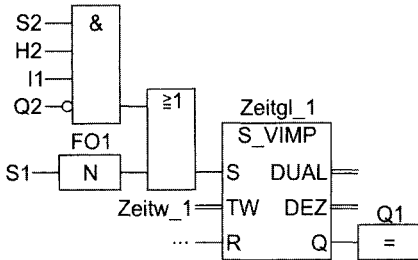
2. Erweiterte RS-Tabelle:

| Zu betätigende Speicher- und Zeitglieder | Bedingungen für das Setzen bzw. Starten | Rücksetz-Bedingungen |
|--|---|----------------------|
| SV-Zeitglied Einfahrt 1 (Zeitgl_1), (Q1) | $(S2 \& H2 \& I1 \& \overline{Q2}) \vee S1(1 \rightarrow 0)$ | |
| SV-Zeitglied Einfahrt 2 (Zeitgl_2), (Q2) | $(\overline{I1} \& \overline{LI1} \& \overline{Q1}) \neq S1(0 \rightarrow 1)$ | $\overline{S1}$ |
| SV-Zeitglied Ausfahrt 1 (Zeitgl_3), (Q3) | $(I2 \& \overline{Q4}) \vee S1(1 \rightarrow 0)$ | |
| SV-Zeitglied Ausfahrt 2 (Zeitgl_4), (Q4) | $(\overline{I2} \& \overline{LI2} \& \overline{Q3}) \neq S1(0 \rightarrow 1)$ | $\overline{S1}$ |

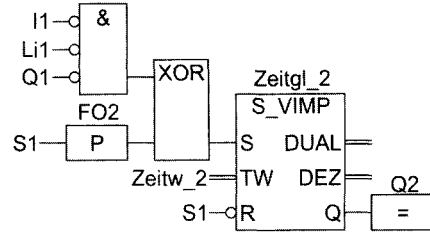
Der erforderliche Zähler wird im eingeschalteten Zustand bei Unterbrechung der Lichtschranke LI1 rückwärts und bei Unterbrechung der Lichtschranke LI2 vorwärts gezählt. Beim Einschalten der Anlage mit S1 wird der Zähler auf die Anzahl(12) der Parkplätze gesetzt. Im ausgeschalteten Zustand wird der Zähler zurückgesetzt. Mit dem Ausgang des Zählers wird die Ampel angesteuert.

3. Funktionsplan:

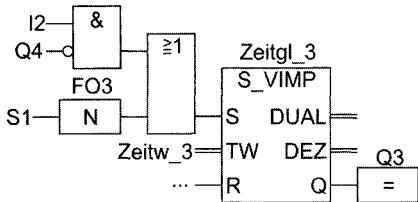
SV-Zeitgl_1: Einfahrt 1 und Q1



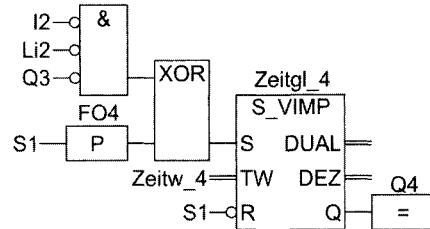
SV-Zeitgl_2: Einfahrt 2 und Q2



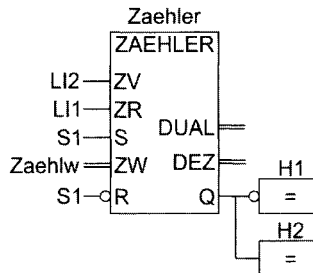
SV-Zeitgl_3: Ausfahrt 1 und Q3



SV-Zeitgl_4: Ausfahrt 2 und Q4



Ansteuerung des Zählers und der Ampel



4. Bausteintyp und Deklarationstabelle:

Für die Flankenauswertung der S1-Signale beim Ein- und Ausschalten der Anlage werden lokale statische Variablen benötigt, so dass der Bausteintyp FB erforderlich ist.

Deklarationstabelle FB 19:

| Name | Datentyp | Anfangswert |
|---------|----------|-------------|
| IN | | |
| S1 | Bool | FALSE |
| S2 | Bool | FALSE |
| I1 | Bool | FALSE |
| LI1 | Bool | FALSE |
| I2 | Bool | FALSE |
| LI2 | Bool | FALSE |
| Zaehler | Counter | |
| Zaehlw | Word | W#16#0 |

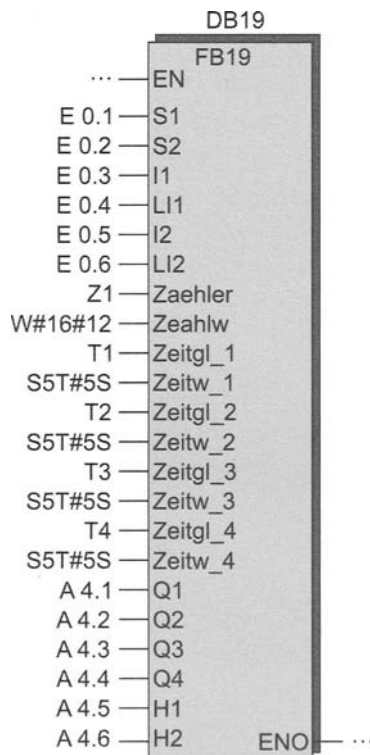
| Name | Datentyp | Anfangswert |
|----------|----------|-------------|
| IN | | |
| Zeitgl_1 | Timer | |
| Zeitw_1 | S5Time | S5T#0s |
| Zeitgl_2 | Timer | |
| Zeitw_2 | S5Time | S5T#0s |
| Zeitgl_3 | Timer | |
| Zeitw_3 | S5Time | S5T#0s |
| Zeitgl_4 | Timer | |
| Zeitw_4 | S5Time | S5T#0s |

Fortsetzung Deklarationstabelle FB 19:

| Name | Datentyp | Anfangswert |
|---------------|----------|-------------|
| IN_OUT | | |
| Q1 | Bool | FALSE |
| Q2 | Bool | FALSE |
| Q3 | Bool | FALSE |
| Q4 | Bool | FALSE |
| H1 | Bool | FALSE |
| H2 | Bool | FALSE |
| STAT | | |
| FO1 | Bool | FALSE |
| FO2 | Bool | FALSE |
| FO3 | Bool | FALSE |
| FO4 | Bool | FALSE |

5. Realisierung:

Aufruf des Bausteins im OB1:



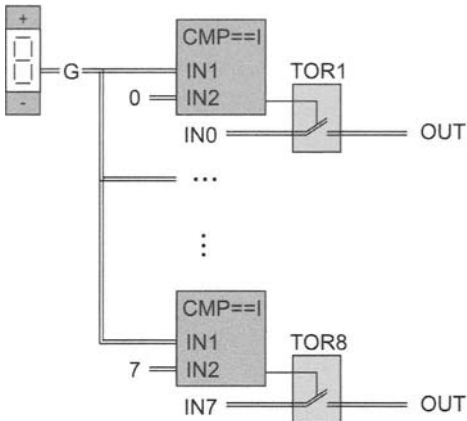
Programmierung und Aufruf des Bausteins siehe beiliegende CD oder
<http://www.automatisieren-mit-sps.de> STEP7-Lösungen → 4_2_09

10.5 Lösungen der Lernaufgaben Kapitel 5

Lösung Lernaufgabe 5.1

Aufg. S. 66

1. Freigrafischer Funktionsplan:



2. Deklarationstabelle FC 36:

| Name | Datentyp | Name | Datentyp |
|------|----------|------|----------|
| IN | | OUT | |
| G | Int | OUT | Real |
| IN0 | Real | | |
| IN1 | Real | | |
| IN2 | Real | | |
| IN3 | Real | | |
| IN4 | Real | | |
| IN5 | Real | | |
| IN6 | Real | | |
| IN7 | Real | | |

3. Anweisungsliste AWL:

```
Vergl. u. Tor 1
U(
  L #G
  L 0
  ==I
)
SPBNB_001
L #IN0
T #OUT
_001:NOP 0
```

```
Vergl. u. Tor 3
U(
  L #G
  L 2
  ==I
)
SPBNB_003
L #IN2
T #OUT
_003:NOP 0
```

```
Vergl. u. Tor 5
U(
  L #G
  L 4
  ==I
)
SPBNB_005
L #IN4
T #OUT
_005:NOP 0
```

```
Vergl. u. Tor 7
U(
  L #G
  L 6
  ==I
)
SPBNB_007
L #IN6
T #OUT
_007:NOP 0
```

```
Vergl. u. Tor 2
U(
  L #G
  L 1
  ==I
)
SPBNB_002
L #IN1
T #OUT
_002: NOP 0
```

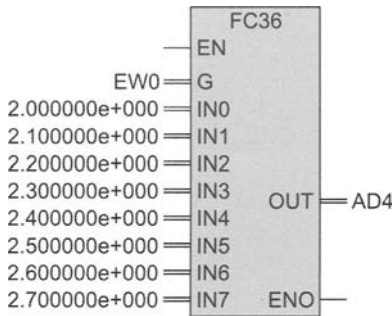
```
Vergl. u. Tor 4
U(
  L #G
  L 3
  ==I
)
SPBNB_004
L #IN3
T #OUT
_004: NOP 0
```

```
Vergl. u. Tor 6
U(
  L #G
  L 5
  ==I
)
SPBNB_006
L #IN5
T #OUT
_006: NOP 0
```

```
Vergl. u. Tor 8
U(
  L #G
  L 7
  ==I
)
SPBNB_008
L #IN7
T #OUT
_008: NOP 0
```

4. Realisierung:

OB1:



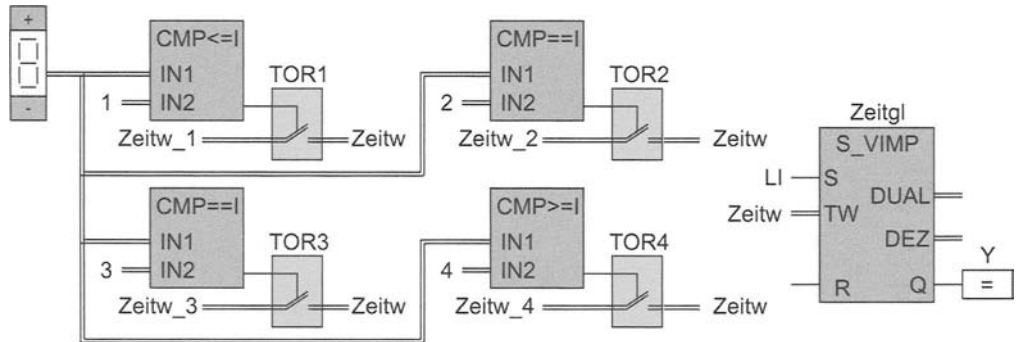
Programmierung und Aufruf des Bausteins siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de>
STEP7-Lösungen → 5_2_01

Lösung Lernaufgabe 5.2 Aufg. S. 66

1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|----------------------|--------|----------|--------------------|---------|
| Lichtschranke | LI | BOOL | Unterbrochen | E 0.0 |
| Zifferneinsteller | EB1 | BYTE | | EB 1 |
| Ausgangsvariable | | | | |
| Ventil der Klebedüse | Y | BOOL | Geöffnet | A 4.1 |

2. Freigrafischer Funktionsplan:



3. Deklarationstabelle FC 12:

| Name | Datentyp |
|---------|----------|
| IN | |
| Groesse | Int |
| LI | Bool |
| Zeitgl | Timer |
| Zeitw_1 | S5Time |
| Zeitw_2 | S5Time |
| Zeitw_3 | S5Time |
| Zeitw_4 | S5Time |

| Name | Datentyp |
|------|----------|
| OUT | |
| Y | Bool |

| Name | Datentyp |
|-------|----------|
| TEMP | |
| Zeitw | S5Time |

4. Anweisungsliste AWL:

```
//Zeit Groesse 1 //Zeit Groesse 2 //Zeit Groesse 3 //Zeit Groesse 3
  U(
    L #Groesse
    L 1
    <=I
  )
  SPBNB_001
  L #Zeitw_1
  T #Zeitw
_001:NOP 0

  U(
    L #Groesse
    L 2
    ==I
  )
  SPBNB_002
  L #Zeitw_2
  T #Zeitw
_002:NOP 0

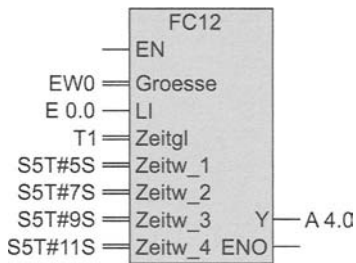
  U(
    L #Groesse
    L 3
    ==I
  )
  SPBNB_003
  L #Zeitw_3
  T #Zeitw
_003: NOP 0

  U(
    L #Groesse
    L 4
    >=I
  )
  SPBNB_004
  L #Zeitw_4
  T #Zeitw
_004: NOP 0

//Zeitglied
  U #LI
  L #Zeitw
  SV #Zeitgl
  U #Zeitgl
  = #Y
```

5. Realisierung:

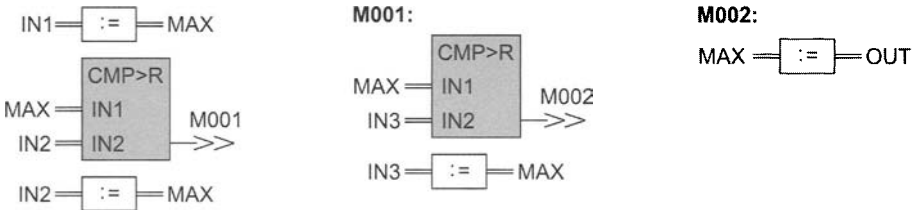
OBI:



Programmierung und Aufruf des Bausteins siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de>
STEP7-Lösungen → 5_2_02

Lösung Lernaufgabe 5.3 Aufg. S. 67

1. Freigrafischer Funktionsplan:



2. Deklarationstabelle FC 25:

| Name | Datentyp |
|------|----------|
| IN | |
| IN1 | Real |
| IN2 | Real |
| IN3 | Real |

| Name | Datentyp |
|------|----------|
| OUT | |
| OUT | Real |

| Name | Datentyp |
|------|----------|
| TEMP | |
| MAX | Real |

3. Anweisungsliste AWL:

```

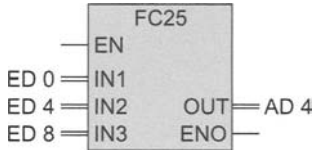
L #IN1
T #MAX
L #MAX
L #IN2
>R
SPB M001

L #IN2
T #MAX
M001: L #MAX
L #IN3
>R
SPB M002

L #IN3
T #MAX
M002: L #MAX
T #OUT
    
```

4. Realisierung:

OB1:



Programmierung und Aufruf des Bausteins siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de>
STEP7-Lösungen → 5_2_03

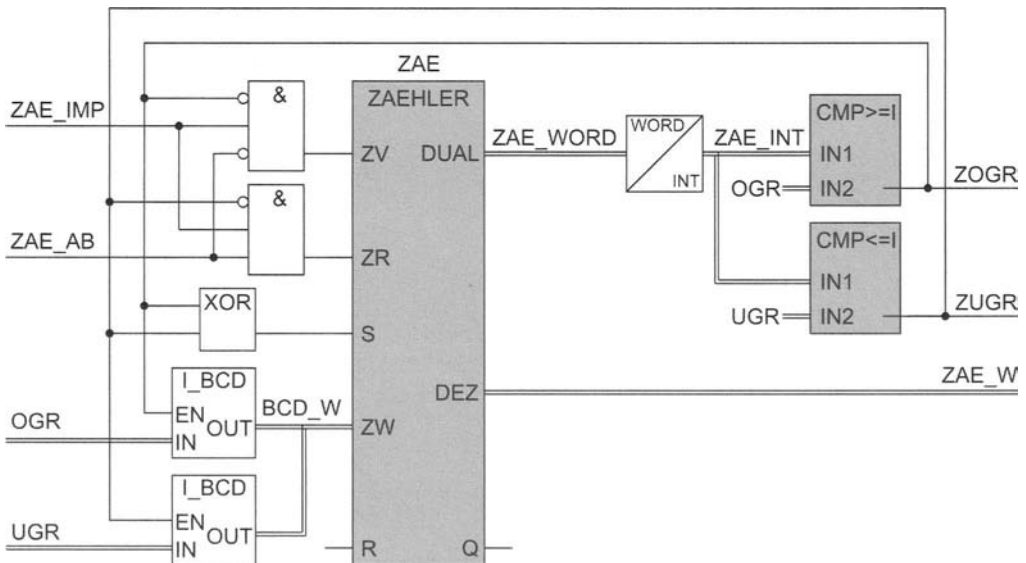
Lösung Lernaufgabe 5.4

Aufg. S. 67

1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|-------------------------|---------|----------|----------------------------|---------|
| Zählimpulse | ZAE_IMP | BOOL | Impulse f = 5Hz | E 0.0 |
| Zählrichtung | ZAE_AB | BOOL | Abwärts ZAE_AB = 1 | E 0.1 |
| Obergrenze | OGR | INT | Maximal: 999 | EW 2 |
| Untergrenze | UGR | INT | Minimal: 0 | EW 4 |
| Ausgangsvariable | | | | |
| Zählerstand | ZAE_W | BCD | 3 stellig | AW 6 |
| Zählerstand Obergrenze | ZOGR | BOOL | Obergr. erreicht ZOGR = 1 | A 4.0 |
| Zählerstand Untergrenze | ZUGR | BOOL | Untergr. erreicht ZOGR = 1 | A 4.1 |

2. Freigrafischer Funktionsplan:



3. Deklarationstabelle FC 83:

| Name | Datentyp |
|-----------|----------|
| IN | |
| ZAE_IMP | Bool |
| ZAE | Counter |
| ZAE_AB | Bool |
| OGR | Int |
| UGR | Int |

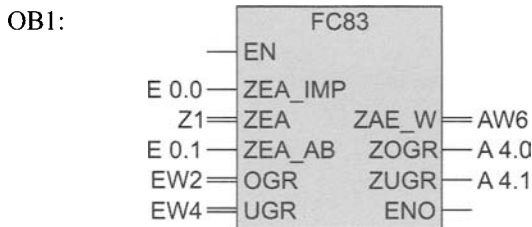
| Name | Datentyp |
|------------|----------|
| OUT | |
| ZAE_W | Word |
| ZOGR | Bool |
| ZUGR | Bool |

| Name | Datentyp |
|-------------|----------|
| TEMP | |
| ZAE_WORD | Word |
| ZAE_INT | Int |
| BCD_W | Word |

4. Anweisungsliste AWL:

```
//Zuweisung OGR //Zaehler //Ausg. Zählwert //Abfrage OGR
    U #ZOGR UN #ZOGR L #ZAE L #ZAE_INT
    SPBNB M001 U #ZAE_IMP T #ZAE_WORD L #OGR
    L #OGR UN #ZAE_AB LC #ZAE >=I
    ITB ZV #ZAE T #ZAE_W = #ZOGR
    T #BCD_W UN #ZUGR
M001: NOP 0 U #ZAE_IMP Formatwandelung Abfrage UGR
    U #ZAE_AB L #ZAE_WORD L #ZAE_INT
//Zuweisung UGR ZR #ZAE T #ZAE_INT L #UGR
    U #ZUGR U( <=I
    SPBNB M002 X #ZOGR = #ZUGR
    L #UGR X #ZUGR
    ITB )
    T #BCD_W L #BCD_W
M002: NOP 0 S #ZAE
```

5. Realisierung:



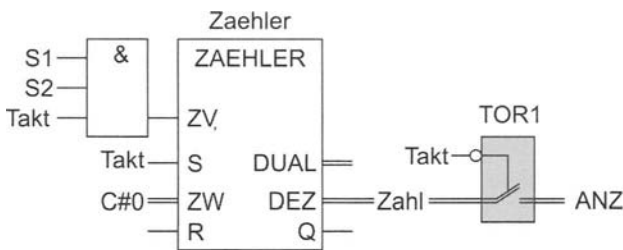
Programmierung und Aufruf des Bausteins siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de>
STEP7-Lösungen → 5_2_04

Lösung Lernaufgabe 5.5 Aufg. S. 68

1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|-------------------------|--------|----------|---------------------|--------------|
| Schalter Messung | S1 | BOOL | Eingeschaltet | S1 = 1 E 0.1 |
| Drehimpulsgeber | S2 | BOOL | Impuls vorhanden | S2 = 1 E 0.2 |
| Ausgangsvariable | | | | |
| Anzeige | AW | WORD | 3 stellige BCD-Zahl | AW 4 |

2. Freigrafischer Funktionsplan:



3. Deklarationstabelle FC 15:

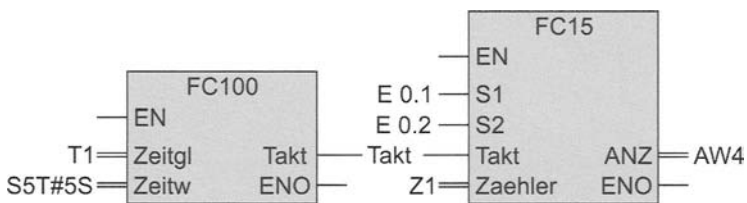
| Name | Datentyp |
|-------------|----------|
| IN | |
| S1 | Bool |
| S2 | Bool |
| Takt | Bool |
| Zaehler | Counter |
| OUT | |
| ANZ | Word |
| TEMP | |
| Zahl | Word |

4. Anweisungsliste AWL:

```

U #S1           U #Takt           UN #Takt
U #S2           L C#0             SPBNB _001
U #Takt         S #Zaehler        L #Zahl
ZV #Zaehler    LC #Zaehler       T #ANZ
               T #Zahl            _001: NOP 0
    
```

5. Aufruf der Bausteine im OB1:



6. Realisierung:

Programmierung und Aufruf der Bausteine siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de> STEP7-Lösungen → 5_2_05

Lösung Lernaufgabe 5.6

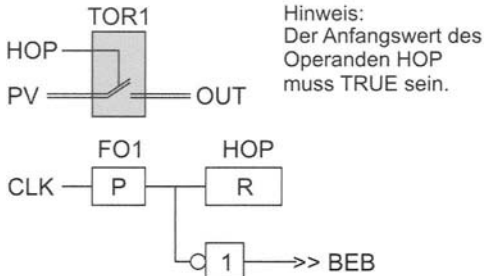
Aufg. S. 69

1. Zuordnungstabelle der Eingänge und Ausgänge für den Test des Bausteins:

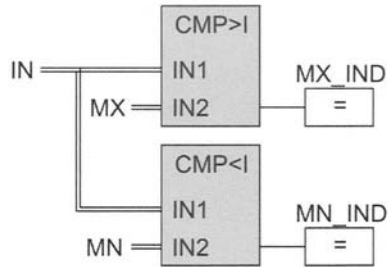
| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|-------------------------|--------|----------|--------------------------|---------|
| Übernahme | CLK | BOOL | Steigende Flanke CLK = 1 | E 0.0 |
| Eingangswert | IN | INT | 16 Bit Integer-Wert | EW 2 |
| Maximalwert | MX | INT | 16 Bit Integer-Wert | EW 4 |
| Minimalwert | MN | INT | 16 Bit Integer-Wert | EW 6 |
| Startwert | PV | INT | 16 Bit Integer-Wert | EW 8 |
| Ausgangsvariable | | | | |
| Überschreitung max | MX_IND | BOOL | Leuchtet MX_IND = 1 | A 4.1 |
| Unterschreitung min | MN_IND | BOOL | Leuchtet MN_IND = 1 | A 4.2 |
| Ausgangswert | OUT | INT | 16 Bit Integer-Wert | AW 6 |

2. Freigrafischer Funktionsplan:

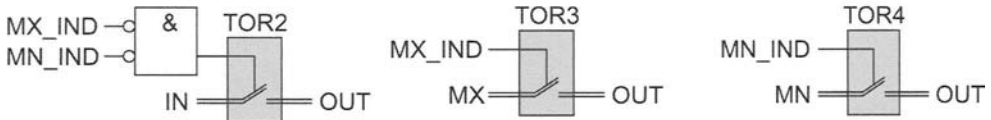
Übergabe des Startwertes PV und Auswertung des Eingangs CLK:



Abfrage des Eingabewertes:



Zuweisung an den Ausgangswert OUT:



3. Deklarationstabelle FB 16:

| Name | Datentyp | Anfangswert |
|------|----------|-------------|
| IN | | |
| CLK | Bool | FALSE |
| IN | Int | 0 |
| MX | Int | 0 |
| MN | Int | 0 |
| PV | Int | 0 |

| Name | Datentyp | Anfangswert |
|--------|----------|-------------|
| OUT | | |
| MX_IND | Bool | FALSE |
| MN_IND | Bool | FALSE |
| OUT | Int | 0 |
| STAT | | |
| FO1 | Bool | FALSE |
| HOP | Bool | TRUE |

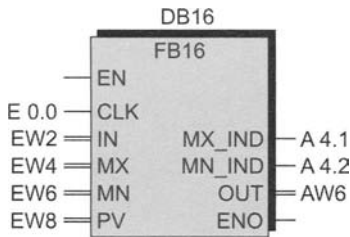
4. Anweisungsliste AWL:

```

Übergabe Startwert          L #MX          Zuweisung MX → OUT
    U HOP          >I          U #MX_IND
    SPBN _001      = #MX_IND    SPBNB _003
    L #PV          L #MX
    T #OUT        T #OUT
_001: NOP 0        L #IN
                  L #MN
                  <I
                  = #MN_IND    Zuweisung MN → OUT
Auswertung CLK-Eingang     U #MN_IND
    U #CLK          U #MN_IND
    FP #FO1        SPBNB _004
    R #HOP        L #MN
    NOT           T #OUT
    BEB          _004: NOP 0
                  L #IN
Abfrage Eingabewert       T #OUT
    L #IN        _002: NOP 0
    
```


5. Realisierung:

OBI:



Programmierung und Aufruf des Bausteins siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de>
STEP7-Lösungen → 5_2_06

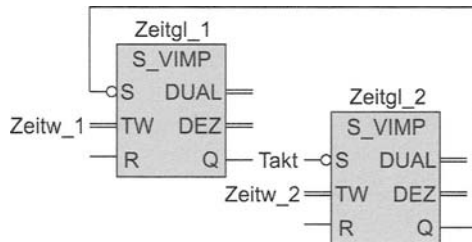
Lösung Lernaufgabe 5.7 Aufg. S. 70

1. Zuordnungstabelle der Eingänge und Ausgänge:

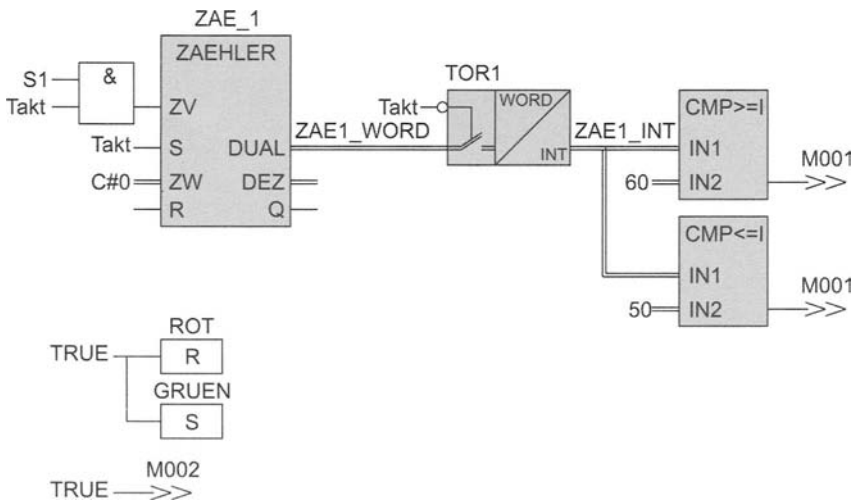
| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|--------------------|--------|----------|--------------------|---------|
| Impulsgeber | S1 | BOOL | Impuls S1 = 1 | E 0.1 |
| Quittiertaster | QUITT | BOOL | Betätigt QUITT = 1 | E 0.2 |
| Ausgangsvariable | | | | |
| Meldeleuchte ROT | H1 | BOOL | Leuchtet H1 = 1 | A 4.1 |
| Meldeleuchte GRUEN | H2 | BOOL | Leuchtet H2 = 1 | A 4.2 |
| Alarmsirene | AL | BOOL | Aktiv AL = 1 | A 4.3 |

2. Freigrafischer Funktionsplan:

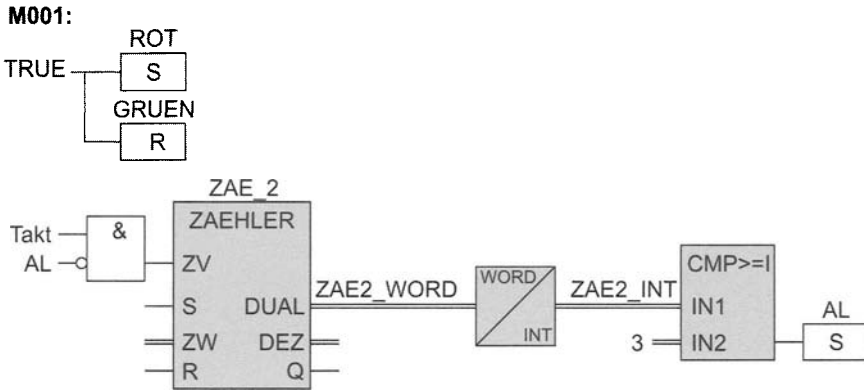
Taktgenerator:



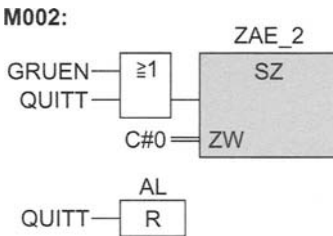
Impulzzählung und Auswertung:



Zählung der Rot-Phasen:



Quittierung:



3. Deklarationstabelle FC 17:

| Name | Datentyp |
|----------|----------|
| IN | |
| S1 | Bool |
| Zeitgl_1 | Timer |
| Zeitgw_1 | S5Time |
| Zeitgl_2 | Timer |
| Zeitgw_2 | S5Time |
| ZAE_1 | Counter |
| ZAE_2 | Counter |
| QUITT | Bool |

| Name | Datentyp |
|--------|----------|
| OUT | |
| ROT | Bool |
| AL | Bool |
| IN_OUT | |
| GRUEN | Bool |

| Name | Datentyp |
|-----------|----------|
| TEMP | |
| ZAE1_WORD | Word |
| ZAE1_INT | Int |
| ZAE2_WORD | Word |
| ZAE2_INT | Int |
| Takt | Bool |

4. Anweisungsliste AWL FC17:

```

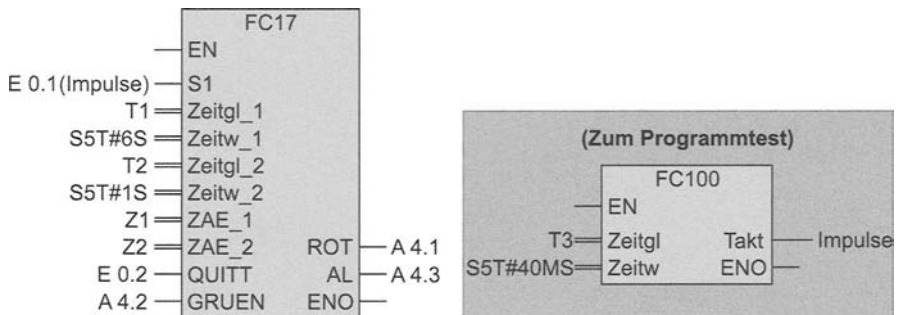
Taktgenerator      IMP-Zähler      IMP-Auswertung      L #ZAHL_INT
UN #Zeitgl_2      U #S1              UN #Takt              L 2
L #Zeitw_1        U #Takt            SPBNB_003            <I
SV #Zeitgl_1      ZV #ZAE_1          L #ZAHL_WORD         SPB  M001
UN #Zeitgl_1      U #Takt            T #ZAHL_INT
L #Zeitw_2        L C#0              _003: L #ZAHL_INT    SET
SV #Zeitgl_2      S #ZAE_1           L 10                  S #GRUEN
U #Zeitgl_1       L #ZAE_1           >I                     R #ROT
= #Takt           T #ZAHL_WORD       SPB  M001             SPA  M002
    
```

Fortsetzung Anweisungsliste AWL FC 17:

| | | |
|------------------------|----------------|----------------|
| Zählung der Rot-Phasen | Auswertung | Quittierung |
| M001: SET | der Rot-Phasen | M002: O #GRUEN |
| S #ROT | L #ROT_WORD | O #QUITT |
| R #GRUEN | T #ROT_INT | L C#0 |
| | | S #ZAE_2 |
| U #Takt | L #ROT_INT | |
| UN #AL | L 3 | U #QUITT |
| ZV #ZAE_2 | >=I | R #AL |
| L #ZAE_2 | S #AL | |
| T #ROT_WORD | | |

5. Realisierung:

OB1:



Programm s. beiliegende CD od. <http://www.automatisieren-mit-sps.de>
 STEP7-Lösungen → 5_2_07

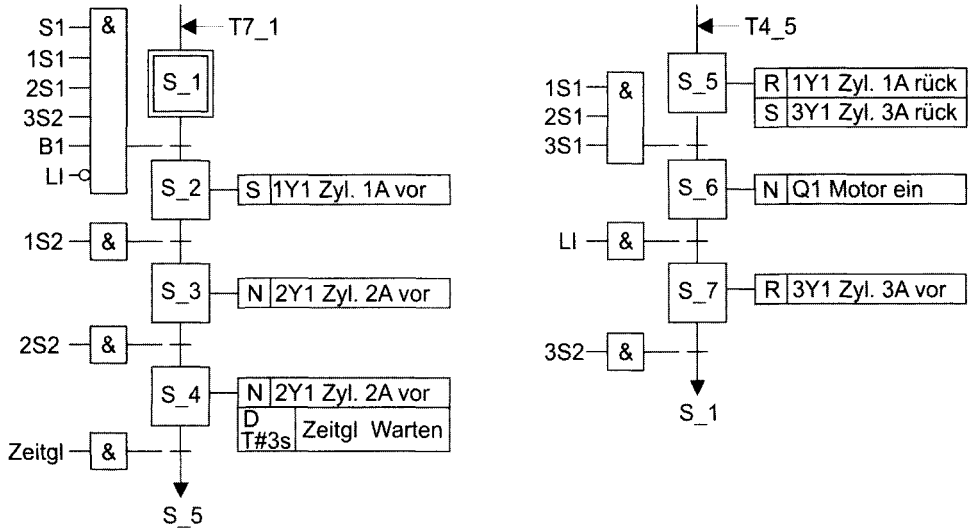
10.6 Lösungen der Lernaufgaben Kapitel 6**Lösung Lernaufgabe 6.1**

Aufg. S. 84

1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | | Adresse |
|-------------------------------|--------|----------|--------------------|-----------|---------|
| Schalter EIN / AUS | S1 | BOOL | Stellung EIN | S1 = 1 | E 0.1 |
| Endlage Zylinder 1A hinten | 1S1 | BOOL | Betätigt | 1S1 = 1 | E 0.2 |
| Endlage Zylinder 1A vorne | 1S2 | BOOL | Betätigt | 1S2 = 1 | E 0.3 |
| Endlage Zylinder 2A hinten | 2S1 | BOOL | Betätigt | 2S1 = 1 | E 0.4 |
| Endlage Zylinder 2A vorne | 2S2 | BOOL | Betätigt | 2S2 = 1 | E 0.5 |
| Endlage Zylinder 3A hinten | 3S1 | BOOL | Betätigt | 3S1 = 1 | E 0.6 |
| Endlage Zylinder 3A vorne | 3S2 | BOOL | Betätigt | 3S2 = 1 | E 0.7 |
| Sensor Deckel im Magazin | B1 | BOOL | Betätigt | B1 = 1 | E 1.0 |
| Lichtschanke | LI | BOOL | Unterbrochen | LI = 1 | E 1.1 |
| Taster Ablaufk. Grundstellung | RESET | BOOL | Betätigt | RESET = 1 | E 0.0 |
| Ausgangsvariable | | | | | |
| Magnetspule Zylinder 1A | 1Y1 | BOOL | Angezogen | 1Y1 = 1 | A 4.1 |
| Magnetspule Zylinder 2A | 2Y1 | BOOL | Angezogen | 2Y1 = 1 | A 4.2 |
| Magnetspule Zylinder 3A | 3Y1 | BOOL | Angezogen | 3Y1 = 1 | A 4.3 |
| Schütz Bandmotor | Q1 | BOOL | Motor ein | Q1 = 1 | A 4.4 |

2. Ablauf-Funktionsplan:

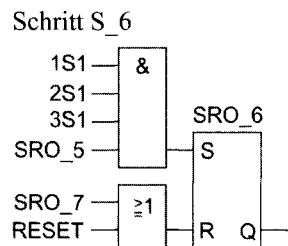
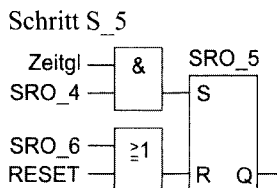
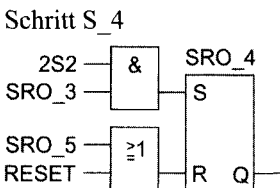
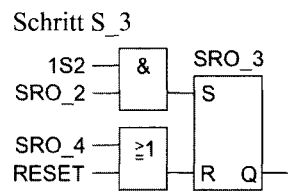
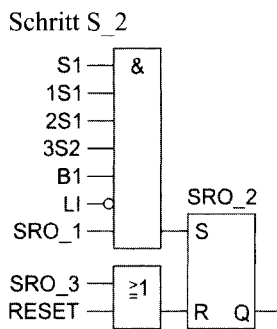
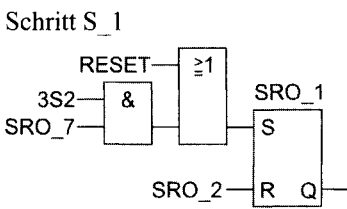


3. Transitionstabelle:

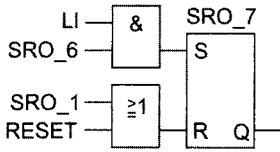
| Transition | Transitionsbedingung |
|------------|---|
| T-1 | $T7_1 = 3S2$ |
| T-2 | $T1_2 = S1 \& \overline{1S1} \& 2S1 \& 3S2 \& B1 \& \overline{LI}$ |
| T-3 | $T2_3 = 1S2$ |
| T-4 | $T3_4 = 2S2$ |

| Transition | Transitionsbedingung |
|------------|-----------------------------|
| T-5 | $T4_5 = \text{Zeitgl}$ |
| T-6 | $T5_6 = 1S1 \& 2S1 \& 3S1$ |
| T-7 | $T6_7 = LI$ |

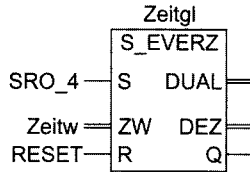
4. Funktionsplan:



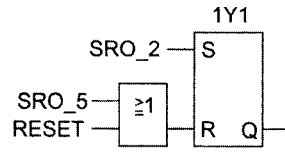
Schritt S_7



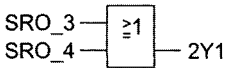
Zeitbildung



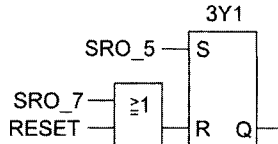
Aktion 1Y1



Aktion 2Y1



Aktion 3Y1



Aktion K1



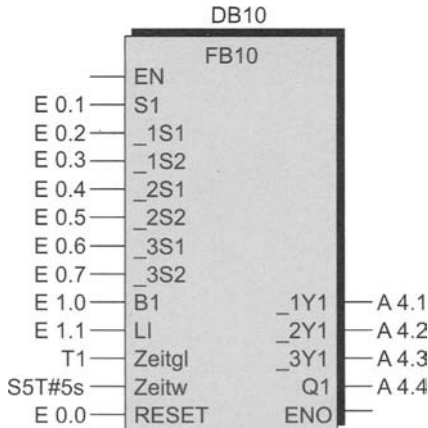
5. Deklarationstabelle FB 10:

| Name | Datentyp | Anfangswert |
|-----------|----------|-------------|
| IN | | |
| S1 | Bool | FALSE |
| _1S1 | Bool | FALSE |
| _1S2 | Bool | FALSE |
| _2S1 | Bool | FALSE |
| _2S2 | Bool | FALSE |
| _3S1 | Bool | FALSE |
| _3S2 | Bool | FALSE |
| B1 | Bool | FALSE |
| LI | Bool | FALSE |
| Zeitgl | Timer | FALSE |
| Zeitw | S5time | S5T#3S |
| RESET | Bool | FALSE |

| Name | Datentyp | Anfangswert |
|-------------|----------|-------------|
| OUT | | |
| _1Y1 | Bool | FALSE |
| _2Y1 | Bool | FALSE |
| _3Y1 | Bool | FALSE |
| Q1 | Bool | FALSE |
| STAT | | |
| SRO_1 | Bool | TRUE |
| SRO_2 | Bool | FALSE |
| SRO_3 | Bool | FALSE |
| SRO_4 | Bool | FALSE |
| SRO_5 | Bool | FALSE |
| SRO_6 | Bool | FALSE |
| SRO_7 | Bool | FALSE |

6. Realisierung:

OB1:



Programmierung und Aufruf des Bausteins siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de>
STEP7-Lösungen → 6_2_01

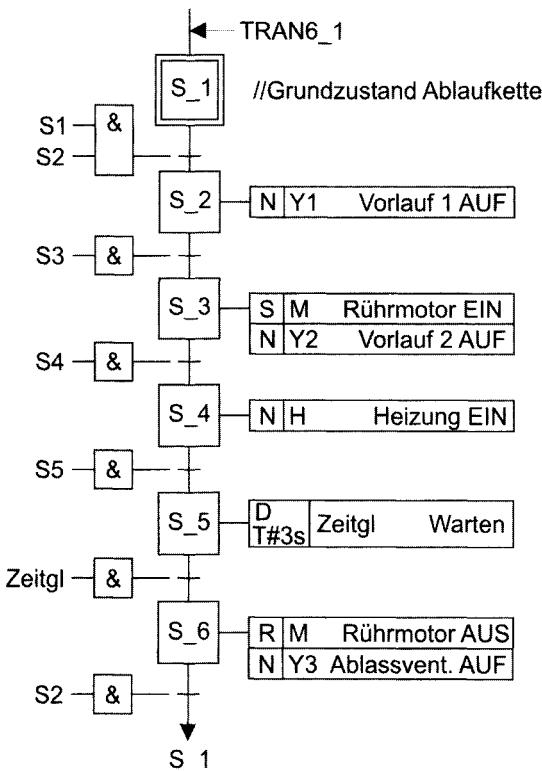
Lösung Lernaufgabe 6.2

Aufg. S. 85

1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | | Adresse |
|-------------------------------|--------|----------|--------------------|-----------|---------|
| Taster Start | S1 | BOOL | Betätigt | S1 = 1 | E 0.1 |
| Leermeldung | S2 | BOOL | Behälter leer | S2 = 1 | E 0.2 |
| Niveauschalter 1 | S3 | BOOL | Niveau 1 erreicht | S3 = 1 | E 0.3 |
| Niveauschalter 2 | S4 | BOOL | Niveau 2 erreicht | S4 = 1 | E 0.4 |
| Temperatursensor | S5 | BOOL | Temp. erreicht | S5 = 1 | E 0.5 |
| Taster Ablaufk. Grundstellung | RESET | BOOL | Betätigt | RESET = 1 | E 0.0 |
| Ausgangsvariable | | | | | |
| Zulaufventil 1 | Y1 | BOOL | Ventil auf | Y1 = 1 | A 4.1 |
| Zulaufventil 2 | Y2 | BOOL | Ventil auf | Y2 = 1 | A 4.2 |
| Ablassventil | Y3 | BOOL | Ventil auf | Y3 = 1 | A 4.3 |
| Heizung | H | BOOL | Heizung an | H = 1 | A 4.4 |
| Rührmotor | M | BOOL | Motor ein | M = 1 | A 4.5 |

2. Ablauf-Funktionsplan



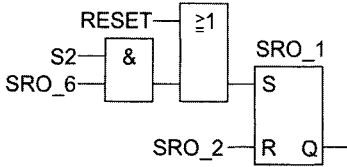
3. Transitionstabelle

| Transition | Weiterschaltbedingungen |
|------------|-------------------------|
| T-1 | T6_1 = S2 |
| T-2 | T1_2 = S1 & S2 |
| T-3 | T2_3 = S3 |
| T-4 | T3_4 = S4 |
| T-5 | T4_5 = S5 |
| T-6 | T5_6 = Zeitgl |

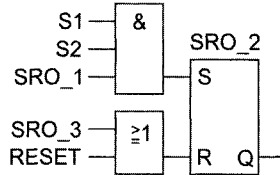
Realisierung mit einem anlagenspezifischen Funktionsbaustein

4. Funktionsplan:

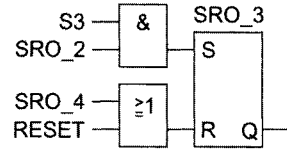
Schritt S_1



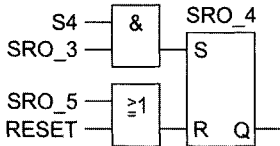
Schritt S_2



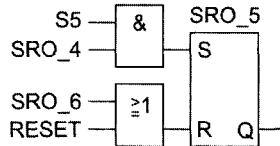
Schritt S_3



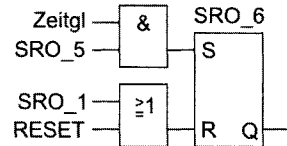
Schritt S_4



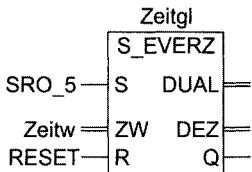
Schritt S_5



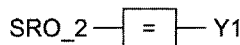
Schritt S_6



Aktion in S_5 Zeitbildung



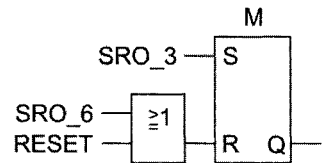
Aktion in S_2



Aktion S_3



Aktion in S_3 und S_6



Aktion in S_4



Aktion in S_6



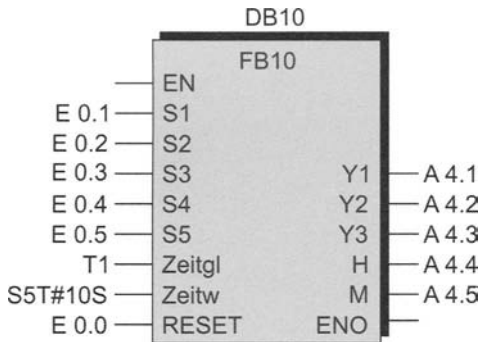
5. Deklarationstabelle FB10:

| Name | Datentyp | Anfangswert |
|-----------|----------|-------------|
| IN | | |
| S1 | Bool | FALSE |
| S2 | Bool | FALSE |
| S3 | Bool | FALSE |
| S4 | Bool | FALSE |
| S5 | Bool | FALSE |
| Zeitgl | Timer | |
| Zeitw | S5Time | S5T#0MS |
| RESET | Bool | FALSE |

| Name | Datentyp | Anfangswert |
|-------------|----------|-------------|
| OUT | | |
| Y1 | Bool | FALSE |
| Y2 | Bool | FALSE |
| Y3 | Bool | FALSE |
| H | Bool | FALSE |
| M | Bool | FALSE |
| STAT | | |
| SRO_1 | Bool | TRUE |
| SRO_2 | Bool | FALSE |
| SRO_3 | Bool | FALSE |
| SRO_4 | Bool | FALSE |
| SRO_5 | Bool | FALSE |
| SRO_6 | Bool | FALSE |

6. Realisierung:

OB1:

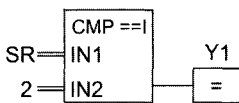


Programmierung und Aufruf des Bausteins siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de>
STEP7-Lösungen → 6_2_02a

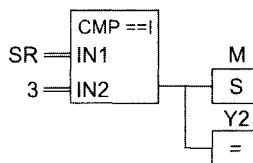
Realisierung mit dem Standard-Funktionsbaustein FB15 und dem Aktionsbaustein FC16

7. Funktionsplan des Aktionsbausteins FC16:

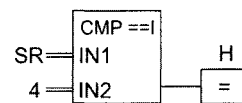
Aktion in S_2



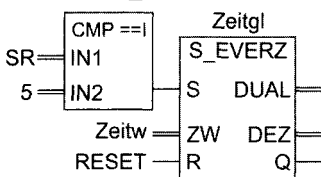
Aktion in S_3



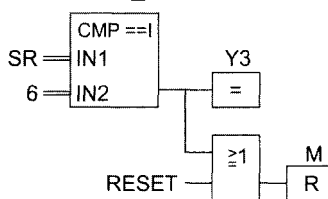
Aktion in S_4



Aktion in S_5



Aktion in S_6



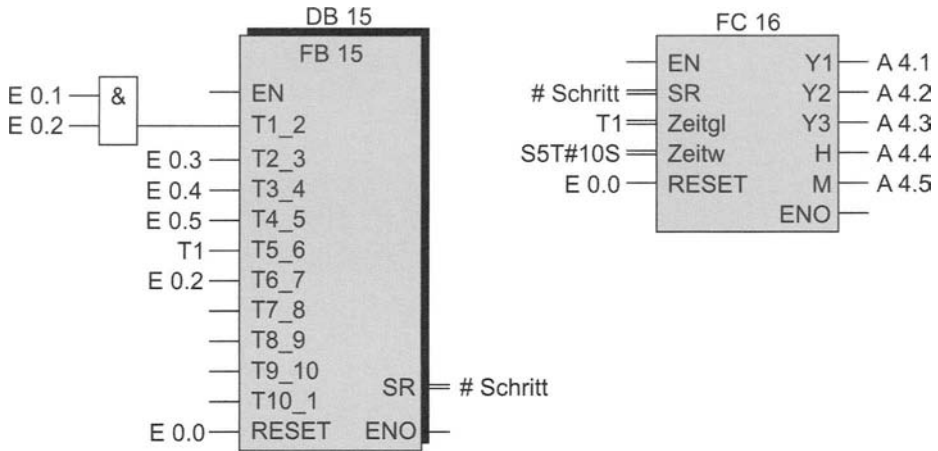
8. Deklarationstabelle Aktionsbaustein FC 16:

| Name | Datentyp |
|-----------|----------|
| IN | |
| SR | Int |
| Zeitgl | Timer |
| Zeitw | S5Time |
| RESET | Bool |

| Name | Datentyp |
|------------|----------|
| OUT | |
| Y1 | Bool |
| Y2 | Bool |
| Y3 | Bool |
| H | Bool |
| M | Bool |

9. Realisierung:

OBI:



Programm siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de>
 STEP7-Lösungen → 6_2_02b

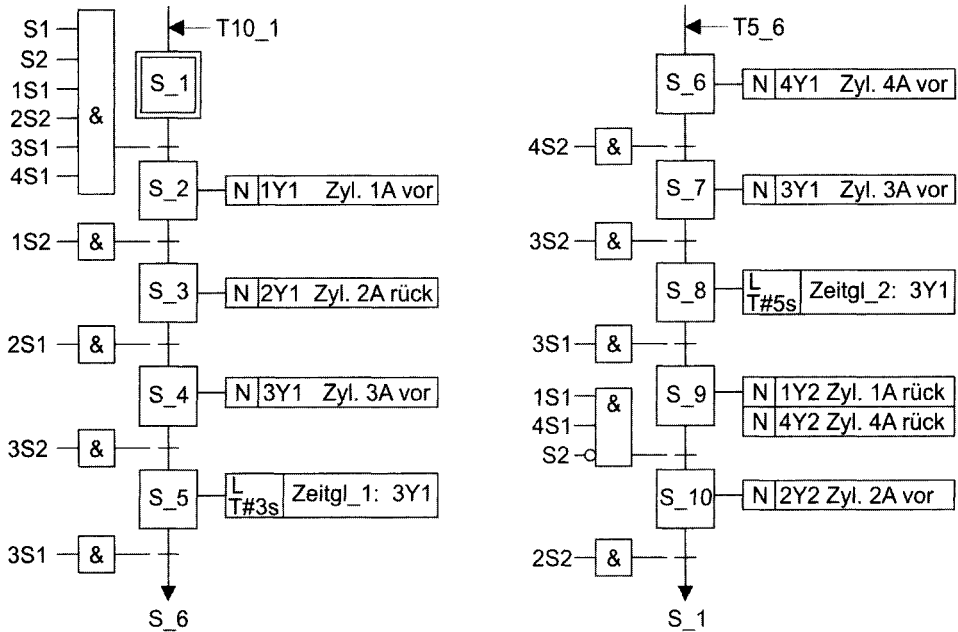
Lösung Lernaufgabe 6.3

Aufg. S. 86

1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | | Adresse |
|-------------------------------|--------|----------|--------------------|-----------|---------|
| Taster Start | S1 | BOOL | Betätigt | S1 = 1 | E 0.1 |
| Sensor Rohr vorhanden | S2 | BOOL | Rohr vorhanden | S2 = 1 | E 0.2 |
| Endlage Zylinder 1A hinten | 1S1 | BOOL | Betätigt | 1S1 = 1 | E 0.3 |
| Endlage Zylinder 1A vorne | 1S2 | BOOL | Betätigt | 1S2 = 1 | E 0.4 |
| Endlage Zylinder 2A hinten | 2S1 | BOOL | Betätigt | 2S1 = 1 | E 0.5 |
| Endlage Zylinder 2A vorne | 2S2 | BOOL | Betätigt | 2S2 = 1 | E 0.6 |
| Endlage Zylinder 3A hinten | 3S1 | BOOL | Betätigt | 3S1 = 1 | E 0.7 |
| Endlage Zylinder 3A vorne | 3S2 | BOOL | Betätigt | 3S2 = 1 | E 1.0 |
| Endlage Zylinder 4A hinten | 4S1 | BOOL | Betätigt | 4S1 = 1 | E 1.1 |
| Endlage Zylinder 4A vorne | 4S2 | BOOL | Betätigt | 4S2 = 1 | E 1.2 |
| Taster Ablaufk. Grundstellung | RESET | BOOL | Betätigt | RESET = 1 | E 0.0 |
| Ausgangsvariable | | | | | |
| Magnetspule Zyl. 1A vor | 1Y1 | BOOL | Angezogen | 1Y1 = 1 | A 4.1 |
| Magnetspule Zyl. 1A zurück | 1Y2 | BOOL | Angezogen | 1Y2 = 1 | A 4.2 |
| Magnetspule Zyl. 2A zurück | 2Y1 | BOOL | Angezogen | 2Y1 = 1 | A 4.3 |
| Magnetspule Zyl. 2A vor | 2Y2 | BOOL | Angezogen | 2Y2 = 1 | A 4.4 |
| Magnetspule Zyl. 3A vor | 3Y1 | BOOL | Angezogen | 3Y1 = 1 | A 4.5 |
| Magnetspule Zyl. 4A vor | 4Y1 | BOOL | Angezogen | 4Y1 = 1 | A 4.6 |
| Magnetspule Zyl. 4A zurück | 4Y2 | BOOL | Angezogen | 4Y2 = 1 | A 4.7 |

2. Ablauf-Funktionsplan:



3. Transitionstabelle:

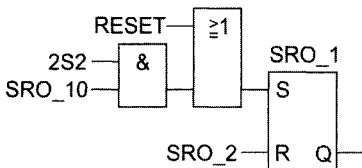
| Transition | Transitionsbedingung |
|------------|---|
| T-1 | T10_1 = 2S2 |
| T-2 | T1_2 = S1 & S2 & 1S1 & 2S2 & 3S1 & 4S1 |
| T-3 | T2_3 = 1S2 |
| T-4 | T3_4 = 2S1 |
| T-5 | T4_5 = 3S2 |

| Transition | Transitionsbedingung |
|------------|-------------------------------------|
| T-6 | T5_6 = 3S1 |
| T-7 | T6_7 = 4S2 |
| T-8 | T7_8 = 3S2 |
| T-9 | T8_9 = 3S1 |
| T-10 | T9_10 = 1S1 & 4S1 & $\overline{S2}$ |

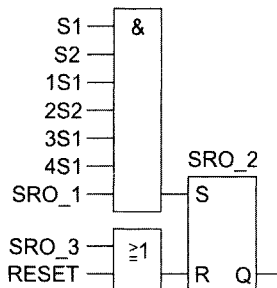
Realisierung mit einem anlagenspezifischen Funktionsbaustein

4. Funktionsplan:

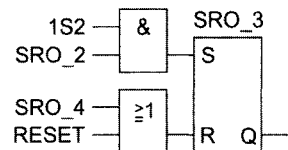
Schritt S_1



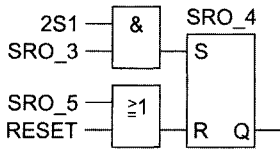
Schritt S_2



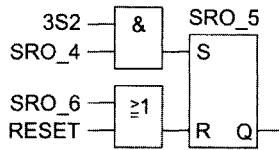
Schritt S_3



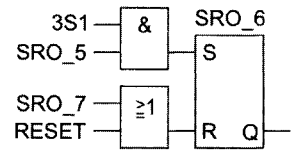
Schritt S_4



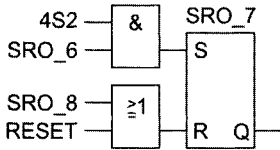
Schritt S_5



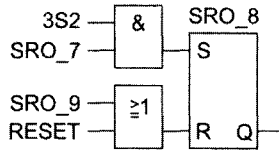
Schritt S_6



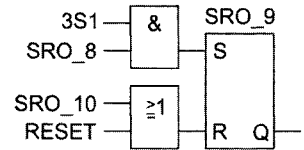
Schritt S_7



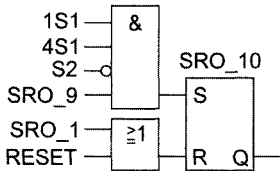
Schritt S_8



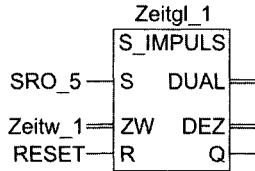
Schritt S_9



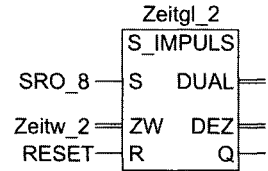
Schritt S_10



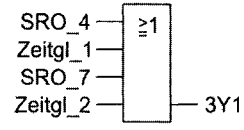
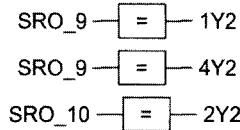
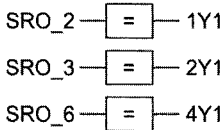
Aktion in S_5 Zeitbildung 1



Aktion in S_8 Zeitbildung 2



Aktionen



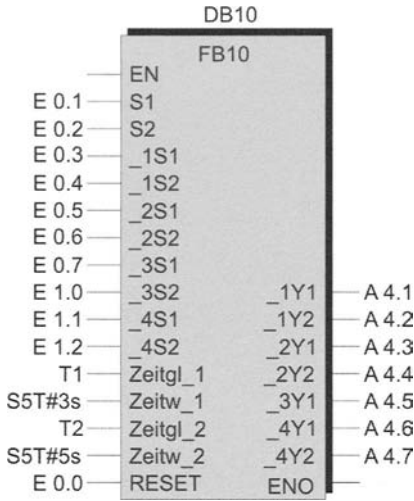
5. Deklarationstabelle FB10:

| Name | Datentyp | Anfangswert |
|----------|----------|-------------|
| IN | | |
| S1 | Bool | FALSE |
| S2 | Bool | FALSE |
| 1S1 | Bool | FALSE |
| 1S2 | Bool | FALSE |
| 2S1 | Bool | FALSE |
| 2S2 | Bool | FALSE |
| 3S1 | Bool | FALSE |
| 3S2 | Bool | FALSE |
| 4S1 | Bool | FALSE |
| 4S2 | Bool | FALSE |
| Zeitgl_1 | Timer | |
| Zeitw_1 | S5Time | S5T#0MS |
| Zeitgl_2 | Timer | |
| Zeitw_2 | S5Time | S5T#0MS |
| RESET | Bool | FALSE |

| Name | Datentyp | Anfangswert |
|--------|----------|-------------|
| OUT | | |
| 1Y1 | Bool | FALSE |
| 1Y2 | Bool | FALSE |
| 2Y1 | Bool | FALSE |
| 2Y2 | Bool | FALSE |
| 3Y1 | Bool | FALSE |
| 4Y1 | Bool | FALSE |
| 4Y2 | Bool | FALSE |
| STAT | | |
| SRO_1 | Bool | TRUE |
| SRO_2 | Bool | FALSE |
| SRO_3 | Bool | FALSE |
| SRO_4 | Bool | FALSE |
| SRO_5 | Bool | FALSE |
| SRO_6 | Bool | FALSE |
| SRO_7 | Bool | FALSE |
| SRO_8 | Bool | FALSE |
| SRO_9 | Bool | FALSE |
| SRO_10 | Bool | FALSE |

6. Realisierung:

OB1:



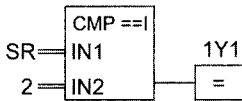
STEP7 Programm siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de>

STEP7-Lösungen → 6_2_03a

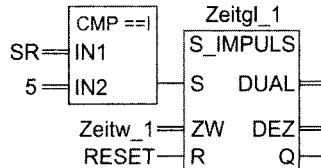
Realisierung mit dem Standard-Funktionsbaustein FB15 und dem Aktionsbaustein FC16

7. Funktionsplan des Aktionsbausteins FC16:

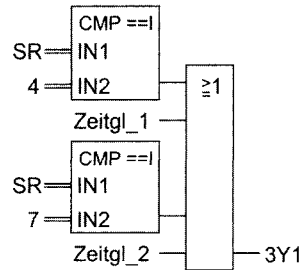
Ansteuerung 1Y1



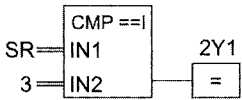
Zeitbildung 1



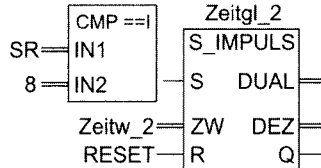
Ansteuerung 3Y1



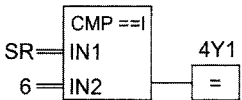
Ansteuerung 2Y1



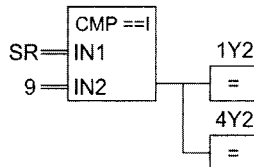
Zeitbildung 2



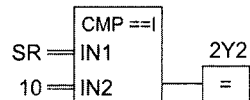
Ansteuerung 4Y1



Ansteuerung 1Y2 und 4Y2



Ansteuerung 2Y2



8. Deklarationstabelle Aktionsbaustein FC 16:

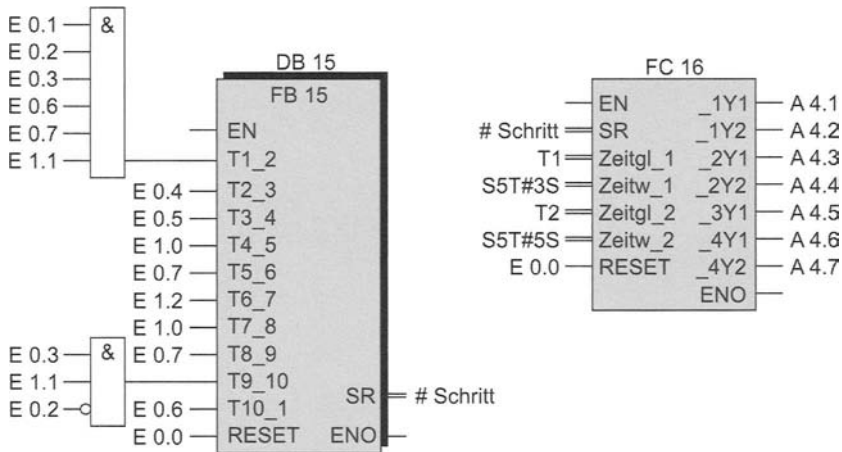
| Name | Datentyp |
|----------|----------|
| IN | |
| SR | Int |
| Zeitgl_1 | Timer |
| Zeitw_1 | S5Time |
| Zeitgl_2 | Timer |
| Zeitw_2 | S5Time |
| RESET | Bool |

| Name | Datentyp |
|------|----------|
| OUT | |
| _1Y1 | Bool |
| _1Y2 | Bool |
| _2Y1 | Bool |
| _2Y2 | Bool |

| Name | Datentyp |
|------|----------|
| OUT | |
| _3Y1 | Bool |
| _4Y1 | Bool |
| _4Y2 | Bool |

9. Realisierung:

OB1:



Programm siehe <http://www.automatisieren-mit-sps.de> unter STEP7-Lösungen → 6_2_03b

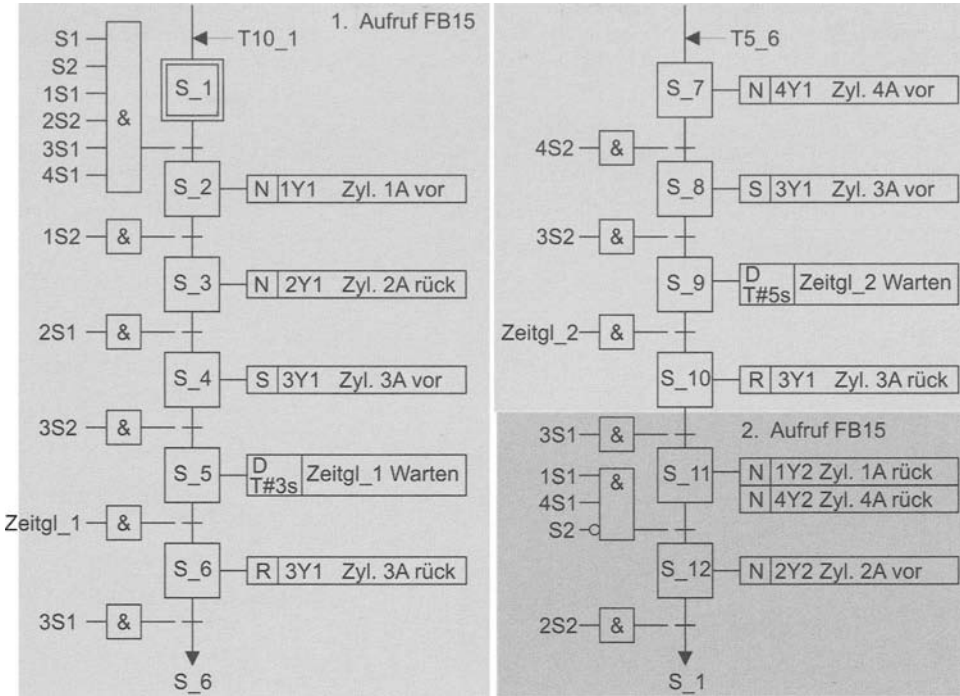
Ergänzung: Kaskadierung des Standardfunktionsbausteins FB15

Wird beim Aufstellen des Ablauf-Funktionsplanes bei der Zeitbildung statt des L-Befehls (Zeitbegrenzung) der D-Befehl (Zeitverzögerung) verwendet, ergeben sich insgesamt 12 Ablaufschritte. Zur Realisierung dieses Ablauf-Funktionsplanes muss der Standard-Funktionsbaustein FB15 um zwei Schritte erweitert werden oder durch zweimaligen Aufruf eine Kaskadierung herbeigeführt werden.

Nachfolgend soll gezeigt werden, wie beim Auftreten von mehr als 10 Ablaufschritten der Ablauf-Funktionsplan durch den zweimaligen Aufruf von FB15 umgesetzt werden kann.

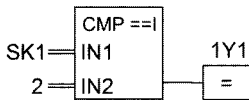
Die Ausgangsgröße SR, welche die aktuelle Schrittnummer des Funktionsbausteins FB15 angibt, wird beim ersten Aufruf der Übergabevariablen #Schr_K1 und beim zweiten Aufruf der Übergabevariablen #Schr_K2 zugewiesen. Die Variable #Schr_K1 gibt somit die Ablaufschritte 1 bis 10 an. Die Ablaufschritte 11 bis 19 werden durch die Variable #Schr_K2 angegeben. Dabei muss beachtet werden, dass der Ablaufschritt 11 aktiv ist, wenn die Übergangsvariable #Schr_K2 den Wert 2 hat. Für die weiteren Schritte gilt Entsprechendes. (Hinweis: der Initialschritt (S_1) kann nicht als Ablaufschritt 11 verwendet werden.)

Veränderte Ablaufkette:

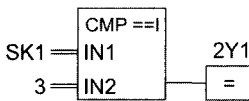


Veränderter Funktionsplan des Aktionsbausteins FC16:

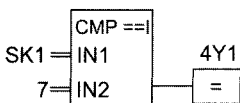
Aktion in S_2



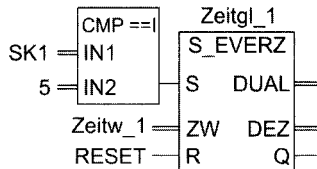
Aktion in S_3



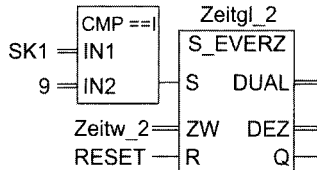
Aktion in S_7



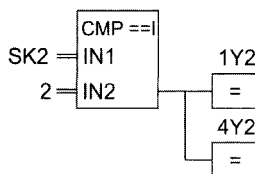
Aktion in S_5 Zeitbildung 1



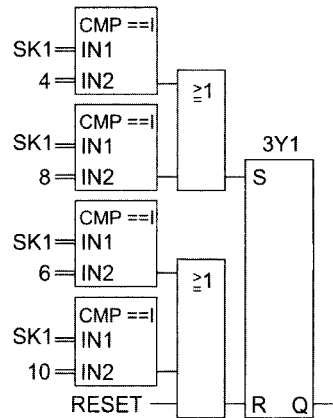
Aktion in S_9 Zeitbildung 2



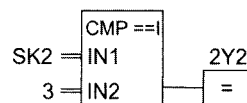
Aktion in S_11



Aktion in S_4, S_6, S_8, S_10



Aktion in S_12



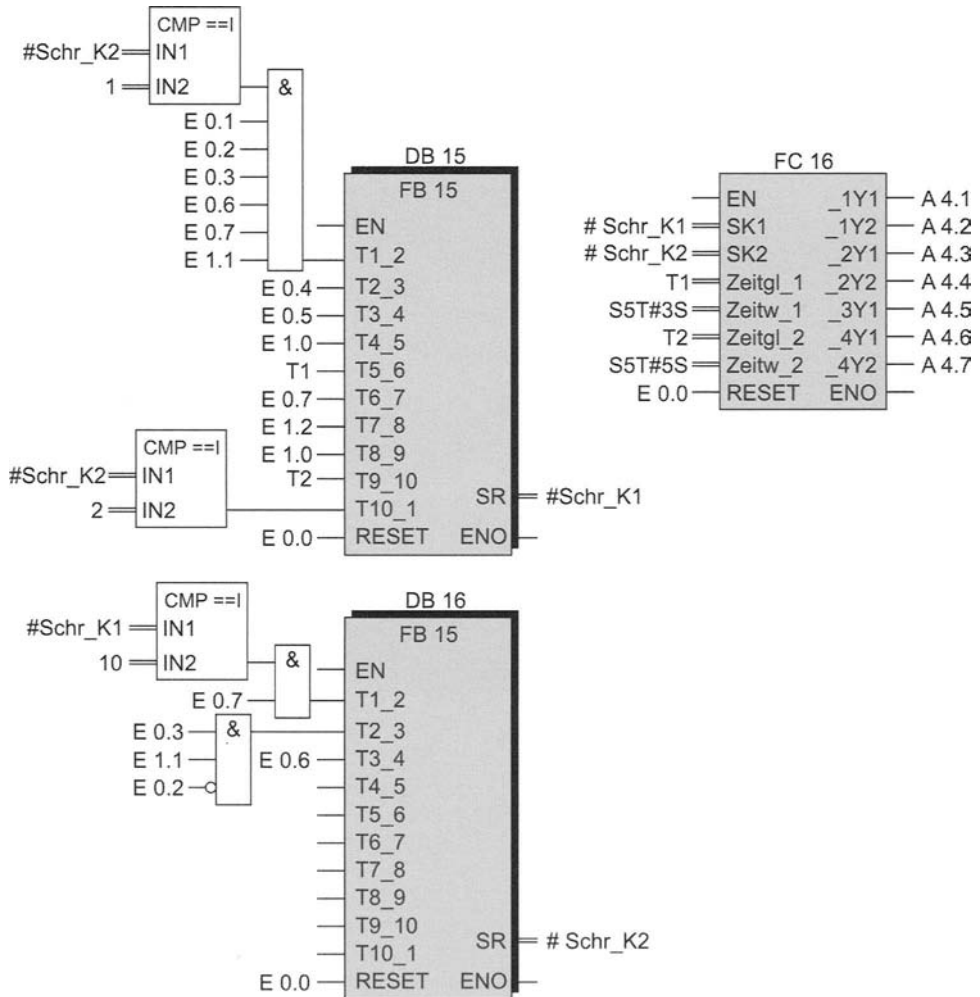
Veränderte Deklarationstabelle Aktionsbaustein FC 16:

| Name | Datentyp |
|-----------|----------|
| IN | |
| SK1 | Int |
| SK2 | Int |
| Zeitgl_1 | Timer |
| Zeitw_1 | S5Time |
| Zeitgl_2 | Timer |
| Zeitw_2 | S5Time |
| RESET | Bool |

| Name | Datentyp |
|------------|----------|
| OUT | |
| _1Y1 | Bool |
| _1Y2 | Bool |
| _2Y1 | Bool |
| _2Y2 | Bool |

| Name | Datentyp |
|------------|----------|
| OUT | |
| _3Y1 | Bool |
| _4Y1 | Bool |
| _4Y2 | Bool |

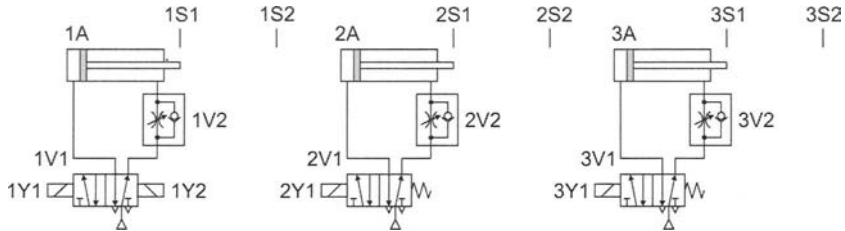
Veränderter Aufruf im OB1:



Lösung Lernaufgabe 6.4

Aufg. S. 87

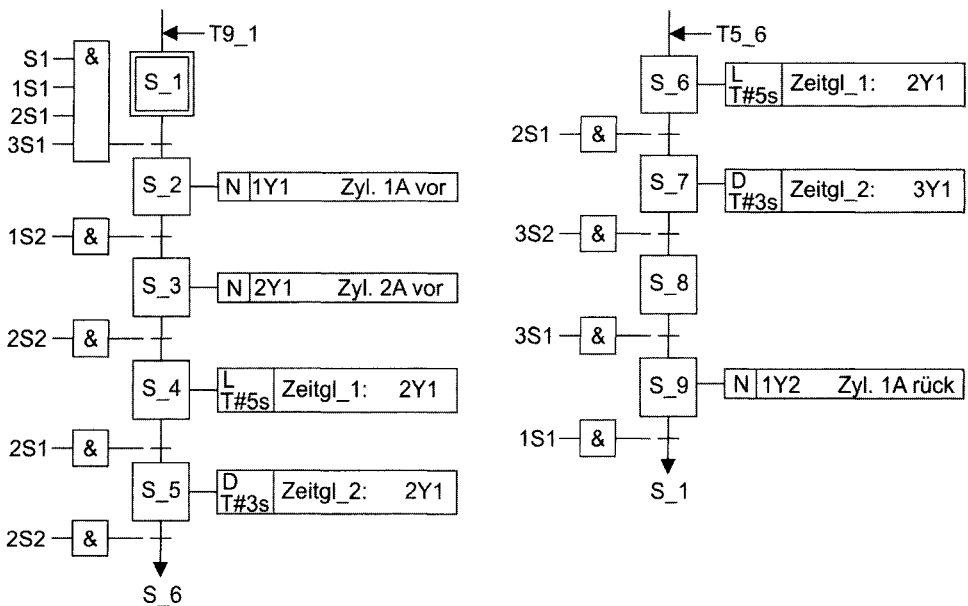
1. Pneumatikplan:



2. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | | Adresse |
|-------------------------------|--------|----------|--------------------|-----------|---------|
| Taster Start | S1 | BOOL | Betätigt | S1 = 1 | E 0.1 |
| Endlage Zylinder 1A hinten | 1S1 | BOOL | Betätigt | 1S1 = 1 | E 0.2 |
| Endlage Zylinder 1A vorne | 1S2 | BOOL | Betätigt | 1S2 = 1 | E 0.3 |
| Endlage Zylinder 2A hinten | 2S1 | BOOL | Betätigt | 2S1 = 1 | E 0.4 |
| Endlage Zylinder 2A vorne | 2S2 | BOOL | Betätigt | 2S2 = 1 | E 0.5 |
| Endlage Zylinder 3A hinten | 3S1 | BOOL | Betätigt | 3S1 = 1 | E 0.6 |
| Endlage Zylinder 3A vorne | 3S2 | BOOL | Betätigt | 3S2 = 1 | E 0.7 |
| Taster Ablaufk. Grundstellung | RESET | BOOL | Betätigt | RESET = 1 | E 0.0 |
| Ausgangsvariable | | | | | |
| Magnetspule Zyl. 1A vor | 1Y1 | BOOL | Angezogen | 1Y1 = 1 | A 4.1 |
| Magnetspule Zyl. 1A zurück | 1Y2 | BOOL | Angezogen | 1Y2 = 1 | A 4.2 |
| Magnetspule Zyl. 2A vor | 2Y1 | BOOL | Angezogen | 2Y1 = 1 | A 4.3 |
| Magnetspule Zyl. 3A vor | 3Y1 | BOOL | Angezogen | 3Y1 = 1 | A 4.4 |

3. Ablauf-Funktionsplan:

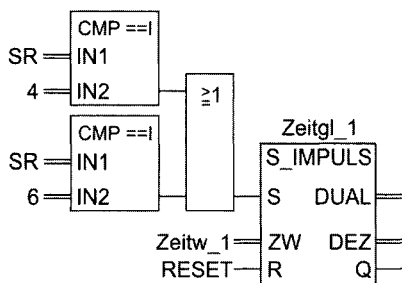


4. Transitionstabelle:

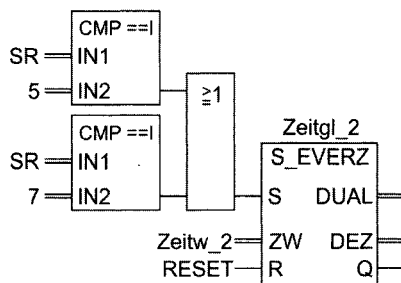
| Transition | Transitionsbedingung |
|------------|-----------------------------|
| T-1 | T9_1 = 1S1 |
| T-2 | T1_2 = S1 & 1S1 & 2S1 & 3S1 |
| T-3 | T2_3 = 1S2 |
| T-4 | T3_4 = 2S2 |
| T-5 | T4_5 = 2S1 |
| T-6 | T5_6 = 2S2 |
| T-7 | T6_7 = 2S1 |
| T-8 | T7_8 = 3S2 |
| T-9 | T8_9 = 3S1 |

5. Funktionsplan der Aktionsausgabe:

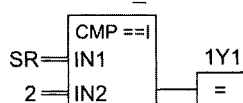
Aktion in S_4, S_6 Zeitbildung 1



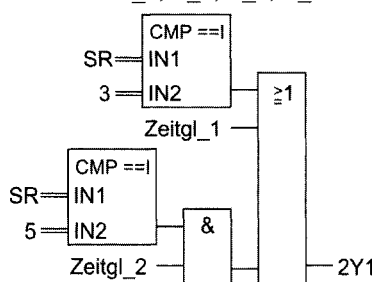
Aktion in S_5, S_7 Zeitbildung 2



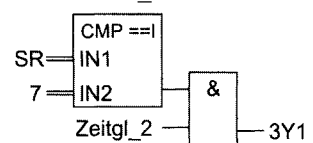
Aktion in S_2



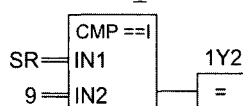
Aktion in S_3, S_4, S_5, S_6



Aktion in S_7



Aktion in S_9



6. Deklarationstabelle Aktionsbaustein FC 16:

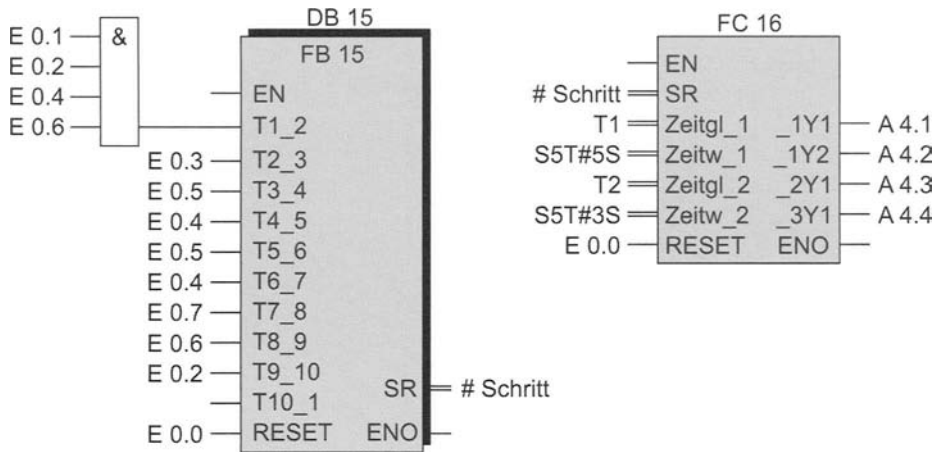
| Name | Datentyp |
|----------|----------|
| IN | |
| SR | Int |
| Zeitgl_1 | Timer |
| Zeitw_1 | S5Time |

| Name | Datentyp |
|----------|----------|
| IN | |
| Zeitgl_2 | Timer |
| Zeitw_2 | S5Time |
| RESET | Bool |

| Name | Datentyp |
|------|----------|
| OUT | |
| _1Y1 | Bool |
| _1Y2 | Bool |
| _2Y1 | Bool |
| _3Y1 | Bool |

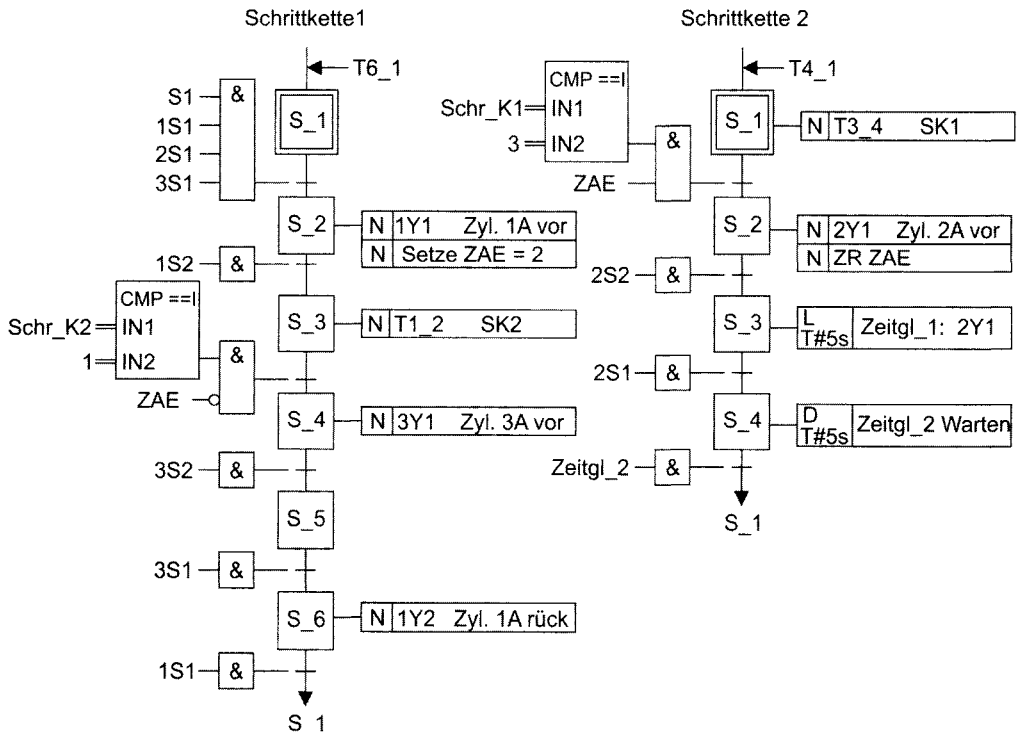
7. Realisierung:

OBI:



Programm siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de>
 STEP7-Lösungen → 6_2_04a

8. Ablauf-Funktionsplan für Lösungsansatz 2:



Hinweis: Die Variable Schr_K1 gibt die aktuelle Schrittnummer der Schrittkeette 1 und die Variable Schr_K2 die aktuelle Schrittnummer der Schrittkeette 2 an.

9. Transitionstabellen:

Schrittkeite 1

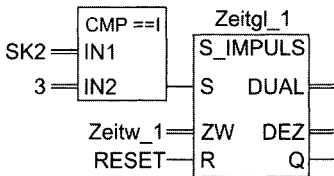
| Transition | Transitionsbedingung |
|------------|---|
| T1_2 | $T1_2 = S1 \ \& \ 1S1 \ \& \ 2S1 \ \& \ 3S1$ |
| T2_3 | $T2_3 = 1S2$ |
| T3_4 | $T3_4 = (Schr_K2=1) \ \& \ \overline{ZAE}$ |
| T4_5 | $T4_5 = 3S2$ |
| T5_6 | $T5_6 = 3S1$ |
| T6_1 | $T6_1 = 1S1$ |

Schrittkeite 2

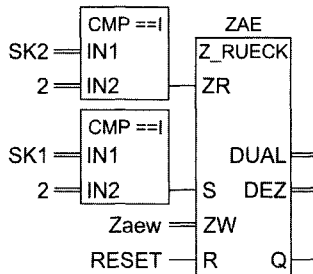
| Transition | Transitionsbedingung |
|------------|-------------------------------------|
| T1_2 | $T1_2 = (Schr_K1 = 3) \ \& \ ZAE$ |
| T2_3 | $T2_3 = 2S2$ |
| T3_4 | $T3_4 = 2S1$ |
| T4_1 | $T4_1 = Zeitgl_2$ |

10. Funktionsplan der Aktionsausgabe:

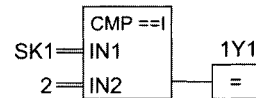
Aktion in S_3 (SK2) Zeit 1



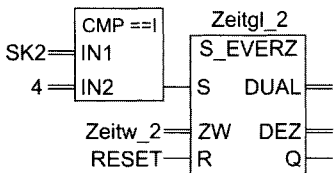
Akt. in S_2 (SK2) S_2 (SK1)



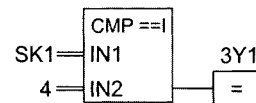
Aktion in S_2 (SK1)



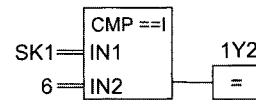
Aktion in S_4 (SK2) Zeit 2



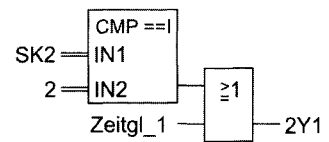
Aktion in S_4 (SK1)



Aktion in S_6 (SK1)



Aktion in S_2 (SK2)



11. Deklarationstabelle Aktionsbaustein FC 16:

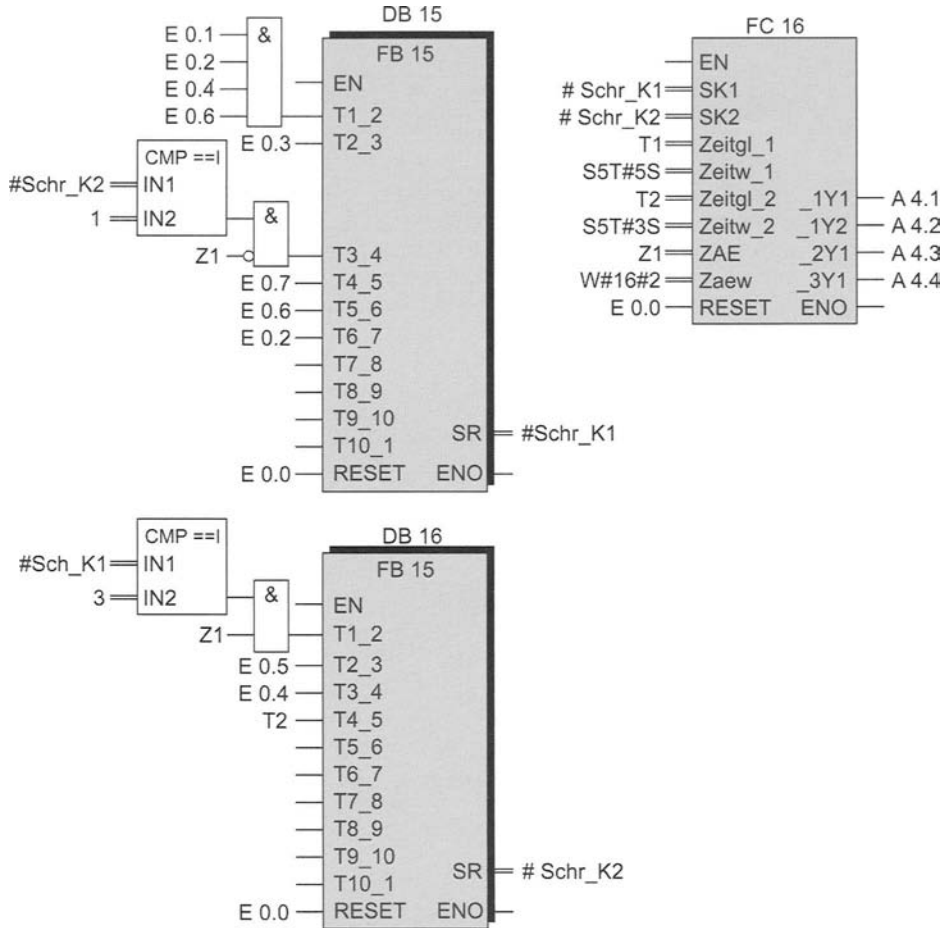
| Name | Datentyp |
|----------|----------|
| IN | |
| SK1 | Int |
| SK2 | Int |
| Zeitgl_1 | Timer |
| Zeitw_1 | S5Time |
| Zeitgl_2 | Timer |
| Zeitw_2 | S5Time |

| Name | Datentyp |
|-------|----------|
| IN | |
| ZAE | Counter |
| Zaew | Word |
| RESET | Bool |

| Name | Datentyp |
|------|----------|
| OUT | |
| _1Y1 | Bool |
| _1Y2 | Bool |
| _2Y1 | Bool |
| _3Y1 | Bool |

12. Realisierung:

OBI:



STEP7 Programm siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de>
 STEP7-Lösungen → 6_2_04b

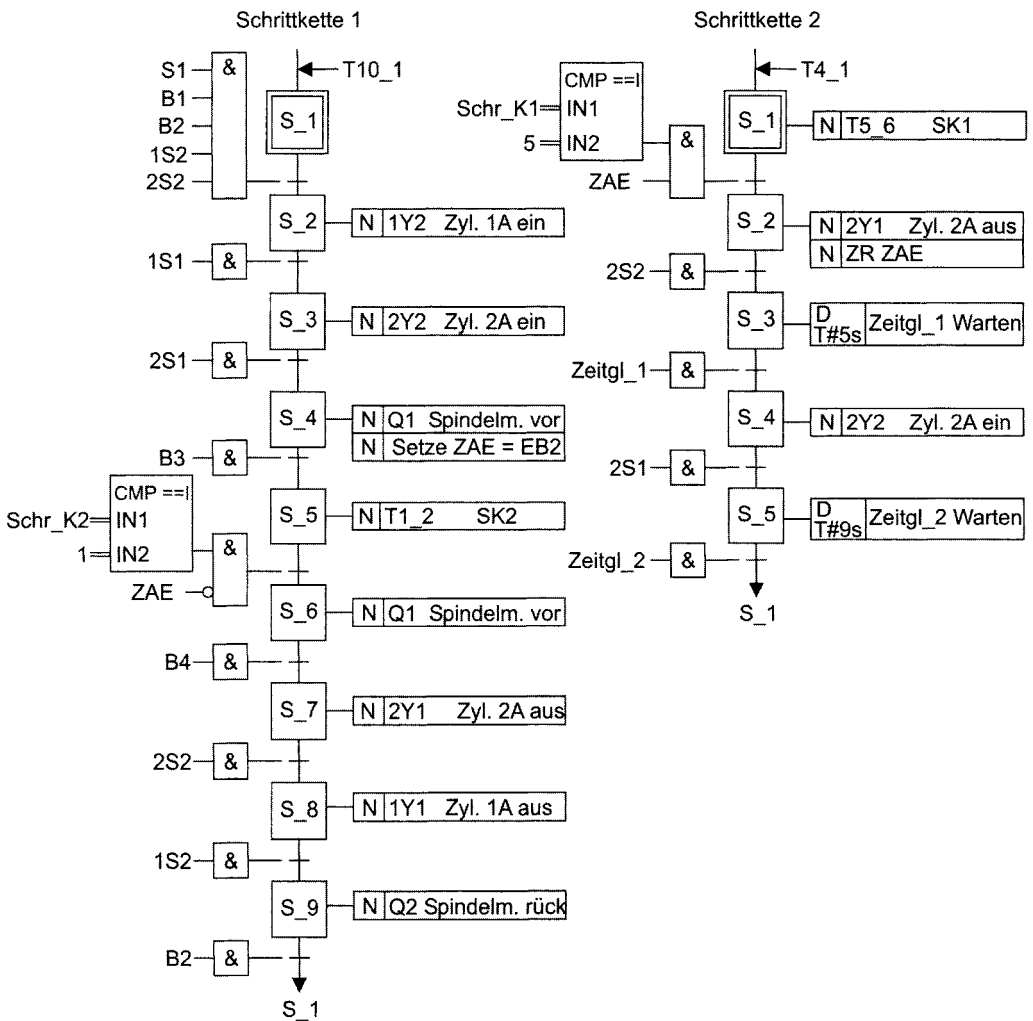
Lösung Lernaufgabe 6.5 Aufg. S. 88

1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|----------------------------|--------|----------|--------------------|---------|
| Taster Start | S1 | BOOL | Betätigt S1 = 1 | E 0.1 |
| Endlage Zylinder 1A hinten | 1S1 | BOOL | Betätigt 1S1 = 1 | E 0.2 |
| Endlage Zylinder 1A vorne | 1S2 | BOOL | Betätigt 1S2 = 1 | E 0.3 |
| Endlage Zylinder 2A hinten | 2S1 | BOOL | Betätigt 2S1 = 1 | E 0.4 |
| Endlage Zylinder 2A vorne | 2S2 | BOOL | Betätigt 2S2 = 1 | E 0.5 |
| Sensor Korb vorhanden | B1 | BOOL | Betätigt B1 = 1 | E 0.6 |
| Sensor Position 1 | B2 | BOOL | Betätigt B2 = 1 | E 0.7 |

| | | | | | |
|-------------------------------|-------|--|-----------------|-----------|-------|
| Sensor Position 2 | B3 | BOOL | Betätigt | B3 = 1 | E 1.0 |
| Sensor Position 3 | B4 | BOOL | Betätigt | B4 = 1 | E 1.1 |
| Taster Ablaufk. Grundstellung | RESET | BOOL | Betätigt | RESET = 1 | E 0.0 |
| Zifferneinsteller | EB2 | Byte | Ziffern 0 bis 9 | | EB 2 |
| Ausgangsvariable | | | | | |
| Magnetspule Zyl. 1A aus | 1Y1 | BOOL | Angezogen | 1Y1 = 1 | A 4.1 |
| Magnetspule Zyl. 1A ein | 1Y2 | BOOL | Angezogen | 1Y2 = 1 | A 4.2 |
| Magnetspule Zyl. 2A aus | 2Y1 | BOOL | Angezogen | 2Y1 = 1 | A 4.3 |
| Magnetspule Zyl. 2A ein | 2Y2 | BOOL </td <td>Angezogen</td> <td>2Y2 = 1</td> <td>A 4.4</td> | Angezogen | 2Y2 = 1 | A 4.4 |
| Spindelmotor vor | Q1 | BOOL | Motor vor | Q1 = 1 | A 4.5 |
| Spindelmotor zurück | Q2 | BOOL | Motor zurück | Q2 = 1 | A 4.6 |

2. Ablauf-Funktionsplan:



3. Transitionstabellen:

Schrittkeite 1

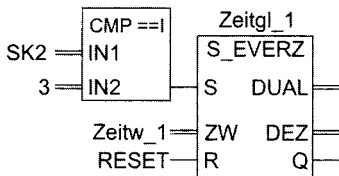
| Transition | Transitionsbedingung |
|------------|--------------------------|
| T-1 | T9_1 = B2 |
| T-2 | T1_2 = S1&B1&B2&1S2&2S2 |
| T-3 | T2_3 = 1S1 |
| T-4 | T3_4 = 2S1 |
| T-5 | T4_5 = B3 |
| T-6 | T5_6 = (Schr_K2=1) & ZAE |
| T-7 | T6_7 = B4 |
| T-8 | T7_8 = 2S2 |
| T-9 | T8_9 = 1S2 |

Schrittkeite 2

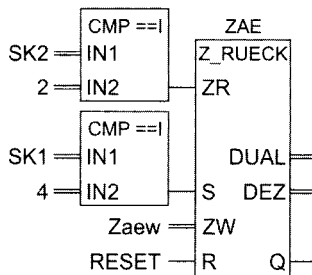
| Transition | Transitionsbedingung |
|------------|----------------------------|
| T-1 | T5_1 = Zeitgl_2 |
| T-2 | T1_2 = (Schr_K1 = 3) & ZAE |
| T-3 | T2_3 = 2S2 |
| T-4 | T3_4 = Zeitgl_1 |
| T-5 | T4_5 = 2S1 |

4. Funktionsplan der Aktionsausgabe:

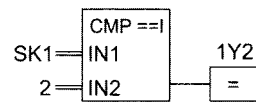
Aktion S_3 (AK2) Zeit 1



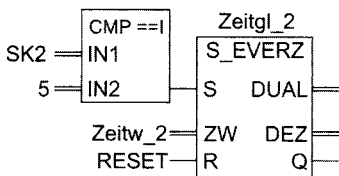
Aktion S_2 (AK2) S_4 (AK1)



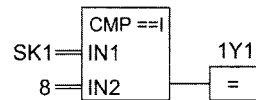
Aktion S_2 (AK1)



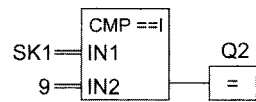
Aktion S_5 (AK2) Zeit 2



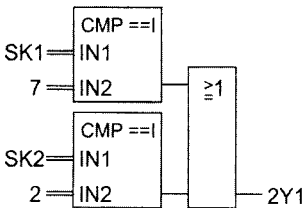
Aktion S_8 (AK1)



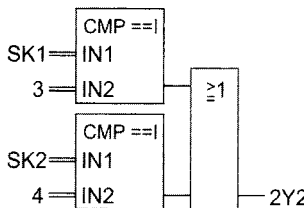
Aktion S_9 (AK1)



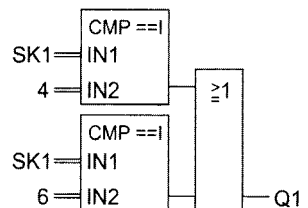
Akt. S_7 (AK1) S_2 (AK2)



Akt. S_3 (AK1) S_4 (AK2)



Akt. S_4 (AK1) S_6 (AK1)

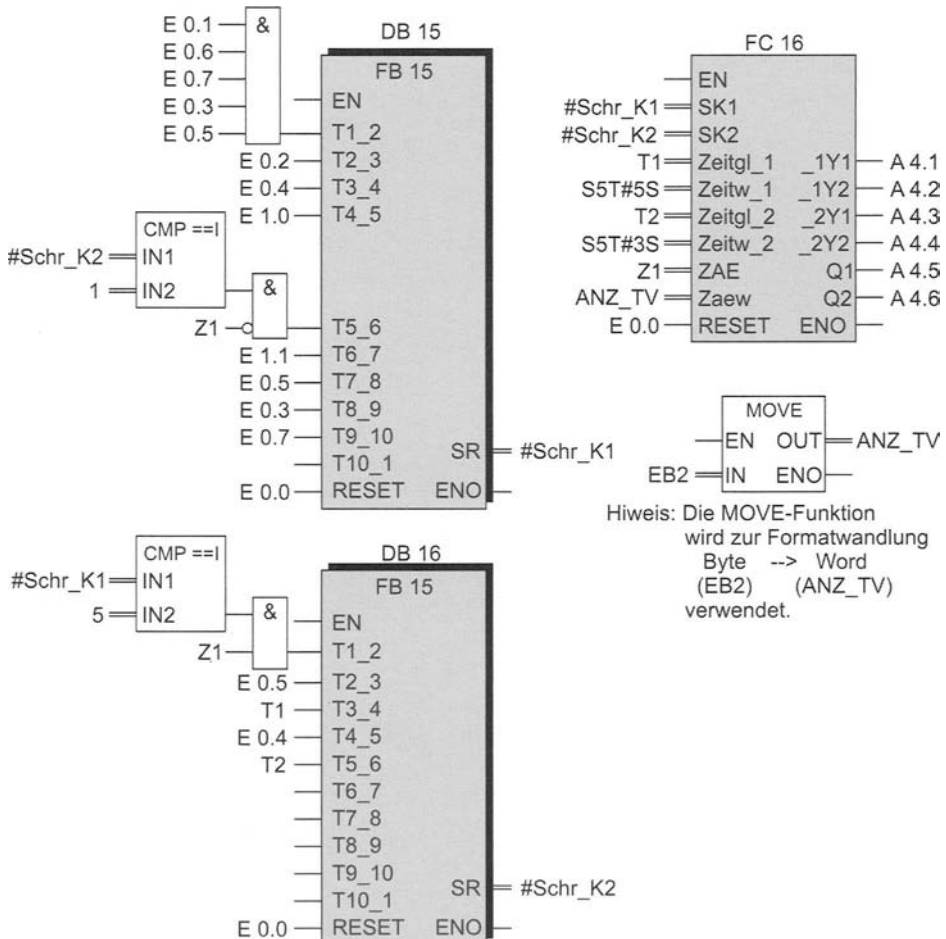


5. Deklarationstabelle Aktionsbaustein FC 16:

| Name | Datentyp |
|----------|----------|
| IN | |
| SK1 | Int |
| SK2 | Int |
| Zeitgl_1 | Timer |
| Zeitw_1 | S5Time |
| Zeitgl_2 | Timer |
| Zeitw_2 | S5Time |
| ZAE | Counter |
| Zaew | Word |
| RESET | Bool |

| Name | Datentyp |
|------|----------|
| OUT | |
| _1Y1 | Bool |
| _1Y2 | Bool |
| _2Y1 | Bool |
| _2Y2 | Bool |
| Q1 | Bool |
| Q2 | Bool |

6. Realisierung: Aufruf im OB1:



10.7 Lösungen der Lernaufgaben Kapitel 7

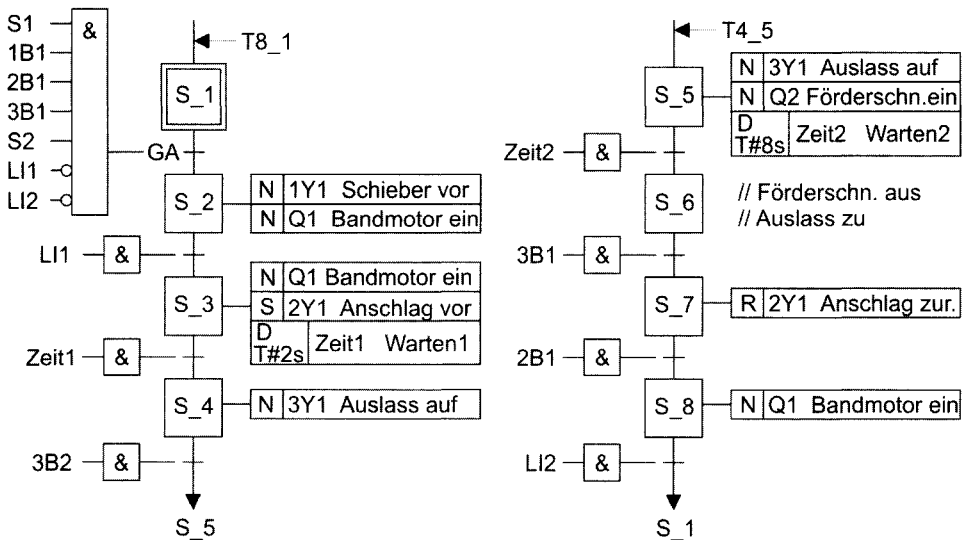
Lösung Lernaufgabe 7.1

Aufg. S. 104

1. Zuordnungstabelle der Anlage:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|------------------------------|--------|----------|---------------------------|---------|
| Sensor Behälter in Magazin | S1 | BOOL | Behälter vorhanden S1 = 1 | E 0.0 |
| Sensor 1A1 hintere Endlage | 1B1 | BOOL | 1A1 eingefahren 1B1 = 1 | E 0.1 |
| Sensor 2A1 hintere Endlage | 2B1 | BOOL | 2A1 eingefahren 2B1 = 1 | E 0.2 |
| Sensor 3A1 vordere Endlage | 3B1 | BOOL | 3A1 ausgefahren 3B1 = 1 | E 0.3 |
| Sensor 3A1 hintere Endlage | 3B2 | BOOL | 3A1 eingefahren 4B2 = 1 | E 0.4 |
| Sensor Pulver im Trichter | S2 | BOOL | Pulver vorhanden S2 = 1 | E 0.5 |
| Lichtschranke Abfüllposition | LI1 | BOOL | Position erreicht LI1 = 1 | E 0.6 |
| Lichtschranke Rollenbahn | LI2 | BOOL | Behälter passiert LI2 = 1 | E 0.7 |
| Ausgangsvariable | | | | |
| Magnetsp. Ventil Zyl 1A1 | 1Y1 | BOOL | Angezogen 1Y1 = 1 | A 4.1 |
| Magnetsp. Ventil Zyl 2A1 | 2Y1 | BOOL | Angezogen 2Y1 = 1 | A 4.2 |
| Magnetsp. Ventil Zyl 3A1 | 3Y1 | BOOL | Angezogen 3Y1 = 1 | A 4.3 |
| Schütz Bandmotor | Q1 | BOOL | Bandmotor läuft Q1 = 1 | A 4.4 |
| Schütz Förderschnecke | Q2 | BOOL | Förderschn. läuft Q2 = 1 | A 4.5 |

2. Ablauf-Funktionsplan:



3. Transitionstabelle:

| Transition | Transitionsbedingung |
|------------|-----------------------------|
| T-1 | T8_1 = LI2 |
| T-2 | T1_2 = GA (Grundst. Anlage) |
| T-3 | T2_3 = LI1 |
| T-4 | T3_4 = Zeit1 |

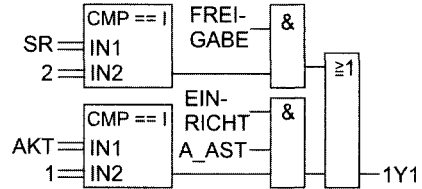
| Transition | Transitionsbedingung |
|------------|----------------------|
| T-5 | T4_5 = 3B2 |
| T-6 | T5_6 = Zeit2 |
| T-7 | T6_7 = 3B1 |
| T-8 | T7_8 = 2B1 |

4. Funktionsplan Aktionsausgabe:

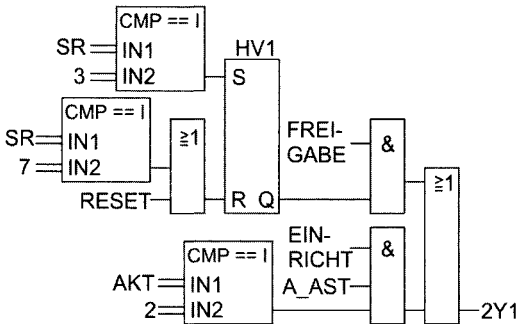
Maskierung Aktoranwahl

L #A ANW
 L W#16#F
 UW
 T #AKT

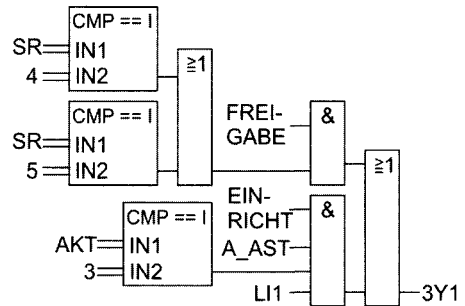
Ventil 1Y1 Zylinder 1A1



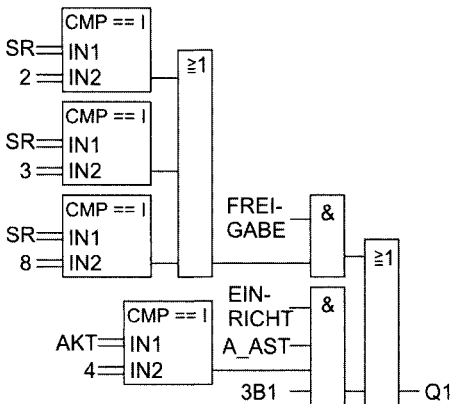
Ventil 2Y1 Zylinder 2A1



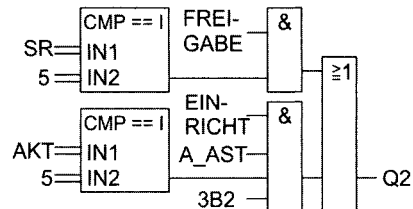
Ventil 3Y1 Zylinder 3A1



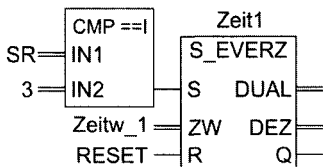
Schütz Q1 (Bandmotor)



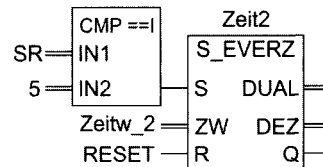
Schütz Q2 (Förderschnecke)



Wartezeit 1



Wartezeit 2



Da die Variable HV1 als statische Variablen zu deklarieren ist, muss der Funktionsplan des Ausgabebausteins in einem Funktionsbaustein programmiert werden.

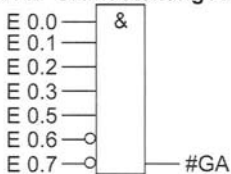
5. Deklarationstabelle Aktionsbaustein FB 26:

| Name | Datentyp | Anfangswert |
|----------|----------|-------------|
| IN | | |
| SR | Int | 0 |
| FREIGABE | Bool | FALSE |
| EINRICHT | Bool | FALSE |
| RESET | Bool | FALSE |
| A_ANW | Bool | FALSE |
| A_AST | Bool | FALSE |
| _3B1 | Bool | FALSE |
| _3B2 | Bool | FALSE |
| LII | Bool | FALSE |
| Zeit1 | Timer | |
| Zeitw_1 | S5Time | S5T#0MS |

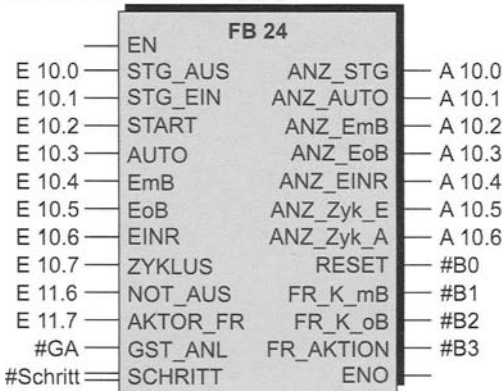
| Name | Datentyp | Anfangswert |
|---------|----------|-------------|
| Zeit2 | Timer | |
| Zeitw_2 | S5Time | S5T#0MS |
| OUT | | |
| _1Y1 | Bool | FALSE |
| _2Y1 | Bool | FALSE |
| _3Y1 | Bool | FALSE |
| Q1 | Bool | FALSE |
| Q2 | Bool | FALSE |
| STAT | | |
| HV1 | Bool | FALSE |
| TEMP | | |
| AKT | Int | |

6. Realisierung:

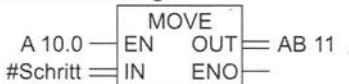
NW1: Grundstellung Anlage



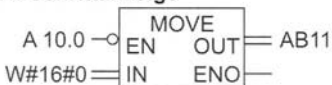
NW2: Betriebsarten DB 24



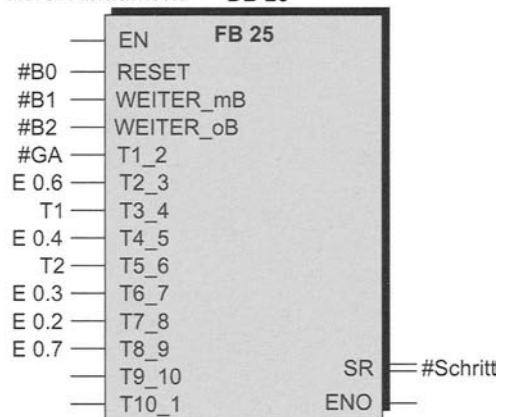
NW3: Schrittanzeige



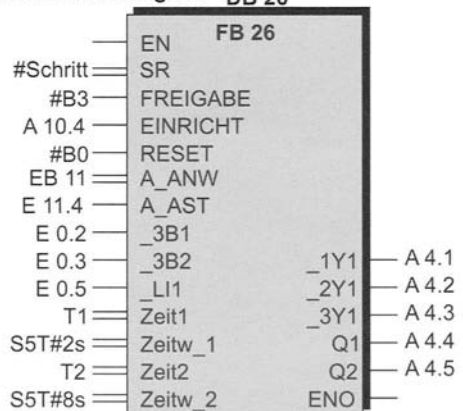
NW4: Schrittanzeige



NW5: Ablaufkette DB 25



NW6: Befehlsausgabe DB 26



Lösung Lernaufgabe 7.2

Aufg. S. 105

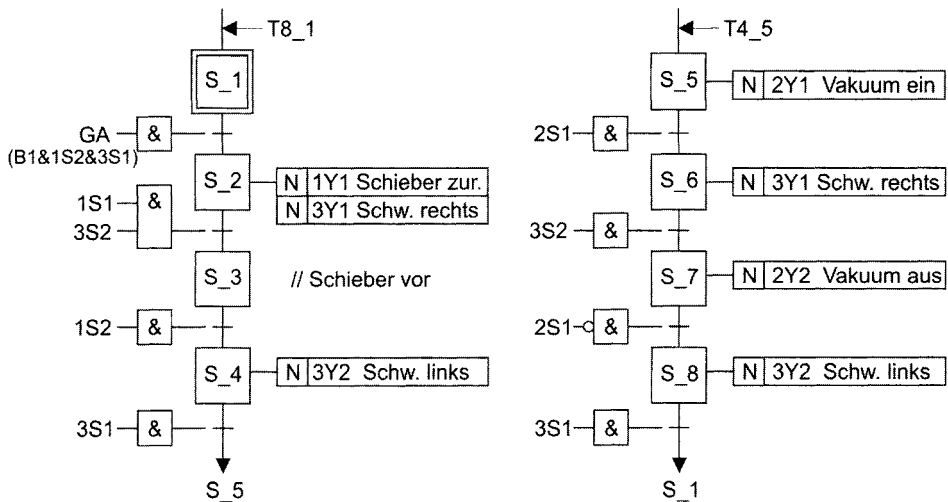
1. Zuordnungstabellen

Eingänge und Ausgänge des Anzeige- und Bedienfeldes: (Siehe Lösung Lernaufgabe 7.1, jedoch ohne Taster Steuerung AUS E0 und anstatt EIN-Taster E1 → Schalter EIN/AUS).

Eingänge und Ausgänge der Zubringeinrichtung:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | | Adresse |
|------------------------------|--------|----------|--------------------|---------|---------|
| Endlage Zylinder 1A hinten | 1S1 | BOOL | Betätigt | 1S1 = 1 | E 0.0 |
| Endlage Zylinder 1A vorne | 1S2 | BOOL | Betätigt | 1S2 = 1 | E 0.1 |
| Unterdruckschalter | 2S1 | BOOL | Betätigt | 2S1 = 1 | E 0.2 |
| Endlage Zylinder 3A Magazin | 3S1 | BOOL | Betätigt | 3S1 = 1 | E 0.3 |
| Endlage Zylinder 3A Band | 3S2 | BOOL | Betätigt | 3S2 = 1 | E 0.4 |
| Sensor Verpackung vorhanden | B1 | BOOL | Betätigt | B1 = 1 | E 0.5 |
| Ausgangsvariable | | | | | |
| Magnetsp. Ventil Zylinder 1A | 1Y1 | BOOL | Angezogen | 1Y1 = 1 | A 4.1 |
| Magnetsp. Ventil Druckaufbau | 2Y1 | BOOL | Angezogen | 2Y1 = 1 | A 4.2 |
| Magnetsp. Ventil Druckabbau | 2Y2 | BOOL | Angezogen | 2Y2 = 1 | A 4.3 |
| Magnetsp. Schwenkarm rechts. | 3Y1 | BOOL | Angezogen | 3Y1 = 1 | A 4.4 |
| Magnetsp. Schwenkarm links | 3Y2 | BOOL | Angezogen | 3Y2 = 1 | A 4.5 |

2. Ablauf-Funktionsplan:



3. Transitionstabelle:

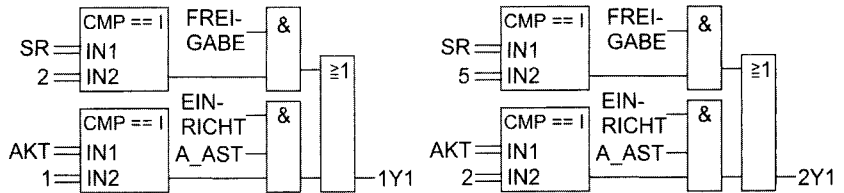
| Transition | Transitionsbedingung |
|------------|----------------------------|
| T-1 | T8_1 = 3S1 |
| T-2 | T1_2 = GA (B1 & 1S2 & 3S1) |
| T-3 | T2_3 = 1S1 & 3S2 |
| T-4 | T3_4 = 1S2 |

| Transition | Transitionsbedingung |
|------------|-------------------------|
| T-5 | T4_5 = 3S1 |
| T-6 | T5_6 = 2S1 |
| T-7 | T6_7 = 3S2 |
| T-8 | T7_8 = $\overline{2S1}$ |

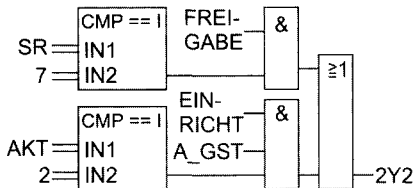
4. Funktionsplan Aktionsausgabe:

Mask. Aktorw. 1Y1 Magnetsp. Ventil Zylinder 1A 2Y1 Magnetsp. Ventil Druckaufbau

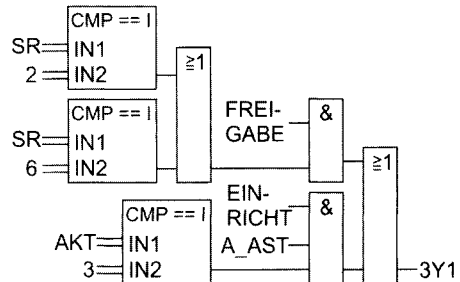
L #A_ANW
L W#16#F
UW
T #AKT



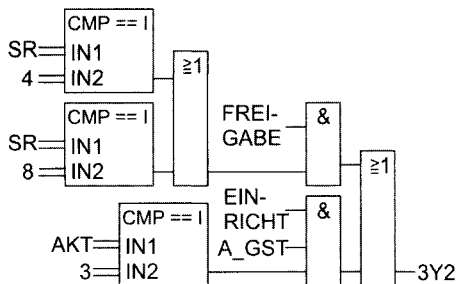
2Y2 Magnetsp. Ventil Druckabbau



3Y1 Magnetsp. Schwenkarm rechts



3Y2 Magnetsp. Schwenkarm links



Da keine statischen Variablen erforderlich sind, kann der Funktionsplan des Aktionsbausteins in einer Funktion FC 26 umgesetzt werden.

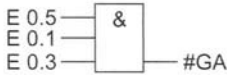
5. Deklarationstabelle Aktionsbaustein FC 26:

| Name | Datentyp |
|----------|----------|
| IN | |
| SR | Int |
| FREIGABE | Bool |
| EINRICHT | Bool |
| RESET | Bool |
| A_ANW | Byte |
| A_AST | Bool |
| A_GST | Bool |

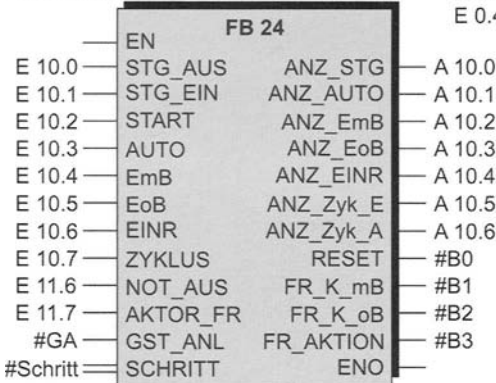
| Name | Datentyp |
|------|----------|
| OUT | |
| _1Y1 | Bool |
| _2Y1 | Bool |
| _2Y2 | Bool |
| _3Y1 | Bool |
| _3Y2 | Bool |
| TEMP | |
| AKT | Int |

6. Realisierung:

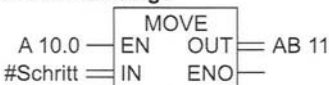
NW1: Grundstellung Anlage



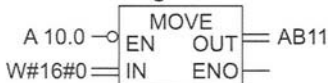
NW2: Betriebsarten DB 24



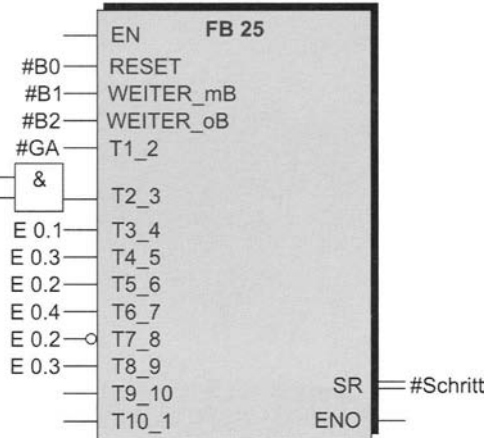
NW3: Schrittanzeige



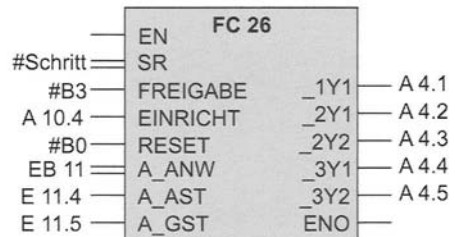
NW4: Schrittanzeige



NW5: Ablaufkette DB 25



NW6: Befehlsausgabe



Programm s. beiliegende CD od. <http://automatisieren-mit-sps.de> STEP7-Lösungen → 7_2_02

Lösung Lernaufgabe 7.3

Aufg. S. 106

1. Zuordnungstabellen:

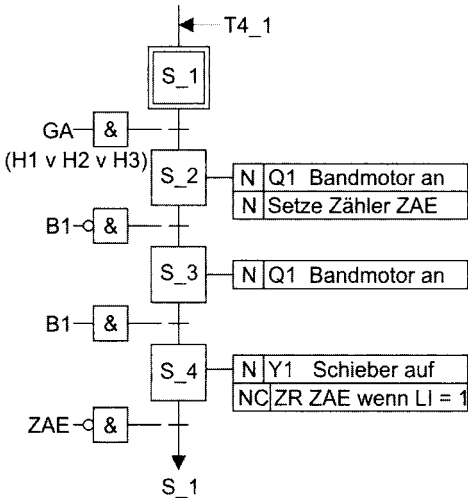
Eingänge und Ausgänge des Anzeige- und Bedienfeldes:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | | Adresse |
|-------------------------|--------|----------|--------------------|--------|---------|
| ... // Standardeinträge | ... | BOOL | ... | ... | ... |
| Taster Vorwahl N1 | S1 | BOOL | Betätigt | S1 = 1 | E 1.1 |
| Taster Vorwahl N2 | S2 | BOOL | Betätigt | S2 = 1 | E 1.2 |
| Taster Vorwahl N3 | S3 | BOOL | Betätigt | S3 = 1 | E 1.3 |
| Ausgangsvariable | | | | | |
| ... // Standardeinträge | ... | ... | ... | ... | ... |
| Anzeige Vorwahl N1 | H1 | BOOL | Anzeige an | H1 = 1 | A 5.1 |
| Anzeige Vorwahl N1 | H2 | BOOL | Anzeige an | H2 = 1 | A 5.2 |
| Anzeige Vorwahl N1 | H3 | BOOL | Anzeige an | H3 = 1 | A 5.3 |

Eingänge und Ausgänge der Tablettenabfülleinrichtung:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | | Adresse |
|------------------------------|--------|----------|--------------------|----------|---------|
| Sensor Röhrrchen vorhanden | B1 | BOOL | Betätigt | $B1 = 1$ | E 0.0 |
| Lichtschranke | LI | BOOL | Betätigt | $LI = 1$ | E 0.1 |
| Ausgangsvariable | | | | | |
| Schütz Bandmotor | Q1 | BOOL | Motor an | $Q1 = 1$ | A 4.0 |
| Elektromagnetischer Schieber | Y1 | BOOL | Angezogen | $Y1 = 1$ | A 4.1 |

2. Ablauf-Funktionsplan:



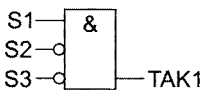
3. Transitionstabelle:

| Transition | Transitionsbedingung |
|------------|---------------------------------------|
| T-1 | $T4_1 = \overline{ZAE}$ |
| T-2 | $T1_2 = GA (H1 \vee H2 \vee H3 = 1)$ |
| T-3 | $T2_3 = \overline{B1}$ |
| T-4 | $T3_4 = B1$ |

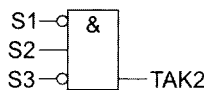
4. Funktionsplan Aktionsausgabe:

Zusätzlich zur Aktionsausgabe muss in diesem Baustein noch die Vorwahl der Tablettenanzahl durch S1, S2 bzw. S3, die zugehörigen Anzeigen H1, H2 bzw. H3 und das Setzen bzw. Rückwärtszählen des Zählers ZAE programmiert werden.

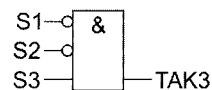
Vorwahl Tablettenzahl N1



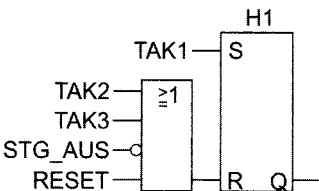
Vorwahl Tablettenzahl N2



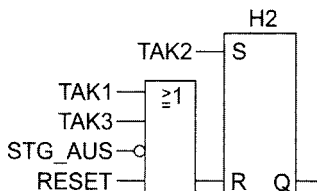
Vorwahl Tablettenzahl N3



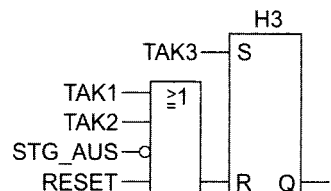
Anzeige N1



Anzeige N2

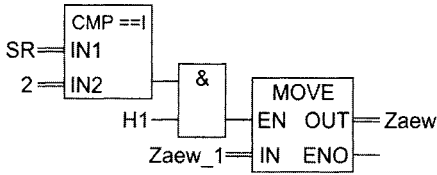


Anzeige N3

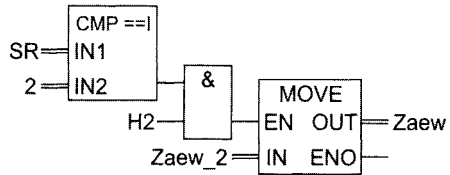


Hinweis: Die Abkürzung TAK steht für Tastenkombination

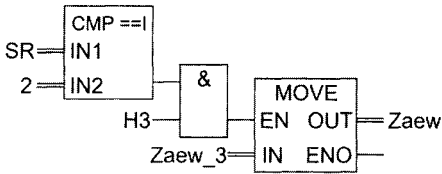
Zählwert „Zaew“ mit N1 laden



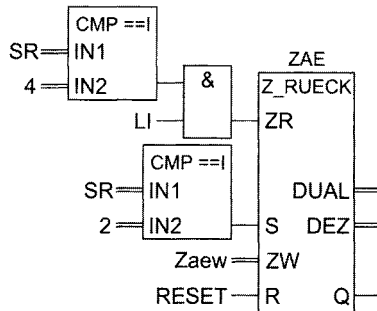
Zählwert „Zaew“ mit N2 laden



Zählwert „Zaew“ mit N3 laden



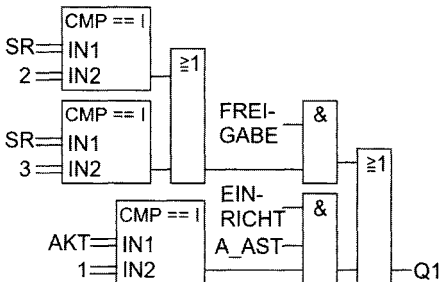
Zähler „Setzen“ und „Rückwärtszählen“



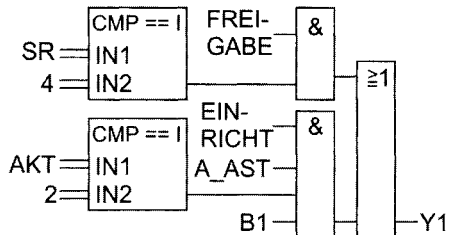
Maskierung Aktoranwahl

- L #A ANW
- L W#16#F
- UW
- T #AKT

Schütz Q1 für Bandmotor



Schieber Y1



5. Deklarationstabelle Aktionsbaustein FC 26:

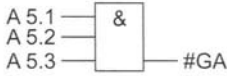
| Name | Datentyp |
|----------|----------|
| IN | |
| SR | Int |
| FREIGABE | Bool |
| EINRICHT | Bool |
| RESET | Bool |
| A_ANW | Bool |
| A_AST | Bool |
| STG_AUS | Bool |
| S1 | Bool |
| S2 | Bool |
| S3 | Bool |

| Name | Datentyp |
|--------|----------|
| B1 | Bool |
| LI | Bool |
| ZAE | Counter |
| Zaew_1 | Word |
| Zaew_2 | Word |
| Zaew_3 | Word |
| OUT | |
| Q1 | Bool |
| Y1 | Bool |

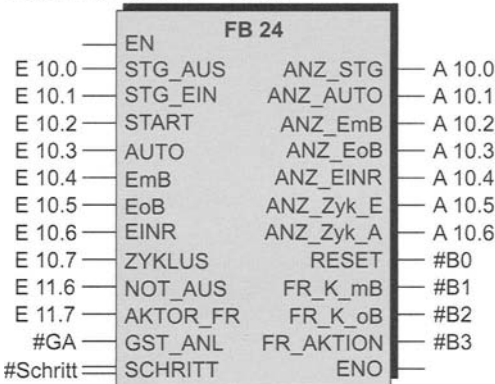
| Name | Datentyp |
|--------|----------|
| IN_OUT | |
| H1 | Bool |
| H2 | Bool |
| H3 | Bool |
| TEMP | |
| TAK1 | Bool |
| TAK2 | Bool |
| TAK3 | Bool |
| Zaew | Word |
| AKT | Int |

6. Realisierung:

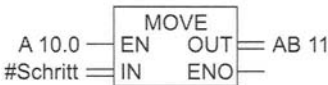
NW1: Grundstellung Anlage



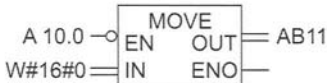
NW2: Betriebsarten DB 24



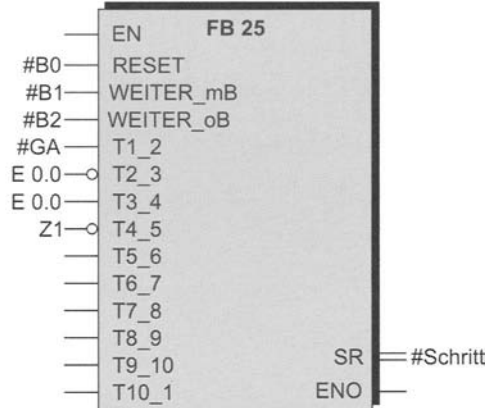
NW3: Schrittanzeige



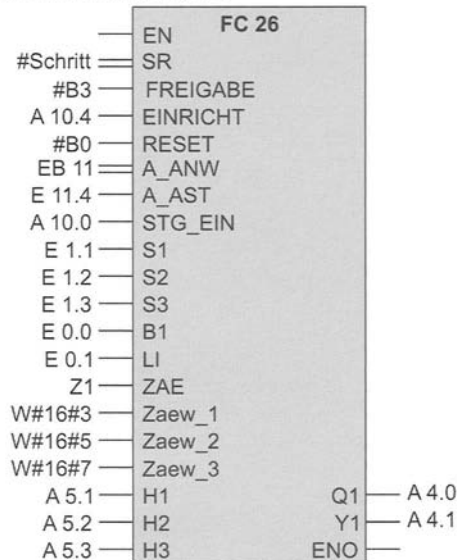
NW4: Schrittanzeige



NW5: Ablaufkette DB 25



NW6: Befehlsausgabe



Programm s. beiliegende CD od. <http://automatisieren-mit-sps.de> STEP7-Lösungen → 7_2_03

Lösung Lernaufgabe 7.4

Aufg. S. 107

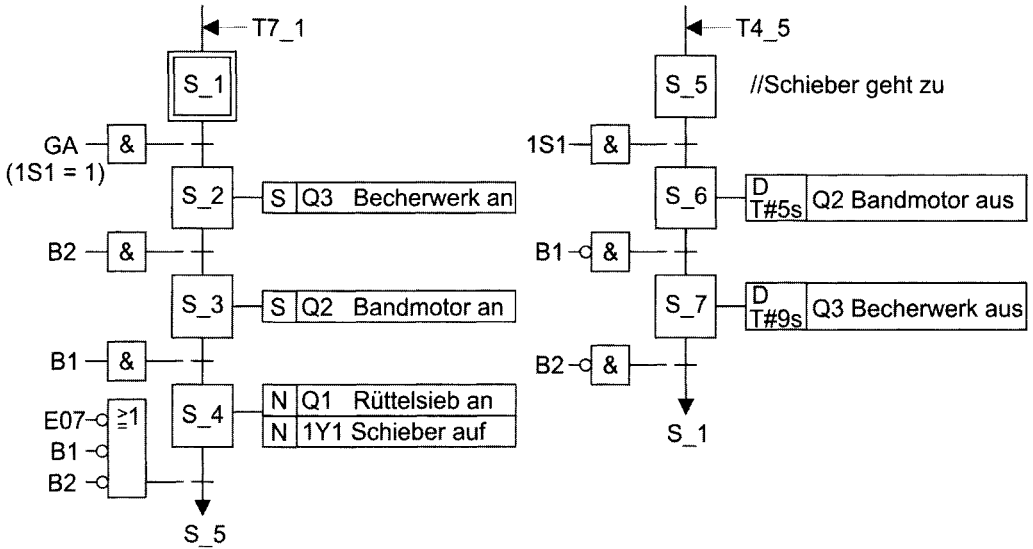
1. Zuordnungstabelle:

Eingänge und Ausgänge des Schotterwerks:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse | |
|---------------------------|--------|----------|--------------------|---------|-------|
| Endlage Schieber 1A vorne | IS1 | BOOL | Betätigt | IS1 = 1 | E 0.1 |
| Drehzahlsensor Förderband | B1 | BOOL | Betätigt | B1 = 1 | E 0.2 |
| Drehzahlsensor Becherwerk | B2 | BOOL | Betätigt | B2 = 1 | E 0.3 |

| AusgangsvARIABLE | | | | | |
|-------------------------------|-----|------|-------------|---------|-------|
| Elektromagnet Ventil Schieber | 1Y1 | BOOL | Angezogen | 1Y1 = 1 | A 4.0 |
| Schütz Motor M1 (Rüttelsieb) | Q1 | BOOL | Motor M1 an | Q1 = 1 | A 4.1 |
| Schütz Motor M2 (Bandmotor) | Q2 | BOOL | Motor M2 an | Q2 = 1 | A 4.2 |
| Schütz Motor M3 (Becherw.) | Q3 | BOOL | Motor M3 an | Q3 = 1 | A 4.3 |

2. Ablauf-Funktionsplan:



3. Transitionstabelle:

| Transition | Transitionsbedingung |
|------------|-------------------------|
| T-1 | $T7_8 = \overline{B2}$ |
| T-2 | $T1_2 = GA (1S1 = 1)$ |
| T-3 | $T2_3 = B2$ |
| T-4 | $T3_4 = B1$ |

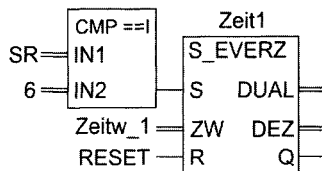
| Transition | Transitionsbedingung |
|------------|--|
| T-5 | $T4_5 = \overline{E07} \vee \overline{B1} \vee \overline{B2}$ |
| T-6 | $T5_6 = 1S1$ |
| T-7 | $T6_7 = \overline{B1}$ |

4. Funktionsplan Aktionsausgabe:

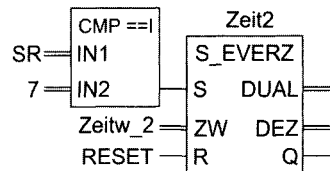
Maskierung
Aktoranwahl

L #A ANW
L W#16#F
UW
T #AKT

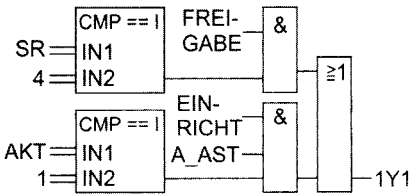
Ansteuerung Zeitglied 1



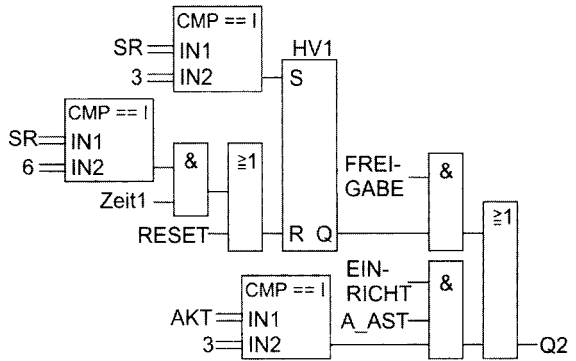
Ansteuerung Zeitglied 2



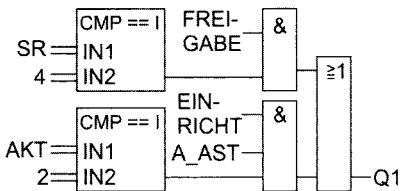
Ansteuerung des Aktors 1Y1



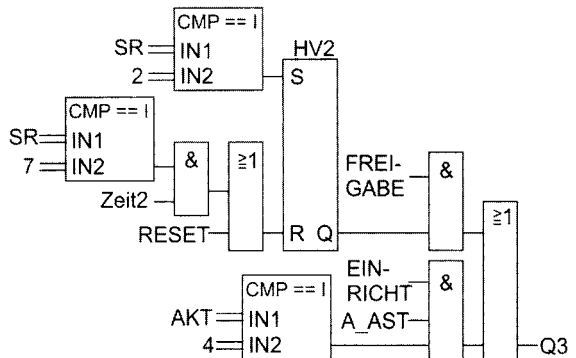
Ansteuerung des Aktors Q2



Ansteuerung des Aktors Q1



Ansteuerung des Aktors Q3



5. Deklarationstabelle Aktionsbaustein FB 26:

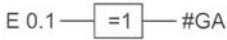
Da die beiden Variablen HV1 und HV2 als statische Variablen zu deklarieren sind, muss der Funktionsplan des Ausgabebausteins in einem Funktionsbaustein programmiert werden.

| Name | Datentyp | Anfangswert |
|-----------|----------|-------------|
| IN | | |
| SR | Int | 0 |
| FREIGABE | Bool | FALSE |
| EINRICHT | Bool | FALSE |
| RESET | Bool | FALSE |
| A_ANW | Bool | FALSE |
| A_AST | Bool | FALSE |
| Zeit1 | Timer | |
| Zeitw_1 | S5Time | S5T#0MS |
| Zeit2 | Timer | |
| Zeitw_2 | S5Time | S5T#0MS |

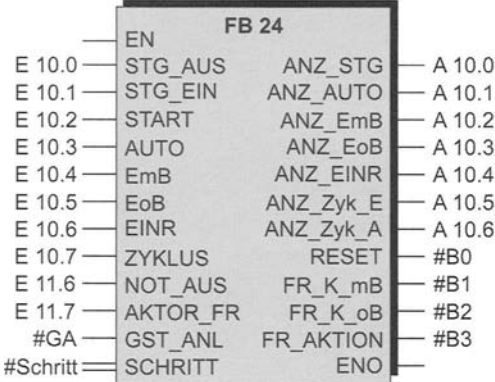
| Name | Datentyp | Anfangswert |
|-------------|----------|-------------|
| OUT | | |
| _1Y1 | Bool | FALSE |
| Q1 | Bool | FALSE |
| Q2 | Bool | FALSE |
| Q3 | Bool | FALSE |
| STAT | | |
| HV1 | Bool | FALSE |
| HV2 | Bool | FALSE |
| TEMP | | |
| AKT | Int | |

6. Realisierung:

NW1: Grundstellung Anlage



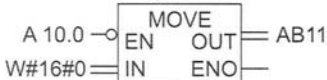
NW2: Betriebsarten DB 24



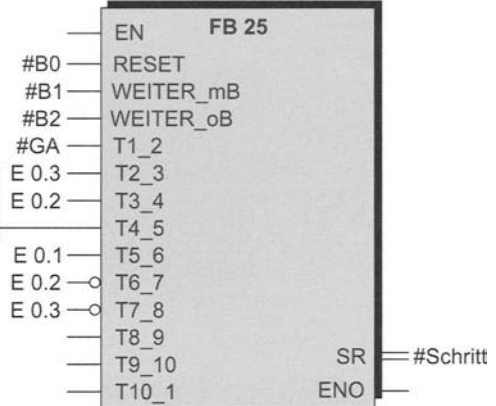
NW3: Schrittanzeige



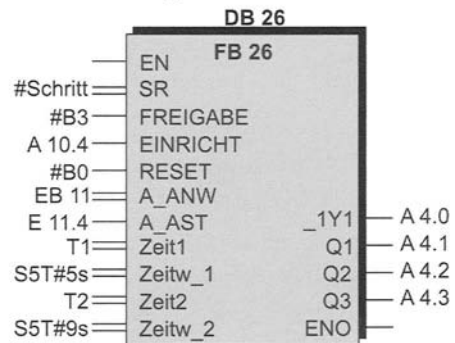
NW4: Schrittanzeige



NW5: Ablaufkette DB 25



NW6: Befehlsausgabe



Programm s. beiliegende CD od. <http://automatisieren-mit-sps.de> STEP7-Lösungen → 7_2_04

Lösung Lernaufgabe 7.5 Aufg. S. 108

1. Zuordnungstabellen.

Eingänge und Ausgänge des Anzeige- und Bedienfeldes:

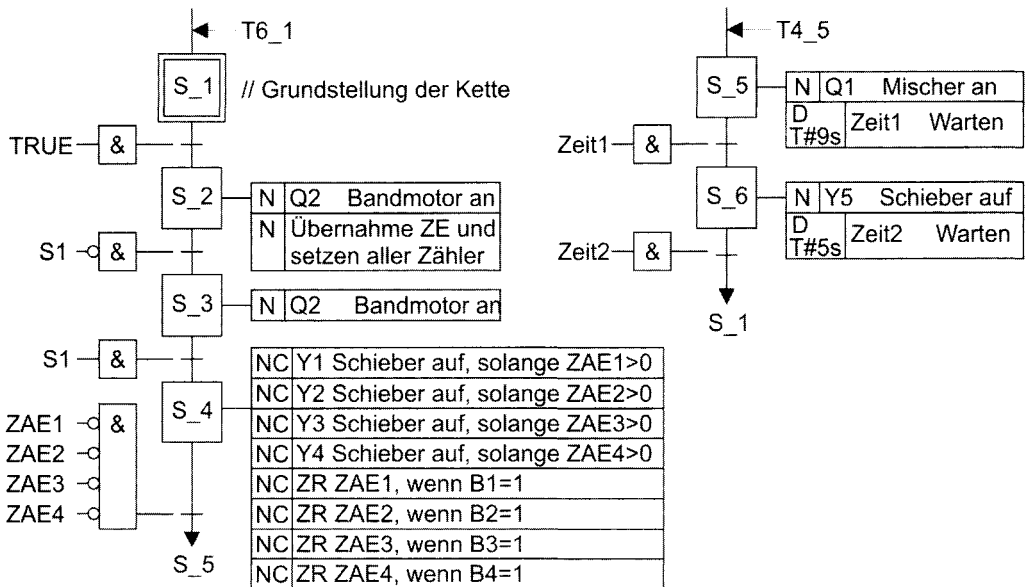
| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|---------------------------|--------|----------|--------------------|---------|
| // Standardeinträge | ... | BOOL | ... | ... |
| Zifferneinsteller Vorwahl | ZE | BOOL | Zahl 0 - 9 | EB2 |
| Ausgangsvariable | | | | |
| ... // Standardeinträge | ... | ... | ... | ... |

Eingänge und Ausgänge der Los-Verpackungsanlage:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|------------------------|--------|----------|--------------------|---------|
| Zählsensor Nieten | B1 | BOOL | Betätigt B1 = 1 | E 0.1 |
| Zählsensor Trostpreise | B2 | BOOL | Betätigt B2 = 1 | E 0.2 |

| | | | | | |
|-------------------------------|----|------|---------------|--------|-------|
| Zählsensor Gewinne | B3 | BOOL | Betätigt | B3 = 1 | E 0.3 |
| Zählsensor Hauptgewinne | B4 | BOOL | Betätigt | B4 = 1 | E 0.4 |
| Sensor Lostüte vorhanden | S1 | BOOL | Betätigt | S1 = 1 | E 0.5 |
| AusgangsvARIABLE | | | | | |
| Zufuhrschieber Silo Nieten | Y1 | BOOL | Silo offen | Y1 = 1 | A 4.1 |
| Zufuhrschieber Silo Trostpr. | Y2 | BOOL | Silo offen | Y2 = 1 | A 4.2 |
| Zufuhrschieber Silo Gewinne | Y3 | BOOL | Silo offen | Y3 = 1 | A 4.3 |
| Zufuhrschieber Silo Hauptgew. | Y4 | BOOL | Silo offen | Y4 = 1 | A 4.4 |
| Abfüllschieber | Y5 | BOOL | Mischer offen | Y5 = 1 | A 4.5 |
| Schütz Mischermotor | Q1 | BOOL | Motor an | Q1 = 1 | A 4.6 |
| Schütz Bandmotor | Q2 | BOOL | Motor an | Q2 = 1 | A 4.7 |

2. Ablauf-Funktionsplan:



3. Transitionstabelle:

| Transition | Transitionsbedingung |
|------------|----------------------------------|
| T-1 | T6_1 =Zeitgl_2 |
| T-2 | T1_2 = TRUE *) |
| T-3 | T2_3 = S1 |
| T-4 | T3_4 = S1 |
| T-5 | T4_5 = ZAE1 & ZAE2 & ZAE3 & ZAE4 |
| T-6 | T5_6 =Zeitgl_1 |

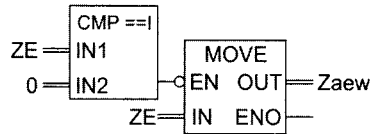
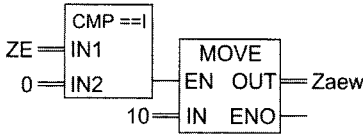
*) Hinweis:

Da die Geber der Anlage keine sinnvolle Grundstellung liefern, wird die Weiterschaltung von Schritt S_1 nach Schritt S_2 unbedingt (TRUE) nach Betätigung der Start-Taste E02 durchgeführt.

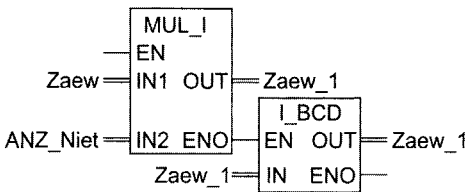
4. Funktionsplan Aktionsausgabe:

Neben der Aktionsausgabe muss in diesem Baustein noch die Gesamtzahl der Lose und das Setzen bzw. Rückwärtszählen der Zähler ZAE1 bis ZAE4 programmiert werden.

Ist am Zifferneinsteller ZE der Wert 0 eingestellt, wird der Faktor 10 in den Multiplikator Zaew geladen. Ansonsten wird der Wert des Zifferneinstellers in den Multiplikator geladen.

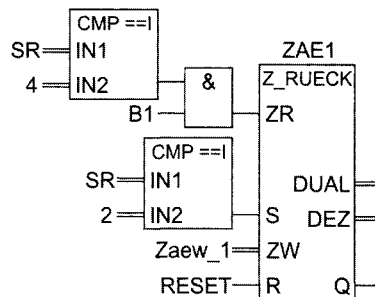


Multiplikation Zaew mit der vorgegebenen Anzahl Nieten pro hundert Lose und Umwandlung in eine BCD-Zahl Zaew_1.



Diese Funktion muss für die drei restlichen Vorgaben (Anzahl Trostpreise, Gewinne und Hauptgewinne) wiederholt werden. Als Ergebnis erhält man die drei BCD-Werte: Zaew_2, Zaew_3, und Zaew_4, mit denen die entsprechenden Zähler gesetzt werden.

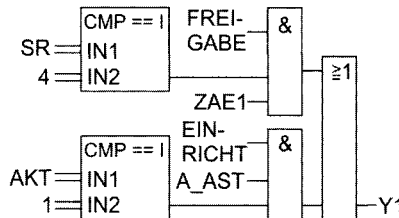
Aktion S_2: Setzen des Zählers auf die BCD-Zahl Zaew_1 und Aktion S_4: Rückwärtszählen, wenn Zähler B1 „1“-Signal meldet.



Zur Zählung der weiteren Losarten müssen noch weiter drei Zählerfunktionen ZAE2, ZAE3 und ZAE4 entsprechend des vorgegebenen Funktionsplanes programmiert werden.

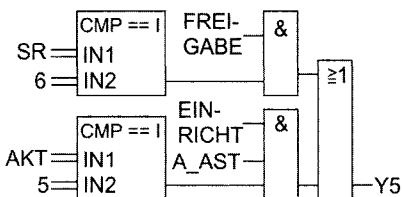
Maskierung Aktoranwahl Ansteuerung Zufuhrschieber

- L #A ANW
- L W#16#F
- UW
- T #AKT

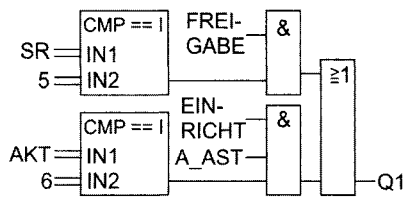


Diese Funktion muss für die drei weiteren Zufuhrschieber Y2, Y3 und Y4 entsprechend programmiert werden

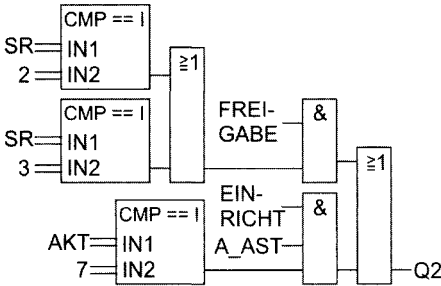
Ansteuerung Y5 Abfüllschieber



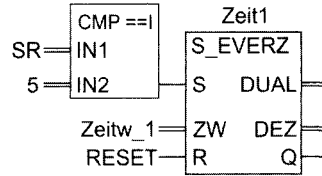
Ansteuerung Q1 Mischeromotor



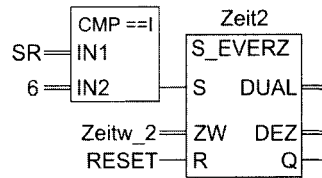
Ansteuerung Q2 Bandmotor



Zeit 1 Warten Schritt S_5



Zeit 2 Warten Schritt S_6



5. Deklarationstabelle Aktionsbaustein FC 26:

| Name | Datentyp |
|-----------|----------|
| IN | |
| SR | Int |
| FREIGABE | Bool |
| EINRICHT | Bool |
| RESET | BOOL |
| A_ANW | Byte |
| A_AST | Bool |
| B1 | Bool |
| B2 | Bool |
| B3 | Bool |
| B4 | Bool |
| ZE | Byte |
| ZAE1 | Counter |
| ZAE2 | Counter |
| ZAE3 | Counter |
| ZAE4 | Counter |

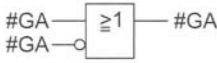
| Name | Datentyp |
|-----------|----------|
| IN | |
| ANZ_Niet | Int |
| ANZ_Tros | Int |
| ANZ_Gew | Int |
| ANZ_HGe | Int |
| Zeit1 | Timer |
| Zeitw_1 | S5Time |
| Zeit2 | Timer |
| Zeitw_2 | S5Time |

| Name | Datentyp |
|-------------|----------|
| OUT | |
| Y1 | Bool |
| Y2 | Bool |
| Y3 | Bool |
| Y4 | Bool |
| Y5 | Bool |
| Q1 | Bool |
| Q2 | Bool |
| Name | |
| TEMP | |
| Zaew_1 | Word |
| Zaew_2 | Word |
| Zaew_3 | Word |
| Zaew_4 | Word |
| Zaew | Int |
| AKT | Int |

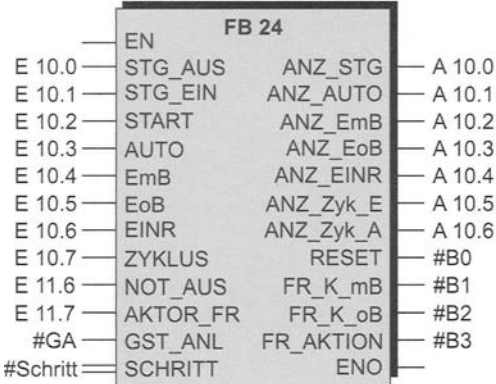
Hinweis: Um die Anzahl der Variablen und die Anzahl der Netzwerke klein zu halten, ist bei der Eingabe des Funktionsplanes für den Aktionsbaustein FC 26 die Formatüberprüfung abzuschalten. So ist es beispielsweise ohne Formatwandlung möglich, die Variable ZE (Byte) an den Vergleichsbaustein oder die Variable Zaew_1 (Word) an den Multiplikationsbaustein zu schreiben, die eigentlich den Datentyp Integer erfordern.

6. Realisierung:

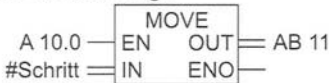
NW1: Grundstellung Anlage



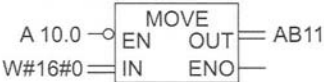
NW2: Betriebsarten DB 24



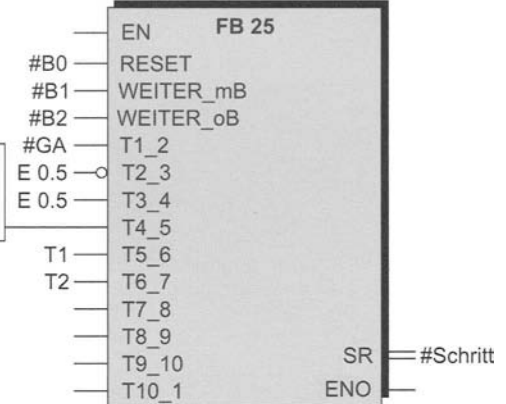
NW3: Schrittanzeige



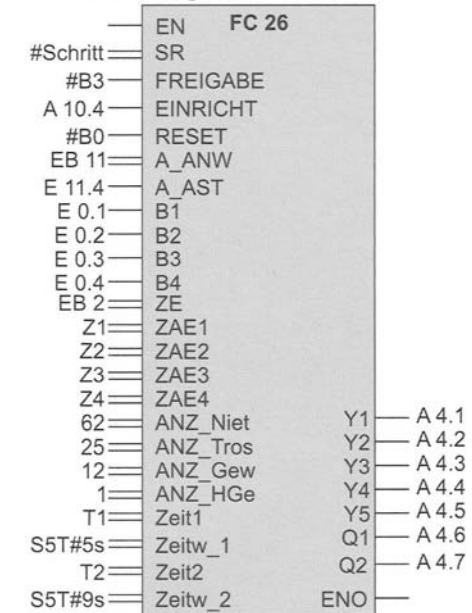
NW4: Schrittanzeige



NW5: Ablaufkette DB 25



NW6: Befehlsausgabe



Programm s. beiliegende CD od. <http://automatisieren-mit-sps.de> STEP7-Lösungen → 7_2_05

Lösung Lernaufgabe 7.6

Aufg. S. 110

1. Zuordnungstabellen:

Eingänge und Ausgänge des Anzeige- und Bedienfeldes:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|-------------------------|--------|----------|-------------------------|---------|
| Taster Steuerung AUS | E0 | BOOL | Betätigt E0 = 0 | E 0.0 |
| Taster Steuerung EIN | E1 | BOOL | Betätigt E1 = 1 | E 0.1 |
| Schalter Automatik/Hand | E2 | BOOL | Automatikbetrieb E2 = 1 | E 0.2 |

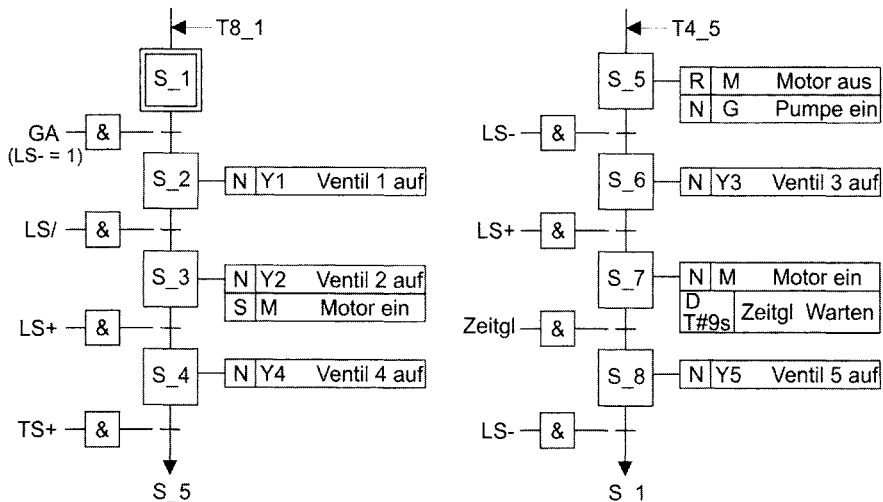
| | | | | | |
|-----------------------------|----|------|---------------------------|--------|-------|
| Taster Start/Einzelschritt | E3 | BOOL | Betätigt | E3 = 1 | E 0.3 |
| Taster Beenden_Autom. | E4 | BOOL | Betätigt | E4 = 1 | E 0.4 |
| Schalter W_Beding_Hand | E5 | BOOL | Weiter_oB | E5 = 1 | E 0.5 |
| Taster Aktion_Freigabe_Hand | E6 | BOOL | Aktion frei | E6 = 1 | E 0.6 |
| NOT_AUS | E7 | BOOL | Betätigt | E7 = 0 | E 0.7 |
| Ausgangsvariable | | | | | |
| Schrittanzeige Wert 1 | A0 | BOOL | Anzeige 2 ⁰ an | A0 = 1 | A 4.0 |
| Schrittanzeige Wert 2 | A1 | BOOL | Anzeige 2 ¹ an | A1 = 1 | A 4.1 |
| Schrittanzeige Wert 4 | A2 | BOOL | Anzeige 2 ² an | A2 = 1 | A 4.2 |
| Schrittanzeige Wert 8 | A3 | BOOL | Anzeige 2 ³ an | A3 = 1 | A 4.3 |
| Anz. Steuerung | A4 | BOOL | Anzeige ein | A4 = 1 | A 4.4 |
| Anz. Automatikbetrieb | A5 | BOOL | Anzeige ein | A5 = 1 | A 4.5 |

Hinweis: Der NOT-AUS Taster befindet sich nicht auf dem Bedienfeld.

Eingänge und Ausgänge des Rührkessels:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|------------------------------|--------|----------|---------------------------|---------|
| Niveauschalter Beh. leer | LS- | BOOL | Behälter leer LS- = 1 | E 1.0 |
| Niveauschalter Beh. halbvoll | LS/ | BOOL | Behälter halbvoll LS/ = 1 | E 1.1 |
| Niveauschalter Beh. voll | LS+ | BOOL | Behälter voll LS+ = 1 | E 1.2 |
| Temperaturgrenzschalter | TS+ | BOOL | Temp. erreicht TS+ = 1 | E 1.3 |
| Ausgangsvariable | | | | |
| Zulaufventil 1 | Y1 | BOOL | Offen Y1 = 1 | A 5.1 |
| Zulaufventil 2 | Y2 | BOOL | Offen Y2 = 1 | A 5.2 |
| Reinigungsmittelventil | Y3 | BOOL | Offen Y3 = 1 | A 5.3 |
| Heißdampfventil | Y4 | BOOL | Offen Y4 = 1 | A 5.4 |
| Ablassventil | Y5 | BOOL | Offen Y5 = 1 | A 5.5 |
| Motor Rührwerk | M | BOOL | Rührwerk an M = 1 | A 5.6 |
| Pumpe | G | BOOL | Pumpe an G = 1 | A 5.7 |

2. Ablauf-Funktionsplan:

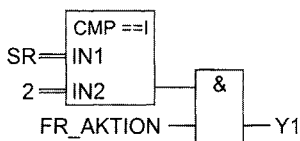


3. Transitionstabelle:

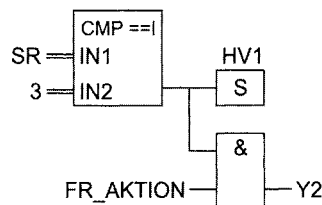
| Transition | Transitionsbedingung |
|------------|--|
| T-1 | T8_1 = LS- |
| T-2 | T1_2 = GA (Grundstellung der Anlage LS- = 1) |
| T-3 | T2_3 = LS/ |
| T-4 | T3_4 = LS+ |
| T-5 | T4_5 = TS+ |
| T-6 | T5_6 = LS- |
| T-7 | T6_7 = LS+ |
| T-8 | T7_8 = Zeitgl |

4. Funktionsplan Aktionsausgabe:

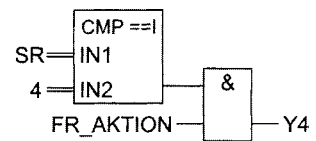
Ansteuerung Ventil 1



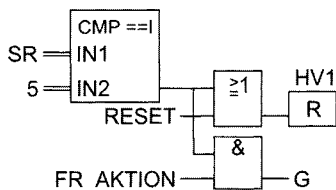
Einschalten Ventil 2



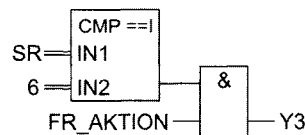
Ansteuerung Ventil 4



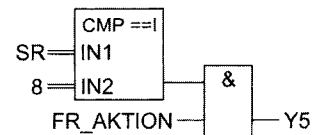
Ausschalten Ventil 1



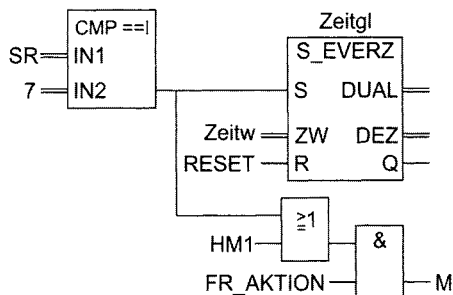
Einschalten Ventil 3



Einschalten Ventil 5



Ansteuerung Zeitfunktion



Wie aus dem Funktionsplan zu ersehen ist, muss für die Ansteuerung des Rührmotors M eine Hilfsvariable HV1 eingeführt werden. Da der Signalzustand dieser Variablen gespeichert wird, ist diese als lokale statische Variable zu deklarieren. Als Aktionsbaustein muss somit ein Funktionsbaustein FB verwendet werden.

5. Deklarationstabelle Aktionsbaustein FB 26:

| Name | Datentyp | Anfangswert |
|------------|----------|-------------|
| IN | | |
| SR | Int | 0 |
| FR_AKTION | Bool | FALSE |
| RESET | Bool | FALSE |
| Zeitgl | Timer | |
| Zeitw | S5Time | S5T#0MS |
| OUT | | |
| Y1 | Bool | FALSE |

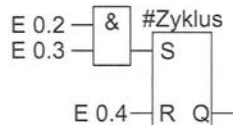
| Name | Datentyp | Anfangswert |
|-------------|----------|-------------|
| Y2 | Bool | FALSE |
| Y3 | Bool | FALSE |
| Y4 | Bool | FALSE |
| Y5 | Bool | FALSE |
| M | Bool | FALSE |
| G | Bool | FALSE |
| STAT | | |
| HV1 | Bool | FALSE |

6. Realisierung:

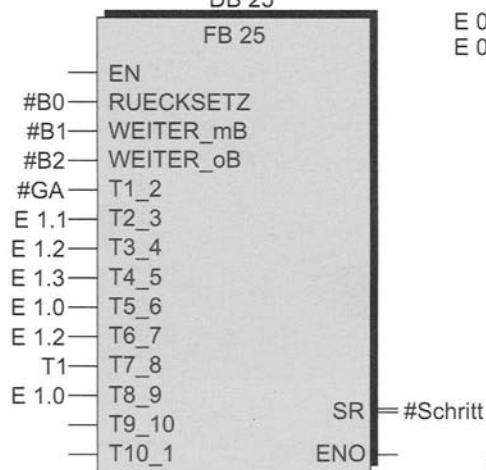
NW1: Grundstellung Anlage



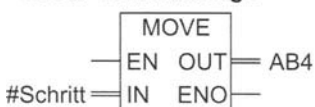
NW2: Zyklus EIN/AUS



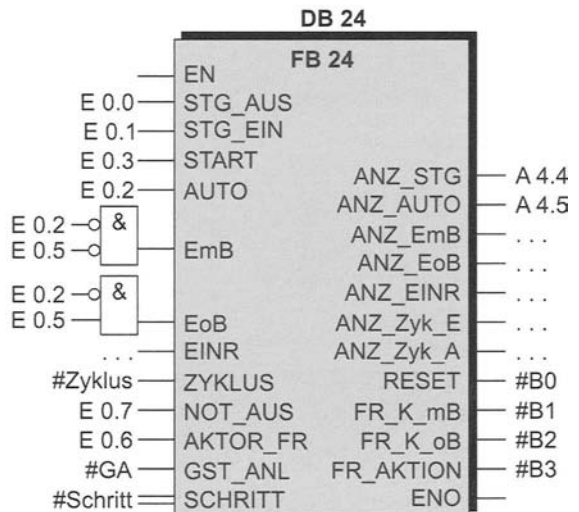
NW3: Ablaufkette



NW4: Schrittanzeige



NW5: Betriebsarten



NW6: Befehlsausgabe



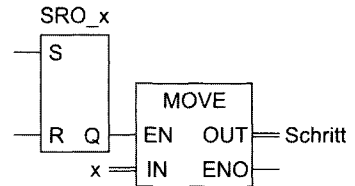
Programm siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de>
unter STEP7-Lösungen → 7_2_06

10.8 Lösungen der Lernaufgaben Kapitel 8

Hinweise: Bei den Lösungsvorschlägen der Aufgaben 8.1, 8.2, 8.3 und 8.4 wird auf die Darstellung des Funktionsplanes für die Ablaufkette, die Darstellung der Deklarationstabelle für den entsprechenden Funktionsbaustein sowie den Aufruf des Bausteins im OB1 aus Umfangsgründen des Buches verzichtet. Alle drei Elemente können Sie dem jeweiligen STEP7-Programm entnehmen, welches Sie auf der beiliegenden CD finden oder unter der Internetadresse <http://www.automatisieren-mit-sps.de> herunterladen können.

Bei der Umsetzung aller vorkommenden Ablauf-Funktionspläne in einen Funktionsbaustein wird der Baustein durch die Ausgangsvariable „Schritt“ erweitert. An diesem Ausgang kann dann der jeweils aktuelle Schritt, in welchem sich die Ablaufkette befindet, abgelesen werden.

Dies ist besonders bei der Erstinbetriebnahme des Programms wichtig, um schnell eventuelle Programmierfehler zu finden. Der nebenstehende Funktionsplan zeigt, wie mit Hilfe der MOVE-Funktion bei jedem Schrittoperanden SRO_x der Variablen „Schritt“ die Schrittnummer zugewiesen wird.



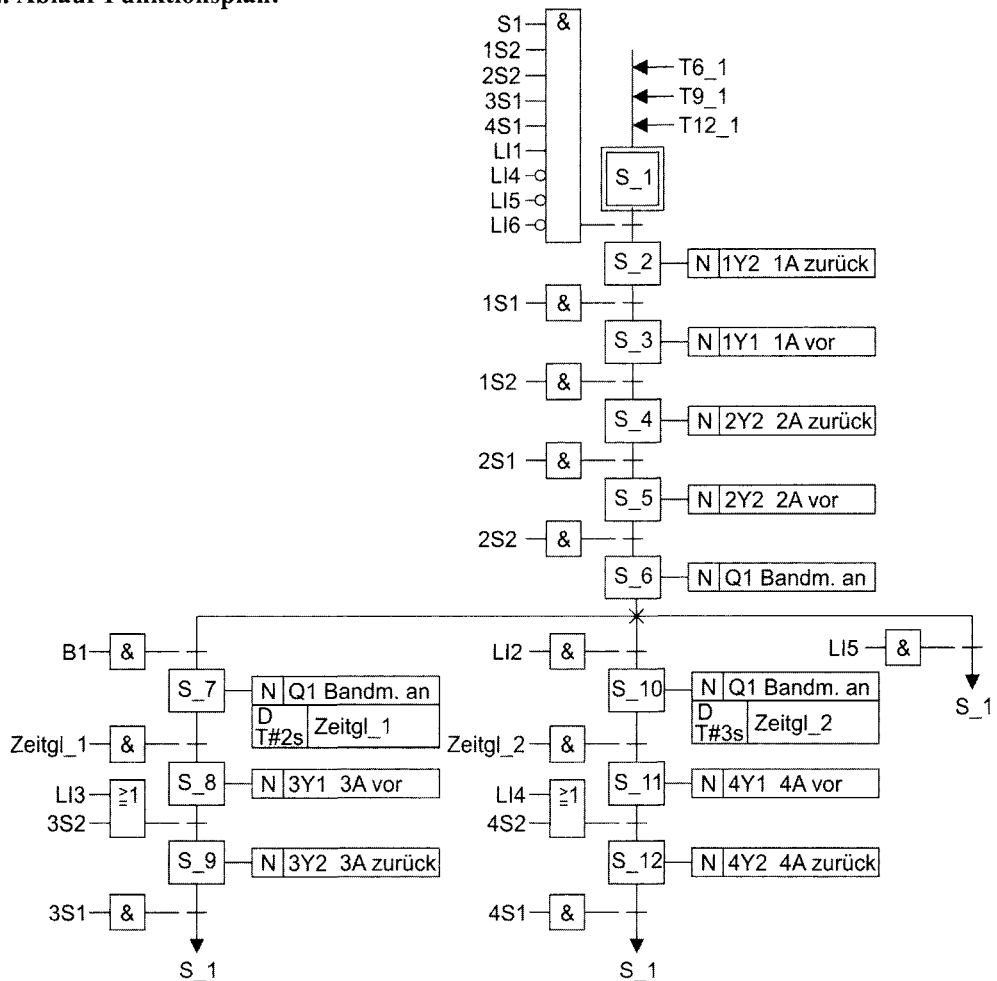
Lösung Lernaufgabe 8.1

Aufg. S. 122

1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|-------------------------------|--------|----------|--------------------------|---------|
| Schlüsselschalter START | S1 | BOOL | Eingeschaltet S1 = 1 | E 0.1 |
| Endsch. Schieber 1A hinten | 1S1 | BOOL | Hintere Endlage 1S1 = 1 | E 0.2 |
| Endsch. Schieber 1A vorne | 1S2 | BOOL | Vordere Endlage 1S2 = 1 | E 0.3 |
| Endsch. Schieber 2A hinten | 2S1 | BOOL | Hintere Endlage 2S1 = 1 | E 0.4 |
| Endsch. Schieber 2A vorne | 2S2 | BOOL | Vordere Endlage 2S2 = 1 | E 0.5 |
| Endsch. Pusher 3A hinten | 3S1 | BOOL | Hintere Endlage 3S1 = 1 | E 0.6 |
| Endsch. Pusher 3A vorne | 3S2 | BOOL | Vordere Endlage 3S2 = 1 | E 0.7 |
| Endsch. Pusher 4A hinten | 4S1 | BOOL | Hintere Endlage 4S1 = 1 | E 1.0 |
| Endsch. Pusher 5A vorne | 4S2 | BOOL | Vordere Endlage 4S2 = 1 | E 1.1 |
| Lichtschanke 1 | LI1 | BOOL | Unterbrochen LI1 = 1 | E 1.2 |
| Lichtschanke 2 | LI2 | BOOL | Unterbrochen LI2 = 1 | E 1.3 |
| Lichtschanke 3 | LI3 | BOOL | Unterbrochen LI3 = 1 | E 1.4 |
| Lichtschanke 4 | LI4 | BOOL | Unterbrochen LI4 = 1 | E 1.5 |
| Lichtschanke 5 | LI5 | BOOL | Unterbrochen LI5 = 1 | E 1.6 |
| Metallsensor | B1 | BOOL | Metallisches Teil B1 = 1 | E 1.7 |
| Taster Ablaufk. Grundstellung | RESET | BOOL | Betätigt RESET = 1 | E 0.0 |
| Ausgangsvariable | | | | |
| Ventil Schieber 1A vor | 1Y1 | BOOL | Schieber 1Avor 1Y1 = 1 | A 4.0 |
| Ventil Schieber 1A zurück | 1Y2 | BOOL | Schieber 1A zur. 1Y2 = 1 | A 4.1 |
| Ventil Schieber 2A vor | 2Y1 | BOOL | Schieber 2Avor 2Y1 = 1 | A 4.2 |
| Ventil Schieber 2A zurück | 2Y2 | BOOL | Schieber 2A zur. 2Y2 = 1 | A 4.3 |
| Ventil Pusher 3A vor | 3Y1 | BOOL | Pusher 3Avor 3Y1 = 1 | A 4.4 |
| Ventil Pusher 3A zurück | 3Y2 | BOOL | Pusher 3A zur. 3Y2 = 1 | A 4.5 |
| Ventil Pusher 4A vor | 4Y1 | BOOL | Pusher 4Avor 4Y1 = 1 | A 4.6 |
| Ventil Pusher 4A zurück | 4Y2 | BOOL | Pusher 5A zur. 4Y2 = 1 | A 4.7 |
| Schütz Förderbandmotor | Q1 | BOOL | Bandmotor an K1 = 1 | A 5.0 |

2. Ablauf-Funktionsplan:



3. Transitionstabelle:

| Transition | Transitionsbedingung |
|------------|---|
| T-1 | T6_1 = LI5 T9_1 = 3S1 T12_1 = 4S1 |
| T-2 | T1_2 = GA |
| T-3 | T2_3 = 1S1 |
| T-4 | T3_4 = 1S2 |
| T-5 | T4_5 = 2S1 |
| T-6 | T5_6 = 2S2 |

| Transition | Transitionsbedingung |
|------------|----------------------|
| T-7 | T6_7 = B1 |
| T-8 | T7_8 = Zeitgl_1 |
| T-9 | T8_9 = LI3 v 3S2 |
| T-10 | T6_10 = LI2 |
| T-11 | T10_11 = Zeitgl_2 |
| T-12 | T11_12 = LI4 v 4S2 |

4.-6. Funktionsplan, Deklarationstabelle, Realisierung:

Siehe STEP7-Programm auf beiliegender CD oder <http://www.automatisieren-mit-sps.de>
STEP7-Lösungen → 8_2_01

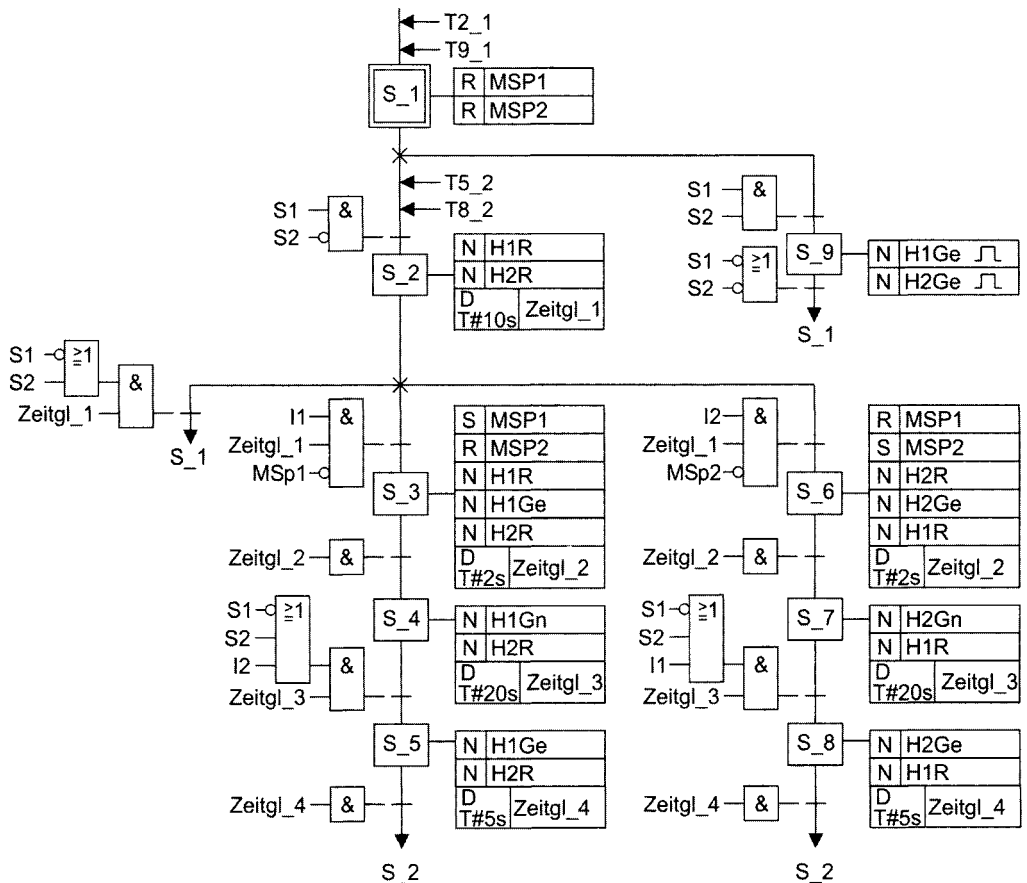
Lösung Lernaufgabe 8.2

Aufg. S. 123

1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | | Adresse |
|-------------------------------|--------|----------|--------------------|-----------|---------|
| Schalter EIN / AUS | S1 | BOOL | Eingeschaltet | S1 = 1 | E 0.1 |
| Schalter TAG / NACHT | S2 | BOOL | Nacht | S2 = 1 | E 0.2 |
| Initiator 1 | I1 | BOOL | Betätigt | I1 = 1 | E 0.3 |
| Initiator 2 | I2 | BOOL | Betätigt | I2 = 1 | E 0.4 |
| Taster Ablaufk. Grundstellung | RESET | BOOL | Betätigt | RESET = 1 | E 0.0 |
| Ausgangsvariable | | | | | |
| Signallampe Rot 1 | H1R | BOOL | Leuchtet | H1R = 1 | A 4.0 |
| Signallampe Gelb 1 | H1Ge | BOOL | Leuchtet | H1Ge = 1 | A 4.1 |
| Signallampe Grün 1 | H1Gn | BOOL | Leuchtet | H1Gn = 1 | A 4.2 |
| Signallampe Rot 2 | H2R | BOOL | Leuchtet | H2R = 1 | A 5.0 |
| Signallampe Gelb 2 | H2Ge | BOOL | Leuchtet | H2Ge = 1 | A 5.1 |
| Signallampe Grün 2 | H2Gn | BOOL | Leuchtet | H2Gn = 1 | A 5.2 |

2. Ablauf-Funktionsplan:



3. Transitionstabelle:

| Transition | Transitionsbedingung |
|------------|--|
| T-1 | $T2_1 = (\overline{S1} \vee S2) \& \text{Zeitgl_1}$ $T9_1 = (S1 \vee \overline{S2})$ |
| T-2 | $T1_2 = S1 \& \overline{S2}$ $T5_2 = \text{Zeitgl_4}$ $T8_2 = \text{Zeitgl_4}$ |
| T-3 | $T2_3 = I1 \& \text{Zeitgl_1} \& \overline{MSP1}$ |

| Trans. | Transitionsbedingung |
|--------|---|
| T-4 | $T3_4 = \text{Zeitgl_2}$ |
| T-5 | $T4_5 = (\overline{S1} \vee S2 \vee I2) \& \text{Zeitgl_3}$ |
| T-6 | $T2_6 = I2 \& \text{Zeitgl_1} \& \overline{MSP2}$ |
| T-7 | $T6_7 = \text{Zeitgl_2}$ |
| T-8 | $T7_8 = (\overline{S1} \vee S2 \vee I1) \& \text{Zeitgl_3}$ |
| T-9 | $S1 \& S2$ |

4.-6. Funktionsplan, Deklarationstabelle, Realisierung:

Siehe STEP7-Programm auf beiliegender CD oder <http://www.automatisieren-mit-sps.de>
STEP7-Lösungen → 8_2_02

Lösung Lernaufgabe 8.3

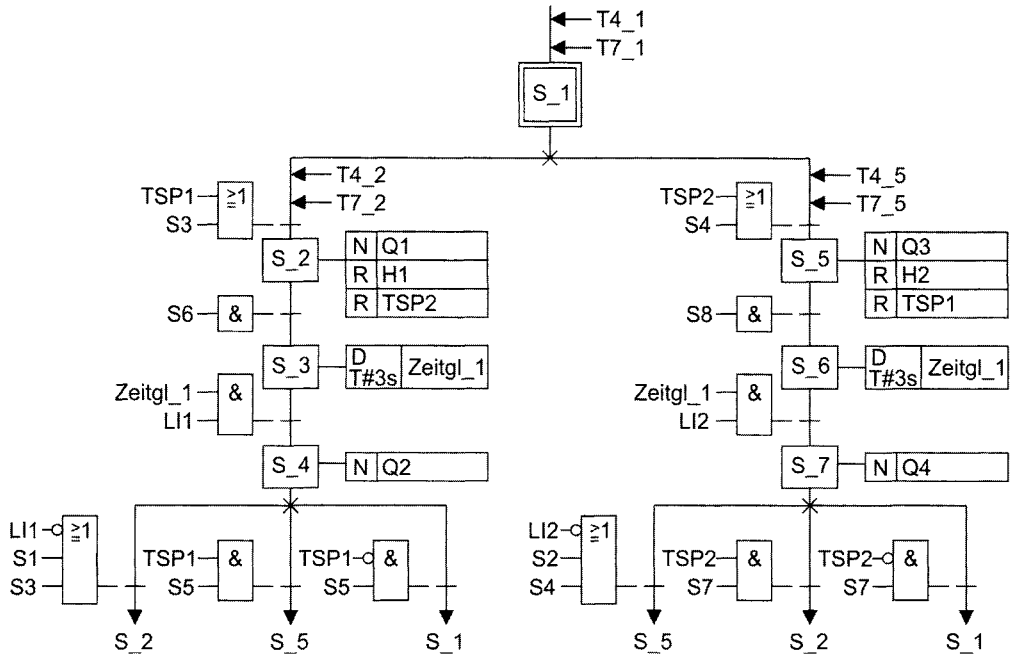
Aufg. S. 124

1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | | Adresse |
|-------------------------------|--------|----------|--------------------|-------------|---------|
| Taster Tür 1 außen | S1 | BOOL | Betätigt | $S1 = 1$ | E 0.1 |
| Taster Tür 2 außen | S2 | BOOL | Betätigt | $S2 = 1$ | E 0.2 |
| Taster Tür 1 innen | S3 | BOOL | Betätigt | $S3 = 1$ | E 0.3 |
| Taster Tür 2 innen | S4 | BOOL | Betätigt | $S4 = 1$ | E 0.4 |
| Endschalter Tür 1 zu | S5 | BOOL | Tür 1 zu | $S5 = 1$ | E 0.5 |
| Endschalter Tür 1 auf | S6 | BOOL | Tür 1 auf | $S6 = 1$ | E 0.6 |
| Endschalter Tür 2 zu | S7 | BOOL | Tür 2 zu | $S7 = 1$ | E 0.7 |
| Endschalter Tür 2 auf | S8 | BOOL | Tür 2 auf | $S8 = 1$ | E 1.0 |
| Lichtschranke Tür 1 | LI1 | BOOL | Unterbrochen | $LI1 = 0$ | E 1.1 |
| Lichtschranke Tür 2 | LI2 | BOOL | Unterbrochen | $LI2 = 0$ | E 1.2 |
| Taster Ablaufk. Grundstellung | RESET | BOOL | Betätigt | $RESET = 1$ | E 0.0 |
| Ausgangsvariable | | | | | |
| Schütz Motor Tür 1 auf | Q1 | BOOL | Tür 1 geht auf | $Q1 = 1$ | A 4.1 |
| Schütz Motor Tür 1 zu | Q2 | BOOL | Tür 1 geht zu | $Q2 = 1$ | A 4.2 |
| Schütz Motor Tür 2 auf | Q3 | BOOL | Tür 2 geht auf | $Q3 = 1$ | A 4.3 |
| Schütz Motor Tür 2 zu | Q4 | BOOL | Tür 2 geht zu | $Q4 = 1$ | A 4.4 |
| Anzeige1 Tastendruck erkannt | H1 | BOOL | Leuchtet | $H1 = 1$ | A 4.5 |
| Anzeige2 Tastendruck erkannt | H2 | BOOL | Leuchtet | $H2 = 1$ | A 4.6 |

2. Ablauf-Funktionsplan:

Vor der Ermittlung des Ablauf-Funktionsplans wird überprüft, ob eine kurzzeitige Betätigung von Eingangsvariablen in einer Signalvorverarbeitung gespeichert werden muss. Für die Taster S1 bzw. S2 trifft dies zu. Die Betätigung des Tasters S1 (S2) bewirkt die speichernde Ansteuerung der Anzeige H1 (H2) und das Setzen eines Tastenspeichers TSP1 (TSP2). Das Rücksetzen der beiden Speicherfunktionen ist aus dem Ablauf-Funktionsplan ersichtlich.



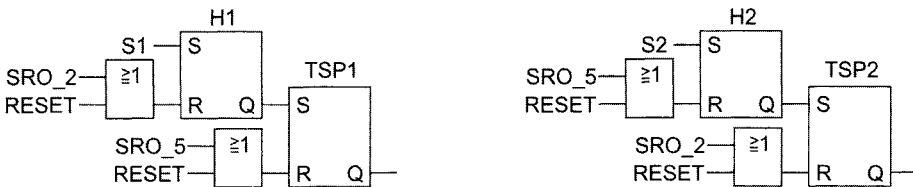
3. Transitionstabelle:

| Transition | Transitionsbedingung |
|------------|--|
| T-1 | $T4_1 = \overline{TSP1} \& S5$ $T7_1 = \overline{TSP2} \& S7$ |
| T-2 | $T1_2 = TSP1 \vee S3$ $T4_2 = \overline{LI1} \vee S1 \vee S3$ $T7_2 = TSP2 \& S7$ |
| T-3 | $T2_3 = S6$ |

| Transition | Transitionsbedingung |
|------------|--|
| T-4 | $T3_4 = Zeitgl_1 \& LI1$ |
| T-5 | $T1_5 = TSP2 \vee S4$ $T4_5 = TSP1 \& S5$ $T7_5 = \overline{LI2} \vee S2 \vee S4$ |
| T-6 | $T5_6 = S8$ |
| T-7 | $T6_7 = Zeitgl_1 \& LI2$ |

4.-6. Funktionsplan, Deklarationstabelle, Realisierung:

Funktionsplan der Signalvorverarbeitung:



Der restliche Funktionsplan für die Umsetzung des Ablauf-Funktionsplanes, die Deklarationstabelle und den Aufruf des Funktionsbausteins siehe STEP7-Programm auf beiliegender CD oder <http://www.automatisieren-mit-sps.de> unter STEP7-Lösungen → 8_2_03

Lösung Lernaufgabe 8.4

Aufg. S. 125

1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|-------------------------------|--------|----------|---------------------------|---------|
| Endschalter Tür oben zu | S1 | BOOL | Betätigt Tür zu S1 = 1 | E 0.1 |
| Endschalter Tür oben auf | S2 | BOOL | Betätigt Tür auf S2 = 1 | E 0.2 |
| Endschalter Tür unten zu | S3 | BOOL | Betätigt Tür zu S3 = 1 | E 0.3 |
| Endschalter Tür unten auf | S4 | BOOL | Betätigt Tür auf S4 = 1 | E 0.4 |
| Endschalter Fahrkorb unten | S5 | BOOL | Betätigt Fk. unten S5 = 1 | E 0.5 |
| Endschalter Fahrkorb oben | S6 | BOOL | Betätigt Fk. oben S6 = 1 | E 0.6 |
| Ruftaster oben aufwärts | S7 | BOOL | Betätigt S7 = 1 | E 0.7 |
| Ruftaster oben abwärts | S8 | BOOL | Betätigt S8 = 1 | E 1.0 |
| Ruftaster unten abwärts | S9 | BOOL | Betätigt S9 = 1 | E 1.1 |
| Ruftaster oben aufwärts | S10 | BOOL | Betätigt S10 = 1 | E 1.2 |
| Lichtschranke Tür oben | LI1 | BOOL | Unterbrochen LI1 = 0 | E 1.3 |
| Lichtschranke Tür unten | LI2 | BOOL | Unterbrochen LI2 = 0 | E 1.4 |
| Taster Ablaufk. Grundstellung | RESET | BOOL | Betätigt RESET = 1 | E 0.0 |
| Ausgangsvariable | | | | |
| Schütz Korbmotor auf | Q1 | BOOL | Fahrkorb auf Q1 = 1 | A 4.1 |
| Schütz Korbmotor ab | Q2 | BOOL | Fahrkorb ab Q2 = 1 | A 4.2 |
| Schütz Türmotor oben auf | Q3 | BOOL | Tür oben geht auf Q3 = 1 | A 4.3 |
| Schütz Türmotor oben zu | Q4 | BOOL | Tür oben geht zu Q4 = 1 | A 4.4 |
| Schütz Türmotor unten auf | Q5 | BOOL | Tür unt. geht auf Q5 = 1 | A 4.5 |
| Schütz Türmotor unten zu | Q6 | BOOL | Tür unt. geht zu Q6 = 1 | A 4.6 |
| Ruf-Anzeige oben aufwärts | H1 | BOOL | Leuchtet H1 = 1 | A 4.7 |
| Ruf-Anzeige oben abwärts | H2 | BOOL | Leuchtet H2 = 1 | A 5.0 |
| Ruf-Anzeige unten aufwärts | H3 | BOOL | Leuchtet H3 = 1 | A 5.1 |
| Ruf-Anzeige unten abwärts | H4 | BOOL | Leuchtet H4 = 1 | A 5.2 |

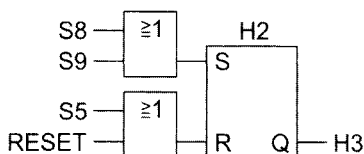
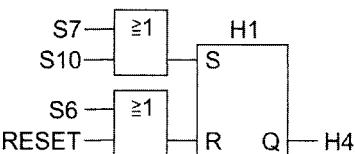
2. Ablauf-Funktionsplan:

Vor der Ermittlung des Ablauf-Funktionsplans wird überprüft, ob eine kurzzeitige Betätigung von Eingangsvariablen mit einer Signalvorverarbeitung gespeichert werden muss. Für die Ruftaster S7 bis S10 trifft dies zu.

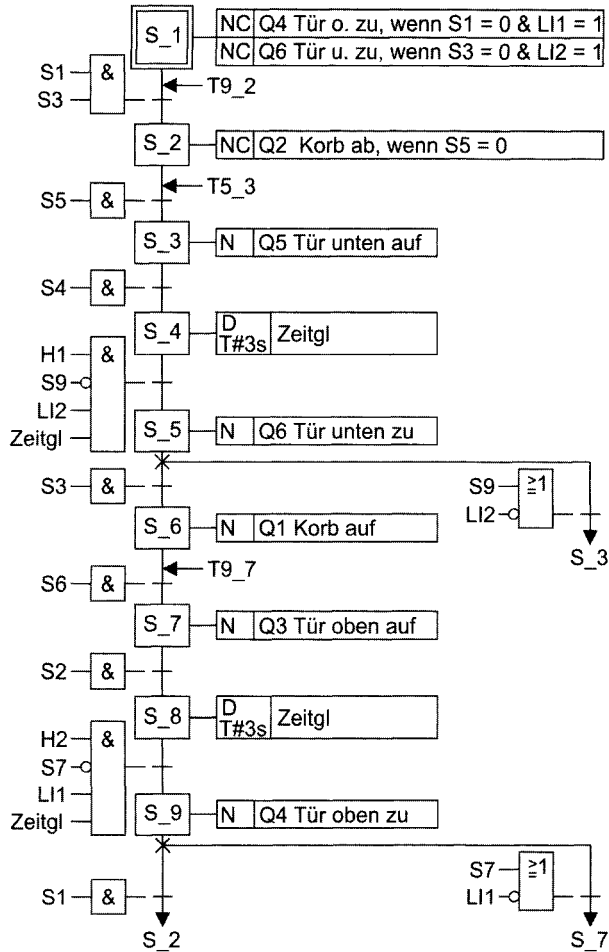
Bei Betätigung der Ruftaster S7 oder S10 (S8 oder S9) werden die Rufanzeigen H1 und H4 (H2 und H3) speichernd angesteuert. Das Rücksetzen der Speicherglieder erfolgt, wenn sich der Aufzugkorb im Restaurant (S6 = 1) bzw. in der Küche (S5 = 1) befindet.

Ist der Korb beispielsweise im Restaurant, erübrigt sich die Speicherung einer etwaigen Aufwärtsanforderung durch S7 bzw. S10. Entsprechendes gilt, wenn sich der Aufzugkorb in der Küche befindet.

Funktionsplan der Signalvorverarbeitung:



Ablauf-Funktionsplan des Speiseaufzugs:



3. Transitionstabelle:

| Transition | Transitionsbedingung |
|------------|------------------------------|
| T-1 | |
| T-2 | T1_2 = S1 & S3 T9_2 = S1 |
| T-3 | T2_3 = S5 T5_3 = S9 v LI2 |
| T-4 | T3_4 = S4 |

| Transition | Transitionsbedingung |
|------------|--|
| T-5 | T4_5 = H1 & $\overline{S9}$ & LI2 & Zeitgl |
| T-6 | T5_6 = S3 |
| T-7 | T6_7 = S6 T9_7 = S7 v LI1 |
| T-8 | T7_8 = S2 |
| T-9 | T8_9 = H2 & $\overline{S7}$ & LI1 & Zeitgl |

4.-6. Funktionsplan, Deklarationstabelle, Realisierung:

Siehe STEP7-Programm auf beiliegender CD oder Internetadresse
<http://www.automatisieren-mit-sps.de> STEP7-Lösungen → 8_2_04

Lösung Lernaufgabe 8.5

Aufg. S. 126

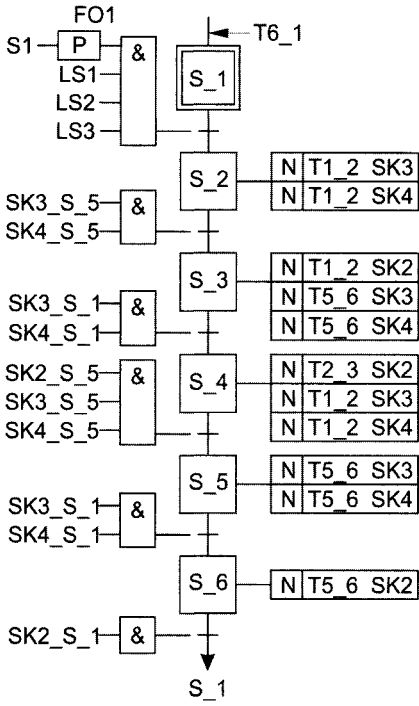
1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | | Adresse |
|-------------------------------|--------|----------|--------------------|-----------|---------|
| Start-Taste | S1 | BOOL | Betätigt | S1 = 1 | E 0.1 |
| Vorlagebehälter 1 | | | | | |
| Behälter leer | LS1 | BOOL | Behälter leer | LS1 = 1 | E 0.2 |
| Dosierzähler | FQS1 | BOOL | Menge erreicht | FQS1 = 1 | E 0.3 |
| Temperatursensor | TS1 | BOOL | Temp. erreicht | TS1 = 1 | E 0.4 |
| Vorlagebehälter 2 | | | | | |
| Behälter leer | LS2 | BOOL | Behälter leer | LS2 = 1 | E 0.5 |
| Dosierzähler | FQS2 | BOOL | Menge erreicht | FQS2 = 1 | E 0.6 |
| Temperatursensor | TS2 | BOOL | Temp. erreicht | TS2 = 1 | E 0.7 |
| Mischkessel | | | | | |
| Behälter leer | LS3 | BOOL | Behälter leer | LS3 = 1 | E 1.0 |
| Temperatursensor 1 | TS3 | BOOL | Temp. erreicht | TS3 = 1 | E 1.1 |
| Temperatursensor 2 | TS4 | BOOL | Abgekühlt | TS4 = 1 | E 1.2 |
| Taster Ablaufk. Grundstellung | RESET | BOOL | Betätigt | RESET = 1 | E 0.0 |
| Ausgangsvariable | | | | | |
| Vorlagebehälter 1 | | | | | |
| Einlassventil | FVK1 | BOOL | Ventil offen | FVK1 = 1 | A 4.0 |
| Heizung | H1 | BOOL | Heizung an | H1 = 1 | A 4.1 |
| Rührwerkmotor | M1 | BOOL | Rührwerk an | M1 = 1 | A 4.2 |
| Auslassventil | FVK2 | BOOL | Ventil offen | FVK2 = 1 | A 4.3 |
| Vorlagebehälter 2 | | | | | |
| Einlassventil | FVK3 | BOOL | Ventil offen | FVK3 = 1 | A 4.4 |
| Heizung | H2 | BOOL | Heizung an | H2 = 1 | A 4.5 |
| Rührwerkmotor | M2 | BOOL | Rührwerk an | M2 = 1 | A 4.6 |
| Auslassventil | FVK4 | BOOL | Ventil offen | FVK4 = 1 | A 4.7 |
| Mischkessel | | | | | |
| Heizung | H3 | BOOL | Heizung an | H3 = 1 | A 5.0 |
| Rührwerkmotor | M3 | BOOL | Rührwerk an | M3 = 1 | A 5.1 |
| Kühlkreislauf | FVK5 | BOOL | Kreislauf offen | FVK5 = 1 | A 5.2 |
| Auslassventil | FVK6 | BOOL | Ventil offen | FVK6 = 1 | A 5.3 |

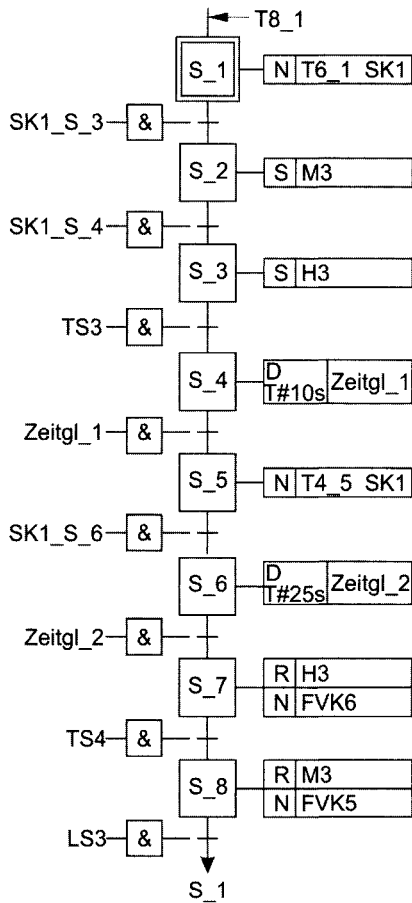
2. Ablauf-Funktionsplan:

Nachfolgend sind drei Ablauf-Funktionspläne dargestellt. Die Koordinations-Schritt看ette SK1, die Schritt看ette für den Mischkesselbetrieb SK2 und die Schritt看ette für den Vorlagebehälter1 SK3. Die Schritt看ette für den Vorlagebehälter2 SK4 entspricht exakt der für den Vorlagebehälter1 und ist nicht dargestellt. Die Koordinations-Schritt看ette SK1 übernimmt die Aufgabe, den Gesamt看etauf der Anlage durch Aufruf und Weiterschaltung der drei anderen Schritt看etten zu steuern.

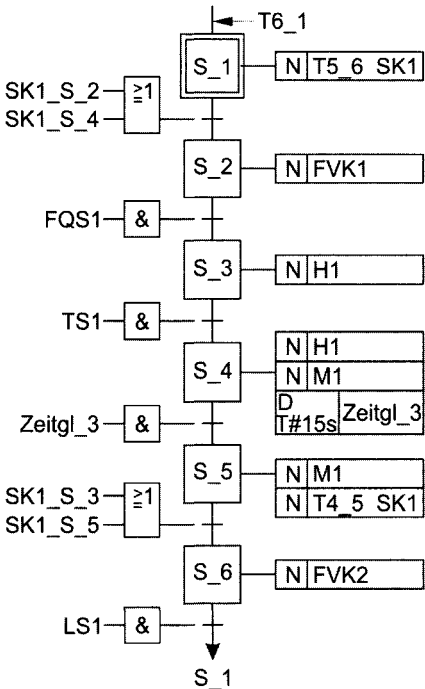
Koordinations-Schritt看ette SK1:



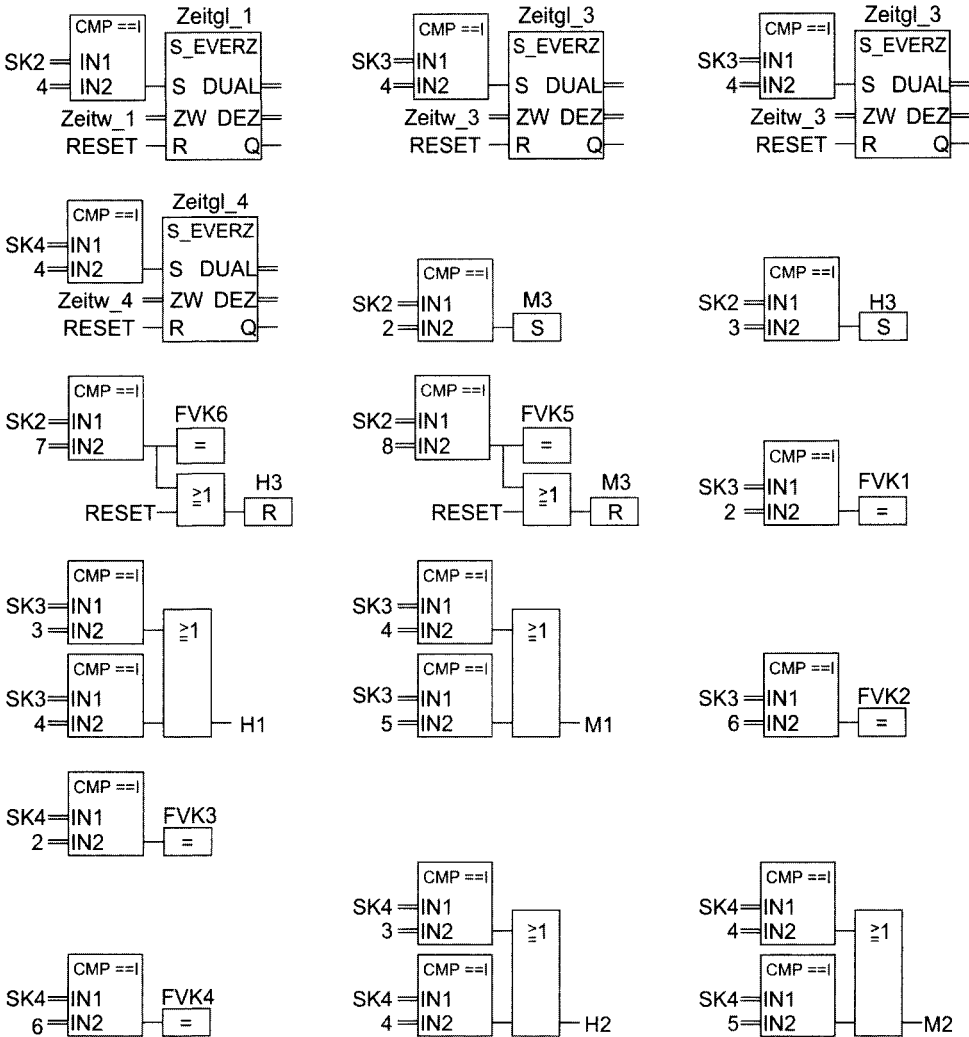
Schritt看ette Mischkessel SK2:



Schritt看ette Vorlagebehälter SK3 (SK4):



3. Funktionsplan Aktionsausgabe:



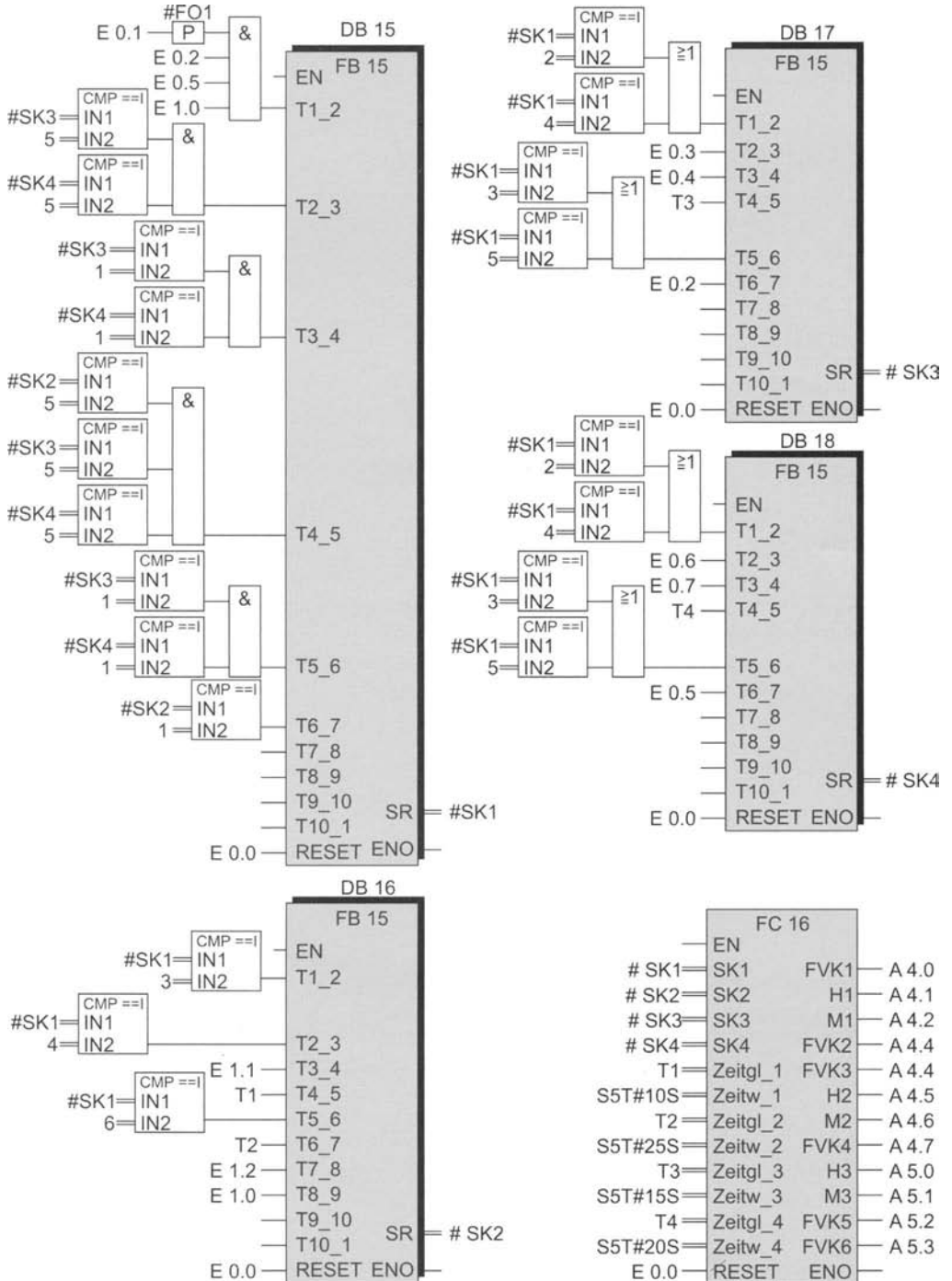
4. Deklarationstabelle Aktionsbaustein FC 16:

| Name | Datentyp |
|----------|----------|
| IN | |
| SK1 | Int |
| SK2 | Int |
| SK3 | Int |
| SK4 | Int |
| Zeitgl_1 | Timer |
| Zeitw_1 | S5Time |
| Zeitgl_2 | Timer |
| Zeitw_2 | S5Time |
| Zeitgl_3 | Timer |

| Name | Datentyp |
|----------|----------|
| IN | |
| Zeitw_3 | S5Time |
| Zeitgl_4 | Timer |
| Zeitw_4 | S5Time |
| RESET | Bool |
| OUT | |
| FVK1 | Bool |
| H1 | Bool |
| M1 | Bool |
| FVK2 | Bool |

| Name | Datentyp |
|------|----------|
| OUT | |
| FVK3 | Bool |
| H2 | Bool |
| M2 | Bool |
| FVK4 | Bool |
| H3 | Bool |
| M3 | Bool |
| FVK5 | Bool |
| FVK6 | Bool |

5. Realisierung: Aufruf im OB1



Vollständiges STEP7-Programm auf beiliegender CD oder

<http://www.automatisieren-mit-sps.de> unter STEP7-Lösungen → 8_2_05

10.9 Lösungen der Lernaufgaben Kapitel 9

Lösung Lernaufgabe 9.1 Aufg. S. 136

1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|----------------------------|--------|----------|------------------------|---------|
| Signalgeber Lüftergruppe 1 | S1 | BOOL | Lüftergruppe an S1 = 1 | E 0.1 |
| Signalgeber Lüftergruppe 2 | S2 | BOOL | Lüftergruppe an S2 = 1 | E 0.2 |
| Signalgeber Lüftergruppe 3 | S3 | BOOL | Lüftergruppe an S3 = 1 | E 0.3 |
| Signalgeber Lüftergruppe 4 | S4 | BOOL | Lüftergruppe an S4 = 1 | E 0.4 |
| Ausgangsvariable | | | | |
| Meldeleuchte P: 10...40 kW | H1 | BOOL | Meldeleuchte an H1 = 1 | A 4.1 |
| Meldeleuchte P = 50 kW | H2 | BOOL | Meldeleuchte an H2 = 1 | A 4.2 |
| Meldeleuchte P >= 60 kW | H3 | BOOL | Meldeleuchte an H3 = 1 | A 4.3 |

2. Bausteintyp:

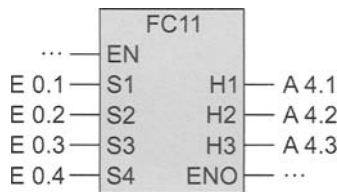
Der Funktionsplan zeigt, dass der zu programmierende Baustein nur Eingabe- und Ausgabevariablen besitzt. Deshalb genügt der Bausteintyp Funktion FC.

3. Deklarationstabelle des Codebausteins FC11:

| Name | Datentyp | Name | Datentyp |
|-----------|----------|------------|----------|
| IN | | IN | |
| S1 | Bool | S3 | Bool |
| S2 | Bool | S4 | Bool |
| | | OUT | |
| | | H1 | Bool |
| | | H2 | Bool |
| | | H3 | Bool |

4. Programmierung und 5. Realisierung:

Aufruf im OBI:



Programmierung und Aufruf des Codebausteins FC siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de>

STEP7-Lösungen → 9_2_01

Lösung Lernaufgabe 9.2 Aufg. S. 137

1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|------------------|--------|----------|--------------------|---------|
| Taster 1 | TA1 | BOOL | Betätigt TA1 = 1 | E 0.1 |
| Taster 2 | TA2 | BOOL | Betätigt TA2 = 1 | E 0.2 |
| Taster 3 | TA3 | BOOL | Betätigt TA3 = 1 | E 0.3 |
| Taster 4 | TA4 | BOOL | Betätigt TA4 = 1 | E 0.4 |
| Taster 5 | TA5 | BOOL | Betätigt TA4 = 1 | E 0.5 |

| | | | | |
|------------------|-----|------|--------------------------|-------|
| AusgangsvARIABLE | | | | |
| Türöffner | TOE | BOOL | Öffner aktiviert TOE = 1 | A 4.1 |

2. Bausteintyp:

Die Signalzustände der lokalen statischen Variablen ZSP1, ZSP2, ZSP3 und ZSP4 müssen gespeichert werden. Das erfordert die Verwendung des Bausteintyps Funktionsbaustein FB.

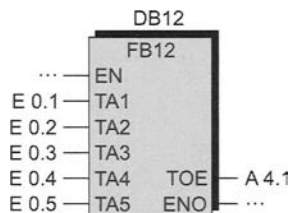
3. Deklarationstabelle des Codebausteins FB12:

| Name | Datentyp | Anfangswert |
|------------|----------|-------------|
| IN | | |
| TA1 | Bool | FALSE |
| TA2 | Bool | FALSE |
| TA3 | Bool | FALSE |
| TA4 | Bool | FALSE |
| TA5 | Bool | FALSE |
| OUT | | |
| TOE | Bool | FALSE |

| Name | Datentyp | Anfangswert |
|-------------|----------|-------------|
| STAT | | |
| ZSP1 | Bool | FALSE |
| ZSP2 | Bool | FALSE |
| ZSP3 | Bool | FALSE |
| ZSP4 | Bool | FALSE |
| ZSP5 | Bool | FALSE |

4. Programmierung und 5. Realisierung:

Aufruf im OB1:



Programmierung und Aufruf des Codebausteins FC siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de> STEP7-Lösungen → 9_2_02.

Lösung Lernaufgabe 9.3

Aufg. S. 137

1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|--|--------|----------|---------------------|---------|
| Taster AUF | AUF | BOOL | Betätigt AUF = 1 | E 0.0 |
| Taster STOPP | STOPP | BOOL | Betätigt STOPP = 0 | E 0.1 |
| Taster ZU Tor | ZU | BOOL | Betätigt ZU = 1 | E 0.2 |
| Endschalter Tor zu | S1 | BOOL | Betätigt S1 = 0 | E 0.3 |
| Endschalter Tor auf | S2 | BOOL | Betätigt S2 = 0 | E 0.4 |
| Wiederholung der 5 Eingangsvariablen für jedes weitere Tor | ... | ... | | |
| Schalter Automatik - Hand | AU_HA | BOOL | Automatik AU_HA = 1 | E 1.7 |
| AusgangsvARIABLE | | | | |
| Schütz Motor auf | Q1 | BOOL | Angezogen Q1 = 1 | A 4.1 |

| | | | | | |
|------------------------|------|------|-----------|----------|-------|
| Schütz Motor zu | Q2 | BOOL | Angezogen | Q2 = 1 | A 4.2 |
| Schütz Motor Tor 2 auf | Q1_2 | BOOL | Angezogen | Q1_2 = 1 | A 4.3 |
| Schütz Motor Tor 2 zu | Q2_2 | BOOL | Angezogen | Q2_2 = 1 | A 4.4 |
| Schütz Motor Tor 3 auf | Q1_3 | BOOL | Angezogen | Q1_3 = 1 | A 4.5 |
| Schütz Motor Tor 3 zu | Q2_3 | BOOL | Angezogen | Q2_3 = 1 | A 4.6 |

2. Bausteintyp:

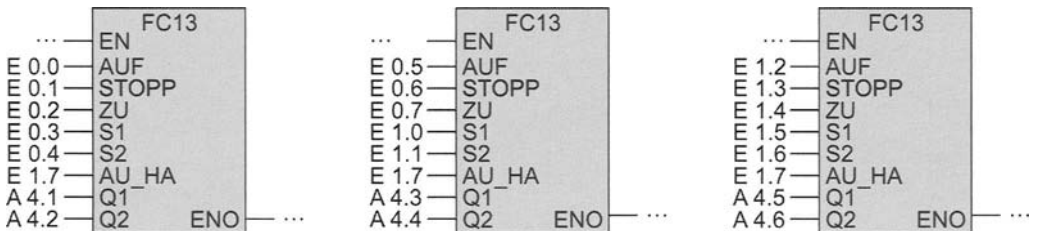
Der zu verwendende Baustein muss Eingabe- und Ausgabevariablen besitzen und die Signal-speicherung der SR-Speicher ermöglichen. An sich muss dazu der Bausteintyp FB (Funktionsbaustein) mit seinem Instanzdatenbaustein verwendet werden. Q1 und Q2 wären dabei als lokale statische Variablen zu deklarieren. Erforderlich sind dann noch zwei abgeleitete Ausgangsvariablen, um die SPS-Ausgänge anzusteuern.

Es gibt aber auch eine einfachere Lösung, bei der der Bausteintyp FC (Funktion) genügt. Dazu werden die Variablen Q1 und Q2 der SR-Speicherfunktionen als IN-OUT-Variablen deklariert. Die Signalspeicherung erfolgt dann in den SPS-Ausgängen. Damit kann der einfachere Bausteintyp FC (Funktion) gewählt werden und es ist kein Instanzdatenbaustein erforderlich.

3. Deklarationstabelle des Codebausteins FC13:

| Name | Datentyp | Name | Datentyp | Name | Datentyp | Name | Datentyp |
|-------|----------|------|----------|-------|----------|--------|----------|
| IN | | IN | | IN | | IN_OUT | |
| AUF | Bool | ZU | Bool | S2 | Bool | Q1 | Bool |
| STOPP | Bool | S1 | Bool | AU_HA | Bool | Q2 | Bool |

4. Realisierung: Aufruf im OB1



Lösung siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de> unter STEP7-Lösungen → 9_2_03.

Lösung Lernaufgabe 9.4 Aufg. S. 137

1. Zuordnungstabelle der Eingänge und Ausgänge:

| Eingangsvariable | Symbol | Datentyp | Logische Zuordnung | Adresse |
|----------------------------|--------|----------|--------------------|--------------|
| Start-Taster | S1 | BOOL | Betätigt | S1 = 1 E 0.1 |
| Endschalter Zyl. 1A hinten | 1S1 | BOOL | Betätigt | S2 = 0 E 0.2 |
| Endschalter Zyl. 1A vorne | 1S2 | BOOL | Betätigt | S3 = 1 E 0.3 |
| Endschalter Zyl. 2A oben | 2S1 | BOOL | Betätigt | S4 = 1 E 0.4 |
| Endschalter Zyl. 1A unten | 2S2 | BOOL | Betätigt | S5 = 0 E 0.5 |
| Endschalter Zyl. 3A unten | 3S1 | BOOL | Betätigt | S6 = 1 E 0.6 |

| | | | | | |
|--------------------------|-----|------|-----------|---------|-------|
| Endschalter Zyl. 1A oben | 3S2 | BOOL | Betätigt | S7 = 1 | E 0.7 |
| Ausgangsvariable | | | | | |
| Magnetventil 1A aus | 1Y1 | BOOL | Angezogen | 1Y1 = 1 | A 4.1 |
| Magnetventil 1A ein | 1Y2 | BOOL | Angezogen | 1Y2 = 1 | A 4.2 |
| Magnetventil 2A aus | 2Y1 | BOOL | Angezogen | 2Y1 = 1 | A 4.3 |
| Magnetventil 2A ein | 2Y2 | BOOL | Angezogen | 2Y2 = 1 | A 4.4 |
| Magnetventil 3A aus | 3Y1 | BOOL | Angezogen | 3Y1 = 1 | A 4.5 |
| Magnetventil 3A ein | 3Y2 | BOOL | Angezogen | 3Y2 = 1 | A 4.6 |

2. Bausteintyp:

Die lokale interne **Variable K2 besitzt eine Selbsthaltung**, da sie auf den Eingang der ODER-Funktion zurückgeführt ist. Der Signalwert dieser Variablen **muss deshalb gespeichert** werden. Das erfordert die Verwendung des Bausteintyps Funktionsbaustein FB.

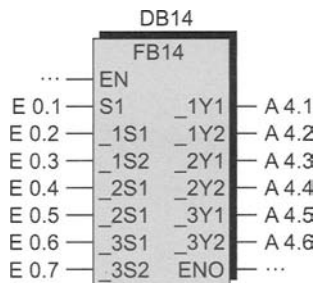
3. Deklarationstabelle des Codebausteins FB14:

| Name | Datentyp | Anfangswert |
|------|----------|-------------|
| IN | | |
| S1 | Bool | FALSE |
| _1S1 | Bool | FALSE |
| _1S2 | Bool | FALSE |
| _2S1 | Bool | FALSE |
| _2S2 | Bool | FALSE |
| _3S1 | Bool | FALSE |
| _3S1 | Bool | FALSE |
| OUT | | |
| _1Y1 | Bool | FALSE |

| Name | Datentyp | Anfangswert |
|------|----------|-------------|
| _1Y2 | Bool | FALSE |
| _2Y1 | Bool | FALSE |
| _2Y2 | Bool | FALSE |
| _3Y1 | Bool | FALSE |
| _3Y2 | Bool | FALSE |
| STAT | | |
| K2 | Bool | FALSE |
| TEMP | | |
| K1 | Bool | FALSE |

4. Realisierung:

Aufruf im OB1:



Programmierung und Aufruf des Codebausteins FB siehe beiliegende CD oder <http://www.automatisieren-mit-sps.de> STEP7-Lösungen → 9_2_04

Anlagensimulation als fester Bestandteil einer professionellen Projektabwicklung

Der Automatisierungstechniker in der Industrie hat in seinem Arbeitsumfeld in der Regel die folgenden, wesentlichen Punkte für einen fehlerfreien Anlagenbetrieb sicher zu stellen:

1. Die Automatisierung**hardware** muss fehlerfrei funktionieren.
2. Die Automatisierungs**software** muss in ihrer Steuerungsfunktion und Ablauffähigkeit korrekt sein.
3. Das **Zusammenspiel** zwischen der Automatisierungstechnologie und der Anlagentechnik muss die gesetzten Anforderungen erfüllen.

Moderne Engineeringkonzepte wie „Simulation Based Engineering“ bringen diese Anforderungen in Einklang. Bevor reale Anlagen oder Maschinen gebaut werden, experimentiert man mit PC-basierten, dynamischen Simulationsmodellen, die zentrale Eigenschaften der Anlage nachbilden. Diese Modelle werden auf industriellen Simulationssystemen von Maschinenbauern, Verfahrenstechnikern, aber auch Automatisierungstechnikern entwickelt und parametrisiert.

Als Beispiel eines solchen industriellen Simulationssystems befindet sich auf der dem Buch beiliegenden CD eine Demo-Version der SIMATIC Engineeringplattform SIMIT¹⁾ der Siemens AG. Die Simulationssoftware SIMIT ist die Standard-Test- und Engineeringplattform für SIMATIC-Anwendungen im industriellen Bereich und findet ihren täglichen Einsatz in der Anlagenplanung sowie in der Anlageninbetriebsetzung. Dies reicht von der Simulation einzelner Maschinen, z.B. eines einzelnen Förderbandes bis hin zu Großanlagen wie beispielsweise eines kompletten Gas- und Dampfturbinenkraftwerkes mit mehreren tausend Ein-/Ausgabesignalen der Leittechnik. Der modulare, objektorientierte Aufbau der Softwareplattform ermöglicht diese hohe Flexibilität im professionellen Einsatz.

Fazit:

Die Simulationssoftware ist ein Hilfsmittel, das den Anwender zielgerichtet bei der Erstellung seines Automatisierungsprogramms unterstützt, und an die „echte“ Anlage heranführt.

In der Kombination von PC-basierter Anlagensimulation im Verbund mit realen Automatisierungskomponenten – sowohl Hardware als auch Software – liegt der Schlüssel zum Erfolg.

Bild: Reale Automatisierungskomponenten und virtuelles Anlagenmodell im Verbund



1) SIMIT = **S**imulation und **I**nformationstechnologie der SIEMENS AG

Anlagensimulation im schulischen Umfeld

Laborarbeitsplätze der Automatisierungstechnik strukturieren sich in der Regel in folgende Teilbereiche:

- Eine Speicherprogrammierbare Steuerung SPS, z.B. SIMATIC S7 300
- Einen PC mit einem SPS-Programmiersystem, z.B. SIMATIC STEP 7
- Optional: Ein Bedien-/Beobachtungssystem, z.B. Bedienpanel
- Ein Anlagenmodell, z.B. ein Förderband als reales Funktionsmodell oder als Simulationsplatte oder jetzt auch als Simulationskonstrukt auf einem PC

Der prinzipielle Aufbau eines Laborplatzes mit einem Anlagenmodell als Simulationskonstrukt lässt sich wie folgt skizzieren:

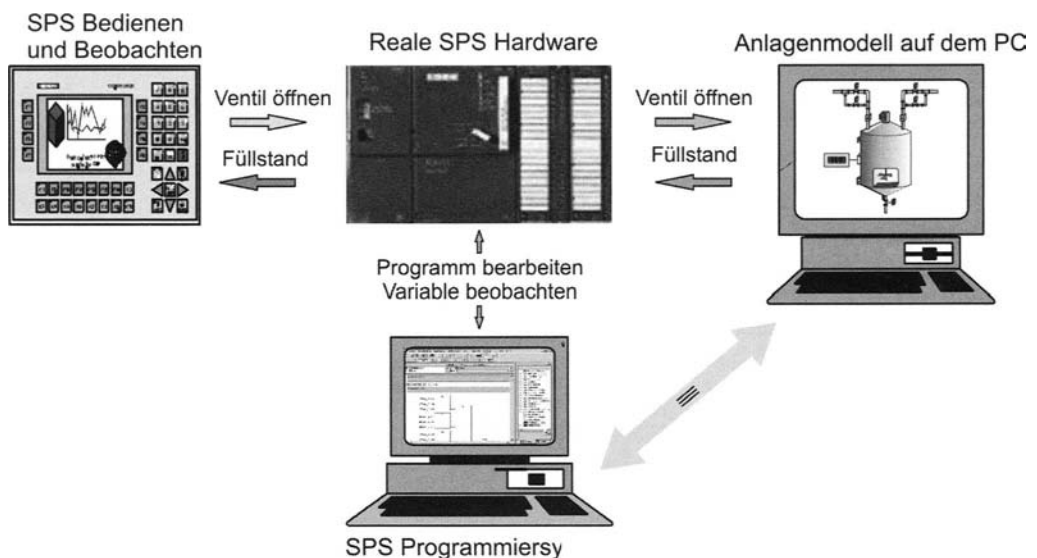


Bild: Prinzipieller Aufbau einer PC-basierten Anlagensimulation im Verbund mit einer SPS einschließlich Bedien- & Beobachtungssystem (z.B. SIMATIC ProTool/Pro)

Beispiel für einen Datenfluss:

Ein Eingabesignal, z.B. das Öffnen des Zulaufventils, verläuft vom Bedien- & Beobachtungssystem zur realen SPS, die programmgemäß ein Ausgabesignal erzeugt, das zum PC mit dem SIMIT-Anlagenmodell gelangt. Dort zeigt ein Farbumschlag an, dass das Ventil geöffnet wurde. Gleichzeitig berechnet das Anlagenmodell den Füllstand gemäß einer mathematischen Beziehung und simuliert so z.B. einen Füllstandssensor. SIMIT meldet den jeweils aktuellen Füllstand an die SPS, die ihrerseits das Signal an das Bedien- und Beobachtungssystem weiterleitet. Das SPS-Programmiersystem und das Anlagen-Simulationsprogramm SIMIT laufen dabei in der Regel auf demselben PC.

Diese in der Industrie bereits gängige Vorgehensweise lässt sich unter Einsatz der SIMATIC-Hardware- und Softwarekomponenten auf den schulischen Laborbereich übertragen. Der folgende Schulungsaufbau zeigt eine mögliche Umsetzung dieses Konzepts und steht exemplarisch für eine Vielfalt möglicher Simulationsmodelle.

Vorgehensweise bei der Inbetriebnahme des Laborplatzes:

1. Erstellung des SPS-Programms im SIMATIC Manager mit STEP 7.
2. Download und Starten des STEP 7-Programms auf die S7-CPU.
3. Start der Simulation des SIMIT-Anlagenmodells mit eingestellter MPI-Kopplung zur S7-CPU.

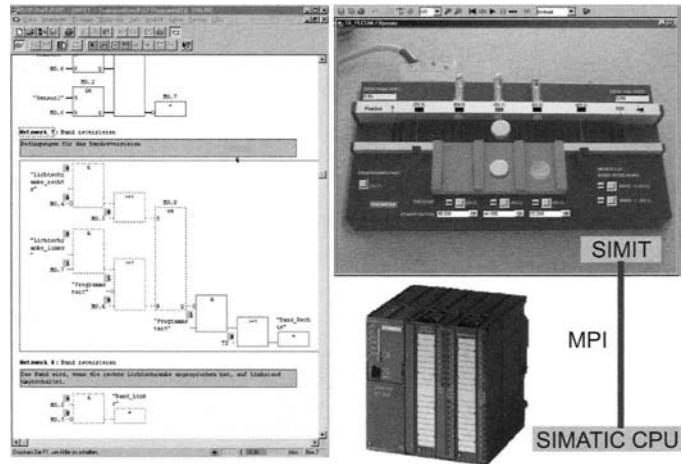


Bild: Ansicht eines Laborplatzes – im Verbund sind dargestellt: SIMATIC STEP 7 / SIMATIC CPU / dynamisiertes SIMIT-Anlagenmodell.

Ein weiterer Abstrahierungsgrad wird erreicht, wenn das Steuerungsprogramm in der Software S7-PLCSIM im Verbund mit STEP 7 und SIMIT auf dem PC abläuft. Dem Anwender stehen in diesem Fall alle Applikationen auf einem PC zur Verfügung und er kann zumindest die logische Softwarefunktionalität vorprüfen. Übertragen auf die schulische Anwendung besteht nun die Möglichkeit, dass dieser „Aufbau“ auch auf dem PC zu Hause vom Schüler/Studenten als Hausaufgaben-Arbeitsplatz genutzt werden kann (virtuelles Automatisierungslabor).

Da ein PC jedoch eine „andere“ Hardware ist als die reale SPS, die sich somit auch „anders“ verhält und insbesondere vom PC-Betriebssystem beeinflusst wird, sollte aus didaktischer Sicht immer ein abschließender Test ohne Simulation auf der realen SPS erfolgen. Nur hier können hardware-spezifische Details wie beispielsweise Buszykluszeiten und die richtige Adressierung der Busteilnehmer oder die korrekte Funktionalität der Hardwarekonfiguration ermittelt werden.

Wird im schulischen Umfeld darüber hinaus beispielsweise noch mit CAD-Software gearbeitet, lassen sich sehr realitätsnahe Projektablaufe umsetzen. Beispielsweise ist ein Transportband mit Aktorik/Sensorik zunächst als technische CAD-Zeichnung zu entwickeln. Diese bildet dann die Grundlage für das dynamische SIMIT Simulationsmodell, mit dem die zugehörige Automatisierung konzipiert und in ihrem dynamischen Verhalten geprüft wird. Die Prüfung kann zunächst auf der abstrakten PC-Ebene im Verbund SIMIT/S7-PLCSIM erfolgen und schließlich mit der realen S7-CPU, was für den Schüler eine klarere Trennung zwischen Automatisierungshardware und Anlagensimulation darstellt.

Abhängig vom Berufsbild des Schülers könnte in der betrieblichen Ausbildung idealerweise auch noch der handwerkliche Aufbau des Transportbandes in Verbindung mit den Automatisierungsgeräten erfolgen. Auf diese Weise wird dem Schüler der komplette Projektablauf vom Design über das simulationsbasierte Engineering der Automatisierung bis hin zum Bau und Betrieb dieser exemplarischen Transportbandanlage transparent dargestellt.

Didaktisches Konzept

Für die Schule/Hochschule sollte das virtuelle Automatisierungslabor nur eine Ergänzung der realen Automatisierungsausrüstung sein, um den Schülern/Studenten einen Einblick in professionelle Projektabwicklung zu geben und um den Bezug der preiswerten Softwareschulversionen nutzen zu können, um damit auch einen schon aus Zeitgründen dringend erforderlichen selbstständigen Umgang der Schüler/Studenten mit der komplexen Software zu ermöglichen. Nicht zu empfehlen ist die Totsimulation als Ersatz für die bewährte Laborpraxis. Mit dem Verschwinden der Automatisierungshardware aus den Automatisierungslaboren würden sich diese in allgemeine Computerräume verwandeln und die diesbezügliche Ausstattung der Schule/Hochschule mit vernetzten Laborräumen bestände dann nur noch aus einigen Softwarepaketen bei den Netzwerk-Administratoren. Zum Bildungsauftrag der Schulen/Hochschule im Bereich der Automatisierungstechnik zählt die Vermittlung eines strukturierten Kernwissens, das auch Lösungsstrategien/-methoden zur Umsetzung von Aufgabenstellungen beinhaltet. Die Anwendung dieser Grundlagen sollte auch weiterhin durch die Realisierung von projekthaften Steuerungsaufgaben mit realen Automatisierungsgeräten (z.B. SPS oder PC mit Feldbussystemen und dezentraler Peripherie) und richtiger Inbetriebnahme der „Anlagen“ unter Anwendung der zur Verfügung stehenden Diagnosesysteme erfolgen.

SIMIT Demo-Beispiele

Ergänzend zu den Ausführungen im Vorwort des Buches seien hier noch einige technische Details zur Durchführung der Demo-Beispiele erwähnt:

Die beigelegte SIMIT-CD (Buchrückseite) enthält unter anderem eine Demo-Version der Siemens SIMATIC Simulationssoftware SIMIT 5.0 SP1, die zwei komplett lauffähige Anlagenmodell beinhaltet. Nach der Installation der Software (bitte die „readme“-Datei auf der CD beachten) führt eine kurze Online-Dokumentation in die einfache Bedienung von SIMIT ein.

Mit einer installierten Basissoftware SIMATIC STEP 7 (ab Version 5.1) haben Sie die Möglichkeit, die beiden SIMIT Modelle (Beispielaufgabe 7.1 und Lernaufgabe 7.1) mit S7-PLCSIM auszuprobieren.

Die beigelegte CD mit der Basissoftware SIMATIC STEP 7 V 5.3 (Buchvorderseite) kann zum Ausprobieren der SIMIT-Modelle und Kennenlernen von Bedienoberfläche auch ohne License Key verwendet werden. Wenn für STEP 7 V 5.3 kein gültiger Licence Key installiert ist, wird ein Trial License Key verwendet, der mit STEP 7 standardmäßig mitgeliefert und installiert wird. Mit diesem License Key kann STEP 7 14 Tage voll genutzt werden.

Weitere Informationen zu SIMIT sowie den SIMATIC Ausbildungskonzepten „Siemens cooperates with education“ findet man auf der CD im Verzeichnis „Dokumente“ oder im Internet unter <http://www.siemens.de/sce> .

Zur Person:

Siemens AG
Bereich Industrial Solutions & Services
Dipl. Phys. Franz Deutel
Senior Manager des Siemens Simulation Centers in Erlangen

Erlangen, im September 2003

SIMIT ist eine eingetragene Marke der Siemens AG.
Das verwendete Bildmaterial dieses Anhangs ist Eigentum der Siemens AG.

Sachwortverzeichnis

- 3/2-Wegeventil 12
- 5/2-Wegeventil 12
- 5/3-Wegeventil 12

- Ablaufkette 74
- Ablaufschritt 74
- Ablaufsteuerungen
 - , Aktionsbaustein 78
 - , anlagenspezifischer Funktionsbaustein 76
 - , Bestimmungszeichen für Aktionen 75
 - , mit Betriebsartenteil 94
 - , mit Verzweigungen 114
 - Alternativverzweigung 114
 - Kettenschleife 115
 - Simultanverzweigung 115
 - , ohne Betriebsartenteil 74
 - , Standard-Ablaufkettenbausteine 77, 95, 98
 - , Standard-Betriebsartenbaustein 95, 96 f.
 - , Struktur 95
- AWL (= Anweisungsliste) 2

- Bausteinfunktionen 62
- Bedienfeld für Ablaufsteuerungen 94
- Betriebsarten bei Ablaufsteuerungen
 - , Automatikbetrieb 94
 - , Einrichten 96
 - , Einzelschrittbetrieb 96

- Datentypumwandlung 61
- Deklaration
 - , globaler Variablen 6
 - , lokaler Variablen 6, 136
- Deklarationstabelle 6, 136
- Deklarationstypen
 - , IN 6, 136
 - , IN_OUT 6, 136
 - , OUT 6, 136
 - , STAT 6, 136
 - , TEMP 6, 136
- DNF (Disjunktive Normalform) 9

- Flankenauswertung 19
- Funktionsbaustein FB 5f, 136

- Funktionsdiagramm (Weg-Schritt) 20
- Funktion FC 5, 136
- Funktionsplan 2
- Freigrafischer Funktionsplan 43

- Grundverknüpfungen, logische
 - , einfache 2
 - , zusammengesetzte 8
- Kontaktplan KOP 2

- Organisationsbaustein OB1 3, 5, 136

- RS-Speicherfunktion
 - , Speichern mit vorrangigem Setzen 18
 - , Speichern mit vorrangigem Rücksetzen 18
- RS-Tabelle 20

- Schaltfolgetabelle 19
- Sprungfunktionen 62
- Stellglieder, elektropneumatische 20

- Symboltabelle 5

- Taktgeber 40
- Transition 74
- Transitionsbedingung 74
- Transitionstabelle 74, 102, 114

- Umsetzungsregeln für Stromlaufpläne 21
- Übertragungsfunktionen
 - , bei AWL 60
 - , bei FUP 61

- Variablendeklaration 6
- Vergleichsfunktionen 62
- Verriegeln von Speichern
 - , gegenseitig 18
 - , Reihenfolge 18

- Weckalarm 40

- Zählfunktionen 41
- Zeitfunktionen 38

Was bietet die beiliegende CD?



1_Benutzerhandbuch.pdf

Kurze Einführung in SIMIT mit einer Anleitung zur Durchführung der DEMO-Projekte Beispiel 7.1 (Biegemaschine) und Lernaufgabe 7.1 (Rührkessel).



2_SIMIT_SCE_Telehome_Demo

Software SIMIT-Demoversion mit ablauffähiger Anlagensimulation für das Beispiel 7.1: Biegewerkzeug und die Lernaufgabe 7.1: Rührkessel.

Betriebssystem-Voraussetzung: Microsoft Windows 2000/XP/NT4.0.

(Die zur Verfügung gestellte SIMIT-Demoversion kann nicht für andere Anlagenmodelle genutzt werden).



3_step7d.zip

Archiviertes STEP7 Steuerungsprogramm für die DEMO-Projekte Beispiel 7.1 und Lernaufgabe 7.1



4_Weitere_Anlagenmodelle

36 SIMIT-Anlagenmodelle für SIMIT 5.0 SP1 im SP-Format für Beispiele und Lernaufgaben. Zusammenstellung siehe Datei: uebersicht.pdf

(Diese Anlagenmodelle müssen importiert werden und erfordern eine Voll-Version der Software SIMIT).



5_s7lv30.zip

Archivierte STEP7-Programme der Lösungsvorschläge für alle Beispiele und Lernaufgaben.



6_L_Kontrollaufgaben_3.pdf

Lösungsvorschläge für alle Kontrollaufgaben.



Readme.txt