

III Ablaufsteuerungen und Zustandsgraph

11 Ablauf-Funktionsplan

11.1 Konzeption und Normungsquellen

Der Ablauf-Funktionsplan ist eine eigenständige Planart zur prozessorientierten Darstellung von Steuerungsaufgaben. Eine verbale Aufgabenstellung soll aus Gründen der Klarheit und Vollständigkeit durch eine grafische Darstellung ersetzt werden, die bei der Planung, Inbetriebnahme und Störungssuche hilfreich sein soll. Ein richtig entworfener Ablauf-Funktionsplan muss bereits die Lösung einer entsprechenden Steuerungsaufgabe darstellen.

Geeignete Steuerungsaufgaben sind solche, bei denen unterscheidbare Aktionen in einer ereignis- oder zeitgesteuerten Reihen- oder auch Parallelfolge ablaufen und die auf Wiederholung gerichtet sind. Anschauliche Beispiele dafür sind Produktionsanlagen oder Verkehrssysteme.

Der Ablauf-Funktionsplan stellt nur die grafischen Elemente für eine Ablaufbeschreibung zur Verfügung. Die Entwurfsmethode besteht darin, dass Steuerungszustände eingeführt und mit Aktionen verknüpft sowie Übergangsstellen zur Berücksichtigung von Steuersignalen vorgesehen werden.

Das Auffinden zutreffender Ablaufstrukturen ist das Hauptproblem dieser Entwurfsmethode und bedarf der entsprechenden Übung an Beispielen. Wie man sich leicht vorstellen kann, liegt schon bei einem einfachen Getränkeautomaten mit mehreren Bedientasten, Geldeinwurf mit verschiedenen Münzen, Geldrückgabe, Getränkeausgabe in Bechern, Mischungen durch Zusatz von Milch und Zucker etc. eine recht unübersichtliche Aufgabenstellung vor. Die Wirksamkeit des Ablauf-Funktionsplans besteht darin, dass er eine das Denken unterstützende anschauliche Darstellungsmethode anbietet.

Die Darstellung von Ablauf-Funktionsplänen kann auf zwei verschiedenen Normen beruhen:

- DIN EN 60848 GRAFCET, Spezifikationssprache für Funktionspläne der Ablaufsteuerung. (Nachfolger der DIN 40719-6). Diese Norm definiert eine grafische Entwurfssprache für die funktionale Beschreibung des Verhaltens des Ablaufteils eines Steuerungssystems. Die Entwurfssprache wird „GRAFCET“ genannt.
- DIN EN 61131, Speicherprogrammierbare Steuerungen, Teil 3: Programmiersprachen, hierin Elemente der Ablaufsprache (AS). Der Zweck der Ablaufsprache ist die Darstellung von Ablauffunktionen in SPS-Programm-Organisationseinheiten des Typs Funktionsbaustein oder Programm. Dazu gibt die Norm zwei Darstellungsvarianten für ihre Elemente an, eine ausführlich behandelte grafische Variante und eine textuelle Variante.

DIN EN 60848 und DIN EN 61131-3 haben jeweils ihren eigenen spezifischen Anwendungsbereich. Während die Entwurfssprache GRAFCET für die Beschreibung des Verhaltens unabhängig von einer speziellen Realisierung (elektronisch, elektromechanisch, pneumatisch oder gemischt) ist, legt IEC 61131-3 die Beschreibungsmittel der Ablaufsprache AS zwecks Pro-

grammrealisierung fest. Beide Darstellungen können jedoch auch über Ihren eigentlichen Anwendungsbereich hinaus verwendet werden. So kann GRAFCET auch für die Beschreibung einer Ablaufsteuerung mit spezifizierten Aktoren verwendet werden und die IEC 1131-3-Darstellung auch für die allgemeine Beschreibung einer Ablaufsteuerung benutzt werden. Der wesentliche Unterschied der beiden Normen besteht in der Darstellung der Aktionsblöcke. In Kapitel 11.2.4 sind die Unterschiede in einer Tabelle aufgeführt.

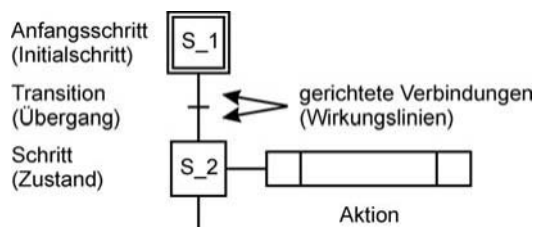
Nachfolgend wird für die grafische Darstellung der Ablauf-Funktionspläne die DIN EN 61131 zu Grunde gelegt, da diese nicht nur eine große Übereinstimmung mit den Programmierertools wie S7-GRAPH bei SIEMENS bzw. AS-Sprache bei CoDeSys hat, sondern auch bei der Programmumsetzung in AWL oder FUP sehr hilfreich ist.

11.2 Grafische Darstellung von Ablaufsteuerungsfunktionen

Zur Darstellung von Ablaufstrukturen werden Ablauf-Funktionspläne (Sequential Function Chart) verwendet. Wegen der geforderten allgemeinen Verständlichkeit ist eine einheitliche, normgerechte Darstellung der wichtigsten grafischen Elemente erforderlich, die der DIN EN 61131-3 entnommen sind, ergänzt mit einigen besseren Darstellungsformen aus S7-GRAPH.

11.2.1 Darstellung von Schritten

Jeder mögliche Zustand einer Steuerung wird im Ablauf-Funktionsplan durch einen *Schritt* dargestellt. Ein Schritt ist entweder aktiv oder inaktiv und stellt einen (Beharrungs-)Zustand der Steuerung dar. Ein Schritt muss grafisch durch einen Block dargestellt werden, der einen Schrittnamen in Form eines Bezeichners enthält.



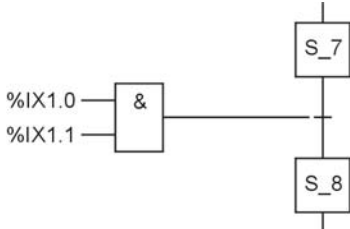
Eine Sonderstellung nimmt der Anfangsschritt ein, der durch eine doppelte Umrahmung gekennzeichnet ist. Der Anfangsschritt muss beim Start des Ablaufs als einziger Schritt aktiv sein. Die Schritte sind durch *gerichtete Verbindungen* in Form von vertikalen Linien miteinander verbunden. Die Eigenschaft eines Schrittes, aktiv oder inaktiv sein zu können, setzt ein Speicherverhalten im Steuerungsprogramm voraus. Jedem Schritt ist normalerweise eine Aktion zugeordnet, die in einem Aktionsblock angegeben und mit dem Schritt verknüpft werden kann. Die Aktion ist nicht Bestandteil des Schrittes.

11.2.2 Darstellung von Übergängen und Übergangsbedingungen

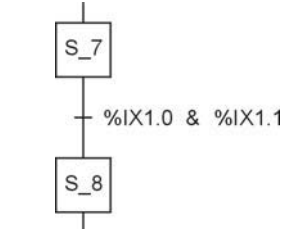
Zwischen den Schritten befindet sich immer ein Übergang. Jeder Übergang (Transition) muss eine Übergangsbedingung haben, die als Ergebnis eine logische „1“ (TRUE) oder eine logische „0“ (FALSE) liefert. Die Übergangsbedingung ist dafür verantwortlich, dass der aktive Zu-

stand von einem Schritt zum Nachfolger wechselt. Die Transition muss durch einen horizontalen Strich in der Verbindungslinie zwischen den Schritten dargestellt werden.

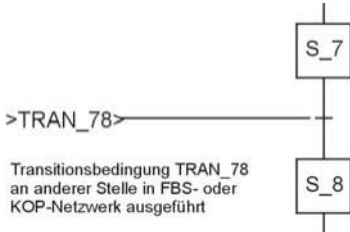
Transitionsbedingung in FBS-(KOP-)Sprache



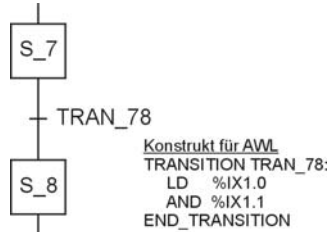
Transitionsbedingung in ST-Sprache



Transitionsbedingung über Konnektor



Transitionsbedingung über Transitionsnamen



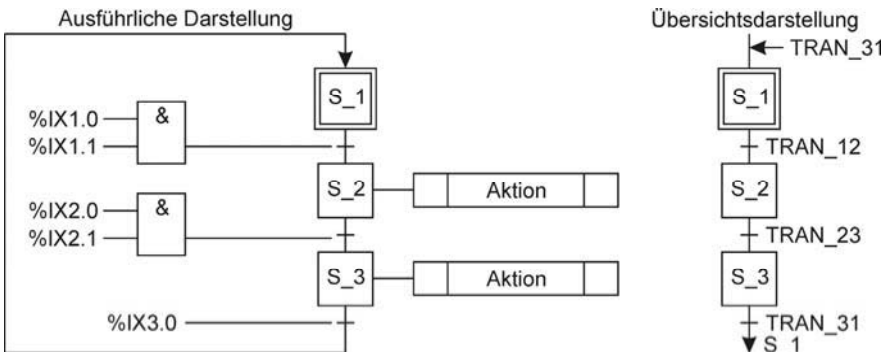
Die Übergangsbedingungen können mit den Darstellungsmitteln des Kontaktplans (KOP), des Funktionsplans (FBS bzw. FUP) und des Strukturierten Textes (ST) angegeben werden. Ein Folgeschritt wird gesetzt, wenn der Vorgängersschritt gesetzt und die Übergangsbedingung erfüllt ist (UND-Bedingung), dabei wird der Vorgängersschritt zurückgesetzt.

Die Norm DIN EN 61131-3 sieht auch die Möglichkeit vor, für einen Übergang einen Transitionsnamen zu vergeben, der auf ein Konstrukt zur Angabe der Transitionsbedingung verweisen muss. Von dieser Möglichkeit soll hier im weiteren aber kein Gebrauch gemacht werden, da mit dem Ablauf-Funktionsplan eine rein grafische Beschreibung einer Steuerungsaufgabe beabsichtigt ist.

11.2.3 Grundformen der Ablaufkette

Eine Ablaufkette besteht aus einer Folge von Schritten und Transitionen, die einfach oder verzweigt sein kann.

Einfache Ablaufkette:



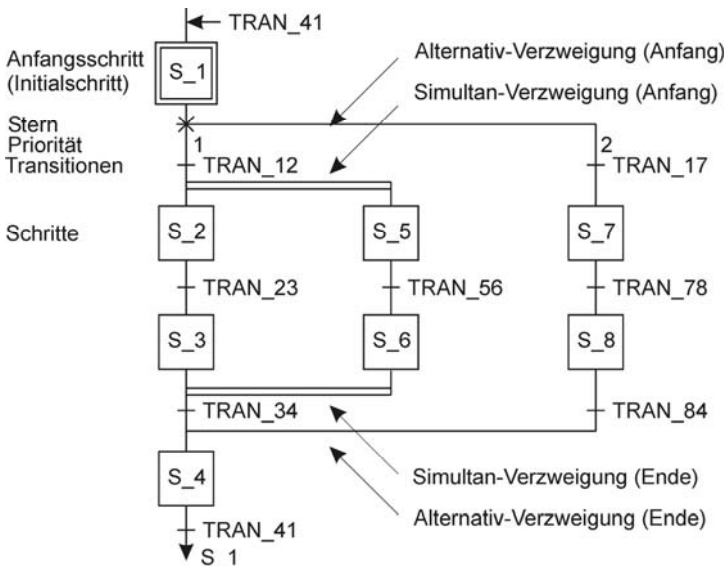
Der Wechsel von Schritt und Transition wird als Folge wiederholt. Es wird eine Kettenschleife gebildet, um wieder zum Anfang zurückzukehren. Die zurückführende Wirkungslinie kann vermieden und durch eine Pfeildarstellung mit Angabe von Schrittnummern ersetzt werden. Des Weiteren kann auch eine Übersichtsdarstellung der Ablaufkette ohne Transitionsbedingungen und Aktionen verwendet werden, wenn es nur um die Ansicht der Ablaufstruktur geht. Das voranstehende Bild zeigt ein Beispiel, bei der in der Übersichtsdarstellung die Pfeilkennzeichnung des Ablaufs aus S7-GRAPH übernommen wurde.

Verzweigte Ablaufkette:

Man unterscheidet je nach Art der Verzweigung zwischen der Alternativ-Verzweigung und der Simultan-Verzweigung.

- *Alternativ-Verzweigung* (1-aus-n-Verzweigung, ODER-Verzweigung); es erfolgt die Auswahl und Bearbeitung nur eines Kettenstranges aus mehreren Kettensträngen. Am Verzweigungsanfang darf zur gleichen Zeit nur eine Transitionsbedingung wahr sein (Verriegelung) oder es muss eine Priorität vorgegeben werden, indem der Strang mit der niedersten Nummer die höchste Priorität hat. Zusätzlich wird mit einem Stern (✱) angegeben, dass die Transitionen von links nach rechts bearbeitet werden. Jedes Strangende muss eine eigene Transitionsbedingung zum Verlassen des Kettenstranges aufweisen. Anfang und Ende von Alternativ-Verzweigungen werden durch waagerechte Einfachlinien dargestellt.
- *Simultan-Verzweigung* (Parallelbearbeitung mehrerer Kettenstränge, UND-Verzweigung); es erfolgt die gleichzeitige Aktivierung der Anfangsschritte mehrerer Kettenstränge, die dann aber unabhängig voneinander bearbeitet werden. Alle Kettenstränge unterliegen auf der Anfangsseite nur einer vorgelagerten gemeinsamen Transitionsbedingung. Bei der Zusammenführung der Kettenstränge (Endeseite) darf nur eine gemeinsame Transitionsbedingung vorhanden sein. Anfang und Ende von Simultan-Verzweigungen werden durch waagerechte Doppellinien dargestellt.

Das nachfolgende Bild zeigt eine Ablaufkette mit Verzweigungen und kurzen Erläuterungen zum Ablaufvorgang.



Die Anwendung der gegebenen Regeln kann nicht verhindern, dass fehlerhafte Ablaufketten konstruiert werden, die irgendwo hängen bleiben oder unerreichbare Schritte aufweisen.

Die Logik solcher Ablauf-Funktionspläne muss daher sehr sorgfältig durchdacht werden.

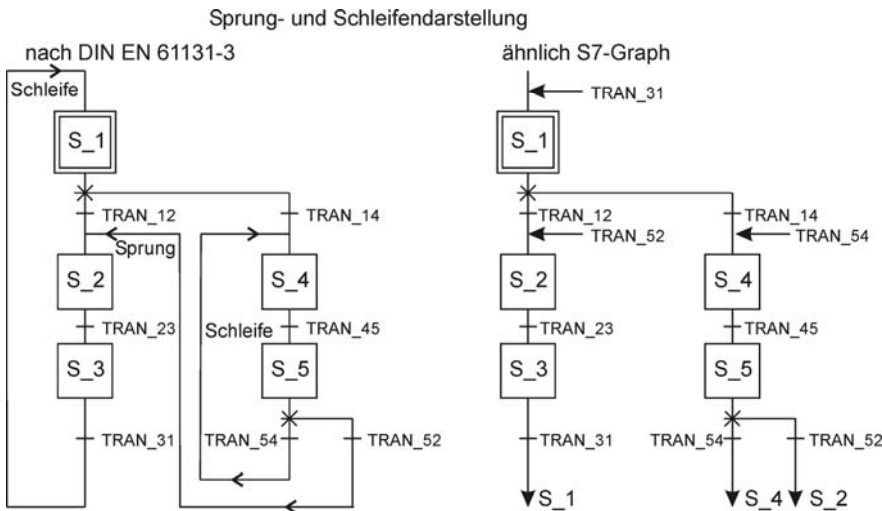
Wenn zum Beispiel der Anfangsschritt S_1 aktiv ist und die Transitionsbedingung von TRAN_17 den booleschen Wert TRUE liefert, während die Transitionsbedingung von TRAN_12 den booleschen Wert FALSE hat, wird der Schritt S_7 aktiviert und Schritt 1 deaktiviert. Damit kann es in dieser Phase des Ablaufs nicht mehr zu einer Aktivierung der Schritte S_2 und S_5 kommen. Der Übergang von Schritt S_8 zu Schritt S_4 erfolgt erst, wenn die Transitionsbedingung von TRAN_84 den booleschen Wert TRUE annimmt. Schritt S_8 wird dann von S_4 zurückgesetzt.

Für den Fall, dass der Anfangsschritt S_1 aktiv ist und die Transitionsbedingungen von TRAN_12 und TRAN_17 den booleschen Wert TRUE haben, wird wegen der festgelegten Priorität die Simultan-Verzweigung bearbeitet. Es werden die Schritte S_2 und S_5 aktiviert und der Vorgängerschritt S_1 deaktiviert. Der Schritt S_4 wird erst erreicht, wenn die Vorgängerschritte S_3 und S_6 aktiv sind und die Transitionsbedingung von TRAN_34 den booleschen Wert TRUE annimmt. Die Schritte S_3 und S_6 werden dann von S_4 zurückgesetzt.

Schleifen und Sprünge

Ein *Sprung* führt unter Steuerung durch eine Transitionsbedingung von einem Schritt zu einem entfernten anderen Schritt, wobei der durch den Sprung gebildete Zweig keine Schritte enthält.

Eine *Schleife* kann die Folge eines Sprunges sein, indem unter Steuerung durch eine Transitionsbedingung auf einen Vorgängerschritt zurückgesprungen wird. Das nachfolgende Bild zeigt zwei Schleifen, wobei die Schleife mit den Schritten 4 und 5 einen Sonderfall darstellt: Bei entsprechenden Transitionsbedingungen können diese beiden Schritte ständig durchlaufen werden. Dabei bereitet Schritt 4 das Setzen von Schritt 5 vor und setzt diesen als Folgezustand auch wieder zurück. Ebenso ist Schritt 5 Vorbereiter von Schritt 4 und auch dessen Rücksetzer. Dieser Sachverhalt spielt beim Entwurf eines Ablauf-Funktionsplanes noch keine Rolle. Erst bei der Umsetzung in ein Steuerungsprogramm unter Verwendung von RS-Speicherglieder darf es am Setz- und Rücksetzeingang nicht zu einer sich aufhebenden Wirkung kommen.



Nachfolgend wird die aufgelöste Darstellungsform von Ablaufketten bevorzugt, um sich kreuzende Wirkungslinien zu vermeiden, die in ungünstigen Fällen auftreten können. Ablaufketten mit einer großen Anzahl von Schritten und mehreren Verzweigungen oder Sprüngen verlieren

an Übersichtlichkeit. Es ist zu prüfen, ob es nicht besser ist, den gesamten Ablauf in einzelne technologische Funktionen zu zerlegen und diese mit mehreren korrespondierenden einfachen Ablaufketten umzusetzen (siehe 11.6.2).

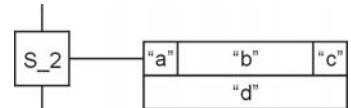
11.2.4 Aktionen, Aktionsblock

Mit einem Schritt ist in der Regel eine Aktion verbunden. Ein Schritt ohne zugehörige Aktion übt eine Warte-Funktion aus bis die nachfolgende Transitionsbedingung erfüllt ist. Da der Ablauf-Funktionsplan beschreiben soll, was in der gesteuerten Anlage zu geschehen hat, wird in der Norm der Begriff *Aktion* anstelle von Befehl verwendet.

Der *Aktionsblock* ist ein grafisches Element zur Darstellung von Aktionen. Der Aktionsblock ist nicht Teil eines Schrittes und damit auch nicht Teil der Ablaufkette. Der Aktionsblock kann mit einem Schritt verknüpft oder als grafisches Element in einer Kontaktplan- bzw. Funktionsplandarstellung verwendet werden. Nachfolgend wird jedoch die Verknüpfung von Schritt und Aktionsblock bevorzugt, wie im Bild dargestellt.

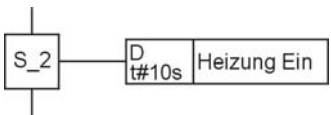
In vollständiger Darstellung besteht der Aktionsblock aus vier Teilflächen, die nicht alle genutzt werden müssen:

- Feld „a“: Bestimmungszeichen
- Feld „b“: Aktionsname
- Feld „c“: AnzeigevARIABLE
- Feld „d“: Beschreibung der Aktion in AWL; ST; KOP, FBS



Im einfachsten Fall wird in Feld „b“ der Name einer booleschen Variablen und in Feld „a“ ein zutreffendes Bestimmungszeichen eingetragen.

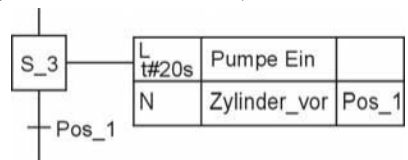
Die Aktion wird ausgeführt, wenn der zugehörige Schritt gesetzt ist und eine Aktionssteuerung die Freigabe erteilt. Die Aktionssteuerung sorgt für die richtige Umsetzung der im Feld „a“ eingetragenen Bestimmungszeichen.



In der Norm DIN EN 6111-3 ist sogar ein entsprechender Funktionsbaustein detailliert beschrieben, der diese umfangreiche Aufgabe wahrnehmen kann. Es wird jedoch nicht gefordert, dass dieser Funktionsbaustein tatsächlich verwendet werden muss; es genügt, dass die Aktion gleichwertig ausgeführt wird.

Im obigen Beispiel lautet die Aktion „Heizung EIN“ und es ist das Bestimmungszeichen D zusammen mit einer Zeitangabe im Feld „a“ eingetragen. Ergebnis: Die Heizung wird entsprechend verzögert eingeschaltet (siehe Tabelle: Bestimmungszeichen für Aktionen).

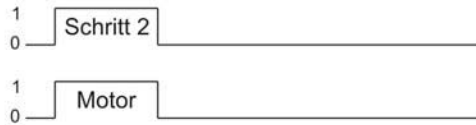
Es ist zulässig, mehrere Aktionen mit einem Schritt zu verbinden. Dies wird grafisch dargestellt durch aneinander gereihete Aktionsblöcke.



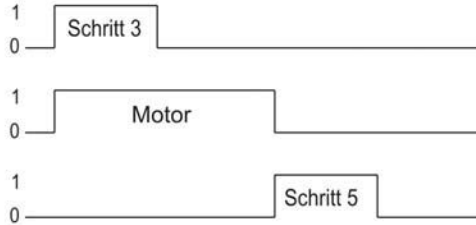
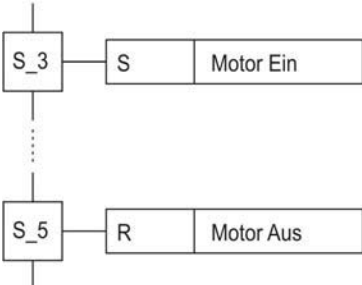
Im Aktionsfeld „c“ kann eine boolesche Anzeige-VARIABLE eingetragen werden, die durch die Aktion gesetzt werden kann, um die Erledigung oder einen Fehlerfall (z. B. Zeitüberschreitung) anzuzeigen. Die Anzeige-VARIABLE kann ggf. auch als Weiterschaltbedingung verwendet werden.

Tabelle 11.1: Bestimmungszeichen für Aktionen (Auswahl)

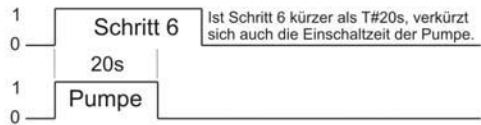
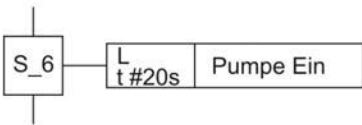
N-Befehl (nichtgespeichert)



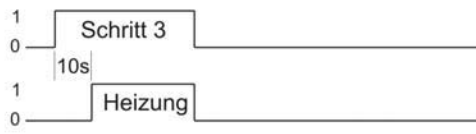
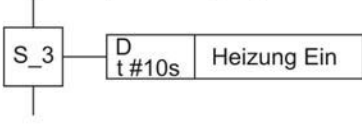
S-Befehl (Setzen, gespeichert)



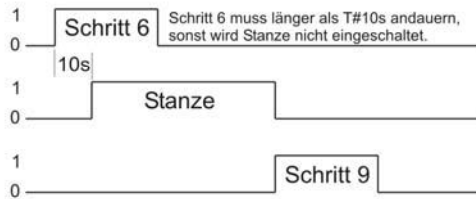
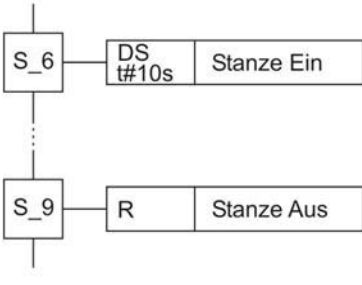
L-Befehl (zeitbegrenzt)



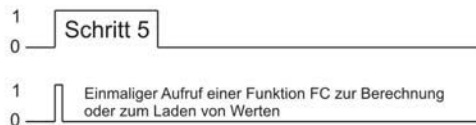
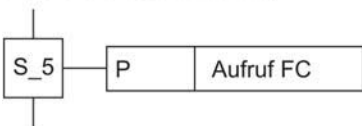
D-Befehl (zeitverzögert)



DS-Befehl (verzögert und gespeichert)



P-Befehl (Impuls Flanke)



Es fehlen: SD = gespeichert und zeitverzögert, SL = gespeichert und zeitbegrenzt.

Die **DIN EN 60848** unterteilt den Aktionsblock nicht. Die Art der Aktionsausgabe wird dort durch Symbole am Aktionsblock festgelegt. Prinzipiell werden dort zwei Arten zur Bildung von Ausgangswerten unterschieden.

Kontinuierlich wirkende Art (Zuweisung durch Zustand)

In der kontinuierlich wirkenden Art zeigt die Verknüpfung einer Aktion mit einem Schritt an, dass eine Ausgangsvariable den Wert TRUE hat, wenn der Schritt aktiv ist und die Zuweisungsbedingung erfüllt ist.

Gespeichert wirkende Art (Zuordnung durch Ereignis)

In der gespeichert wirkenden Art wird die Verknüpfung einer Aktion mit internen Ereignissen angewendet, um anzuzeigen, dass eine Ausgangsvariable den auferlegten Wert annimmt und behält, wenn eines dieser Ereignisse einsetzt.

Gegenüberstellung der unterschiedlichen Darstellung der wichtigsten Aktionen der Normen IEC 61131-3 und GRAFCET (EN 60848):

IEC 61131-3	EN 60848	Beschreibung
		Nicht gespeicherte Aktion Nicht gespeicherte kontinuierlich wirkende Aktion.
		Bedingte nicht gespeicherte Aktion Nicht gespeicherte kontinuierlich wirkende Aktion mit Zuweisungsbedingung.
		Zeitverzögerte Aktion Die Zuweisungsbedingung wird erst nach der Zeit t = 3 s ausgehend von der Aktivierung des Schritts erfüllt.
		Zeitbegrenzte Aktion Die Zuweisungsbedingung wird während der Dauer von t = 3 s ausgehend von der Aktivierung des Schritts erfüllt.
		Speichernde Aktion (Setzen) Der booleschen Variablen VAR1 wird der Wert 1 mit der Aktivierung des Schrittes zugeordnet.
		Speichernde Aktion (Rücksetzen) Der booleschen Variablen VAR1 wird der Wert 0 mit der Aktivierung des Schrittes zugeordnet.
		Speichernd zeitverzögerte Aktion Erst 3 Sekunden nach Aktivierung des Schrittes wird der booleschen Variable VAR2 eine „1“ speichernd zugewiesen.

11.3 Umsetzung des Ablauf-Funktionsplans mit SR-Speichern

Ein unter Berücksichtigung der technologischen Bedingungen richtig entworfener Ablauf-Funktionsplan stellt die Lösung einer entsprechenden Steuerungsaufgabe dar. Wie wird dann aus dem Entwurf ein ablauffähiges Steuerungsprogramm?

In der Praxis wird man ein Programmiersystem wählen, zu dessen Sprachenumfang auch die Ablaufsprache (AS) zählt, z. B. S7-GRAPH bei STEP 7 oder AS bei CoDeSys. Im Prinzip kann man dann den Entwurf direkt eingeben. Man braucht sich z. B. nicht um die Umsetzung der Ablaufkette und der Aktionsausgabe unter Berücksichtigung der Bestimmungszeichen zu kümmern, dies wird alles vom Programmier- und Betriebssystem der SPS übernommen.

Möglich ist jedoch auch eine Realisierung von Ablauf-Funktionsplänen mit grundlegenden Programmiermitteln wie SR-Speichergliedern und zugehörigen Grundverknüpfungen, wie anschließend dargestellt. Im Abschnitt 11.4 wird dann gezeigt, wie sich die Umsetzung des Ablauf-Funktionsplans durch einen wiederverwendbaren Baustein standardisieren lässt.

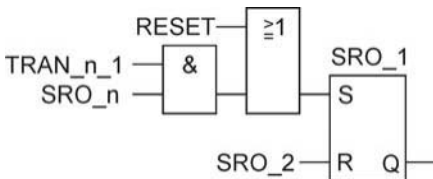
11.3.1 Umsetzungsregeln

Eine mit jedem Programmiersystem mögliche Umsetzungsmethode ist die Zuweisung eines Schrittes der Ablaufkette an einen SR-Speicher und die Realisierung von Übergangsbedingungen durch logische Grundverknüpfungen. Damit die Ablaufkette zu jedem Zeitpunkt in die Grundstellung (Initialschritt aktiv und alle andern Schritte inaktiv) gebracht werden kann, wird eine Variable RESET eingeführt.

Folgende Regeln sind zu beachten:

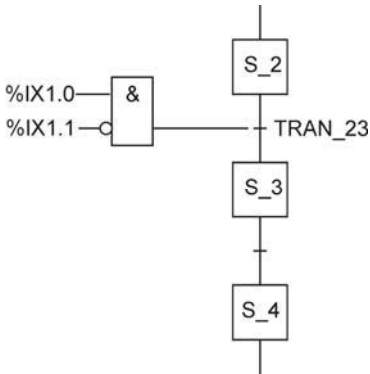
- Es werden so viele SR-Speicher verwendet, wie es Schritte im Ablauf-Funktionsplan gibt, einschließlich eines Anfangsschrittes.
- Beim Einschalten der Steuerung muss für die Grundstellung der Ablaufkette gesorgt werden. Es bieten sich zwei Möglichkeiten an:
 1. Eine Anweisungsfolge erzeugt einen einmalig auftretenden Richtimpuls beim Einschalten der Steuerung und führt zum Setzen des Anfangsschrittes sowie zum Rücksetzen der anderen Schrittspeicher:


```
UN #FO      // # = Vorsatzzeichen für lokale Variablen
= #IO      // Richtimpuls
S #FO      // IO = Impuls-Operand, FO = Flankenoperand
```
 2. Die statische Lokalvariable des 1. Schrittspeichers SRO_1 erhält in der Deklarationstabelle den Anfangswert TRUE, alle anderen Schrittspeicher erhalten den Anfangswert FALSE.
- Umsetzung des Initialschrittes S_1 in ein SR-Speicherglied. Es wird angenommen, dass die statische lokale Variable SRO_1 den Anfangswert „TRUE“ erhält.

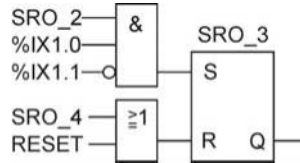


TRAN_n_1: Transitionsbedingung des letzten Schrittes n zu Schritt 1
 SRO_n: Schrittoperand des letzten Schrittes n
 SRO_2: Schrittoperand des 2. Schrittes

- Stellvertretend für alle weiteren Schritte sei die Umsetzung des Schrittes 3 mit einem SR-Speicherglied dargestellt:

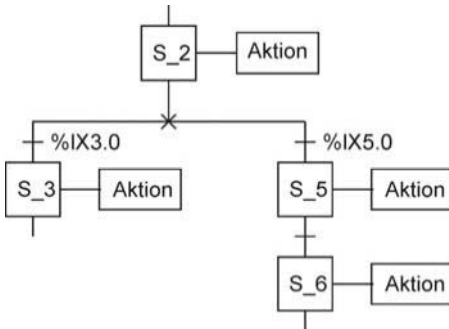


Setzbedingung: Am Setzeingang von Speicherglied 3 ist die Übergangsbedingung der Transition TRAN23 mit dem Ausgang von Speicherglied 2 UND-verknüpft.

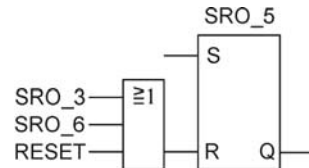


Rücksetzbedingung: Speicherglied 3 wird an seinem Rücksetzeingang vom Folgezustand angesteuert. Erst wenn Zustand 4 erreicht ist, wird Zustand 3 gelöscht.

- Verriegelung von Schritten, deren Übergangsbedingungen gleichzeitig auftreten können.



Verriegelung: Zustand 3 setzt sich gegen Zustand 5 durch, indem er diesen im Konfliktfall zurücksetzt.



- Die Realisierung der Befehlsausgabe erfolgt gesondert durch Zuweisung des Schrittoperanden SRO_x zum Steuerungsausgang der Aktion.
- Wird ein Steuerungsausgang von mehreren Schritten aus angesteuert, werden die entsprechenden Schrittoperanden „ODER“-verknüpft.



Mehrfache Einzelzuweisungen wären falsch, weil sich dabei immer nur die letzte Anweisung durchsetzt.

- Ist das Bestimmungszeichen der Aktion D (zeitverzögert) oder L (zeitbegrenzt), so erfolgt die Befehlsausgabe mit einem SE- oder mit einem SI-Zeitglied bzw. mit dem TON- oder TP-Funktionsblock.

Bestimmungszeichen D:



Bestimmungszeichen L:



11.3.2 Realisierung

Die Ablaufkette mit allen Schritten, Transitionen und Aktionen wird in einem Funktionsbaustein umgesetzt. Da die Transitionen und Schritte anlagenabhängig sind, kann der Baustein dann nur für diese Anlage verwendet werden. Sind mehrere gleichartige Anlagen anzusteuern, kann der Baustein jedoch mehrmals im Steuerungsprogramm aufgerufen werden.

Regeln für die Programmierung des anlagenabhängigen Funktionsbausteins:

- Die IN-Variablen des Funktionsbausteins werden durch die Sensoren der Anlage, den RESET-Eingang und eventuell benötigte Zeit- und Zähler-Variablen gebildet.
- Die OUT-Variablen des Funktionsbausteins steuern die Aktoren der Anlage an.
- Der Schrittoperand des Initialschrittes SRO_1 wird mit „TRUE“ vorbelegt.
- Die Grundstellung der Ablaufkette wird entweder durch den betriebsmäßigen Ablauf oder durch Ansteuerung mit einem RESET-Signal erreicht. RESET setzt den Speicher des Initialschrittes und rücktsetzt alle anderen Schrittspeicher.
- Die Schrittspeicher steuern die Ausgänge (Aktoren) an. Ist ein Ausgang von mehreren Schrittspeichern anzusteuern, so sind diese durch ODER zu verknüpfen.

11.3.3 Beispiel

■ Beispiel 11.1: Reaktionsprozess

In einem Reaktionsbehälter werden zwei unterschiedliche chemische Ausgangsstoffe zusammengeführt, auf eine bestimmte Temperatur erwärmt und durch Umrühren vermischt.

Technologieschema:

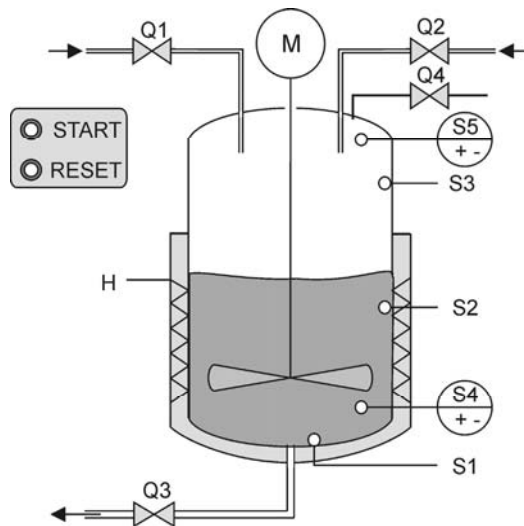


Bild 11.1: Reaktionsbehälter

Prozessablaufbeschreibung:

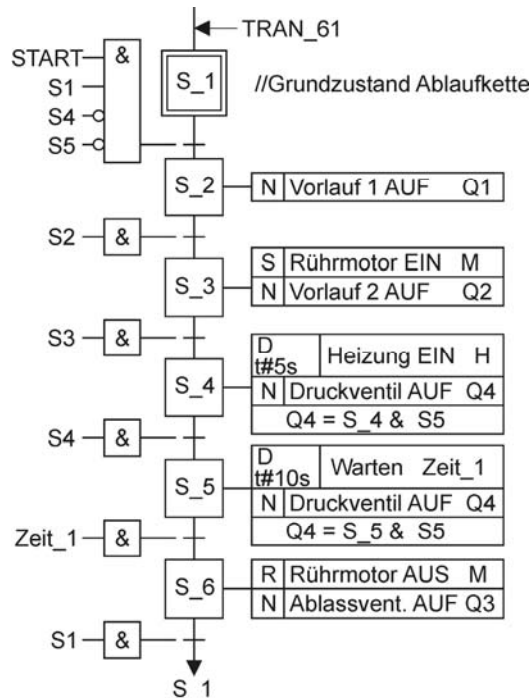
Nach Betätigung des Tasters START wird, sofern eine Leermeldung des Behälters durch S1 vorliegt und der Temperatursensor S4 sowie Drucksensor S5 keine Grenzwerte melden, das Vorlaufventil Q1 solange geöffnet, bis der Niveauschalter S2 („1“) anspricht. Danach schaltet der Rührwerkmotor M ein und das Ventil Q2 wird geöffnet. Spricht der Niveauschalter S3 („1“) an, schließt das Ventil Q2 wieder. Nach einer Wartezeit von 5 Sekunden wird die Heizung H eingeschaltet, bis der Temperatursensor S4 („1“) das

Erreichen der vorgegebenen Temperatur meldet. Meldet während des Aufheizvorgangs der Drucksensor S5 einen Überdruck im Reaktionsbehälter, wird das Druckablassventil Q4 geöffnet bis keine Meldung mehr vorliegt. Nach Ablauf des Aufheizvorganges läuft das Rührwerk noch 10 Sekunden weiter, bevor das Ventil Q3 öffnet. Ist der Behälter leer (Niveauschalter S1 = „1“), wird das Ventil Q3 wieder geschlossen und der Prozessablauf kann wiederholt werden. Mit der RESET-Taste lässt sich die Ablaufkette in die Grundstellung bringen.

Zuordnungstabelle der Eingänge und Ausgänge:

Eingangsvariable	Symbol	Datentyp	Logische Zuordnung	Adresse
Start-Taster	START	BOOL	Betätigt START = 1	E 0.0
Leermeldung	S1	BOOL	Behälter leer S1 = 1	E 0.1
Niveauschalter 1	S2	BOOL	Niveau 1 erreicht S2 = 1	E 0.2
Niveauschalter 2	S3	BOOL	Niveau 2 erreicht S3 = 1	E 0.3
Temperatursensor	S4	BOOL	Temp. erreicht S4 = 1	E 0.4
Drucksensor	S5	BOOL	Überdruck erreicht S5 = 1	E 0.5
Taster Ablaufk. Grundstellung	RESET	BOOL	Betätigt RESET = 1	E 0.7
Ausgangsvariable				
Zulaufventil 1	Q1	BOOL	Ventil auf Q1 = 1	A 4.1
Zulaufventil 2	Q2	BOOL	Ventil auf Q2 = 1	A 4.2
Ablassventil	Q3	BOOL	Ventil auf Q3 = 1	A 4.3
Druckablassventil	Q4	BOOL	Ventil auf Q4 = 1	A 4.4
Heizung	H	BOOL	Heizung an H = 1	A 4.5
Rührmotor	M	BOOL	Motor ein M = 1	A 4.6

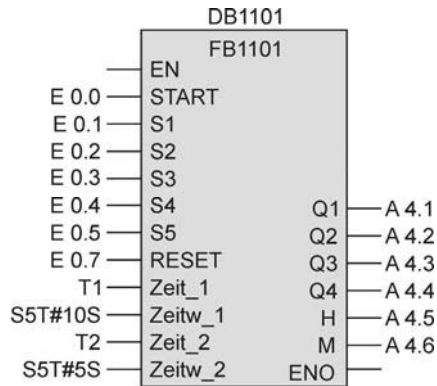
Ablauf-Funktionsplan:



Die Umsetzung des Ablauf-Funktionsplanes erfolgt im Funktionsbaustein FB 1101 nach den beschriebenen Regeln. Nachfolgend ist die Lösung für STEP 7 und CoDeSys gezeigt.

Lösung in STEP 7

**Aufruf des Funktionsbausteins
FB 1101 im OB 1:**



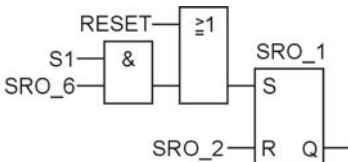
Deklarationstabelle des Funktionsbausteins FB 1101:

Name	Datentyp	Anfangswert
IN		
START	Bool	FALSE
S1	Bool	FALSE
S2	Bool	FALSE
S3	Bool	FALSE
S4	Bool	FALSE
S5	Bool	FALSE
RESET	Bool	FALSE
Zeit_1	Timer	
Zeitw_1	S5Time	S5T#10S
Zeit_2	Timer	
Zeitw_2	S5Time	S5T#5S

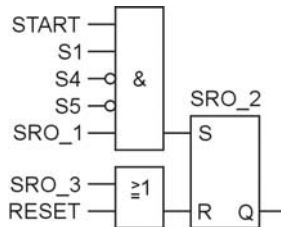
Name	Datentyp	Anfangswert
OUT		
Q1	Bool	FALSE
Q2	Bool	FALSE
Q3	Bool	FALSE
Q4	Bool	FALSE
H	Bool	FALSE
M	Bool	FALSE
STAT		
SRO_1	Bool	TRUE
SRO_2	Bool	FALSE
SRO_3	Bool	FALSE
SRO_4	Bool	FALSE
SRO_5	Bool	FALSE
SRO_6	Bool	FALSE

Funktionsplan:

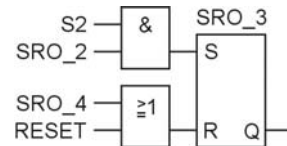
Schritt S_1



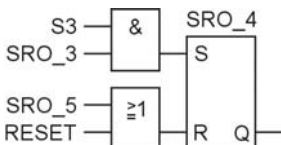
Schritt S_2



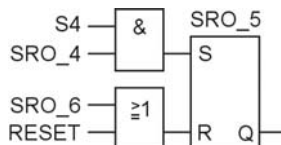
Schritt S_3



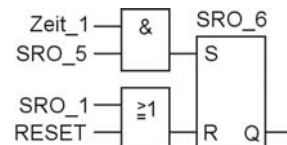
Schritt S_4



Schritt S_5



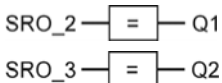
Schritt S_6



Zeitbildung Zeit_1



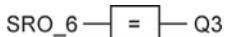
Ansteuerung Ventil Y1 u. Y2



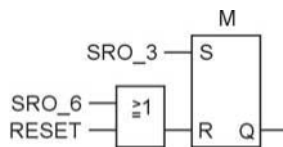
Ansteuerung Heizung H



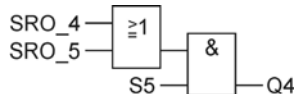
Ansteuerung Ventil Y3



Ansteuerung Motor M



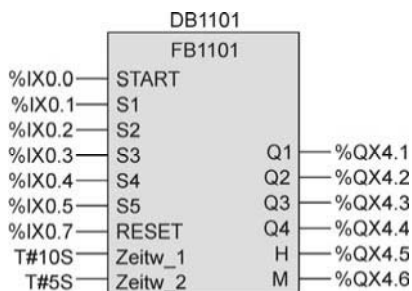
Ansteuerung Ventil Y4



Lösung in CoDeSys

Aufruf des Funktionsbausteins

FB 1101 im PLC_PRG:



Der Funktionsplan der STEP 7 Lösung kann bis auf die Verwendung der SE-Zeitfunktionen für CoDeSys übernommen werden. Statt der beiden SE-Zeitfunktionen werden dort zwei Funktionsblöcke TON aus der Standardbibliothek verwendet.

FB 1101 AWL:

```

FUNCTION_BLOCK FB1101
VAR_INPUT
    LD RESET
    START, S1, S2,
    S3, S4, S5,
    RESET:BOOL;
    ZEITW_1: TIME;
    ZEITW_2: TIME;
END_VAR
VAR_OUTPUT
    Q1, Q2, Q3,
    Q4, H, M:BOOL;
END_VAR
VAR
    ANDN S4
    ANDN S5
    SRO_1:BOOL:=TRUE;
    SRO_2, SRO_3,
    SRO_4, SRO_5,
    SRO_6:BOOL;
    Zeit1: TON;
    Zeit2: TON;
END_VAR
LD RESET
OR( S1
AND SRO_3
)
LD RO_5
OR RESET
R SRO_4
CAL Zeit1
(IN := SRO_5,
PT := ZEITW_1)
CAL Zeit2
(IN := SRO_4,
PT := ZEITW_2)
LD START
AND S1
AND SRO_3
LD SRO_4
S SRO_5
LD SRO_6
OR RESET
R SRO_5
LD Zeit1.Q
AND SRO_5
S SRO_6
LD SRO_1
OR RESET
LD SRO_2
ST Q1
LD SRO_3
ST Q2
LD SRO_6
ST Q3
LD SRO_4
OR SRO_5
AND S5
ST Q4
LD S2
AND SRO_2
S SRO_3
LD SRO_4
OR RESET
R SRO_3

```

11.4 Umsetzung des Ablauf-Funktionsplans mit standardisierter Baueinbaueinheit

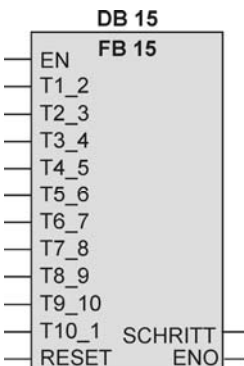
Für lineare Ablaufketten kann eine standardisierte Lösungsstruktur gefunden werden, bestehend aus einem allgemein gültigen Funktionsbaustein FB zur Umsetzung der Schritte mit den Transitionsbedingungen und einem anlagenspezifischen Ausgabebaustein. Der allgemein gültige Funktionsbaustein (Bibliotheks-Funktionsbaustein) enthält keine Befehlsausgabe, da diese anlagenabhängig ist. Ergebnis und Ausgabe dieses Bausteins ist nur die Schrittnummer, welche den jeweils aktuellen Schritt als Integer-Zahl ausgibt. Ein aufgabengemäß programmierter Befehlsausgabebaustein muss die aktuelle Schrittnummer auswerten und die Befehlsausgabe ausführen. Der vorgestellte Bibliotheks-Funktionsbaustein FB 15 ist für zehn Schritte konzipiert. Eine Erweiterung des Bausteins für mehr Schritte ist jedoch leicht ausführbar.

11.4.1 Regeln für die Programmierung des Bibliotheks-Schrittkettenbausteins FB 15: KoB (Kette ohne Betriebsartenwahl)

- Die IN-Variablen des Schrittkettenbausteins werden durch zehn boolesche Variablen T1_2, T2_3 bis T10_1 und einen RESET-Eingang gebildet. An den Eingang T1_2 wird die Transitionsbedingung von Schritt 1 nach Schritt 2 geschrieben. Für die restlichen Eingänge T2_3 bis T10_1 gilt Entsprechendes. Die IN-Variablen T2_3 bis T10_1 erhalten in der Vorbelegung der Deklarationstabelle den Wert „TRUE“. Damit wird automatisch weitergeschaltet, wenn ein solcher Eingang nicht beschaltet ist.
- Einzige OUT-Variable des Schrittkettenbausteins ist SCHRITT (Schrittnummer) mit dem Datenformat INTEGER. Dieser Ausgang gibt den jeweils aktuellen Schritt an.
- Für jeden Ablaufschritt ist ein SR-Speicherglied mit der Logik der Transitionsbedingungen zum Setzen des Nachfolgespeichers vorzusehen. Über den Rücksetzeingang wird der Vorgängerspeicher von seinem Nachfolgerspeicher gelöscht. Für die SR-Speicherglieder sind Schrittoperanden als statische Variablen in der Deklarationstabelle einzuführen. Der Schrittoperand des Initialschrittes SRO_1 wird dabei mit „TRUE“ vorbelegt.
- Die Grundstellung der Ablaufkette wird entweder durch den betriebsmäßigen Ablauf oder durch Ansteuerung mit einem RESET-Signal erreicht. RESET setzt den Speicher des Initialschrittes und rückt alle anderen Schrittspeicher.

Die Umsetzung des standardisierten Schrittkettenbausteins erfolgt im Funktionsbaustein FB 15.

Funktionsbaustein FB 15:



Beschreibung der Übergabeparameter:

T1_2	Eingangsvariable für die Transitionsbedingung von Schritt 1 nach 2
T2_3 ...	Eingangsvariablen der entsprechenden Transitionsbedingungen der Transitionen
T10_1	T2_3 bis T_10_1 aus der Schrittfolge
RESET	Eingangsvariable, um die Schrittfolge in die Grundstellung zu setzen
SCHRITT	INTEGER-Variable zur Ausgabe der aktuellen Schrittnummer

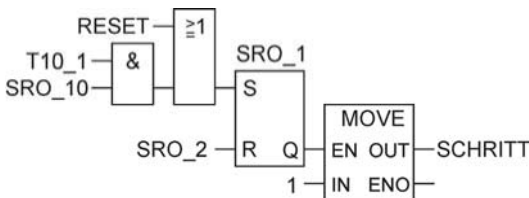
Deklarationstabelle des Schrittketten-Funktionsbausteins FB 15:

Name	Datentyp	Anfangswert
IN		
T1_2	Bool	FALSE
T2_3	Bool	TRUE
T3_4	Bool	TRUE
T4_5	Bool	TRUE
T5_6	Bool	TRUE
T6_7	Bool	TRUE
T7_8	Bool	TRUE
T9_10	Bool	TRUE
T10_1	Bool	TRUE
RESET	Bool	FALSE
OUT		
SCHRITT	Integer	0

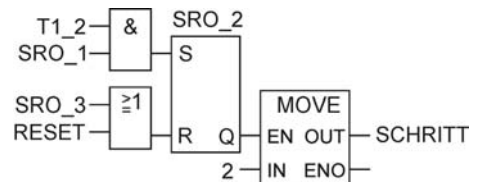
Name	Datentyp	Anfangswert
STAT		
SRO_1	Bool	TRUE
SRO_2	Bool	FALSE
SRO_3	Bool	FALSE
SRO_4	Bool	FALSE
SRO_5	Bool	FALSE
SRO_6	Bool	FALSE
SRO_7	Bool	FALSE
SRO_8	Bool	FALSE
SRO_9	Bool	FALSE
SRO_10	Bool	FALSE

Funktionsplan des Schrittketten-Funktionsbausteins FB 15 für STEP 7:

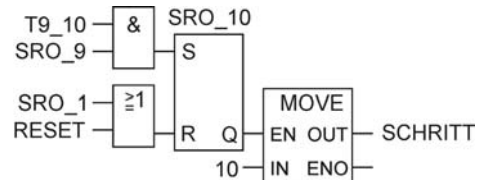
Schritt 1:



Schritt 2:



Schritt 10:



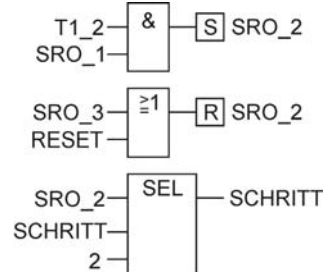
Die Programmteile für die Schritte 3 bis 9 sind entsprechend aufgebaut. Auf eine Darstellung wird deshalb verzichtet.

Funktionsplan des Schrittketten-Funktionsbausteins FB 15 für CoDeSys:

Da bei CoDeSys die MOVE-Box im Funktionsplan keinen EN-Eingang besitzt, wird zur Übergabe der Schrittnummer bei einem aktiven Schritt die SEL-Funktion verwendet.

Beispielhaft ist nebenstehend die Umsetzung des Schrittes 2 gezeigt.

Schritt 2:

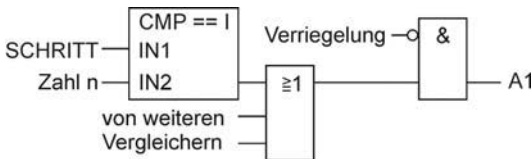


Wird die Ablaufkette mit dem Ablaufkettenbaustein FB 15 realisiert, müssen die Ausgabeaktionen außerhalb des FB 15 gebildet werden. Es fördert nicht nur die Übersichtlichkeit, wenn die Ausgabeaktionen in einem Befehlsausgabebaustein zusammengefasst werden, sondern beide Bausteine können zur Ansteuerung gleicher Anlagen wiederverwendet werden.

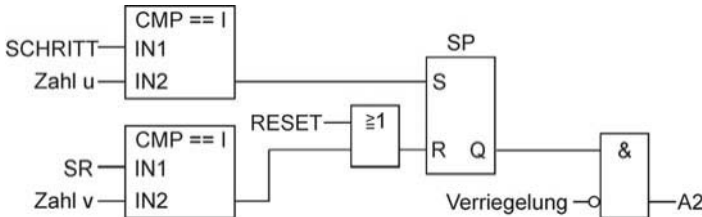
11.4.2 Regeln für die Programmierung des Befehlsausgabebausteins

Als Bausteintyp kann in den meisten Fällen eine Funktion FC (z. B. FC 16) verwendet werden. Sind allerdings lokale statische Variable in dem Baustein erforderlich, muss ein Funktionsbaustein FB (z. B. FB 16) verwendet werden.

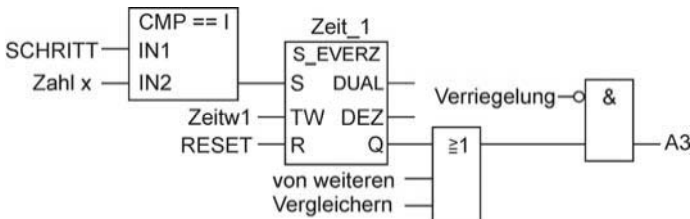
- Die Auswertung der aktuellen Schrittnummer erfolgt durch einen Vergleich.
- Ist das Bestimmungszeichen der Aktion ein „N“, muss eine einfache Zuweisung programmiert werden. Wird der Aktor von mehreren Ablaufschritten nichtspeichernd angesteuert, müssen die Einzelzuweisungen ODER-verknüpft werden.



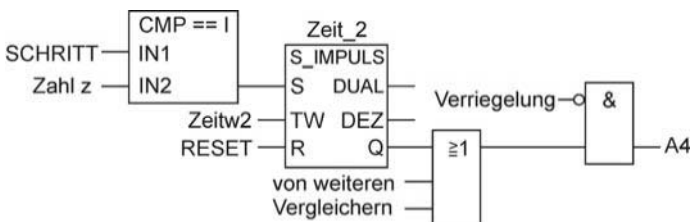
- Ist das Bestimmungszeichen der Aktion ein „S“, muss eine SR-Speicherfunktion verwendet werden, die durch eine nachfolgende Aktion „R“ beendet wird.



- Ist das Bestimmungszeichen ein „D“, wird die Zeit durch die Zeitfunktion SE bzw. durch einen TON-Funktionsblock gebildet.



- Ist das Bestimmungszeichen ein „L“, wird die Zeit durch eine Zeitfunktion SI bzw. durch einen TP-Funktionsblock gebildet.



11.4.3 Realisierung

Zur Umsetzung der Ablaufkette in ein Steuerungsprogramm wird der universelle Schrittketten-Funktionsbaustein FB 15 aufgerufen und an den Eingängen anlagenspezifisch parametrierbar. Dabei sind einfach die Transitionsbedingungen an die entsprechenden Eingänge T1_2 bis T10_1 zu legen. Nicht benötigte Transitionen können dabei unbeschaltet bleiben.

Der eigentliche Programmieraufwand liegt in der Erstellung des Ausgabebausteins FC 16 bzw. FB 16. Die Ausgänge dieses Bausteins steuern die Aktoren der Anlage an.

Diese Art der Realisierung liefert mit der Integer-Variablen „Schritt“, welche die Verbindung der beiden Bausteine FB 15 und FC 16 herstellt, die Möglichkeit zur aktuellen Schrittanzeige. Transferiert man diese Variable an ein Ausgangsbyte (z. B. AB6), so kann dort die Schrittnummer dualcodiert angezeigt werden.

11.4.4 Beispiel

■ Beispiel 11.2: Prägemaschine

Mit einer Prägemaschine soll auf Werkstücken eine Kennzeichnung eingepreßt werden. Dazu schiebt ein Schieber ein Werkstück aus dem Magazin in die Prägeform. Wenn die Prägeform belegt ist, stößt der Prägestempel abwärts und bewegt sich nach einer Wartezeit von 3 Sekunden wieder nach oben. Nach dem Prägevorgang stößt der Auswerfer das fertige Teil aus der Form, sodass es anschließend von dem Luftstrom aus der Luftdüse in den Auffangbehälter geblasen wird. Eine Lichtschranke spricht an, wenn das Teil in den Auffangbehälter fällt. Danach kann der nächste Prägevorgang beginnen.

Technologieschema:

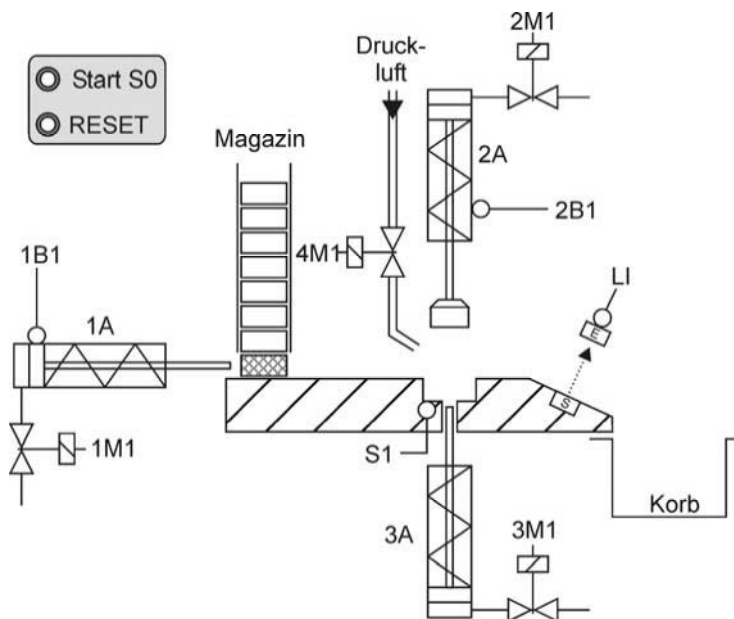
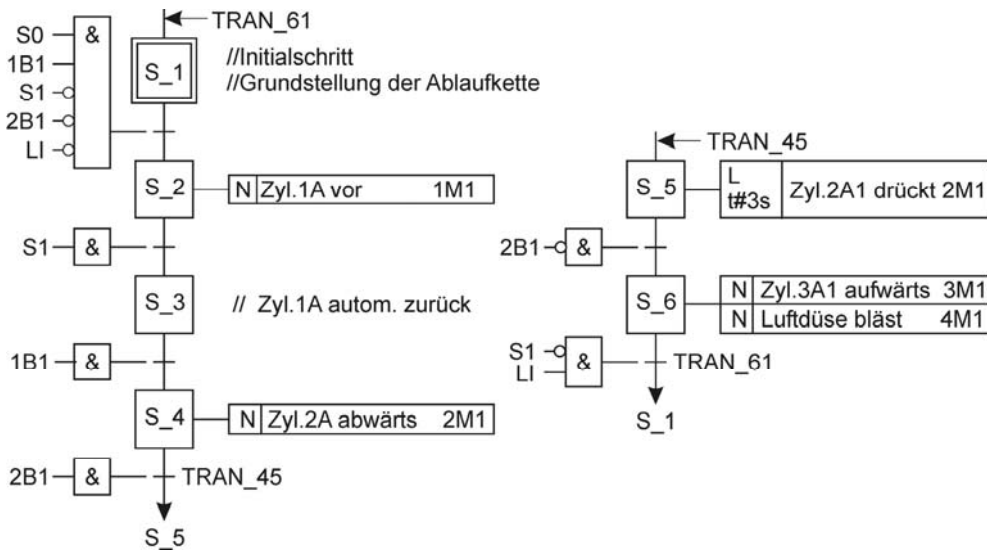


Bild 11.2: Prägemaschine (Die eingesetzten drei 3/2-Wege-Ventile sind nur schematisch dargestellt.)

Zuordnungstabelle der Eingänge und Ausgänge:

Eingangsvariable	Symbol	Datentyp	Logische Zuordnung		Adresse
Start-Taster	S0	BOOL	Betätigt	S0 = 1	E 0.0
Hintere Endl. Zyl. 1A	1B1	BOOL	Endlage erreicht	1B1 = 1	E 1.1
Prägeform belegt	S1	BOOL	Form belegt	S1 = 1	E 1.2
Vordere Endl. Zyl. 2A	2B1	BOOL	Endlage erreicht	2B1 = 1	E 1.3
Lichtschranke	LI	BOOL	Unterbrochen	LI = 1	E 1.4
RESET-Taster	RESET	BOOL	Betätigt	RESET = 1	E 0.7
Ausgangsvariable					
Magnetspule Ventil 1	1M1	BOOL	Zyl.1A fährt aus	1M1 = 1	A 5.1
Magnetspule Ventil 2	2M1	BOOL	Zyl.2A fährt aus	2M1 = 1	A 5.2
Magnetspule Ventil 3	3M1	BOOL	Zyl.3A fährt aus	3M1 = 1	A 5.3
Magnetspule Luftdüse	4M1	BOOL	Ventil offen	4M1 = 1	A 5.4

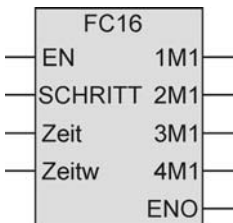
Ablauf-Funktionsplan:



Da der Schrittketten-Funktionsbaustein FB 15 für die Ablaufkette verwendet wird, muss nur der Befehlsausgabebaustein FC 16 entworfen werden.

Lösung in STEP 7

Deklarationstabelle des Ausgabebausteins FC 16:



Name	Datentyp
IN	
SCHRITT	Int
Zeit	Timer
Zeitw	SSTime
RESET	Bool

Name	Datentyp
OUT	
1M1	Bool
2M1	Bool
3M1	Bool
4M1	Bool

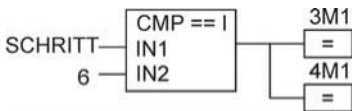
Das Programm der Funktion FC 16 für die Aktionen besteht aus den Ansteuerungen der Aktoren und der Programmierung der Zeitfunktion in Abhängigkeit von der Schrittnummer, wie im Ablauf-Funktionsplan vorgegeben.

Funktionsplan des Ausgabebausteins FC 16:

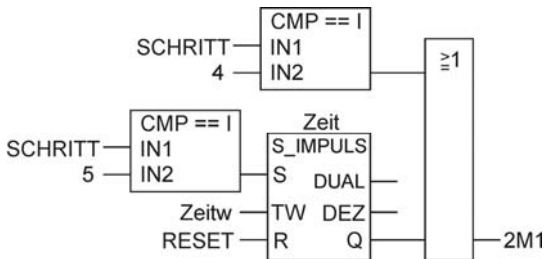
Zuweisung Ventil 1M1



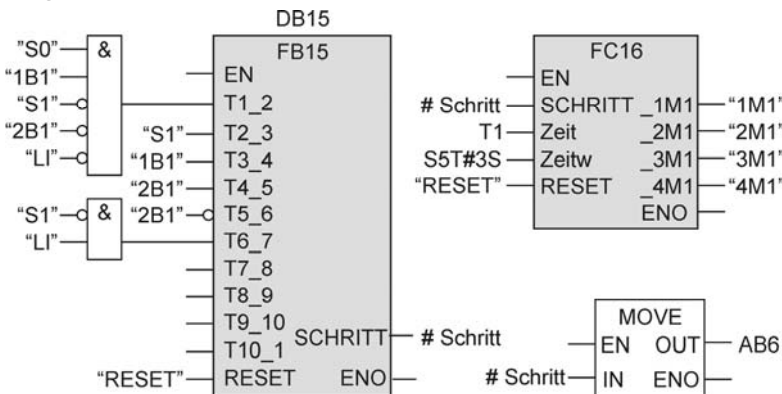
Zuweisung Ventil 3M1 und 4M1



Zuweisung Ventil 2M1



Programmstruktur im OB 1:

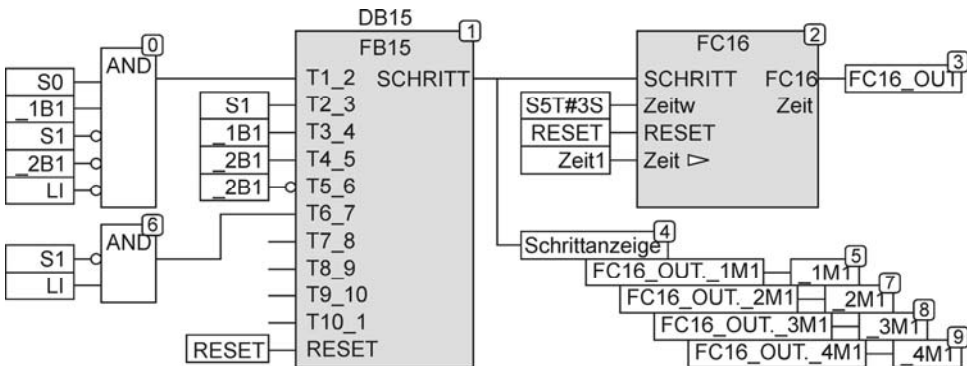


Die mit „#“ versehene Variable muss im OB 1 als temporäre Variable mit dem Datentyp Integer deklariert werden. Für die Eingangs- und Ausgangsoperanden (siehe Zuordnungstabelle) werden Symbole in der Symboltabelle vereinbart (z. B. E 0.0 → „S0“).

Lösung in CoDeSys

Auch bei CoDeSys ist bei Verwendung des Schrittkettenbausteins FB 15 aus der Bibliothek nur der Ausgabebaustein FC 16 zu programmieren. Das Programm entspricht dem oben dargestellten Funktionsplan. Statt der SE-Zeitfunktion wird dabei der TON-Funktionsblock verwendet, dessen Instanz „Zeit“ als IN_OUT-Variable deklariert wird.

Aufruf der Bausteine im PLC-PRG in der Programmiersprache CFC:



Hinweis: Die Programme und Bibliotheken für STEP 7 und CoDeSys können auf der Web-Seite: www.automatisieren-mit-sps.de herunter geladen werden.

11.5 Ablaufsteuerungen mit wählbaren Betriebsarten

11.5.1 Grundlagen

Fertigungstechnische oder verfahrenstechnische Ablaufsteuerungen sind Steuerungen mit einem zwangsweisen Ablauf der inneren Zustände der Steuerungsprogramme, die in eindeutig funktioneller und zeitlicher Zuordnung zu den technologischen Abläufen in der Anlage stehen und mit einer übergeordneten Funktionsebene zur Bedienerführung mit verschiedenen Betriebsarten ausgestattet sind.

Steuerungslösungen in der Form von Ablaufsteuerungen zeichnen sich durch einen übersichtlichen Programmaufbau mit typischen Steuerungsstrukturen aus. Beim praktischen Einsatz der Ablaufsteuerungen erweisen sich diese als wartungsfreundlich, da sie ein schnelles Erkennen von Fehlerursachen erlauben, die zumeist in fehlenden Weiterschaltbedingungen an den Schritzübergängen zu suchen sind. Allerdings bilden bereits kleinere Ablaufsteuerungen ein komplexes Steuerungsprogramm, da neben der Ablaufkette eine übergeordnete Funktionsebene für die Vorgabe unterschiedlicher Betriebsarten vorhanden ist.

Die Darstellung der Ablaufkette erfolgt mit den Elementen des Ablauf-Funktionsplanes und kann demgemäß übersichtlich gestaltet werden. Der Betriebsartenteil ist dagegen nur als Verknüpfungssteuerung realisierbar und gewöhnlich ohne Funktionsbeschreibung nicht leicht verstehbar. In der Praxis wird der Entwurf von Ablaufsteuerungen durch den Einsatz spezieller Software-Tools (z. B. S7-GRAPH) unterstützt, die es dem Anwendungsprogrammierer gestatten, sich ganz auf die technologische Lösung zu konzentrieren, da die Umsetzung auf die Steuerungsprogrammebene vom Programmiersystem geleistet wird und der Betriebsartenteil bereits integriert ist. Im diesem Lehrbuch wird auf den Einsatz solch mächtiger Software-Tools verzichtet und dafür die Ablaufsteuerung aus einzelnen Programmteilen zusammengesetzt.

11.5.2 Struktur

Eine Ablaufsteuerung kann in drei Programmteile gegliedert werden:

- Betriebsartenteil
- Ablaufkette
- Befehlsausgabe

Es liegt im Prinzip immer die nachfolgende, vereinfacht dargestellte Programmstruktur vor:

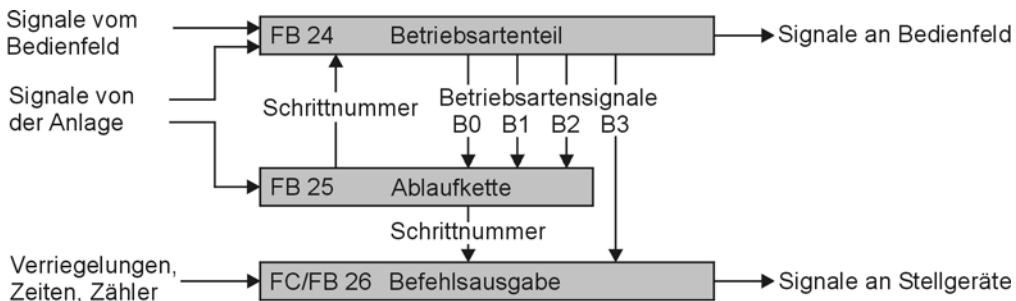


Bild 11.3: Struktur einer Ablaufsteuerung mit wählbaren Betriebsarten

Aufgaben des Betriebsartenteils:

Hier werden die Bedingungen für die Steuerungsabläufe vorgegeben. In der Regel lassen sich Automatikbetrieb, Einzelschrittbetrieb und Einrichtbetrieb zur Anlagensteuerungen wählen. Während im Einrichtbetrieb die Aktoren einer Anlage über Befehlsgeber direkt ohne Berücksichtigung der Ablaufkette angesteuert werden können, unterscheiden sich die beiden Betriebsarten Automatik und Einzelschritt durch Bedingungen der Schritt-Weiterschaltung der Ablaufkette. Beim Automatikbetrieb läuft die Schrittkette ohne Einwirkung eines Bedieners automatisch ab. Beim Einzelschrittbetrieb erfolgt die Weiterschaltung von einem zum nächsten Schritt nur unter der Leitung des Anlagenpersonals. Dabei kann bei der Weiterschaltung gewählt werden, ob die Übergangsbedingung erfüllt sein muss oder nicht. Zur Einstellung der Betriebsarten erzeugt ein *Betriebsartenteil* verschiedene Steuerbefehle für die Ablaufkette. Der Betriebsartenteil stellt somit das Bindeglied zwischen Bedienfeld und Ablaufkette dar.

Aufgaben der Ablaufkette:

Kernstück einer Ablaufsteuerung ist die Ablaufkette. In dieser wird das Programm für den schrittweisen Funktionsablauf der Steuerung bearbeitet. Die einzelnen Ablaufschritte werden abhängig von Übergangsbedingungen (Transitionen) in einer festgelegten Reihenfolge aktiviert. Für die Betriebsarten Automatik, Einzelschritt mit Bedingung und Einzelschritt ohne Bedingung steuern Signale aus dem Betriebsartenteil zusätzlich die Weiterschaltung.

Aufgaben der Befehlsausgabe:

In der Befehlsausgabe werden die den einzelnen Ablaufschritten zugeordneten Aktionen durch Ansteuerung der Stellgeräte ausgeführt. Dabei müssen Betriebsartenvorgaben genauso berücksichtigt werden wie Verriegelungssignale aus der Anlage, um die Steuerungssicherheit zu gewährleisten.

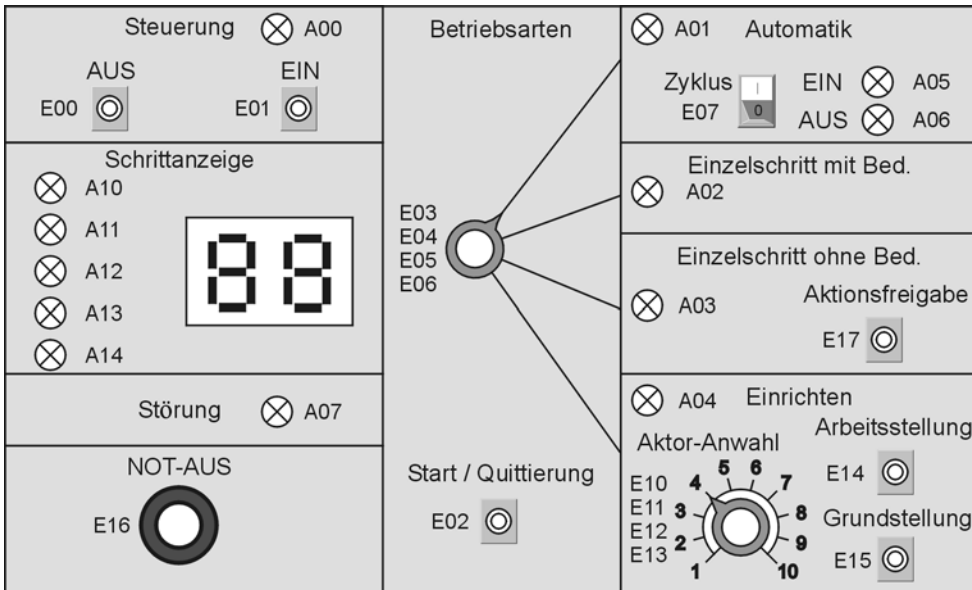
11.5.3 Bedien- und Anzeigefeld

Die Einstellung der Betriebsarten erfolgt von einem Bedienfeld aus. Neben den Bedienelementen besitzen Bedienfelder noch Leuchtmelder zur Anzeige der Betriebszustände und eine Schrittanzeige, an der der aktuelle Schritt abgelesen werden kann.

In der Praxis werden zur Bedienung und Beobachtung von Ablaufsteuerungen meist Panels eingesetzt, die maschinennah angebracht sind. Nachfolgend wird ein Bedien- und Anzeigefeld vorgestellt, das sich sowohl hardwaremäßig mit Tastern, Schaltern und Anzeigeleuchten als auch softwaremäßig mit einem Bedien- und Beobachtungssystem realisieren lässt

Neben einem EIN- und einem AUS-Taster für das Ein-/Ausschalten der Steuerung befindet sich auf dem Bedienfeld ein Wahlschalter, mit dem die Betriebsart Automatik, Einzelschritt mit Bedingung, Einzelschritt ohne Bedingung oder Einrichten eingestellt werden kann. Leuchtmelder geben jeweils eine Rückmeldung, welche Betriebsart gerade aktuell ist.

Für die Betriebsart „Einrichten“ ist ein Drehschalter zur Auswahl von 10 Aktoren auf dem Bedienfeld angebracht. Unter der Aktorauswahl ist hierbei beispielsweise ein Motor oder ein Zylinder zu verstehen. Mit den Tasten Arbeitsstellung bzw. Grundstellung kann im Einrichtbetrieb der Motor dann im Rechts- bzw. Linkslauf angesteuert oder der Zylinder in die Arbeitsstellung oder Grundstellung gefahren werden. Um „Crash-Situationen“ zu vermeiden, sind entsprechende Verriegelungen bei der Befehlsausgabe zu berücksichtigen.

Bedien- und Anzeigefeld:**Bild 11.4:** Bedien- und Anzeigefeld

Um Verwechslungen mit den Bezeichnungen von Signalgebern in Anlagen zu vermeiden, die in den Technologieschemata oftmals mit S0 ... Sn bezeichnet werden, sind die Eingänge des Bedienfeldes mit E00 bis E07 und E10 bis E17 benannt. Ebenso die Ausgänge von A00 bis A06 und A10 bis A14.

Beabsichtigte Funktionen der Befehlsgeber und Anzeigen des Bedienfeldes:**Taster E00: Steuerung_AUS**

Schaltet die Steuerung aus. (Anmerkung: In der Praxis besteht der Befehlsgeber für das EIN- und AUS-Schalten der Steuerung meist aus einem Schlüsselschalter).

Taster E01: Steuerung_EIN

Schaltet die Steuerung ein. Steuerung einschalten nach NOT-AUS oder Steuerung AUS.

Taster E02: Start / Quittierung

Automatikbetrieb: Ist die Betriebsart mit E03 vorgewählt, wird beim Betätigen des Tasters der Automatikbetrieb gestartet.

Einzelschrittbetrieb: Bei entsprechender Betriebsartenwahl erfolgt durch Taster-Betätigung eine Einzelschritt-Weiterschaltung der Schrittkette.

Bei ausgeschalteter Steuerung nach einer unregelmäßigen Betriebssituation wird durch Betätigen der Start/Quitt-Taste das RESET Betriebsartensignal B0 zum Rücksetzen der Schrittkette und der gespeicherten Aktionen ausgelöst.

Wahlschalter E03, E04, E05 und E06: Betriebsart

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

stoppt die Ansteuerung der Aktoren. Durch den Wahlschalter ist gewährleistet, dass stets nur einer der Eingänge E03, E04, E05 und E06 ein „1“-Signal annehmen kann.

Taster E07: Zyklus EIN / AUS

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

Wahlschalter E10 ... E13: Aktor-Anwahl

In der Betriebsart Einrichten können 10 verschiedene Aktoren angewählt werden. An den Eingängen E10 ... E13 liegt der Wert dualcodiert vor. Die Auswahl betrifft z. B. anzusteuernde Zylinder oder Motoren mit Rechts- und Linkslaufmöglichkeit.

Taster E14: Aktor-Arbeitsstellung

Die Bewegung wird nur bei gedrückter Taste ausgeführt. Der gewählte Zylinder fährt in die Arbeitsstellung bzw. der gewählte Motor dreht „rechts“. Anlagenspezifische Verriegelungen verhindern oder beenden die Bewegung bei „Crash“-Gefahr.

Taster E15: Aktor-Grundstellung

Die Bewegung wird nur bei gedrückter Taste ausgeführt. Der gewählte Zylinder fährt in die Grundstellung bzw. der gewählte Motor dreht „links“. Anlagenspezifische Verriegelungen verhindern oder beenden die Bewegung bei „Crash“-Gefahr.

Taster E16: NOT-AUS-Taster

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

Taster E17: Aktionsfreigabe

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

Anzeige A00: Leuchtmelder „Steuerung EIN“

Zeigt den eingeschalteten Zustand der Steuerung an.

Anzeige A01: Leuchtmelder „Automatik“

Leuchtet bei Zyklusbearbeitung im Automatikbetrieb.

Anzeige A02: Leuchtmelder „Einzelschritt mit Bedingung“

Leuchtet in der Betriebsart Einzelschritt mit Bedingung.

Anzeige A03: Leuchtmelder „Einzelschritt ohne Bedingung“

Leuchtet in der Betriebsart Einzelschritt ohne Bedingung.

Anzeige A04: Leuchtmelder „Einrichten“

Leuchtet in der Betriebsart Einrichten.

Anzeige A05: Leuchtmelder „Zyklus EIN“

Leuchtet in der Betriebsart Automatik, wenn Schalter „Zyklus“ auf EIN steht.

Anzeige A06: Leuchtmelder „Zyklus AUS“

Leuchtet in der Betriebsart Automatik, wenn der Schalter „Zyklus“ auf AUS steht und ein Bearbeitungszyklus abläuft.

Anzeige A07: Leuchtmelder „Störung“

Leuchtet in allen Betriebsarten, wenn eine Störung auftritt.

Schrittanz.: A10 ... A14

Sind für die dualcodierte Anzeige des jeweiligen Schrittes vorgesehen, 5 Bit für 32 Schritte (0 ... 31).

11.5.4 Betriebsartenteil-Baustein (FB 24: BETR)

Aufgabe des Bausteins für den Betriebsartenteil ist die programmgesteuerte Erzeugung von Betriebsartensignalen. Der Betriebsartenteil stellt das Bindeglied zwischen Anzeige- und Bedienfeld und der Schrittkette dar und erzeugt abhängig von den Befehlsgebern des Bedienfeldes die Betriebsartensignale.

Der Betriebsartenteil wird in einem Funktionsbaustein FB 24 bibliotheksfähig realisiert. Bei den nachfolgenden Beispielen wird dieser Baustein unverändert übernommen. Da das Programm für den Betriebsartenbaustein FB 24 empirisch entwickelt wurde, macht es wenig Sinn, sich ausführlich mit dessen Programm zu beschäftigen. Sowohl in der Ausbildung wie in der Praxisanwendung genügt es, die funktionale Beschreibung des Bausteins zu kennen.

Wie aus der Struktur der Ablaufsteuerung (Bild 11.3) zu sehen ist, sind folgende Betriebsartensignale in dem Funktionsbaustein FB 24 zu generieren:

- **Signal Rücksetzen RESET (B0)**

Mit dem Signal RESET (B0) werden alle Schrittspeicher einer Ablaufkette mit Ausnahme von Schritt 1 (Initialschritt) und alle gespeicherten Ausgabezuweisungen zurückgesetzt. Der Initialschritt S_1 wird durch das B0-Signal gesetzt. Dieser Vorgang heißt *Initialisierung der Ablaufkette* und muss zu Beginn der Inbetriebnahme einer Ablaufsteuerung durchgeführt werden.

- **Signal Weiterschalten mit Bedingungen W_mB (B1)**

Das Signal B1 wirkt in einer UND-Verknüpfung mit den Weiterschaltbedingungen auf die Ablaufkette. Das Weiterschalten von Schritt n nach Schritt n+1 setzt die erfüllte Weiterschaltbedingung und das Weiterschaltsignal B1 voraus. Nur wenn B1 den Signalwert „1“ hat, kann bei erfüllten Weiterschaltbedingungen in der Ablaufkette der nächste Schritt gesetzt werden. Umgekehrt kann durch Vorgabe des Signalwertes „0“ von B1 die Schrittweiterschaltung gesperrt werden, obwohl alle Weiterschaltbedingungen für diesen Schritt erfüllt sind.

- **Signal Weiterschalten ohne Bedingungen W_oB (B2)**

Mit dem Signal B2 *Weiterschalten ohne Bedingungen* soll in der Ablaufkette der nächste Schrittspeicher und nur dieser gesetzt werden, ohne dass die entsprechende Weiterschaltbedingung erfüllt ist. Eine solche Funktionalität hat nur unter bestimmten Randbedingungen wie z. B. einer Inbetriebnahme ihren Sinn. Man möchte beispielsweise von der Grundstellung aus gezielt in den vierten Schritt gehen, und zwar schrittweise durch hintereinander gegebene B2-Signale. Dazu muss das B2-Signal unabhängig von seiner Erzeugungsart immer ein Impulssignal sein, das für einen Programmzyklusdurchlauf den Signalwert „1“ hat. Ein B2-Signal als gespeichertes 1-Signal würde einen Kettenkreislauf ohne gezielte Anhaltmöglichkeit erzeugen.

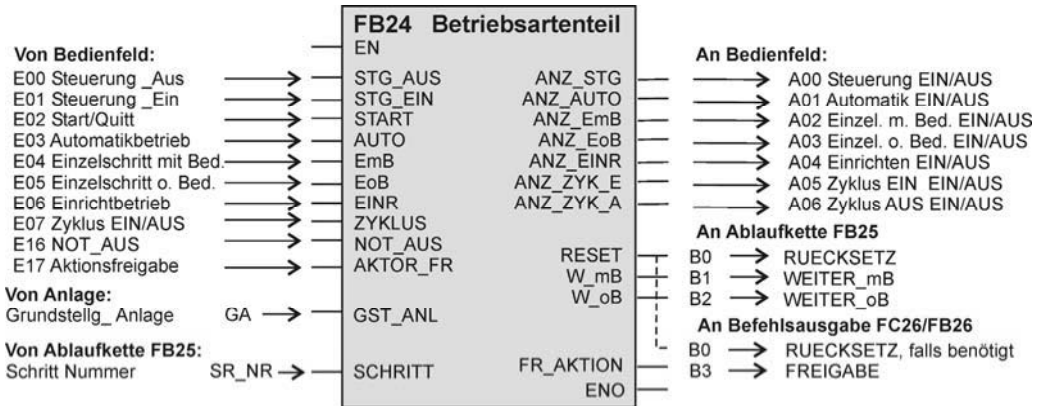
- **Signal Freigabe der Aktion FR_AKTION (B3)**

Mit dem Signal B3 *Aktionsfreigabe* wird im Ausgabebaustein die Ansteuerung der Aktoren eines Schrittes freigegeben. Im Einzelschrittbetrieb ohne Bedingungen hat dieses Signal erst den Wert „TRUE“, wenn die Aktionsfreigabetaste gedrückt ist. Außerdem wird durch dieses Signal verhindert, dass bei ausgeschalteter Steuerung ein Aktor angesteuert wird, wenn die Schrittkette in einem beliebigen Schritt steht.

Diese Betriebsartensignale werden im Ablaufkettenbaustein FB 25 und Befehlsausgabebaustein FC/FB 26 verarbeitet. Neben diesen Signalen werden vom Betriebsartenbaustein FB 24

noch die Anzeigesignale für das Bedienfeld generiert. Die Ausnahme hierbei bildet das Signal für die Störungsanzeige. Dieses muss je nach Anforderung der Störungsdiagnose in einem eigenen Baustein erzeugt werden. Denkbar sind Methoden der Zeitüberwachung, d. h., wie lange ein Schritt gesetzt sein darf, oder der Auswertung widersprüchlicher Sensormeldungen usw. Störungsdiagnose und Störmeldungen sind so komplex, dass die Betrachtung den Rahmen dieses Abschnittes sprengen würde.

Schnittstellen des Betriebsarten-Funktionsbausteins FB 24:

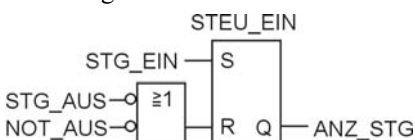


Deklarationstabelle des Betriebsarten-Funktionsbausteins FB 24:

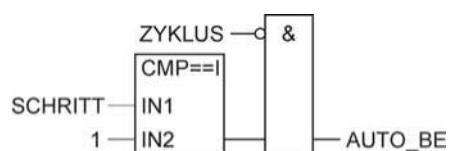
Name	Datentyp	Name	Datentyp	Name	Datentyp
IN		OUT		STAT	
STG_AUS	Bool	ANZ_STG	Bool	STEU_EIN	Bool
STG_EIN	Bool	ANZ_AUTO	Bool	AUTO_EIN	Bool
START	Bool	ANZ_EmB	Bool	AUTO_BE	Bool
AUTO	Bool	ANZ_EoB	Bool	START_SP	Bool
EmB	Bool	ANZ_EINR	Bool	FO1	Bool
EoB	Bool	ANZ_ZYK_E	Bool	FO2	Bool
EINR	Bool	ANZ_ZYK_A	Bool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

Funktionsplan des Betriebsartenbausteins FB 24:

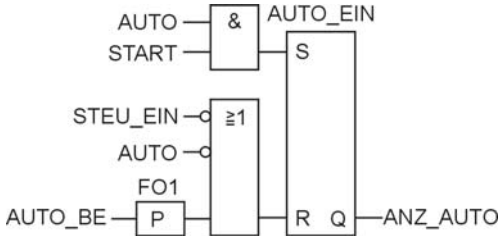
Steuerung EIN-AUS



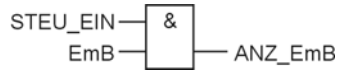
Vorwahl Automatikbetrieb beenden



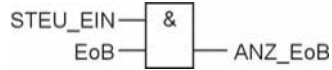
Automatikbetrieb



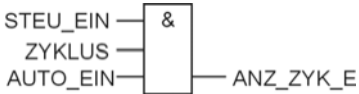
Anzeige Einzelschritt mit Bedingung



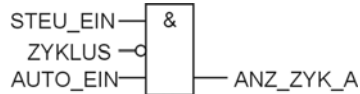
Anzeige Einzelschritt ohne Bedingung



Anzeige Zyklus EIN



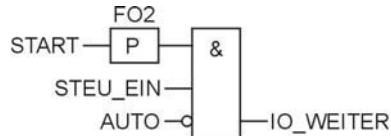
Anzeige Zyklus AUS



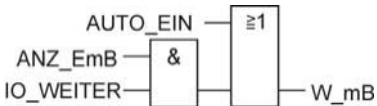
Anzeige Einrichtbetrieb



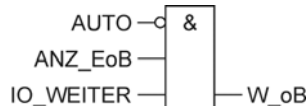
Einzelschritt Weiterschalt-Impuls



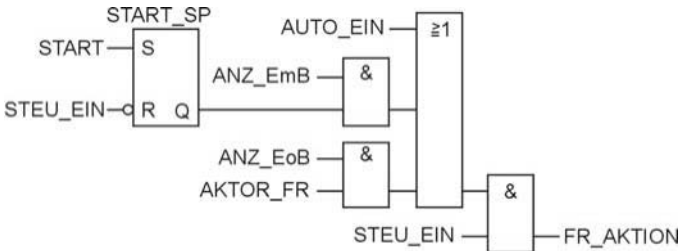
Weiterschalten Kette mit Bedingungen



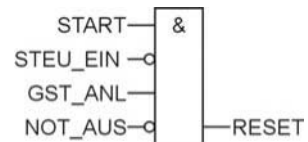
Weiterschalten Kette ohne Bedingungen



Aktionsfreigabe

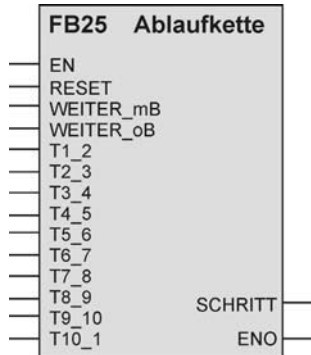


RESET



11.5.5 Ablaufkettenbaustein (FB 25: KET_10)

Für lineare Ablaufketten liefert in Abschnitt 11.4.1 dargestellte Funktionsbaustein FB 15 eine *standardisierte Lösung* für die Realisierung der Schrittfolge. Dieser Baustein muss für die Anwendung für Ablaufsteuerungen mit mehreren Betriebsarten noch durch die Betriebsartensignale B0 (RESET), B1 (Weiterschalten mit Bedingungen) und B2 (Weiterschalten ohne Bedingungen) ergänzt werden. Der so entstandene, nachfolgend dargestellte Funktionsbaustein FB 25 ist wieder für 10 Schritte konzipiert. Reicht die Anzahl der Schritte nicht aus, können entweder zwei Schrittfolgenbausteine FB 25 hintereinander gekoppelt oder die Anzahl der Schritte im Funktionsbaustein FB 25 erweitert werden. Bei verzweigten Schrittfolgen kann ebenfalls der Schrittfolgenbaustein FB 25 verwendet werden, wenn durch mehrmaligen Aufruf des Schrittfolgenbausteins FB 25 die verzweigte Kette in einzelne voneinander abhängige lineare Ketten zerlegt wird.

Funktionsbaustein FB 25:**Schnittstellen des Funktionsbausteins:****Beschreibung der Übergabeparameter:**

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

Die folgenden Tabellen geben die erforderlichen Datentypen und die Anfangswerte der Eingangs-, Ausgangs- und statischen Variablen an, welche in der Deklarationstabelle des Bausteins festzulegen sind.

Tabelle 11.2: Eingangsvariablen des Ablaufkettenbausteines FB 25

Parameter	Typ	Anfangswert	Beschreibung
RESET	BOOL	FALSE	Grundstellung der Ablaufkette (B0)
WEITER_mB	BOOL	FALSE	Weiterschalten mit Bedingungen (B1)
WEITER_oB	BOOL	FALSE	Weiterschalten ohne Bedingungen (B2)
T1_2	BOOL	FALSE	Weiterschaltbedingung Schritt1 nach Schritt2
T2_3 bis T10_1	BOOL	TRUE	Weiterschaltbedingungen: Nicht benötigte Schritte bleiben offen und werden durch den Anfangswert TRUE weitergeschaltet.

Tabelle 11.3: Ausgangsvariablen des Ablaufkettenbausteines FB 25

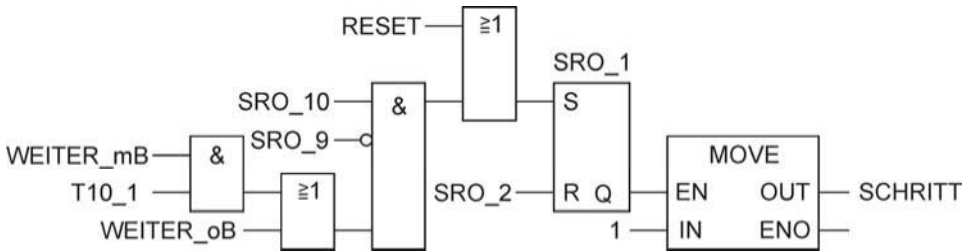
Parameter	Typ	Anfangswert	Beschreibung
SCHRITT	INT	0	Nummer des aktiven Schrittes

Tabelle 11.4: Statische Variablen des Ablaufkettenbausteines FB 25 (nach außen nicht sichtbar)

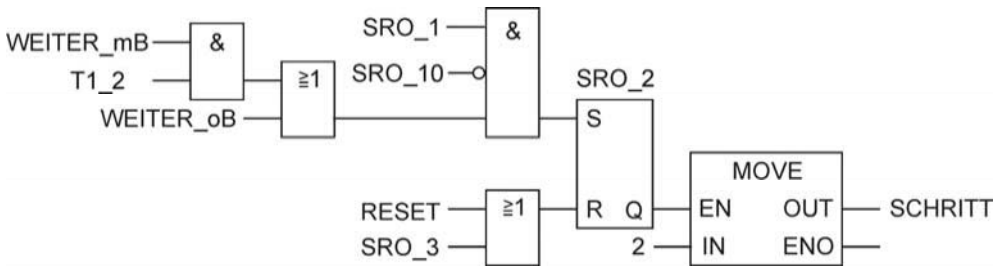
Statische Variablen	Typ	Anfangswert	Beschreibung
SRO_1	BOOL	TRUE	Schritt-Operand 1: Initialschritt
SRO_2 bis SRO_10	BOOL	FALSE	Schritt-Operanden 2 bis 10

Steuerungsprogramm des Ablaufketten-Funktionsbausteins FB 25:

Schritt 1: Initialschritt

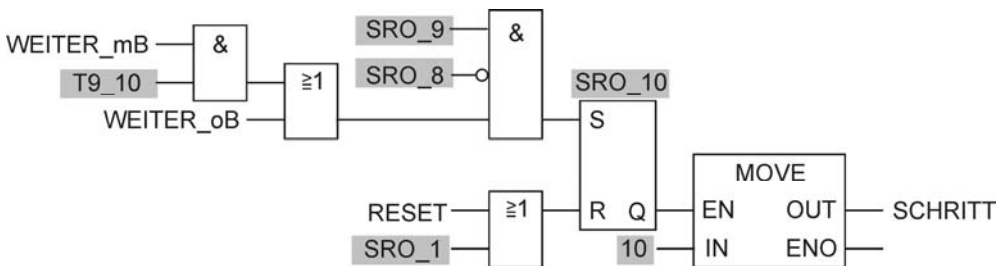


Schritt 2:



Die Funktionspläne der Schritte 3 bis Schritt 10 haben alle die gleiche Struktur wie Schritt 2. Im nachfolgend dargestellten Funktionsplan für Schritt 10 sind die Variablen grau hinterlegt, die bei dem jeweiligen Schritt angepasst werden müssen.

Schritt 10:



Hinweis: Bei Programmierung des Schrittketten-Funktionsbausteins mit CoDeSys wird statt der MOVE-Funktion die Funktion SEL verwendet, wie in Kapitel 11.4.1 gezeigt.

Funktionsbeschreibung:

Die Schrittkette wird bei erstmaliger Bearbeitung des Ablaufketten-Funktionsbausteines FB 25 statisch in den Grundzustand gesetzt: SRO_1 = 1 und alle anderen SRO_n = 0. Dies erfolgt durch die bei der Deklaration der statischen Variablen festgelegten Anfangswerte: SRO_1 = TRUE und SRO_n = FALSE. Ein Rücksetzsignal (B0) vom Betriebsartenteil kann die Schrittkette über den Eingangsparameter RESET ebenfalls in die Grundstellung bringen.

Für jeden Schritt ist ein Schritt-Speicher erforderlich, hier durch ein SR-Glied realisiert. Für jedes SR-Speicherglied ist ein Schritt-Operand als statische Lokalvariable deklariert (z. B. SRO_3 für Schritt 3), damit der Zustand des Schrittes für weitere Bearbeitungszyklen sowie bei mehrmaliger Verwendung des Funktionsbausteins gespeichert werden kann.

Der Speicher wird gesetzt, wenn die Übergangsbedingung erfüllt ist und der Vorgängerschritt SRO_2 gesetzt sowie der Vor-Vorgängerschritt SRO_1 rückgesetzt ist. Die Abfrage des Vor-Vorgängerschrittes ist nötig, um ein Durchschalten der Ablaufkette beim Weiterschalten ohne Bedingungen (B2) zu verhindern.

Die Übergangsbedingung kann auf zweifache Art erfüllt sein:

Der Eingangsparameter WEITER_mB ist „1“ und es liegt die Weiterschaltbedingung T2_3 = 1 vor. Der Eingangsparameter WEITER_mB wird von der gewählten Betriebsart beeinflusst. Bei Automatikbetrieb und bei Handbetrieb im Modus Weiterschalten mit Bedingungen wird vom Betriebsartenprogramm das Signal FR_K_mB erzeugt und mit dem Verknüpfungssignal B1 zum Ablaufkettenbaustein übertragen.

Der Eingangsparameter WEITER_oB ist „1“. Das Signal darf für die Weiterschaltung um einen Schritt jeweils nur für einen Zyklus anstehen (positive Flankenwertung erforderlich!). In diesem Fall hat die Weiterschaltbedingung T2_3 keinen Einfluss. Das ist denkbar, wenn beispielsweise die Betriebsart Einzelschritt ohne Weiterschaltbedingungen gewählt wurde. Es liegt dann die Absicht vor, die Schrittkette auf einen bestimmten Schritt einzustellen.

Das Rücksetzen des Schritt-Speichers erfolgt durch den Nachfolgerschritt oder durch das Rücksetzsignal bei RESET = „1“. Erfolgt das Setzen eines Schrittoperanden im Zyklus n, so wird der Vorgänger im Zyklus n+1 zurückgesetzt.

Die MOVE-Box dient der Übergabe der Schrittnummer des aktiven Schrittes, vom Eingang IN an den Ausgang OUT und dort an den Ausgangsparameter SCHRITT, der bei der Deklaration den Datentyp INTEGER erhalten hat. Die Steuerung der MOVE-Box erfolgt über ihren Freigabe-Eingang EN. Nur wenn ein Schritt-Speicher gesetzt, d. h., der Schritt aktiv ist, wird die Funktion der dortigen MOVE-Box ausgeführt. Es kommt nicht zu einer Überschreibung des Ausgangsparameters SCHRITT durch die Bearbeitung der MOVE-Box des nachfolgenden Schrittes, solange dieser nicht aktiv ist.

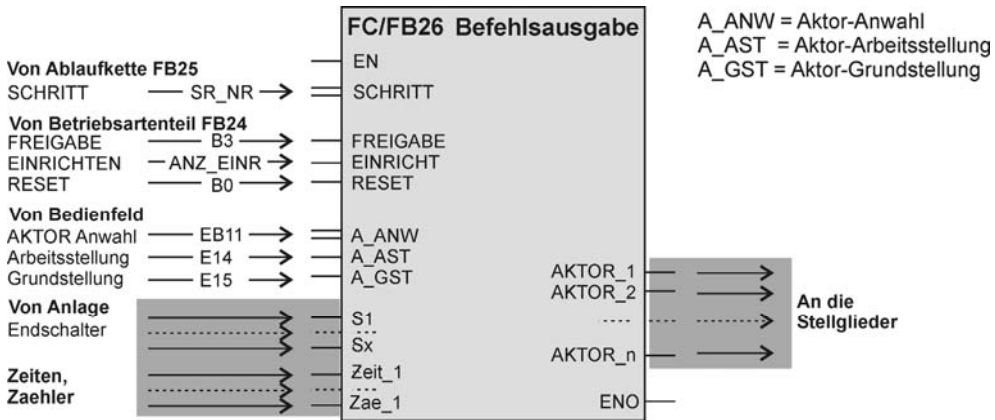
11.5.6 Befehlsausgabe

Im Befehlsausgabeteil der Ablaufsteuerung sollen die Befehle für die Stellgeräte (Aktoren) programmiert werden, und zwar entsprechend den Vorgaben der in den Aktionsblöcken der Ablaufkette angegebenen Bestimmungszeichen.

Für den Betriebsartenteil und die Ablaufkette wurden zwei Programme in bibliotheksfähigen Funktionsbausteinen vorgestellt. Für den Befehlsausgabeteil scheint dies nicht möglich zu sein, da zu unterschiedliche Anforderungen an einen solchen Baustein gestellt werden. Trotzdem ist es sinnvoll, die Befehlsausgaben in einem Baustein (z. B. FC 26/FB 26) zu kapseln, um dem Steuerungsprogramm eine übersichtliche Struktur zu geben. Zudem sind die nachfolgend angegebenen Regeln und bestimmte Teile des dargestellten Steuerungsprogramms anlagenunabhängig und können somit für jeden Befehlsausgabebaustein übernommen werden. Treten lokale statische Variablen in dem Ausgabebaustein auf, muss statt der Funktion FC 26 ein Funktionsbaustein FB 26 verwendet werden.

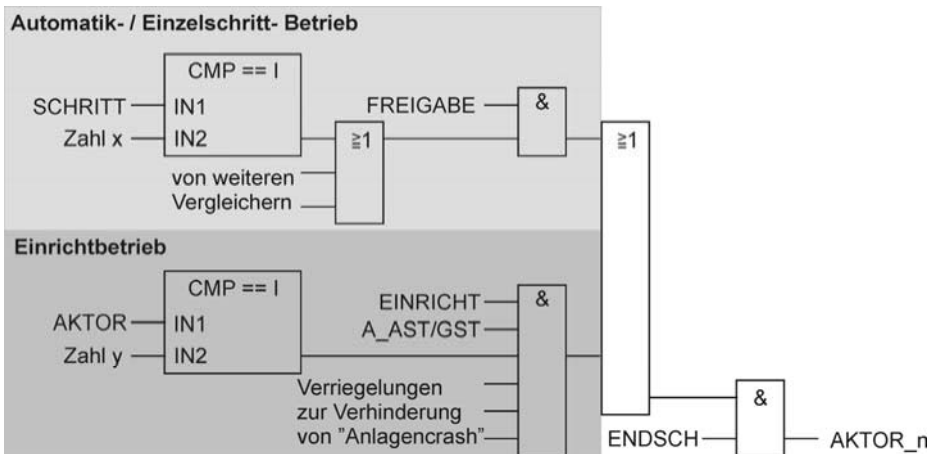
Der Befehlsausgabebaustein FC 26/FB 26 realisiert die Ansteuerung der Stellglieder unter Berücksichtigung der eingestellten Betriebsart und der erforderlichen Verriegelungsbedingungen. Einige Schnittstellen des Bausteins müssen auf die jeweilige Anlage angepasst werden. In der nachfolgenden Darstellung des Bausteins sind die Anschlüsse der anlagenabhängigen Ein- und Ausgangs-Operanden sowie mögliche STEP 7 Zeiten und Zähler dunkelgrau hinterlegt.

Schnittstellen des Befehlsausgabebausteins FC 26/FB 26:



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

Struktur der Ansteuerung eines Stellgliedes (Aktor_n) im Befehlsausgabebaustein:



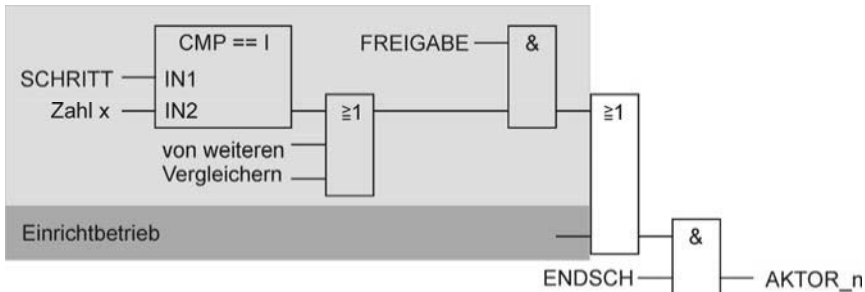
In den Betriebsarten Automatik, Einzelschritt und Einrichten ist bei der Ansteuerung von Stellgliedern, die eine Bewegung zur Folge haben und bestimmte Endschalter nicht überfahren dürfen, zusätzlich eine Endschalter-Verriegelung erforderlich. Endschalter liefern bei Betätigung ein 0-Signal durch einen zwangsöffnenden Kontakt. Deshalb ist der Endschalter an der UND-Verknüpfung bejaht abzufragen.

Nachfolgend werden die Regeln zur Ansteuerung eines Stellgliedes in den Betriebsarten Automatik, Einzelschritt (hellgraue Fläche) und Einrichten (dunkelgraue Fläche) getrennt beschrieben.

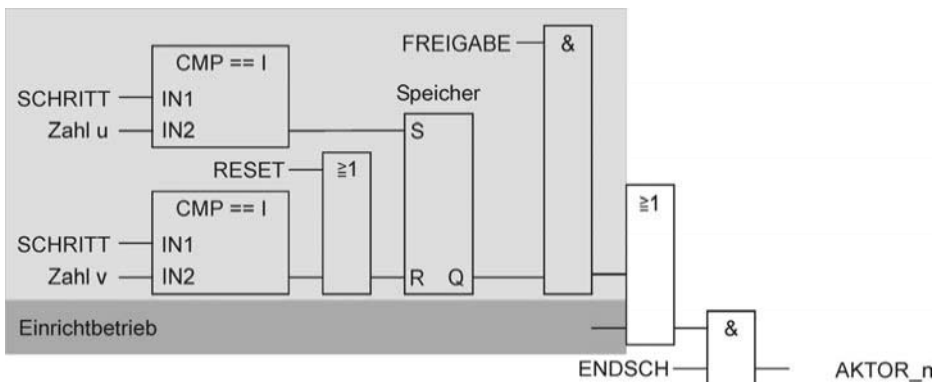
Ansteuerung eines Stellgliedes (Aktor_n) im Automatik bzw. Einzelschrittbetrieb:

Abhängig von den Bestimmungszeichen der Aktion in der Ablaufkette muss der hellgrau hinterlegte Teil der Stellglied-Ansteuerung ausgeführt werden. Dabei gelten folgende Regeln:

1. Jeder von der Ablaufkette beeinflusste Teil der Befehlsausgabe beginnt mit der Auswertung der Schrittnummer durch einen Vergleich. Da die Schrittnummer als Integerzahl übermittelt wird, ist ein Vergleichler auf „ist gleich“ für Integerzahlen einzusetzen.
2. Wird ein Stellglied von mehreren Ablaufschritten nichtspeichernd (Bestimmungszeichen „N“) angesteuert, müssen die Einzelwertzuweisungen mit einer ODER-Verknüpfung zusammengefasst werden.

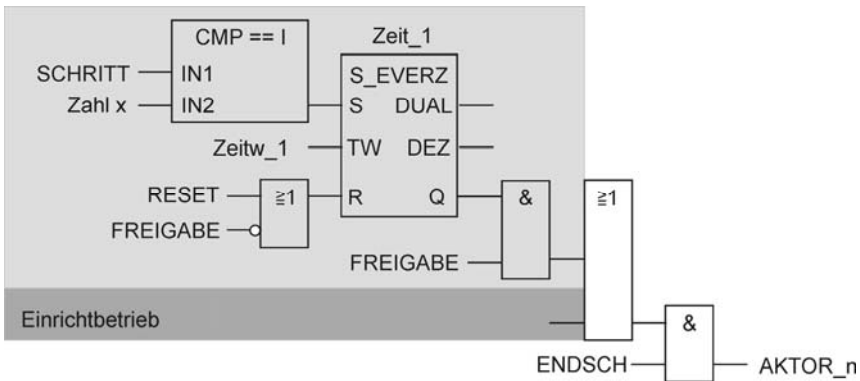


3. Ist die im Schritt auszuführende Aktion mit dem Bestimmungszeichen „S“ für gespeicherte Aktion gekennzeichnet, dann muss eine SR-Speicherfunktion verwendet werden. Der boolesche Ausgang des Vergleiches führt auf den Setzeingang des Speichers. Wichtig ist, dass ein später folgender Schritt der Ablaufkette die auszuführende Aktion beendet. Dazu muss im Aktionsblock für diese Aktion das Bestimmungszeichen „R“ für vorrangiges Rücksetzen angegeben sein.

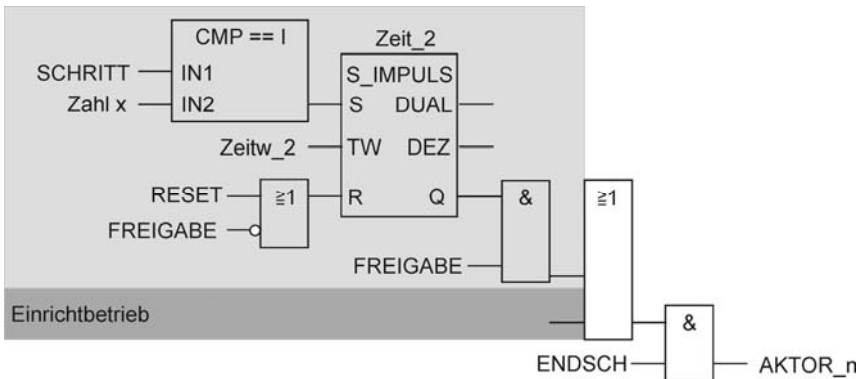


4. Zeigt der Aktionsblock eines Schrittes das Bestimmungszeichen „D“ für zeitverzögertes Einschalten eines Stellgliedes, so muss der Vergleicherausgang auf ein Zeitglied mit Einschaltverzögerung geführt werden (CoDeSys: TON; STEP 7: S_EVERZ). Am Zeitgliedausgang liegt wieder das bereits erwähnte UND-Glied. Im **Einzelschrittbetrieb** kann der Fall vorkommen, dass der aktive Ablaufschritt den Zeitgliedeingang aktiviert hält und der Zeitablauf nicht abgewartet werden soll. Das erfordert eine Rücksetzmöglichkeit für das Zeitglied z. B. mit dem Signal FREIGABE (B3) oder auch durch das Signal RESET (B0).

Ansteuerung eines Aktors, wenn das Bestimmungszeichen „D“ ist:



5. Zeigt der Aktionsblock eines Schrittes das Bestimmungszeichen „L“ für zeitbegrenztes Einschalten eines Stellgliedes, so ist der Vergleicherausgang auf ein Impuls-Zeitglied zu führen.(CodeSys: TP; STEP 7: S_IMPULS).



Für andere Bestimmungszeichen in den Aktionsblöcken, sind entsprechende Verknüpfungen zu bilden.

Steuerungsstruktur im Einrichtbetrieb:

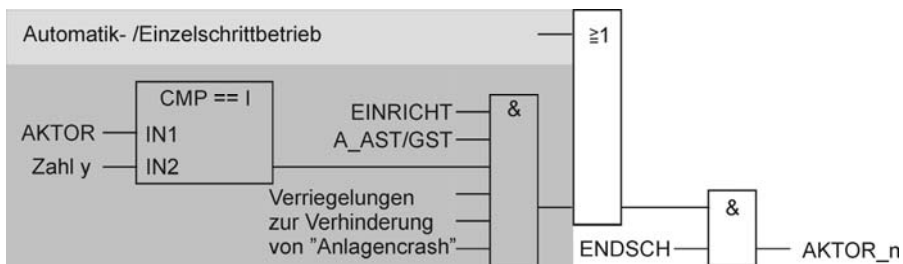
Der in den vorangegangenen Bildern noch unbestimmt gebliebene, dunkelgrau hinterlegte Teil für den Einrichtbetrieb muss anlagenabhängig realisiert werden.

Folgende Regeln sind beim Entwurf zu berücksichtigen:

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

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

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



Hinweis: Der Teil der UND-Verknüpfung, der die Freigabe durch das Einricht-Signal, das Signal Aktor in die Grundstellung (A-GST) bzw. Aktor in die Arbeitsstellung (A_AST) und die Aktorauswahl enthält, ist bei allen Aktor-Ansteuerungsfunktionen gleich.

3. Zu berücksichtigen sind noch mögliche „Crash-Situationen“, die im Einrichtbetrieb vorkommen können. Diese sind durch Verriegelungen am UND-Glied zu verhindern.

Nach Fertigstellung des Steuerungsprogramms sind die Ansteuerungsbedingungen der Stellglieder im Einrichtbetrieb nochmals einer genauen Prüfung zu unterziehen, ob alle Sicherheitsanforderungen tatsächlich auch erfüllt sind.

11.5.7 Realisierung

Zur Umsetzung einer Ablaufsteuerung mit wählbaren Betriebsarten in ein Steuerungsprogramm werden die universellen Funktionsbausteine Betriebsarten FB 24 und Ablaufkette FB 25 aufgerufen. Bei der Parametrierung der Bausteine sind dabei nur der Eingang GA für die Grundstellung der Anlage beim Betriebsarten-Baustein FB 24 und die Transitionsbedingungen des Ablaufkettenbausteins FB 25 anlagenabhängig.

Der eigentliche Programmieraufwand liegt in der Anpassung des Ausgabebausteins FC 26/FB 26 an die speziellen Anlagenanforderungen. Als Ausgangsvariablen müssen alle Stellglieder der Anlage deklariert werden. Die erforderlichen Verriegelungen müssen als zusätzliche Eingänge des Bausteins vereinbart werden.

Zu prüfen ist für die zu steuernde Anlage noch, ob alle vier vorgestellten Betriebsarten im konkreten Anwendungsfall auch sinnvoll sind. Ist dies nicht der Fall, können entsprechende Befehlsgeber und Anzeigen auf dem Bedienfeld außer Betracht bleiben. Der dargestellte standardisierte Betriebsarten-Funktionsbaustein FB 24 kann aber trotzdem weiter verwendet werden. Die zugehörigen Ein- bzw. Ausgänge des Bausteins bleiben dann unbeschaltet.

11.5.8 Beispiel

■ Beispiel 11.3: Biegemaschine

Auf einer Biegemaschine werden Bleche gebogen. Die Zuführung und der Abtransport der Bleche ist nicht Gegenstand dieser Steuerungsaufgabe. Meldet Sensor S1, dass ein Blech eingelegt ist, fährt das Schutzgitter nach einer Wartezeit von 5 Sekunden durch Ansteuerung des Schützes Q1 zu. Ist das Schutzgitter geschlossen, fährt Zylinder 1A aus und hält das Blech fest. Zylinder 2A biegt das Blech zunächst um 90°, bevor Zylinder 3A das Blech in die endgültige Form bringt. Zylinder 3A verharrt dabei drei Sekunden in der vorderen Endlage. Ist Zylinder 3A wieder in der oberen Endlage, fährt auch Zylinder 1A zurück. Danach wird das Schutzgitter durch Ansteuerung von Q2 geöffnet. Nach Entfernen des gebogenen Blechs, kann der Biegevorgang wiederholt werden.

Zylinder 1A wird durch ein 5/2-Wegeventil mit beidseitiger elektromagnetischer Betätigung angesteuert. Die Zylinder 2A und 3A werden durch 5/2-Wegeventile mit einseitiger elektromagnetischer Betätigung und Federrückstellung angesteuert. Aus dem Technologieschema ist ersichtlich, welcher Elektromagnet der 5/2-Wegeventile jeweils angesteuert werden muss.

Technologieschema:

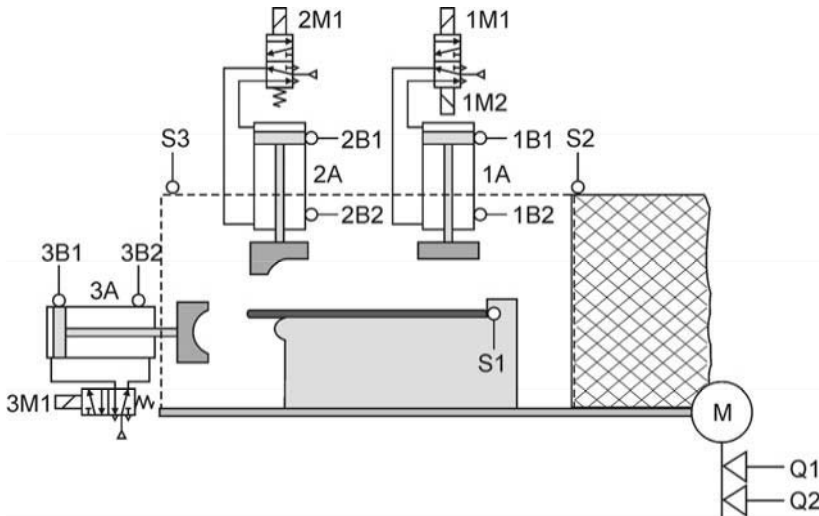


Bild 11.5: Biegemaschine

Zuordnungstabelle der Eingänge und Ausgänge der Anlage:

Eingangsvariable	Symbol	Datentyp	Logische Zuordnung	Adresse	
Hintere Endl. Zylinder 1	1B1	BOOL	Endlage erreicht	1B1 = 1	E 0.0
Vordere Endl. Zylinder 1	1B2	BOOL	Endlage erreicht	1B2 = 1	E 0.1
Hintere Endl. Zylinder 2	2B1	BOOL	Endlage erreicht	2B1 = 1	E 0.2
Vordere Endl. Zylinder 2	2B2	BOOL	Endlage erreicht	2B2 = 1	E 0.3
Hintere Endl. Zylinder 3	3B1	BOOL	Endlage erreicht	3B1 = 1	E 0.4
Vordere Endl. Zylinder 3	3B2	BOOL	Endlage erreicht	3B2 = 1	E 0.5
Sensor Blech vorhanden	S1	BOOL	Blech vorhanden	S1 = 1	E 0.6
Sensor Schutzgitter auf	S2	BOOL	Schutzgitter ist auf	S2 = 0	E 0.7
Sensor Schutzgitter zu	S3	BOOL	Schutzgitter ist zu	S3 = 0	E 1.0
Ausgangsvariable					
Magnetspule 1 Zyl. 1A	1M1	BOOL	Zyl. 1A fährt aus	1M1 = 1	A 4.1
Magnetspule 2 Zyl. 1A	1M2	BOOL	Zyl. 1A fährt zurück	1M2 = 1	A 4.2
Magnetspule 1 Zyl. 2A	2M1	BOOL	Zyl. 2A fährt aus	2M1 = 1	A 4.3
Magnetspule 1 Zyl. 3A	3M1	BOOL	Zyl. 3A fährt aus	3M1 = 1	A 4.4
Motor Schutzgitter zu	Q1	BOOL	Schutzgitter fährt zu	Q1 = 1	A 4.5
Motor Schutzgitter auf	Q2	BOOL	Schutzgitter fährt auf	Q2 = 1	A 4.6

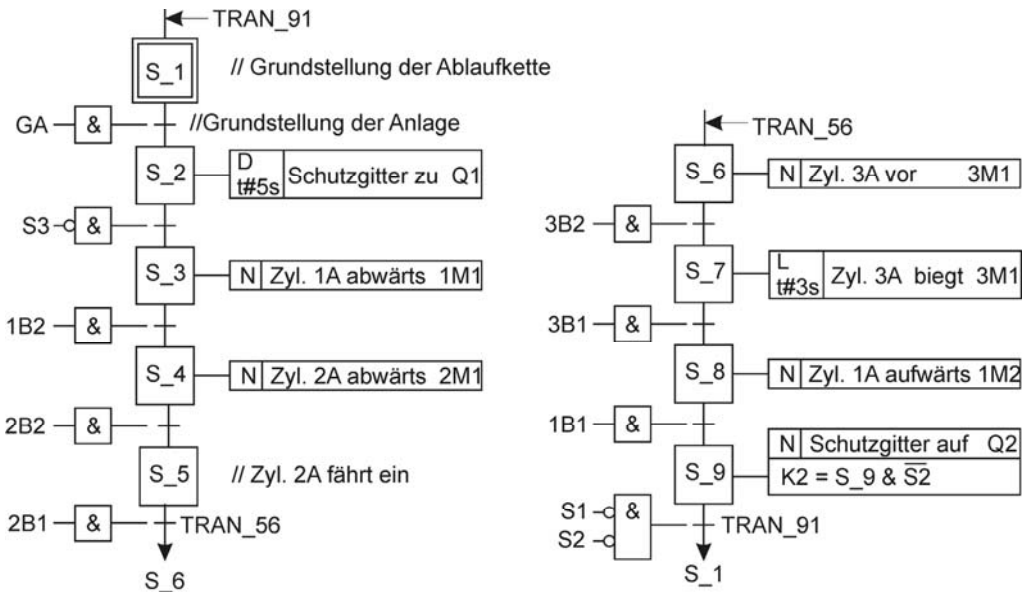
Für die Biegemaschine sollen die Betriebsarten Automatik, Einzelschritt und Einrichten möglich sein. Dazu wird das in Abschnitt 11.5.3 vorgestellte Bedien- und Anzeigefeld verwendet.

Zuordnungstabelle der Eingänge und Ausgänge des Bedien- und Anzeigefeldes:

Eingangsvariable	Symbol	Datentyp	Logische Zuordnung	Adresse
Steuerung EIN	<u>E00</u>	BOOL	Betätigt <u>E00 = 0</u>	E 10.0
Steuerung AUS	<u>E01</u>	BOOL	Betätigt <u>E01 = 1</u>	E 10.1
Taster Start/Quittierung	<u>E02</u>	BOOL	Betätigt <u>E02 = 1</u>	E 10.2
Auswahl Automatik	<u>E03</u>	BOOL	Ausgewählt <u>E03 = 1</u>	E 10.3
Auswahl Einzelschr. m.B.	<u>E04</u>	BOOL	Ausgewählt <u>E04 = 1</u>	E 10.4
Auswahl Einzelschr. o.B.	<u>E05</u>	BOOL	Ausgewählt <u>E05 = 1</u>	E 10.5
Auswahl Einrichten	<u>E06</u>	BOOL	Ausgewählt <u>E06 = 1</u>	E 10.6
Wahlschalter Zyklus	<u>E07</u>	BOOL	Zyklus ein <u>E07 = 1</u>	E 10.7
Aktor-Anwahl	E10_E13	½ BYTE	Dualzahl von 1 bis 10	E11.0–11.3
Taster Arbeitsstellung	<u>E14</u>	BOOL	Betätigt <u>E14 = 1</u>	E 11.4
Taster Grundstellung	<u>E15</u>	BOOL	Betätigt <u>E15 = 1</u>	E 11.5
NOT-AUS	<u>E16</u>	BOOL	Betätigt <u>E16 = 0</u>	E 11.6
Taster Aktionsfreigabe	<u>E17</u>	BOOL	Betätigt <u>E14 = 1</u>	E 11.7
Ausgangsvariable				
Anzeige Steuerung EIN	<u>A00</u>	BOOL	Leuchtet <u>A00 = 1</u>	A 10.0
Anzeige Automatik	<u>A01</u>	BOOL	Leuchtet <u>A01 = 1</u>	A 10.1
Anzeige Einzelschr. m. B.	<u>A02</u>	BOOL	Leuchtet <u>A02 = 1</u>	A 10.2
Anzeige Einzelschr. o. B.	<u>A03</u>	BOOL	Leuchtet <u>A03 = 1</u>	A 10.3
Anzeige Einrichten	<u>A04</u>	BOOL	Leuchtet <u>A00 = 1</u>	A 10.4
Anzeige Zyklus EIN	<u>A05</u>	BOOL	Leuchtet <u>A01 = 1</u>	A 10.5
Anzeige Zyklus AUS	<u>A06</u>	BOOL	Leuchtet <u>A02 = 1</u>	A 10.6
Schrittanzeige	A10_A14	5xBool	Dualzahl von 0 ... 31	A11.0–11.4

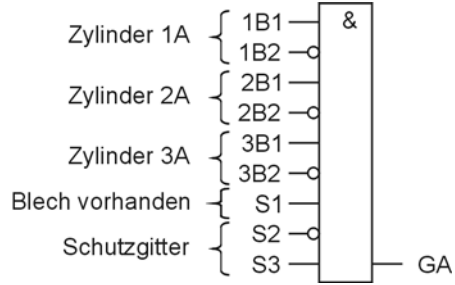
Hinweis: Damit die Ein- Ausgabeoperanden für das Bedienfeld als Symbole in die Symboltabelle übernommen werden können, wird jeweils ein Unterstrich vor das Symbol hinzugefügt.

Ablauf-Funktionsplan:



Grundstellung der Anlage

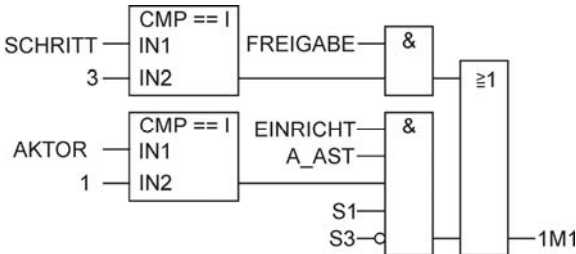
Das Signal GA für die Grundstellung der Anlage wird durch die Sensor-Abfrage der Anlage gebildet. Die drei Zylinder müssen sich in der hinteren Endlage befinden (Abfrage durch 1B1, 2B1 und 3B1). Die negierte Abfrage der Sensoren für die vordere Endlage überprüft, ob die entsprechenden Sensoren das richtige Signal geben. Mit Sensor S1 wird abgefragt, ob sich ein Blech in der Biegemaschine befindet. Mit den Sensoren S2 und S3 wird die offene Stellung des Schutzgitters abgefragt.



Die Umsetzung des Ablauf-Funktionsplans erfolgt durch die Verwendung der vorgestellten Bausteine FB 24 Betriebsarten, FB 25 Ablaufkette und FC 26 Befehlsausgabe. Während die Bausteine FB 24 und FB 25 lediglich parametrisiert werden müssen, sind bei dem Befehlsausgabebaustein die notwendigen Verriegelungen bei der Ansteuerung der Stellglieder zu bedenken.

Funktionsplan Befehlsausgabebaustein FC 26:

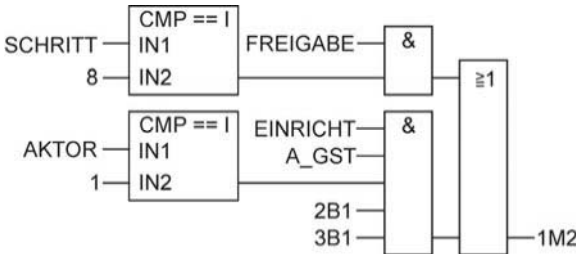
Ansteuerung 1M1 Zylinder 1A vor



Zylinder 1A ist Aktor Nr. 1

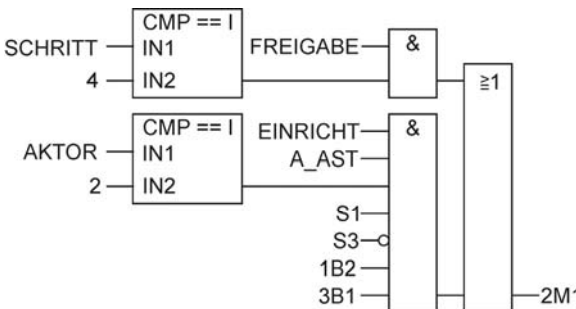
Im Einrichtbetrieb darf der Zylinder 1A nur ausfahren, wenn die Schutztür geschlossen und ein Blech eingelegt ist.

Ansteuerung 1M2 Zylinder 1A zurück



Im Einrichtbetrieb darf der Zylinder 1A nur einfahren, wenn sich die beiden anderen Zylinder im eingefahrenen Zustand befinden.

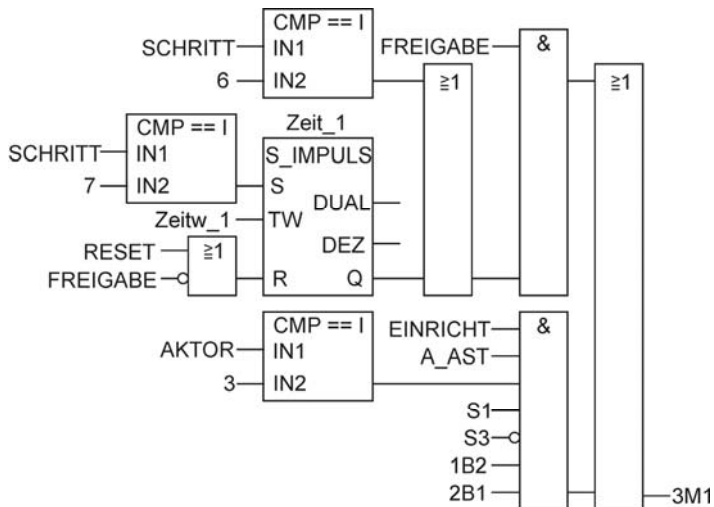
Ansteuerung 2M1 Zylinder 2A vor



Zylinder 2A ist Aktor Nr. 2

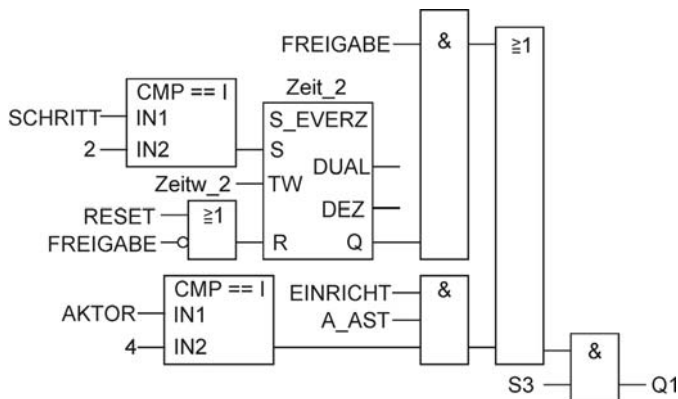
Im Einrichtbetrieb darf der Zylinder 2A nur ausfahren, wenn die Schutztür geschlossen und ein Blech eingelegt ist. Außerdem muss das Blech gespannt und Zyl. 3A eingefahren sein.

Ansteuerung 3M1 Zylinder 3A vor



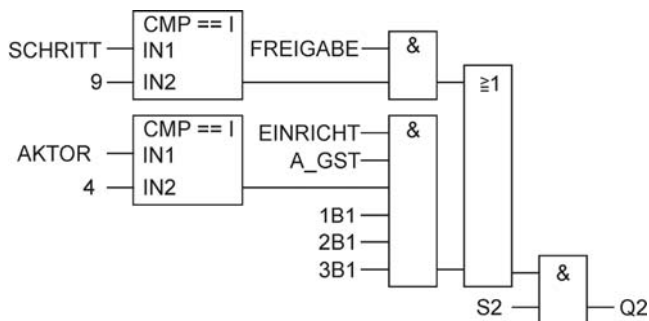
Zylinder 3A ist Aktor Nr. 3
 Im Einrichtbetrieb darf der Zylinder 3A nur ausfahren, wenn die Schutztür geschlossen und ein Blech eingelegt ist. Außerdem muss das Blech gespannt und Zyl. 2A eingefahren sein.

Ansteuerung Q1 Motorschütz Schutzgitter zu



Der Motor M ist Aktor Nr. 4
 Im Einrichtbetrieb darf das Schutzgitter ohne besondere Bedingung zufahren. Der Endschalter S3 verriegelt in allen Betriebsarten die Ansteuerung von Q1.

Ansteuerung Q2 Motorschütz Schutzgitter auf



Der Motor M ist Aktor Nr. 4
 Im Einrichtbetrieb darf die Schutzgitter nur aufgesteuert werden, wenn sich alle drei Zylinder in der oberen Endlage befinden. Der Endschalter S2 verriegelt in allen Betriebsarten die Ansteuerung von Q2.

Im ersten Netzwerk des Befehlsausgabebausteins FC 26 ist noch die Eingangsvariable A_ANW mit B#16#0F zu maskieren und der temporären Variablen AKTOR zuzuweisen (siehe nebenstehende AWL).

L #A_ANW
 L B#16#0F
 UW
 T #Aktor

Deklarationstabelle des Befehlsausgabebausteins FC 26:

Name	Datentyp
IN	
SCHRITT	Int
FREIGABE	Bool
EINRICHT	Bool
RESET	Bool
AKT_ANW	Byte
A_AST	Bool
A_GST	Bool

Name	Datentyp
IN	
<u>1B2</u>	Bool
<u>2B1</u>	Bool
<u>3B1</u>	Bool
S1	Bool
S2	Bool
S3	Bool
<u>Zeit_1</u>	Timer
<u>Zeitw_1</u>	S5Time
<u>Zeit_2</u>	Timer
<u>Zeitw_2</u>	S5Time

Name	Datentyp
OUT	
<u>1M1</u>	Bool
<u>1M2</u>	Bool
<u>2M1</u>	Bool
<u>3M1</u>	Bool
Q1	Bool
Q2	Bool

Name	Datentyp
TEMP	
AKTOR	Int

Hinweis: Da die Syntax von STEP 7 und CoDeSys in der Deklarationstabelle keine Zahlen als erstes Zeichen beim Namen der lokalen Variablen zulässt, beginnen die entsprechenden Variablen mit einem Unterstrich. (Beispiel: statt 1B1 wird 1B1 geschrieben.)

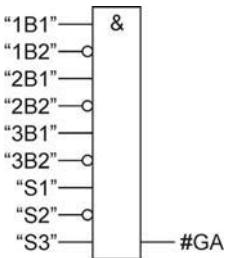
Lösung in STEP 7

Nachfolgend ist das Programm des Organisationsbausteins OB 1 im Funktionsplan angegeben. Neben den Bausteinaufrufen von FB 24 Betriebsartenteil, FB 25 Schrittkette und FC 26 Befehlsausgabe sind noch die Netzwerke für die Grundstellung und die Schrittanzeige dargestellt.

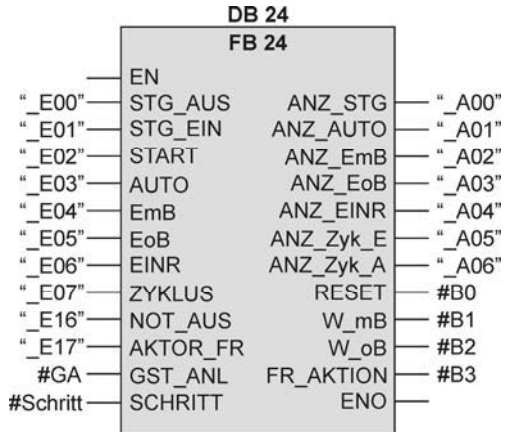
Für die SPS-Operanden sind die in der Zuordnungstabelle der Aufgabe gegebenen Symbole verwendet worden. Die mit „#“ gekennzeichneten Variablen müssen im OB 1 als temporäre Variablen mit dem Datentyp BOOL bzw. INT deklariert werden.

Aufruf der Bausteine und Zuweisungen im OB 1:

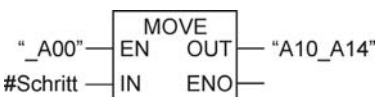
Netzwerk 1: Grundstellung der Anlage GA



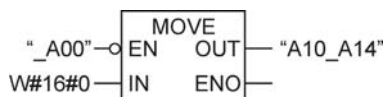
Netzwerk 2: Betriebsarten-Funktionsbaustein FB 24



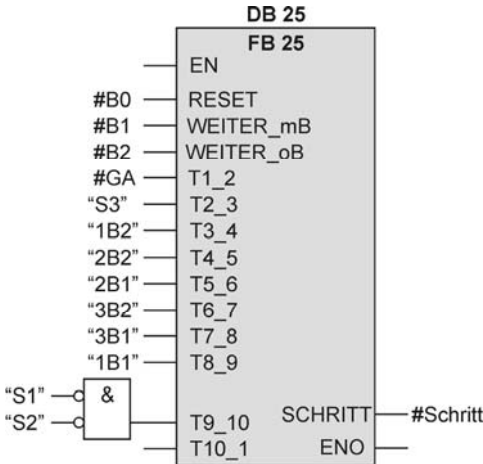
Netzwerk 3: Schrittanzeige bei Steuerung EIN



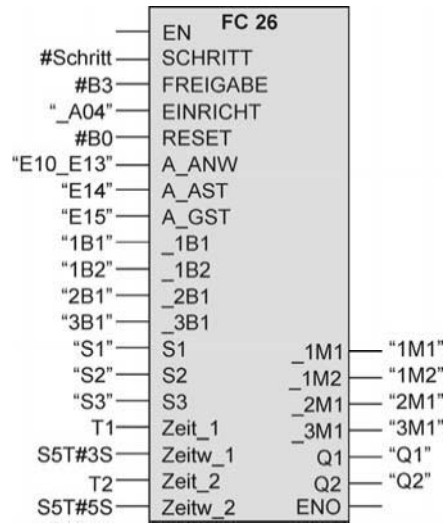
Netzwerk 4: Schrittanzeige bei Steuerung AUS



Netzwerk 5: Schrittketten-Funktionsbaustein

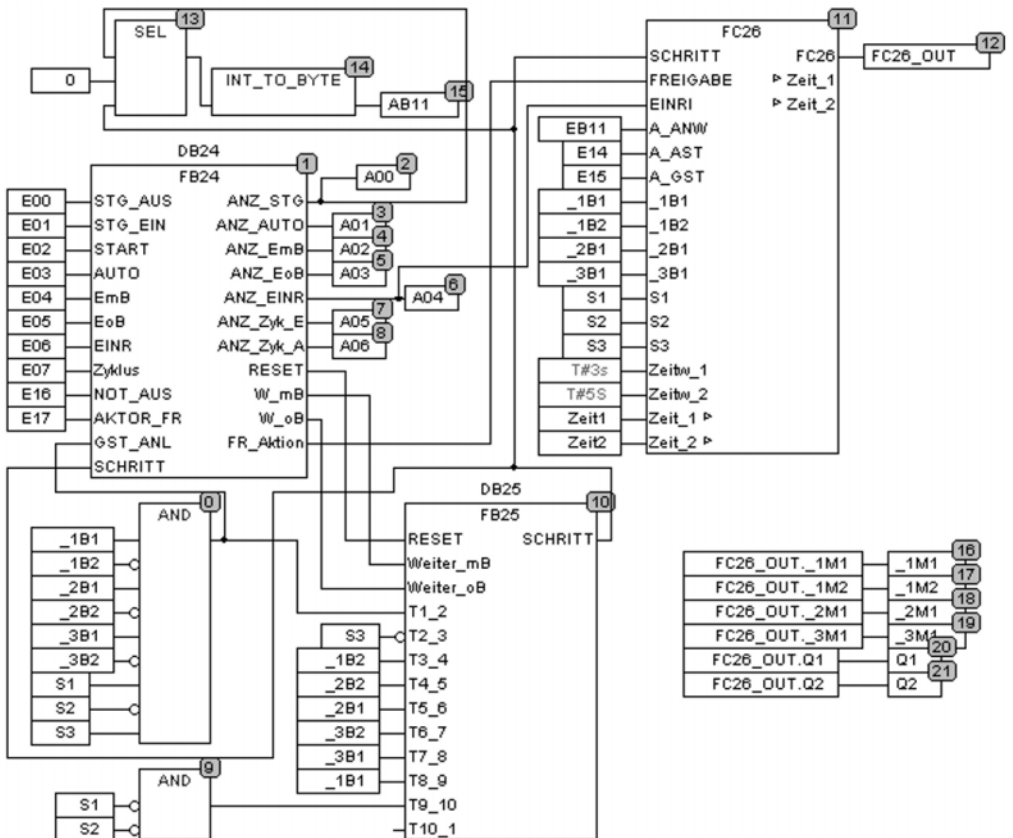


Netzwerk 6: Ausgabe-Funktion



Lösung in CoDeSys

Aufruf der Bausteine im PLC-PRG in der Programmiersprache CFC:



11.6 Komplexe Ablaufsteuerungen

11.6.1 Ablaufsteuerung mit Betriebsartenteil und Signalvorverarbeitung

Nicht immer sind die Weiterschaltbedingungen (Transitionen) der Ablaufkette nur abhängig von einzelnen Signalzuständen bestimmter Sensoren oder gestarteter Zeitglieder, wie bisher angenommen. Der allgemeinere Fall ist gekennzeichnet durch die Vorverarbeitung beliebiger Signale und deren Überprüfung auf bestimmte Kriterien oder Überwachung ihrer Ausführung. Dazu gehören beispielsweise die Verarbeitung von Zählimpulsen bei Mengenerfassungen, die Auswertung von Analogsignalen bei Temperaturmessungen, die Bildung von Weiterschaltsignalen aus komplexen Bedingungen verschachtelter Ablaufketten sowie die Bereitstellung von Daten oder Rezeptwerten. Damit kann sich schnell ein unübersichtliches Programm ergeben. In solchen Fällen ist es zweckmäßig, die Ablaufsteuerung um einen Funktionsbaustein für die Aufgaben einer Signalvorverarbeitung zu erweitern. Zur Normierung der Analogwerte können die Normierungsbausteine aus der Bausteinbibliothek verwendet werden. Die Gesamtstruktur einer Ablaufsteuerung erhält somit folgendes Aussehen:

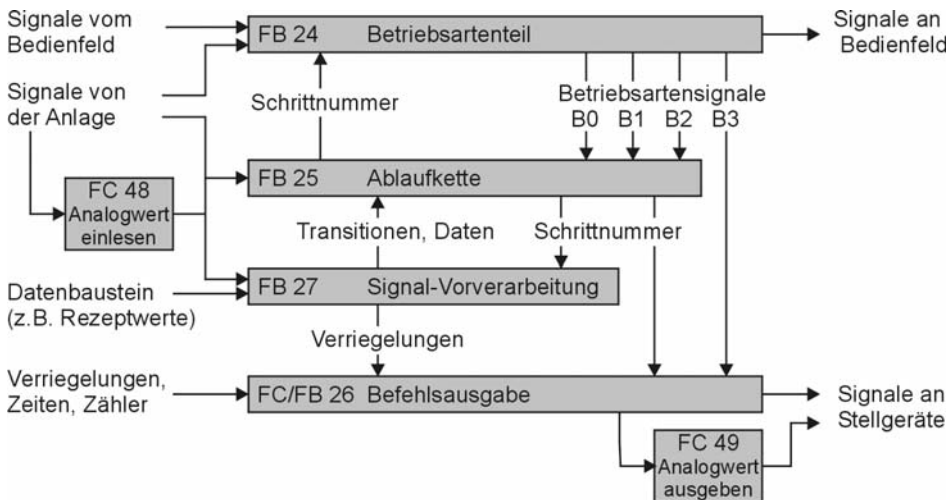


Bild 11.6: Programmstruktur einer Ablaufsteuerung mit standardisierten Funktionsbausteinen für den Betriebsartenteil (FB 24) und die Ablaufkette (FB 25) sowie den anlagenspezifischen Bausteinen für die Befehlsausgabe (FC 26/FB 26) und Signalvorverarbeitung (FB 27).

Der Signalvorverarbeitungsbaustein FB 27 ist wie der Befehlsausgabebaustein FC 26/FB 26 anlagen- bzw. prozessabhängig. Sensorsignale, die vorverarbeitet werden müssen, sind als Eingangsvariablen des Bausteins FB 27 zu deklarieren. Ergänzend zu den bereits genannten Beispielen soll auf zwei komplizierte Fälle hingewiesen werden:

1. Ist die Aufbereitung von Transitionsbedingungen und Daten von den Schritten der Ablaufkette abhängig, dann ist auch die Schrittnummer als weiterer Bausteineingang im Datenformat Integer zu deklarieren.

- Sind bei der Anlage beispielsweise Rezeptwerte aus einem Datenbaustein zu laden und als Daten der Ablaufkette zur Verfügung zu stellen, so sind auch die Datenbausteinnummer und die Rezeptnummer als zusätzliche Eingänge zu deklarieren.

Einfache Weiterschaltensignale wie z. B. das Erreichen eines bestimmten Füllstandes, das Ablauf einer Wartezeit usw. werden jedoch nicht in die Signalvorverarbeitung einbezogen und direkt als Weiterschaltensignale der Ablaufkette zugeführt.

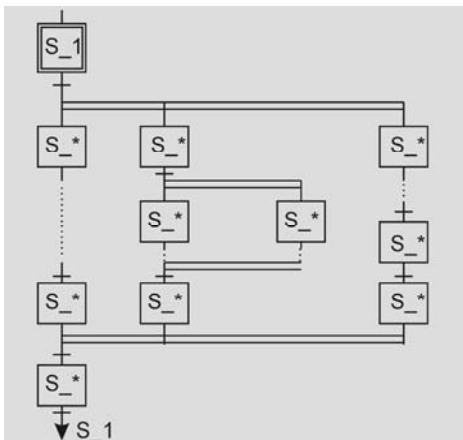
In Kapitel 11.6.4 zeigt Beispiel 11.4 eine Ablaufsteuerung mit einer erforderlichen Signalvorverarbeitung in einem dafür vorgesehenen Funktionsbaustein FB 27.

11.6.2 Ablaufsteuerungen mit korrespondierenden Ablaufketten

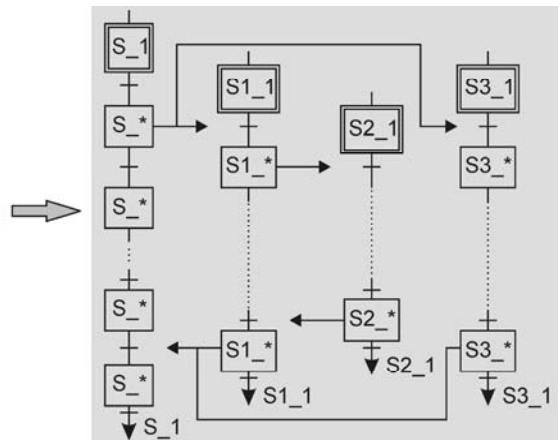
Das Beschreibungsmittel *Ablauf-Funktionsplan* eignet sich besonders gut zur Lösung von Steuerungsaufgaben für Maschinen und Anlagen, bei denen mechanische Bewegungen und zeitliche Abläufe vorrangig sind. Die funktionale Beschreibung des Steuerungsablaufs kann dabei von dem Maschinenkonstrukteur, dem Inbetriebsetzer und dem Wartungsexperten gleichermaßen verwendet werden. Damit kann von der Aufgabenstellung über die Programmierung und Inbetriebsetzung bis hin zum Service durchgängig gearbeitet werden. Dies begründet unter anderem den häufigen Einsatz dieses Beschreibungsmittels in der Praxis.

Für komplexere Maschinen oder Anlagen kann jedoch der Ablauf-Funktionsplan durch eine Vielzahl von Schritten und Verzweigungen leicht unübersichtlich werden. Ablaufpläne, die über eine große Anzahl von Schritten verfügen oder viele parallele Zweige und Sprünge besitzen, sind nur sehr schwer lesbar. Es empfiehlt sich deshalb bei solchen Steuerungsaufgaben für die Maschine oder Anlage einzelne Funktionseinheiten zu bilden und für jede dieser Funktionseinheiten einen eigenständigen Ablauf-Funktionsplan zu erstellen. Die Verbindung und Synchronisation der einzelnen Ablaufpläne übernimmt wiederum ein quasi übergeordneter Ablauf-Funktionsplan (Haupt-Ablaufkette), der den Gesamtablauf der Maschine oder Anlage beschreibt. Die Aktionen der Hauptkette bestehen dabei aus Start- oder Weiterschaltbedingungen für die Ablaufketten der einzelnen Funktionseinheiten. Die Weiterschaltbedingungen der Haupt-Ablaufkette ergeben sich aus Schritten oder Kombinationen von Schritten der einzelnen Ablaufketten.

Statt Ketten mit Verzweigungen:



Korrespondierende lineare Ketten:



Die Vorgehensweise beim Entwurf einer Ablaufsteuerung mit korrespondierenden Ablaufketten kann nach folgenden Schritten erfolgen:

- Darstellung der mechanischen und elektrischen Elemente der Anlage in Technologieschemata.
- Zerlegen der Anlage in Funktionseinheiten und Festlegen der zugehörigen Sensoren und Aktoren.
- Für jede einzelne Funktionseinheit einen Ablauf-Funktionsplan (Ablaufkette) entwerfen.
- Den übergeordneten Ablauf-Funktionsplan (Haupt-Ablaufkette) für die Koordination der einzelnen Funktionseinheiten entwerfen.

Je nach Bedarf können die beiden letzten Schritte auch vertauscht oder zeitgleich ausgeführt werden. Der Vorteil dieser Methode ist nicht nur eine übersichtlichere Darstellung des gesamten Steuerungsablaufs, sondern auch der, dass die einzelnen Ablaufketten leichter handhabbar, änderbar und bei gleichen Funktionseinheiten auch kopierbar sind.

In Kapitel 11.6.4 zeigt Beispiel 11.5 die Anwendung der beschriebenen Methode.

11.6.3 Ablaufbeschreibung für Verknüpfungssteuerungen

Steuerungen mit kombinatorischem Charakter sowie Speicher- oder Zeitverhalten, jedoch ohne zwangsläufig schrittweisen Ablauf bezeichnet man allgemein als Verknüpfungssteuerung. Da bei solchen Steuerungen der Prozess selbst keine Steuerungsstruktur vorgibt, ist es schwierig, aus der Aufgabenstellung heraus das Steuerungsprogramm zu finden.

Ziel dieses Abschnitts ist es, Steuerungsaufgaben vom Typ Verknüpfungssteuerungen in Form einer Ablaufstruktur darzustellen und diese dann in bekannter Weise in ein Steuerungsprogramm umzusetzen. Dazu werden Schritte oder besser Steuerungszustände eingeführt. Die Steuerungszustände können entweder in einer Ablaufkette oder einem Zustandsgraphen (siehe Kapitel 12) angeordnet werden. Welche der beiden Darstellungsarten jeweils ausgewählt wird, ist abhängig von der Steuerungsaufgabe und der gewählten Programmiersoftware (S7-GRAPH oder S7-HiGraph).

Der Übergang von einer mit Steuerungszuständen realisierten Verknüpfungssteuerung zu einer Ablaufsteuerung ist fließend. Hier soll gezeigt werden, wie Steuerungsaufgaben ohne zwangsläufig schrittweisen Ablauf durch die Einführung Steuerungszuständen systematisch gelöst werden können.

Folgende Regeln sind dabei zu beachten:

- Die Steuerung nimmt zu einem bestimmten Zeitpunkt einen ganz bestimmten Zustand ein.
- Mit einer Ablaufbeschreibung (Ablaufkette) werden die unterschiedlichen Zustände und Bedingungen für die Beibehaltung oder Änderung eines Zustandes angegeben.
- Von jedem Zustand kann in jeden beliebigen Zustand übergegangen werden. Je nach Übersichtlichkeit der Ablaufkette können die Übergänge auch als Sprünge dargestellt werden.
- Sind mehrere Folgezustände möglich, so sind Prioritäten einzuführen. In der Ablaufkette hat der links stehende Folgezustand Vorrang. Folgezustände sind gegenseitig zu verriegeln.
- Der Grundzustand S_1 (Initialzustand) wird beim Einschalten der Programmbearbeitung ohne Bedingung gesetzt.
- Die Umsetzung der Ablaufkette in ein Steuerungsprogramm erfolgt nach den in Kapitel 11.3 angegebenen Regeln.
- Die Anwendung eines Betriebsartenprogramms entfällt.

Die Grenzen dieser Beschreibungsmethode liegen in der Anzahl der Zustände und vor allem in der Anzahl der Verzweigungen. Haben die einzelnen Zustände jeweils sehr viele Folgezustände, so wird die Darstellung in der Ablaufkette schnell sehr unübersichtlich. Besser geeignet ist dann das Beschreibungsmittel Zustandsgraph, das in Kapitel 12 dargestellt wird. Solche Zustandsgraphen sind in der Darstellung Petri-Netzen sehr ähnlich und können mit der Programmiersoftware HiGraph realisiert werden.

Im folgenden Kapitel zeigt Beispiel 11.6 die Anwendung dieser Entwurfsmethode an der Ansteuerung einer Bedarfsampelanlage z. B. für Baustellen. Weitere Beispiele finden Sie in dem Buch „Automatisieren mit SPS, Übersichten und Übungsaufgaben“.

11.6.4 Beispiele

■ Beispiel 11.4: Mischbehälter mit wählbaren Rezeptwerten

Ein Behälter wird mit den beiden Flüssigkeiten 1 und 2 gleichzeitig gefüllt, umgerührt und auf eine bestimmte Temperatur erhitzt. Eine Rezeptur besteht aus den Sollwerten für die Menge Q1 (in %), Menge Q2 (in %), Temperatur TEMP (in °C) und Reaktionszeit T (in Sekunden). Zehn solcher Rezepturen sind in einem Datenbaustein DB hinterlegt und können mit einem einstelligen Ziffereinsteller (0 ... 9) ausgewählt werden. Zur Bedienung der Anlage soll das in Abschnitt 11.5 vorgestellte Bedien- und Anzeigefeld mit den vier wählbaren Betriebsarten verwendet werden.

Technologieschema:

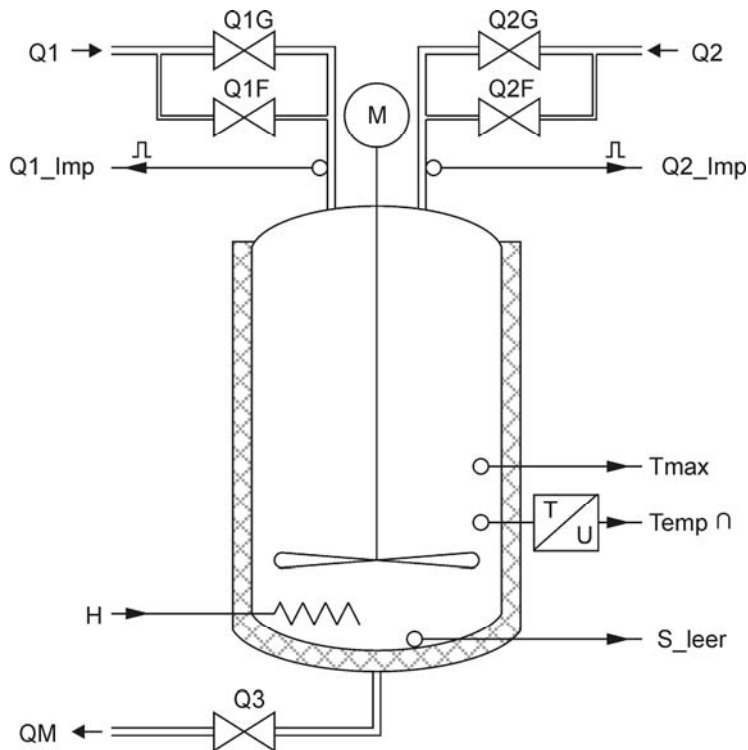


Bild 11.6: Mischbehälter

Mischprozess:

Nach dem Einschalten der Steuerung und der Betriebsartenwahl, z. B. Automatikbetrieb, sowie der Übernahme der Betriebsart mit E03, beginnt der Mischvorgang durch Betätigen der Starttaste E02.

Die durch den Rezeptureinsteller bestimmten Rezeptwerte werden aus dem Datenbaustein geladen und die beiden Grobventile Q1G sowie Q2G geöffnet.

Nach einer Wartezeit von 8 s wird das Rührwerk M eingeschaltet.

Zwei Durchflussmengengeber Q1_Imp und Q2_Imp liefern der Steuerung Impulse bis maximal 10 Hz, die abhängig von der Füllgeschwindigkeit sind. Durch Zählen der Impulse kann die jeweilige Zuflussmenge ermittelt werden. Zehn Impulse entsprechen der Füllmenge von 1 %.

Ist 90 % der Sollwertmenge Q1 (Q2) erreicht, wird das Grobventil Q1G (Q2G) geschlossen.

Sind beide Grobventile geschlossen, werden die Feinventile Q1F und Q2F geöffnet, bis jeweils die restlichen 10 % der Sollwertmengen aufgefüllt sind.

Nach Schließen der Feinventile wird die Heizung H eingeschaltet und bei Erreichen des Temperatur-Sollwertes abgeschaltet.

Nach Ablauf der zur Rezeptur zugehörigen Reaktionszeit wird das Rührwerk M ausgeschaltet und die Mischung über das Ventil Q3 abgelassen, bis die Leermeldung erfolgt.

Zuordnungstabelle der Eingänge und Ausgänge des Mischbehälters:

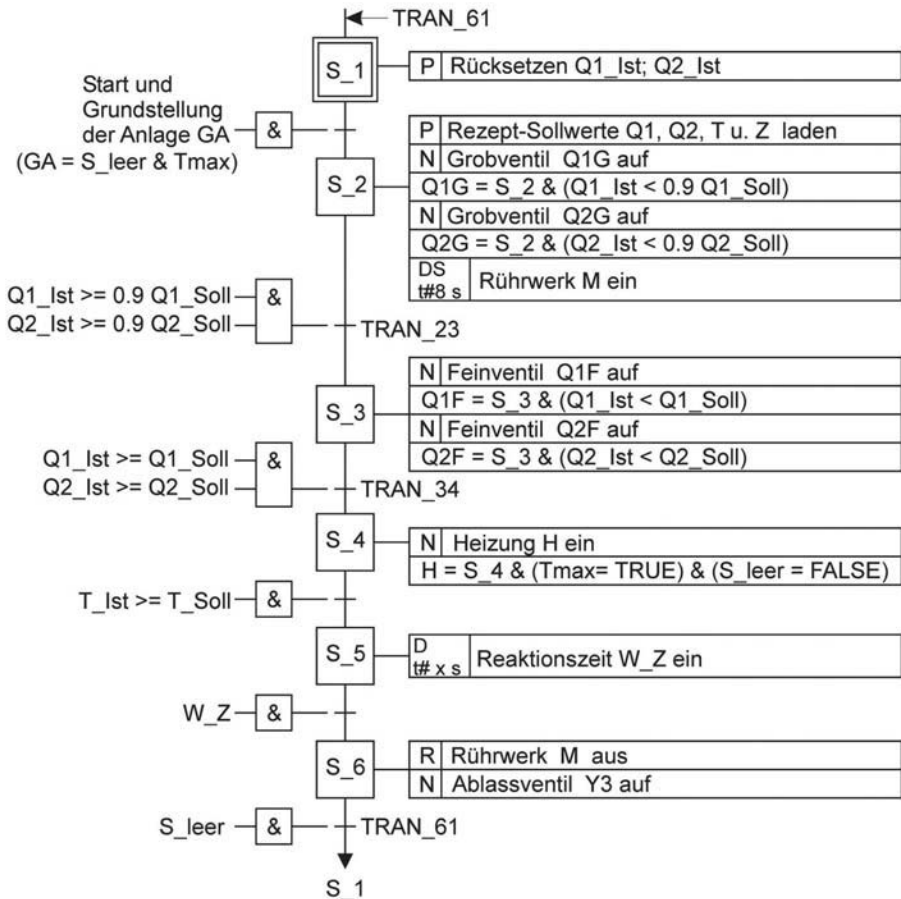
Eingangsvariable	Symbol	Datentyp	Logische Zuordnung		Adresse
Durchflussmengengeber 1	Q1_Imp	BOOL	Impulse		E 0.1
Durchflussmengengeber 2	Q2_Imp	BOOL	Impulse		E 0.2
Temperaturgrenzwert	Tmax	BOOL	Erreicht	Tmax = 0	E 0.3
Leermeldung/Behälter	S_leer	BOOL	Behälter leer	S_leer = 1	E 0.4
Mischtemperatur	Temp	WORD	Analogspannung	0 ... 10 V	PEW 320
Rezeptureinsteller	ZE	½ Byte	Eine Stelle BCD	0 ... 9	EB 8
Ausgangsvariable					
Grobventil 1	Q1G	BOOL	Offen	Q1G = 1	A 4.1
Grobventil 2	Q2G	BOOL	Offen	Q2G = 1	A 4.2
Feinventil 1	Q1F	BOOL	Offen	Q1F = 1	A 4.3
Feinventil 2	Q2F	BOOL	Offen	Q2F = 1	A 4.4
Ablassventil	Q3	BOOL	Offen	Q3 = 1	A 4.5
Rührwerk	M	BOOL	Motor läuft	M = 1	A 4.6
Heizung	H	BOOL	Heizung ein	H = 1	A 4.7

Die Zuordnungstabelle des Bedienfeldes ist im Beispiel 11.3 (Biegemaschine) angegeben.

Bei STEP 7 befinden sich die Rezepturen in einem Datenbaustein (z. B. DB 10). Die Eingabe der Werte für die 10 Rezepturen in den Datenbaustein ist nicht Gegenstand dieser Aufgabe.

Bei CoDeSys werden die Werte der 10 Rezepturen in ein Datenfeld geschrieben. Dazu wird ein eigener Datentyp mit der Struktur eines Rezeptes (Q1: INT, Q2: INT, TEMP: REAL und ZEIT: TIME) angelegt. Im PLC-PRG wird dann die Variable „REZEPTE“ als ARRAY mit den entsprechenden Rezeptwerten als Initialisierung deklariert.

Die Rezeptwerte Menge Q1 und Menge Q2 müssen als Integer-Werte im Bereich von 0 ... 100 (%) angegeben werden. Beide Werte zusammen dürfen bei der Rezeptwerteingabe 100% nicht überschreiten. Die Werte sind im Signalvorverarbeitungs-Baustein bei der Übernahme als aktueller Sollwert auf die anlagen-spezifische Anzahl von Impulsen für 100 % umzurechnen. Bei der hier verwendeten Modellanlage wären dies 1000 Impulse für 100 %. Die Rezeptwerte für Q1 und Q2 müssen demnach mit 10 multipliziert werden.

Ablaufkette:

Die Transitionsbedingungen TRAN_23 und TRAN_34 sind Weberschaltensignale, welche sich aus der Zählung der Impulse und einem Vergleich mit den Sollwerten ergeben. Die dort stehenden Bedingungen „ $Q1_Ist \geq 0,9Q1_Soll$ “ bis „ $Q2_Ist \geq Q2_Soll$ “ werden in einem Signalvorverarbeitungsbaustein FB 27 gebildet. Weitere Aufgaben des Bausteins FB 27 sind die Zählung der Impulse zur Ermittlung der Istmengen Q1 und Q2 und das Zurücksetzen der Istwerte im Initialschritt Schritt 1. Das Laden der Sollwerte aus dem Datenbaustein entsprechend der gewählten Rezeptur wird ebenfalls in diesem Baustein ausgeführt.

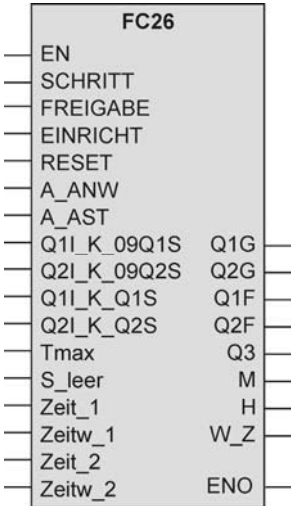
Die Auswertung des Analogsignals des Temperatursensors der Mischanlage erfolgt durch den Bibliotheksbaustein FC 48 Analogwert einlesen. Mit diesem wird die Normierung der Werte 10 °C bis 110 °C ($0\text{ V} \dots 10\text{ V}$) des Sensors in steuerungsinterne Gleitpunktzahlen von 10.0 bis 110.0 ausgeführt.

Für die Steuerungsaufgabe des Mischbehälters muss der Signalvorverarbeitungsbaustein FB 27 neu entworfen werden. Der Ausgabebaustein FC 26 ist an die Gegebenheiten der Anlage anzupassen. Die weiteren erforderlichen Bausteine FB 24 Betriebsarten, FB 25 Schrittkette und FC 48 Analogwert einlesen können aus der Bibliothek unverändert übernommen werden.

Nachfolgend sind deshalb nur die Programme für den Ausgabebaustein FC 26 und für den Signalvorverarbeitungsbaustein FB 27 dargestellt.

Ausgabebaustein FC 26

Übergabevariablen der Ausgabefunktion FC 26:



Die Standard-Eingangsvariablen des Bausteins werden erweitert durch Eingänge für die Verriegelungsbedingungen:

Q1I_K_09Q1S für $Q1_Ist < 0,9 Q1_Soll$,

Q2I_K_09Q2S für $Q2_Ist < 0,9 Q2_Soll$,

Q1I_K_Q1S für $Q1_Ist < Q1_Soll$ und

Q2I_K_Q2S für $Q2_Ist < Q2_Soll$.

An diese Eingänge sind die entsprechenden Bedingungen $Q1Ist \leq 0,9 Q1Soll$ ($Q1Ist_GL_09Q1Soll$) etc. des Signalvorverarbeitungsbausteins FB 27 **negiert** anzulegen. Damit werden die Einlassventile beim Erreichen des jeweiligen Sollwertes geschlossen.

Weitere Eingangsvariablen für Verriegelungen sind der Temperaturgrenzwertgeber Tmax und das Sensorsignal S_leer.

Die Ausgangsvariablen des Bausteins werden durch die Aktoren der Mischanlage und der Reaktionszeit W_Z als Transitionsbedingung gebildet.

Hinweis: Bei CoDeSys werden die Standard-Zeitfunktionsblöcke TON zur Bildung der Zeit benutzt und die Instanzen der Funktionsbausteine als IN_OUT-Variablen deklariert.

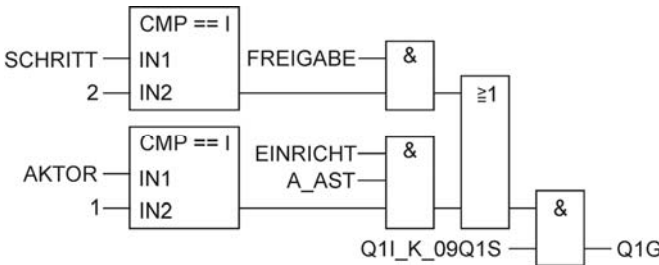
Steuerungsprogramm:

Maskierung des 1/2-Bytes zur Aktorauswahl:

```
L #A_ANW
L W#16#F
UW
T #AKTOR
```

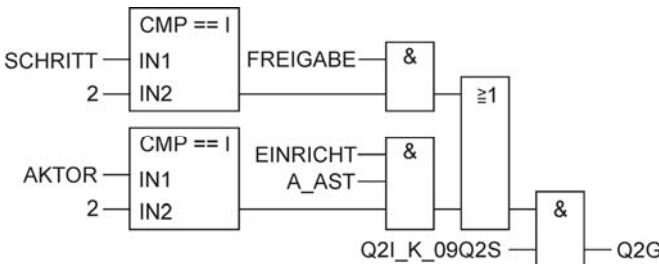
Damit die übrigen Binärstellen des Bytes noch verwendet werden können, sind die ersten 4 Bits der Eingangsvariablen A_ANW mit W#16#000F maskiert und der lokalen Variablen AKTOR zugewiesen.

Ansteuerung Grobventil Q1G:



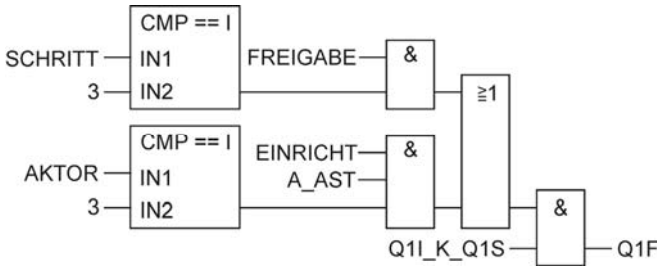
Grobventil Q1G ist Aktor Nr. 1
 Im Einrichtbetrieb kann das Ventil Q1G ohne Bedingungen geöffnet werden.
 In allen Betriebsarten wird beim Erreichen von 90 % des Sollwertes Q1 das Ventil geschlossen.

Ansteuerung Grobventil Q2G:



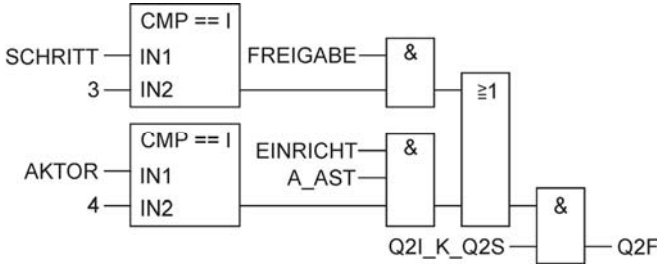
Grobventil Q2G ist Aktor Nr. 2
 Im Einrichtbetrieb kann das Ventil Q2G ohne Bedingungen geöffnet werden.
 In allen Betriebsarten wird beim Erreichen von 90 % des Sollwertes Q2 das Ventil geschlossen.

Ansteuerung Feinventil Q1F:



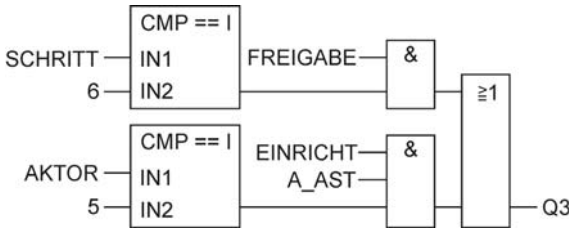
Feinventil Q1F ist Aktor Nr. 3
 Im Einrichtbetrieb kann das Ventil Q1F ohne Bedingungen geöffnet werden.
 In allen Betriebsarten wird beim Erreichen des Sollwertes Q1 das Ventil geschlossen.

Ansteuerung Feinventil Y1G:



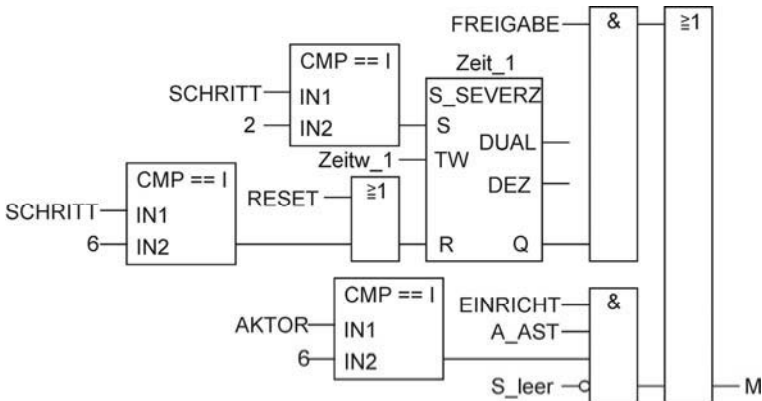
Feinventil Q2F ist Aktor Nr. 4
 Im Einrichtbetrieb kann das Ventil Q2F ohne Bedingungen geöffnet werden.
 In allen Betriebsarten wird beim Erreichen des Sollwertes Q2 das Ventil geschlossen.

Ansteuerung Ablassventil Q3:



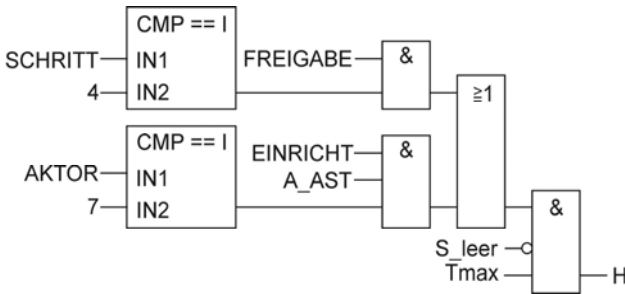
Ablassventil Q3 ist Aktor Nr. 5
 Im Einrichtbetrieb kann das Ventil Q3 ohne Bedingungen geöffnet werden.

Ansteuerung Rührwerk M:



Rührwerk M ist Aktor Nr. 6
 Im Einrichtbetrieb darf das Rührwerk nur laufen, wenn der Behälter nicht leer ist.

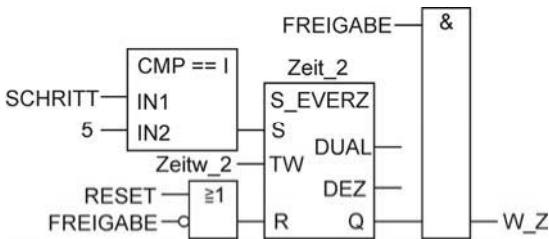
Ansteuerung Heizung H:



Heizung H ist Aktor Nr. 7

In allen Betriebsarten kann die Heizung H nur eingeschaltet werden, wenn der Behälter nicht leer ist und die Grenztemperatur nicht erreicht ist.

Bildung der Reaktionszeit:



Bei der Bildung der Reaktionszeit muss der Einrichtbetrieb nicht berücksichtigt werden. Deshalb entfällt die entsprechende Ansteuerung.

Deklarationstabelle:

Name	Datentyp
IN	
SCHRITT	Int
FREIGABE	Bool
EINRICHT	Bool
RESET	Bool
A_ANW	Byte
A_AST	Bool
Q1I_K_09Q1S	Bool
Q2I_K_09Q2S	Bool
Q1I_K_Q1S	Bool
Q2I_K_Q2S	Bool
Tmax	Bool
S_leer	Bool
Zeit 1	Timer
Zeitw 1	S5Time
Zeit 2	Timer
Zeitw 2	S5Time

Name	Datentyp
OUT	
Q1G	Bool
Q2G	Bool
Q1F	Bool
Q2F	Bool
Q3	Bool
M	Bool
H	Bool
W Z	Bool
TEMP	
Aktor	Int

Durch die bei STEP 7 und CodeSys unterschiedliche Speicherung der Rezeptwerte unterscheiden sich auch die Programme und die Übergabevariablen des Signalvorverarbeitungs-Bausteins FB 27.

Der weitere Lösungsweg wird deshalb auf den folgenden Seiten für STEP 7 und CoDeSys getrennt beschrieben.

Lösung in STEP 7

Signalvorverarbeitungs-Baustein FB 27:

Übergabevariablen:



Beschreibung der Übergabeparameter:

DBNR (INT): Angabe der Datenbausteinnummer, der die Rezeptwerte enthält**RZNR (BYTE):** Anschluss für den einstelligen Zifferneinsteller**SCHRITT (INT):** Angabe der aktuellen Schrittnummer der Ablaufkette**TEMP_Soll (REAL) u. Zeit_Soll (S5Time):** Ausgabe des Temperatur- bzw. ZeitsollwertesAlle anderen Ein- und Ausgangsvariablen sind vom Datentyp **BOOL**.

Steuerungsprogramm:

Rücksetzen der Istwerte Q1 und Q2 im Schritt 1 (AWL):

```

L #SCHRITT
L 1
==I
FP #FO1
SPBN M001
L 0
T #Q1_Ist
T #Q2_Ist
M001: NOP 0

```

Das Bestimmungszeichen P in der Aktion des 1. Schrittes gibt die flankengesteuerte Löschung der Ist-Mengenähler Q1_Ist und Q2_Ist vor.

Nur beim Eintritt in den Schritt 1 werden die Zähler gelöscht. Wird beispielsweise im Einrichtbetrieb ein Zuluftventil geöffnet, ist somit gewährleistet, dass die zugelaufene Menge in den entsprechenden Zählern Q1_Ist bzw. Q2_Ist auch in Schritt 1 erfasst werden.

Laden der Sollwerte aus dem Datenbaustein (AWL):

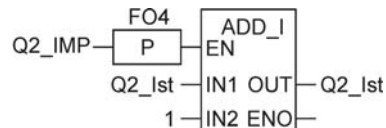
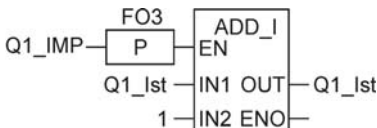
```

L #SCHRITT          L DBW [#ZEIG]      SRW 3
L 2                 L 10                L 2
==I                 *I                    +I
FP #FO2             T #Q1_Soll           SLW 3
SPBN M002           L #ZEIG              L #ZEIG
L #DBNR             SRW 3                L DBD [#ZEIG]
T #DBNRW            L 2                  T #Temp_Soll
AUF DB[#DBNRW]     +I                    L #ZEIG
L #RZNR             SLW 3                SRW 3
L B#16#F            T #ZEIG              L 4
UW                  L DBW [#ZEIG]       +I
L 10                L 10                SLW 3
*I                  *I                    T #ZEIG
SLW 3               T #Q2_Soll           L DBW [#ZEIG]
T #ZEIG             L #ZEIG              T #Zeit_Soll
M002: NOP 0

```

Die jeweils erste Adresse einer Rezeptur (Menge Q1) ergibt sich aus der Multiplikation der Rezeptnummer RZNR mit dem Faktor 10 (siehe nachfolgenden Datenbaustein DB 10). Mit der lokalen Variable ZEIG werden die einzelnen Rezeptwerte indirekt adressiert. Da 10 Impulse der Menge von 1 % entsprechen, werden die Rezeptwerte Q1 und Q2 je mit 10 multipliziert.

Istwertmengenähler Q1_Ist und Q2_Ist:



Berechnung der 90 % von Q1_Soll bzw. Q2_Soll (AWL):

```

L #Q1_Soll      L #Q2_Soll
L 9             L 9
*I             *I
L 10           L 10
/I            /I
T #_9Q1Soll    T #_9Q2Soll
    
```

Zur Berechnung des 90%-Wertes von Q1_Soll wird der Wert mit 9 multipliziert, dann durch 10 dividiert und der lokalen Variablen _9Q1Soll zugewiesen. Für _9Q2Soll gilt Entsprechendes.

Bildung der Bedingungen Q1_Ist >= 0.9 Q1_Soll und Q2_Ist >= 0.9 Q2_Soll:



Bildung der Bedingungen Q1_Ist >= Q1_Soll und Q2_Ist >= Q2_Soll:



Deklarationstabelle FB 27:

Name	Datentyp	A -Wert
IN		
DBNR	Int	0
RZNR	Byte	B#16#0
Q1_IMP	Bool	FALSE
Q2_IMP	Bool	FALSE
SCHRITT	INT	0
OUT		
Q1Ist_GL_09Q1Soll	Bool	FALSE
Q2Ist_GL_09Q2Soll	Bool	FALSE
Q1Ist_GL_Q1Soll	Bool	FALSE
Q2Ist_GL_Q2Soll	Bool	FALSE
Temp_Soll	REAL	0.0
Zeit_Soll	S5Time	S5T#0S

Name	Datentyp	A.-Wert
STAT		
Q1_Ist	Int	0
Q2_Ist	Int	0
Q1_Soll	Int	0
Q2_Soll	Int	0
FO1	Bool	FALSE
FO2	Bool	FALSE
FO3	Bool	FALSE
FO4	Bool	FALSE
TEMP		
DBNRW	WORD	
ZEIG	DWORD	
_9Q1Soll	Int	
_9Q2Soll	Int	

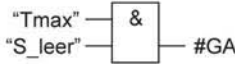
DB 10: Rezeptwertspeicher

Die zehn Rezepturen mit den jeweils vier Rezeptwerten für eine Mischung stehen im Datenbaustein DB10 und können dort von Hand oder von einem Bedien- und Beobachtungssystem über eine Kommunikationsverbindung auftragsbezogen geändert werden.

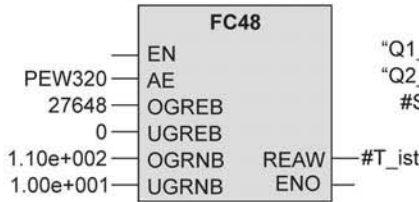
Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	Menge_Q1_0	INT	30	30%
+2.0	Menge_Q2_0	INT	40	40%
+4.0	Temperatur_0	REAL	5.000e+001	50.0°C
+8.0	Reaktionszeit_0	S5TIME	S5T#10S	10 s
+10.0	Menge_Q1_1	INT	31	31%
...
+90.0	Menge_Q1_9	INT	39	39%
+92.0	Menge_Q2_9	INT	49	49%
+94.0	Temperatur_9	REAL	39.000e+001	59.0°C
+98.0	Reaktionszeit_9	S5TIME	S5T#19S	19 s
=100.0		END_STRUCT		

Funktionsplan Organisationsbaustein OB 1:

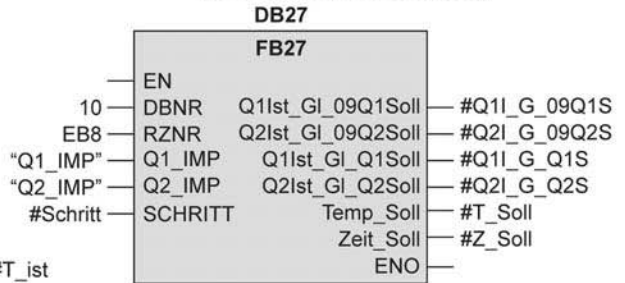
NW1: Grundstellung Anlage



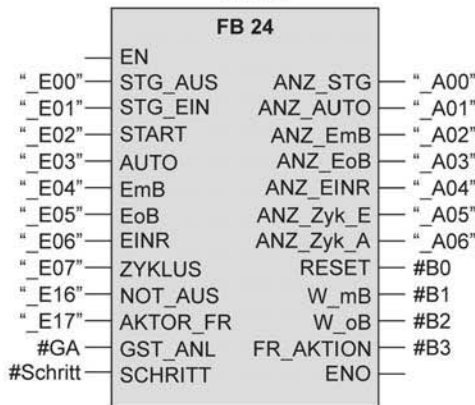
NW3: Analogwert einlesen



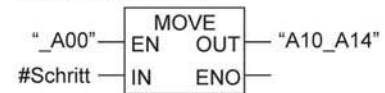
NW2: Signalvorverarbeitung



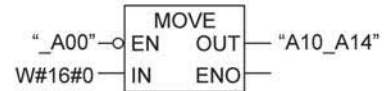
NW4: Betriebsarten DB 24



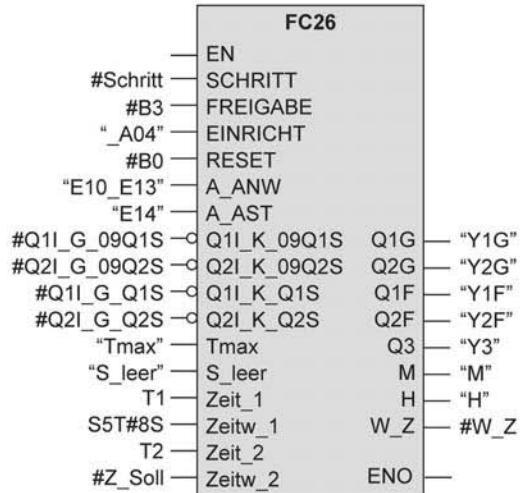
NW6: Schrittanzeige



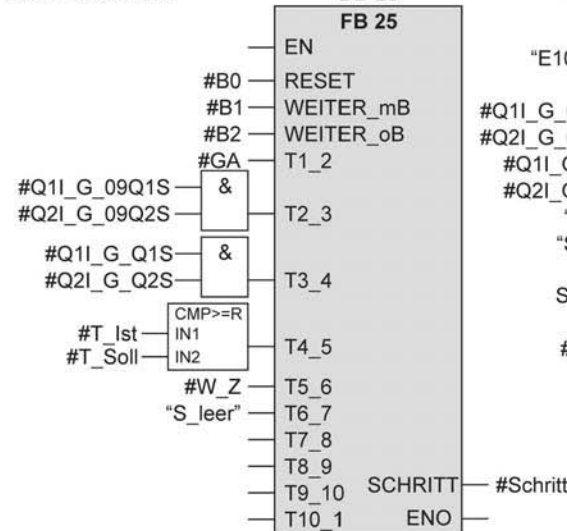
NW7: Schrittanzeige



NW8: Befehlsausgabe



NW5: Abaufkette



Hinweis: Die Reihenfolge der Netzwerke ist unbedingt einzuhalten (Aufruf FB 27 vor FB 25).

Lösung in CoDeSys

Signalvorverarbeitungs-Baustein FB 27:

Übergabevariablen:



Beschreibung der Übergabeparameter:

REZEPT (ARRAY[0..9]OF REZW): Übergabe der Feldvariablen, welche die Rezeptwerte enthält

RZNR (INT): Vorgabe der gewünschten Rezeptnummer

SCHRITT (INT): Angabe der aktuellen Schrittnummer der Ablaufkette

TEMP_Soll (REAL) und Zeit_Soll (TIME): Ausgabe des Temperatur- bzw. Zeitsollwertes

Alle anderen Ein- und Ausgangsvariablen haben den Datentyp **BOOL**.

Datentyp REZW und Programm des Funktionsbausteins FB 27 in AWL:

```

TYPE REZW :
STRUCT
  Q1, Q2:INT;
  TEMP:REAL;
  ZEIT:TIME;
END_STRUCT
END_TYPE

FUNCTION_BLOCK FB27
VAR_INPUT
  REZEPTE:ARRAY [0..9]
    OF REZW;
  RZNR:INT;
  Q1_IMP, Q2_IMP:BOOL;
  SCHRITT:INT;
END_VAR
VAR_OUTPUT
  Q1Ist_GL_09Q1Soll:BOOL;
  Q2Ist_GL_09Q2Soll:BOOL;
  Q1Ist_GL_Q1Soll:BOOL;
  Q2Ist_GL_Q2Soll:BOOL;
  Temp_Soll:REAL;
  Zeit_Soll:TIME;
END_VAR
VAR
  FO1, FO2: R_TRIG;
  FO3, FO4: R_TRIG;
  Q1_Ist,Q2_Ist: INT;
  Q1_Soll, Q2_Soll: INT;
  _9Q1Soll, _9Q2Soll: INT;
  _INT_0:INT;
END_VAR

//Anweisungsteil
ADD 1
STRUCT
  LD SCHRITT
  EQ 1
  ST FO1.CLK
  CAL FO1
  LD FO1.Q
  NOT
  JMP M001
  LD 0
  ST Q1_Ist
  ST Q2_Ist
  M001:
  LD SCHRITT
  EQ 2
  ST FO2.CLK
  CAL FO2
  LD FO2.Q
  NOT
  JMP M002
  LD Rezepte[RZNR].Q1
  MUL 10
  ST Q1_Soll
  LD Rezepte[RZNR].Q2
  MUL 10
  ST Q2_Soll
  LD Rezepte[RZNR].TEMP
  ST Temp_Soll
  LD Rezepte[RZNR].ZEIT
  ST Zeit_Soll
  M002:
  CAL FO3 (CLK := Q1_IMP)
  LD Q1_Ist
  ST _INT_0
  LD FO3.Q
  SEL Q1_Ist,_INT_0
  ST Q1_Ist
  CAL FO4 (CLK := Q2_IMP)
  LD Q2_Ist
  ADD 1
  ST _INT_0
  LD FO4.Q
  SEL Q2_Ist,_INT_0
  ST Q2_Ist
  LD Q1_Soll
  MUL 9
  DIV 10
  ST _9Q1Soll
  LD Q2_Soll
  MUL 9
  DIV 10
  ST _9Q2Soll
  LD Q1_Ist
  GE _9Q1Soll
  ST Q1Ist_GL_09Q1Soll
  LD Q2_Ist
  GE _9Q2Soll
  ST Q2Ist_GL_09Q2Soll
  LD Q1_Soll
  ST Q1Ist_GL_Q1Soll
  LD Q2_Soll
  ST Q2Ist_GL_Q2Soll

```

Rezeptwertspeicher

Die zehn Rezepturen mit den jeweils vier Rezeptwerten für eine Mischung stehen in einem im PLC-PRG deklarierten Datenfeld mit Initialwerten.

Beispiel für die Initialisierung der ersten beiden Rezeptwerte:

```

VAR REZEPTE:ARRAY[1..10] OF REZW :=
  (Q1:=30,Q2:=40,TEMP:=50.0,ZEIT:=T#10S),
  (Q1:=31,Q2:=41,TEMP:=51.0,ZEIT:=T#11S);  END_VAR

```

Der Aufruf aller Bausteine und deren Verschaltung im PLC_Programm entspricht dem STEP 7 Programm im OB 1. Auf eine Darstellung wird deshalb verzichtet und auf den Download der Programme auf der WEB-Seite www.automatisieren-mit-sps.de verwiesen.

■ Beispiel 11.5: Chargenprozess

Zur Herstellung eines bestimmten Kunststoffes arbeiten zwei Vorlagebehälter für Einsatzstoffe mit einem Mischkessel zusammen. In den Vorlagebehältern erfolgt das Dosieren und Aufheizen der Rohprodukte. Je zwei Füllungen von Vorlagebehälter 1 und Vorlagebehälter 2 werden im Mischkessel gesammelt, auf die Reaktionstemperatur gebracht. Nach einer vorgegebenen Mischzeit ist das gewünschte Produkt fertig und kann in den Wertproduktbehälter abgelassen werden.

Technologieschema:

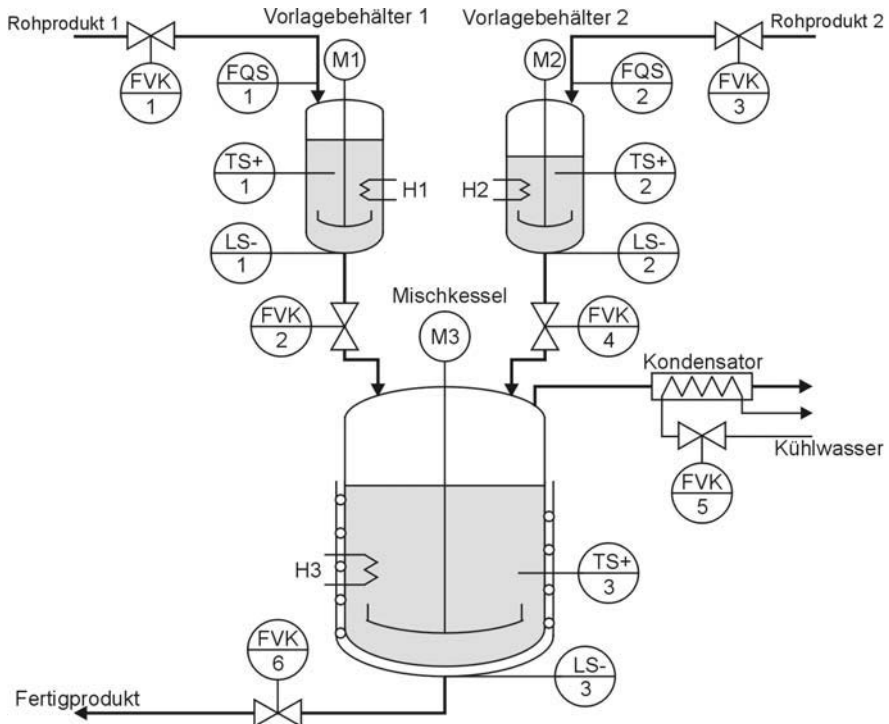


Bild 11.7: Chargenprozess (Legende: FVK = binäres Stellventil (F = Durchfluss, V = Stellgerätfunktion, K = binär); FQS = Durchflussmengenähler (FQ = Durchflussmenge, S = Frequenz); TS = Temperatursensor (T = Temperatur, S = Schaltung); LS = Standmessung (L = Stand, S = Schaltung))

Funktionsablauf:

Vorlagebehälter

Die zwei Vorlagebehälter beginnen ihren Betrieb mit dem Prozessstart. Da in den Behältern der gleiche Prozessablauf stattfindet, gilt die Beschreibung für beide.

Zunächst wird das Einlassventil FVK1 für die Zufuhr des Einsatzstoffes geöffnet. Meldet der Dosierzähler FQS1 das Erreichen der eingestellten Dosiermenge, wird das Einlassventil wieder geschlossen, die Heizung H1 und das Rührwerk M1 eingeschaltet. Erreicht die Temperatur im Vorlagebehälter den eingestellten Wert (Temperatursensor TS+1 = TRUE), dann ist der Stoff für den Mischkessel aufbereitet.

Erst wenn in beiden Vorlagebehältern die Füllungen fertig sind, werden die Auslassventile FVK3 bzw. FVK4 geöffnet. Bis dahin soll bei dem zuerst fertigen Vorlagebehälter die Heizung abgeschaltet, das Rührwerk jedoch weiterlaufen. Während des Entleervorgangs wird das Rührwerk ausgeschaltet. Bei leerem Behälter (LS1 = 1 bzw. LS2 = 1) ist das Auslassventil wieder zu schließen. Sind beide Vorlagebehälter vollständig entleert, kann mit der zweiten Füllung begonnen werden.

Mischkesselbetrieb

Während der ersten Befüllung des Mischkessels wird das Rührwerk (M3) eingeschaltet. Ist die Befüllung beendet, wird die Heizung H3 eingeschaltet. Nach Erreichen der erforderlichen Temperatur im Mischkessel (TS3 = 1) wird das Kühlwasserventil FVK5 des Kondensators geöffnet und die zweite Befüllung mit den fertigen Stoffen der Vorlagebehälter kann beginnen. Ist die zweite Befüllung vollständig durchgeführt, muss die gesamte Mischung bei eingeschalteter Heizung H noch 25 Sekunden gerührt werden. Danach ist das Produkt fertig. Das Rührwerk M3, die Heizung H, der Kühlkreislauf FVK5 sind abzuschalten und das Auslassventil FVK6 zu öffnen. Ist der Mischkessel vollständig entleert, kann der Chargenprozess wiederholt werden.

Die Vorlagebehälter und der Mischkessel sollen mit dem in Abschnitt 11.5 vorgestellten Bedienfeld und dem dazugehörigen Betriebsartenteil betrieben werden.

Zuordnungstabelle der Eingänge und Ausgänge:

Eingangsvariable	Symbol	Datentyp	Logische Zuordnung		Adresse
Vorlagebehälter 1					
Behälter leer	LS1	BOOL	Behälter leer	LS1 = 1	E 0.0
Dosierzähler	FQS1	BOOL	Menge erreicht	FQS1 = 1	E 0.1
Temperatursensor	TS1	BOOL	Temperatur erreicht	TS1 = 1	E 0.2
Vorlagebehälter 2					
Behälter leer	LS2	BOOL	Behälter leer	LS2 = 1	E 0.3
Dosierzähler	FQS2	BOOL	Menge erreicht	FQS2 = 1	E 0.4
Temperatursensor	TS2	BOOL	Temperatur erreicht	TS2 = 1	E 0.5
Mischkessel					
Behälter leer	LS3	BOOL	Behälter leer	LS3 = 1	E 0.6
Temperatursensor	TS3	BOOL	Temperatur erreicht	TS3 = 1	E 0.7
Ausgangsvariable					
Vorlagebehälter 1					
Einlassventil	FVK1	BOOL	Ventil geöffnet	FVK1 = 1	A 4.0
Heizung	H1	BOOL	Heizung an	H1 = 1	A 4.1
Rührwerkmotor	M1	BOOL	Rührwerk an	M1 = 1	A 4.2
Auslassventil	FVK2	BOOL	Ventil geöffnet	FVK2 = 1	A 4.3
Vorlagebehälter 2					
Einlassventil	FVK3	BOOL	Ventil geöffnet	FVK3 = 1	A 4.4
Heizung	H2	BOOL	Heizung an	H2 = 1	A 4.5
Rührwerkmotor	M2	BOOL	Rührwerk an	M2 = 1	A 4.6
Auslassventil	FVK4	BOOL	Ventil geöffnet	FVK4 = 1	A 4.7
Mischkessel					
Heizung	H3	BOOL	Heizung an	H3 = 1	A 5.0
Rührwerkmotor	M3	BOOL	Rührwerk an	M3 = 1	A 5.1
Kühlwasserventil	FVK5	BOOL	Kreislauf offen	FVK5 = 1	A 5.2
Auslassventil	FVK6	BOOL	Ventil geöffnet	FVK6 = 1	A 5.3

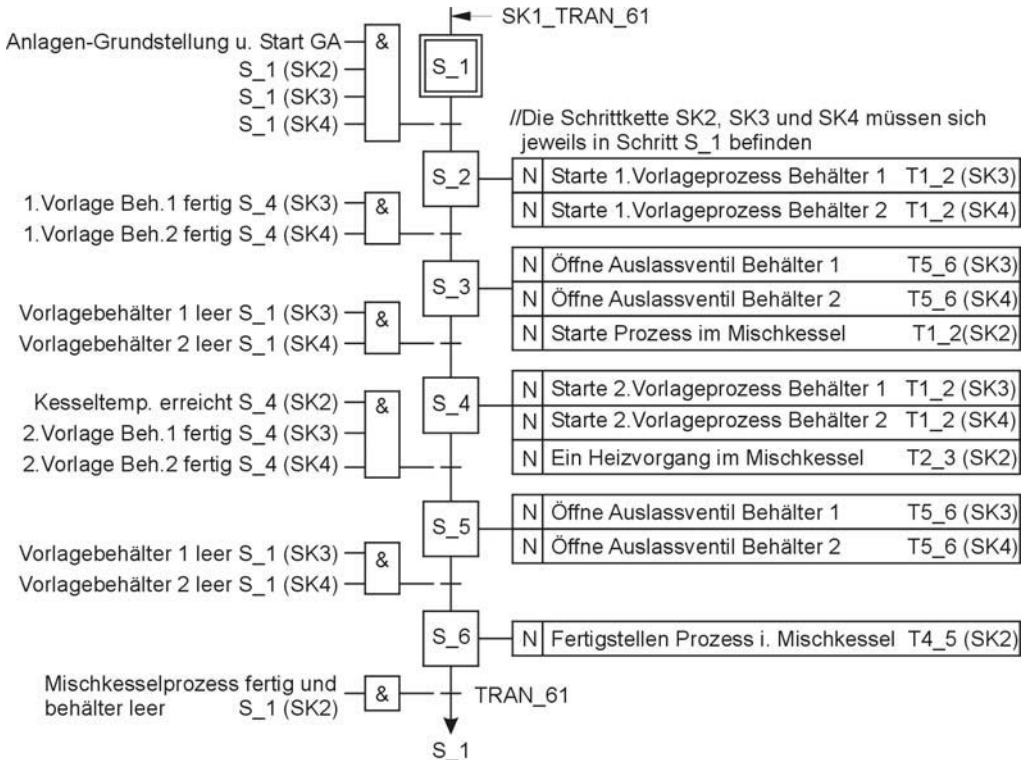
Die Zuordnungstabelle des Bedien- und Anzeigefeldes siehe Abschnitt 11.5.

Lösung

Der Chargenprozess kann in drei Funktionseinheiten Mischkessel, Vorlagebehälter 1 und Vorlagebehälter 2 gegliedert werden. Jede dieser Funktionseinheiten wird mit einer eigenen Ablaufkette gesteuert. Die Ablaufkette für den Vorlagebehälter 1 (SK3) und die Ablaufkette für den Vorlagebehälter 2 (SK4) unterscheiden sich dabei nur durch die zum jeweiligen Behälter gehörenden Sensoren und Aktoren. Die Koordination der drei Ablaufketten und somit die Steuerung des Gesamtablaufs übernimmt die Haupt-Ablaufkette (SK1).

Der Entwurf der vier Ablaufketten erfolgt aus der Funktionsbeschreibung am günstigsten zeitgleich, da Transitionen und Aktionen der einzelnen Ketten miteinander korrespondieren. Nachfolgend sind die einzelnen Ablaufketten nacheinander in der Reihenfolge: Haupt-Ablaufkette (SK1), Ablaufkette für den Mischkesselbetrieb (SK2) und Ablaufkette für den Vorlagebehälter 1 (SK3) dargestellt. Auf die Darstellung der Ablaufkette für den Vorlagebehälter 2 (SK4) wird verzichtet, da diese bis auf die entsprechenden Bezeichnungen für die Sensoren und Aktoren der Ablaufkette (SK3) für den Vorlagebehälter 1 entspricht.

Haupt-Ablaufkette SK1:



Bis auf die erste Weiterschaltbedingung bestehen alle Transitionen der Haupt-Ablaufkette aus der Abfrage von aktiven Schritten der Schrittketten SK2, SK3 und SK4. Dabei bedeutet beispielsweise S_5 (SK3), dass Schritt 5 der Schrittkette 3 aktiv sein muss, damit die Bedingung erfüllt ist.

Die Aktionen der Koordinations-Schrittkette steuern keine Stellglieder an, sondern bilden Bedingungen für die Transitionen der Schrittketten SK2, SK3 und SK4.

Die Bezeichnungen haben dabei folgende Bedeutung:

T1_2 (SK3) ist die Weiterschaltbedingung von Schritt 1 nach Schritt 2 der Schrittkette 3.

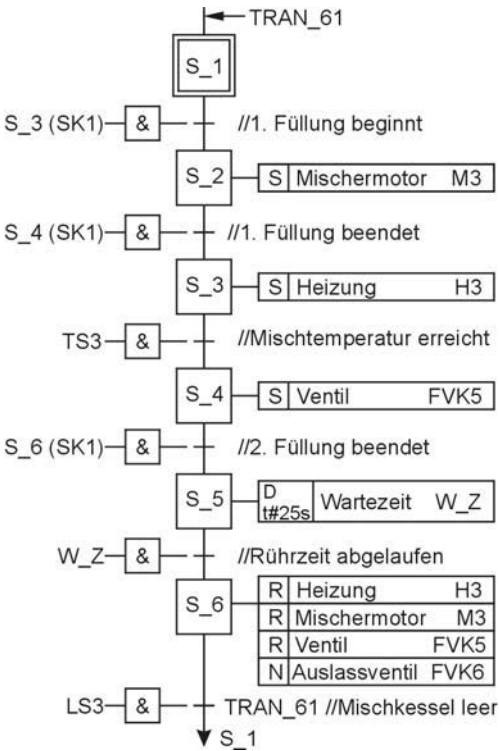
T1_2 (SK4) ist die Weiterschaltbedingung von Schritt 1 nach Schritt 2 der Schrittkette 4.

T5_6 (SK3) ist die Weiterschaltbedingung von Schritt 5 nach Schritt 6 der Schrittkette 3.

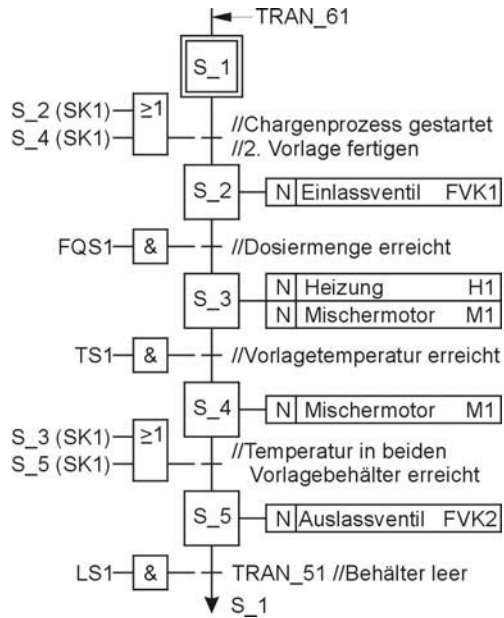
T5_6 (SK4) ist die Weiterschaltbedingung von Schritt 5 nach Schritt 6 der Schrittkette 4.

Ablaufketten für die Funktionseinheiten Mischkessel und Vorlagebehälter:

Mischkessel SK2:



Ablaufkette Vorlagebehälter SK3 (SK4):



Für die Umsetzung der vier Ablaufketten in ein Steuerungsprogramm wird der Bibliotheks-Funktionsbaustein FB 25 mit unterschiedlichen Instanzen für jede Ablaufkette aufgerufen und parametrisiert.

Die Bedingungen aus dem Betriebsartenteil werden durch den Aufruf des Betriebsarten-Funktionsbausteins FB 24 gebildet. Im Einzelschrittbetrieb ohne Bedingungen erfolgt die Schrittweitschaltung bei allen vier Ablaufketten gleichzeitig. Eine getrennte Weitschaltung würde eine entsprechende gegenseitige Verriegelung über Auswahlshalter bedingen. Um die Anzahl der Eingabevariablen zu begrenzen, wurde bei der dargestellten Realisierung auf eine getrennte Weitschaltung der einzelnen Ablaufketten im Einzelschrittbetrieb ohne Bedingungen verzichtet.

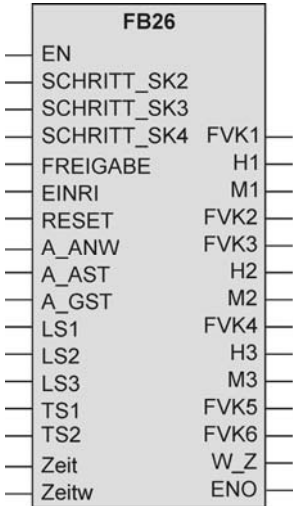
Der Ausgabebaustein FC 26/FB 26 muss an die Bedingungen des Chargenprozesses angepasst werden. Die jeweils aktiven Schritte der Ablaufketten SK2, SK3 und SK4 bestimmen die Bedingungen für die Ansteuerung der Aktoren der Anlage.

Deshalb sind die drei Schrittvariablen SCHRITT_SK2, SCHRITT_SK3 und SCHRITT_SK4 als Eingangsvariablen des Bausteins zu deklarieren.

Die Bestimmungszeichen „S“ und „R“ bei den Aktionen der Schrittke 2 des Mischkessels erfordern die Deklaration von statischen Lokalvariablen als Hilfsspeicher. Somit muss ein Funktionsbaustein FB 26 für die Befehlsausgabe verwendet werden.

Ausgabebaustein FB 26:

Ein- und Ausgangsvariablen:



Steuerungsprogramm:

Maskierung des 1/2-Bytes zur Aktorauswahl:

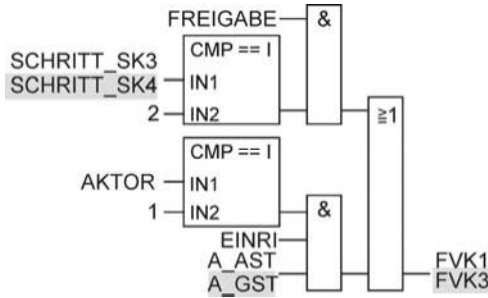
```

L #A_ANW
L W#16#F
UW
T #AKTOR
    
```

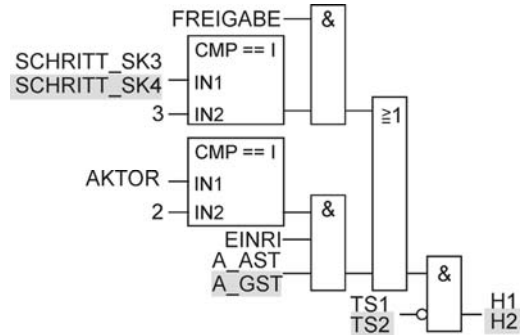
Das Steuerungsprogramm für die Aktoren der beiden Vorlagebehälter ist nur in einem Funktionsplan dargestellt. Die grau hinterlegten Variablen sind dabei für den Vorlagebehälter 2 bestimmt.

Funktionsplan für die Ansteuerung der Aktoren des Chargenprozesses:

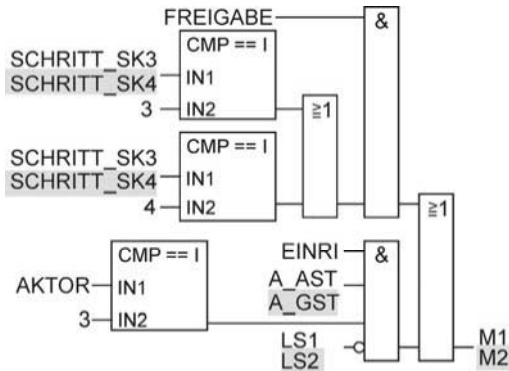
Einlassventil FVK1 bzw. FVK3



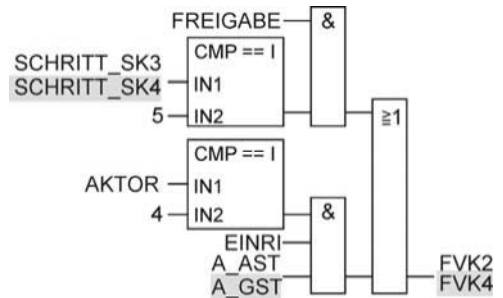
Heizung H1 bzw. H2



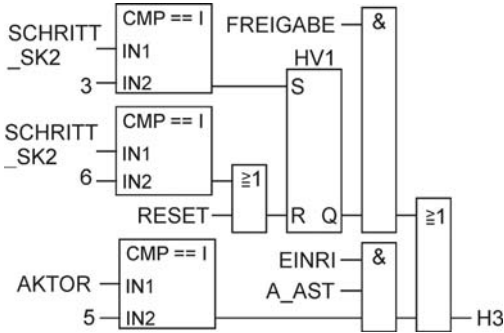
Rührwerkmotor M1 bzw. M2



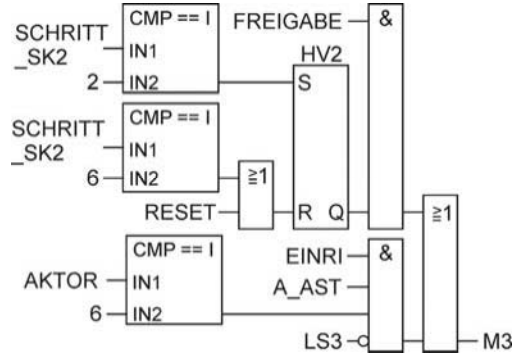
Auslassventil FVK2 bzw. FVK4



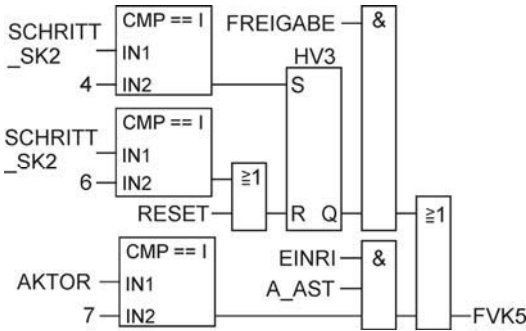
Mischkessel-Heizung H3



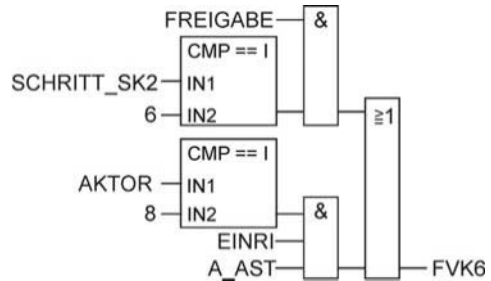
Mischkessel-Rührwerkmotor M3



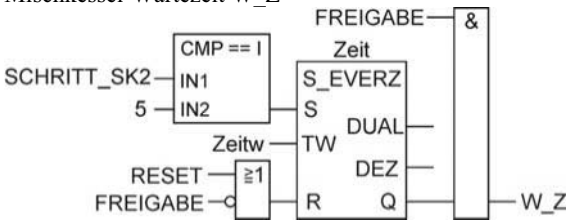
Mischkessel-Kühlwasserventil FVK5



Mischkessel-Auslassventil FVK6

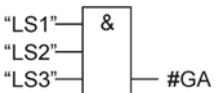


Mischkessel-Wartezeit W_Z

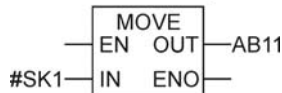


Nachfolgend ist der Aufruf der Bausteine für STEP 7 im Organisationsbaustein **OB 1** im Funktionsplan angegeben. Für die SPS-Operanden sind die in der Zuordnungstabelle der Aufgabe gegebenen Symbole verwendet worden. Die in mit „#“ gekennzeichneten Variablen müssen im **OB 1** als temporäre Variable mit entsprechendem Datentyp deklariert werden.

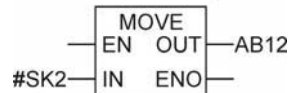
NW 1: Anlagengrundst.



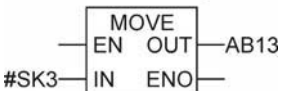
NW 2: Schrittanzeige 1



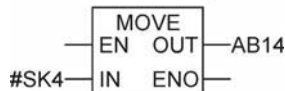
NW 3: Schrittanzeige 2

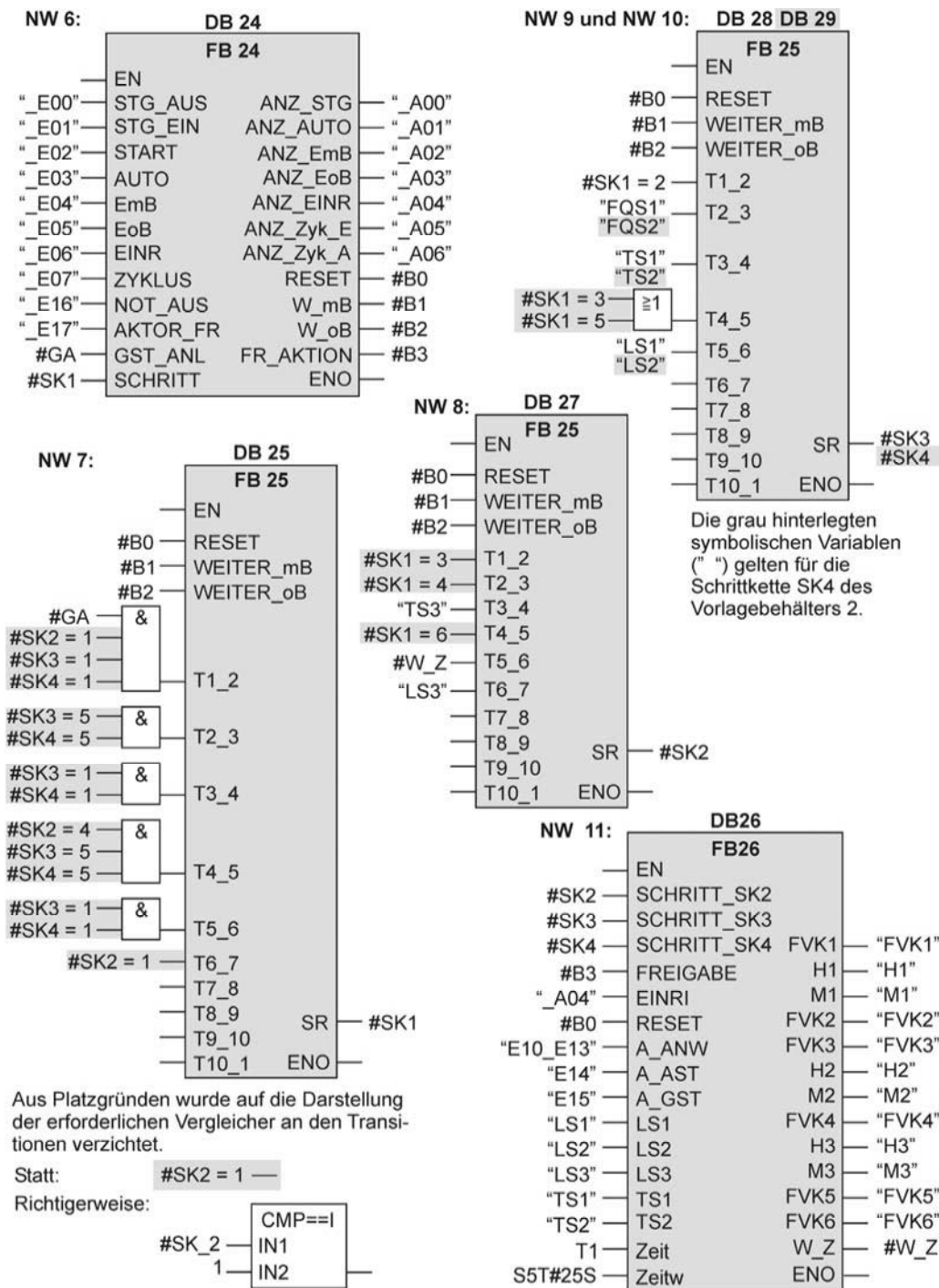


NW 4: Schrittanzeige 3



NW 5: Schrittanzeige 4





Hinweis: Auf die AWL-Darstellung der Steuerungsprogramme für STEP 7 und CoDeSys wird verzichtet. Diese stehen auf der Seite www.automatisieren-mit-sps.de zum Download zur Verfügung.

■ Beispiel 11.6: Bedarfsampelanlage

Wegen Bauarbeiten muss der Verkehr auf einer Zufahrtsstraße zu einer Fabrik über eine Fahrspur geleitet werden. Da am Tage das Verkehrsaufkommen sehr hoch ist, wird eine Bedarfsampelanlage installiert. Beim Einschalten der Anlage sollen beide Ampeln Rot signalisieren. Wird ein Initiator betätigt, soll die entsprechende Ampel nach 10 s auf Grün schalten. Die Grün-Phase soll mindestens 20 s andauern, bevor durch eventuelle Betätigung des anderen Initiators beide Signallampen wieder Rot zeigen. Nach 10 s wird dann die andere Fahrspur mit Grün bedient. Liegt keine Meldung eines Initiators vor, so bleibt die Ampelanlage in ihrem jeweiligen Zustand. Das Ausschalten der Anlage soll nur nach der Grün-Phase einer Fahrspur möglich sein.

Technologieschema:

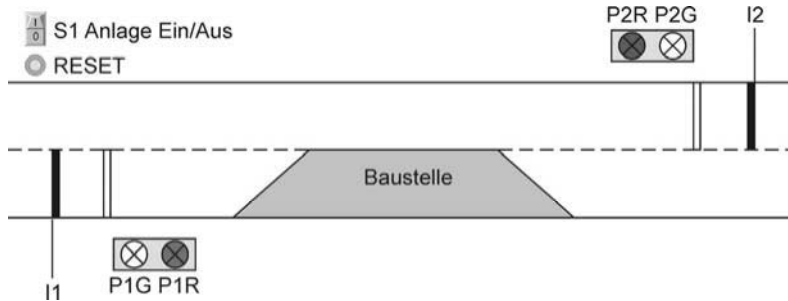


Bild 11.8: Bedarfsampelanlage

Zuordnungstabelle der Eingänge und Ausgänge:

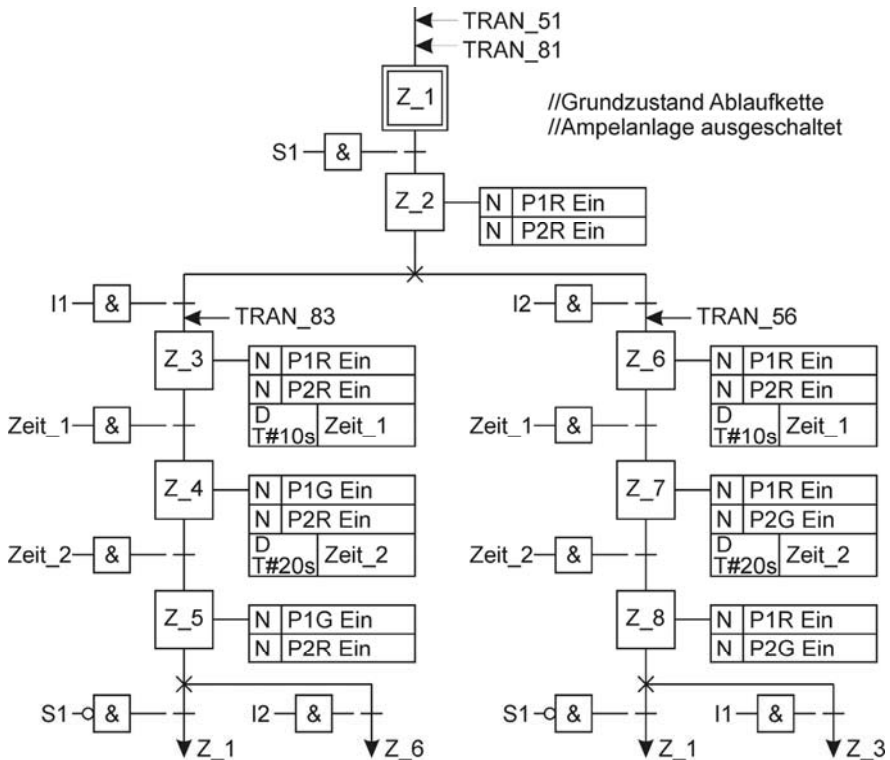
Eingangsvariable	Symbol	Datentyp	Logische Zuordnung	Adresse
Anlage Ein/Aus	S1	BOOL	Eingeschaltet	S1 = 1 E 0.0
Initiator 1	I1	BOOL	Betätigt	I1 = 1 E 0.1
Initiator 2	I2	BOOL	Betätigt	I2 = 1 E 0.2
Rücksetzen Ablaufkette	RESET	BOOL	Betätigt	RESET = 1 E 0.7
Ausgangsvariable				
Lampe Rot 1	P1R	BOOL	Leuchtet	P1R = 1 A 4.1
Lampe Rot 2	P2R	BOOL	Leuchtet	P2R = 1 A 4.6
Lampe Grün 1	P1G	BOOL	Leuchtet	P1G = 1 A 4.2
Lampe Grün 2	P2G	BOOL	Leuchtet	P2G = 1 A 4.7

Die Steuerungszustände für die Bedarfsampelanlage werden hier nicht als Schritte bezeichnet, sondern ganz neutral Zustände genannt. Mit dem Begriff Schritt wird bewusst eine Verbindung zu Fertigungs- bzw. Verfahrensschritten entsprechender Anlagen hergestellt. Bei der Bedarfsampelanlage kann man schlecht von Verkehrsschritten, sondern nur von Grün- und Rot-Phasen sprechen. Gleichwohl kann die Aufgabe Bedarfsampelanlage durch Einführen von Zuständen in eine übersichtliche Ablaufstruktur gebracht und mit den Beschreibungsmitteln des Ablauf-Funktionsplanes dargestellt werden. Folgende Steuerungszustände lassen sich dabei für die Baustellenampel einführen:

- Zustand 1 (Z_1): Initialzustand. Die Bedarfsampelanlage ist ausgeschaltet.
- Zustand 2 (Z_2): Die Bedarfsampelanlage wurde eingeschaltet. Beide Ampeln zeigen Rot.
- Zustand 3 (Z_3): Es liegt ein Bedarf von Initiator I1 vor. Da die Fahrspur erst nach einer Wartezeit von 10 Sekunden freigegeben werden kann, zeigen beide Ampeln Rot.
- Zustand 4 (Z_4): Die gesicherte Grünphase von 20 Sekunden beginnt. Ampel 1 zeigt Grün und Ampel 2 zeigt Rot.

- Zustand 5 (Z_5): Die gesicherte Grünphase von 20 Sekunden ist abgelaufen. Ausschalten der Ampelanlage oder Umschalten bei Bedarf auf der andern Seite (I2) ist möglich. Ampel 1 zeigt Grün und Ampel 2 zeigt Rot.
- Zustand 6 (Z_6): Entspricht Zustand 3, jedoch Bedarf von Initiator I2.
- Zustand 7 (Z_7): Entspricht Zustand 4, jedoch Ampel 1 Rot und Ampel 2 Grün.
- Zustand 8 (Z_8): Entspricht Zustand 5, jedoch Ampel 1 Rot und Ampel 2 Grün.

Ablauf-Funktionsplan:



Verriegelungen:

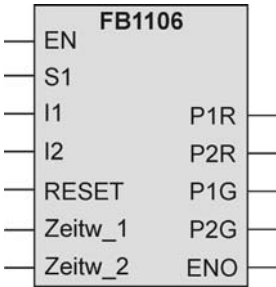
Bei diesem Funktionsablaufplan ist die Verriegelung der jeweils zwei möglichen Folgezustände unbedingt erforderlich. Befindet sich der Steuerungsprozess beispielsweise in Zustand 2, so könnten die Übergangsbedingungen I1 und I2 gleichzeitig erfüllt sein. Im nachfolgenden Programm wurde dem Zustand 3 der Vorrang gegenüber Zustand 6 gegeben, indem Zustand 6 durch Zustand 3 zurückgesetzt wird.

Auch die Nachfolgezustände von Zustand 5 und Zustand 8 müssen gegenseitig verriegelt werden. Zustand 1 hat Vorrang gegenüber Zustand 3 bzw. Zustand 6. Ist die Steuerung in Zustand 5 und treten beide Übergangsbedingungen S1 und I2 gleichzeitig auf, so setzt sich nur Zustand 1 durch. Zustand 6 wird in diesem Konfliktfall von Zustand 1 zurückgesetzt. Entsprechend ist dies bei den Nachfolgern von Zustand 8 vorgesehen.

Die Umsetzung des Ablauf-Funktionsplans erfolgt im Funktionsbaustein FB 1106.

Übergabeparameter:

Beschreibung der Parameter:



S1, I1, I2	BOOL	EIN-Schalter und Sensoren
RESET	BOOL	Rücksetzeingang
Zeitw1	TIME	Zeitvorgabe beide Ampeln ROT
Zeitw2	TIME	Zeitvorgabe beide Ampeln GRUEN
P1R, P2R	BOOL	ROT-Leuchte für Ampel 1 und 2
P1G, P2G	BOOL	GRÜN-Leuchte für Ampel 1 und 2

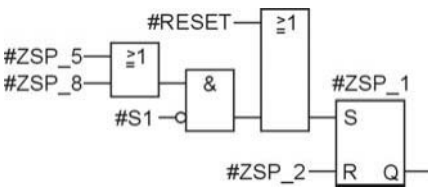
Deklarationstabelle FB 1106:

Name	Datentyp	Anfangswert
IN		
S1	Bool	FALSE
I1	Bool	FALSE
I2	Bool	FALSE
RESET	Bool	FALSE
Zeit 1	Timer	
Zeitw 1	S5Time	S5T#0MS
Zeit 2	Timer	
Zeitw 2	S5Time	S5T#0MS
OUT		
P1G	Bool	FALSE
P2G	Bool	FALSE
P1R	Bool	FALSE
P2R	Bool	FALSE

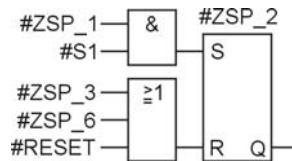
Name	Datentyp	Anfangswert
STAT		
ZSP 1	Bool	TRUE
ZSP 2	Bool	FALSE
ZSP 3	Bool	FALSE
ZSP 4	Bool	FALSE
ZSP 5	Bool	FALSE
ZSP 6	Bool	FALSE
ZSP 7	Bool	FALSE
ZSP 8	Bool	FALSE

Funktionsplan des Funktionsbausteins FB 1106:

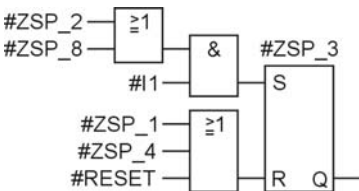
Zustand 1 (Initialzustand)



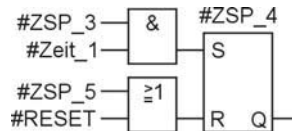
Zustand 2



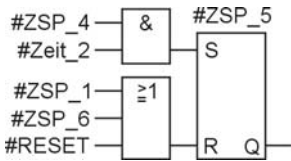
Zustand 3



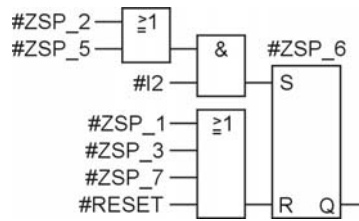
Zustand 4



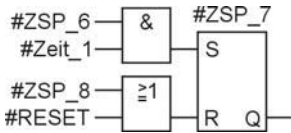
Zustand 5



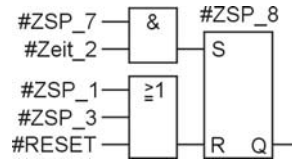
Zustand 6



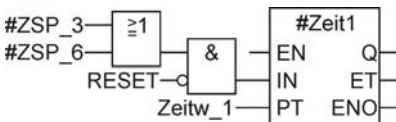
Zustand 7



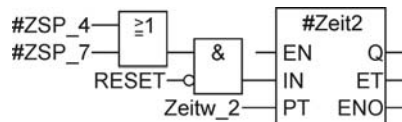
Zustand 8



Zeitfunktion 1



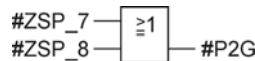
Zeitfunktion 2



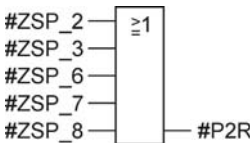
Ausgangszuweisung Grün 1



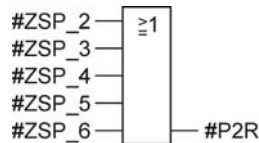
Ausgangszuweisung Grün 2



Ausgangszuweisung Rot 1



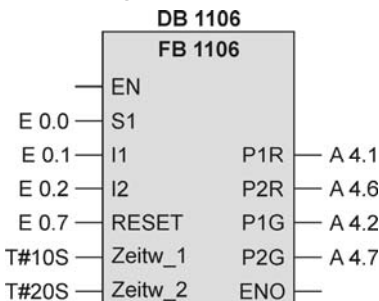
Ausgangszuweisung Rot 2



Durch den Aufruf des Systemfunktionsbausteins TON zur Bildung der beiden Zeit kann der Funktionsplan des Bausteins FB 1106 für STEP 7 und CoDeSys unverändert übernommen werden.

Aufruf des Funktionsbausteins FB 1106 im OB 1 bzw. PLC_PRG:

STEP7 Programm OB 1 im FUP:



CoDeSys Programm PLC_PRG im FUP:

