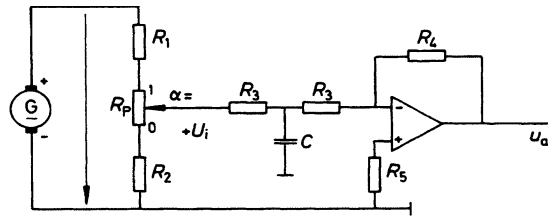


**Abb. 4.1** Zum Messbereich einer Messwerterfassung

**Abb. 4.2** Drehzahlerfassung mit Tachodynamo



Die dynamischen Eigenschaften einer Messeinrichtung für regeltechnische Aufgaben sind durch die Umsetzzeit und Eigenzeitkonstanten gegeben. Eine Regelung ist nur möglich, wenn diese Zeitkonstanten nicht die Größenordnung der wesentlichen Regelstreckenzeitkonstanten aufweisen.

### 4.1.1 Drehzahlmessung

Die Drehzahl ist ein wichtiger Messwert in der Antriebstechnik. Ihre analoge Abbildung ist mit einem Tachodynamo möglich und in Abb. 4.2 dargestellt.

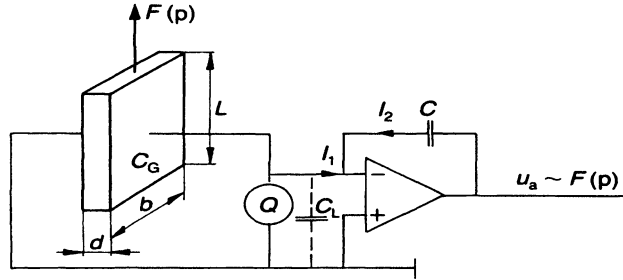
Bei Drehzahlen von 1000/min liefert ein Tachodynamo je nach Type 100 V. Daher erfolgt über einen Spannungsteiler die Normierung der Tachospannung auf 10 V, sodass sich ergibt:

$$U_i = U_T \frac{\alpha \cdot R_p + R_2}{R_1 + R_2 + R_p} \quad (4.1)$$

Die Drehzahl ist bei dieser Messwerterfassung in einem Bereich von ca. 10/min bis 1500/min einstellbar. Voraussetzung sind außerdem Widerstände  $R_1$  und  $R_2$  mit sehr niedrigen Toleranzen, damit das Teilverhältnis nicht verfälscht wird.



**Abb. 4.4** Kraftmessung mit dem piezoelektrischen Effekt



### 4.1.3 Druck- und Zugkraftmessung

Bei zahlreichen mess- und regeltechnischen Aufgaben ist das Beherrschen von Druck- und Zugkraftbeanspruchung wichtig. Zum Messen solcher nichtlinearer Größen wird oft der piezoelektrische Effekt herangezogen (Abb. 4.4).

Deformiert man einen Piezokristall (Quarkristall) mit dem Druck  $p$  oder einer Kraft  $F$ , entsteht eine Gitterverschiebung, die sich an den gegenüberliegenden Stirnflächen als Ladung unterschiedlicher Polarität bemerkbar macht. Diese Ladung hängt von der Materialdicke  $d$ , der Länge  $l$  und der Zugkraft  $F$  sowie einer Materialkonstanten  $k$  ab. Sie kann als Strom  $I$  entnommen werden.

$$Q_p = \frac{k \cdot l}{d} \cdot F = I_1 \cdot t$$

Unter Berücksichtigung der Leitungskapazität  $C_L$  und der Quarzkapazität  $C_p$  folgt:

$$Q_p = \frac{k \cdot C_L}{C_L + C_p} \cdot F \quad \text{bzw.} \quad U_a = -\frac{k}{C_L + C_p} \cdot F = -\frac{Q_p}{C_L} \quad (4.3)$$

Piezoelektrische Messwertaufnehmer sind in der Anordnung nach Abb. 4.4 besonders für dynamische Messungen geeignet. Es lassen sich große Kraftmessbereiche realisieren. Da piezoelektrische Sensoren außerordentlich hochohmig sind und nur sehr kleine Ladungsmengen erzeugen, benötigt man zur Signalverarbeitung Verstärker mit FET-Eingangsstufen.

Statische Kräfte werden bevorzugt mit Dehnungsmessstreifen (DMS) erfasst. Ein Leiter der Länge  $l$  und des Querschnitts  $A$  an den Stirnflächen hat den Widerstand:

$$R = \frac{l \cdot \rho}{A} = \frac{4 \cdot l}{d^2 \pi} \cdot \rho$$

Wird der Leiter durch eine Kraft  $F$  um eine Längungsänderung  $\Delta l$  gedehnt, ergibt sich auch eine Querkontraktion  $\Delta d$ . Auch der spezifische Widerstand  $\rho$  ändert sich bei der Kontraktion. Dieser Effekt ist bei piezoresistiven Halbleitermaterialien besonders ausgeprägt. Es ergibt sich insgesamt für die Widerstandsänderung [17]:

$$\frac{\Delta R}{R} = \frac{\Delta \rho}{\rho} + \frac{\Delta l}{l} - \frac{2 \cdot \Delta d}{d}$$

## 7.1 Speicher

Für die Speicherung von Informationen steht je nach Anwendungsgebiet eine Reihe von CMOS-Schaltkreisen zur Verfügung.

Die einfachste Art, mit einem Standardschaltkreis eine Information zu speichern, stellt Abb. 7.1 dar. Sobald ein 1-Signal am Eingang  $E$  des ODER-Gatters anliegt, zeigt der Ausgang  $A$  ebenfalls 1-Signal. Danach bleibt unabhängig vom Zustand des Eingangssignals der Ausgang  $A$  permanent auf 1-Signal stehen. Dieser Speicher ist nur durch Abschalten der Speisspannung löschar.

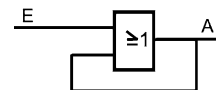
### 7.1.1 RS-Flip-Flop

Normalerweise ist es notwendig, einen Speicher im Betrieb über ein logisches Signal zu löschen und setzen zu können. Diese Funktion erhält man beispielsweise beim RS-Flip-Flop. Die Schaltung mit NOR-Gattern ist in Abb. 7.2 dargestellt. Sie wird mit zwei Rückkopplungsschleifen realisiert, die wie in Abb. 7.1 angedeutet, das Speichern ermöglichen.

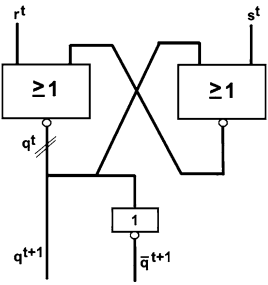
Zum Verständnis der Schaltung verwendet man die Wahrheitstabelle des Flip-Flops (Tab. 7.1). Dabei werden sowohl die Signale Setzen ( $s$ ) und Rücksetzen ( $r$ ) als auch der Ausgang ( $q$ ) als Variable zur Zeit  $t$  betrachtet.

Es ist zu sehen, dass ein Setzsignal  $s^t = 1$  den Ausgang im nächsten Augenblick  $t + 1$  (im nächsten Schritt) auf  $q^{t+1} = 1$  schaltet, wenn gleichzeitig  $r^t = 0$  ist. Ebenso bewirkt ein Reset-signal  $r^t = 1$  das Rücksetzen des Ausgangs auf  $q^{t+1} = 0$ , und zwar unabhängig vom Zustand

**Abb. 7.1** Permanenter Speicher mit ODER-Gatter



**Abb. 7.2** RS-Flip-Flop mit NOR-Gattern 4001 (reset dominant)



**Tab. 7.1** Wahrheitstabelle des RS-Flip-Flops mit NOR-Gattern

$r^t$	$s^t$	$q^t$	$q^{t+1}$
0	0	0	0
0	1	0	1
1	0	0	0
1	1	0	0
0	0	1	1
0	1	1	1
1	0	1	0
1	1	1	0

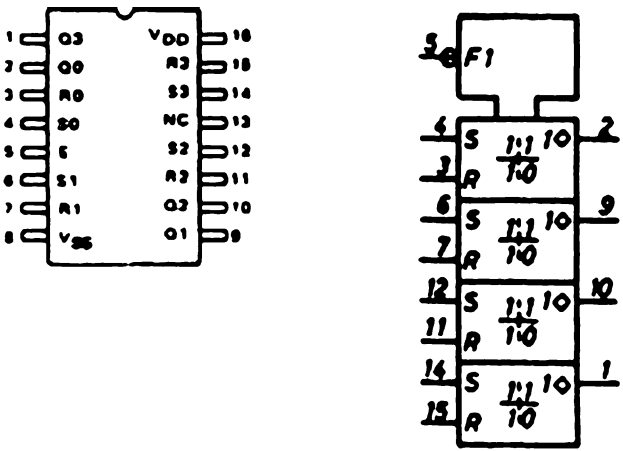
$q^{t+1}$	$r^t$	$s^t$
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

des Setzeingangs. Es überwiegt somit der Restbefehl, das Flip-Flop ist „reset dominant“. Dieser Zustand wird im Gegensatz zu den meisten Literaturangaben bei Ablaufsteuerungen bewusst eingesetzt. Als 4-fach RS-Flip-Flop (set dominant) ist der Schaltkreis verfügbar (Abb. 7.3).

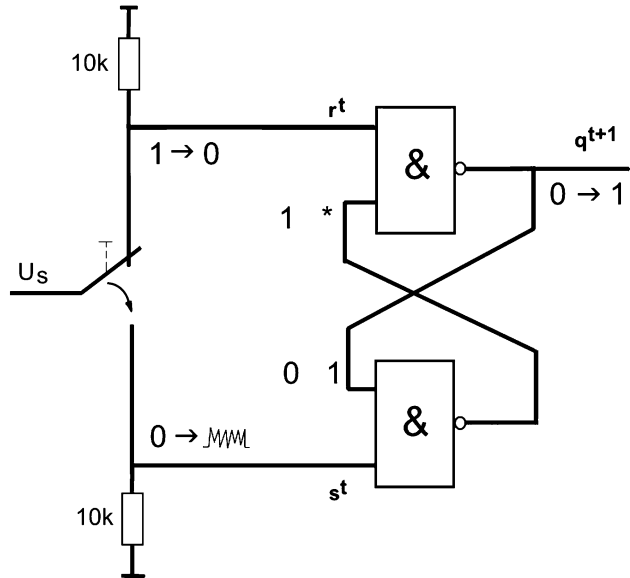
Überträgt man die Wahrheitstabelle in ein Veitch-Diagramm, lässt sich eine Funktionsgleichung des RS-Flip-Flops angeben. Sie lautet:

$$q^{t+1} = \bar{r}^t s^t + \bar{r}^t q^t . \tag{7.1}$$

**Abb. 7.3** 4-fach RS-Flip-Flop 4043



**Abb. 7.4** RS-Flip-Flop mit NAND-Gattern 4011



Bei Aufgabenstellungen wie z. B. einer Ablaufsteuerung kann man die Funktionsgleichung des RS-Flip-Flops zur Schaltungsdimensionierung mithilfe der booleschen Algebra verwenden.

Ein RS-Flip-Flop mit NAND-Gattern soll anhand eines Beispiels erläutert werden. Durch mechanisches Schwingen an Kontaktflächen beim Schalten zwischen 1- und 0-Signal entstehen unerwünschte Impulse, die zur Fehlfunktion von Schaltungen führen können (Kontaktprellen). Das Problem wird mit der folgenden Schaltung gelöst (Abb. 7.4).

Wenn der Schalter betätigt wird und Kontaktprellen entsteht, hat dies keinen Einfluss auf das Ausgangssignal  $q^{t+1}$ . Es wechselt nur einmal von 0-Signal nach 1-Signal.

Die beiden Widerstände von je 10 k $\Omega$  gewährleisten, dass die Eingänge der NAND-Gatter stets auf festem Potenzial liegen (siehe Handhabungsregeln).

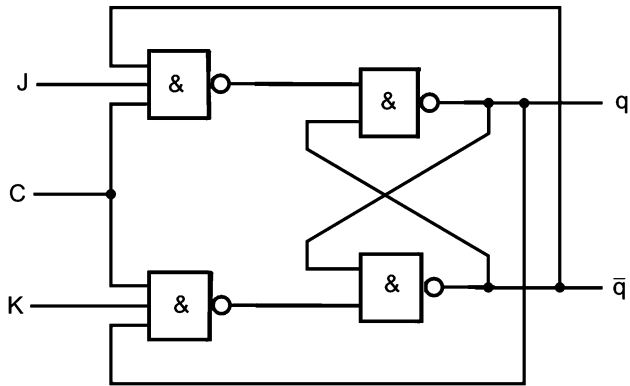
## 7.1.2 JK-Flip-Flop

Das JK-Flip-Flop ist ein getakteter Speicher, dessen Ausgang  $q$  durch eine ansteigende Flanke in Abhängigkeit von den Steuer- bzw. Vorbereitungseingängen  $J$  und  $K$  beeinflusst wird (Abb. 7.5)

Zum Verständnis der Schaltung verwendet man die Wahrheitstabelle des Flip-Flops (Tab. 7.2). Dabei werden nicht nur die beiden Vorbereitungseingänge  $J^t$  und  $K^t$  als Eingangsvariable zur Zeit  $t$  betrachtet, sondern auch der Ausgang als  $q^t$ .

Wie aus der Wahrheitstabelle (untere Zeile) ersichtlich, reagiert das Flip-Flop nur bei ansteigenden Flanken am Eingang  $C$ .

**Abb. 7.5** JK-Flip-Flop mit NAND-Gattern



**Tab. 7.2** Wahrheitstabelle des JK-Flip-Flops mit NAND-Gattern

C	$J^t$	$K^t$	$q^t$	$q^{t+1}$
1	1	*	0	1
1	*	0	1	1
1	0	*	0	0
1	*	1	1	0
1	*	*	$q^t$	$q^t$

$q^{t+1}$	$J^t$				$K^t$			
	0	1	1	0		0	1	0
	1	1	0	0		1	1	0

Zeigt der Vorbereitungseingang  $J^t = 1$ , wird unabhängig vom Zustand des Eingangs  $K^t$  das Flip-Flop gesetzt (erste Zeile der Wahrheitstabelle Tab. 7.2). Reset erfolgt, wenn der Vorbereitungseingang  $K^t = 1$  aufweist, unabhängig vom Zustand des Eingangs  $J^t$  (zweite Zeile von unten in der Wahrheitstabelle).

Überträgt man die Wahrheitstabelle in ein Veitch-Diagramm, lässt sich eine Funktionsgleichung des JK-Flip-Flops angeben. Sie lautet:

$$q^{t+1} = J^t \bar{q}^t + \bar{K}^t q^t . \tag{7.2}$$

Bei Aufgaben wie beispielsweise einer Schrittkette bei Verpackungsanlagen kann man die Funktionsgleichung des JK-Flip-Flops zur Schaltungsdimensionierung mithilfe der booleschen Algebra einsetzen.

Als Schaltkreis 4027 ist ein 2-fach JK-Flip-Flop mit übergeordnetem RS-Flip-Flop (set dominant) verfügbar (Abb. 7.6).

**7.1.3 D-Flip-Flop**

Das D-Flip-Flop ist ebenfalls ein getakteter Speicher, dessen Ausgang  $q$  durch eine ansteigende Flanke in Abhängigkeit vom Dateneingang  $D$  beeinflusst wird (Abb. 7.7).

Zum Verständnis der Schaltung verwendet man auch hier die Wahrheitstabelle des Flip-Flops (Tab. 7.3). Das am Dateneingang zur Zeit  $t$  anliegende Signal  $D^t$  wird mit jeder

Die zugehörige Programmierung als Anweisungsliste (AWL) hat in STEP7 und CoDeSys dann folgende Struktur:

### STEP7 Programm CoDeSys Programm

```
FUNCTION FC400: BOOL
VAR_INPUT
  S1: BOOL;
  S2: BOOL;
  S3: BOOL;
  S4: BOOL;
END_VAR
BEGIN
  U #S1;
  U #S2;
  O ;
  O #S3;
  O #S4;
  = RET_VAL; //A bzw. P1
END_FUNCTION
```

```
FUNCTION FC400: BOOL
VAR_INPUT
  S1: BOOL;
  S2: BOOL;
  S3: BOOL;
  S4: BOOL;
END_VAR
LD S1
AND S2
OR S3
OR S4
ST FC400
```

Diese UND-ODER-LOGIK kann nun auch als Funktion aufgerufen und in anderen Schaltungsteilen mehrfach verwendet werden.

### Aufruf der Funktion in STEP7

```
ORGANIZATION:_BLOCK
VAR_TEMP
...//Standardeingaben
END_VAR
BEGIN
  CALL FC400 (
    S1 : = E0.1,
    S2 : = E0.2,
    S3 : = E0.3,
    S4 : = E0.4,
    RET_VAL : = A4.0);
END_ORGANIZATION_BLOCK
```

### Aufruf der Funktion in CoDeSys

```
PROGRAMM PLCPRG
VAR
... (* Standards *)
END_VAR
LD %IX0.1
FC400 %IX0.2, %IX0.3, %IX0.4
ST %QX4.0
```

---

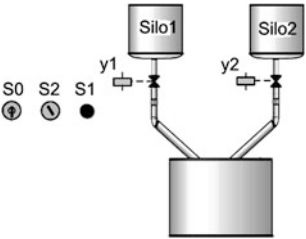
## 10.2 Anwendungen

### 10.2.1 Einfacher Mischvorgang

Ein Schüttgut soll aus zwei Komponenten gemischt werden (Abb. 10.4). Mit einem Wahlschalter S2 wird eine Auswahl zwischen den zwei verschiedenen Schüttgütern getroffen [41].



**Abb. 10.4** Einfacher Mischvorgang aus zwei Komponenten



**Tab. 10.4** Wahrheitstabelle der Schütz-Ansteuerung

S1	S2	S3	y1	y2
0	0	0	*	*
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	1	0
1	1	0	0	0
1	1	1	0	1

Steht S2 auf 0-Signal, gelangt das Schüttgut von Silo1 in den Mischbehälter, wenn gleichzeitig der Taster S1 betätigt wird. Schüttgut aus Silo2 gelangt mit der Schalterstellung S2 auf 1-Signal in den Mischbehälter, bei gleichzeitigem Betätigen von S1. Mit dem Schalter S0 wird der Mischvorgang abgeschaltet.

Die zugehörige Wahrheitstabelle für die Ansteuerung der Schütze y1 und y2, die die Ventile betätigen, hat demnach die Eingangsvariablen S0, S1 und S2 und ist in Tab. 10.4 dargestellt.

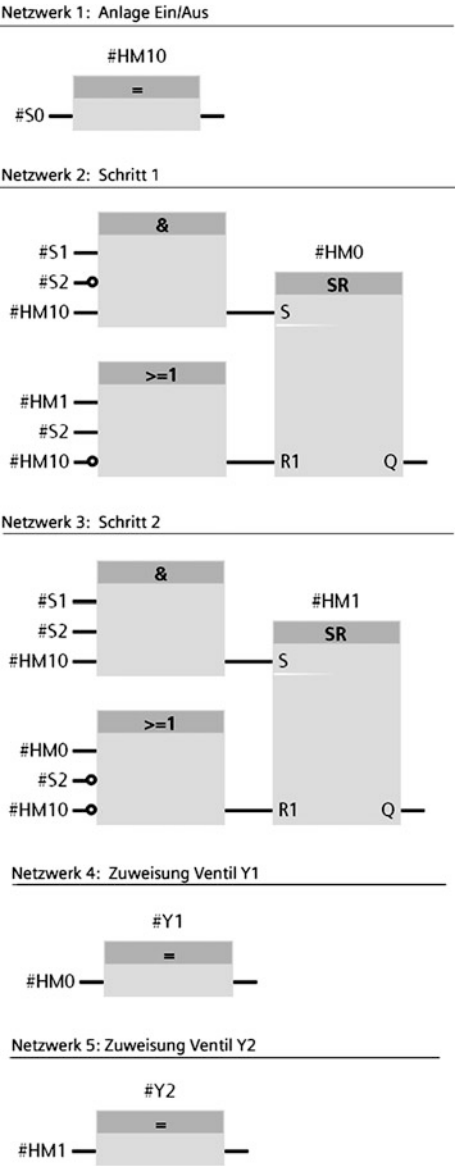
Aus dem Funktionsplan (FUP) ist zu erkennen, dass man die Schalterstellungen als RS-Flip-Flop-Funktion realisiert (Abb. 10.5).

**10.2.2 Automatische Stern-Dreieck-Schaltung**

Bei schwachen Drehstrom-Netzen und mit Rücksicht auf die einschlägigen Bestimmungen der Elektroversorgungsunternehmen muss der Anlaufstrom größerer Asynchronmotoren begrenzt werden.

Dies kann durch eine Stern-Dreieck-Schaltung geschehen. In Abb. 10.6 ist eine automatische Stern-Dreieck-Schützschaltung mit Wirkschalt- und Stromlaufplan dargestellt. Dabei wird das Moment und der Strom beim Anfahren auf 1/3 reduziert [15]. Mit dem Taster S1 wird der Motor zunächst eine begrenzte Zeit (am Zeitrelais Z eingestellt) in Stern-Schaltung angefahren. Danach geht die Schützschaltung automatisch auf die Dreieck-Schaltung über. Mit Taster SO kann der Motor jederzeit angehalten werden.

**Abb. 10.5** Funktionsplan (FUP) des Mischvorgangs



Die Deklaration der Variablen und Symbole ist in Tab. 10.5 und der FUP der Schaltung in Abb. 10.7 dargestellt.

Eine andere Variante des FUP der Stern-Dreieck-Schaltung zeigt Abb. 10.8.

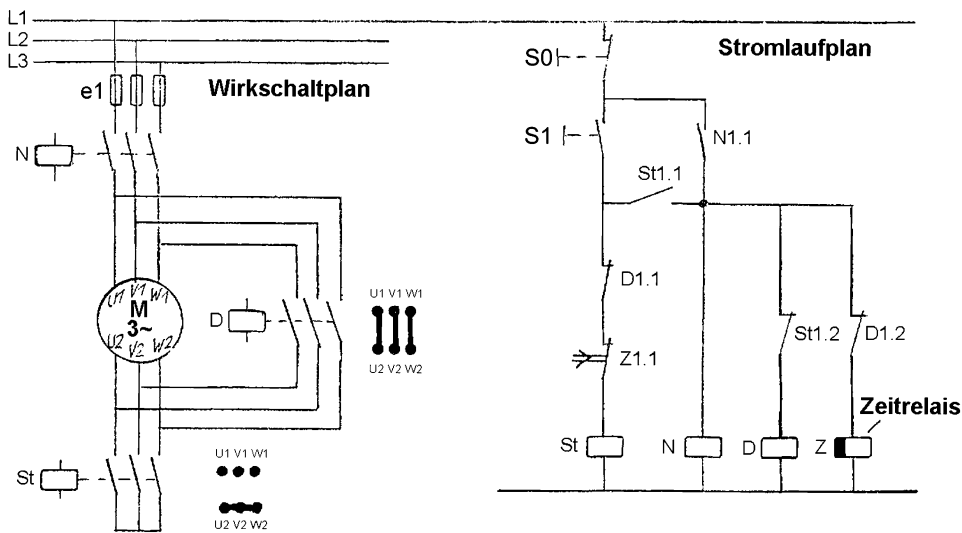


Abb. 10.6 Wirkschluplan und Stromlaufplan der Stern-Dreieck-Schütz-Schaltung

Tab. 10.5 Datentypen und Symbole

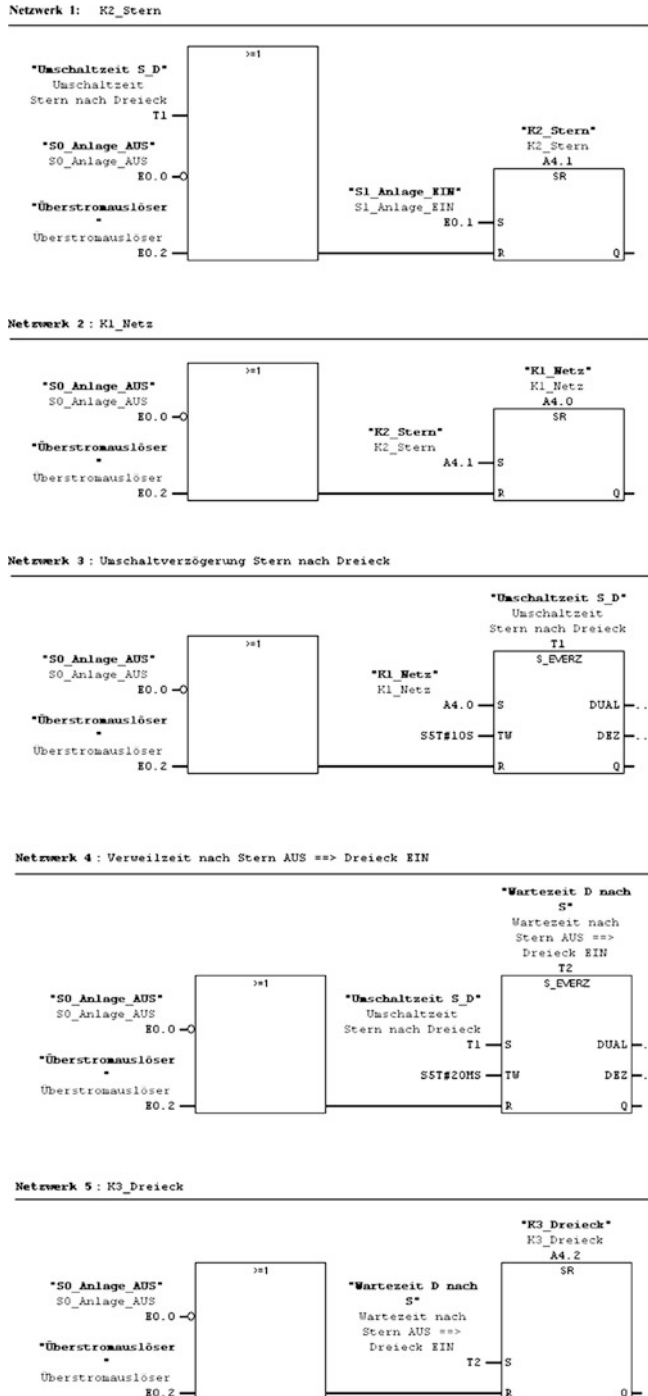
Symbol 	Adresse	Datentyp	Kommentar
K1_Netz (N)	A 4.0	BOOL	K1_Netz
K2_Stern (ST)	A 4.1	BOOL	K2_Stern
K3_Dreieck (D)	A 4.2	BOOL	K3_Dreieck
S0_Anlage_AUS	E 0.0	BOOL	S0_Anlage_AUS
S1_Anlage_EIN	E 0.1	BOOL	S1_Anlage_EIN
Stern-Dreieck-Anlauf	FC 1	FC 1	
Überstromauslöser	E 0.2	BOOL	Überstromauslöser
Umschaltzeit S_D	T 1	TIMER	Umschaltzeit Stern nach Dreieck
Wartezeit D nach S	T 2	TIMER	Wartezeit nach Stern AUS ==> Dreieck EIN

10.2.3 Torsteuerung

Ein Werktor soll automatisch betätigt werden. Zwei Endlagenschalter geben die Endpositionen wieder und eine Sicherheitskontaktleiste sorgt für den Personenschutz. Der Antrieb über einen Drehstrommotor wird über die beiden Schütze K1 und K2 angesteuert (Abb. 10.9) [42].

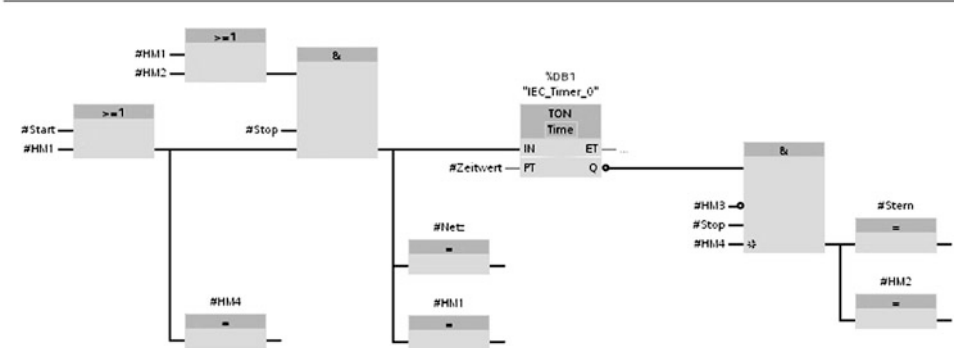
Die Anlage wird über die Taster S0 (E0.0)/S1 (E0.1) ein- bzw. ausgeschaltet. Der Ein-Zustand wird mit der Meldeleuchte H0 (A5.0) signalisiert.

Im Automatikbetrieb S3 (E0.3) EIN, läuft das Tor in die mit den Tastern S4 (E0.4) und S5 (E0.5) vorgewählte Richtung bis in die zugehörige Endlage.



**Abb. 10.7** Funktionsplan (FUP) der Stern-Dreieck-Schaltung [42]

Netzwerk 1:



Netzwerk 2:

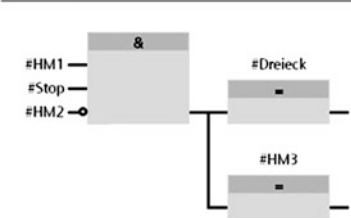


Abb. 10.8 FUP einer Stern-Dreieck-Schaltung nach [41]

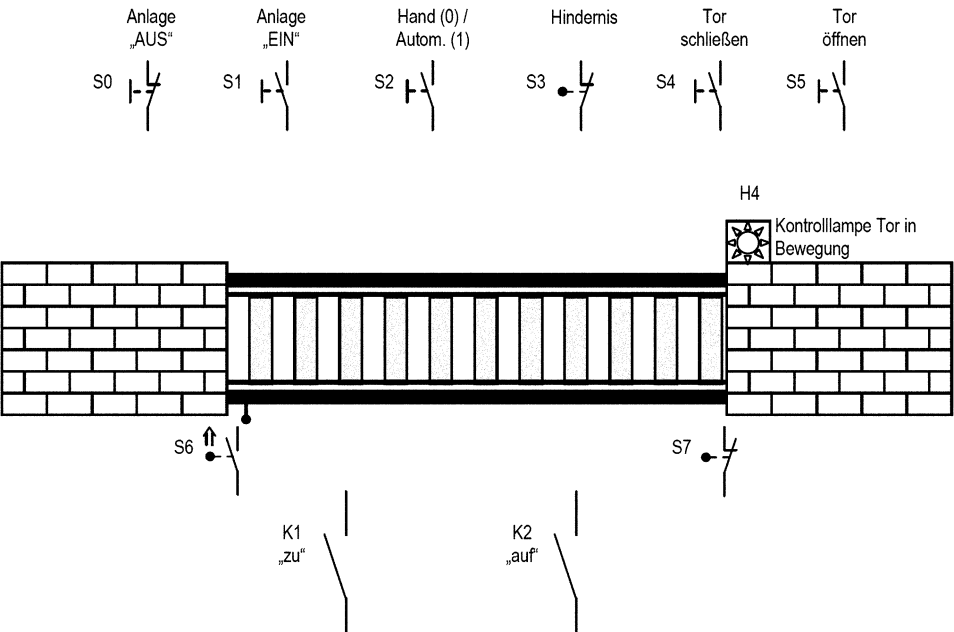


Abb. 10.9 Prinzipaufbau der Torsteuerung

**Tab. 10.6** Symbol-, Adress- und Datentyp-Zuordnung

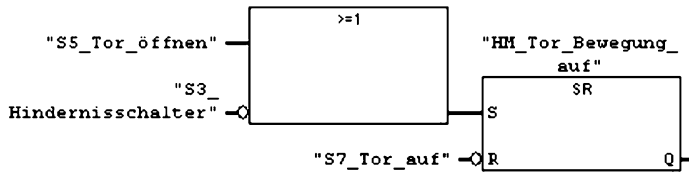
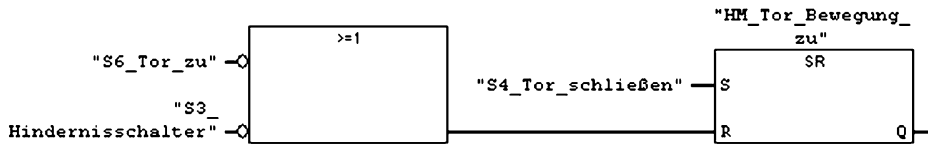
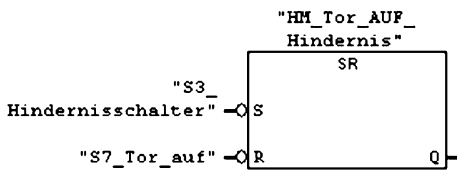
	Symbol	Adresse	Datentyp	Kommentar
1	S0 Anlage_AUS	E0.0	BOOL	Anlage „AUS“ / 0 = AUS
2	S1 Anlage_EIN	E0.1	BOOL	Anlage „EIN“ / 1 = EIN
3	S2 Anlage_Autom	E0.2	BOOL	Anlage Automatik / 1 = Autom.
4	S3 Hindernisschalter	E0.3	BOOL	Hindernisschalter / 0 = Hindemis
5	S4_Tor_schließen	E0.4	BOOL	Taster Tor schließen / 1 = zu
6	S5_Tor_öffnen	E0.5	BOOL	Taster Tor öffnen / 1 = auf
7	S6_Tor_zu	E0.6	BOOL	Endschalter Tor zu / 0 = zu
8	S7_Tor_auf	E0.7	BOOL	Endschalter Tor auf / 0 = offen
9	K1_Tor_zu	A4.0	BOOL	Ansteuerung Ventil Tor zu
10	K2_Tor_auf	A4.1	BOOL	Ansteuerung Ventil Tor auf
11	K3_Hauptluft	A4.2	BOOL	Ansteuerung Ventil Luftzufuhr
12	H0_ML_Anlage_EIN	A5.0	BOOL	ML Anlage ist eingeschaltet
13	H1_ML_Anlage_Auto	A5.1	BOOL	ML Anlage ist im Automatikbetrieb
14	H2_Blinkleuchte_TOR	A5.2	BOOL	Blinklampe TOR in Bewegung
15	H3_ML_Tor_zu	A5.3	BOOL	ML Tor ist geschlossen
16	H4_ML_Tor_auf	A5.4	BOOL	ML Tor ist geöffnet
17	HM_1Hz	M100.5	BOOL	Blinktakt 5Hz
18	HM_5Hz	M100.1	BOOL	Blinktakt 1Hz
19	Immer low	M0.0	BOOL	Immer low
20	Immer high	M0.1	BOOL	Immer high
21	HM_Anlage_EIN	M0.2	BOOL	Hilfsmerker Anlage ist eingeschaltet
22	HM_Anlage_Auto	M0.3	BOOL	Hilfsmerker Anlage ist in Automatik
23	HM_Tor_Bewegung_zu	M0.4	BOOL	Hilfsmerker Tor läuft zu
24	HM_Tor_Bewegung_auf	M0.5	BOOL	Hilfsmerker Tor läuft auf
25	HM_Tor_AUF_Hindernis	M0.6	BOOL	HM Tor läuft auf nach Hindernis erkan

Wenn bei dem Zulaufen die Sicherheitskontaktleiste S3 (E0.3) anspricht, läuft das Tor sofort wieder auf, die Kontrolllampe H2 (A5.2) am Tor blinkt dann mit 5 Hz, bis sich das Tor wieder in der geöffneten Stellung befindet. Während dieser Zeit kann kein erneuter Schließ-Befehl ausgelöst werden.

Die Endlagen Tor AUF und ZU werden über die Endschalter S6 (E0.6)/S7 (E0.7) an den Meldeleuchten H3 (A5.3) für ZU und H4 (A5.4) für AUF signalisiert. Im Handbetrieb wirkt die gleiche Funktion wie im Automatikbetrieb, jedoch ist die Sicherheitskontaktleiste außer Funktion (Tab. 10.6).

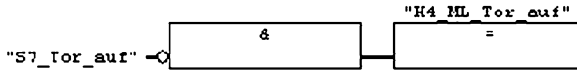
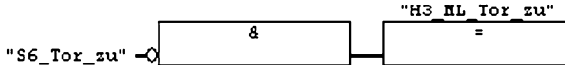
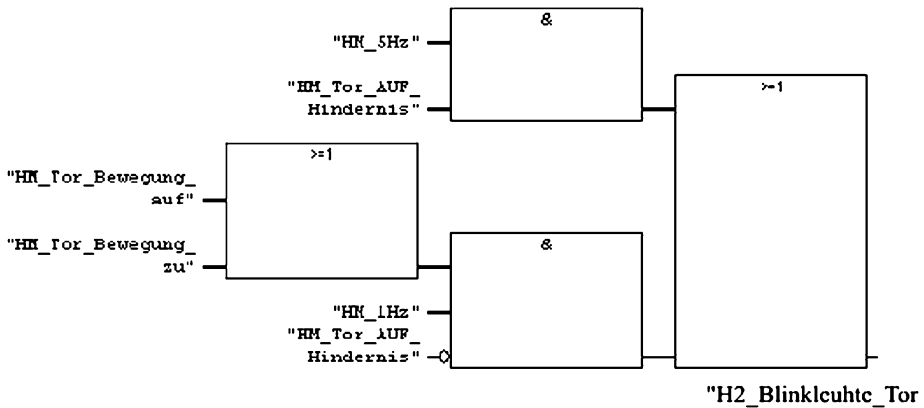
Beim Normalbetrieb des Öffnens und Schließern blinkt die Kontrolllampe H2 (A5.2) am Tor mit einer Frequenz von 1 Hz.

Der Funktionsplan (FUP) für Automatikbetrieb ist in den Abb. 10.10 und 10.11 dargestellt.

**Netzwerk 1:** Hilfsmarker Tor 1 läuft auf**Netzwerk 2:** Hilfsmerker Tor läuft zu**Netzwerk 3:** HM Tor läuft auf nach Hindernis erkannt**Netzwerk 4:** Ansteuerung Tor auf**Abb. 10.10** FUP der Torsteuerung (Teil 1)**10.2.4 Ampelsteuerung**

Ein Fußgängerüberweg soll mit einer Ampelanlage versehen werden. Die Funktionsweise wurde bereits für eine Hardware-Lösung in CMOS-Technik im Abschn. 8.4 erklärt.

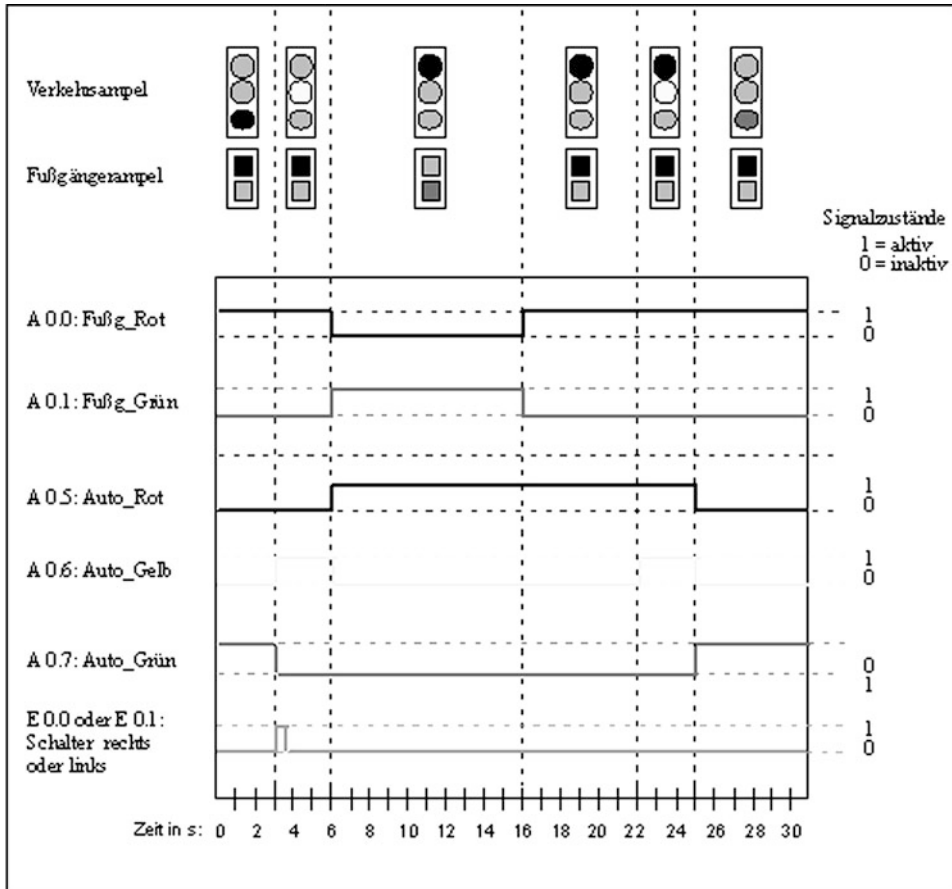
Am Tage wird der reguläre Zyklus gefahren. Durch den Schalter am Eingang 0.0 wird für Nachtbetrieb ein Blinkvorgang eingestellt, der die Straßenbenutzer auf einen Fußgängerüberweg hinweisen soll [42]. Um für den Zähler einen Takt zu erhalten rufen wir über STEP7, SIMATIC S7 300 die Hardware-Konfiguration auf. Über die aufgerufene CPU wird das Register Zyklus/Taktmerker geöffnet, der Taktmerker aktiviert und das MB 100 zugeordnet, danach gespeichert und übertragen.

**Netzwerk 5: Ansteuerung Tor zu****Netzwerk 6: ML Tor ist geöffnet****Netzwerk 7: ML Tor ist geschlossen****Netzwerk 8: Blinklampe TOR in Bewegung****Abb. 10.11** FUP der Torsteuerung (Teil 2)

Da das Taktmerkerbyte Frequenzen von 0.5 Hz im Bit 7 bis 10 Hz im Bit 0 anbietet, wird in dieser Anwendung die Frequenz von 1 Hz also das Bit 5 ausgewählt.

Nachdem der Zähler zu zählen begonnen hat, müssen entsprechend der Logik für die Ampelphasen die verschiedenen Leuchten (Rot, Gelb, Grün) angesteuert werden. Der





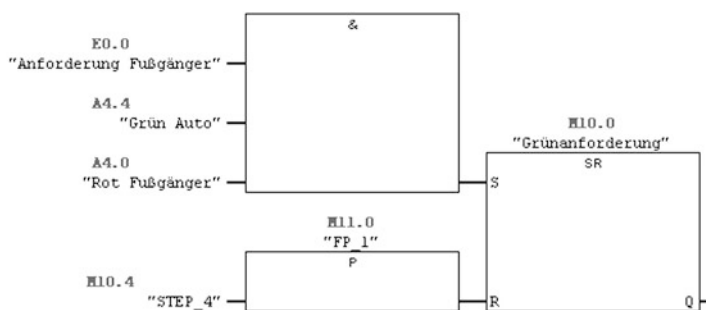
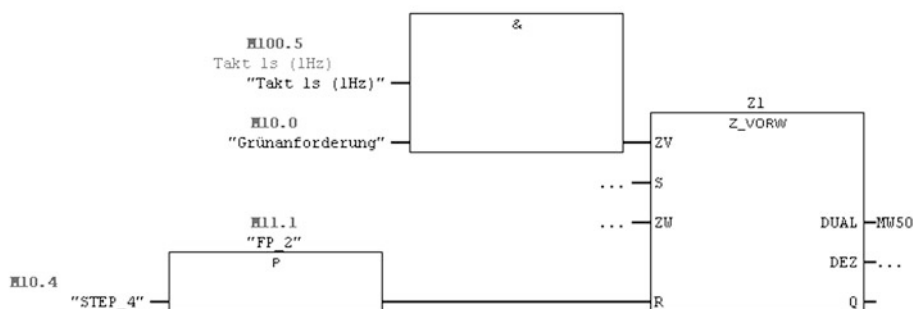
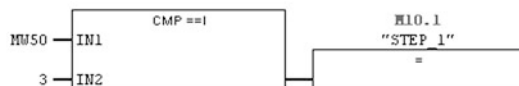
**Abb. 10.12** Ablaufplan der Ampelphasen

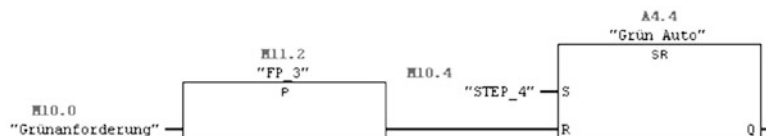
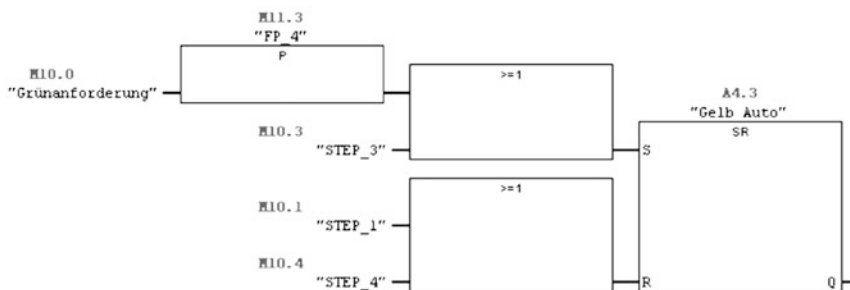
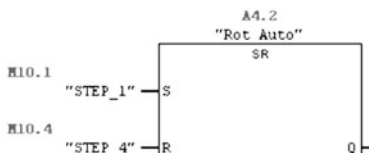
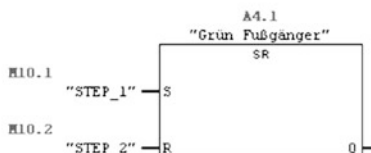
Ablaufplan der einzelnen Ampelphasen für die Fußgänger- und Verkehrsampel ist in Abb. 10.12 dargestellt.

Die Festlegung der Symbolik, Adressen, Datentypen und zugehörigen Kommentare zur Programmierung zeigt die folgende Tab. 10.7.

In den Abb. 10.13 und 10.14 ist der Funktionsplan (FUP) zur Steuerung der Ampelphasen für den Fußgängerüberweg und den querenden Autoverkehr dargestellt.

Weitere Varianten zum Thema Ampelsteuerung finden sich in [9] und [41].

**Netzwerk 1: Grünanforderung durch Fußgänger****Netzwerk 2: Taktgeber****Netzwerk 3: Verzögerung bis Gelb für Auto****Netzwerk 4: Fußgänger Grünphase****Abb. 10.13** FUP der Ampelsteuerung (Teil 1)

**Netzwerk 5:** Verzögerung bis Gelb/Rot für Auto**Netzwerk 6:** Verzögerung bis Grün für Auto**Netzwerk 7:** Grün Auto**Netzwerk 8:** Gelb Auto**Netzwerk 9:** Rot Auto**Netzwerk 10:** Grün Fußgänger**Abb. 10.14** FUP der Ampelsteuerung (Teil 2)

**Tab. 10.7** Symbole, Adressen und Datentypen zur Ampelsteuerung

Symbole	Adressen	Datentypen	Kommentare
ROT_Fussgaenger	A4.0	BOOL	Rot für Fußgänger
GRUEN_Fussgaenger	A4.1	BOOL	Grün für Fußgänger
ROT_Auto	A4.2	BOOL	Rot für Pkw
GELB_Auto	A4.3	BOOL	Gelb für Pkw
GRUEN_Auto	A4.4	BOOL	Grün für Pkw
Anforderung Fussgaenger	E0.0	BOOL	Anforderung Fußgänger
Gruenanforderung	M10.0	BOOL	
STEP_1	M10.1	BOOL	
STEP_2	M10.2	BOOL	
STEP_3	M10.3	BOOL	
STEP_4	M10.4	BOOL	
FP_1	M11.0	BOOL	
FP_2	M11.1	BOOL	
FP_3	M11.2	BOOL	
FP_4	M11.3	BOOL	
Takt 1 s (1 Hz)	M100.5	BOOL	Taktfrequenz 1 Hz

## 10.2.5 Ablaufsteuerung

Bereits in Abschn. 8.6 wurde eine Ablaufsteuerung (Schritt看ette) in CMOS-Technik besprochen.

Jeder Schritt einer solchen Steuerung muss abgearbeitet sein, bevor der nächste freigegeben wird. Dazu sind Speicher notwendig. Es gibt zwei Varianten, den Ablauf auch über Leuchtmelder darzustellen. Entweder zeigen die Leuchten nacheinander, Schritt für Schritt an, oder nach jedem Schritt wird der vorhergehende Speicher wieder gelöscht bzw. der zugehörige Leuchtmelder ausgeschaltet.

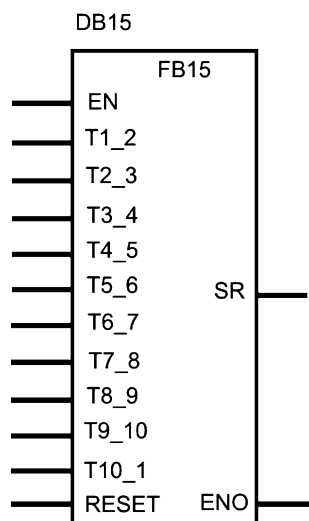
Für einfache gleichartige Abläufe ohne Ablauf-Verzweigungen, -Schleifen oder -Sprünge kann auch der Standard-Ablaufkettenbaustein FB15 benutzt werden (Abb. 10.15 und Tab. 10.8).

Mit einem RESET-Befehl wird die interne Schritt看ette auf den Anfangszustand gesetzt. Es gibt jedoch keine Befehlsausgabe für jeden Schritt, sondern nur die Ausgabe einer Schritt看ummer am Ausgang SR als INTEGER-Zahl.

Für eine Schritt看ette mit dem Einsatz des FB15 trennt man die 2 Arbeitsbereiche einer Schritt看ette auf in:

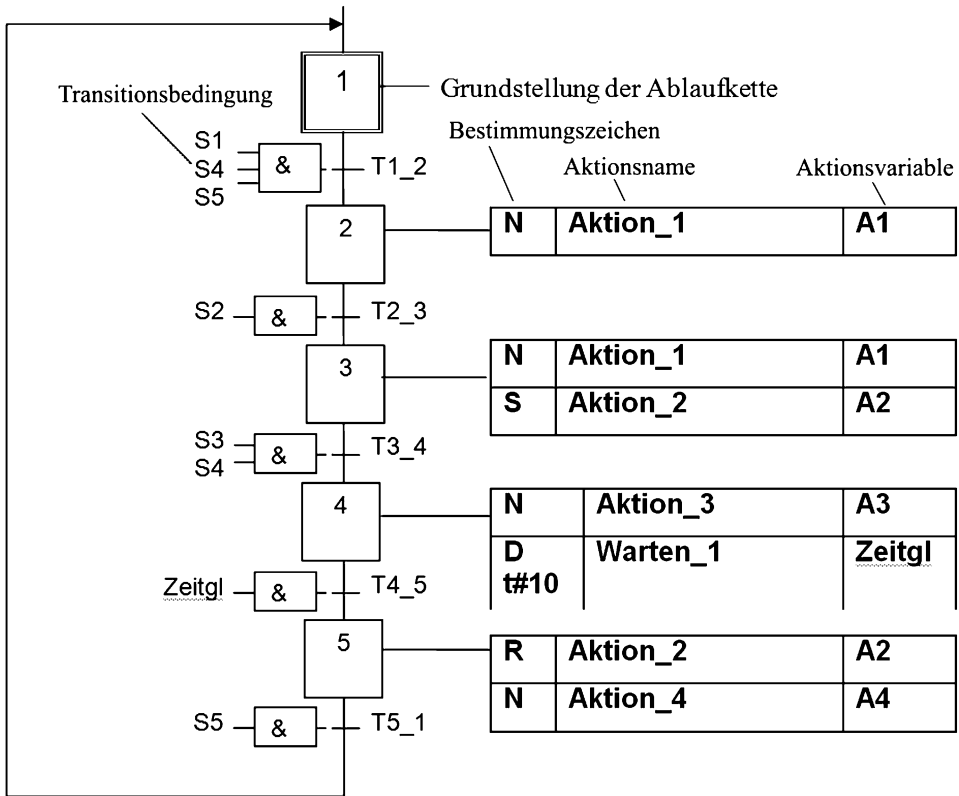
- Kontrolle der Übergangsbedingungen und Erzeugen einer Schritt看ummer
- Ansteuerung der Ausgabebefehle in Abhängigkeit der Schritt看ummer

**Abb. 10.15** Funktionsblock  
FB15



**Tab. 10.8** Deklaration des Schrittkettenbausteins FB15

FB 15 Netzwerk 1						
Adresse	Deklaration	Name	Typ	Anfangswert	Kommentar	
0.0	in	T1_2	BOOL	FALSE		
0.1	in	T2_3	BOOL	FALSE		
0.2	in	T3_4	BOOL	FALSE		
0.3	in	T4_5	BOOL	FALSE		
0.4	in	T5_6	BOOL	FALSE		
0.5	in	T6_7	BOOL	FALSE		
0.6	in	T7_8	BOOL	FALSE		
0.7	in	T8_9	BOOL	FALSE		
1.0	in	T9_10	BOOL	FALSE		
1.1	in	T10_1	BOOL	FALSE		
1.2	in	RESET	BOOL	FALSE		
2.0	out	Schritt	INT	1		
	in_out					
4.0	stat	SR01	BOOL	TRUE		
4.1	stat	SR02	BOOL	FALSE		
4.2	stat	SR03	BOOL	FALSE		
4.3	stat	SR04	BOOL	FALSE		
4.4	stat	SR05	BOOL	FALSE		
4.5	stat	SR06	BOOL	FALSE		
4.6	stat	SR07	BOOL	FALSE		
4.7	stat	SR08	BOOL	FALSE		
5.0	stat	SR09	BOOL	FALSE		
5.1	stat	SR10	BOOL	FALSE		
	temp					



**Abb. 10.16** Prinzipdarstellung einer Schrittkette

Auf diese Weise kann man das Umsetzen der Übergangsbedingungen in eine Schrittnummer mittels des Bausteins FB15 realisieren.

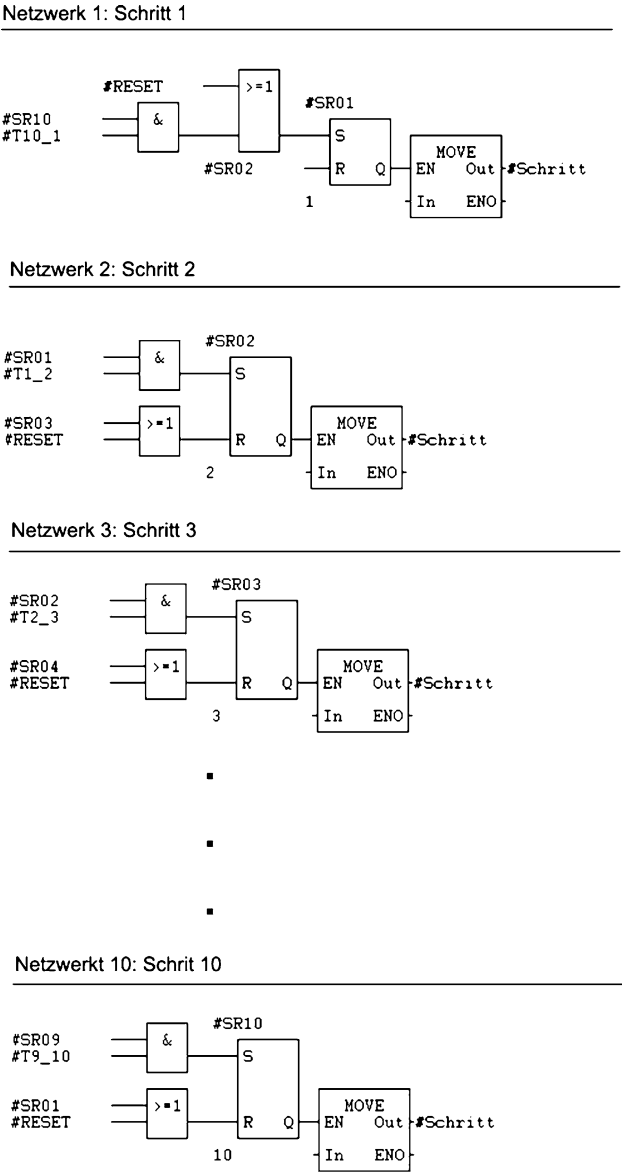
Der Baustein FB15 wird im OB1 aufgerufen und dort mit den Transitionsbedingungen beschaltet. Er hat die Aufgabe, die Transitionsbedingungen zu verarbeiten, d. h. die aktuelle Schrittnummer auszugeben. Die zugehörige Variable muss im OB1 deklariert werden (z. B. OB\_SRNR).

Eine Schrittkette besteht also nun aus zwei Bausteinen, die im OB1 aufgerufen werden:

- FB15: Hier wird der Schritkettenablauf realisiert, d. h. immer wenn eine Transitionsbedingung erfüllt ist, wird die Variable Schrittnummer erhöht.
- FC1: Hier werden die Aktionen festgelegt über Vergleiche, d. h. wenn beispielsweise die Schrittnummer 3 anliegt, dann wird z. B. eine Leuchte aktiviert, bei Schrittnummer 4 z. B. ein Motor für Linkslauf aktiviert usw.

Prinzipiell hat eine Schrittkette dann die in Abb. 10.16 dargestellte Form, wenn nach dem letzten Schritt wieder auf die Anfangstellung gegangen werden soll.

**Abb. 10.17** FUP einer Schritt-  
kette bei Verwendung des FB15  
in STEP7



Die Realisierung einer solchen Schritt-kette mit zyklischem Ablauf, also dem Restart nach dem letzten Schritt, ist in Abb. 10.17 dargestellt. Die Schaltung bzw. Programmierung der Schritte 2 bis 9 sehen gleich aus.

Dabei muss die Transitionsbedingung von 1 nach 2 um ein ODER im OB1 bei der Beschaltung von FB15 ergänzt werden. Der erste Eingang dieses ODER ist z. B. ein START-Taster, der zweite Eingang ist ein Startmerker, sodass beim nächsten Durchlauf wegen des

gesetzten Startmerkers gleich in Schritt 2 übergeleitet wird. Der Startmerker wird im OB1 über ein RS-Flip-Flop gesetzt als zusätzliches Netzwerk im OB1.

Die für jeden Schritt notwendigen RS-Flip-Flops werden durch die Transitionsbedingungen gesetzt und durch den jeweils nachfolgenden Schritt wieder rückgesetzt (SR01 bis SR10). Nach ausgeführtem Schritt 10 wird das zugehörige Flip-Flop durch den Rücksprung auf Schritt 1 wieder rückgesetzt und der Zyklus der Schrittkette kann von Neuem beginnen.

Weitere Beispiele zu Ablaufsteuerungen finden sich in [8, 9 und 41].

## 10.2.6 Regelung mit PID-Regler

Die Gl. 2.41 des PID-Reglers

$$y(t) = K_R \left[ x_d(t) + \frac{1}{T_N} \int_0^t x_d(\tau) \cdot d\tau + T_V \frac{dx_d(t)}{dt} \right]$$

geht in den folgenden digitalen PID-Stellungsalgorithmus über [6]:

$$y(kT_z) = K_R \left\{ x_d(kT_z) + \frac{T_z}{T_N} \cdot \sum_{i=0}^k x_d(iT_z) + \frac{T_V}{T_z} \cdot [x_d(kT_z) - x_d(kT_z - 1)] \right\} \quad (10.1)$$

Darin sind:

$k$ : Laufindex

$T_z$ : Zeit für eine Rechenoperation bzw. Abtastzeit

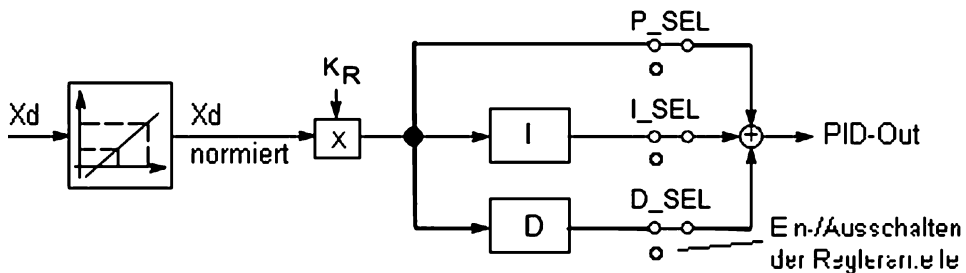
$T_N$ : Nachstell- bzw. Integrationszeitkonstante

$T_V$ : Vorhalt- bzw. Differenziationszeitkonstante

$y$ : Stellgröße

$x_d$ : Regeldifferenz

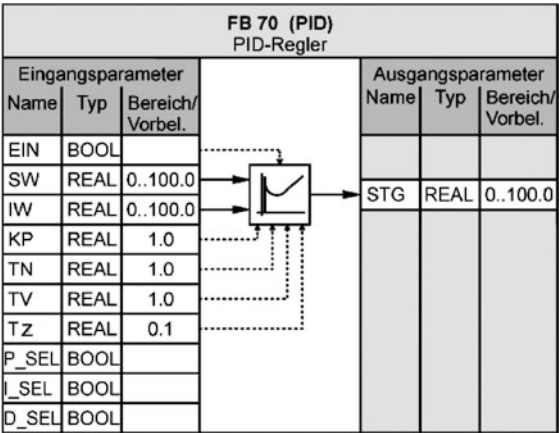
$K_R$ : Proportionalverstärkung



**Abb. 10.18** PID-Regler als Summenverzweigung



**Tab. 10.9** Deklaration des Reglerbausteins FB70



Der Algorithmus lässt sich aus den drei Zweigen Proportionalanteil, Integralanteil und Differenzialanteil mit der gemeinsamen Reglerverstärkung  $K_R$  darstellen (Abb. 10.18).

Durch Zu- und Abschalten einzelner Zweige können auch der PI-, PD- oder der P-Regler realisiert werden.

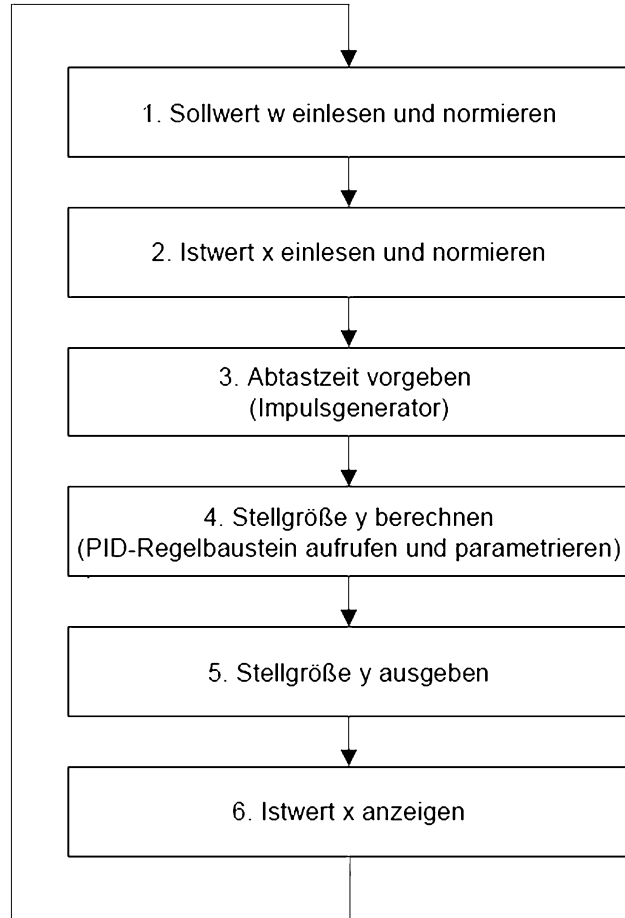
Der PID-Stellungsalgorithmus steht als Baustein FB70 zur Verfügung. Die Deklaration der Variablen ist in Tab. 10.9 dargestellt.

Die Programmierung in STEP7 sieht wie folgt aus:

**Quelltext**

```
FUNCTION_BLOCK FB70 (*PID*)
(* xd(kTz)=XD xd(kTz-1)=XD1 *)
VAR_INPUT
    EIN: BOOL;
    P_SEL, I_SEL, D_SEL: BOOL;
    W, X: REAL;
    KR: REAL:=1.0; TN: REAL:=1.0;
    TV: REAL:=1.0; TZ: REAL:=1.0;
END_VAR
VAR_OUTPUT
    STG: REAL;
END_VAR
VAR
    EK: REAL;
    STGI: REAL;
    STGD: REAL;
    ESUM: REAL;
END_VAR
IF EIN = FALSE THEN (*Programmierung des FB70*)
    STG:=0.0; XD1:=0.0; XDSUM:=0.0;
    RETURN;
END_IF;
STG:=0.0; XD:=KR*(W - X); XDSUM:=XDSUM + XD;
STGI:=XDSUM*TZ/TN; STGD:=TV/TZ*(XD - XD1); XD1:=XD;
IF P_SEL=TRUE THEN STG:=STG+EK; END_IF;
IF I_SEL=TRUE THEN STG:=STG+STGI; END_IF;
IF D_SEL=TRUE THEN ST:=STG+STGD; END_IF;
IF STG<0.0 THEN STG:=0.0
    ELSIF STG>100.0 THEN STG:=100.0;
END_IF;
END_FUNCTION_BLOCK
```

**Abb. 10.19** Ablaufplan einer Regelung mit STEP7

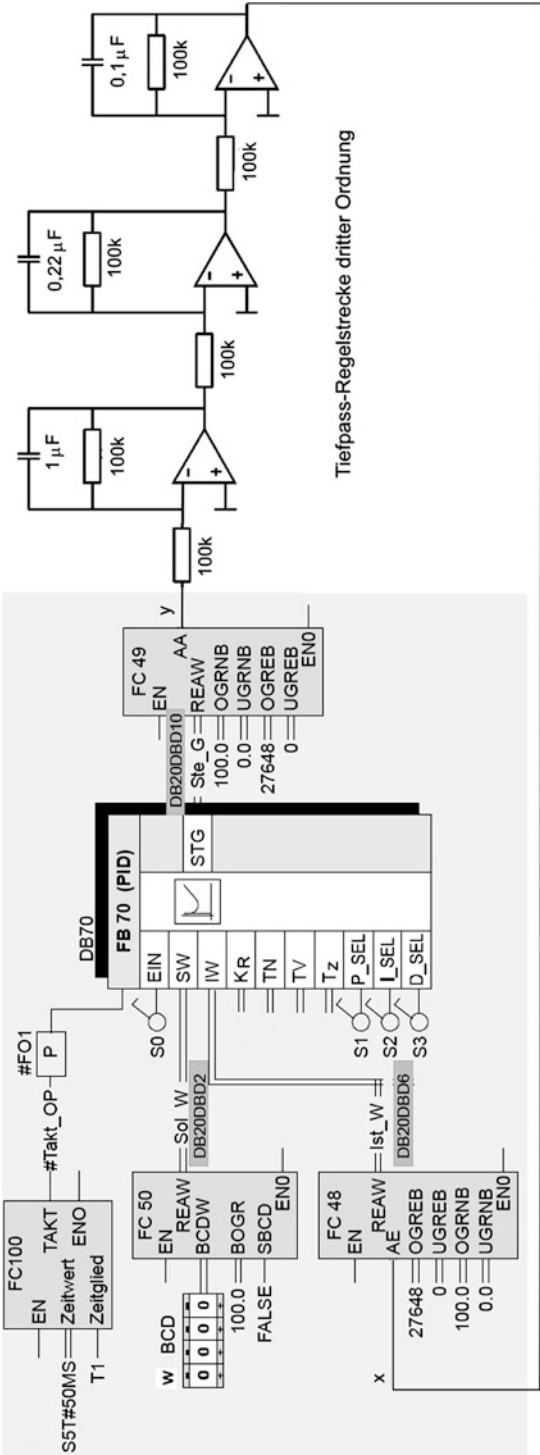


Der Ablaufplan zur Realisierung einer Regelung in STEP7 ist im folgenden Diagramm dargestellt (Abb. 10.19).

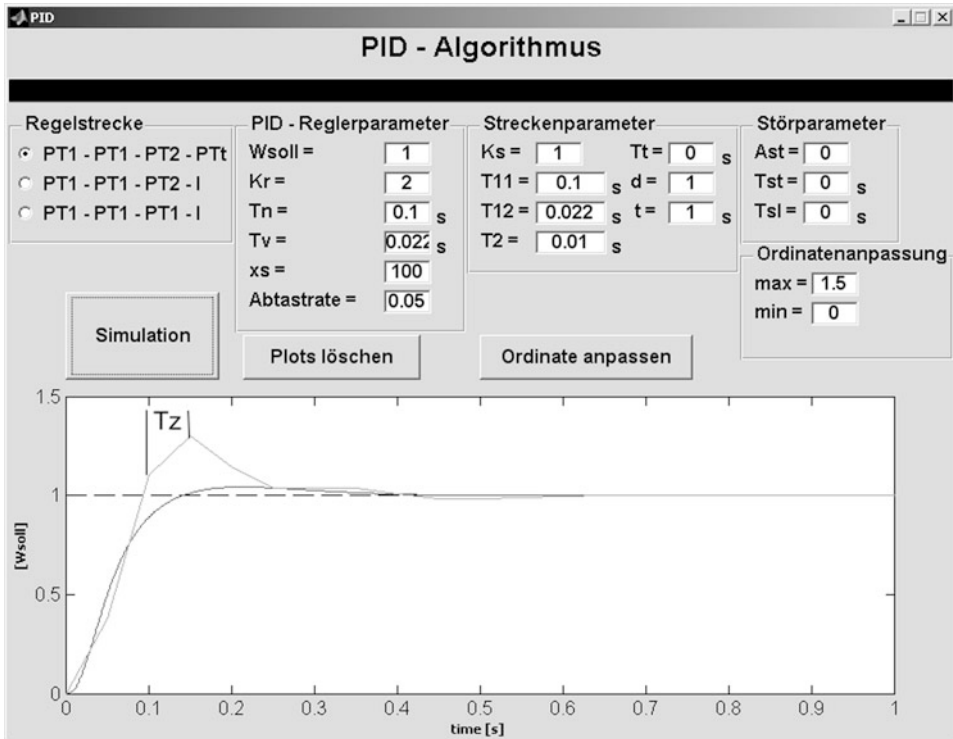
Mit dem Funktionsbaustein FB50 wird der im BCD-Code vorliegende Sollwert (Führungsgröße  $w$ ) eingelesen und auf das Datenformat REAL normiert. Der Istwert (Regelgröße  $x$ ) kann mit dem Funktionsbaustein FB48 eingelesen und die Stellgröße  $y$  mit dem Funktionsbaustein FB49 ausgegeben werden.

Eine Simulation, bei der der Reglerbaustein FB70 auf eine Strecke dritter Ordnung wirkt, soll den Einfluss der Abtastzeit  $T_z$  auf die Sprungantwort des Regelkreises zeigen (Abb. 10.20).

Die Regelstrecke wird dabei durch drei aktive Tiefpässe mit den Zeitkonstanten  $T_{11} = 0,1 \text{ s}$ ,  $T_{12} = 0,022 \text{ s}$  und  $T_{13} = 0,01 \text{ s}$  entsprechend Abb. 2.35 und Gl. 2.56 dargestellt. Der PID-Regler nach Abb. 2.22 und Gl. 2.38 wird so eingestellt, dass  $T_N = T_{11}$  und  $T_V = T_{12}$  sind, d. h. die Bedingung  $T_N \gg T_V$  erfüllt wird. Die Reglerverstärkung wird auf



**Abb. 10.20** FUP des PID-Reglers mit FB70 und Strecke dritter Ordnung



**Abb. 10.21** Sprungantworten der Regelung aus PID-Regler und  $PT_3$ -Strecke

$K_R = 2$  eingestellt. Diese Optimierung ergibt eine Sprungantwort ohne größeres Überschwingen [6].

Mit einer MATLAB Simulink Oberfläche [18] kann nun der Einfluss von  $T_z$  auf die Sprungantwort der Regelung untersucht werden (Abb. 10.21).

Es zeigt sich, dass die Abtastzeit  $T_z$  demnach einen erheblichen Einfluss auf die Güte der gesamten Regelung hat. Insbesondere, wenn  $T_z$  in der Größenordnung der beteiligten Prozesszeitkonstanten (Streckenzeitkonstanten) liegt, wird das Übergangsverhalten erheblich gestört (hier bei  $T_z = 0,05$  s).

Weiterführende Literatur und Links zum Themenbereich SPS finden sich unter [41, 44, 45].

---

## 11.1 Schnittstellen- und Bussysteme

Die verschiedensten Pegelumsetzer wurden in den Abschn. 3.8 und 7.9 bereits beschrieben. Die wesentlichen Übertragungsformen von Signalen und Daten sowie deren Pegel werden hier nochmals ergänzend dargestellt.

**Analoge Schnittstellen** In der Analogtechnik ist die Information als Spannung oder Strom eingepreßt. Es gibt industriell genutzt die drei folgenden Pegelvarianten:

0 V bis + 10 V  
–10 V bis + 10 V  
0(4) mA bis 20 mA .

**RS232, RS422, RS485 Serielle Schnittstellen** RS232 ist ein Standard für eine bei Computern oft vorhandene serielle Schnittstelle, die in den 1960er-Jahren von einem US-amerikanischen Standardisierungskomitee eingeführt wurde.

Die Signalleitung übertragen folgende Pegel:

+ 3 V bis + 15 V ( + 12 V üblich) aktiver Zustand  
– 3 V bis – 15 V ( – 12 V üblich) inaktiver Zustand .

Die angegebenen Spannungen beziehen sich auf die Empfänger (Eingänge). Bei den Sendern (Ausgänge) muss die Spannung mindestens +5 V bzw. –5 V an einer Last von 3 ... 7 k $\Omega$  betragen, um genügend Störabstand zu gewährleisten.

Die Norm RS232 beschreibt die serielle Verbindung zwischen einem Datenendgerät (DTE) und einer Daten-Übertragungseinrichtung (DCE) mit ihren elektrischen und mechanischen Eigenschaften. Obwohl die Norm lediglich diesen Verbindungstyp definiert, hat sich die RS232-Schnittstelle als genereller Standard für serielle Datenübertragungen

über kurze Distanzen etabliert. DTE und DCE unterscheiden sich grundsätzlich in der Belegung ihrer Steckverbinder.

PCs, Drucker, Plotter oder der Main Port eines Terminals sind mit einer DTE-Belegung ausgestattet, während Modems und Drucker-Ports von Terminals DCE-Belegungen aufweisen. Die RS232-Norm definiert als Standard-Steckverbindung einen 25-poligen SUB-D-Stecker.

Die erzielbare Entfernung zwischen zwei RS232-Geräten ist wie bei allen seriellen Übertragungsverfahren vom verwendeten Kabel und der Baudrate abhängig. Als Richtmaß sollte bei einer Übertragungsrate von 9600 Baud eine Distanz von 15 bis 30 Metern nicht überschritten werden.

RS232-Schnittstellen besitzen eine Vielzahl von Handshake-Leitungen, die jedoch in Ihrer Gesamtheit lediglich zur Verbindung eines Modems mit einem Datenendgerät benötigt werden. Der weitaus häufigere Fall der Verbindung zweier Datenendgeräte miteinander lässt sich in der Regel mit einer reduzierten Anzahl von Handshake-Leitungen ohne Probleme realisieren. Nicht benötigte Handshake-Eingänge werden einfach durch Verbindung mit den eigenen Handshake-Ausgängen auf Freigabepegel gelegt.

Im Gegensatz zur RS232 arbeitet die RS423 jedoch lediglich mit Ausgangspegeln von  $\pm 4 \dots 6$  Volt, während die Empfängerbausteine, die baugleich mit RS422-Empfängern sind, noch Pegel von  $\pm 200$  mV als gültiges Signal erkennen müssen.

Die RS423 ist zur Übertragung von Daten mit einer Geschwindigkeit von bis zu 100 kBaud und über eine Entfernung von bis zu 1200 Metern geeignet. Maximal 10 Empfänger dürfen gleichzeitig mit einem Sender verbunden werden. RS423-Schnittstellen sind in der Praxis eher selten anzutreffen, da die mit gleichem Aufwand verbundene RS422-Schnittstelle gegenüber RS423-Verbindungen den Vorteil der deutlich höheren Übertragungssicherheit bietet.

Die RS485-Schnittstelle stellt eine Erweiterung der RS422-Definition dar. Während die RS422 lediglich den unidirektionalen Anschluss von bis zu 10 Empfängern an einen Sendebaustein zulässt, ist die RS485 als bidirektionales Bussystem mit bis zu 32 Teilnehmern konzipiert. Physikalisch unterscheiden sich beide Schnittstellen nur unwesentlich.

Da mehrere Sender auf einer gemeinsamen Leitung arbeiten, muß durch ein Protokoll sichergestellt werden, dass zu jedem Zeitpunkt maximal ein Datensender aktiv ist. Alle anderen Sender müssen sich zu dieser Zeit in hochohmigem Zustand befinden. Die Aktivierung der Senderbausteine kann, durch Schalten einer Handshake-Leitung oder Datenfluss gesteuert, automatisch erfolgen. Eine Terminierung des Kabels ist bei RS422-Leitungen nur bei hohen Baudraten und großen Kabellängen, bei RS485-Verbindungen dagegen grundsätzlich nötig. Obwohl für große Entfernungen bestimmt, zwischen denen Potenzialverschiebungen unvermeidbar sind, schreibt die Norm für keine der beiden Schnittstellen eine galvanische Trennung vor.

Da die Empfängerbausteine empfindlich auf Verschiebung des Massepotenzials reagieren, ist für zuverlässige Installationen eine galvanische Trennung unbedingt empfehlenswert. Bei der Installation muss auf korrekte Polung der Aderpaare geachtet werden, da eine falsche Polung zur Invertierung der Daten- und Handshake-Signale führt.

RS422- und RS485-Schnittstellen sind für die serielle Hochgeschwindigkeits-Datenübertragung über große Entfernungen entwickelt worden und finden im industriellen Bereich zunehmend Verbreitung. Die seriellen Daten werden ohne Massebezug als Spannungsdifferenz zwischen zwei korrespondierenden Leitungen übertragen. Für jedes zu übertragende Signal existiert ein Aderpaar, das aus einer invertierten und einer nicht invertierten Signalleitung besteht. Die invertierte Leitung wird in der Regel durch den Index „A“ oder „-“ gekennzeichnet, während die nicht invertierte Leitung mit „B“ oder „+“ bezeichnet wird.

Der Empfänger wertet lediglich die Differenz zwischen beiden Leitungen aus, sodass Gleichtakt-Störungen auf der Übertragungsleitung nicht zu einer Verfälschung des Nutzsymbols führen. Durch die Verwendung von abgeschirmtem, paarig verdrehtem Level-5-Kabel lassen sich Datenübertragungen über Distanzen von bis zu 1200 Metern bei einer Geschwindigkeit von bis zu 100.000 Baud realisieren. RS422-Sender stellen unter Last Ausgangspegel von  $\pm 2$  Volt zwischen den beiden Ausgängen zur Verfügung; die Empfängerbausteine erkennen Pegel von  $\pm 200$  mV noch als gültiges Signal.

**USB Serielle Schnittstelle** USB erlaubt es einem Gerät, Daten mit 1,5 Mbit/s, 12 Mbit/s oder 480 Mbit/s zu übertragen. Der USB-3.0-Standard ergänzt einen SuperSpeed-Modus mit 4000 Mbit/s. Diese Raten basieren auf dem Systemtakt der jeweiligen USB-Geschwindigkeit und stellen die physikalische Datenübertragungsrate dar. Der tatsächliche Datendurchsatz liegt infolge des Protokoll-Overhead darunter.

Im USB-Standard ist eine maximale theoretische Datenlast bei High-Speed unter idealen Bedingungen von 49.152.000 Byte/s beziehungsweise 53.248.000 Byte/s angegeben. Dazu kommt die Verwaltung der Geräte, sodass bei aktuellen Systemen für USB 2.0 eine nutzbare Datenrate in der Größenordnung von 320 Mbit/s (40 MB/s) und für USB 3.0 2400 Mbit/s (300 MB/s) zur Verfügung steht.

**CAN-Bus** Da bei einer herkömmlichen Verkabelung für jede Information je eine Leitung benötigt wird, steigt mit zunehmendem Funktionsumfang z. B. in der Kfz-Elektronik die Länge und das Gewicht des Kabelbaumes sowie die Zahl der Anschlüsse an den Steuergeräten. Abhilfe schafft hier der CAN-Bus, der sämtliche Informationen über lediglich zwei Leitungen überträgt. Datenbusse, auch CAN (Controlled Area Network) genannt, verbinden bis zu 100 verschiedene Steuergeräte miteinander.

- CAN Bus Klassen

Aktuelle Kraftfahrzeuge vernetzen bereits eine große Zahl von Steuergeräten miteinander, die unterschiedliche Anforderungen mit sich bringen. Daher werden im Kfz mehrere CAN-Bussysteme verbaut. Diese unterscheiden sich vor allem in der Übertragungsgeschwindigkeit. Sie werden in die drei Klassen eingeteilt.

- CAN A < 10 kBit/s Diagnose (konventionell)

Kfz mit CAN-Bus verfügen über ein Diagnosesystem. Solche Systeme lesen Fehlerspeicher aus und ermöglichen eine Stellglieddiagnose. Die Datenübertragungsgeschwindigkeit

keit ist nicht so wichtig, da die Daten nur gelegentlich in der Werkstatt zu Wartungs- und Diagnosezwecken ausgelesen werden. Der Diagnoseanschluss (auch K-Leitung und L-Leitung genannt) muss aber robust und fehlertolerant sein. Bei neueren Fahrzeugen wird die Diagnose direkt an der eigentlichen Busleitung (CAN C) durchgeführt.

- CAN B < 125 kBit/s Komfort, Display, Karosserie  
Über diesen (Low-Speed-CAN) Bus kommunizieren z. B. Steuergeräte für Beleuchtung, Klimaanlage, Verriegelung und Armaturen. Hier ist eine Übertragung wichtiger Daten bei nicht so hoher Geschwindigkeit wichtig (z. B. K-CAN, Karosserie-CAN, Komfort-CAN). Der Bus muss trotzdem ausfallsicher und robust sein. Daher arbeitet er im Kfz meist nach dem fehlertoleranten Standard ISO 11899-3.
- CAN C < 1 MBit/s Motor, Getriebe, Diagnose (Bus)  
An diesem (High-Speed-CAN) Bus sind z. B. die Steuergeräte für Motormanagement, Getriebe, ESP, ASR und ABS angeschlossen. Der Bus muss echtzeitfähig sein, d. h. die Datenübertragung darf sich durch den Bus nur extrem kurz verzögern. Mittlerweile ist auch eine Echtzeit-Diagnose über einen eigenen Diagnose-Bus möglich. Dieser Bus muss schnell sein, weil große Datenmengen in kurzer Zeit übertragen werden müssen. Im Kfz kommt meist der Standard ISO 11898-2 zum Einsatz.

**Profibus** Profibus (Process Field Bus) ist der universelle Feldbus, der breite Anwendung in der Fertigungs-, Prozess-, und Gebäudeautomatisierung findet. Profibus wurde durch Siemens und die Profibus-Nutzerorganisation entwickelt und in der internationalen Normenreihe IEC 61158 standardisiert. Profibus ermöglicht die Kommunikation von Geräten verschiedener Hersteller ohne besondere Schnittstellenanpassungen [37].

Der Profibus legt die technischen Merkmale eines seriellen Feldbussystems fest, mit dem verteilte digitale Automatisierungsgeräte von der Feldebene bis zur Zellenebene miteinander vernetzt werden können. Profibus ist ein Multi-Master-System und ermöglicht dadurch den gemeinsamen Betrieb von mehreren Automatisierungs-, Engineering- oder Visualisierungssystemen mit den dezentralen Peripheriegeräten an einem Bus.

Der Profibus basiert auf anerkannten internationalen Standards. Die Protokollarchitektur orientiert sich am OSI (Open System Interconnection) Referenzmodell, entsprechend dem internationalen Standard ISO 7498. Dabei übernimmt jede Übertragungsschicht genau festgelegte Aufgaben. Die Schicht 1 (Physical Layer) definiert die Übertragungsphysik, Schicht 2 (Data Link Layer) das Buszugriffsprotokoll und Schicht 7 (Application Layer) die Anwendungsfunktionen.

Profibus ist sowohl für schnelle, zeitkritische Anwendungen als auch für komplexe Kommunikationsaufgaben geeignet. Nachfolgend werden die Grundlagen von Profibus und den technischen Weiterentwicklungen DPV1 und DPV2 dargestellt.

Profibus unterscheidet folgende Gerätetypen:

- Master-Geräte: Sie bestimmen den Datenverkehr auf dem Bus. Ein Master darf Nachrichten ohne externe Aufforderung aussenden, wenn er im Besitz der Buszugriffsberechtigung (Token) ist. Master werden auch als aktive Teilnehmer bezeichnet.



- **Slave-Geräte:** Sie sind Peripheriegeräte wie beispielsweise Ein-/Ausgangsgeräte, Ventile, Antriebe und Messumformer. Sie erhalten keine Buszugriffsberechtigung, d. h. sie dürfen nur empfangene Nachrichten quittieren oder auf Anfrage eines Masters Nachrichten an diesen übermitteln. Slaves werden als passive Teilnehmer bezeichnet. Sie benötigen nur einen geringen Anteil des Busprotokolls, dadurch wird eine aufwandsarme Implementierung ermöglicht.

**SSI-Schnittstelle** Beim Interfacemodul mit SSI-Schnittstelle werden die Daten zwischen dem Lesekopf und dem Interfacemodul seriell mit der RS485-Schnittstelle und vom Interfacemodul zur Steuerung mit SSI-Protokoll (Serial Synchron Interface) übertragen. Die Daten werden im Binär-Code (WCS-IS310/320) oder im Gray-Code zur Steuerung übertragen.

An das Interfacemodul wird ein Lesekopf vom Typ LS211-0, an das WCS-IS320/321 ein Lesekopf vom Typ LS221-0 angeschlossen. Das Interfacemodul wird immer mit RS485-Abschlusswiderstand geliefert.

Weitere Hinweise zur Schnittstellenkonfiguration sind in [38] angegeben.

## 11.2 Laborversuch Inverter

In Abschn. 2.2.1 wurde eine Inverterschaltung mit Operationsverstärker gezeigt. Mit einem OP des Typs  $\mu A741$  soll dieser Laborversuch durchgeführt werden (Abb. 11.1).

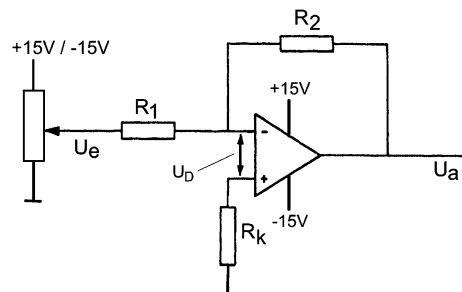
Bei der Messung der Ein- und Ausgangsspannung der Schaltung geht es darum, die Gl. 2.14 des OPs

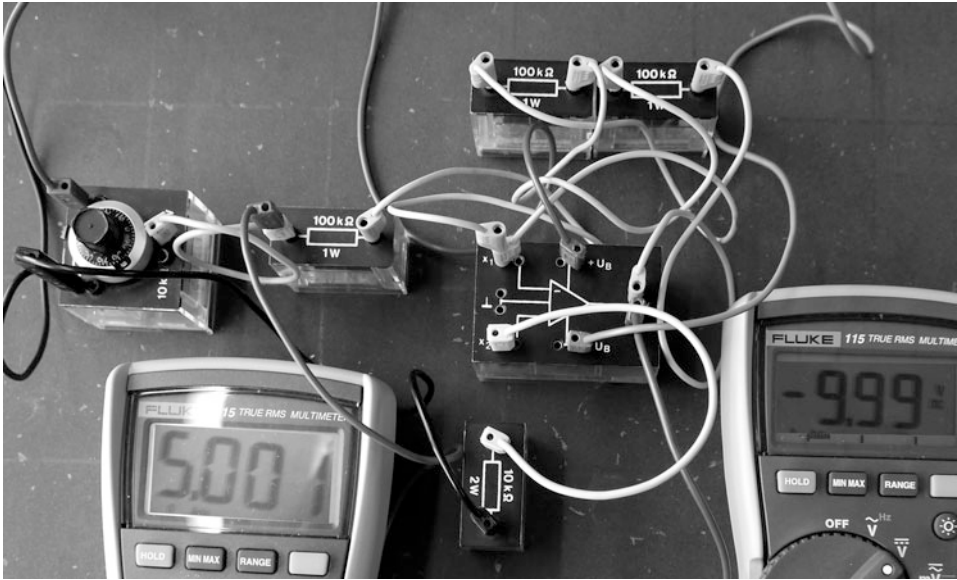
$$U_a = -U_e \frac{R_2}{R_1}$$

zu bestätigen bzw. deren Grenze zu erkennen. Zunächst wird der Offsetabgleich am OP vorgenommen. Tab. 11.1 liefert dann folgende Ergebnisse der Messungen.

Die Messungen bestätigen die Funktion der Gleichung des Inverters unterhalb der beiden Stellgrenzen  $+13,xx$  V und  $-12,yy$  V und bei einem Fehler, der maximal im 10 mV-Bereich liegt.

**Abb. 11.1** Inverter





**Abb. 11.2** Aufbau der Inverterschaltung und Messung für  $K_p = 2$

Der Widerstand  $R_k$  ist für den dargestellten Messbereich, wie sich zeigt, nicht relevant. Der Plus-Eingang des OPs kann daher direkt auf Masse gelegt werden.

Die Verstärkung wurde zunächst  $K_p = R_2/R_1 = 1$  und dann auf  $K_p = 2$  gesetzt. Die Messungen stimmen mit Gl. 2.14 überein (Abb. 11.2).

Die Differenzeingangsspannung  $U_D$  wird in der Analogtechnik häufig zur Berechnung von OP-Schaltungen herangezogen. Dabei setzt man bei einer die entsprechende OP-Gleichung erfüllenden Schaltung  $U_D = 0$  voraus. Wie die Messung zeigt, ist diese Bedingung erfüllt ( $U_D = 0,001$  mV gemessen).

Insgesamt gelten somit die in Abschn. 2.2.1 bereits zusammengestellten Bedingungen für den Betrieb eines Inverters bzw. allgemein für OP-Schaltungen.

### 11.3 Laborversuch Bandpass

Im Abschn. 2.2 wurde der aktive Bandpass bereits besprochen (Abb. 11.3).

Die Gl. 2.64a–d gibt den zugehörigen Frequenzgangbetrag wieder.

$$\frac{|F(j\omega)|}{\text{dB}} = 20 \lg \left( K_p \cdot \frac{\omega T_1}{\sqrt{(1 - \omega^2 T_1 T_2)^2 + \omega^2 (T_1 + T_2)^2}} \right).$$

Setzt man die Eckfrequenz  $\omega_1$  des Hochpasses und die des Tiefpasses  $\omega_2$  gleich, liegt das Maximum der asympthotischen Amplitude bei  $\omega_g$ . Die Darstellung der Kurven zeigt

Praktische Elektronik

Analogtechnik und Digitaltechnik für die industrielle  
Praxis

Orlowski, P.F.

2013, X, 314 S. 262 Abb. Mit Online-Extras., Softcover

ISBN: 978-3-642-39004-3