

Institut für Automatisierungs- und Systemtechnik

Grundlagenpraktikum

Versuch AS – G22

„Steuerung einer Flaschenfüllanlage mit Speicherprogrammierbarer
Steuerung SPS TSX Compact“

Fachverantwortlicher: Dr.-Ing. K. Wulff

Versuchsbetreuer: Dipl.-Ing. A. Fink

Ausgabe: Mai 2015

Inhalt

1	Einleitung.....	2
2	Prozessbeschreibung der Flaschenfülleinrichtung ET 706	2
3	Funktionsbeschreibung der Flaschenfülleinrichtung	4
4	Hardware der SPS TSX-Compact	7
4.1	Hardwareaufbau.....	7
5	Programmiersystem Concept 2.6.....	7
5.1	FBD = Funktions-Block-Diagramm	8
5.2	EFB – Elementare Funktionen und Elementare Funktionsbausteine.....	9
6	Datentypen -Konstanten -Variablen	10
6.1	Datentypen	10
6.2	Konstanten	10
6.3	Variablen	10
6.4	Literal.....	11
6.5	Direkt adressierte Variablen.....	11
6.6	Variablenliste Flaschenfüllanlage.....	12
7	Aufgabenstellung.....	13
7.1	Freigabe	13
7.2	Aufgabe 1: Fülleinrichtung abwärts	13
7.3	Aufgabe 2: Dosieren:	13
7.4	Aufgabe 3: Fülleinrichtung aufwärts	13
7.5	Aufgabe 4: Förderbandantrieb	13
7.6	Aufgabe 6: Flaschenzähler.....	13
8	Versuchsvorbereitung	14
9	Versuchsdurchführung	14
10	Anhang.....	15

1 Einleitung

Der Praktikumsversuch "Flaschenfülleinrichtung" dient dazu, die Kenntnisse der digitalen Schaltungstechnik in einem SPS-Programm an einem praktischen Beispiel kennenzulernen und anzuwenden. Die dynamische Prozess-Simulationseinheit ET 706 stellt realistisch das statische und dynamische Verhalten des auf der Frontplatte abgebildeten Prozesses dar. Mit Hilfe eines Ablauf- oder Verriegelungssteuerprogramms soll der Prozess, der wirklichkeitsnah auf die Eingangssignale reagiert, so ablaufen, dass es zu keiner Fehlfunktion der Flaschenfüllvorrichtung kommt. Dabei sind verschiedene steuerungstechnische Bausteine, wie positive und negative Flankenerkennung, Timer, Zähler und RS-Flip-Flops im Programm anzuwenden.

2 Prozessbeschreibung der Flaschenfülleinrichtung ET 706

Der Modellbaustein ET 706 simuliert den Prozess einer Flaschenfüllanlage von der Bereitstellung leerer Flaschen, ihrer Zubringung zur Abfülleinrichtung, anschließender Dosierung, bis zum Abstellen der gefüllten und gezählten Flaschen im Kasten. Die Steuerung des Prozesses wird von der SPS TSX Compact übernommen. Diese fragt die Zustände des Modells ab, reagiert über ein Programm darauf und steuert die Eingänge des Modells an.

Der Flaschenfüllvorgang läuft folgendermaßen ab:

Nach dem Einschalten des Modells befindet sich die Anlage im Freigabezustand.

Das bedeutet:

- leere Flaschen stehen bereit
- Förderband, Dosiereinrichtung und Flaschenzubringer sind in der Ausgangsposition
- ein leerer Kasten befindet sich am Ende des Zubringers
- die Freigabeanzeige leuchtet.

Wird die Taste "Freigabe" betätigt, so transportiert das Förderband die erste leere Flasche zur Füllposition, die Freigabeanzeige wird zurückgesetzt. Erreicht sie die Füllposition, so stoppt das Förderband. Die Fülleinrichtung beginnt nach unten zu fahren und wird in der unteren Position angehalten. Jetzt wird die Dosiereinrichtung eingeschaltet und die Flasche gefüllt. Danach fährt die Fülleinrichtung wieder nach oben, bis die obere Position erreicht ist. Die volle Flasche wird vom Flaschenzubringer zum Kasten transportiert und im Kasten abgestellt. Der Vorgang beginnt vom neuen und wird erst, nachdem die zwölfte Flasche im Kasten abgestellt wurde, beendet.

Eingänge (aus Sicht des Simulationsmoduls ET706):

A1: Dosiereinrichtung (Durchlaßventil) "EIN"

A2 : Fülleinrichtung "abwärts"

A3 : Fülleinrichtung "aufwärts"

A4: Förderband (Antrieb) "EIN"

A5 : Freigabeanzeige = Freigabeanforderung

Ausgänge (aus Sicht des Simulationsmoduls ET706):

E1: Oberer Grenztaster (Öffner)

E2: Unterer Grenztaster (Öffner)

E3: Positionstaster (Schließer)

E4: Kastenfreigabe (Schließer) "Freigabetaster"

E5: Flasche im Kasten abgestellt (Impuls)

Einschaltnormierung

Nach dem Einschalten der Stromversorgung werden die Speicher der Simulationseinheit selbsttätig in definierte Anfangszustände gesetzt.

3 Funktionsbeschreibung der Flaschenfülleinrichtung

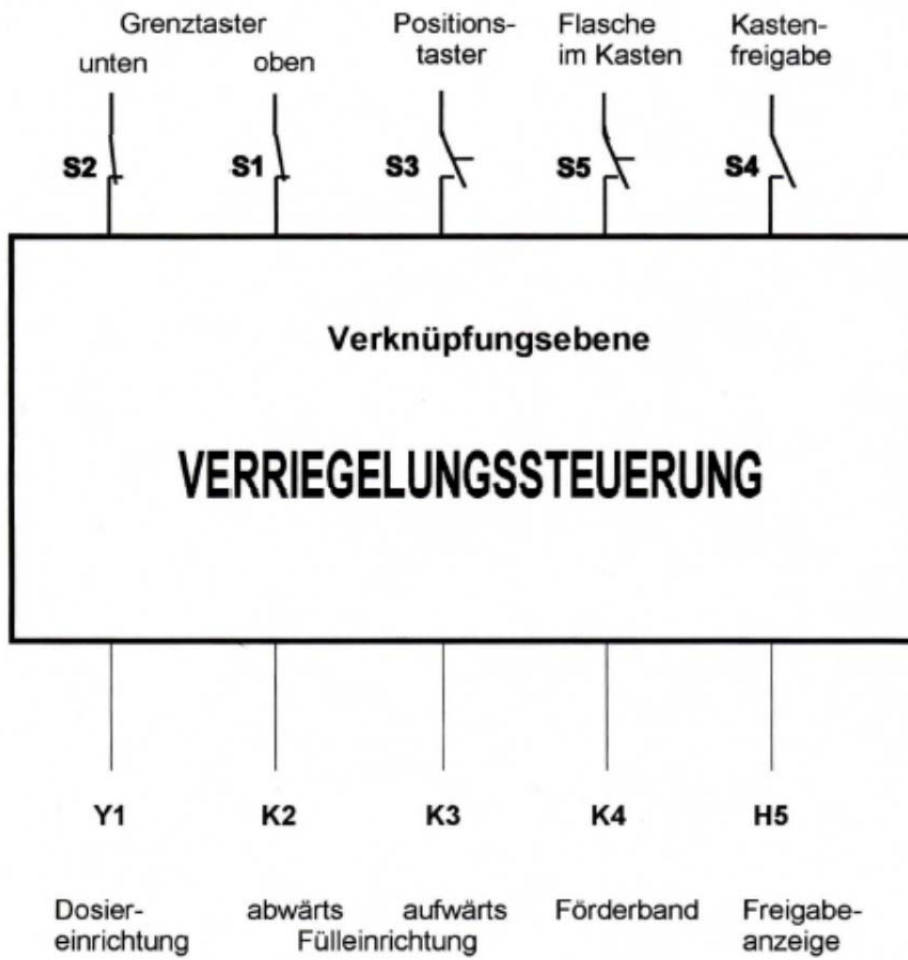
Das Modell ist mit der SPS Compact zu verschalten, dazu sind die Modelleingänge mit den SPS-Ausgängen zu verbinden.

SPS	Modell ET706	Signal	Kommentar
DE1	E1	S1_GO	Oberer Grenztaster
DE2	E2	S2_GU	Unterer Grenztaster
DE3	E3	S3_Postas	Positionstaster Flasche an Füllposition
DE4	E4	S4_Freigabe	Freigabetaster
DE5	E5	S5_flimp	Flaschenzählimpuls, Flasche im Kasten
DA1	A1	Y1_dosein	Dosiereinrichtung einschalten
DA2	A2	K2_fuellab	Fülleinrichtung abwärts fahren
DA3	A3	K3_fuellauf	Fülleinrichtung aufwärts fahren
DA4	A4	K4_bandein	Förderband anschalten
DA5	A5	H5_freianz	Freigabeanzeige

Wird die Betriebsspannung eingeschaltet, so befindet sich das Modell im Grundzustand. Der obere Grenztaster **S1_GO** ist geöffnet und der Positionstaster **S3_Postas** geschlossen. Wird die Anlage mit dem Freigabetaster **S4_Freigabe** in Betrieb genommen, so transportiert das Förderband **K4_bandein** leere Flaschen heran, und die Freigabeanzeige H5_freianz wird zurückgesetzt. Das Förderband löst am Positionstaster **S3_Postas** immer dann ein Signal aus, wenn sich eine Flasche unter der Fülleinrichtung befindet. Der Positionstaster stoppt das Förderband und bleibt bis zur nächsten Förderbandbewegung geschlossen. Die Fülleinrichtung fährt abwärts **K2_fuellab**, wird vom unteren Grenztaster **S2_GU** angehalten und die Dosiereinrichtung **Y1_dosein** nach einer Einschaltverzögerung von 1 Sekunde angesteuert. Der Füllvorgang **zeit5s** für eine Flasche muss 5 Sekunden dauern. Nach dieser Zeit fährt die Fülleinrichtung **K3_fuellauf** nach oben und wird vom oberen Grenztaster **S1_GO** angehalten. Nun wird die gerade gefüllte Flasche von einer Fördereinrichtung, die nicht zu steuern ist, im Kasten abgestellt, und ein Flaschenzählimpuls **S5_flimp** wird erzeugt.

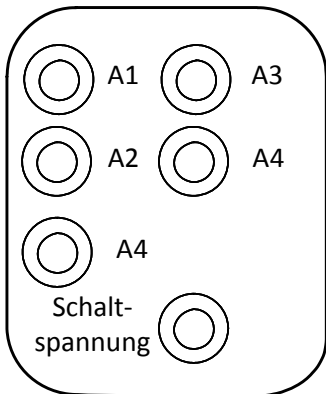
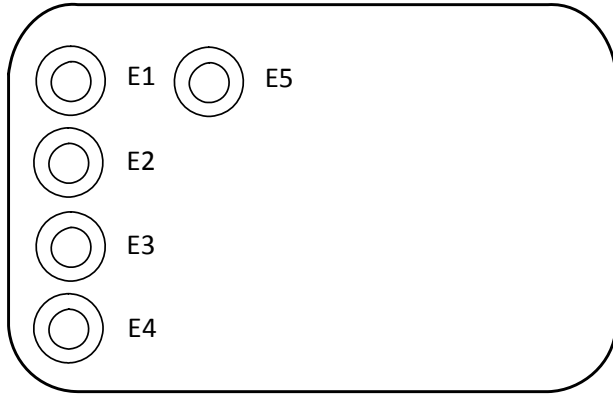
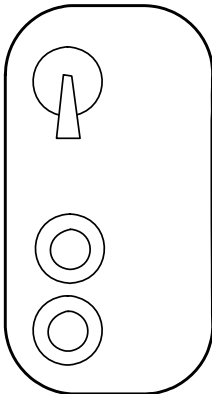
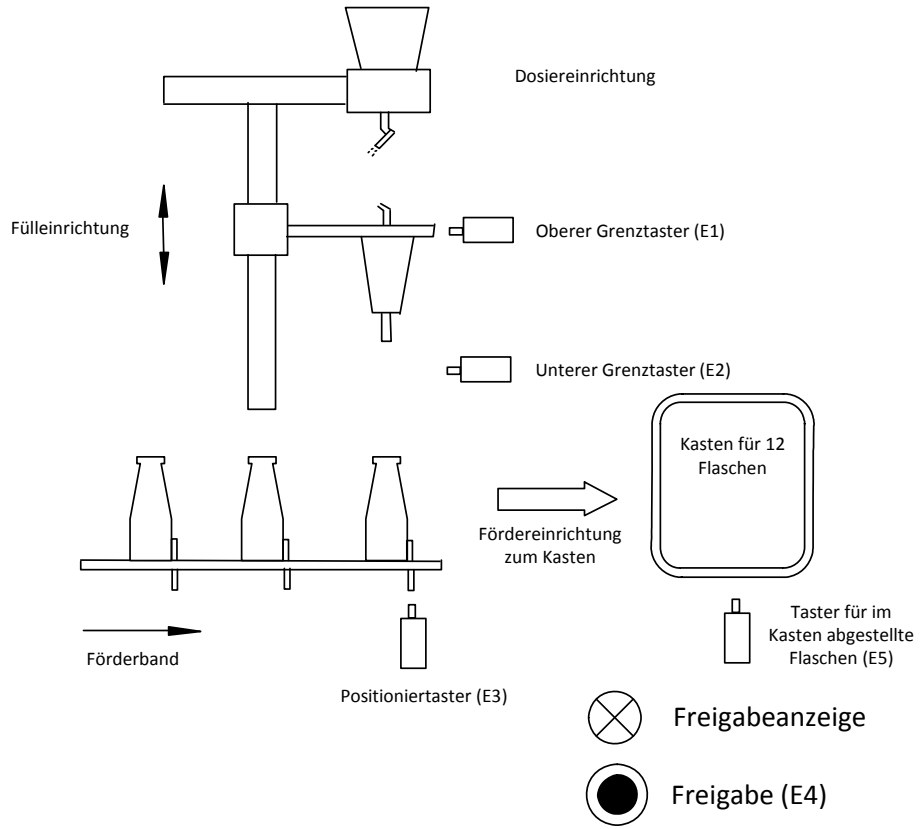
Nach einer Verzögerungszeit von 1 Sekunde läuft das Band erneut an und der Vorgang beginnt von vorn. Sind 12 Flaschen im Kasten, ist der Füllprozess beendet und die Freigabeanzeige **H5_freianz** wird erneut gesetzt. Der Vorgang lässt sich erneut starten, wenn der Freigabetaster **S4_Freigabe** betätigt wird.

Eingabeebene



Ausgabebene

Flaschenfülleinrichtung



4 Hardware der SPS TSX-Compact

Der Grundträger und die Erweiterungsträger einer SPS sind mit verschiedenen Bausteinen, wie CPU mit integrierten Netzteil, Kommunikations-, digitale und analoge Ein- und Ausgabebaugruppen bestückt.

Die CPU und das Netzteil befinden sich auf den ersten beiden Steckplätzen, dann folgt meist eine Kommunikationsbaugruppe. Die nachfolgenden Steckplätze können beliebig mit Prozessbaugruppen belegt werden.

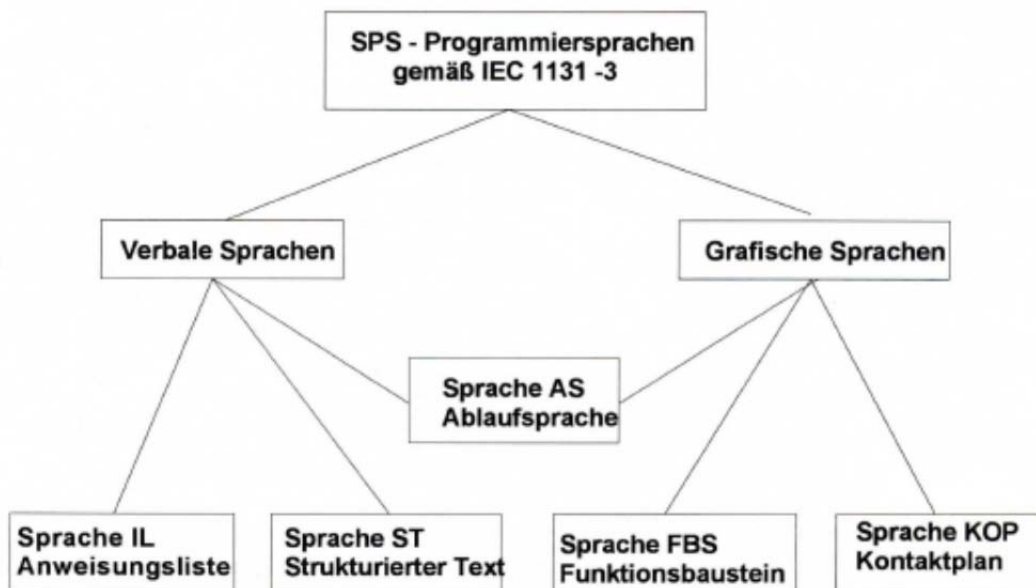
4.1 Hardwareaufbau

Steckplatz	Baugruppentyp	Baugruppenname
1 – 1, 2	CPU TSX Compact	PC-E984-258
1 - 4	Digitale Ausgabeeinheit	DAP 216
1 - 5	Digitale Eingabeeinheit	DEP 216

Für jede eingesetzte Baugruppe muss eine Adresszuweisung für die Kanäle erfolgen. Wird die jeweilige Anfangsadresse eingetragen, so wird die Endadresse anhand der eingetragenen Baugruppe ermittelt.

5 Programmiersystem Concept 2.6

Das Programmierwerkzeug "Concept" arbeitet mit Menüsteuerung. Standardmenüs, wie Datei, Fenster und Hilfe sind konstant in allen Editoren. Windows-Standards, wie Druckersteuerung und Druckbefehle sind ohne Modifikation verwendbar. Hell/Dunkelsteuerung für ausführbare und nicht ausführbare Befehle ist eine weitere optische Hilfe. Weitere Steuerungsmöglichkeiten von Concept auf Windows-Basis sind Dialogfelder mit Kontrollkästchen, Optionsschaltflächen Text und Listfelder.



Im Praktikum wird nur die grafische Sprache FBD = Funktions-Block-Diagramm verwendet.

5.1 FBD = Funktions-Block-Diagramm

Ein Projekt besteht in der Regel aus mehreren SPS-Organisationseinheiten - Sectionen - genannt.

In einer Section befinden sich eine Menge von

EFB – Elementare Funktionsbausteine

DFB – Abgeleitete Funktionsbausteine

UDEFB – Benutzerdefinierte Funktionen und Funktionsbausteine

Im Praktikum werden nur EFB verwendet.

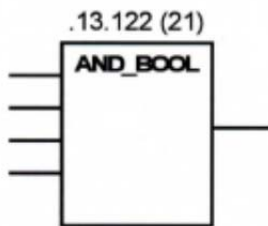
5.2 EFB – Elementare Funktionen und Elementare Funktionsbausteine

Sie werden von Concept in Form von Bibliotheken zur Verfügung gestellt. Die IEC -Bibliothek enthält nachfolgend aufgeführte Funktionen

Arithmetic	Bistable
Comparison	Converter
Counter	Edge detection
Logik	Numerical
Timer	Selection

Die EFB werden grafisch als Rahmen mit Ein- und Ausgängen dargestellt. Je nach Funktion verfügen sie über interne Zustände (z.B. Zählvariable bei Counter).

Beispiel EFB



6 Datentypen -Konstanten -Variablen

6.1 Datentypen

Concept, der neuen Norm folgend, klassifiziert die bisherigen Datentypen Eingänge, Ausgänge und Merker zum Typ "BOOL".

Bezeichnung	Datentyp	Bits (Wertebereich)
BOOL	Boolean	1 (0,1)
BYTE	Bitfolge	8
WORD	Bitfolge	16
DWORD	Bitfolge	32
LWORD	Bitfolge	64
SINT	Ganze Zahl	8 (0 bis 255)
INT	Ganze Zahl	16 (-32768 bis +32767)
DINT	Ganze Zahl	32 (-2147483648 bis +2147483647)
LINT	Ganze Zahl	64
USINT	Ganze Zahl ohne Vorzeichen	8
UINT	Ganze Zahl ohne Vorzeichen	16 (0 bis 65535)
UDINT	Ganze Zahl ohne Vorzeichen	32
ULINT	Ganze Zahl ohne Vorzeichen	64
REAL	Gleitkommazahl	32
LREAL	Gleitkommazahl	64
TIME	Zeitdauer	
DATE	Datum	
TIME_OF_DAY	Uhrzeit	
String	Zeichenfolge	

6.2 Konstanten

Die Konstante lässt sich Ein-und Ausgängen zuordnen und der Wert wird im Variablen-Editor festgelegt.

6.3 Variablen

Die located Variablen, die dem Signalspeicher der SPS zugeordnet sind, lesen Signalzustände der SPS aus oder werden an die SPS ausgegeben.

Die unlocated Variablen (früher Merker) werden nur verwaltet und gespeichert, ihre Adressen auf der SPS sind nicht bekannt, sie besitzen nur symbolische Namen.

6.4 Literal

Literale weisen den Eingängen nur direkte Werte zu, dabei ist der vom Baustein geforderte Datentyp einzuhalten.

Für einen Zeitbaustein ist als Sollwert der Datentyp TIME notwendig.

Das Literal lautet :

t#3h20m23s6ms für 3 Stunden, 20 Minuten, 23 Sekunden, und 6 Millisekunden

6.5 Direkt adressierte Variablen

Für direkt adressierte Variablen gelten folgende Eingangs- und Ausgangsreferenzen.

Variablenart	Präfix	im FBD; LD	Variable
Ausgangsbit	0	%0:00008	Ausgangsbit 8
Eingangsbit	1	%1:00014	Eingangsbit 14
Eingangswort	3	%3:00002	Eingangswort 2
Ausgangswort	4	%4:00007	Ausgangswort 7

6.6 Variablenliste Flaschenfüllanlage

	Variablenname	Datentyp	Adresse	Initialwert	Kommentar
Ausgänge	Y1_dosein	BOOL	000001		Dosiereinrichtung An
	K2_fuellab	BOOL	000002		Fülleinrichtung abwärts
	K3_fuellauf	BOOL	000003		Fülleinrichtung aufwärts
	K4_bandein	BOOL	000004		Förderband An
	H5_freianz	BOOL	000005		Freigabeanzeige An
Eingänge	S1_GO	BOOL	100001		Grenztaster Oben
	S2_GU	BOOL	100002		Grenztaster Unten
	S3_Postas	BOOL	100003		Positionstaster
	S4_Freigabe	BOOL	100004		Freigabetaster
	S5_flimp	BOOL	100005		Flaschenzählimpuls
Merker	dosaus	BOOL			Signal Dosiereinrichtung Aus
	flaimp	BOOL			Flanke Flaschenimpuls
	flapos	BOOL			Flanke Positionstaster
	Kasten_voll	BOOL		TRUE	
	fuellsoll	INT		12	Flaschensollwert=12
	zist	INT			Zustand im Zähler
Literal	zeit1s	TIME		t#1s	Wartezeit 1s
	zeit5s	TIME		t#5s	Wartezeit 5s

Achtung: Diese Variablenliste ist nicht vollständig, es sind noch weitere Merker in der Liste enthalten, welche verwendet werden können.

7 Aufgabenstellung

7.1 Freigabe

Zu Beginn soll die Anlage gesperrt sein und die Freigabeanzeige leuchten. Nach Überprüfung des Grundzustandes (siehe Beschreibung in Punkt 3) soll die Anlage über den Freigabetaster gestartet werden und ihren Betrieb aufnehmen.

7.2 Aufgabe 1: Fülleinrichtung abwärts

Wenn das Förderband eine leere Flasche herantransportiert hat und der Förderbandantrieb abgeschaltet hat (dies wird durch den Eingang **S3_Postas** signalisiert), muss die Fülleinrichtung **K2_fuellab** abwärts fahren, bis der untere Grenztaster **S2_GU** öffnet.

7.3 Aufgabe 2: Dosieren:

Wenn der untere Grenztaster **S2_GU** öffnet, ist nach einer **Verzögerung von 1 Sekunde** die Dosiereinrichtung **Y1_dosein** für **5 Sekunden** einzuschalten.

7.4 Aufgabe 3: Fülleinrichtung aufwärts

Wenn die Dosiereinrichtung **Y1_dosein** abschaltet, muss die Fülleinrichtung **K3_fuellauf** fahren, bis der obere Grenztaster **S1_GO** öffnet.

7.5 Aufgabe 4: Förderbandantrieb

Das Förderband **K4_bandein** geht mit einer **Verzögerung von 1 Sekunde** in Betrieb, wenn die Freigabeanzeige **H5_freianz** aus ist und eine Flasche im Kasten abgestellt wurde (**S5_flimp**). Das Förderband hält an, wenn der Positionstaster **S3_Postas** ausgelöst wird.

Achtung: S3_Postas wird erst bei Bewegung des Förderbandes auf LOW gesetzt (Eigenart des Simulationsmodells) !

7.6 Aufgabe 6: Flaschenzähler

Die Flaschen, die sich im Kasten befinden, müssen gezählt werden. Als Zähltakt kann der Flaschenimpuls **S5_flimp** genutzt werden. Mit der Kastenfreigabe ist für den Zähler ein Setzimpuls zu erzeugen, damit der Zählersollwert **fuellsoll** übernommen wird. Es müssen 12 Flaschen im Kasten sein. Über eine Variable soll der aktuelle Zählerstand überwacht werden.

Wenn der Sollwert (12 Flaschen) erreicht ist, soll der gesamte Prozess gestoppt werden und die Freigabeanzeige leuchten. Über den Freigabetaster soll die Anlage für einen nächsten Durchlauf gestartet werden.

8 Versuchsvorbereitung

Skizzieren Sie für jede Teilaufgabe die Ein- und Ausgangssignale und eine mögliche Verknüpfung zur Realisierung der Teilaufgabe.

Skizzieren Sie für jede Teilaufgabe die Zeitverläufe der Ein- und Ausgangssignale.

Beachten Sie die Schaltung der Signalgeber (Öffner/ Schließer)!

9 Versuchsdurchführung

Erstellen Sie für jede Teilaufgabe eine Programm-Section unter Verwendung der grafischen Programmiersprache FBD.

Zu verwendende Grundfunktionen:

AND: UND –Funktion

OR: ODER –Funktion

RS: Bistabiler Funktionsbaustein (Rücksetzen dominant)

SR: Bistabiler Funktionsbaustein (Setzen dominant)

R _ TRIG: Erkennung steigender Flanken

F _ TRIG: Erkennung fallender Flanken

CTU: Aufwärts-Zähler

TON: Einschaltverzögerung

TOF: Ausschaltverzögerung

Laden Sie das erstellte Programm auf die angeschlossene SPS.

Simulieren Sie das Verhalten der erstellten Section mit dem angeschlossenen Schalterblock und testen Sie Ihr Programm.

Nach Abarbeitung aller Teilaufgaben verschalten Sie die SPS mit der Flaschenfüllanlage und überprüfen das Verhalten des Programms an der Flaschenfüllanlage.

10 Anhang

Anleitung Programm testen und in die SPS übertragen

Auszug Concept Programmierhandbuch „Erstellung eines Programms mit der Funktionsbausteinsprache FBD“

Auszug Concept Programmierhandbuch IEC Bausteinbibliothek - verwendete Grundfunktionen

Programm testen und in die SPS übertragen

Verbindung zur SPS herstellen:

Online -> Verbinden

Modbus, SPS-TIn.-Nr. 001, RTU-Modus, COM 1 (9600, 8, gerade, 1, keine)

Programm übertragen:

Online -> Laden

IEC Program muss angewählt sein

Initialwerte müssen angewählt sein

nach Neukonfiguration muss auch Konfiguration angewählt sein

Onlineanzeige der SPS:

Button Io betätigen – die angewählte Section wird grau hinterlegt und die Signale der SPS werden angezeigt

Grün = 1 – Signal

Rot = 0 - Signal

Logiksignale ändern:

Button Io deaktivieren, Signale ändern, Änderung laden

Doppelklick auf zu ändernden Eingang und aus Variablenliste andere Variable auswählen

Signalzustand ändern:

Referenzdaten – Editor wählen

Variable ändern oder ergänzen

Freischalten mit Disable

Eintrag in **Value** ON oder OFF

SPS ein- oder ausschalten:

Online -> Online-Steuerung

SPS Stoppen, Starten, Löschen

7.6 Erstellen eines Programms mit der Funktionsbausteinsprache FBD

Erstellen eines Programms in der Funktionsbausteinsprache FBD

Einleitung

Die nachfolgende Beschreibung gibt ein Beispiel zur Erstellung eines Programms in der Funktionsbausteinsprache (FBD). Das Erstellen eines Programms in der Funktionsbausteinsprache FBD gliedert sich in 2 Hauptschritte:

Schritt	Aktion
1	Erzeugen einer Section (siehe <i>Erzeugen einer Section</i> , S. 225)
2	Erstellen der Logik (siehe <i>Erstellen der Logik</i> , S. 226)

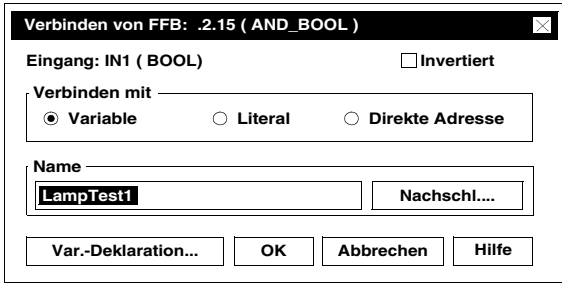
Erzeugen einer Section

Zum Erzeugen einer Section führen Sie die folgenden Schritte aus:

Schritt	Aktion
1	<p>Erzeugen Sie mit dem Menübefehl Datei → Neue Section... eine neue Section und geben Sie einen Section-Namen ein.</p> <p>Hinweis: Der Section-Name (max. 32 Zeichen) muß im gesamten Projekt eindeutig sein, dabei wird nicht zwischen Groß-/Kleinschreibung unterschieden. Existiert der eingegebene Section-Name schon, werden Sie gewarnt und Sie müssen einen anderen Namen wählen. Der Section-Name muß den IEC-Namenskonventionen entsprechen, andernfalls erfolgt eine Fehlermeldung.</p> <p>Hinweis: Nach IEC1131-3 sind als erstes Zeichen von Namen nur Buchstaben zulässig. Falls Sie aber auch Zahlen als erstes Zeichen verwenden wollen, können Sie dieses mit dem Menübefehl Optionen → Voreinstellungen → IEC-Erweiterungen... → Führende Ziffern in Bezeichnern zulässig freigeben.</p>

Erstellen der Logik

Zum Erstellen der Logik führen Sie die folgenden Schritte aus:

Schritt	Aktion
1	<p>Um einen FFB in die Section einzufügen, wählen Sie den Menübefehl Objekte → FFB wählen...</p> <p>Reaktion: Das Dialogfeld FFBs von Bibliothek wird geöffnet.</p> 
2	<p>In diesem Dialogfeld können Sie mit der Befehlsschaltfläche Bibliothek... eine Bibliothek wählen und einen FFB daraus auswählen. Sie können sich aber auch mit der Befehlsschaltfläche DFB die von Ihnen erzeugten DFBs anzeigen lassen und einen auswählen.</p>
3	<p>Plazieren Sie nun den gewählten FFB in der Section.</p>
4	<p>Wenn Sie alle FFBs plaziert haben schließen Sie das Dialogfeld mit Schließen.</p>
5	<p>Aktivieren Sie mit Objekte → Anwahl-Modus den Anwahl-Modus, klicken Sie auf den FFB, und schieben Sie die FFBs an die gewünschte Position.</p>
6	<p>Aktivieren Sie mit Objekte → Verbindung den Verbindungs-Modus, und stellen Sie die Verbindungen zwischen den FFBs her.</p>
7	<p>Nun aktivieren Sie mit Objekte → Anwahl-Modus wieder den Anwahl-Modus, und führen Sie einen Doppelklick auf einem der unverknüpften Ein-/Ausgänge aus.</p> <p>Reaktion: Es wird das Dialogfeld Verbinden von FFB geöffnet, in dem Sie dem Ein-/Ausgang einen Aktualparameter zuordnen können.</p> 

Schritt	Aktion
8	<p>Je nach Programmlogik können Sie dem Ein-/Ausgang zuordnen:</p> <ul style="list-style-type: none"> ● Variable <ul style="list-style-type: none"> ● Located Variable Über eine Located Variable können Sie dem Ein-/Ausgang des FFBs ein Hardware-Ein-/Ausgangssignal zuordnen. Der Name der Variablen wird Ihnen am Ein-/Ausgang im Editor-Fenster angezeigt. ● Unlocated Variable Sie können die dem Ein-/Ausgang des FFBs zugeordnete Unlocated Variable als Merker verwenden, d.h. zum Auflösen von Schleifen oder zur Übergabe von Werten zwischen verschiedenen Sections. Der Name der Variablen wird Ihnen am Ein-/Ausgang im Editor-Fenster angezeigt. ● Konstante Sie können dem Eingang des FFBs eine Konstante zuordnen. Sie können die Konstante an andere Sections übergeben. Den Wert der Konstanten bestimmen Sie im Variablen-Editor. Der Name der Konstanten wird Ihnen am Eingang im Editor-Fenster angezeigt. ● Literal Sie können dem Eingang ein Literal zuordnen, d.h. dem Ein-/Ausgang direkt einen Wert zuordnen. Der Wert wird Ihnen am Eingang im Editor-Fenster angezeigt. ● Direkte Adresse Über eine Adresse können Sie dem Ein-/Ausgang ein Hardware-Ein-/Ausgangssignal zuordnen. Die Adresse wird Ihnen am Ein-/Ausgang im Editor-Fenster angezeigt. <p>Hinweis: Beispiel für den Aufruf von Multielement-Variablen siehe <i>Aufruf von Abgeleiteten Datentypen</i>, S. 593.</p> <p>Hinweis: Nicht verknüpfte FFB-Eingänge sind standardmäßig mit einer "0" belegt.</p>
9	<p>Sichern Sie nun die FBD-Section mit dem Menübefehl Datei → Projekt speichern.</p>

Auszug Concept Programmierhandbuch IEC Bausteinbibliothek

- verwendete Grundfunktionen

AND_***:	Logische UND-Funktion
CTU:	Aufwärts-Zähler
F-TRIG:	Erkennung fallender Flanken
OR_***:	Logische ODER-Funktion
R-TRIG:	Erkennung steigender Flanken
RS:	Bistabiler Funktionsbaustein, Rücksetzen dominant
SR:	Bistabiler Funktionsbaustein, Setzen dominant
TOF:	Ausschaltverzögerung
TON:	Einschaltverzögerung

(*** = BOOL, BYTE, WORD)

Kurzbeschreibung

Funktionsbeschreibung

Die Funktion verknüpft (nach UND-Logik) die Bit-Folgen an den Eingängen und gibt das Ergebnis am Ausgang aus. Die Verknüpfung erfolgt bitweise.

Es können die Datentypen der Gruppe ANY_BIT verarbeitet werden.

Hinweis: Diese Funktion ist mit booleschen Variablen in der Programmiersprache LD (Ladder Diagram) nicht verfügbar, da dort die gleiche Funktionalität mit Kontakten realisiert werden kann.

Die Datentypen aller Eingangswerte und der des Ausgangswertes müssen gleich sein. Für die Verarbeitung der verschiedenen Datentypen steht jeweils eine spezielle Funktion zur Verfügung.

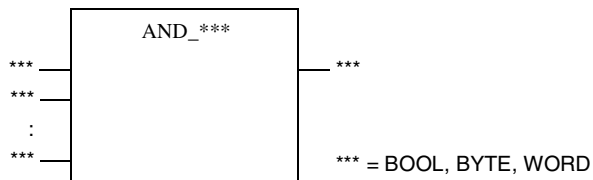
Die Anzahl der Eingänge kann erhöht werden.

Als zusätzliche Parameter können EN und ENO projiziert werden.

Darstellung

Symbol

Darstellung des Bausteins:



Formel

$$\text{OUT} = \text{IN1} \& \text{IN2} \& \text{INn}$$

Parameterbeschreibung

Beschreibung der Bausteinparameter

Parameter	Datentyp	Bedeutung
IN1	BOOL, BYTE, WORD	Eingangs-Bit-Folge
IN2	BOOL, BYTE, WORD	Eingangs-Bit-Folge
INn	BOOL, BYTE, WORD	Eingangs-Bit-Folge
OUT	BOOL, BYTE, WORD	Ausgangs-Bit-Folge

Kurzbeschreibung

Funktionsbeschreibung

Der Funktionsbaustein wird zum Aufwärtszählen von INT-Werten verwendet.

Die Funktionsbausteine zum Zählen von DINT, UDINT und UINT-Werten finden Sie in der Bibliothek Extended.

Bei "1"-Signal am Eingang R wird der Wert "0" dem Ausgang CV zugewiesen. Bei jedem Übergang von "0" nach "1" am Eingang CU wird der Wert von CV um 1 erhöht.

Bei $CV \geq PV$ wird der Ausgang Q auf "1" gesetzt.

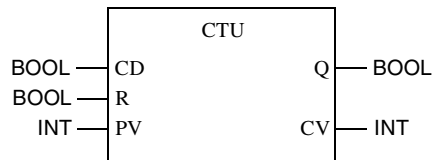
Hinweis: Der Zähler arbeitet nur bis zum maximal Wert des Ausgangs CV. Es findet kein Überlauf statt.

Als zusätzliche Parameter können EN und ENO projektiert werden.

Darstellung

Symbol

Darstellung des Bausteins:



Parameterbeschreibung

Beschreibung der Bausteinparameter:

Parameter	Datentyp	Bedeutung
CU	BOOL	Trigger-Eingang
R	BOOL	Reset
PV	INT	Voreinstellungswert
Q	BOOL	Ausgang
CV	INT	Zählwert (Istwert)

Kurzbeschreibung

Funktionsbeschreibung

Der Funktionsbaustein wird zur Erkennung fallender Flanken 1 -> 0 verwendet.

Hinweis: Dieser Funktionsbaustein ist in der Programmiersprache LD (Ladder Diagram) nicht verfügbar, da es dort den "Kontakt - Negative Flanke" gibt, welcher die gleiche Funktionalität hat.

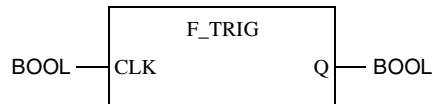
Der Ausgang Q wird "1", wenn ein Übergang von "1" nach "0" am Eingang CLK erfolgt. Der Ausgang bleibt von einer Ausführung des Funktionsbausteins bis zur nächsten Ausführung auf "1"; danach kehrt der Ausgang zu "0" zurück.

Als zusätzliche Parameter können EN und ENO projiziert werden.

Darstellung

Symbol

Darstellung des Bausteins:



Parameterbeschreibung

Beschreibung der Bausteinparameter:

Parameter	Datentyp	Bedeutung
CLK	BOOL	Takt-Eingang
Q	BOOL	Ausgang

Kurzbeschreibung

Funktionsbeschreibung

Die Funktion verknüpft (nach ODER-Logik) die Bit-Folgen an den Eingängen und gibt das Ergebnis am Ausgang aus. Die Verknüpfung erfolgt bitweise.

Es können die Datentypen der Gruppe ANY_BIT verarbeitet werden.

Hinweis: Diese Funktion ist mit booleschen Variablen in der Programmiersprache LD (Ladder Diagram) nicht verfügbar, da dort die gleiche Funktionalität mit Kontakten realisiert werden kann.

Die Datentypen aller Eingangswerte und der des Ausgangswertes müssen gleich sein. Für die Verarbeitung der verschiedenen Datentypen steht jeweils eine spezielle Funktion zur Verfügung.

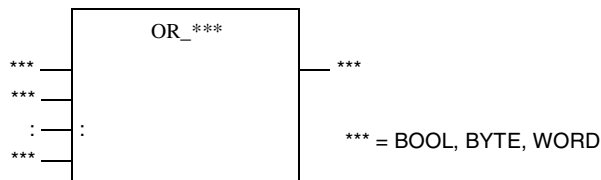
Die Anzahl der Eingänge kann erhöht werden.

Als zusätzliche Parameter können EN und ENO projiziert werden.

Darstellung

Symbol

Darstellung des Bausteins:



Formel

$$\text{OUT} = \text{IN1 OR IN2 OR .. OR INn}$$

Parameterbeschreibung

Beschreibung der Bausteinparameter:

Parameter	Datentyp	Bedeutung
IN1	BOOL, BYTE, WORD	Eingangs-Bit-Folge
IN2	BOOL, BYTE, WORD	Eingangs-Bit-Folge
INn	BOOL, BYTE, WORD	Eingangs-Bit-Folge
OUT	BOOL, BYTE, WORD	Ausgangs-Bit-Folge

Kurzbeschreibung

Funktionsbeschreibung

Der Funktionsbaustein wird zur Erkennung steigender Flanken 0 -> 1 verwendet.

Hinweis: Dieser Funktionsbaustein ist in der Programmiersprache LD (Ladder Diagram) nicht verfügbar, da es dort den "Kontakt - Positive Flanke" gibt, welcher die gleiche Funktionalität hat.

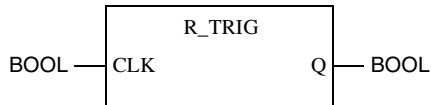
Der Ausgang Q wird "1", wenn ein Übergang von "0" nach "1" am Eingang CLK erfolgt. Der Ausgang bleibt von einer Ausführung des Funktionsbausteins bis zur nächsten Ausführung auf "1" (einen Zyklus); danach kehrt der Ausgang zu "0" zurück.

Als zusätzliche Parameter können EN und ENO projiziert werden.

Darstellung

Symbol

Darstellung des Bausteins:



Parameterbeschreibung

Beschreibung der Bausteinparameter:

Parameter	Datentyp	Bedeutung
CLK	BOOL	Takt-Eingang
Q	BOOL	Ausgang

Kurzbeschreibung

Funktionsbeschreibung

Der Funktionsbaustein wird als RS-Speicher mit der Eigenschaft "Rücksetzen dominant" verwendet.

Der Ausgang Q1 wird "1", wenn der Eingang S "1" wird. Dieser Zustand bleibt auch erhalten, wenn der Eingang S wieder "0" wird. Der Ausgang Q1 wird erst wieder "0", wenn der Eingang R1 "1" wird. Sind die Eingänge S und R1 gleichzeitig "1", setzt der dominierende Eingang R1 den Ausgang Q1 auf "0".

Der Anfangszustand von Q1 beim ersten Aufruf des Funktionsbausteins ist "0".

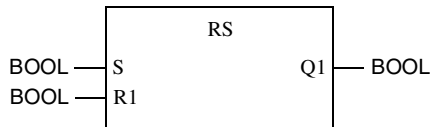
Als zusätzliche Parameter können EN und ENO projiziert werden.

Hinweis: Dieser Funktionsbaustein arbeitet mit einer internen Unlocated Variable und hat somit ein speicherndes-Verhalten. Das heißt, wenn der Ausgang "Q1" mit einem Hardwareausgang verbunden wird, bleibt beim Aus-/Einschalten der SPS der Ausgang auf Wert "1".

Darstellung

Symbol

Darstellung des Bausteins:



Parameterbeschreibung

Beschreibung der Bausteinparameter:

Parameter	Datentyp	Bedeutung
S	BOOL	Setzen
R1	BOOL	Rücksetzen (dominant)
Q1	BOOL	Ausgang

Kurzbeschreibung

Funktionsbeschreibung

Der Funktionsbaustein wird als SR-Speicher mit der Eigenschaft "Setzen dominant" verwendet.

Der Ausgang Q1 wird "1", wenn der Eingang S1 "1" wird. Dieser Zustand bleibt auch erhalten, wenn der Eingang S1 wieder "0" wird. Der Ausgang Q1 wird erst wieder "0", wenn der Eingang R "1" wird. Sind die Eingänge S1 und R gleichzeitig "1", setzt der dominierende Eingang S1 den Ausgang Q1 auf "1".

Der Anfangszustand von Q1 beim ersten Aufruf des Funktionsbausteins ist "0".

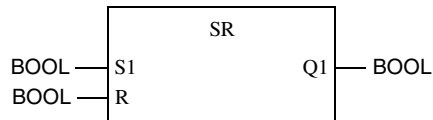
Als zusätzliche Parameter können EN und ENO projiziert werden.

Hinweis: Dieser Funktionsbaustein arbeitet mit einer internen Unlocated Variable und hat somit ein speicherndes-Verhalten. Das heißt, wenn der Ausgang "Q1" mit einem Hardwareausgang verbunden wird, bleibt beim Aus-/Einschalten der SPS der Ausgang auf Wert "1".

Darstellung

Symbol

Darstellung des Bausteins:



Parameterbeschreibung

Beschreibung der Bausteinparameter:

Parameter	Datentyp	Bedeutung
S1	BOOL	Setzen (dominant)
R	BOOL	Rücksetzen
Q	BOOL	Ausgang

Kurzbeschreibung

Funktionsbeschreibung

Der Funktionsbaustein wird als Ausschaltverzögerung verwendet.

Eine 0 -> 1 Flanke am Eingang IN löst einen Reset aus.

Eine 1 -> 0 Flanke am Eingang IN startet die Timerfunktion.

Erreicht die abgelaufene Zeit (Ausgang ET) den am Eingang PT vorgegebenen Wert, wird der Ausgang Q auf "0" gesetzt.

Der Anfangszustand von ET beim ersten Aufruf des Funktionsbausteins ist "0".

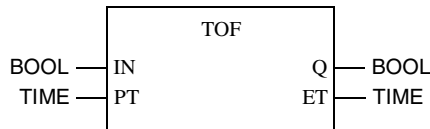
Als zusätzliche Parameter können EN und ENO projiziert werden.

Hinweis: Der Eingang EN kann nicht als Pause-Funktion für den Funktionsbaustein verwendet werden.
 Auch wenn der Eingang EN "0" wird, wird die abgelaufene Zeit weiter gemessen.
 Wenn der Eingang EN wieder "1" wird, wird der Ausgang ET aktualisiert und führt somit einen Sprung aus.
 Wenn Sie eine Pause-Funktion benötigen, steht Ihnen dafür der Funktionsbaustein TOF_P der Bausteinbibliothek EXTENDED zur Verfügung.

Darstellung

Symbol

Darstellung des Bausteins:



Parameterbeschreibung

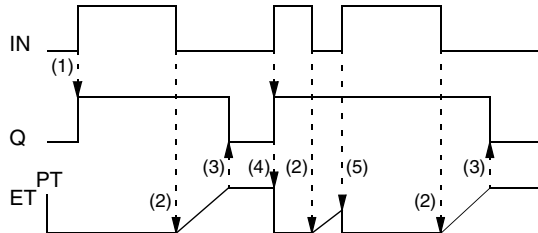
Beschreibung der Bausteinparameter:

Parameter	Datentyp	Bedeutung
IN	BOOL	0 -> 1: Reset 1 -> 0: Timer starten
PT	TIME	Voreinstellung der Verzögerungszeit
Q	BOOL	Ausgang
ET	TIME	abgelaufene Zeit

Detailbeschreibung

Taktdiagramm

Darstellung der Ausschaltverzögerung TOF:



- (1) Wenn IN "1" wird, wird Q "1".
- (2) Wenn IN "0" wird, wird die interne Zeit (ET) gestartet.
- (3) Erreicht die interne Zeit den Wert von PT, wird Q "0".
- (4) Wenn IN "1" wird, wird Q "1" und die interne Zeit gestoppt/rückgesetzt.
- (5) Wird IN "1" bevor die interne Zeit den Wert von PT erreicht hat, wird die interne Zeit gestoppt/rückgesetzt, ohne daß Q "0" wurde.

Kurzbeschreibung

Funktionsbeschreibung

Der Funktionsbaustein wird als Einschaltverzögerung verwendet.

Eine 1 -> 0 Flanke am Eingang IN löst einen Reset aus.

Eine 0 -> 1 Flanke am Eingang IN startet die Timerfunktion.

Erreicht die abgelaufene Zeit (Ausgang ET) den am Eingang PT vorgegebenen Wert, wird der Ausgang Q auf "1" gesetzt.

Der Anfangszustand von ET beim ersten Aufruf des Funktionsbausteins ist "0".

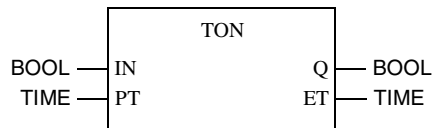
Als zusätzliche Parameter können EN und ENO projiziert werden.

Hinweis: Der Eingang EN kann nicht als Pause-Funktion für den Funktionsbaustein verwendet werden.
Auch wenn der Eingang EN "0" wird, wird die abgelaufene Zeit weiter gemessen.
Wenn der Eingang EN wieder "1" wird, wird der Ausgang ET aktualisiert und führt somit einen Sprung aus.
Wenn Sie eine Pause-Funktion benötigen, steht Ihnen dafür der Funktionsbaustein TON_P der Bausteinbibliothek EXTENDED zur Verfügung.

Darstellung

Symbol

Darstellung des Bausteins:



Parameterbeschreibung

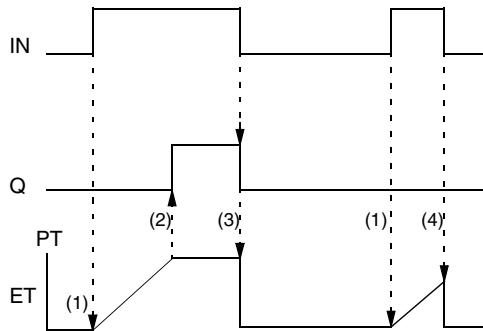
Beschreibung der Bausteinparameter:

Parameter	Datentyp	Bedeutung
IN	BOOL	0 -> 1: Timer starten 1 -> 0: Reset
PT	TIME	Voreinstellung der Verzögerungszeit
Q	BOOL	Ausgang
ET	TIME	abgelaufene Zeit

Detailbeschreibung

Taktdiagramm

Darstellung der Einschaltverzögerung TON:



- (1) Wenn IN "1" wird, wird die interne Zeit (ET) gestartet.
- (2) Erreicht die interne Zeit den Wert von PT, wird Q "1".
- (3) Wird IN "0", wird Q "0" und die interne Zeit gestoppt/rückgesetzt.
- (4) Wird IN "0" bevor die interne Zeit den Wert von PT erreicht hat, wird die interne Zeit gestoppt/rückgesetzt, ohne daß Q "1" wurde.