

SIEMENS

Microsystem SIMATIC S7-200

2-Stunden-Fibel

Ausgabe 01/2000



microsystems

Sicherheitstechnische Hinweise

Die 2-Stunden-Fibel wurde für den schnellen Einstieg in die S7-200-Welt geschaffen und ist bewußt kurz gehalten. Sie ist kein Ersatz für das S7-200-Handbuch.

Bitte beachten Sie deshalb unbedingt die im S7-200-Handbuch gemachten Angaben, insbesondere die sicherheitstechnischen Hinweise.

Marken

SIMATIC[®] und SIMATIC NET[®] sind eingetragene Marken der Siemens AG.

Die übrigen Bezeichnungen in dieser Schrift können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen können.

Copyright © Siemens AG 2000 All rights reserved

Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts ist nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte vorbehalten, insbesondere für den Fall der Patenterteilung oder GM-Eintragung.

Siemens AG
Bereich Automatisierungs- und Antriebstechnik
Geschäftsgebiet Industrie-Automatisierungssysteme
Postfach 4848, D-90327 Nürnberg

Haftungsausschluß

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so daß wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden regelmäßig überprüft, und notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten. Für Verbesserungsvorschläge sind wir dankbar.

© Siemens AG 2000

Technische Änderungen bleiben vorbehalten

Übungsbeispiele

Die Übungsbeispiele für die 2-Stunden-Fibel finden Sie im Verzeichnis:
"PRIM_200/2H_EX/"

Inhaltsverzeichnis

Wiederholung	Ein paar Worte zur Wiederholung 5 Hier sind die Bits 6 Stromfluss im Kontaktplan 7 Der SPS-Zyklus 9
↓	
Selbsthaltung	Einführung 13 Öffner-Kontakt 14 Beschreibung der Lösung und Test 16 Selbsthaltung mal anders... 17
↓	
Stromstoß-Schaltung	Einführung 21 Lösung im Überblick 22 Flankenerfassung 23 Merker 25 Beschreibung der Lösung und Test 27
↓	
Ausschaltverzögerung	Einführung 29 Speichern unter... 31 Netzwerk einfügen 32 Beschreibung der Lösung 33 Kommentare eingeben 36
↓	
Schrittkeite	Einführung 39 Grundlagen 41 Arbeiten mit Schrittketten 45 Modifikation 50 Lösungsbeschreibung, Beispiel 51 Test 55
↓	
Anhang	Tipps A1 Index B1



Die Gliederung der 2-Stunden-Fibel finden Sie in den Fußzeilen aller Seiten wieder.
 Das Kapitel, in dem Sie sich befinden, ist jeweils hervorgehoben.

Vorwort

Sehr geehrte S7-200-Anwenderin, sehr geehrter S7-200-Anwender, die Effizienz beim Einsatz moderner Micro-Steuerungen hängt maßgebend davon ab, wie schnell und sicher Sie in der Lage sind, eine Steuerung einzusetzen. Damit auch Anfänger schnell und einfach die S7-200 handhaben können, haben wir die 1- und die 2-Stunden-Fibel geschaffen.

Die Ihnen vorliegende 2-Stunden-Fibel wird Sie, aufbauend auf der 1-Stunden-Fibel, in kurzer Zeit mit der Arbeitsweise der Kleinsteuerung S7-200 vertraut machen. Die Fibel zeigt Ihnen an einigen Beispielaufgaben, wie die Steuerung arbeitet und wie diese für einfache Aufgaben effektiv eingesetzt wird.

Es wird Ihnen leicht fallen, nach dem Durcharbeiten der 2-Stunden-Fibel typische Steuerungsaufgaben selbstständig zu lösen.

Wir wünschen Ihnen viel Erfolg beim Bearbeiten der einzelnen Aufgaben.

Im Anhang befindet sich eine Diskette, von der Sie die angesprochenen Beispiele laden können.



Das S7-200-Team wünscht
Ihnen viel Erfolg!

Wiederholung

Selbsthaltung

Stromstoß-
Schaltung

Ausschalt-
Verzögerung

Schrittkeite

Anhang

1

2

Wiederholung

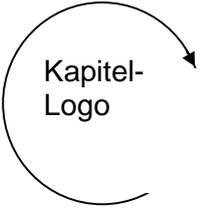
Selbsthaltung

Stromstoß-Schaltung

Ausschalt-Verzögerung

Schrittkeite

Anhang


 Kapitel-
Logo

Fibel-Symbole

In der 2-Stunden-Fibel werden zahlreiche Symbole und Text hervorhebungen verwendet, deren Bedeutung wir Ihnen auf dieser Seite kurz erklären.

Sehen Sie sich am besten gleich den Seitenkopf an. Jede Seite hat prinzipiell einen identischen Seitenkopf. Die Überschrift in großen Buchstaben kennzeichnet den aktuellen Unterpunkt des Kapitels. Der Bereich "Neues" im rechten Kopfteil zeigt Ihnen die Inhalte der vorhergehenden Seiten, schwarz hervorgehoben den Inhalt der aktuellen Seite und anschließend die Inhalte der nächste(n) Seite(n).

Text mit grauer Hinterlegung fordert Sie auf, selbst aktiv zu werden, z.B. für Eingaben per Maus oder über die Tastatur.



Dieses Symbol zeigt Ihnen, dass die linke Maustaste einmal (Klick) für eine Aktion betätigt werden muss (z.B. zum Markieren von Feldern).



2 x

Dieses Symbol zeigt Ihnen, dass die linke Maustaste zweimal schnell hintereinander (Doppelklick) für eine Aktion betätigt werden muss.



Hier ist die Eingabetaste (auch ENTER oder RETURN genannt) Ihrer Tastatur zu betätigen.



Dies kennzeichnet, dass Sie mit Maus oder wahlweise mit Tastatur (Funktionstasten, Pfeiltasten) vorgegebene Listenpunkte der Bildschirmoberfläche auswählen können.

F2

Dies bedeutet, dass Sie die Funktionstaste "F2" (es gibt die Funktionstasten F1 ... F12) betätigen müssen. Sie werden feststellen, dass trotz komfortabler Mausbedienung die Verwendung der Tastatur in manchen Situationen ein zügigeres Arbeiten erlaubt.



In Kombination mit einem Seitenverweis finden Sie hier weitere Informationen zu einem bestimmten Thema.

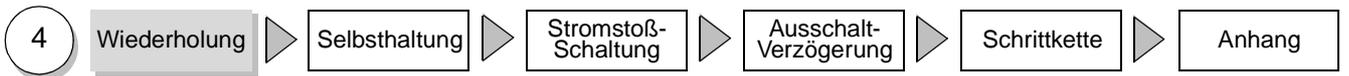


An diesen Stellen werden Sie gebeten, Eingaben in Textfelder der Bildschirmoberfläche zu tätigen. Oder Sie werden erinnert, dass Sie hier bei Ihren eigenen Projekten Notizen aufzeichnen sollten.



Menü

Ein Menü-Punkt der Bildschirmoberfläche ist mit der linken Maustaste schrittweise (Oberpunkt, Unterpunkt) zu aktivieren.



- Ein paar Worte zur Wiederholung
- Hier sind die Bits
- Stromfluss im Kontaktplan
- Der SPS-Zyklus

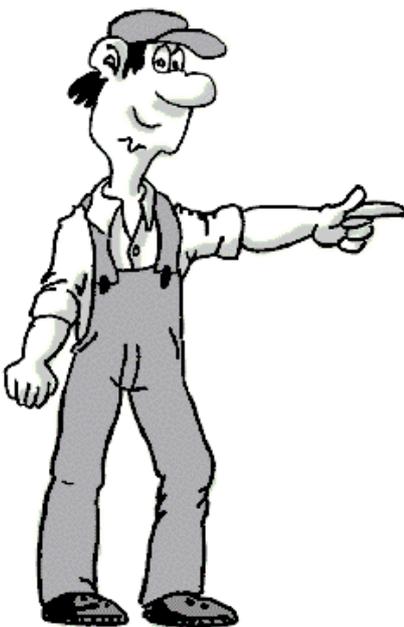


Ein paar Worte zur Wiederholung ...

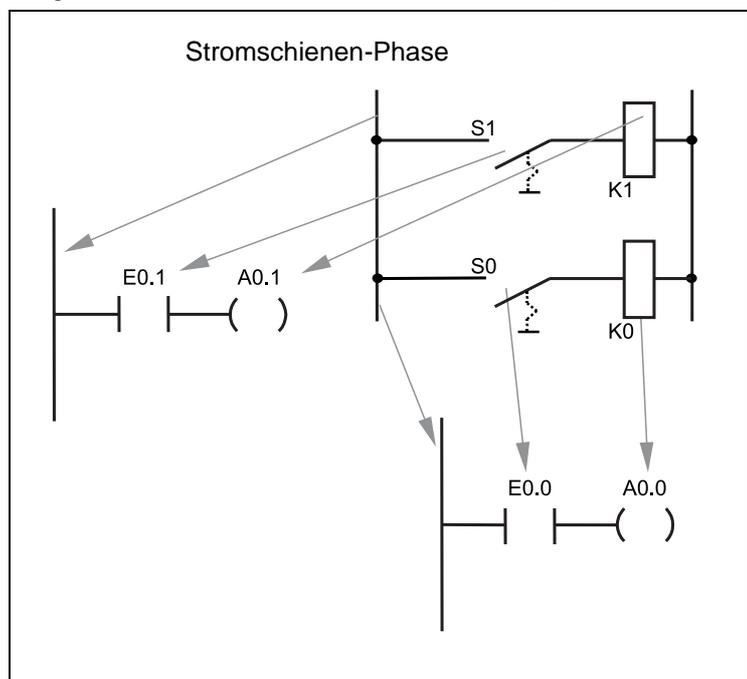
In der 1-Stunden-Fibel haben Sie gesehen, dass der Stromlaufplan für Schütz-Steuerungen verwandt mit dem Kontaktplan zur Programmierung von Speicherprogrammierbaren Steuerungen (SPS) ist.

Er ist lediglich eine Darstellung mit anderen Symbolen.

Außerdem waren Sie schon in der Lage, kleine Verknüpfungen selbst zu programmieren. Selbst Zeitglieder lernten Sie in der kurzen Zeit schon kennen.



Vergleichen Sie auch mit der 1-Stunden-Fibel, Seite 24:



- Ein paar Worte zur Wiederholung
- Hier sind die Bits
- Stromfluss im Kontaktplan
- Der SPS-Zyklus

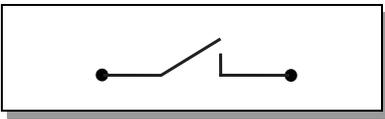


Hier sind die Bits

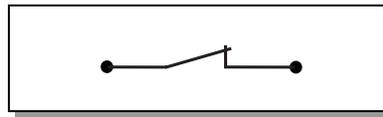
Die kleinste zu verarbeitende Einheit ist das Bit.
Das Bit kann zwei Zustände annehmen:

- 1) "1" bedeutet "Bit gesetzt" oder Zustand ist "wahr",
- 2) "0" bedeutet "Bit nicht gesetzt" oder Zustand ist "unwahr".

In einer Ihnen vertrauten Darstellungsweise lassen sich die beiden binären Zustände "1" und "0" als elektrische Schaltkreise, d.h. durch Schalter darstellen.

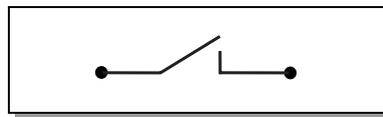


Dabei bedeutet ein geschlossener Schalter:
Stromdurchgang, also Zustand Bit = "1"



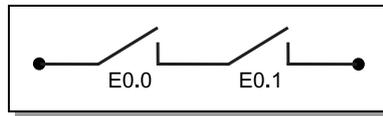
"1" = "wahr"
Stromdurchgang

und ein offener Schalter:
kein Stromdurchgang, also Zustand Bit = "0".



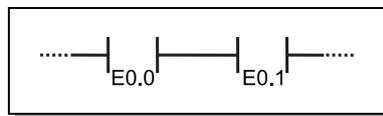
"0" = "unwahr"
kein Stromdurchgang

Es ist jetzt nur noch ein kleiner Schritt, dass auch logische Verknüpfungen, z.B. die Reihenschaltung von zwei Kontakten als Schaltkreise dargestellt werden können. Die UND-Verknüpfung der Eingänge E0.0 und E0.1 wird wie nebenstehend dargestellt.



UND-
Verknüpfung

Dies entspricht in der KOP-Darstellung:



Zum Schluss noch eine Festlegung.

Bei positiver Logik gilt:

24V = High-Pegel = "1" und

0V = Low-Pegel = "0".

Bei negativer Logik gilt:

0V = Low-Pegel = "1"

24V = High-Pegel = "0".

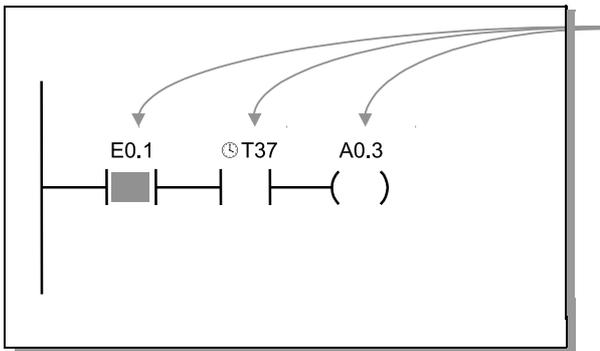
positive Logik

negative Logik

- Ein paar Worte zur Wiederholung
- Hier sind die Bits
- **Stromfluss im Kontaktplan**
- Der SPS-Zyklus

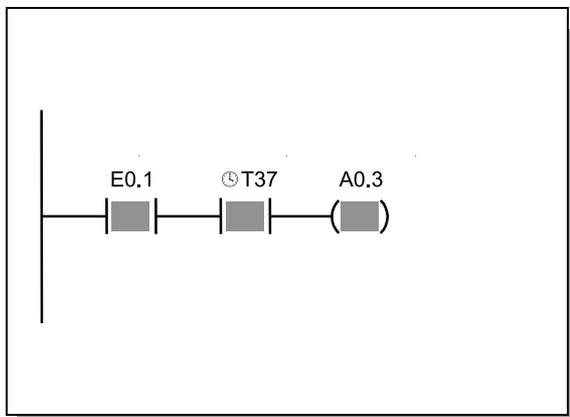


Stromfluss im Kontaktplan (1)



In diesem Beispiel wird der Ausgang A0.3 eingeschaltet oder "1", wenn der Kontakt an E0.1 geschlossen wird (DC 24 V am Eingang E0.1) UND gleichzeitig das Zeitbit T37 aktiviert ist.

Der Eingang E0.1 ist jetzt "1", d.h. der Kontakt E0.1 ist geschlossen. T37 ist im Bild nicht aktiv d.h. "0". Deshalb bleibt A0.3 auch ausgeschaltet d.h. "0".



Wenn auch die Zeit T37 "1" ist (Timer T37 ist abgelaufen), wird das Ergebnis der UND-Verknüpfung "1" und damit der Ausgang A0.3 auch "1".

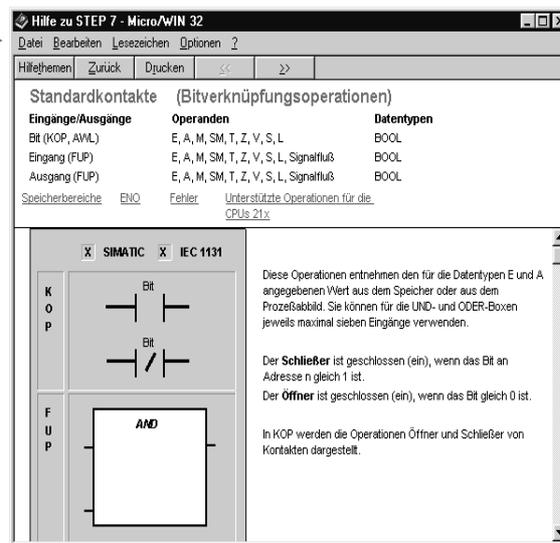
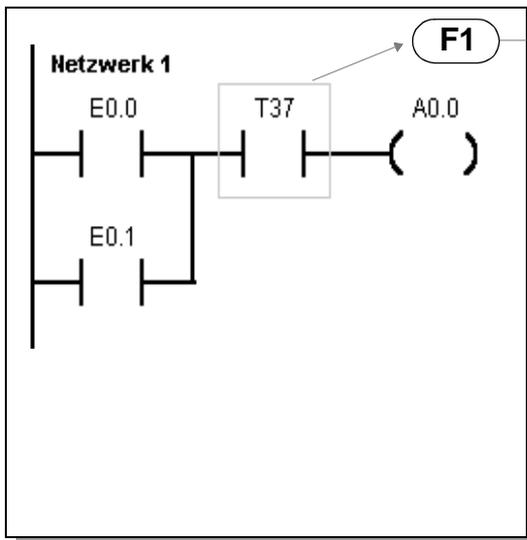
Das Ausgangs-Bit ist dann ebenfalls "wahr", hat also den Wert "1". Die graue Hinterlegung der KOP-Statusanzeige symbolisiert den Wert "1" des Ausgangs-bits.

Die KOP-Statusanzeige haben Sie bereits in der 1-Stunden-Fibel zum Testen Ihrer Programme genutzt.

- Ein paar Worte zur Wiederholung
- Hier sind die Bits
- **Stromfluss im Kontaktplan**
- Der SPS-Zyklus



Stromfluss im Kontaktplan (2) (Hilfefunktion nutzen)



Hilfe anzeigen

1. Element
markieren

2. F1



Falls Sie erneut die online-Hilfe zu einem Kontaktsymbol oder anderen Funktionen einsehen möchten:

Markieren Sie den Kontakt

- im Kontaktplan (KOP) oder
- im Funktionsplan (FUP) bzw.
- markieren Sie den Kontakt im Operationsbaum von STEP 7-Micro/WIN

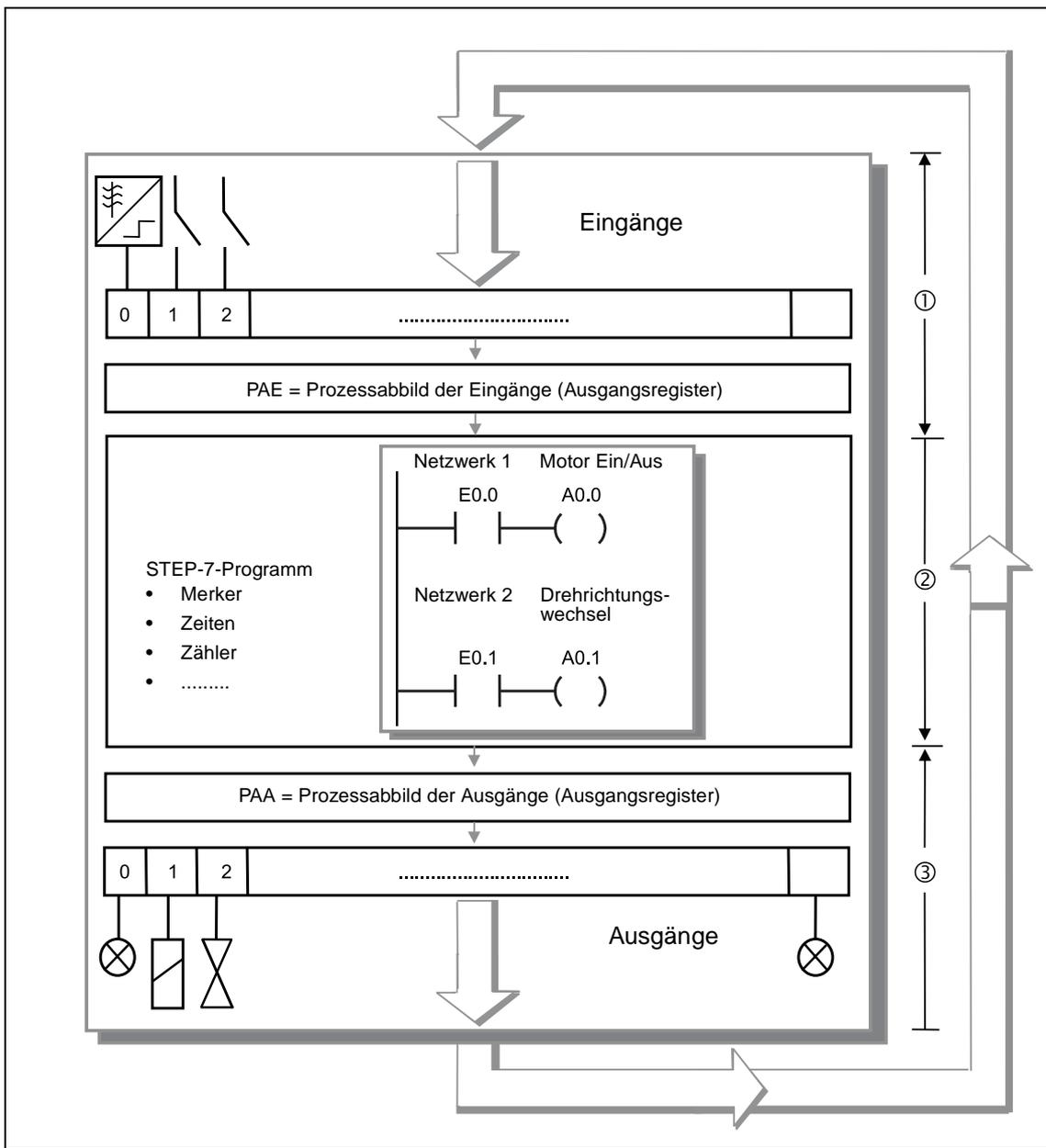
mit einem einfachen Mausklick und drücken dann die F1-Taste.

F1 Online-Hilfe

- Ein paar Worte zur Wiederholung
- Hier sind die Bits
- Stromfluss im Kontaktplan
- Der SPS-Zyklus



Der SPS-Zyklus (1)



Alle SIMATIC-Automatisierungsgeräte arbeiten prinzipiell zyklisch. In diesem zyklischen Betrieb werden zuerst die Schaltzustände an den Eingängen gelesen und im Prozessabbild der Eingänge (PAE) gespeichert. Mit diesen Informationen wird anschließend das Steuerungsprogramm versorgt und abgearbeitet.

Prozessabbild der Eingänge: PAE

- Ein paar Worte zur Wiederholung
- Hier sind die Bits
- Stromfluss im Kontaktplan
- Der SPS-Zyklus



Der SPS-Zyklus (2)

Entsprechend der Schaltlogik im Programm werden die Ausgänge im Prozessabbild der Ausgänge (PAA) beschrieben. Die Zustände im PAA werden im letzten Schritt an die physikalischen Ausgänge übertragen. Der Zyklus beginnt anschließend wieder von vorn.

Prozessabbild
der Ausgänge:
PAA



Ein Zyklus dauert typischerweise zwischen 3 und 10 ms. Die Dauer richtet sich nach Anzahl und Art der verwendeten Anweisungen.

Der Zyklus besteht aus zwei Hauptkomponenten:

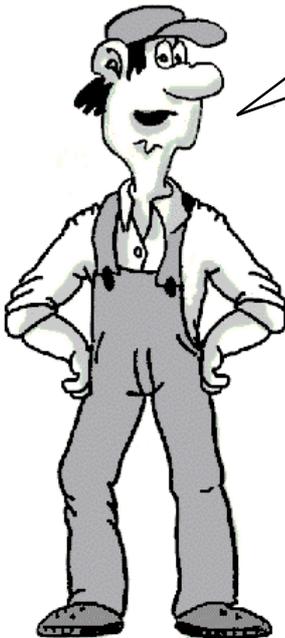
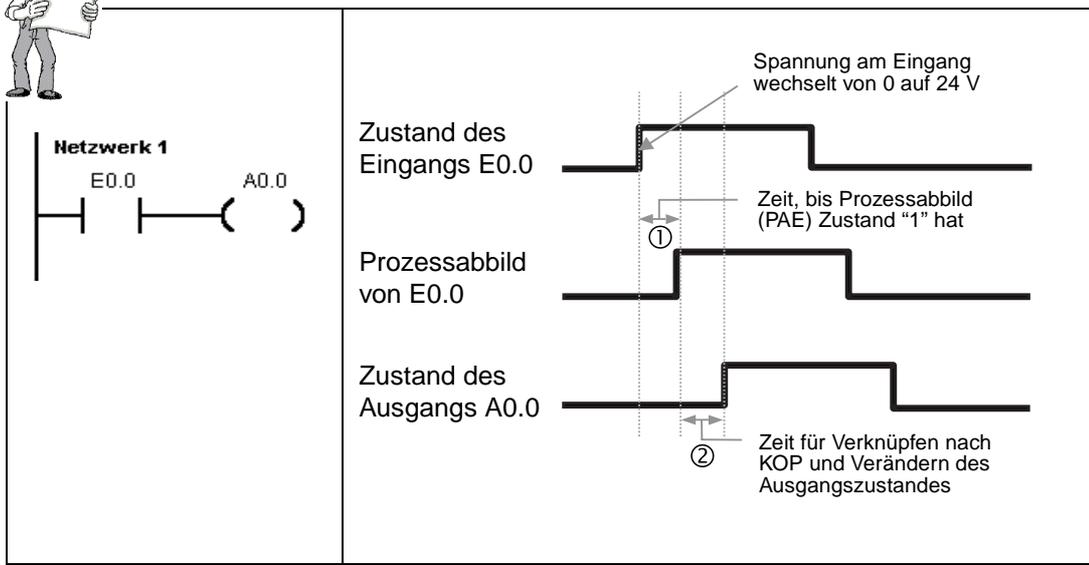
- 1) Betriebssystemzeit, typ. 1 ms; entspricht Phasen ① und ③ (siehe Seite 9).
- 2) Zeit für die Bearbeitung der Befehle; entspricht Phase ② (siehe Seite 9).

Außerdem werden Zyklen nur bearbeitet, wenn das Automatisierungsgerät "arbeitet", sich also im Betriebszustand "RUN" befindet.

- Ein paar Worte zur Wiederholung
- Hier sind die Bits
- Stromfluss im Kontaktplan
- Der SPS-Zyklus



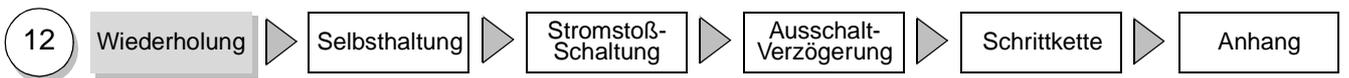
Der SPS-Zyklus (3)



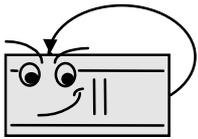
Signaländerungen an Eingängen, die während eines Zyklus stattfinden, werden im nachfolgenden Zyklus in das Eingangs-Register übernommen. Dort werden die Signalzustände für diesen Zyklus "eingefroren". Dies ist das Prozessabbild der Eingänge PAE (siehe ①).
Im nächsten Zyklus werden die übernommenen Zustände nach dem Kontaktplan verknüpft (siehe ②) und die Ausgänge entsprechend der Verknüpfungsergebnisse aktualisiert.

Ausgangsänderung erst am Ende des nachfolgenden Zyklus

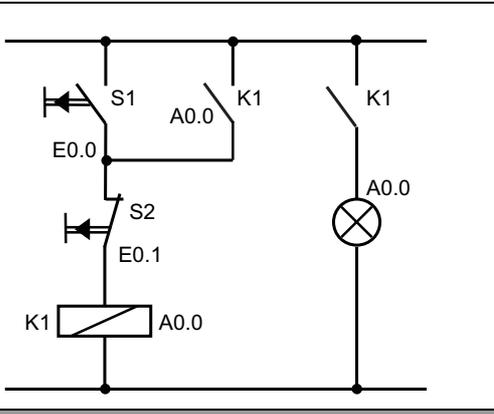
Wiederholung  **Notizen**



- Einführung
- Öffner-Kontakt
- Beschreibung der Lösung und Test
- Selbsthaltung mal anders



Einführung



Hier lernen Sie, wie eine Standard-Selbsthaltung programmiert wird.

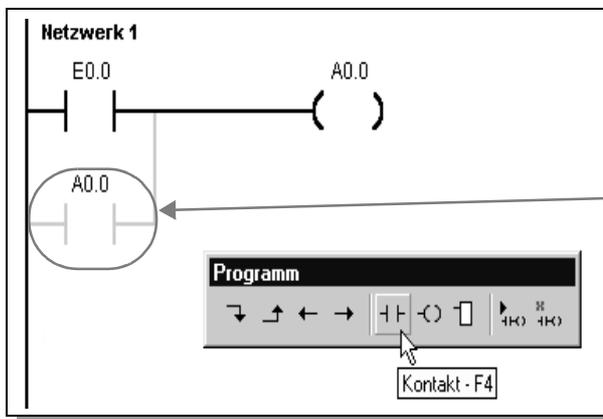
Standard-Selbsthaltung

Das Beispiel hierzu:

Der Ausgang A0.3 soll eingeschaltet werden, sobald S1 an Eingang E0.0 betätigt wird. Durch Selbsthaltung soll A0.0 solange aktiv bleiben, bis S2 an Eingang E0.1 betätigt wird und damit die Selbsthaltung unterbricht.



Öffnen Sie in STEP 7-Micro/WIN das erste Übungsprojekt "a:\d01.mwp" von Diskette. Im Programm fehlen noch ein paar Elemente. Als kleine Übung fügen Sie nun die fehlenden KOP-Elemente ein.



Damit die Selbsthaltung funktioniert, muss der Ausgang (hier A0.0), sobald er aktiviert wurde, selbst dafür sorgen, dass er seinen "wahren" Zustand beibehält und damit aktiviert bleibt.

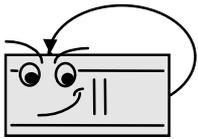
Ausgang A0.0 als Eingang sorgt für Selbsthaltung

Dies erfolgt dadurch, dass der Ausgang (hier A0.0) als Kontakt parallel zum auslösenden Eingang geschaltet wird, ganz wie bei einer herkömmlichen Schütz-Schaltung (A0.0 ist vergleichbar mit unserem Schütz K1).

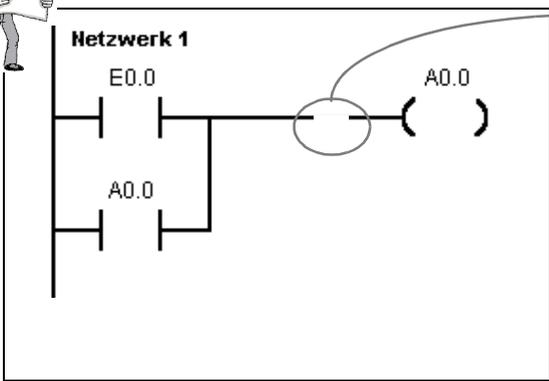
Fügen Sie an der gekennzeichneten Stelle zuerst als Parallelschaltung zu E0.0 einen Kontakt A0.0 ein (grau angedeutet).

- 1) Zur Eingabe des Kontakts klicken Sie mit der linken Maustaste auf das Kontaktplan-Feld und betätigen Sie das Schließerkontakt-Symbol (F4) von STEP 7-Micro/WIN. Wie auf dem Symbol gekennzeichnet, können Sie statt der Maus auch die Funktionstaste F4 verwenden.
- 2) Um die senkrechte Linie einzugeben, markieren Sie mit der Maus das Kontaktplan-Feld von E0.0 und betätigen das entsprechende Symbol oder die Funktionstaste F7.

- Einführung
- Öffner-Kontakt
- Beschreibung der Lösung und Test
- Selbsthaltung mal anders



Öffner-Kontakt (1)



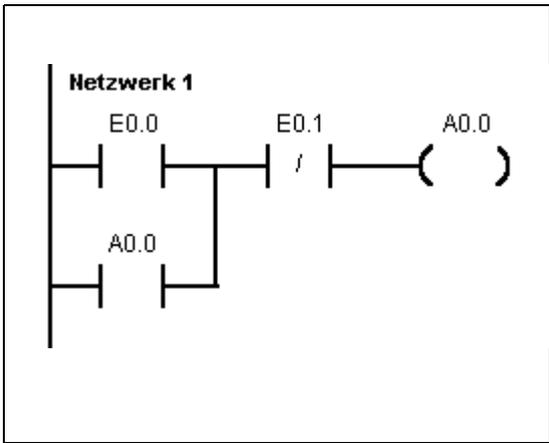
Damit die Selbsthaltung wieder beendet werden kann, soll der Eingang E0.1 bei Betätigung wie eine Unterbrechung des Strompfads wirken. Wird ein Strompfad unterbrochen (d.h. Zustand "0" liegt vor), wenn ein Schalter betätigt wird, spricht man von einem Öffner.

Es muss folglich ein Element eingefügt werden, welches im Kontaktplan als Öffner wirkt, wenn DC 24V ("wahr") am Eingang E0.1 anliegen.

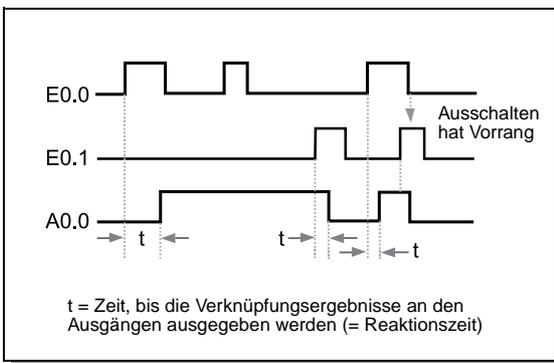
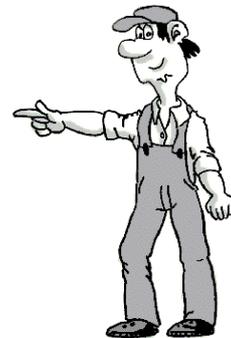
Öffner:



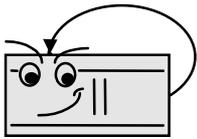
Ergänzen Sie einen Öffner-Kontakt für den Schalter S1 an Eingang E0.1. Dies ist auf der nächsten Seite beschrieben



So sieht die fertige Selbsthaltung aus! Darunter sehen Sie die Funktionsweise im Zeit-Diagramm.



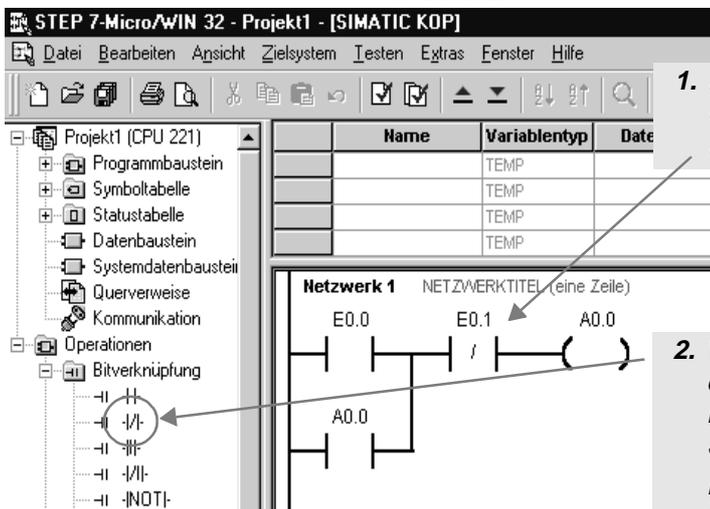
- Einführung
- Öffner-Kontakt
- Beschreibung der Lösung und Test
- Selbsthaltung mal anders



Öffner-Kontakt (2)



Ein Öffner-Kontakt unterbricht den "Stromfluss" im Kontaktplan, wenn der ihm zugeordnete Ein- oder Ausgang "wahr" ist.
Geben Sie den Öffner-Kontakt folgendermaßen ein:



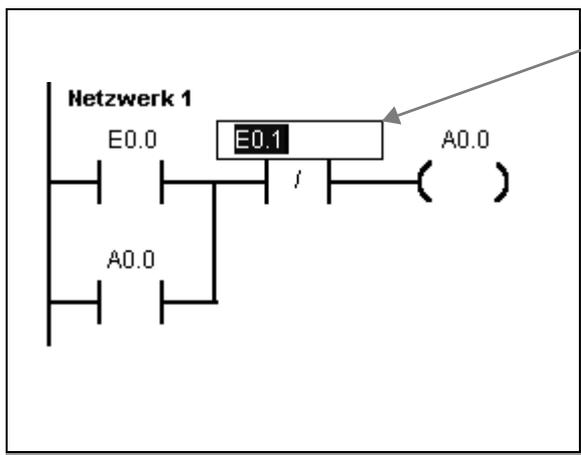
1. Markieren Sie mit einem Mausklick die Stelle, die durch einen Öffner-Kontakt ersetzt werden soll.



2. Wählen Sie mit einem Mausklick den Öffner-Kontakt aus der Kontaktplan-Symboleiste von STEP 7-Micro/WIN.

Der Öffner-Kontakt wird daraufhin im markierten Feld platziert.

3. Zuletzt muss dem Öffner-Kontakt noch das gewünschte Element (hier E0.1) zugewiesen werden. Dies geschieht durch Eingabe in das bereits aktivierte und markierte Textfeld.



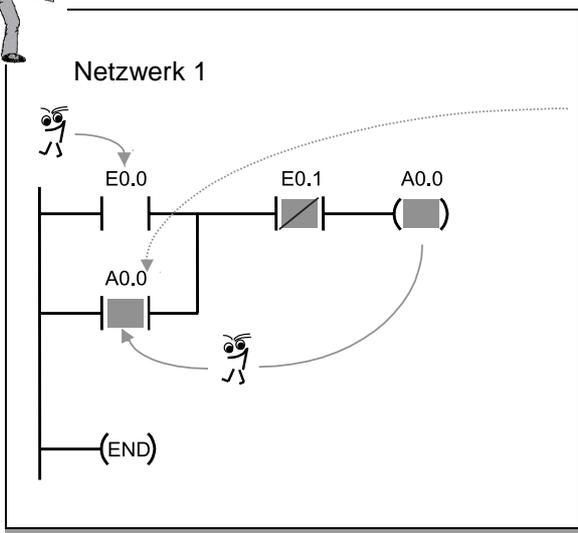
4. Schließen Sie die Textfeld-Eingaben immer durch Betätigung der Eingabetaste ab.



- Einführung
- Öffner-Kontakt
- Beschreibung der Lösung und Test
- Selbsthaltung mal anders



Beschreibung der Lösung und Test



Wie in der Schütz-Schaltung haben Sie auch hier einen Kontakt des Ausgangs (A0.0) parallel zu dem auslösenden Element (E0.0) geschaltet. Ausgang A0.0 parallel zum Eingang hält sich selbst

Wenn in einem Zyklus durch Betätigen des Schalters S1 an E0.0, der Ausgang A0.0 aktiviert wurde, erscheint bereits im darauffolgenden Zyklus (wenige Millisekunden später) der Kontakt A0.0 parallel zu E0.0 geschlossen. Dies bewirkt die Selbsthaltung. Der Öffner E0.1 kann bei Betätigung des damit verbundenen Schalters S2 diese Selbsthaltung beenden.



Speichern Sie Ihr vervollständigtes Programm auf Festplatte. Dadurch können Sie es jederzeit wieder komplett laden und weiter bearbeiten. Wir benötigen das Programm für unser Beispiel "Ausschaltverzögerung" wieder.



Übertragen Sie das Programm anschließend in das Automatisierungsgerät zum Test der Funktion.



Zum Test schalten Sie das Automatisierungsgerät in den Betriebszustand „RUN“.

Testen Sie Ihr Programm durch Betätigung der beiden Schalter des Simulators, die an E0.0 und E0.1 angeschlossen sind.

Beobachten Sie die Leuchten der S7-200 bzw. den KOP-Status.

Beginnen Sie mit dem Einschalten von E0.0.

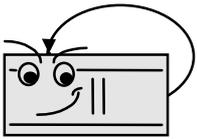
E0.1 muss ausgeschaltet sein. LED an E0.0 muss leuchten.

A0.0 wird dann leuchten.

Sobald E0.1 eingeschaltet wird, wird A0.0 = "0".



- Einführung
- Öffner-Kontakt
- Beschreibung der Lösung und Test
- **Selbsthaltung mal anders**

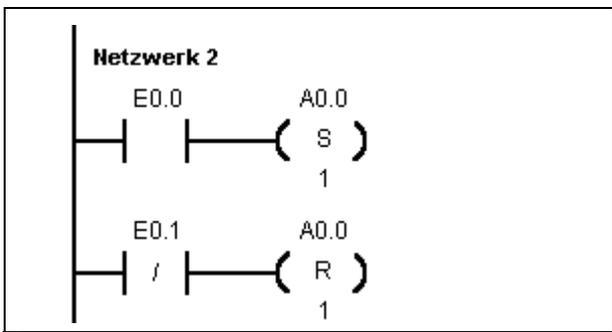


Selbsthaltung mal anders ... (1)



In der Automatisierungstechnik wird die Selbsthaltung oft auch in einer anderen Variante verwirklicht:

Statt einer Rückführung des Ausgangs - wie im vorhergehenden Beispiel - werden hier einfach die Funktionen "Setzen" und "Rücksetzen" verwendet. Betrachten Sie zuerst folgenden Kontaktplan.



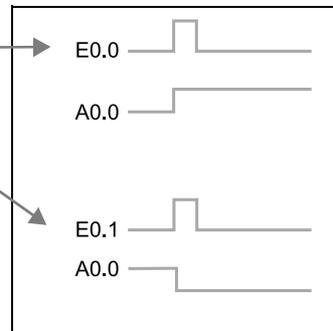
Ein Schaltimpuls an E0.0 bewirkt durch die Operation "Setzen" - (S), dass A0.0 bleibend eingeschaltet wird.

-(S)
Setzen

Ein Schaltimpuls an E0.1 bewirkt dagegen durch die Operation "Rücksetzen" - (R), dass A0.0 wieder ausgeschaltet wird.

-(R)
Rücksetzen

Die "Spulen" - (S) Setzen von A0.0 auf "1"
- (R) Rücksetzen von A0.0 zu "0"
werden in der Automatisierungstechnik häufig genutzt, um Ausgänge oder Merker bei kurzzeitiger Aktivierung (Impuls) durch einen vorgeschalteten Kontakt bleibend ein- bzw. auszuschalten.



-(S) ⇔ 1

-(R) ⇔ 0

Ein "gesetzter" Ausgang oder Merker bleibt so lange "gesetzt", bis er durch die - (R) Anweisung wieder zurückgesetzt ("unwahr") wird.

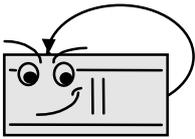
Wert Setzen mit (S)
Rücksetzen mit (R)

Liegt an der Setz- **und** an der zugehörigen Rücksetz-Spule eines Ausganges das Signal "1", hat die in der Reihenfolge letzte Operation im Programm Vorrang.

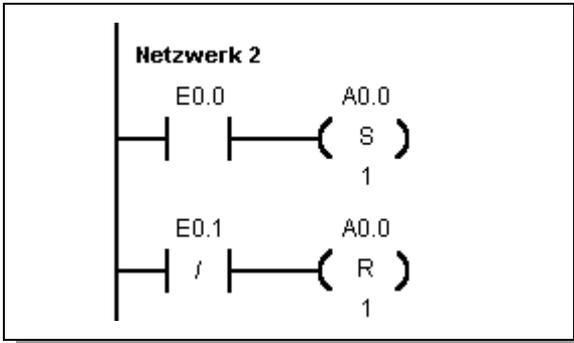
Letzte Operation im Zyklus hat Vorrang



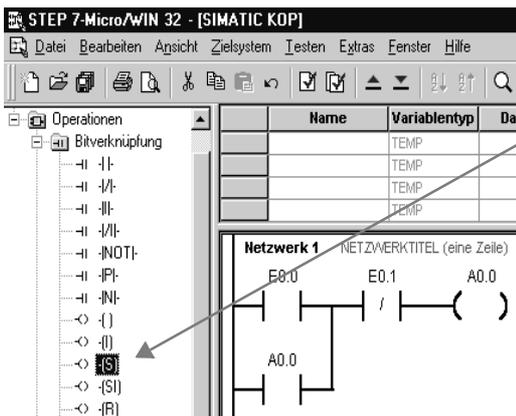
- Einführung
- Öffner-Kontakt
- Beschreibung der Lösung und Test
- **Selbsthaltung mal anders**



Selbsthaltung mal anders ... (2)



Wie Sie E0.0 und E0.1 eingeben, haben Sie bereits gelernt. Geben Sie die Setz- und Rücksetz-Spule wie folgt ein:



1. Nach Markieren des gewünschten Feldes wählen Sie mit einem Mausklick “-(S)-” aus der Liste für Operationsfamilien.

☞ Markieren

☞ -(S)-

2. Tragen Sie in das bereits aktivierte Textfeld die zu beeinflussende Ausgangsadresse, hier A0.0, ein.

☞ Adresse

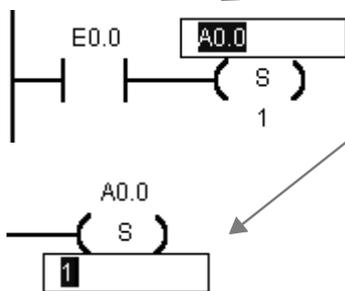
☞ Enter

3. Abschließend legen Sie noch fest, welche Anzahl (max. 255) Ausgänge, bzw. Merker ab der gewählten Adresse gleichartig beeinflusst werden sollen (hier nur ein Bit).

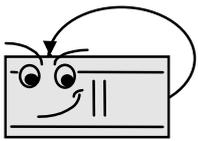
Mit einem Befehl bis zu 255 Ausgänge, Merker, Zeiten setzen (S) oder zurücksetzen (R)

☞ Anzahl (1...255)

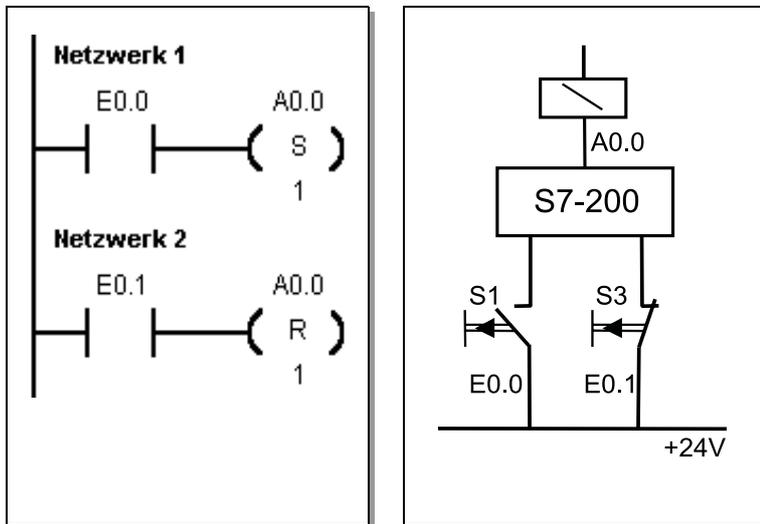
☞ Enter



- Einführung
- Öffner-Kontakt
- Beschreibung der Lösung und Test
- Selbsthaltung mal anders



Sicherheitsaspekte Abschalten bei Drahtbruch an Verbindung zu S3 (3)



Schalter mit Öffner-Kontakt, der beim Betätigen das Signal "0" liefert.

Im KOP wird dieses Signal durch den Öffner-Kontakt E0.1 ---|/|--- umgekehrt.

D.h. wenn Sie den Schalter S3 betätigen, wird A0.0 zurückgesetzt.



Hinweise zur Sicherheit

- Zum Rücksetzen wurde in obiger Lösung ein Öffner-Schalter S3 verwendet. Bei Betätigung von E0.0 wird der Ausgang A0.0 bleibend gesetzt. Liegt an E0.1 +24 V an, liefert der "Öffner-Kontakt" im KOP den Zustand "0". Der Ausgang A0.0 wird nicht zurückgesetzt. Der KOP-"Stromfluss" ist unterbrochen und die Spule zum Rücksetzen ist deaktiviert. Liegt kein Signal (0V) an E0.1 (S3 ist geöffnet), ist der Öffner von E0.1 im KOP = "1" und der Ausgang wird zurückgesetzt.

Bei Verwendung eines Öffner-Schalters an E0.1 wird der selbsthaltende Ausgang A0.0 zurückgesetzt (wieder ausgeschaltet):

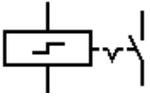
- wenn der Schalter S3 betätigt wird (E0.1 = "0") oder
 - wenn eine Verbindungsleitung zwischen E0.1 und dem Öffner-Schalter bricht. Auch bei Drahtbruch ist dadurch gewährleistet, dass zum Beispiel ein in Selbsthaltung betriebener Anlagenteil, z.B. Motor, abgeschaltet wird.
- Die Operation "Rücksetzen von A0.0" wurde in der Reihenfolge nach der Operation 'Setzen von A0.0' eingegeben, da hiermit im Falle gleichzeitiger Betätigung der beiden Schalter die Beendigung der Selbsthaltung Vorrang hat.



Öffnen Sie in STEP 7-Micro/WIN das fertige Übungsbeispiel "a:ld02.mwp" von Diskette und probieren Sie die Funktionen.

Selbsthaltung ► **Notizen**

- Einführung
- Lösung im Überblick
- Flankenerfassung
- Merker
- Beschreibung der Lösung und Test

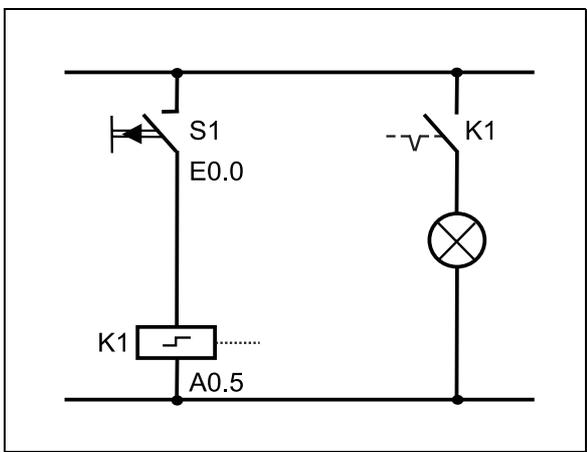


Einführung



Hier werden Sie eine sogenannte Stromstoß-Schaltung realisieren. In diesem Zusammenhang werden Sie die Flankenerkennung und Merker kennenlernen.

Funktionsweise



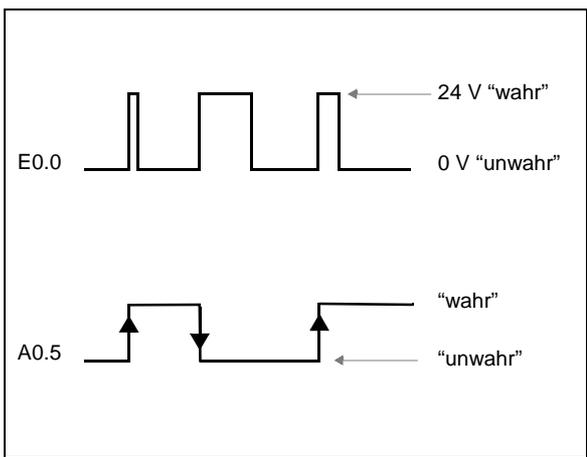
Eine Leuchte an Ausgang A0.5 soll eingeschaltet werden, sobald S1 an Eingang E0.0 kurzzeitig betätigt wird.

Wird S1 (E0.0) erneut betätigt, soll A0.5 abfallen und die Leuchte erlöschen.

Bei jeder Betätigung des Schalters S1 soll A0.5 seinen Zustand umkehren.

Es handelt sich um eine sogenannte "Stromstoß-Schaltung".

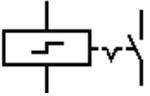
Zeitdiagramm



Ausgang A0.5 soll immer, wenn der Schalter an E0.0 von "geöffnet" zu "geschlossen" übergeht, seinen aktuellen Zustand **einmal** umkehren.

Bleibt der Schalter geschlossen oder offen, findet keine Änderung statt.

- Einführung
- Lösung im Überblick
- Flankenerfassung
- Merker
- Beschreibung der Lösung und Test



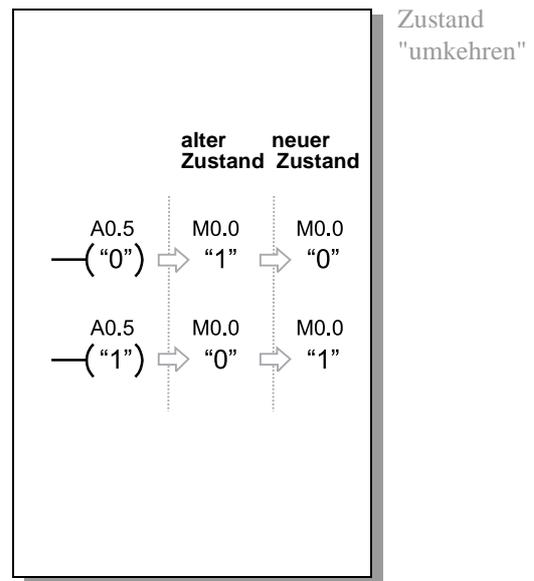
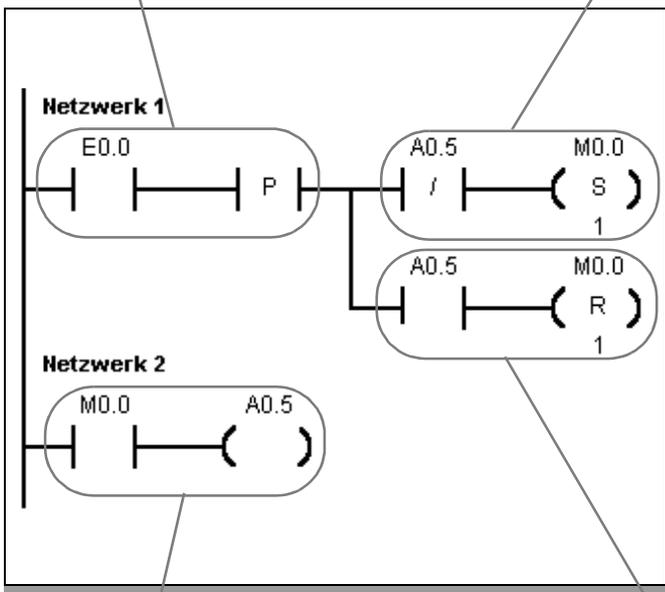
Lösung im Überblick



Wir zeigen Ihnen vor der stufenweisen Lösung der Aufgabe zunächst die fertige Lösung, um Ihnen dadurch einen Überblick zu geben.

Erfassen, ob an E0.0 eine Zustandsänderung von "0" nach "1" (= positive Flanke) erfolgt ist.

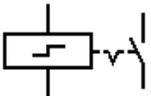
Wenn Ausgang A0.5 "0" ist, wird Merker M0.0 gesetzt. Dies "merkt vor" dass A0.5 in Netzwerk 2 "1" werden soll.



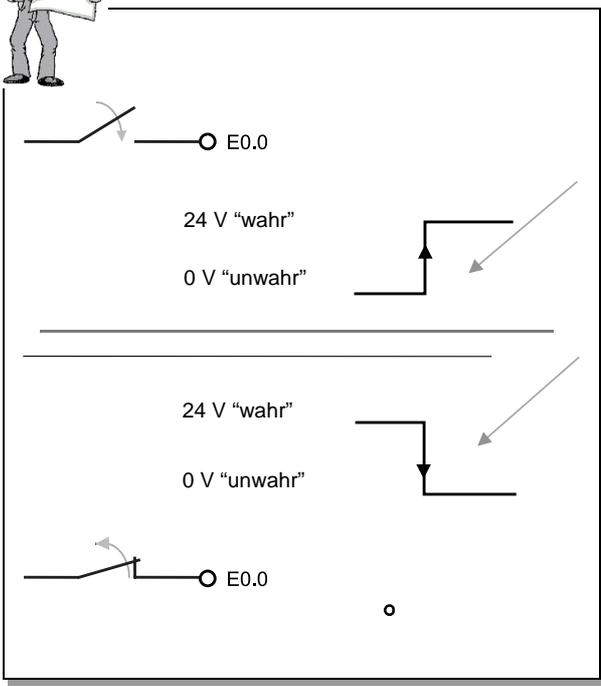
Zustand von Merker M0.0 dem Ausgang A0.5 zuweisen.

Wenn Ausgang A0.5 "1" ist, wird Merker M0.0 zurückgesetzt. Dies "merkt vor", dass A0.5 in Netzwerk 2 "0" werden soll.

- Einführung
- Lösung im Überblick
- Flankenerfassung
- Merker
- Beschreibung der Lösung und Test



Flankenerfassung (1)



Der Moment des Überganges eines Kontaktes (Eingang, Ausgang ...) von "offen" zu "geschlossen" bzw. von "unwahr" zu "wahr" wird als steigende oder positive Flanke bezeichnet.

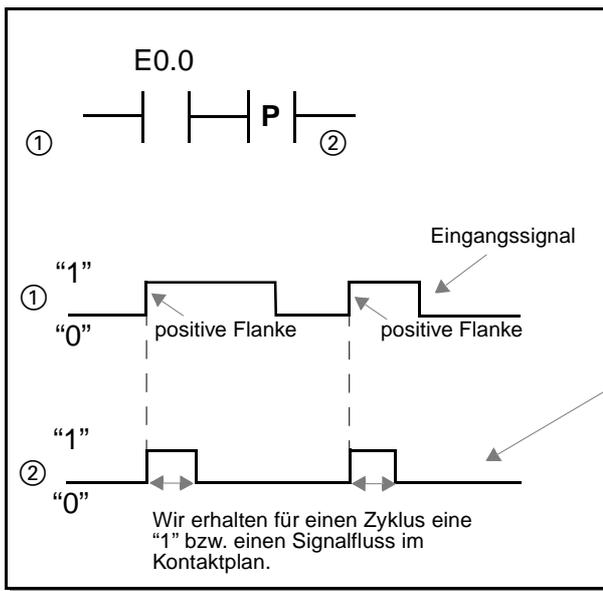


Entsprechend wird der Übergang von "geschlossen" zu "offen" bzw. von "wahr" zu "unwahr" als fallende oder negative Flanke bezeichnet.



Für die Erfassung dieser Zustände werden die zwei Funktionen $\neg P$ (steigende Flanke) und $\neg N$ (fallende Flanke) von der S7-200 bereitgestellt.

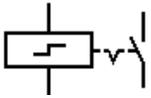
In unserem Beispiel benutzen wir die $\neg P$ - Funktion wie folgt:



Und so sieht das Signal aus, das die $\neg P$ Funktion erzeugt.



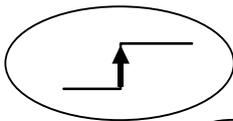
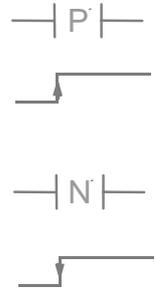
- Einführung
- Lösung im Überblick
- Flankenerfassung
- Merker
- Beschreibung der Lösung und Test



Flankenerfassung (2)

Der Kontakt --|P| zur Erkennung steigender Flanken wird beim Übergang von "unwahr" zu "wahr" des vorgeschalteten Kontakts für die Dauer eines Zyklus geschlossen.

Entsprechend ist der Kontakt --|N| zur Erkennung fallender Flanken bei Übergängen von "wahr" zu "unwahr" für die Dauer eines Zyklus geschlossen.

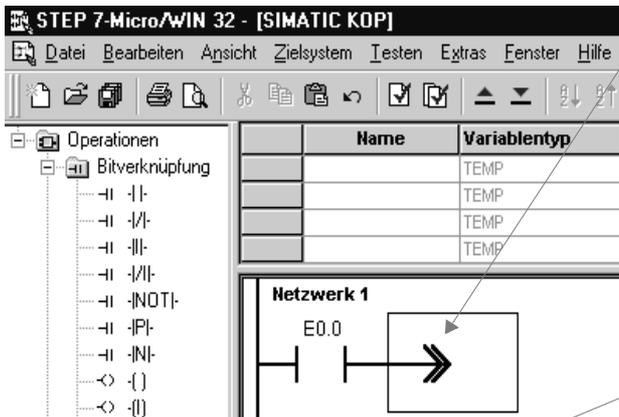


In unserer Übung "Stromstoß-Schaltung" wird --|P| also verwendet, um nur im Moment der Betätigung des Tasters an E0.0 ein Signal an die nachfolgenden Verknüpfungen weiterzureichen.

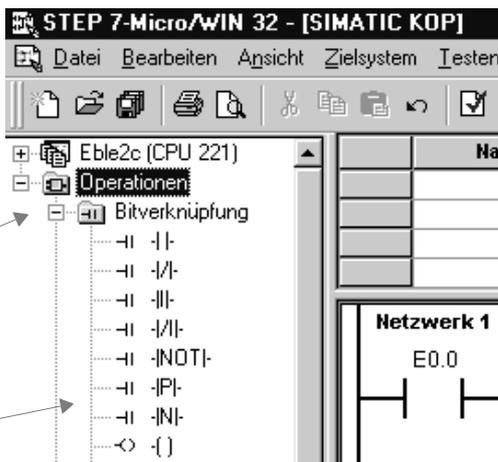
Und so wird es eingegeben:



Öffnen Sie in STEP 7-Micro/WIN das Übungsprojekt "a:\d03.mwp" von Diskette. Auch dieses Projekt ist noch unvollständig und wird Schritt für Schritt ergänzt.



1. Markieren Sie mit einem Mausklick die Stelle, die durch eine Flankenerkennung ersetzt werden soll.



☞ Markieren

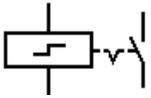
2. Markieren Sie mit einem Mausklick "Bitverknüpfung" im Operationsbaum.

☞ Markieren

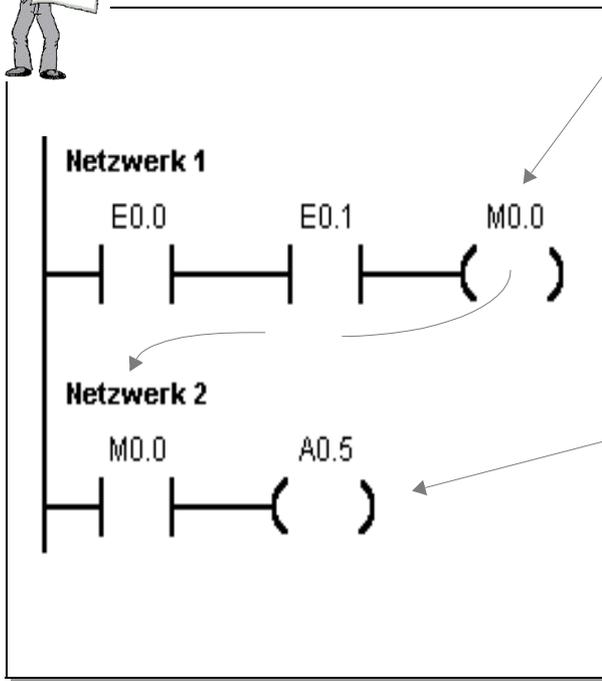
3. Aus der danach erscheinenden Liste für Operationen wählen Sie "Steigende Flanke" |P| bzw. "Fallende Flanke" |N|.

☞ Flanke

- Einführung
- Lösung im Überblick
- Flankenerfassung
- **Merker**
- Beschreibung der Lösung und Test



Merker (1)



Für die Stromstoß-Schaltung benötigen Sie Merker. Wie man mit Merkern arbeitet, wird hier an einem Beispiel kurz gezeigt.

Der Bit-Merker "M0.0" wird statt als Ausgang als SPS-internaler Ablageplatz (Speicher) für das Zwischenergebnis der Verknüpfung "E0.0 UND E0.1" genutzt.

In diesem Netzwerk wird der Bit-Merker als "Eingangs-Schließer" verwendet und steuert damit den Ausgang A0.5. Der Merker kann noch an beliebig anderen Stellen im Programm verwendet werden.

Beliebig oft als Öffner oder Schließer verwendbar

Merker werden, ähnlich wie der Speicher eines Taschenrechners, zum Speichern von Zwischenergebnissen genutzt.

Merker werden in der SPS-Technik wie Ausgänge verwendet und sind in ihrer Wirkung mit Hilfsschützen vergleichbar. Ein Merker kann an beliebigen Stellen beliebig oft als Öffner oder Schließer verwendet werden.

Verwendung wie Ausgänge

Wirkung eines Hilfsschützen

Bei einer Unterbrechung der Betriebsspannung geht der Merkerinhalt verloren. Um dies zu verhindern, gibt es "Remanenz."

Der Inhalt von Merkern steht sofort (im gleichen Zyklus) für nachfolgende Verknüpfungen bereit.

Inhalt sofort aktualisiert

Merker werden verwendet, wenn das (Zwischen-) Ergebnis eines Netzwerks in anderen Netzwerken weiterverarbeitet werden soll (wie Zwischensummen beim handschriftlichen Addieren von Zahlen) oder um ausgewertete Folgezustände zwischenzuspeichern.

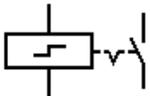
Mit -(S) oder -(R) mehrmals beschreibbar
Mit -()- nur **einmal** zuweisen



Stromstoß-Schaltung

Stromstoß-Schaltung

- Einführung
- Lösung im Überblick
- Flankenerfassung
- **Merker**
- Beschreibung der Lösung und Test



Merker (2)

Sie kennen nun die Funktion von Merkern und können damit die Lösung der Stromstoß-Schaltung verstehen.

Die $\neg|P|$ Funktion ermöglicht pro Tastendruck an E0.0 für einen Zyklus den Signalfluss (Flankenerfassung) im Netzwerk 1.



Pro $\neg|P|$ Flanke soll A0.5 seinen Zustand umkehren.

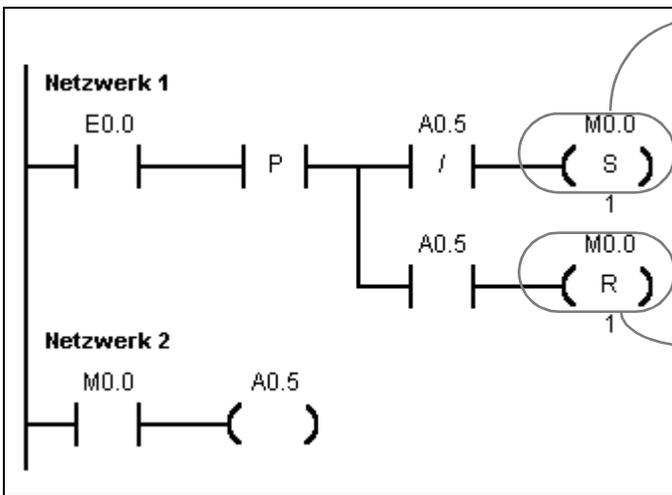
Wir schreiben den umgekehrten Zustand (Folgezustand) nicht direkt in den Ausgang A0.5, weil der im "oberen" Zweig soeben gesetzte Ausgang im "unteren" Zweig sofort wieder zurückgesetzt würde. Wir schreiben den Folgezustand deshalb in den Merker M0.0 (= Rettung vor Überschreiben).

Im Netzwerk 2 wird der "gesetzte" Zustand des Merkers dem Ausgang zugewiesen.

-(S)
Setzen

-(R)
Rücksetzen

Folgezustand vor Überschreiben "retten"



An dieser Stelle muss eine Spule zum Setzen des Merkers M0.0 platziert werden. Die Zahl unter der Spule gibt an, wieviele Elemente ab der angegebenen Anfangsadresse gesetzt werden sollen. Hier: Setzen eines Bits ab Merker M0.0.

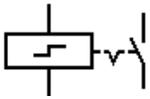
M0.0 wird gesetzt, wenn A0.5 nicht aktiv ("unwahr") war

Da der untere Zweig die umgekehrte Funktion des oberen Zweiges realisiert, muss das Bit von Merker M0.0 zurückgesetzt, also ausgeschaltet werden, wenn bei einem Tasterdruck dieser Zweig "stromführend" ist.

M0.0 wird zurückgesetzt, wenn A0.5 aktiv ("wahr") war

Ergänzen Sie abschließend das Beispiel in Ihrem aktuellen Übungsprojekt in STEP 7-Micro/WIN wie oben gezeigt.

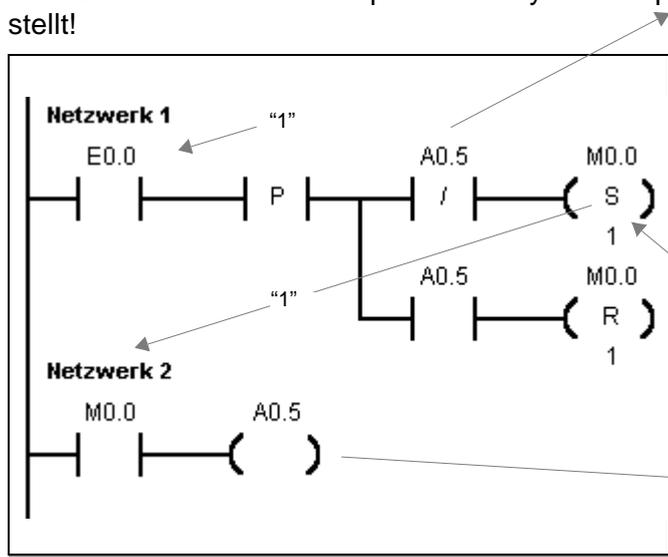
- Einführung
- Lösung im Überblick
- Flankenerfassung
- Merker
- Beschreibung der Lösung und Test



Beschreibung der Lösung und Test

Zusammenfassend wird am Beispiel des oberen Zweiges von Netzwerk 1 (endet mit (S), Einschaltvorgang) nochmals die Funktion unseres nun vollständigen Programmes erläutert:

Der "Stromfluss" im Kontaktplan ist im Zyklus der positiven Flanke an E0.0 dargestellt!



Wenn E0.0 gerade betätigt wird (-IPI- Flankenerkennung)

und

A0.5 im aktuellen Zyklus "0" ist (oberer Zweig ist durch Abfrage mit Öffner-Kontakt wahr),

dann...

Folgezustand von A0.5 durch Setzen von Merker M0.0 vormerken: **-(S)** Setzen eines Bits ab M0.0.

M0.0 hat hier schon den Folgezustand von A0.5.

A0.5 erhält den neuen Zustand erst am Ende des Zyklus zugewiesen und erscheint deshalb erst im nächsten Zyklus als "wahr" oder "1" in der KOP-Darstellung.



Speichern Sie das fertige Programm auf Festplatte.



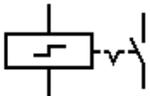
Übertragen Sie das Programm in die SPS.



Zum Testen schalten Sie die SPS in den Betriebszustand "RUN".

Testen Sie Ihr Programm: Betätigen Sie den Schalter an E0.0 und beobachten Sie dabei den Ausgang A0.5.

- Einführung
- Lösung im Überblick
- Flankenerfassung
- Merker
- Beschreibung der Lösung und Test



Zeigen Sie jetzt, was Sie bereits wissen

... denn inzwischen haben Sie einiges dazugelernt!

Lesen Sie sich deshalb die folgenden Fragen durch und beantworten Sie diese.

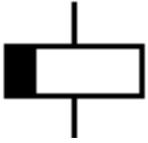
- ✓ Was ist der Zyklus einer SPS?
Welche drei Hauptbestandteile sind im "Zyklus" enthalten? Siehe Seite 9
- ✓ Wie wird in der SPS-Technik eine Selbsthaltung realisiert? Siehe Seite 13
- ✓ Öffner-Kontakt: Wie sieht er im Kontaktplan (KOP) aus, was bewirkt er und welche Sicherheitsmaßnahme kann damit erreicht werden? Siehe Seite 14
- ✓ Was ist eine Flanke, womit wird diese festgestellt und wozu? Siehe Seite 23
- ✓ Was sind Merker und wozu werden diese gebraucht? Siehe Seite 25
- ✓ Wie werden die Spulen "Setzen" und "Rücksetzen" eingegeben und was bewirken diese? Siehe Seite 26



Diese Fragen konnten Sie sicher leicht beantworten. Auch wenn Sie dazu noch einmal auf den entsprechenden Seiten nachgeschlagen haben.

Spätestens jetzt haben Sie alles verstanden.

- Einführung
- Speichern unter ...
- Netzwerk einfügen
- Beschreibung der Lösung
- Kommentare eingeben

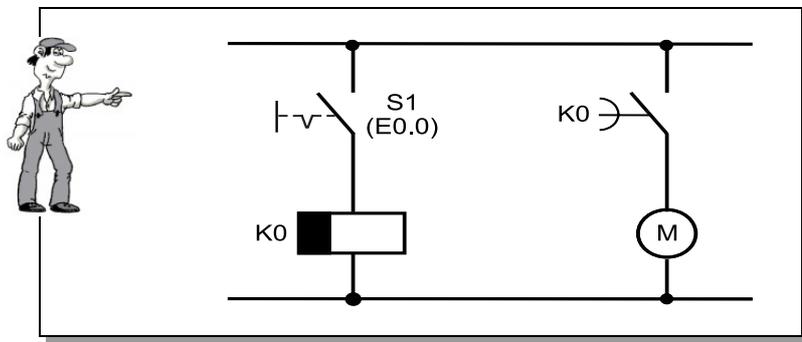


Einführung

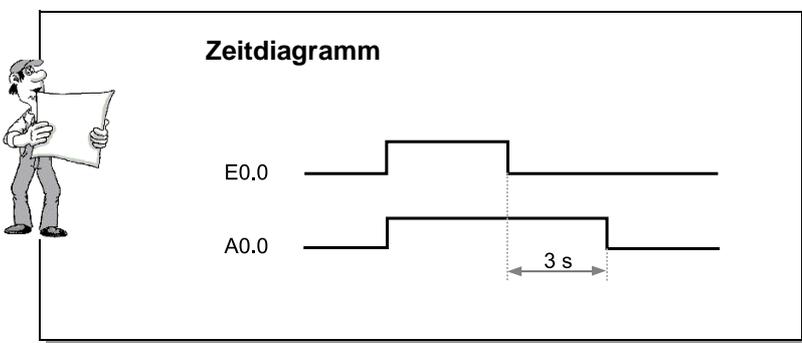


Die Einschaltverzögerung kennen Sie bereits aus der 1-Stunden-Fibel. Wir realisieren nun gemeinsam eine Ausschaltverzögerung.

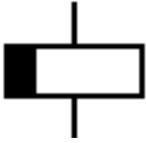
Betätigen von S1 (E0.0) aktiviert einen Lüftermotor an Ausgang A0.0. Wenn S1 (E0.0) abgeschaltet wird, soll der Lüfter noch 3 Sekunden weiterlaufen und dann stoppen.



Wird S1 abgeschaltet, soll der Lüfter 3 Sekunden weiterlaufen



- Einführung
- Speichern unter ...
- Netzwerk einfügen
- Beschreibung der Lösung
- Kommentare eingeben



Einführung



Vorgehensweise

- 1) Zuerst laden Sie die vervollständigte Selbsthalteschaltung aus unserem ersten Beispiel von der Festplatte.
- 2) Anschließend speichern Sie das Beispiel unter einem neuen Namen auf der Festplatte ab.
- 3) Danach schaffen Sie Platz durch "Netzwerk einfügen"
- 4) Abschließend vervollständigen Sie die Ausschaltverzögerung und Kommentierung.
- 5) Zuletzt werden Sie das Programm testen.

Auf den nächsten Seiten werden wir alle Schritte gemeinsam bearbeiten, um die Ausschaltverzögerung sicher zu realisieren.

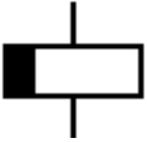
Wir wünschen Ihnen viel Erfolg.



Ausschaltverzögerung

Ausschaltverzögerung

- Einführung
- **Speichern unter ...**
- Netzwerk einfügen
- Beschreibung der Lösung
- Kommentare eingeben



Speichern unter ...



Als Grundlage für unser Projekt werden wir die Selbsthaltungsschaltung aus dem ersten Kapitel nutzen. Das komplette Projekt duplizieren Sie, indem Sie es laden und gleich darauf unter einem neuen Namen speichern.



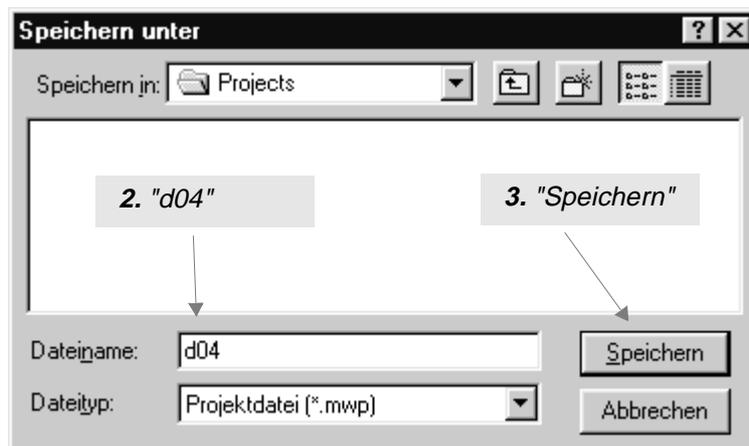
Laden Sie in STEP 7-Micro/WIN Ihr Projekt "d01.mwp" (Selbsthaltungsschaltung) von der Festplatte. Sie haben es im ersten Kapitel dort abgespeichert.

Sie möchten das Projekt jetzt unter einem neuen Namen ablegen. Speichern Sie das Projekt wie nachfolgend beschrieben unter der Bezeichnung "d04.mwp".



1. Rufen Sie die Menü-Funktion "Datei >Speichern unter..." auf.

→ Menü:
Datei,
Speichern
unter...



d04.mwp
Speichern

Wiederholung

Selbsthaltung

Stromstoß-Schaltung

Ausschalt-Verzögerung

Schritt看te

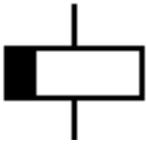
Anhang

31

Ausschaltverzögerung

Ausschaltverzögerung

- Einführung
- Speichern unter ...
- **Netzwerk einfügen**
- Beschreibung der Lösung
- Kommentare eingeben



Netzwerk einfügen

Damit wir die Ausschaltverzögerung realisieren können, muss an der Stelle von Netzwerk 2 ein zusätzliches Netzwerk eingefügt werden. Dazu sind folgende Schritte nötig:

1. Aktivieren Sie durch einen einfachen Mausklick das Titelfeld von Netzwerk 2.

2. Lassen Sie anstelle von Netzwerk 2 ein neues Netzwerk einfügen.

Netzwerk einfügen

Markieren

Netzwerk-Schaltfläche in Werkzeugleiste (F10)

Sie haben Platz geschaffen für das neue Netzwerk 2, welches Sie für die Realisierung der Ausschaltverzögerung verwenden werden. Die Inhalte des ursprünglichen Netzwerks 2 sind um ein Netzwerk "weitergewandert".

Hinweis:

Um Platz für die Eingabe von KOP-Elementen zu schaffen, gibt es auch noch folgende Möglichkeit:

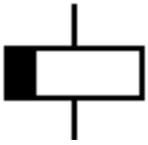
3. Wählen Sie aus dem Menü Bearbeiten den Unterpunkt "Inhalte einfügen".

Menü: Bearbeiten, Inhalte einfügen/Netzwerk

Name	Variablentyp	Da
	TEMP	
	TEMP	
	TEMP	

4. "Netzwerk" selektieren.

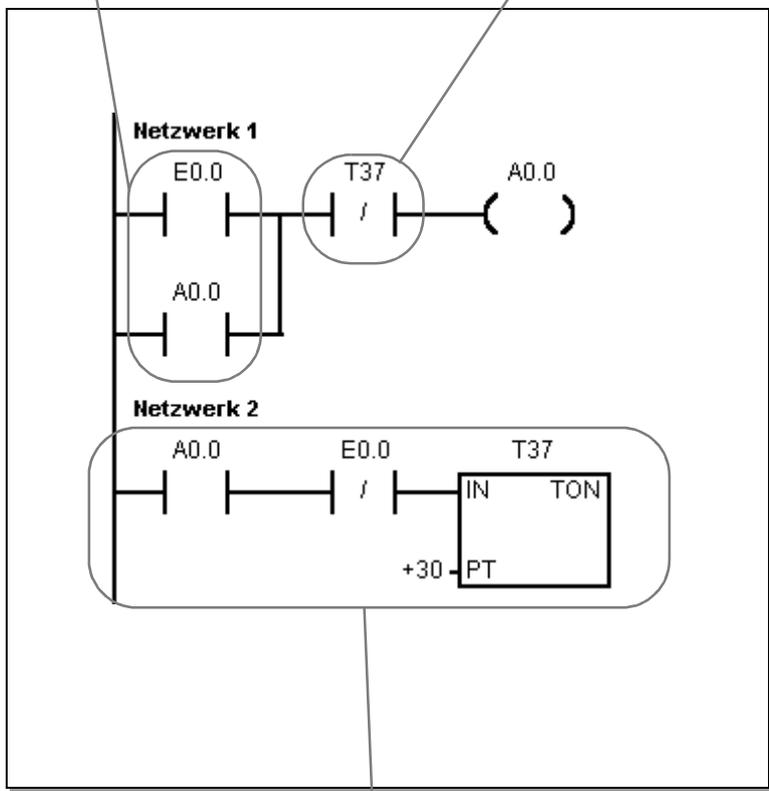
- Einführung
- Speichern unter ...
- Netzwerk einfügen
- **Beschreibung der Lösung**
- Kommentare eingeben



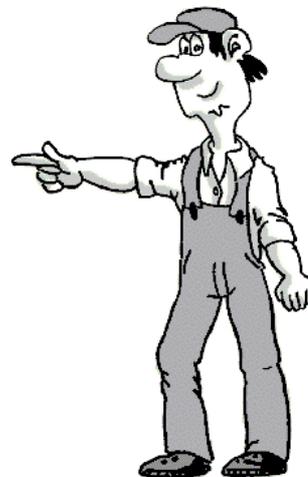
Lösung im Überblick

E0.0 aktiviert A0.0.
A0.0 hält sich selbst, da er auch gleichzeitig parallel zu E0.0 geschaltet ist.

Wenn T37 abgelaufen ist, wird die Selbsthaltung über diesen Kontakt aufgetrennt. Der Motor stoppt.
Ist T37 nicht abgelaufen, "hält" die Selbsthaltung.

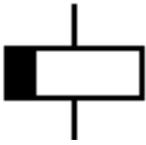


So sieht das fertige Programm aus.



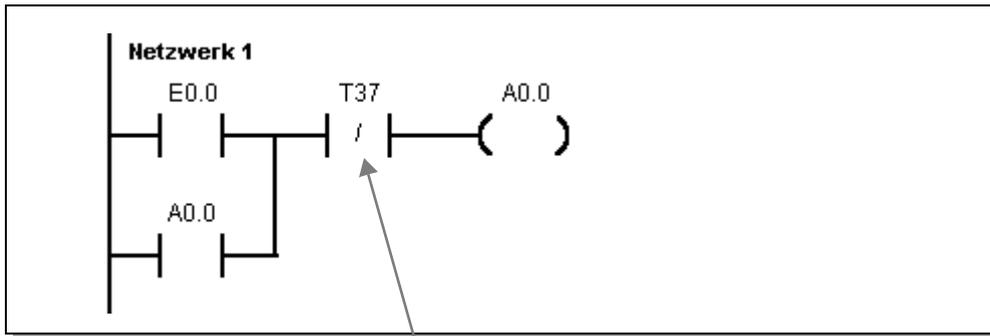
Wenn A0.0 betätigt ist und E0.0 wieder "0" ist (S1 ist nicht mehr betätigt), beginnt die Zeit T37 zu laufen.

- Einführung
- Speichern unter ...
- Netzwerk einfügen
- **Beschreibung der Lösung**
- Kommentare eingeben



Lösung - Programm eingeben

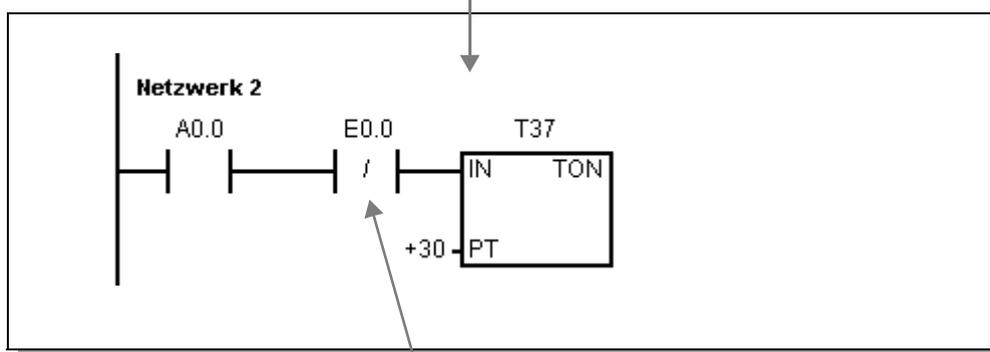
Netzwerk 1 muss so aussehen:



Überschreiben Sie E0.1 der Selbsthalteschaltung mit T37.

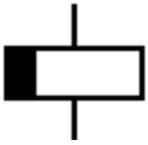
Tragen Sie in Netzwerk 2 folgendes Programm ein:

T37 geben Sie ein mit:
F2 Zeiten/Zähler und
F3 Zeit als Einschaltverzögerung.



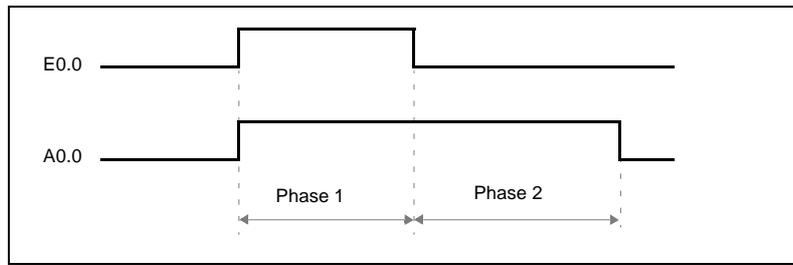
T37 hat eine Zeitbasis von 100 ms (siehe auch "1-Stunden-Fibel", Seite 36).
Der Zeitwert ist deshalb $30 * 100 \text{ ms} = 3 \text{ s}$.

- Einführung
- Speichern unter ...
- Netzwerk einfügen
- **Beschreibung der Lösung**
- Kommentare eingeben

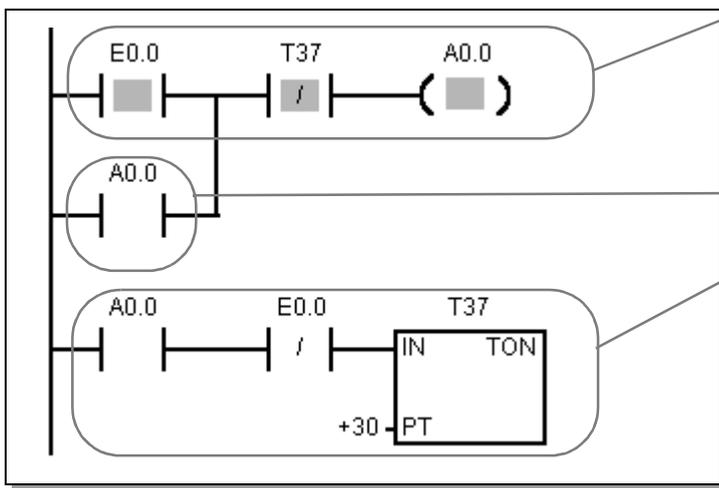


Beschreibung der Lösung

So funktioniert unser Programm. Es hat zwei aktive Phasen.

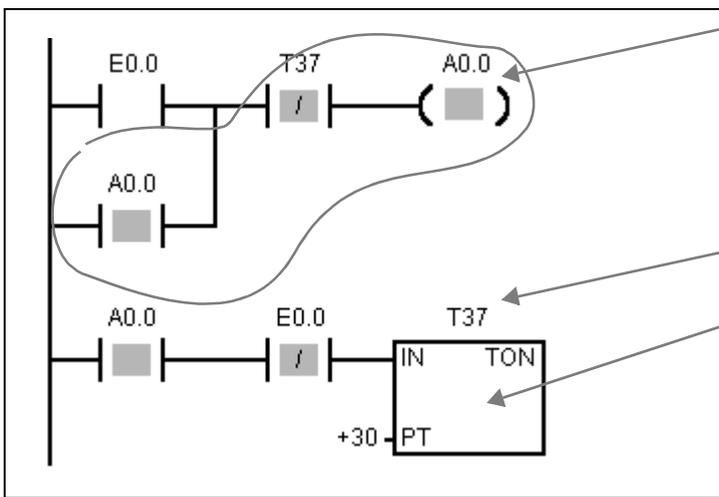


Phase 1: Aktivieren der Selbsthaltung, E0.0 ist "1"
(wir gehen davon aus, dass A0.0 nicht aktiv ist).



Wenn E0.0 betätigt wird UND T37 nicht abgelaufen ist DANN wird A0.0 aktiviert (= "1"). Über diesen Kontakt hält sich A0.0 selbst. T37 läuft noch nicht, weil E0.0 noch "1" ist.

Phase 2:



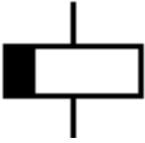
E0.0 ist nicht mehr betätigt. Die Selbsthaltung bleibt solange bestehen, bis T37 abgelaufen ist. Solange die Zeit läuft, ist T37 "0" und der Öffner auf Stromdurchlass. Im Testbetrieb sieht man hier die Zeit laufen. Wenn A0.0 aktiv ist UND E0.0 nicht mehr betätigt ist, dann läuft die Zeit T37.



Ausschaltverzögerung

Ausschaltverzögerung

- Einführung
- Speichern unter ...
- Netzwerk einfügen
- Beschreibung der Lösung
- Kommentare eingeben

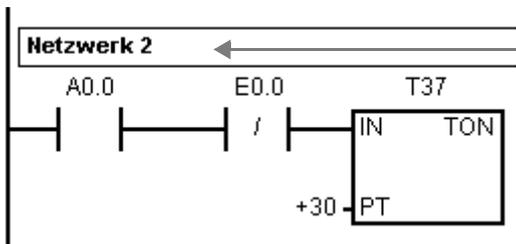


Kommentare eingeben (1)



Speichern und probieren Sie Ihr neues Programm aus. Wenn Sie E0.0 betätigen, wird A0.0 aktiviert. Wenn E0.0 ausgeschaltet wird, erlischt A0.0 nach 3 Sekunden.

Gut gemacht! Vielleicht wüssten Sie sich auch schon, dass es für spätere Arbeiten (Änderungen o.a.) hilfreich wäre, im Programm Anmerkungen zur Funktionsweise zu hinterlegen. Klar, auch daran haben wir gedacht. Dafür gibt es die Möglichkeit, Titel und Kommentare zu jedem Netzwerk zu ergänzen. Ich zeige Ihnen, wie dies geht.

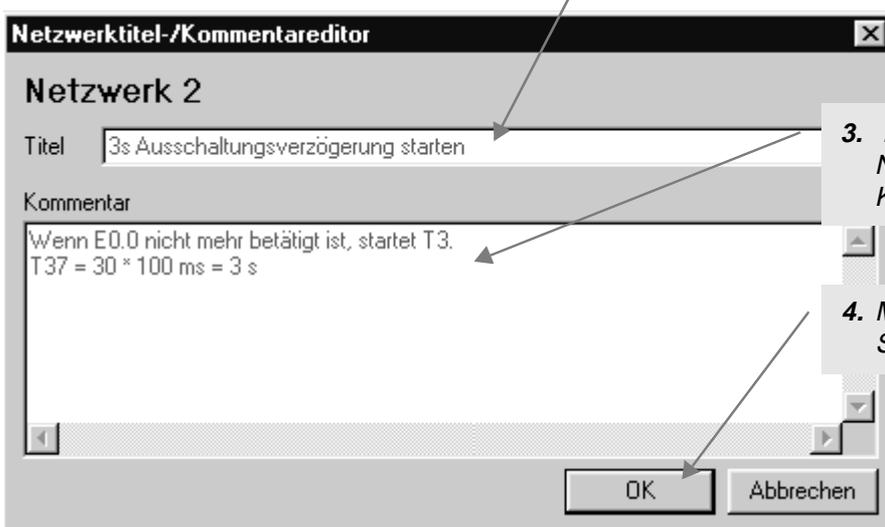


1. Klicken Sie doppelt auf das Titelfeld von Netzwerk 2.

2x

2. Der Kommentar-Editor wird jetzt angezeigt. Tragen Sie hier den Netzwerktitel ein ...

Titel



3. ... und hier den Netzwerktitel-Kommentar.

Kommentar

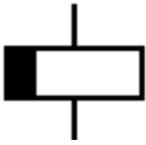
4. Mit OK schließen Sie die Eingabe ab.

OK

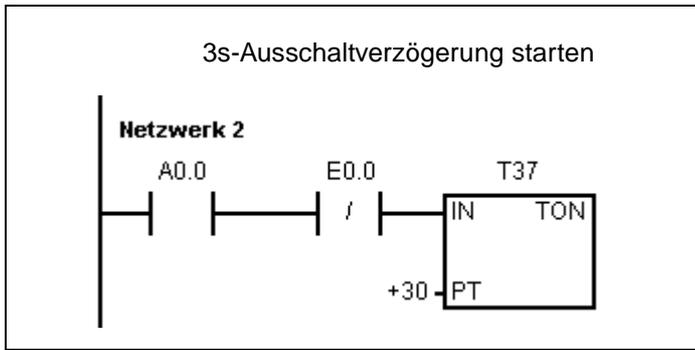
Ausschaltverzögerung

Ausschaltverzögerung

- Einführung
- Speichern unter ...
- Netzwerk einfügen
- Beschreibung der Lösung
- **Kommentare eingeben**

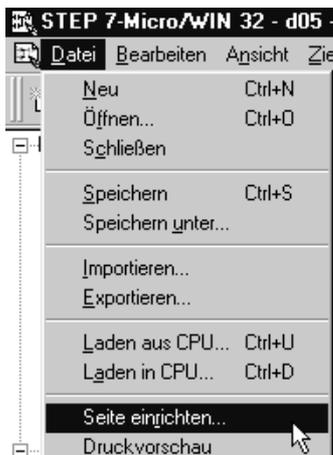


Kommentare eingeben (2)



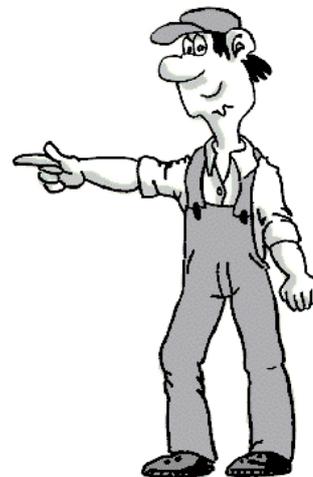
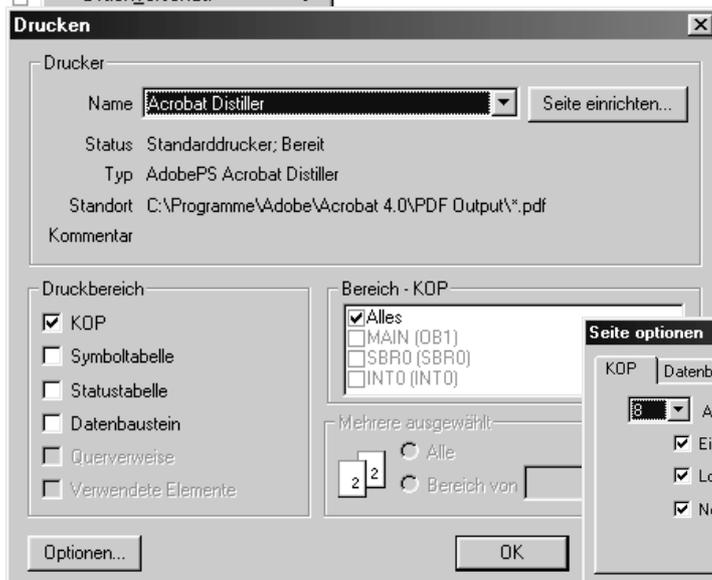
Nach dem Hinzufügen des Kommentars ist am Bildschirm lediglich der Netzwerktitel sichtbar.

Die Kommentare können später durch erneutes Aktivieren des Kommentar-Editors sichtbar gemacht werden.



Wenn Sie Ihre Kommentare mit auf Papier ausgeben wollen, aktivieren Sie die Menü-Funktion "Datei/Seite einrichten".

→ Menü:
Projekt,
Seite einrichten



Netzwerk-
kommentare
drucken



OK

Wiederholung

Selbsthaltung

Stromstoß-
Schaltung

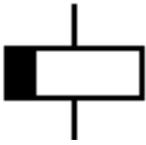
Ausschalt-
Verzögerung

Schritt-kette

Anhang

37

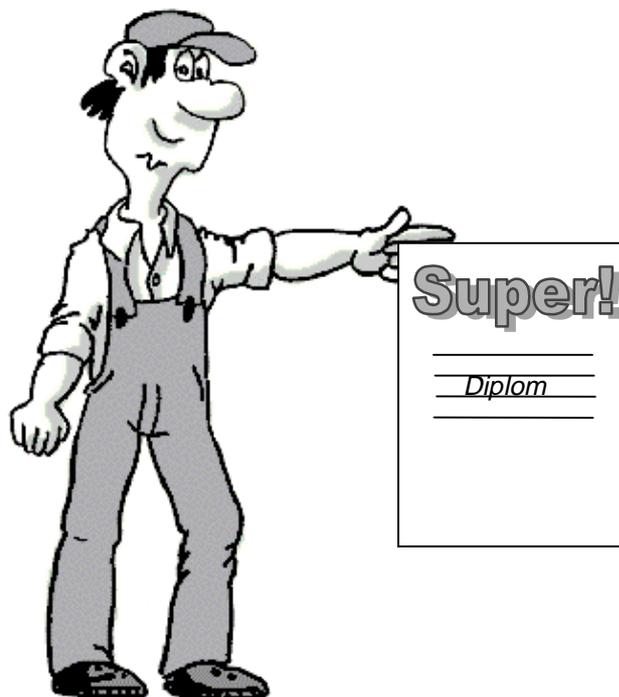
- Einführung
- Speichern unter ...
- Netzwerk einfügen
- Beschreibung der Lösung
- **Kommentare eingeben**



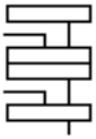
Zeigen Sie jetzt, was Sie bereits wissen

Lesen Sie sich deshalb die folgenden Fragen durch und beantworten Sie diese:

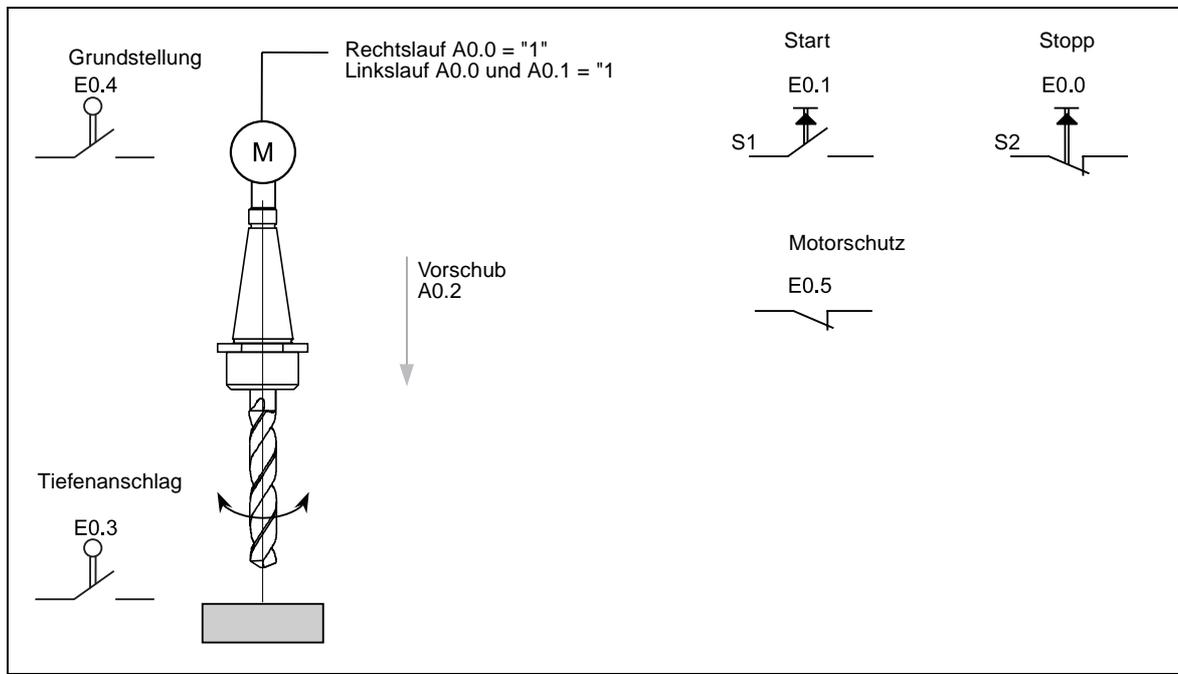
- ✓ Wie lässt sich eine Ausschaltverzögerung realisieren? Zeichnen Sie den Kontaktplan für zwei mögliche Lösungen. Einmal mit der normalen Spule —()— und einmal mit **(S)** und **(R)**. Siehe Seite 29
- ✓ Wie wird ein Projekt gespeichert? Siehe Seite 31
- ✓ Wie wird der Wert eines Zeitgliedes festgelegt? Siehe Seite 36 in der "1-Stunden-Fibel"
- ✓ Welche Kommentare sind zu Netzwerken möglich? Siehe Seite 36



- Einführung
- Grundlagen
- Arbeiten mit Schrittketten
- Modifikation
- Lösungsbeschreibung, Beispiel
- Test



Einführung



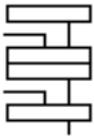
Wir realisieren nun gemeinsam eine Schrittkette.

Der Motor einer Bohrmaschine wird mit S1 im Rechtslauf gestartet. Nach 3s wird der Vorschub eingeschaltet.

Ist der Tiefenanschlag an E0.3 erreicht, wird der Vorschub ausgeschaltet. Eine Feder zieht die Maschine in die Grundstellung zurück. Dabei dreht der Antrieb im Linkslauf (A0.0 und A0.1 sind "1").

Ist die Grundstellung E0.4 = "1" erreicht, läuft der Antrieb noch 1s nach, bis die Maschine komplett abgeschaltet wird. Mit Stopp kann die Maschine immer abgeschaltet werden (Aktivierung mit E0.0 = "0").

- Einführung
- Grundlagen
- Arbeiten mit Schrittfolgen
- Modifikation
- Lösungsbeschreibung, Beispiel
- Test

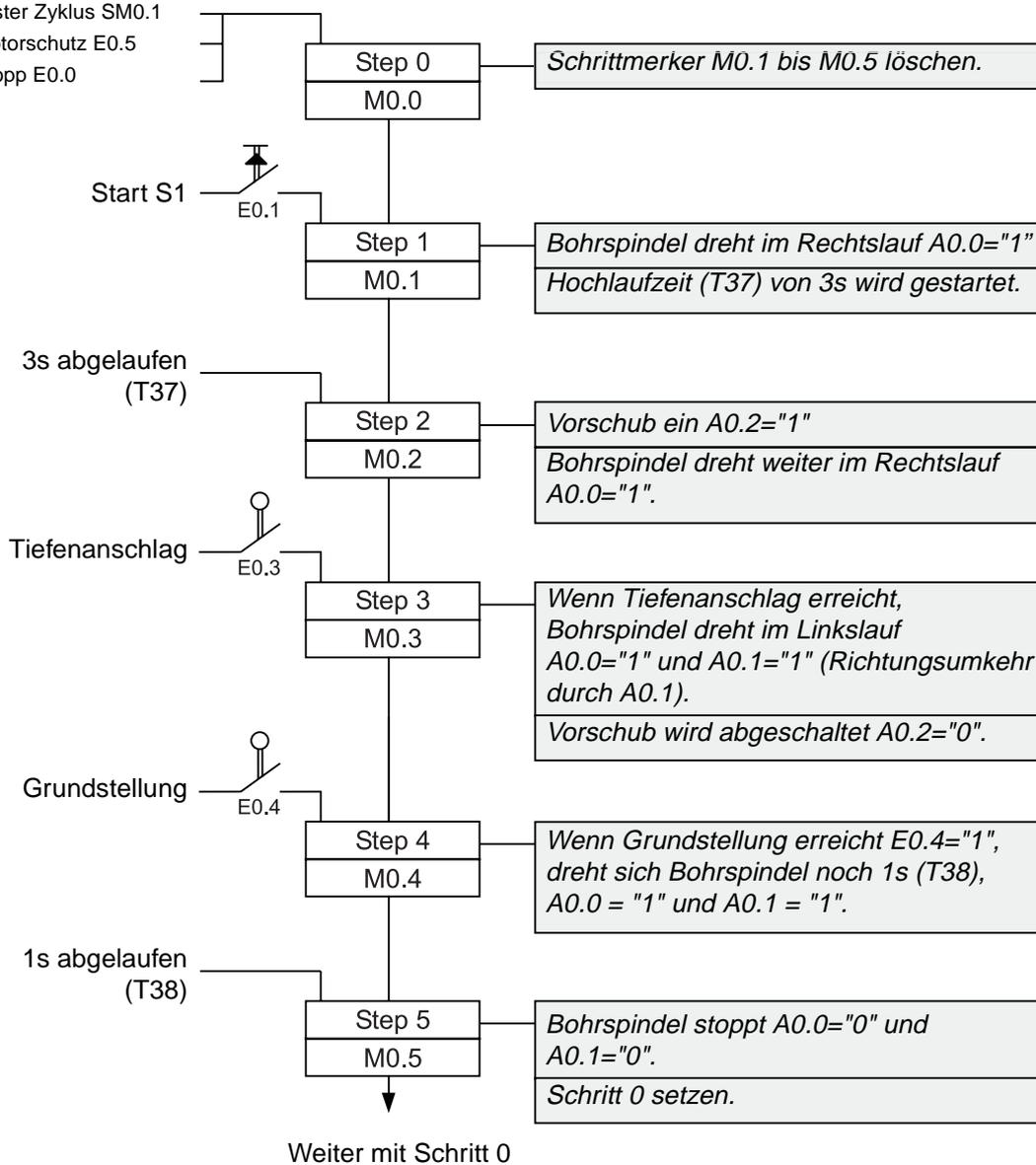


Lösungsansatz

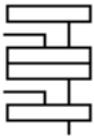


So sieht die Lösung für die Schrittfolge des Bohrmaschinen-Beispiels aus.

Erster Zyklus SM0.1
Motorschutz E0.5
Stopp E0.0



- Einführung
- Grundlagen
- Arbeiten mit Schrittfolgen
- Modifikation
- Lösungsbeschreibung, Beispiel
- Test



Grundlagen (1)

Wir lösen nun die Bohrmaschinensteuerung mit einer Schrittfolge.



Was ist eine Schrittfolgensteuerung?

- Eine Steuerungsmethode, bei der eine Aufgabe in sehr kleine, meist aufeinanderfolgende Teilaufgaben zerlegt wird (z.B. Motor ein, Vorschub ein, Vorschub aus, ...).
- Die Teilaufgaben (Funktionen) werden Schritte genannt.
- Üblicherweise muss ein Schritt abgearbeitet sein, bevor der nächste begonnen wird.
- Ein neuer Schritt wird aktiv, wenn die entsprechende Wechselschaltbedingung aktiv ist.
- Ein Schritt ist aktiv wenn der zugehörige Schrittmerker z.B. M0.1 = "1" ist.

Schritte

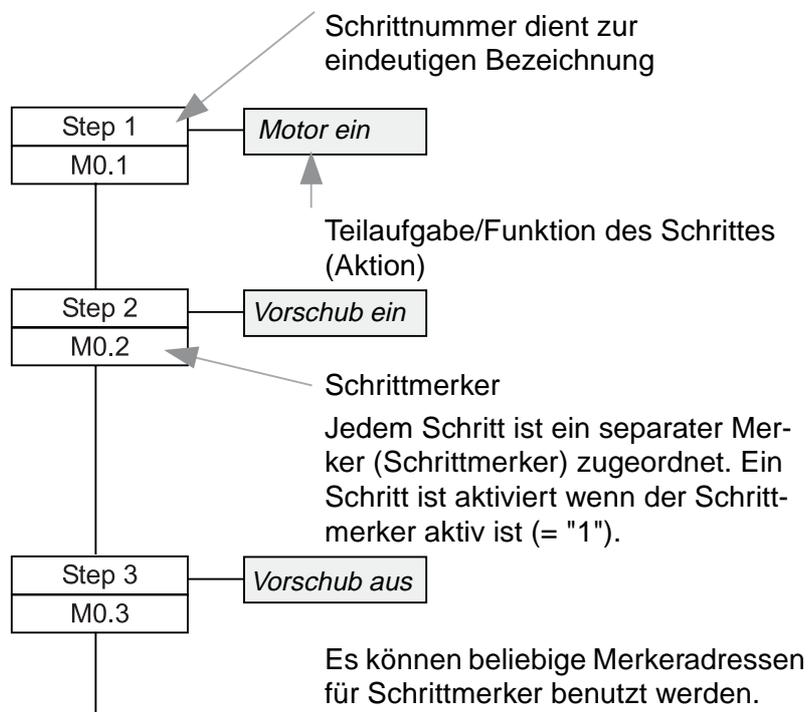
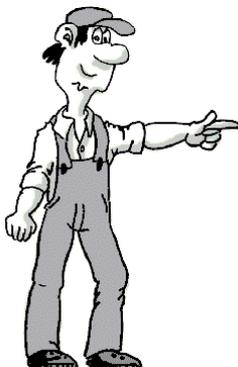
Wechselschaltbedingung

Aktiver Schritt

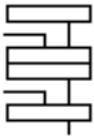


Schrittmerker
MX.Y = "1"

Für jeden wichtigen Zustand wird ein Schritt definiert.



- Einführung
- Grundlagen
- Arbeiten mit Schrittfolgen
- Modifikation
- Lösungsbeschreibung, Beispiel
- Test



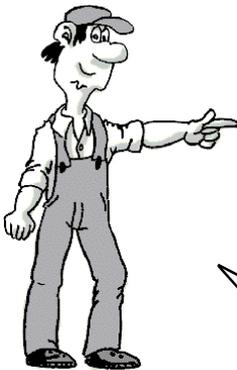
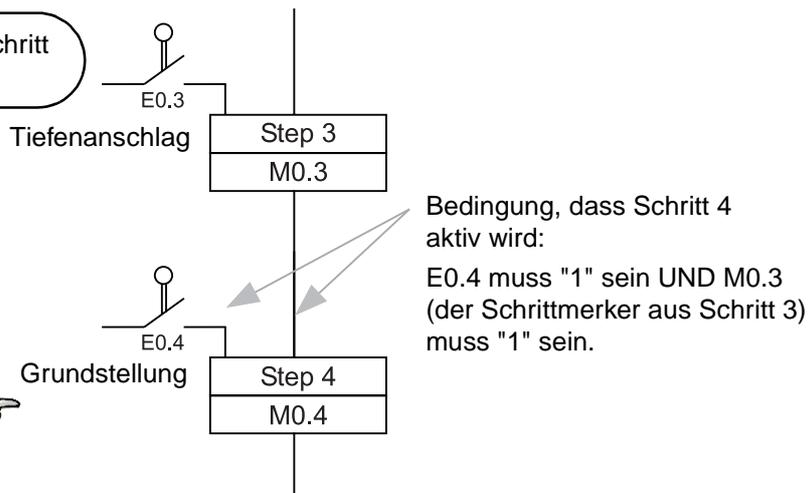
Grundlagen (2)



Was ist eine Weiterschaltbedingung?

- Jeder Schritt wird durch eine Bedingung gestartet (aktiviert). Die Bedingung wird meist aus den Zuständen der Maschine abgeleitet. Diese können betätigte Endschalter, Bedientasten, erreichte Temperaturen, Zeiten oder andere sein. Weiterschaltbedingung aktiviert Schrittmerker
- Fast immer ist auch ein aktiver vorangegangener Schritt Teil der Bedingung. Aktiver Schrittmerker "1"
- Wird ein neuer Schrittmerker gesetzt, wird der Schrittmerker des vorhergehenden Schrittes zurückgesetzt.

Immer nur einen Schritt aktivieren.

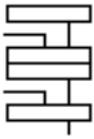


Ist diese Bedingung erfüllt, z.B. Zeit abgelaufen oder Endschalter betätigt, wird ein neuer Schritt aktiviert. Üblicherweise wird dann ein anderer aktiver Schritt zurückgesetzt.

Beim Weiterschalten der Schrittfolge kümmern wir uns zunächst noch nicht um die Aktivierung der Ausgänge. Dies wird in einem weiteren Programmteil erledigt. Das bedeutet, dass eine Steuerung mit Schrittfolgen aus zwei Programmteilen besteht:

- 1) Das eigentliche Weiterschalten der einzelnen Schritte, wenn die dafür notwendigen Bedingungen erfüllt sind (Weiterschaltbedingung).
- 2) Der Aktivierung der Ausgänge (Ventile und Antriebe ansteuern).

- Einführung
- Grundlagen
- Arbeiten mit Schritt-ketten
- Modifikation
- Lösungsbeschreibung, Beispiel
- Test



Grundlagen (3)

Die zwei Programmteile einer Schritt-ketten-steuerung:

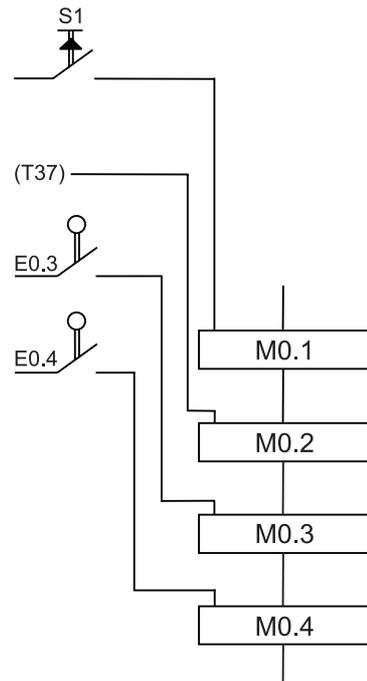
- 1) Die Bedingungen für die Aktivierung der einzelnen Schritte (Teilaufgaben) werden logisch mit den einzelnen Schritt-merkern verknüpft.

Werden die Merker M0.1... nacheinander aktiv, wird die gesamte Schritt-kette abgearbeitet.

Damit ist der gesamte Ablauf der Aufgabe festgelegt.

Start S1 E0.1, 3s Verzögerung, Tiefenanschlag E0.3, Grundstellung E0.4, jeweils vorhergehender Schritt.

Schritt-merker M0.1, M0.2, M0.3, M0.4



1. Programmteil

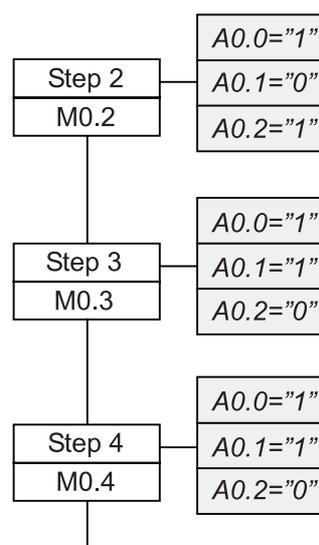
Start

Schritt-kette

- 2) Die aktiven Merker werden jeweils den Ausgängen des Automatisierungsgeräts zugewiesen, die dann ihrerseits z.B. Schütze oder Ventile ansteuern.

Dies ist die Schnittstelle zur Anlage/ Maschine.

A0.1, A0.2, A0.0

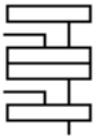


2. Programmteil

Befehlsausgabe

z.B. Motoren, Ventile

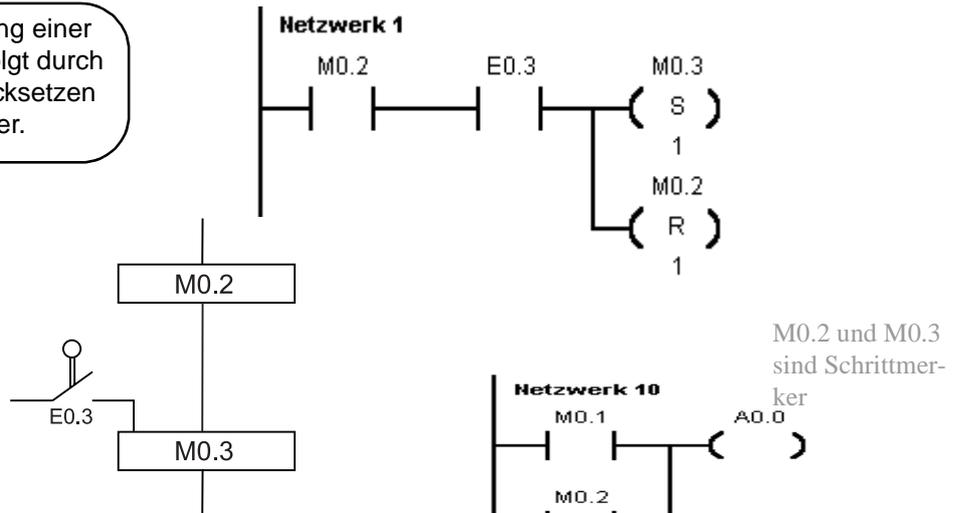
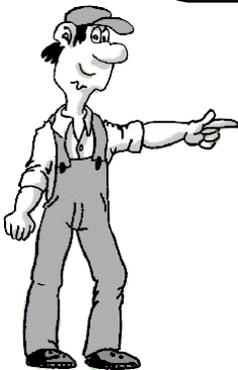
- Einführung
- Grundlagen
- Arbeiten mit Schrittketten
- Modifikation
- Lösungsbeschreibung, Beispiel
- Test



Grundlagen (4)

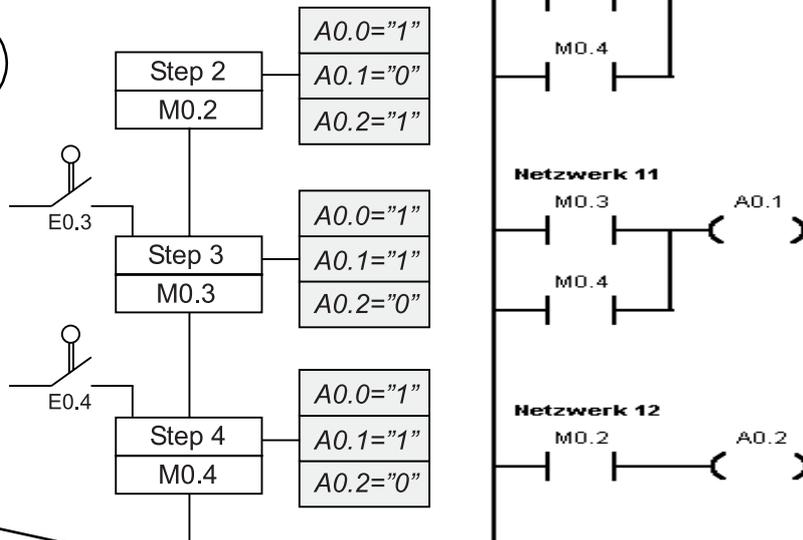
1) Steuerung/Weiterschaltung der Schrittke

Die Fortschaltung einer Schrittke erfolgt durch Setzen und Rücksetzen der Schrittmerker.



2) Ansteuerung der Ausgänge über die Schrittmerker

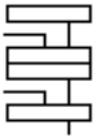
Wenn ein Ausgang "0" ist, wird er nicht angesteuert.



Ausgänge werden nur von den Schrittmerkern angesteuert. Die Zuweisung von Ausgängen mit einer normalen Spule —()— stellt sicher, dass der Ausgang nur im jeweiligen Schritt aktiviert ist.

Muss ein Ausgang in mehreren Schritten "1" sein (z.B. A0.0), werden die jeweiligen Schrittmerker mit "ODER" verknüpft und dem Ausgang zugewiesen.

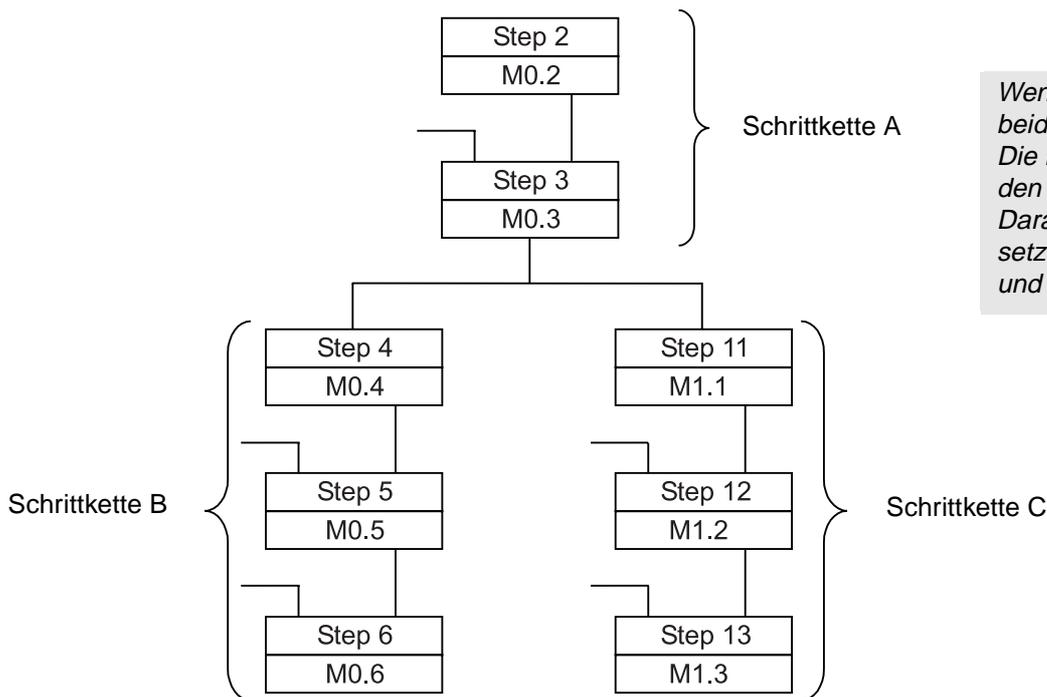
- Einführung
- Grundlagen
- **Arbeiten mit Schrittfolgen**
- Modifikation
- Lösungsbeschreibung, Beispiel
- Test



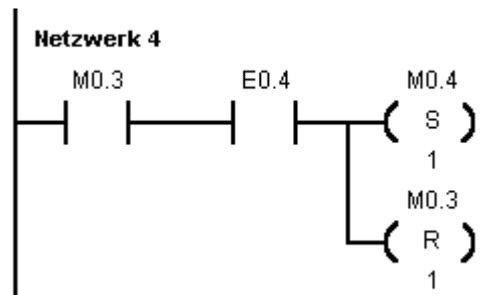
Arbeiten mit Schrittfolgen (1)



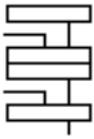
- Jedem Schritt ist ein separater Merker (Schrittmerker) zugeordnet. Dieser ist "1", wenn der Schritt aktiv ist.
- Der Übersicht wegen sollte in einer Schrittfolge gleichzeitig immer nur ein Schritt aktiv sein. D.h. nur ein Schrittmerker sollte "1" sein.
- Ist die Aufgabe komplizierter, sollte günstigerweise eine weitere Schrittfolge benutzt werden.
- Sind zwei oder mehrere Vorgänge gleichzeitig und unabhängig zu steuern, arbeitet man mit separaten Schrittfolgen. Dies wird in der folgenden Grafik gezeigt.



Wenn M0.3="1" ist, starten die beiden Schrittfolgen B und C. Die Merker M0.4 und M1.1 werden durch M0.3 gesetzt. Daraufhin wird M0.3 zurückgesetzt und die Schrittfolgen B und C laufen unabhängig weiter.



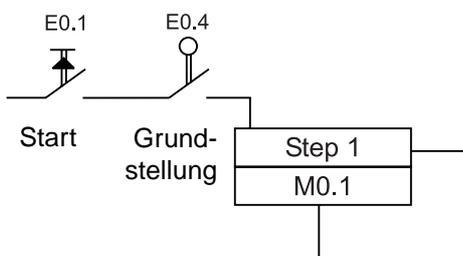
- Einführung
- Grundlagen
- **Arbeiten mit Schritt-ketten**
- Modifikation
- Lösungsbeschreibung, Beispiel
- Test



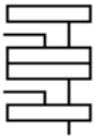
Arbeiten mit Schritt-ketten (2)

Die Weiterschaltbedingung ist in der Praxis auch aus mehreren Kontakten zusammengestellt.

Unser Beispiel kann z.B. dahingehend erweitert werden, dass der Start nur dann erfolgen kann, wenn die Bohrmaschine in Grundstellung steht. Die Schritt-kette sieht an dieser Stelle dann wie folgt aus:



- Einführung
- Grundlagen
- **Arbeiten mit Schritt看etten**
- Modifikation
- Lösungsbeschreibung, Beispiel
- Test

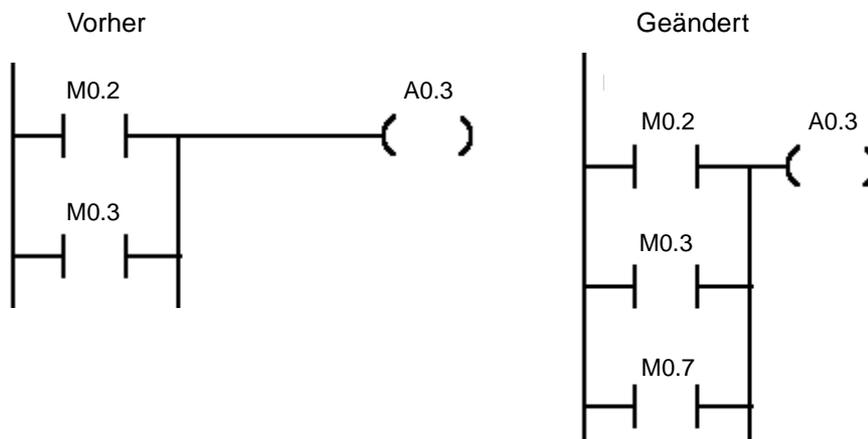


Arbeiten mit Schritt看etten (3)



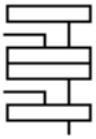
Vorteile

- **Steuerteil der Schritt看ette und Ansteuerung der Ausgänge sind getrennt**
 - Wenn ein Ausgang statt in Schritt 1, 2 und 3 nun auch in Schritt 7 aktiv sein muss, ist nur an einer Stelle das Programm zu ändern.



- Änderungen am Steuerteil der Schritt看ette beeinflussen die Ansteuerung der Ausgänge nicht.
- **Das Programm ist leicht testbar**
 - Jeder Schritt ist am Programmiergerät verfolgbar.
 - Wenn die Weiterschaltung nicht arbeitet, ist leicht erkennbar, welche Bedingung fehlt.
- **Die Stillstandszeiten der Maschine sind geringer**
 - Arbeitet eine Maschine nicht, ist aus der mechanischen Position der Maschine und dem aktiven Schrittmarker leicht erkennbar, welche Weiterschaltbedingung fehlt.
- **Weniger Programmierfehler, schnellere Inbetriebnahme**
 - Die Benutzung der Schritt看ette zwingt zu einer Programmstrukturierung. Dies minimiert Programmierfehler.

- Einführung
- Grundlagen
- **Arbeiten mit Schrittketten**
- Modifikation
- Lösungsbeschreibung, Beispiel
- Test

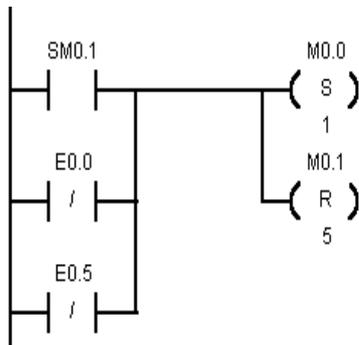


Wichtig für sicheres Arbeiten (1)



Im ersten Schrittmerker (Grundstellung) sollten z.B. keine Antriebe oder Ventile aktiviert werden. In unserem Beispiel ist dies Schritt 0 bzw. Schrittmerker M0.0.

Wenn "STOP" betätigt wird oder ein Motorschutz anspricht, muss der erste Schrittmerker (in unserem Beispiel M0.0) nur gesetzt werden und alle Antriebe stehen. Gleichzeitig müssen alle anderen Schrittmerker zurückgesetzt werden.



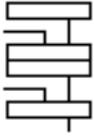
M0.0 wird gesetzt, M0.1 bis M0.5 werden zurückgesetzt

- im ersten Zyklus nach Spannungswiederkehr durch SM0.1 oder
- wenn E0.0="0" oder
- wenn E0.5="0".

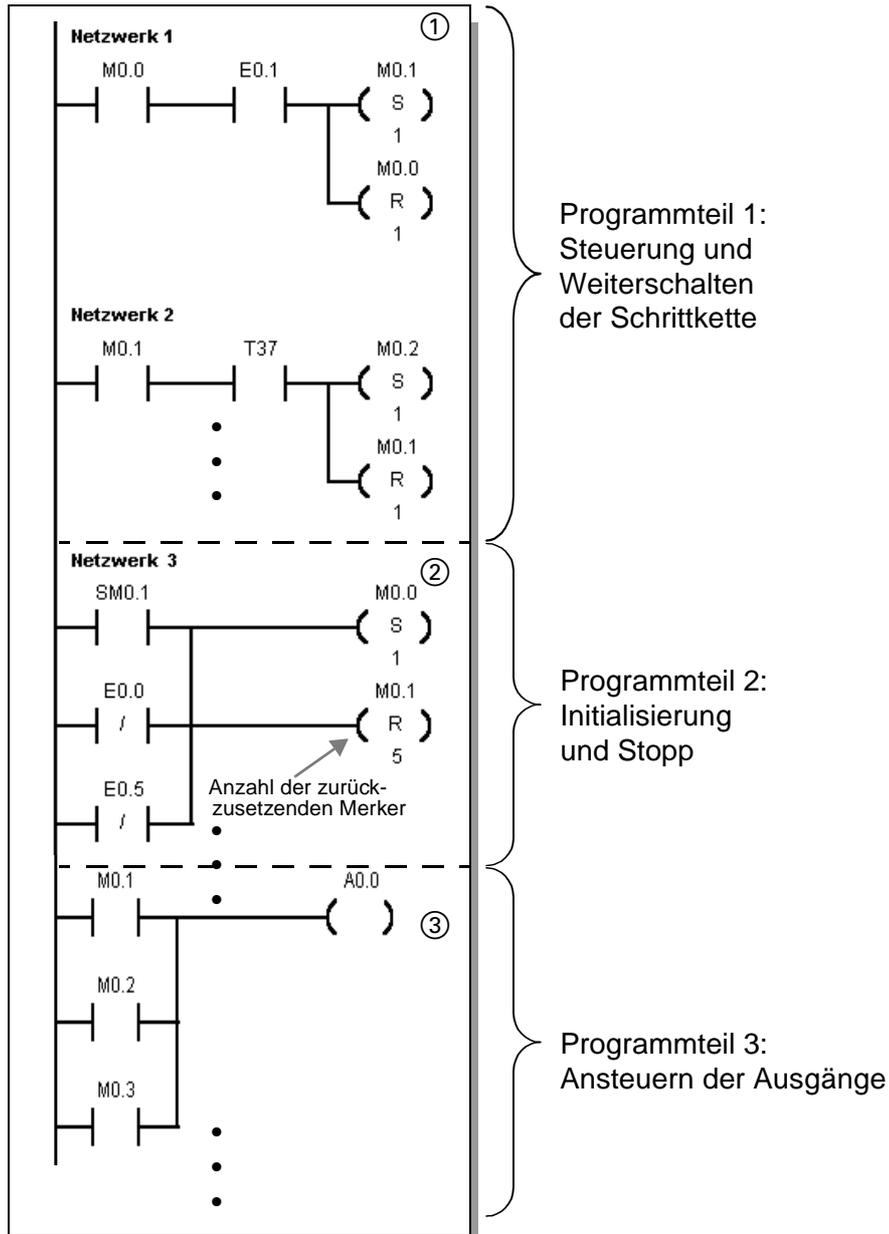
SM0.1 liefert im ersten Zyklus nach Wiedereinschalten für einen Zyklus Zustand "1"

Das im Beispiel gezeigte Programmteil muss am Ende der "normalen" Fortschaltbedingungen der Schrittke stehen. Damit ist sichergestellt, dass vor der Aktivierung der Ausgänge eine evtl. notwendige Abschaltung erfolgt.

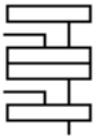
- Einführung
- Grundlagen
- **Arbeiten mit Schrittfolgen**
- Modifikation
- Lösungsbeschreibung, Beispiel



Wichtig für sicheres Arbeiten (2)



- Einführung
- Grundlagen
- Arbeiten mit Schrittketten
- **Modifikation**
- Lösungsbeschreibung, Beispiel
- Test



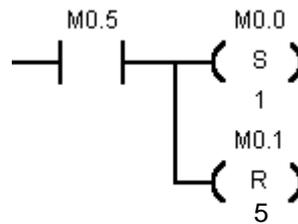
Modifikation



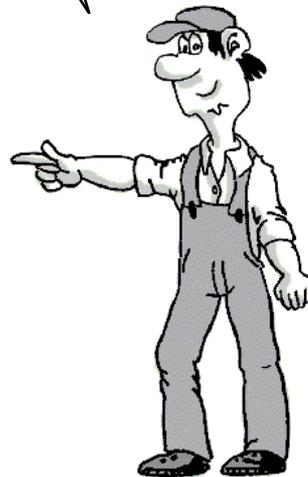
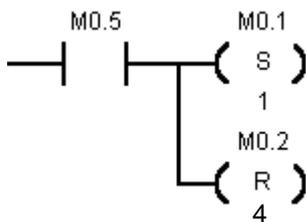
Netzwerk 6 bestimmt, in welchen Schritt das Programm nach Schritt 5 springt. Im Beispiel springt es in Schritt 0.

Dies wird gesteuert durch:

Setzen von M0.0 und Rücksetzen von M0.1 bis M0.5.

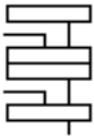


Wenn das Programm automatisch nach Schritt 5 zu Schritt 1 springen soll, muss Netzwerk 6 so aussehen.



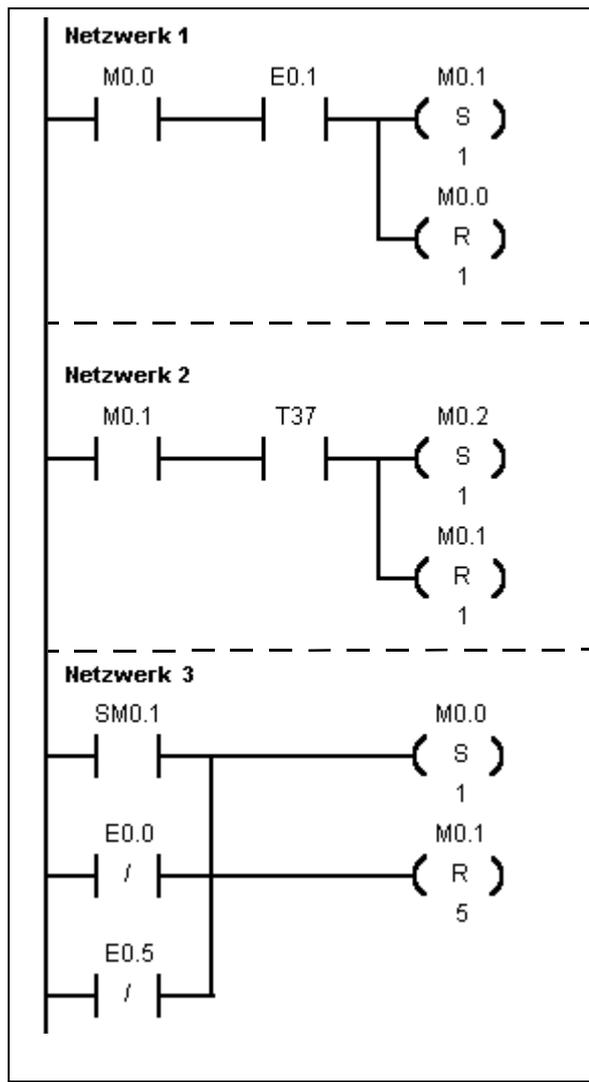
Mit dieser Modifikation läuft der "Bohrautomat" automatisch, bis er durch E0.0 oder E0.5 gestoppt wird.

- Einführung
- Grundlagen
- Arbeiten mit Schrittfolgen
- Modifikation
- Lösungsbeschreibung, Beispiel
- Test



Lösungsbeschreibung, Beispiel (1)

Programmteil 1 - Weiterschaltung der Schrittfolge



Aktivieren von Schritt 1

Der Schrittmerker M0.1 wird gesetzt, wenn die Schrittfolge in Grundstellung ist (M0.0 = "1") UND E0.1 betätigt wird. Gleichzeitig wird M0.0 der Schrittmerker der Grundstellung zurückgesetzt.

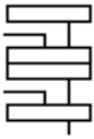
Aktivieren von Schritt 2

Der Schrittmerker M0.2 wird gesetzt, wenn die Schrittfolge in Schritt 1 steht (M0.1 = "1") UND die Zeit T37 abgelaufen ist. Gleichzeitig wird der Schrittmerker M0.1 zurückgesetzt.

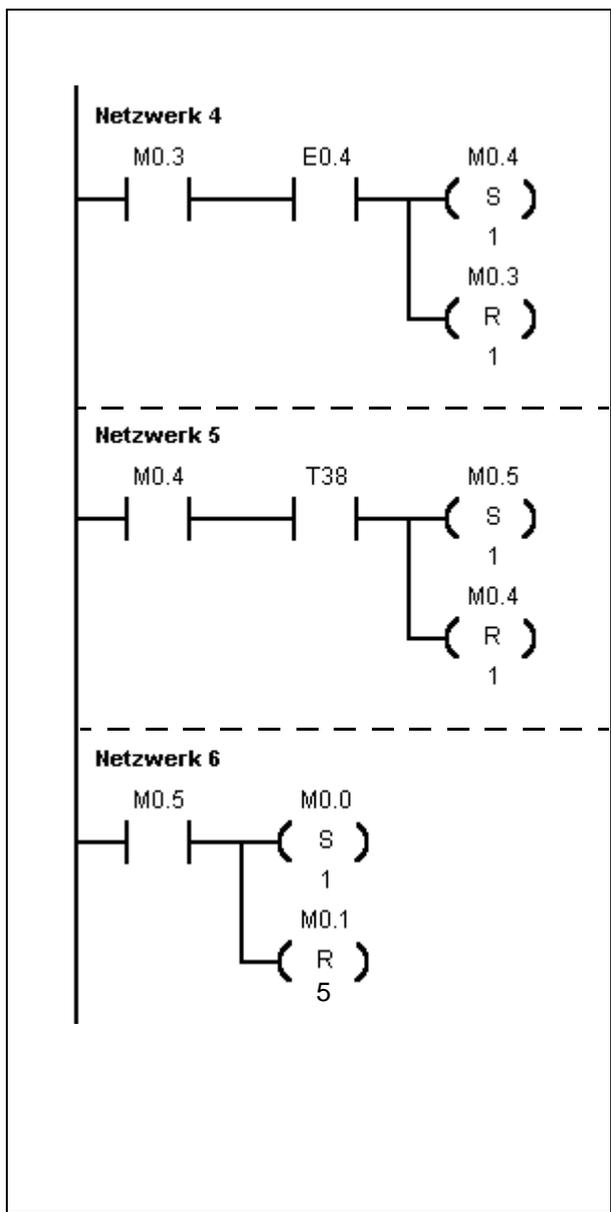
Aktivieren von Schritt 3

Der Schrittmerker M0.3 wird gesetzt, wenn die Schrittfolge in Schritt 2 steht (M0.2 = "1") UND Eingang E0.3 Tiefenanschlag "1" wird. Gleichzeitig wird M0.2 zurückgesetzt.

- Einführung
- Grundlagen
- Arbeiten mit Schrittfolgen
- Modifikation
- Lösungsbeschreibung, Beispiel
- Test



Lösungsbeschreibung, Beispiel (2)



Aktivieren von Schritt 4

Der Schrittmerker M0.4 wird gesetzt, wenn die Schrittfolge in Schritt 3 steht (M0.3 = "1") UND Eingang E0.4 (Grundstellung) "1" wird. Gleichzeitig wird M0.3 zurückgesetzt.

Aktivieren von Schritt 5

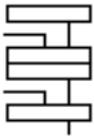
Der Schrittmerker M0.5 wird gesetzt, wenn die Schrittfolge in Schritt 4 steht (M0.4 = "1") UND die Zeit T38 abgelaufen ist. Gleichzeitig wird der Schrittmerker M0.4 zurückgesetzt.

Aktivieren von Schritt 0

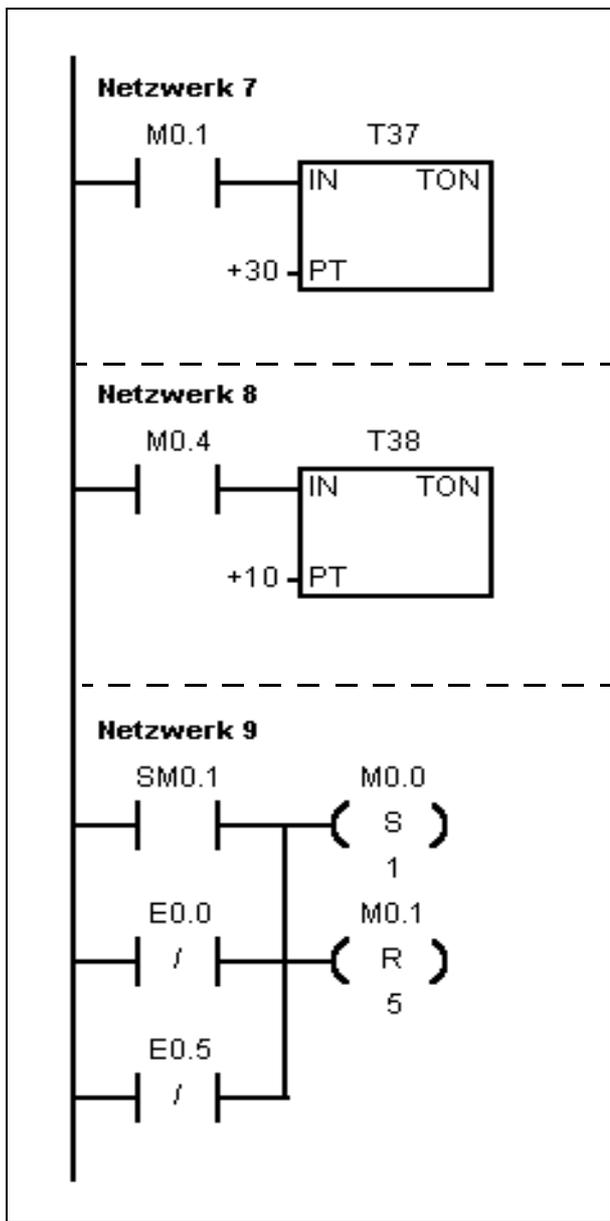
Wenn Schrittmerker M0.5 aktiv ist (Nachlaufzeit T38 ist beendet), dann wird von der Schrittfolge aus der Schritt 0 (Initialisierungsschritt) aktiviert. Dieser Schritt in Netzwerk 6 wurde bewusst eingebaut, weil an dieser Stelle auch noch weitere Bedingungen, z.B. die Entnahme des Werkstückes, abgefragt werden könnten, bevor Schritt 0 wieder aktiviert wird.

Diese Bedingung müsste in Reihe zu Kontakt M0.5 geschaltet werden.

- Einführung
- Grundlagen
- Arbeiten mit Schrittketten
- Modifikation
- Lösungsbeschreibung, Beispiel
- Test



Lösungsbeschreibung, Beispiel (3)



Aktivierung der Zeit T37

Wenn Schritt 1 aktiv ist ($M0.1 = "1"$), wird die Zeit T37 gestartet.

Aktivierung der Zeit T38

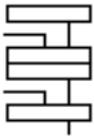
Wenn Schritt 4 aktiv ist ($M0.4 = "1"$), wird die Zeit T38 gestartet.

Initialisierung der Schrittke

Der Schrittmerker M0.0 wird gesetzt

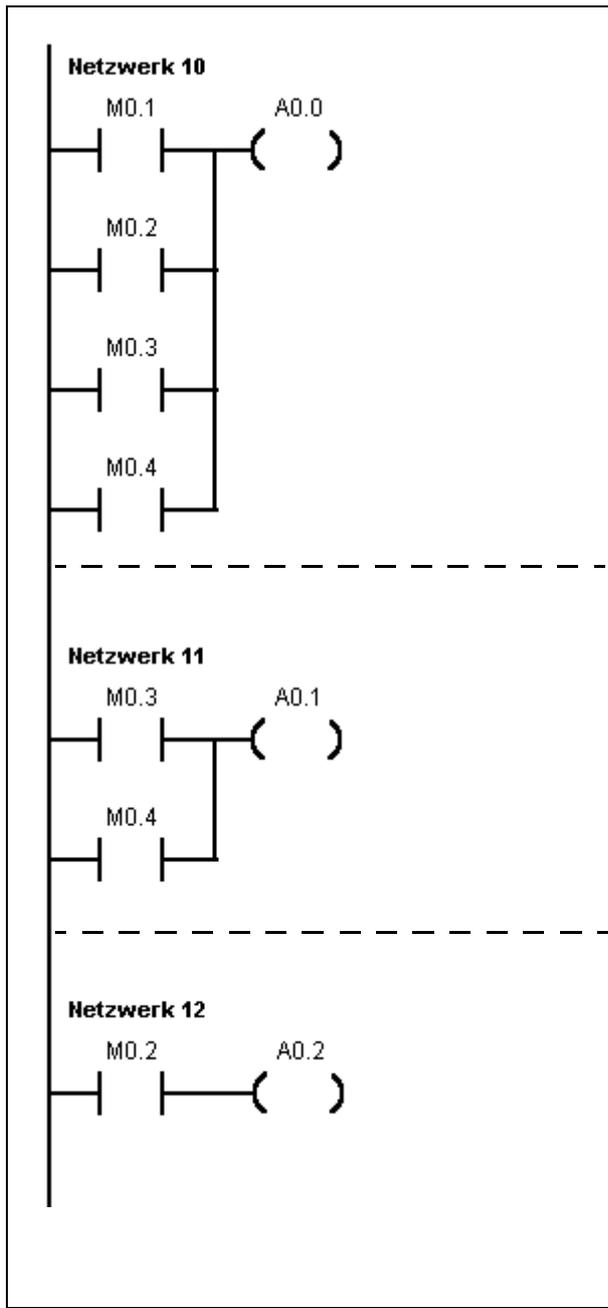
- 1) Im ersten Zyklus (SM0.1 ist hier für einen Zyklus "1")
ODER
- 2) Wenn Stopp betätigt wird ($E0.0 = "0"$)
ODER
- 3) Wenn der Motorschutz angesprochen hat ($E0.5 = "0"$). Gleichzeitig werden die Schrittmerker M0.1 bis M0.5 zurückgesetzt.

- Einführung
- Grundlagen
- Arbeiten mit Schrittfolgen
- Modifikation
- Lösungsbeschreibung, Beispiel
- Test



Lösungsbeschreibung, Beispiel (4)

Programmteil 2 - Ansteuern der Ausgänge



Ausgang A0.0 aktivieren (Antrieb Rechtslauf)

Ausgang A0.0 ist "1" in den Schritten 1, 2, 3, 4, d.h. wenn M0.1 oder M0.2 oder M0.3 oder M0.4 = "1" sind.

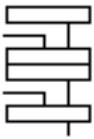
Ausgang A0.1 aktivieren (Richtungsumkehr)

Ausgang A0.1 ist "1" in den Schritten 3 und 4, d.h. wenn M0.3 oder M0.4 = "1" sind.

Ausgang A0.2 aktivieren (Vorschub ein)

Wenn Merker M0.2 = "1", dann wird Ausgang A0.2 = "1".

- Einführung
- Grundlagen
- Arbeiten mit Schritt-ketten
- Modifikation
- Lösungsbeschreibung, Beispiel
- Test



Test

Sie können das Programm selbst eingeben oder von der Diskette die Datei "d05.mwp" laden. Beachten Sie, dass der Stopp-Schalter E0.0 und der Motorschutz E0.5 "Öffner-Kontakte" sind. Dies wird aus Sicherheitsgründen so realisiert. Ein Leitungsbruch zwischen Schalter und SPS stoppt die Maschine.

E0.5 und E0.0 müssen zum Test "1" sein, d.h. die Eingangsleuchtdioden müssen leuchten.

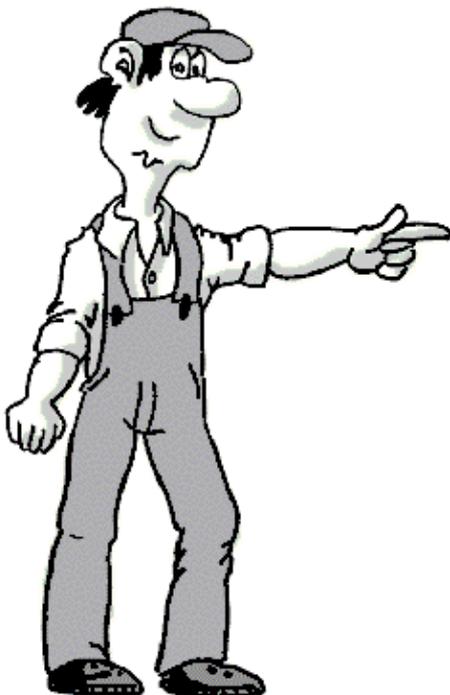
Mit kurzen Betätigen von E0.1 startet der Antrieb. Nach 3 s schaltet sich der Vorschub A0.2 ein. Nach Betätigen von E0.3 kehrt der Antrieb seine Richtung um und der Vorschub A0.2 stoppt.

Wird die Grundstellung erreicht (kurzes Betätigen von E0.4), stoppt der Antrieb nach 1s.

E0.0 und E0.5 stoppen den Antrieb in jeder Phase.

Beobachten Sie das Programm im Testbetrieb. Sie werden jeweils genau sehen, welche Eingang notwendig ist, damit die Schritt-kette weiterschaltet.

Probieren Sie es selbst aus !



Schrittke

Notizen





Geschafft.

Jetzt können Sie mit der S7-200 selbstständig Aufgaben lösen. Wenn Sie kompliziertere Schüttschaltungen umsetzen wollen, finden Sie im Anhang einige nützliche Tipps.

Wiederholung



Selbsthaltung



Stromstoß-Schaltung



Ausschalt-Verzögerung



Schritt看ette



Anhang

57

Wiederholung



Selbsthaltung



Stromstoß-Schaltung



Ausschalt-Verzögerung



Schrittkeite



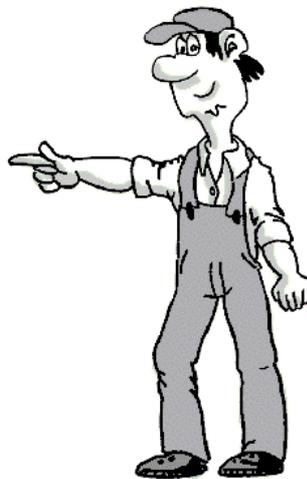
Anhang

Lust auf mehr?

Weitere Beispiele finden Sie im Verzeichnis "Samples" in Ihrem STEP 7-Micro/WIN-Verzeichnis oder in den "Tipps & Tricks" zur S7-200. Die "Tipps & Tricks" sind über Ihren SIMATIC-Partner erhältlich.

Weitere Informationen finden Sie in den Handbüchern zur S7-200. Umfassend weiterbilden können Sie sich bei einem S7-200-Kurs in Ihrem Siemens-Trainingscenter oder bei Ihrem SIMATIC-Partner.

Offene Fragen oder
technische Probleme:
Die SIMATIC-Partner
helfen Ihnen gerne.



Wenden Sie sich bitte an Ihren SIMATIC-Partner, bei dem Sie das Einsteiger-Paket bezogen haben. Er wird Ihnen gerne weiterhelfen.

Sollte Ihr Partner nicht erreichbar sein, rufen Sie bitte unsere SIMATIC-Hotline, Tel.: 0911/895-7000, an.

Wiederholung



Selbsthaltung



Stromstoß-Schaltung



Ausschalt-Verzögerung



Schrittkeite



Anhang

Im folgenden haben wir Ihnen einige Beispiele zusammengestellt, die es Ihnen leicht machen, auch kompliziertere "Schalterverknüpfungen" im Kontaktplan zu realisieren.



Wiederholung



Selbsthaltung



Stromstoß-Schaltung



Ausschalt-Verzögerung



Schritt-kette



Anhang

61

Wiederholung



Selbsthaltung



Stromstoß-Schaltung



Ausschalt-Verzögerung



Schrittkeite



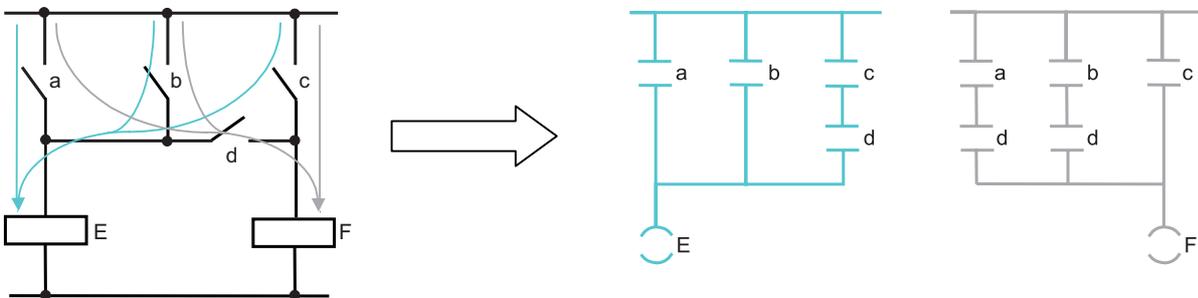
Anhang



Brückenschaltung

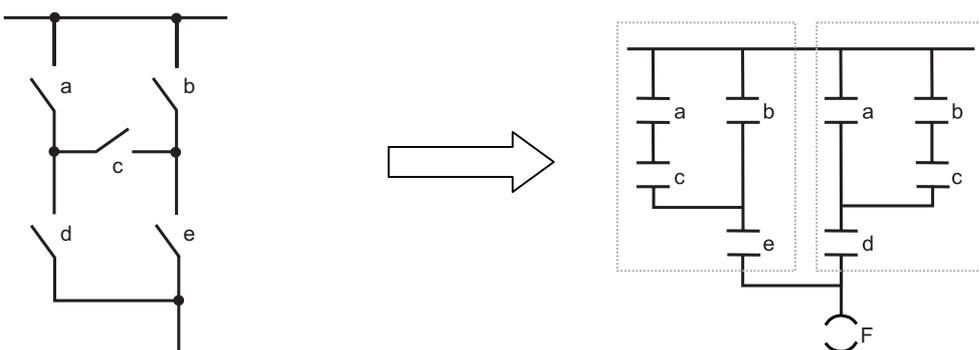
Wenn Sie Umsteiger von der Schütz- zur SPS-Technik sind, begegnen Ihnen höchstwahrscheinlich Schalterverknüpfungen, die im Kontaktplan nicht direkt umsetzbar sind. Dazu zählt die Brückenschaltung. Für die einfache und für die komplexe Brückenschaltung wird hier jeweils kurz ein Lösungsweg skizziert.

1) Einfache Brückenschaltung



Die einfache Brückenschaltung (links) wird mit zwei Netzwerken umgesetzt. Es werden einfach die einzelnen möglichen Strompfade aufgegliedert. Zur einfacheren Vergleichbarkeit wurde der Kontaktplan ebenfalls senkrecht abgebildet.

2) Komplexe Brückenschaltung



Die zwei möglichen Strompfade wurden wieder umgewandelt und neu kombiniert. Zum einen a, c parallel b, zum anderen b, c parallel a. Zur einfacheren Vergleichbarkeit wurde der Kontaktplan senkrecht abgebildet.

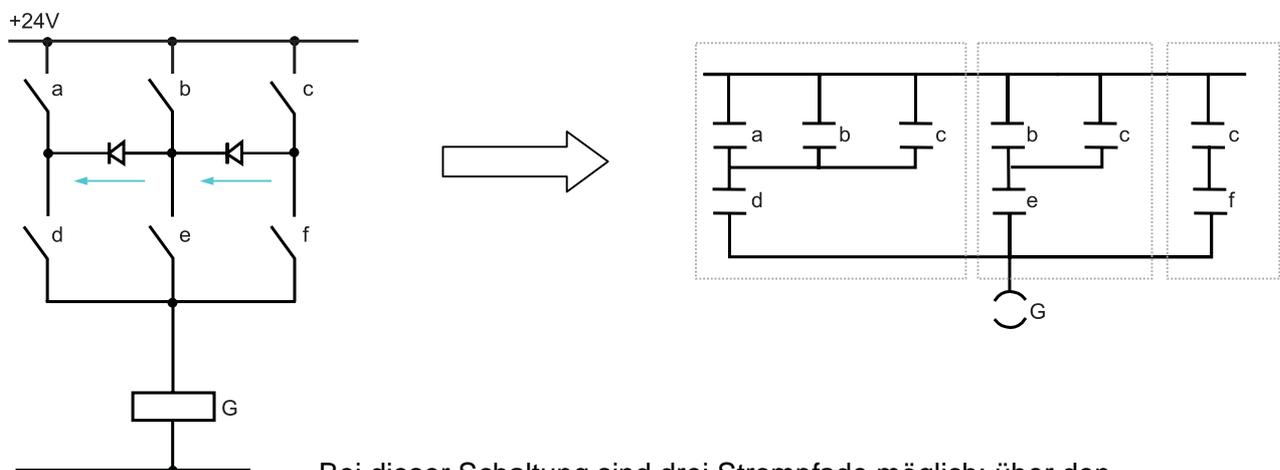
Vermeiden Sie bei neuen Projekten die Brückenschaltung im Stromlaufplan! Denken Sie möglichst gleich "im Kontaktplan".



Diodenschaltung

Wenn in "alten" Schaltplänen Dioden verwendet wurden, ist deren Umsetzung in den Kontaktplan nicht ohne weiteres möglich.

Da Dioden im Prinzip Verbindungsleitungen darstellen, jedoch nur in einer Richtung Strom leiten, wird hier ein ähnlicher Lösungsweg beschrieben, wie bei der Brückenschaltung. Der Kontaktplan ist zur besseren Vergleichbarkeit mit dem Stromlaufplan wieder senkrecht abgebildet.



Bei dieser Schaltung sind drei Strompfade möglich: über den Schalter d, den Schalter e sowie über den Schalter f.

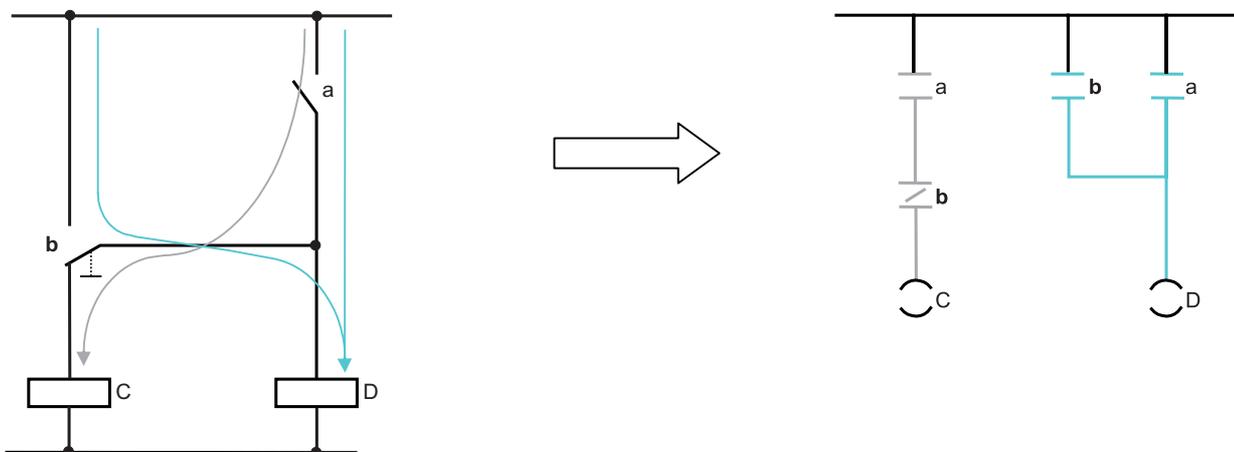
Der Strom durch die Dioden kann nur von b in Richtung d bzw. von c in Richtung e fließen.

Durch die drei Strompfade ergeben sich auch die drei in der Kontaktplan-Lösung eingerahmten Teilnetzwerke. Da Schalter d, e und f auf der selben Schiene wie der Ausgang G liegen, sind auch diese drei Teilnetzwerke zu einem Netzwerk verbunden.



Umschalter

Umschalter sollten Ihnen auch kein Kopfzerbrechen bereiten, wenn Sie einen Stromlaufplan in den Kontaktplan umsetzen. Nachfolgend wird diese Transformation kurz erläutert.



Zur Hervorhebung des Stromfades wurde dieser grafisch hervorgehoben.

Der Umschalter b wird danach aufgeteilt in einen Öffner-Kontakt, der in Serie zu a gilt und am Ausgang C zur Wirkung beiträgt, bzw. einen Schließer-Kontakt, der parallel zu a Gültigkeit erlangt und D schaltet.

Somit kann ein Umschalter grundsätzlich durch einen Öffner-Kontakt und einen Schließer-Kontakt derselben Eingangs-Adresse im Kontaktplan umgesetzt werden.

Anhang

Tipps
Notizen.



Notizen

A4

Wiederholung

Selbsthaltung

Stromstoß-Schaltung

Ausschalt-Verzögerung

Schrittkeite

Anhang



Index A...H

In diesem Index wurden die wichtigsten Stichworte zur Programmierung der S7-200 aufgenommen. Sie finden kurze Erklärungen zu den verwendeten Abkürzungen der Fibel sowie teilweise Querverweise zur 1-Stunden-Fibel.

Folgendes Symbol wird im Index verwendet:

1h- Hinweise auf Seiten der 1-Stunden-Fibel

- A**
- A: Ausgang, z.B. A0.0
 - AB: Ausgangsbyte (8 Bit), z.B. AB0
 - Ablaufsteuerung: Steuerung, die aus Ereignissen Schritte ableitet bzw. weiterschaltet. Diese aktivieren wiederum vorgegebene Aktionen.
 - Arbeiten mit Schrittketten: 45 ff.
 - Ausschaltverzögerung: 29 ff.
 - Ausschneiden von Elementen: 1h- 32
 - AW: Ausgangswort (16 Bit), z.B. AW0
 - AWL: Anweisungsliste (engl.: STL)
- B**
- Betriebsartenschalter: Schalter an der S7-200 mit drei Stellungen: STOP, TERM, RUN.
 - Binär: Darstellung von Zahlen in Bits (zwei mögliche Werte, 0 oder 1)
 - Bit: Binäre Ziffer: 6
 - Brückenschaltung: A1
 - BuB: Bedienen und Beobachten, z.B. mit Text Displays, Operator Panels und Touch Panels; heute HMI
 - Byte: 8 Bit breiter Wert: 1h- 48
- C**
- CPU: Central Processing Unit, Zentraleinheit, z.B. die S7-200
- D**
- Datenbaustein: Variablenspeicher der S7-200, hier können Werte für die Nutzung im Steuerprogramm abgelegt werden
 - DB1: Datenbaustein der S7-200
 - Diodenschaltung: A2
 - DIV: Division berechnen
- E**
- E: Eingang, z.B. E0.0
 - EB: Eingangsbyte (8 Bit), z.B. EB0
 - Einfügen von Elementen: 1h- 30
 - Einschaltverzögerung: 1h- 35
 - EW: Eingangswort (16 Bit), z.B. EW0
- F**
- Flanken: 21,22
- G**
- Grundlagen der Schrittkette: 39-42
- H**
- Halbautomat: Steuerung, die gewisse Abläufe selbständig durchläuft, an anderen Stellen jedoch auf Eingaben vom Nutzer angewiesen ist.
 - HMI: Human-Machine-Interface (Bedienen und Beobachten)



Index K...S

K

Kommentare eingeben: 36 ff.
 Kontaktplan: 1h-📖 25
 KOP-Status: 7, 1h-📖 26

L

Lehrmodell: 1h-📖 7
 Lösung der Ausschaltverzögerung: 29 ff.
 Lösung der Schrittkette: 39 ff.
 Lösung der Selbsthaltung: 15 ff.
 Lösung der Stromstoß-Schaltung: 21 ff.

M

MB: Merker-Byte (8 Bit)
 MD: Merker-Doppelwort (32 Bit)
 Merker: 25 ff.
 MW: Merker-Wort (16 Bit)

N

Netzwerk einfügen: 32

O

OB1: Organisationsbaustein der S7-200
 ODER-Verknüpfung: 53
 Öffner-Kontakt: 14, 15
 On-line-Hilfe: 8

P

PAA: Prozessabbild der Ausgänge: 10
 PAE: Prozessabbild der Eingänge: 9
 PLC: Programmable Logic Controller,
 englische Abkürzung für SPS.
 Prozessabbild: Programm einer SPS arbeitet
 auf Abbild der Peripherie. Zu Beginn des
 Zyklus wird Abbild der Eingänge einge-
 lesen, am Ende wird Abbild der Ausgänge
 an Peripherie ausgegeben: 9 ff.

R

Remanenz: 23
 RET: Return, Unterprogramm beenden
 Rücksetzen, Setzen: 16 ff.
 RUN: Stellung des Betriebsartenschalters der
 S7-200 zum manuellen (Wieder-)Anlauf der
 Steuerung

S

SBR: Sub-Routine, Unterprogramm
 Schließerkontakt: 8
 Schrittkette: In sich meist geschlossene Kette
 von Schritten, die in Ablaufsteuerung
 schrittweise abgearbeitet wird: 39 ff.
 Schrittmerker: 41
 Selbsthaltung: 13 ff.
 Setzen, Rücksetzen: 17 ff.
 Sicherheitsaspekte: 19
 SMB: Sondermerker-Byte (8 Bit), z.B. SMB28
 SMB28: Potentiometer der S7-200
 SMW: Sondermerker-Wort (16 Bit)
 SMD: Sondermerker-Doppelwort (32 Bit)
 SPS: Speicherprogrammierbare Steuerung:
 1h-📖 5
 Speichern des Programms: 1h-📖 41
 Spule: Darstellung für ein Ausgangselement
 im Kontaktplan (vergleichbar mit einem
 Schütz): 17
 Status: Erlaubt das Beobachten eines
 Prozesses auf Programmebene oder in
 gesonderter Statustabelle. Nützlich für Test
 und Diagnose.
 Status im Kontaktplan: 1h-📖 26
 STOP: Stellung des Betriebsartenschalters
 der S7-200 zum manuellen Anhalten der
 Steuerung.
 Stromfluss im Kontaktplan: 7
 Stromstoß-Schaltung: 21 ff.
 Symbole der Fibel: 3



Index T...Z

T

T37 (Timer): 29 ff.
 TERM: Stellung des Betriebsartenschalters der S7-200. Erlaubt Beeinflussung der Steuerung aus STEP 7-Micro/WIN heraus
 Timer: engl. Begriff für Zeitgeber.
 TON: Zeitschalter der S7-200, auch Timer genannt: 1h- 36 f.
 TONR: speichernde Einschaltverzögerung

U

UND-Verknüpfung: 6
 Unwahr, Wahr: 6

V

V: Variablen-Bit, z.B. V0.0
 VB: Variablen-Byte, z.B. VB0
 VD: Variablen-Doppelwort, z.B. VD45
 Verknüpfungssteuerung: Steuerung, die anhand Überprüfung der Eingangs-/Ausgangszustände Aktionen schaltet.
 V-Speicher: Datenbaustein der S7-200
 VW: Variablen-Wort, z.B. VW45

W

Wahr, Unwahr: 6
 Weiterschaltbedingung: 40
 Wort: Ein Wert, der durch 2 Bytes (16 Bit) repräsentiert wird.

X

XOR: Exklusiv-ODER, logische Verknüpfung, schaltet nur bei verschiedenen Zuständen (Antivalenz) am Eingang

Z

Z0: Einfacher Zähler (CTU)
 Zeit: Datentyp für alle Zeitfunktionen (hier auch "Timer" genannt).
 Zeitbit: 7
 Zyklus einer SPS (3...10ms): 9 ff.

An
Siemens AG
A&D AS MVM
Gleiwitzer Str. 555

Fax: (0911) 895-2786

90475 Nürnberg

Resonanz "2-Stunden-Fibel"

Sehr geehrte Anwenderin, sehr geehrter Anwender der Micro-SPS S7-200,

wir haben die 2-Stunden-Fibel geschaffen, um Ihnen - aufbauend auf die 1-Stunden-Fibel - einen einfachen Einstieg in die Programmierung der Micro-SPS SIMATIC S7-200 zu ermöglichen.

Wir sind sicher, dass Sie mit der Fibel typische Steuerungsaufgaben einfach lösen können. Wenn Sie weitere persönliche Anregungen haben, ist uns Ihre Meinung wichtig.

Bitte informieren Sie uns mit diesem Schreiben. Nennen Sie uns bitte auch Ihren Namen und Ihren Absender, damit wir uns bei Rückfragen direkt an Sie wenden können.

Vielen Dank

A&D AS MVM

Anregungen, Verbesserungshinweise, Rückmeldungen

Von

Name _____ Funktion _____

Firma _____ Telefon _____

Straße _____ Ort _____

Meine Anregungen:

Anhang



Tipps
Notizen.

