

SIMATIC

Automatisierungssystem S7-300

Beispielprogramme für Technologische Funktionen

Inhaltsverzeichnis

Beispielprogramme für die Technologie "Positionieren"	1
Beispielprogramme für die Technologie "Zählen, Frequenzmessen und Pulsweitenmodulation"	2
Beispielprogramme für die Technologie "Punkt-zu-Punkt-Kopplung"	3
Beispielprogramme für die Technologie "Regeln"	4
Index	

Diese Dokumentation ist Bestandteil des
Dokumentationspaketes mit der Bestellnummer:
6ES7398-8FA10-8AA0

Ausgabe 10/2001
A5E00130042-01

Sicherheitstechnische Hinweise

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise sind durch ein Warndreieck hervorgehoben und je nach Gefährungsgrad folgendermaßen dargestellt:



Gefahr

bedeutet, dass Tod, schwere Körperverletzung oder erheblicher Sachschaden eintreten **werden**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.



Warnung

bedeutet, dass Tod, schwere Körperverletzung oder erheblicher Sachschaden eintreten **können**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.



Vorsicht

bedeutet, dass eine leichte Körperverletzung oder ein Sachschaden eintreten können, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Vorsicht

bedeutet, dass ein Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Achtung

ist eine wichtige Information über das Produkt, die Handhabung des Produktes oder den jeweiligen Teil der Dokumentation, auf den besonders aufmerksam gemacht werden soll.

Qualifiziertes Personal

Inbetriebsetzung und Betrieb eines Gerätes dürfen nur von **qualifiziertem Personal** vorgenommen werden. Qualifiziertes Personal im Sinne der sicherheitstechnischen Hinweise dieses Handbuchs sind Personen, die die Berechtigung haben, Geräte, Systeme und Stromkreise gemäß den Standards der Sicherheitstechnik in Betrieb zu nehmen, zu erden und zu kennzeichnen.

Bestimmungsgemäßer Gebrauch

Beachten Sie Folgendes:



Warnung

Das Gerät darf nur für die im Katalog und in der technischen Beschreibung vorgesehenen Einsatzfälle und nur in Verbindung mit von Siemens empfohlenen bzw. zugelassenen Fremdgeräten und -komponenten verwendet werden.

Der einwandfreie und sichere Betrieb des Produktes setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung und Montage sowie sorgfältige Bedienung und Instandhaltung voraus.

Marken

SIMATIC®, SIMATIC HMI® und SIMATIC NET® sind Marken der Siemens AG.

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

Copyright © Siemens AG 2001 All rights reserved

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

Haftungsausschluss

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

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

Siemens Aktiengesellschaft

© Siemens AG 2001
Technische Änderungen bleiben vorbehalten

A5E00130042



Inhaltsverzeichnis

1	Beispielprogramme für die Technologie "Positionieren"	
1.1	Einführung	1-1
1.2	Positionieren mit Analogausgang	1-3
1.2.1	Analog 1 Erste Schritte	1-4
1.2.2	Analog 2 Betriebsarten	1-8
1.2.3	Analog 3 Ladeportal	1-11
1.3	Positionieren mit Digitalausgängen	1-16
1.3.1	Digital 1 Erste Schritte	1-17
1.3.2	Digital 2 Betriebsarten	1-21
1.3.3	Digital 3 Pufferspeicher	1-24
2	Beispielprogramme für die Technologie "Zählen, Frequenzmessen und Pulsweitenmodulation"	
2.1	Einführung	2-1
2.2	Zählen	2-3
2.2.1	Zählen 1 Erste Schritte	2-3
2.2.2	Zählen 2 Abfüllstation	2-9
2.3	Frequenzmessen	2-14
2.3.1	Frequenz 1 Erste Schritte	2-14
2.3.2	Frequenz 2 Drehzahlüberwachung mit variabler Sollwertvorgabe	2-19
2.4	Pulsweitenmodulation	2-23
2.4.1	PWM 1 Erste Schritte	2-23
2.4.2	PWM 2 Aufheizen einer Flüssigkeit	2-27
3	Beispielprogramme für die Technologie "Punkt-zu-Punkt-Kopplung"	
3.1	Einführung	3-1
3.2	Punkt-zu-Punkt-Kopplung	3-3
3.2.1	CPU 31xC ASCII	3-5
3.2.2	CPU 31xC ASCII BCC	3-6
3.2.3	CPU 31xC RK512	3-8
4	Beispielprogramme für die Technologie "Regeln"	
4.1	Einführung	4-1
4.2	Regeln	4-3
4.2.1	Regeln 1 CONT_S Schrittreger mit Streckensimulation	4-3
4.2.2	Regeln 2 CONT_C Kontinuierlicher Regler mit Streckensimulation	4-8
4.2.3	Regeln 3 PULSEGEN Kontinuierlicher Regler CONT_C mit nachgeschaltetem Impulsformer PULSEGEN und Streckensimulation....	4-13

Index

1 Beispielprogramme für die Technologie "Positionieren"

1.1 Einführung

Übersicht

In diesem Kapitel finden Sie Anwendungsbeispiele zur Technologie Positionieren.

Die Beispiele befinden sich auf der Ihrer Dokumentation beigelegten CD. Falls Ihnen die CD nicht vorliegt, können Sie die Beispiele auch über das Internet beziehen. Das Projekt besteht aus mehreren kommentierten S7-Programmen verschiedener Komplexität und Zielrichtung.

Die Installation der Beispiele ist in der Liesmich.wri der CD beschrieben. Nach der Installation befinden sich die Beispiele im Katalog
`...\\STEP7\\EXAMPLES\\ZDt26_03_TF_____31xC_Pos.`

Voraussetzungen

- Sie haben eine S7-Station bestehend aus einer Stromversorgungsbaugruppe sowie einer CPU 314C aufgebaut und verdrahtet.
- Auf Ihrem PG ist STEP 7 ($\geq$ V5.1 + Service Pack 2) korrekt installiert.
- Das PG ist an die CPU angeschlossen.
- Sie haben zur Sicherheit von Anlage und Bedienerpersonal Hardware-endschalter und NOT-AUS-Schalter installiert.

Vorbereiten der Beispiele

1. Öffnen Sie das Beispielprojekt **ZDt26_03_TF_____31xC_Pos** im Ordner **...\\STEP7\\EXAMPLES** mit dem SIMATIC Manager und kopieren Sie es unter einem geeigneten Namen in Ihr Projektverzeichnis (**Datei > Speichern unter**). Nutzen Sie die Darstellung **Ansicht > Detail** zur Anzeige der kompletten Informationen.
2. Fügen Sie in Ihr Projekt eine Station entsprechend Ihrem Hardware-Aufbau ein.
3. Wählen Sie ein Beispielprogramm aus und kopieren Sie das S7-Programm in diese Station.
4. Konfigurieren Sie die Hardware vollständig mit **HW Konfig** und wählen Sie dabei das TF-Submodul **Positionieren** aus.
5. Parametrieren Sie dieses Submodul entsprechend Ihrer Anlagenkonfiguration.
6. Speichern Sie die Hardware-Konfiguration und laden Sie diese in die CPU.
7. Laden Sie den Bausteinbehälter in die CPU.

Code der Beispiele

Die Beispiele sind in AWL/FUP geschrieben. Sie können sie direkt über den KOP/AWL/ FUP-Editor anschauen.

Wählen Sie in diesem Editor über **Ansicht > Anzeigen mit** "Symbolischer Darstellung", "Symbolauswahl" und "Kommentar". Wenn Sie genügend Platz auf dem Bildschirm haben, können Sie sich auch noch die "Symbolinformation" anzeigen lassen.

Anwenden eines Beispiels

In den Beispielprogrammen sind Variablentabellen (VAT) vorhanden, mit denen Sie die Werte ansehen und verändern können.

Um die Fahrtsignale (z. B. Tippen) einfacher steuern zu können, schließen Sie Taster oder Schalter an ein Eingangsbyte an.

z. B:	L EB 0	// Tast-/Schaltelemente für Steuern der Achse
	T DB6.DBB4	// Steuersignale für Beispiel 1

Weiterverwenden eines Beispiels

Sie können den Code der Beispiele direkt als Anwenderprogramm weiterverwenden.

Der Code der Beispiele ist jedoch nicht optimiert und auch nicht für alle Möglichkeiten ausgelegt. Fehlerauswertungen sind in den Beispielprogrammen nicht ausführlich programmiert, um die Programme nicht zu umfangreich werden zu lassen.

Das Beispielprogramm "Betriebsarten" ist als Kopiervorlage verwendbar, in dem Sie durch Verändern und Streichen die Funktionen zusammenstellen können, die dann das Grundgerüst für Ihr Anwenderprogramm bilden.

1.2 Positionieren mit Analogausgang

Für die Positionierungsart finden Sie drei kommentierte S7-Programme:

- **Analog 1 Erste Schritte**
- **Analog 2 Betriebsarten**
- **Analog 3 Ladeportal**

Im Beispiel "Analog 1 Erste Schritte" wird nur der SFB aufgerufen und mit Parametern versorgt. Sie lernen vor allem die Parametrierung und die Funktionen des SFB ANALOG kennen.

Das Beispiel "Analog 2 Betriebsarten" baut auf das vorherige Beispiel auf. Hier sehen Sie, wie man den SFB versorgen kann, um die einzelnen Betriebsarten zu realisieren.

Das Beispiel "Analog 3 Ladeportal" ist eine typische Anwendung und besteht aus folgenden Teilen:

- Schrittkette für den Automatikablauf,
- Betriebsartenteil mit Positionszuordnung/Berechnung,
- Aufruf des Systemfunktionsbausteins (SFB),
- Auswertung "Achse in Position" (POS_RCD).

Es baut ebenfalls auf die vorherigen Beispiele "Erste Schritte" und "Betriebsarten" auf und zeigt die Ausarbeitung einer S7-Anwendung mit Positionierungsaufgabe.

Bausteinstruktur

Für die Positionierart ANALOG sind folgende Bausteine vorbereitet:

Baustein	Name (in der Symbolleiste)	Beschreibung
OB 1	CYCLE_EXC	Zyklisches Programm
OB 100	COMPLETE RESTART	Neustart: Rücksetzen der Steuersignale
FC 1	GETST_A	Beispiel 1: Erste Schritte ANALOG
FB 2	MODE_A	Beispiel 2: Betriebsarten
DB 2	DI_MODE_A	Beispiel 2: Instanz-DB zu MODE_A
FB 3	PORTAL	Beispiel 3: Ladeportal
DB 3	DI_PORTAL	Beispiel 3: Instanz-DB zu PORTAL
SFB 44	ANALOG	Systemfunktionsbaustein ANALOG
DB 6	DI_ANALOG	Instanz-DB zu SFB ANALOG

1.2.1 Analog 1 Erste Schritte

Ziel

Mit diesem Beispiel nehmen Sie die Technologie **"Positionieren mit Analogausgang"** in Betrieb.

Sie lernen die Parametrierung des Systemfunktionsbausteins (SFB) ANALOG kennen.

Sie haben die Möglichkeit, folgende Betriebsarten durch Anlegen der entsprechenden Parameter am SFB zu testen:

- Tippen
- Referenzpunktfahren
- Schrittmaßfahrt relativ
- Schrittmaßfahrt absolut

Voraussetzungen

- Sie haben die Parametrierung mit Hilfe der Parametriermasken durchgeführt.
- Sie haben im Baustein COMPLETE RESTART (OB100) die Werte für Geschwindigkeit, Beschleunigung, Verzögerung, Um- und Abschalt Differenz angepasst.
- Es liegt kein Prozessfehler an (ERR = 0). Anliegende Prozessfehler müssen mit **ERR_A = 1** quittiert werden.

Anlauf

Im Anlauf-OB (OB 100) werden die Steuersignale, die z. B. über die Variablentabelle beeinflusst werden können, zurückgesetzt.

Bedienung "Betriebsart Tippen"

Die Parameter des SFB ANALOG können Sie über die vorbereitete Variablentabelle VAT_GETST_A ändern.

Setzen und aktivieren Sie folgende Eingangsparameter des SFB:

Parameter	Einstellung	Bedeutung
MODE_IN	1	Betriebsart 1 = Tippen
DRV_EN	1	Antriebsfreigabe einschalten
SPEED	xx xxx xxx	Geschwindigkeit in Impulse/s eintragen

Zum Starten des Antriebs mit unterschiedlichen Richtungen müssen Sie entweder den Eingangsparameter **DIR_P** oder **DIR_M** setzen und aktivieren.

Parameter	Einstellung	Bedeutung	Einstellung	Bedeutung
DIR_P	1	Tippen in Richtung Plus	0	Tippen in Richtung Minus
DIR_M	0		1	

Rückmeldungen:

Parameter	Wert	Bedeutung
ST_ENBLD	1	Startfreigabe
MODE_OUT	1	Aktive/eingestellte Betriebsart "Tippen"
WORKING	1	Fahrt läuft
ACT_POS	xx xxx xxx	Aktueller Lageistwert
ERROR	1	Fehler beim Starten/Fortsetzen einer Fahrt
STATUS	xxxx	Fehlernummer
ERR	xxxx	Prozessfehler

Bedienung "Betriebsart Referenzpunktfahren"

Setzen und aktivieren Sie folgende Eingangsparameter des SFB:

Parameter	Einstellung	Bedeutung
MODE_IN	3	Betriebsart 3 = Referenzpunktfahren
DRV_EN	1	Antriebsfreigabe einschalten
SPEED	xx xxx xxx	Startgeschwindigkeit in Impulse/s

Zum Starten des Antriebs mit unterschiedlichen Richtungen müssen Sie entweder den Eingangsparameter **DIR_P** oder **DIR_M** setzen und aktivieren.

Parameter	Einstellung	Bedeutung	Einstellung	Bedeutung
DIR_P	1	Referenzpunktfahrt in Richtung Plus	0	Referenzpunktfahrt in Richtung Minus
DIR_M	0		1	

Rückmeldungen:

Parameter	Wert	Bedeutung
ST_ENBLD	1	Startfreigabe
MODE_OUT	3	Aktive/eingestellte Betriebsart "Referenzpunktfahren"
WORKING	1	Fahrt läuft
SYNC	1	Achse ist synchronisiert
ACT_POS	xx xxx xxx	Aktueller Lageistwert
ERROR	1	Fehler beim Starten/Fortsetzen einer Fahrt
STATUS	xxxx	Fehlernummer
ERR	xxxx	Prozessfehler

Bedienung "Betriebsart Schrittmäßfahrt relativ"

Setzen und aktivieren Sie folgende Eingangsparameter des SFB:

Parameter	Einstellung	Bedeutung
MODE_IN	4	Betriebsart 4 = Schrittmäßfahrt relativ
DRV_EN	1	Antriebsfreigabe einschalten
TARGET	xx xxx xxx	Ziel/Wegstück in Impulse
SPEED	xx xxx xxx	Startgeschwindigkeit in Impulse/s

Zum Starten des Antriebs mit unterschiedlichen Richtungen müssen Sie entweder den Eingangsparameter **DIR_P** oder **DIR_M** setzen und aktivieren.

Parameter	Einstellung	Bedeutung	Einstellung	Bedeutung
DIR_P	1	Schrittmäßfahrt rel. in Richtung Plus	0	Schrittmäßfahrt rel.
DIR_M	0		1	in Richtung Minus

Rückmeldungen:

Parameter	Wert	Bedeutung
ST_ENBLD	1	Startfreigabe
MODE_OUT	4	Aktive/eingestellte Betriebsart "Schrittmäßfahrt relativ"
WORKING	1	Fahrt läuft
POS_RCD	1	Position erreicht
ACT_POS	xx xxx xxx	Aktueller Lageistwert
ERROR	1	Fehler beim Starten/Fortsetzen einer Fahrt
STATUS	xxxx	Fehlernummer
ERR	xxxx	Prozessfehler

Bedienung "Betriebsart Schrittmäßfahrt absolut"

Voraussetzung:

Die Achse ist synchronisiert (**SYNC** = TRUE)

Setzen und aktivieren Sie folgende Eingangsparameter des SFB:

Parameter	Einstellung	Bedeutung
MODE_IN	5	Betriebsart 5 = Schrittmäßfahrt absolut
DRV_EN	1	Antriebsfreigabe einschalten
TARGET	xx xxx xxx	Ziel/Wegstück in Impulse
SPEED	xx xxx xxx	Startgeschwindigkeit in Impulse/s

Linearachse

Zum Starten des Antriebs müssen Sie den Eingangsparameter **START** setzen und aktivieren.

Parameter	Einstellung	Bedeutung
START	1	Schrittmäßfahrt absolut

Rundachse

Zum Starten des Antriebs müssen Sie entweder den Eingangsparameter **START**, **DIR_P** oder **DIR_M** setzen und aktivieren.

Parameter	Einstellung	Bedeutung
START	1	Schrittmäßfahrt absolut auf "kürzestem Weg" zum Ziel
DIR_P	1	Schrittmäßfahrt absolut in "Richtung Plus" zum Ziel
DIR_M	1	Schrittmäßfahrt absolut in "Richtung Minus" zum Ziel

Rückmeldungen:

Parameter	Wert	Bedeutung
ST_ENBLD	1	Startfreigabe
MODE_OUT	5	Aktive/eingestellte Betriebsart "Schrittmäßfahrt absolut"
WORKING	1	Fahrt läuft
POS_RCD	1	Position erreicht
ACT_POS	xx xxx xxx	Aktueller Lageistwert
ERROR	1	Fehler beim Starten/Fortsetzen einer Fahrt
STATUS	xxxx	Fehlernummer
ERR	xxxx	Prozessfehler

1.2.2 Analog 2 Betriebsarten

Ziel

In diesem Beispiel wird gezeigt, wie Sie den SFB ANALOG versorgen müssen, um die einzelnen Betriebsarten zu realisieren.

Wenn Sie in Ihrer Anlage eine Achse einsetzen, erzeugen Sie die Steuersignale für Tippen, Referenzpunktfahren, Schrittmaßfahren meist durch eine Verknüpfungslogik oder eine Schrittkette. Diese Steuersignale können Sie in diesem Beispiel direkt zur Ansteuerung einer Fahrt benutzen.

Um eine Fahrt auszuführen, brauchen Sie nur **ein Bit** anzusteuern.

Sie sehen auch, wie die Parametrierung der Aufträge "Bezugspunkt setzen" und "Längenmessung" realisiert wird.

Voraussetzungen

- Sie haben die Parametrierung mit Hilfe der Parametriermasken durchgeführt.
- Sie haben im Baustein COMPLETE RESTART (OB100) die Werte für Geschwindigkeit, Beschleunigung, Verzögerung, Um- und Abschalt Differenz angepasst.
- Es liegt kein Prozessfehler an (ERR=0). Anliegende Prozessfehler müssen mit **ERR_A = 1** quittiert werden.
- Für die Betriebsart "Schrittmaßfahren absolut" muss die Achse synchronisiert sein.
- Sie haben für die Funktion "Längenmessung" die Art der Flanke mit dem Parameter "Längenmessung" über die Parametriermasken eingestellt.

Programmaufbau

Das Programm Betriebsarten (MODE_A) hat folgenden Aufbau:

- Reset der Startsignale
- Betriebsartenparameter für Referenzpunktfahren
- Betriebsartenparameter für Tippen
- Betriebsartenparameter für Schrittmaßfahrt relativ
- Betriebsartenparameter für Schrittmaßfahrt absolut
- Auftrag Bezugspunkt setzen
- Aufruf des SFB ANALOG

Bedienung "Betriebsarten"

Tragen Sie die gewünschten Positionswerte der Schrittmaßfahrten und die Geschwindigkeit für schnelle und langsame Fahrt in die Variablentabelle VAT_MODE_A ein. Setzen und aktivieren Sie den Parameter S_DRV_EN.

Parameter	Einstellung	Bedeutung
Pos1a	xx xxx xxx	Sollwert: Positionswert 1 absolut
Pos2a	xx xxx xxx	Sollwert: Positionswert 2 absolut
Pos1r	xx xxx xxx	Sollwert: Positionswert 1 relativ
Pos2r	xx xxx xxx	Sollwert: Positionswert 2 relativ
Speed_FAST	xx xxx xxx	Sollwert: Geschwindigkeit schnell
Speed_SLOW	xx xxx xxx	Sollwert: Geschwindigkeit langsam
Pos_REFP	xx xxx xxx	Sollwert: Bezugspunkt
S_DRV_EN	1	Steuersignal: Antriebsfreigabe

Durch Setzen und Aktivieren **eines** der folgenden Steuersignale starten Sie die entsprechende Funktion:

Parameter	Einstellung	Bedeutung
S_REF	1	Start Referenzpunktfahrt
S_DIR_PF	1	Start Tippen vorwärts schnell
S_DIR_MF	1	Start Tippen rückwärts schnell
S_DIR_PS	1	Start Tippen vorwärts langsam
S_DIR_MS	1	Start Tippen rückwärts langsam
S_POS1a	1	Start Pos 1 anfahren (absolut)
S_POS2a	1	Start Pos 2 anfahren (absolut)
S_POS1rP	1	Start Pos 1 anfahren (relativ vorwärts)
S_POS1rM	1	Start Pos 2 anfahren (relativ rückwärts)
S_REFP_EN	1	Auftrag Bezugspunkt setzen

Rückmeldungen:

Parameter	Wert	Bedeutung
ST_ENBLD	1	Startfreigabe
MODE_OUT	x	Aktive/eingestellte Betriebsart
WORKING	1	Fahrt läuft
POS_RCD	1	Position erreicht
SYNC	1	Achse synchronisiert
ACT_POS	xx xxx xxx	Aktueller Lageistwert
ERROR	1	Fehler beim Starten/Fortsetzen einer Fahrt
STATUS	xxxx	Fehlernummer
ERR	xxxx	Prozessfehler

Längenmessung

Die Funktion "Längenmessung" wird direkt mit dem Eingang (DI + 0.3) "Längenmessung" gesteuert.

Rückmeldungen:

Parameter	Wert	Bedeutung
MSR_DONE	1	Längenmessung beendet
BEG_VAL	1	Lageistwert Längenmessung Beginn
END_VAL	1	Lageistwert Längenmessung Ende
LEN_VAL	1	Gemessene Länge

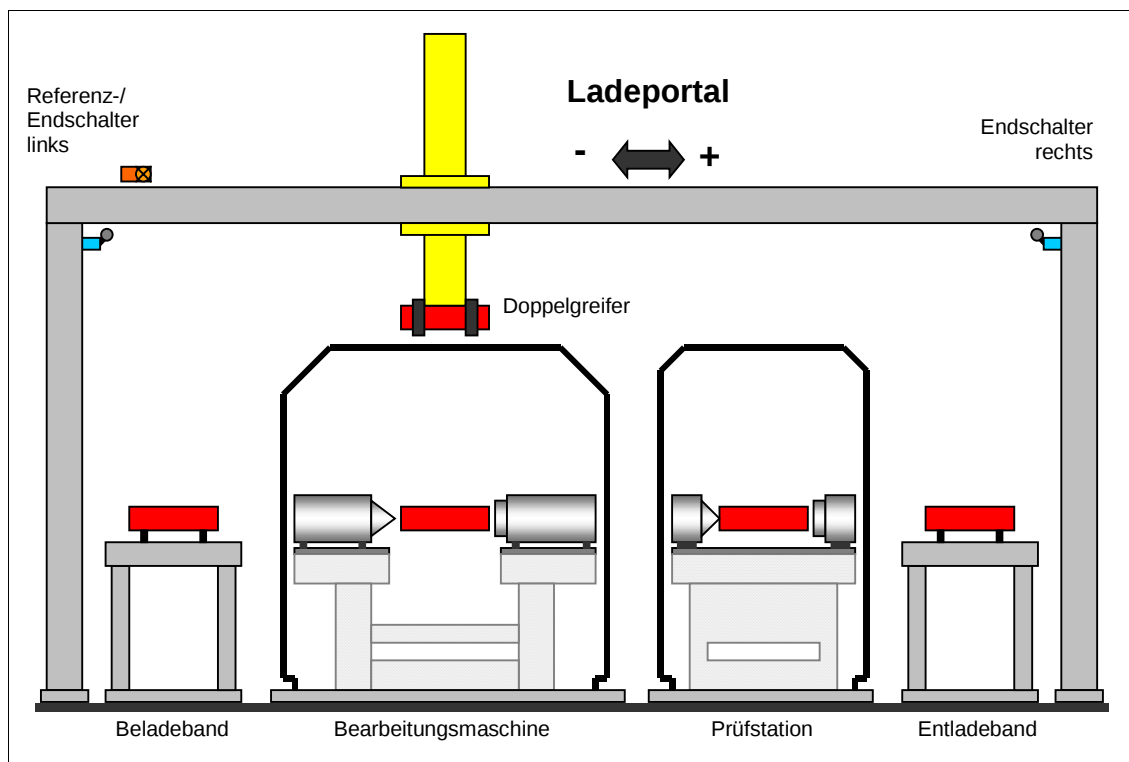
1.2.3 Analog 3 Ladeportal

Ziel

Mit dem vorliegenden Beispiel soll eine Anwendung für die Technologie **"Positionieren mit Analogausgang"** gezeigt werden. Hier werden die Steuersignale für den Automatikbetrieb durch eine Schrittkette erzeugt. Handfunktionen wie Tippen oder Schrittmaßfahren sind ebenfalls möglich.

Aufgabenstellung

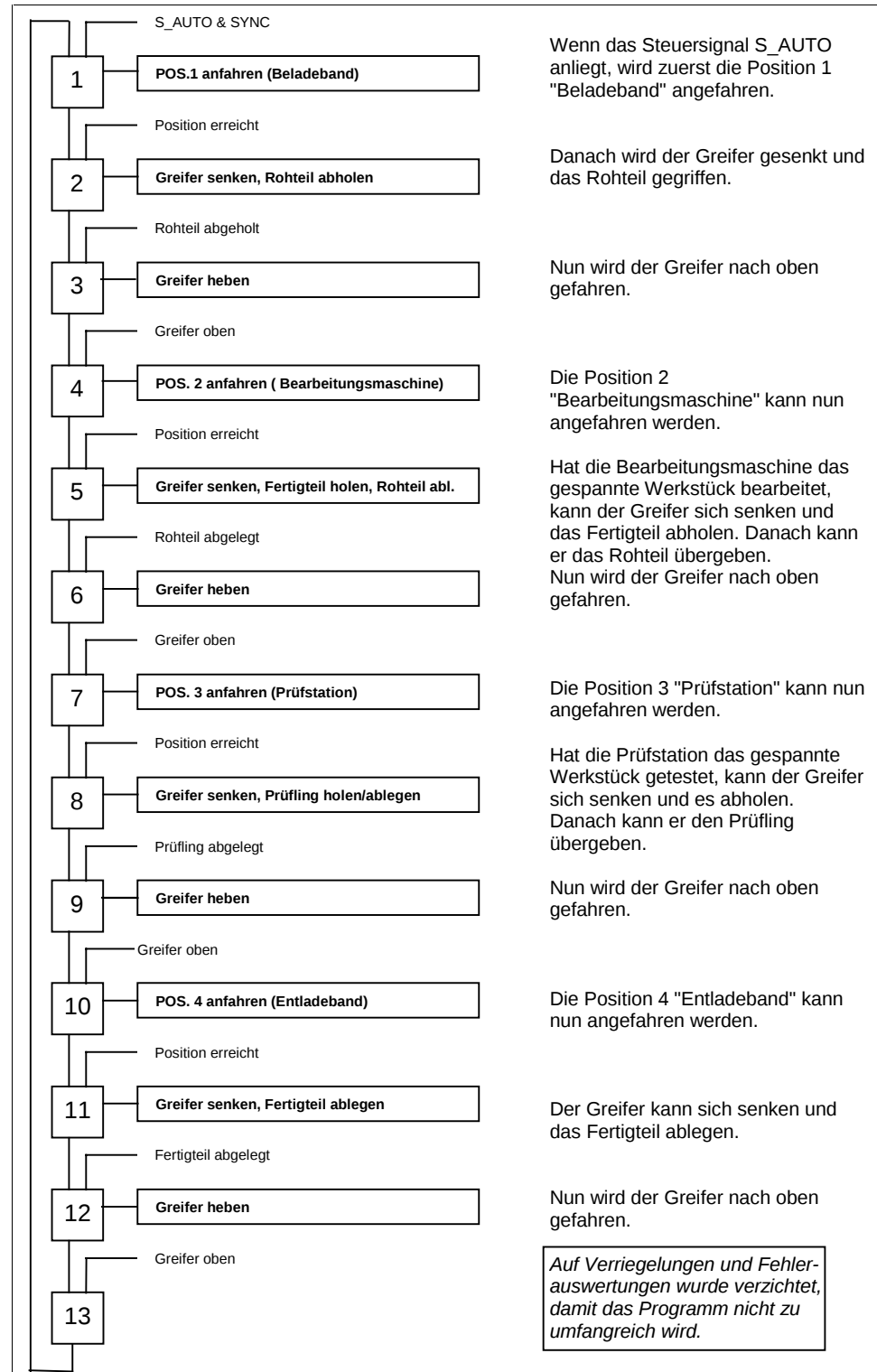
Ein Ladeportal hat die Aufgabe die zu bearbeitenden Werkstücke vom Beladeband abzuholen, in die Bearbeitungsmaschine und Prüfstation zu befördern und anschließend auf dem Entladeband wieder abzulegen.



Das Ladeportal ist mit einem Doppelgreifer (Greifer 1: Rohteil/Greifer 2: Fertigteil) ausgestattet. Der Greifer wird horizontal durch einen **Servomotor**, vertikal durch einen Pneumatikzylinder bewegt.

Ablaufdiagramm

Wenn die Achse nicht synchronisiert ist, muss vor dem Starten zuerst die Referenzpunktfahrt ausgeführt werden.



Umrechnung des Weges von mm in Impulse (Weginkremente)

Der Inkrementalgeber liefert 2500 Inkremente pro Geberumdrehung und ist nach dem Getriebe angebaut. Das Ladeportal bewegt sich bei einer Umdrehung 100 mm.

$100 \text{ mm} / 2500 \text{ Inkremente} = 0,040 \text{ mm}$

1 Inkrement = 4 Impulse

$0,040 \text{ mm} / 4 = 0,010 \text{ mm}$

Ein Impuls entspricht somit einer Wegstrecke von 0,010 mm.

Zielpositionen

Es werden folgende Positionssollwerte benötigt:

Zielpositionen	Umrechnung in Impulse (Weginkremente)
Position 1 Beladeband	$100 \text{ mm} / 0,01 \text{ mm/Imp.} = \mathbf{10\ 000}$ Impulse
Position 2 Bearbeitungsmaschine	$1800 \text{ mm} / 0,01 \text{ mm/Imp.} = \mathbf{180\ 000}$ Impulse
Position 3 Prüfstation	$3000 \text{ mm} / 0,01 \text{ mm/Imp.} = \mathbf{300\ 000}$ Impulse
Position 4 Entladeband	$4500 \text{ mm} / 0,01 \text{ mm/Imp.} = \mathbf{450\ 000}$ Impulse

Parametrieren mit STEP 7

Mit STEP 7 parametrieren Sie die CPU wie folgt:

Parameter	Eingabe
Technologieart	Positionieren mit Analogausgang
Zielbereich	100 Impulse ($1 \text{ mm} / 0,01 \text{ mm/Impulse} = 100$)
Überwachungszeit	2000 ms
Maximalgeschwindigkeit	10000 Impulse/s
Schleich-/Referenziergeschwindigkeit	100 Impulse/s
Achsart	Linearachse
Software-Endschalter Anfang	-5000 Impulse Arbeitsbereichsgrenze Anfang ($-50 \text{ mm} / 0,01 \text{ mm/Impulse} = -5000 \text{ Impulse}$)
Software-Endschalter Ende	500 000 Impulse Arbeitsbereichsgrenze Ende ($5000 \text{ mm} / 0,01 \text{ mm/Impulse} = 500\ 000 \text{ Impulse}$)
Referenzpunktlage zu Referenzpunktschalter	In Richtung Plus
Inkremente pro Geberumdrehung	2500
Zählrichtung	Normal (Spurwechsel Spur A vor B = Istwert positiv)

Um-/Abschaltdifferenz

Die **Umschaltdifferenz** definiert den Umschaltpunkt, ab dem der Antrieb mit Schleichgang verfährt.

Diese Umschaltdifferenz ist im Beispiel mit 50 mm festgelegt.

50 mm / 0,010 mm/Imp. = **5000** Impulse

Die **Abschaltdifferenz** ist die Wegstrecke vom Abschaltpunkt bis zum Ziel. Der Antrieb wird im Abschaltpunkt abgeschaltet.

Damit die Zielposition so genau wie möglich getroffen wird, müssen Sie so vorgehen:

1. Geben Sie über das Anwenderprogramm eine Abschaltdifferenz $\geq 1/2$ Zielbereich vor.
2. Fahren Sie den Lader in der Betriebsart "Schrittmaßfahrt absolut" einmal auf eine Position.
3. Ermitteln Sie die Differenz zwischen tatsächlich erreichter Zielposition und vorgegebener Zielposition.
4. Addieren Sie diese Differenz zur Abschaltdifferenz.

Ermittelt wurde eine Differenz von 60 Impulsen für links und rechts.

Im Anwenderprogramm werden folgende Eintragungen vorgenommen:

Umschaltdifferenz +	=	5000	Impulse
Umschaltdifferenz -	=	5000	Impulse
Abschaltdifferenz +	=	60	Impulse
Abschaltdifferenz -	=	60	Impulse

Hinweis

Die Zielposition sollten Sie aus anderer Richtung evtl. erneut anfahren und die Abschaltdifferenz überprüfen

Bedienung "Ladeportal"

Tragen Sie die gewünschten Positionswerte der Schrittmaßfahrten und die Geschwindigkeit für schnelle und langsame Fahrt in die Variablentabelle VAT_PORTAL ein. Setzen und aktivieren Sie den Parameter S_DRV_EN.

Parameter	Einstellung	Bedeutung
Pos1	xx xxx xxx	Sollwert: Positionswert 1 (Beladeband)
Pos2	xx xxx xxx	Sollwert: Positionswert 2 (Bearbeitungsmaschine)
Pos3	xx xxx xxx	Sollwert: Positionswert 3 (Prüfstation)
Pos4	xx xxx xxx	Sollwert: Positionswert 4 (Entladeband)
Speed_FAST	xx xxx xxx	Sollwert: Geschwindigkeit schnell
Speed_SLOW	xx xxx xxx	Sollwert: Geschwindigkeit langsam
S_DRV_EN	1	Steuersignal: Antriebsfreigabe

Falls die Achse nicht synchronisiert ist, stellen Sie die Anlage auf Handbetrieb (S_AUTO = 0) und starten Sie die Referenzpunktfahrt.

Parameter	Einstellung	Bedeutung
S_AUTO	0	Handbetrieb = 0 / Automatikbetrieb = 1
S_REF	1	Start Referenzpunktfahrt

Automatikbetrieb können Sie mit folgendem Signal starten:

Parameter	Einstellung	Bedeutung
S_AUTO	1	Automatikbetrieb

Die folgenden Steuersignale können Sie setzen und aktivieren, wenn der Automatikbetrieb nicht gestartet ist. (S_AUTO = 0 → Handbetrieb)

Parameter	Einstellung	Bedeutung
S_REF	1	Start Referenzpunktfahrt
S_DIR_PF	1	Start Tippen vorwärts schnell
S_DIR_MF	1	Start Tippen rückwärts schnell
S_DIR_PS	1	Start Tippen vorwärts langsam
S_DIR_MS	1	Start Tippen rückwärts langsam
S_POS1	1	Start Pos 1 anfahren (Beladeband)
S_POS2	1	Start Pos 2 anfahren (Bearbeitungsmaschine)
S_POS3	1	Start Pos 3 anfahren (Prüfstation)
S_POS4	1	Start Pos 4 anfahren (Endladeband)

Rückmeldungen:

Parameter	Wert	Bedeutung
ST_ENBLD	1	Startfreigabe
MODE_OUT	x	Aktive/eingestellte Betriebsart
WORKING	1	Fahrt läuft
POS_RCD	1	Position erreicht
SYNC	1	Achse synchronisiert
ACT_POS	xx xxx xxx	Aktueller Lageistwert
ERROR	1	Fehler beim Starten/Fortsetzen einer Fahrt
STATUS	xxxx	Fehlernummer
ERR	xxxx	Prozessfehler

1.3 Positionieren mit Digitalausgängen

Für die Positionierungsart finden Sie drei kommentierte S7-Programme:

- **Digital 1 Erste Schritte**
- **Digital 2 Betriebsarten**
- **Digital 3 Pufferspeicher**

Im Beispiel "Digital 1 Erste Schritte" wird nur der SFB aufgerufen und mit Parametern versorgt. Sie lernen vor allem die Parametrierung und die Funktionen des SFB DIGITAL kennen.

Das Beispiel "Digital 2 Betriebsarten" baut auf das vorherige Beispiel auf. Hier sehen Sie, wie man den SFB versorgen kann, um die einzelnen Betriebsarten zu realisieren.

Das Beispiel "Digital 3 Pufferspeicher" ist eine typische Anwendung und besteht aus folgenden Teilen:

- einer Schrittkette für den Automatikablauf,
- einem Betriebsartenteil mit Positionszuordnung/Berechnung,
- dem Aufruf des Systemfunktionsbausteins (SFB),
- und der Auswertung "Achse in Position" (POS_RCD).

Es baut ebenfalls auf die vorherigen Beispiele "Erste Schritte" und "Betriebsarten" auf und zeigt die Ausarbeitung einer S7-Anwendung mit Positionierungsaufgabe.

Bausteinstruktur

Für die Positionierart STEP sind folgende Bausteine vorbereitet:

Baustein	Name (in der Symbolleiste)	Beschreibung
OB 1	CYCLE_EXC	Zyklisches Programm
OB 100	COMPLETE RESTART	Neustart: Rücksetzen der Steuersignale
FC 1	GETST_D	Beispiel 1: Erste Schritte DIGITAL
FB 2	MODE_D	Beispiel 2: Betriebsarten
DB 2	DI_MODE_D	Beispiel 2: Instanz-DB zu MODE_S
FB 3	STORAGE	Beispiel 3: Pufferspeicher
DB 3	DI_STORAGE	Beispiel 3: Instanz-DB zu Pufferspeicher
SFB 46	DIGITAL	Systemfunktionsbaustein DIGITAL
DB 6	DI_DIGITAL	Instanz-DB zu SFB DIGITAL

1.3.1 Digital 1 Erste Schritte

Ziel

Mit diesem Beispiel nehmen Sie die Technologie **"Positionieren mit Digitalausgang"** in Betrieb.

Sie lernen die Parametrierung des Systemfunktionsbausteins (SFB) DIGITAL kennen.

Sie haben die Möglichkeit, folgende Betriebsarten durch Anlegen der entsprechenden Parameter am SFB zu testen:

- Tippen
- Referenzpunktfahren
- Schrittmaßfahrt relativ
- Schrittmaßfahrt absolut

Voraussetzungen

- Sie haben die Parametrierung mit Hilfe der Parametriermasken durchgeführt.
- Sie haben im Baustein COMPLETE RESTART (OB 100) die Werte für Um- und Abschalt Differenz angepasst.
- Es liegt kein Prozessfehler an (ERR = 0). Anliegende Prozessfehler müssen mit **ERR_A = 1** quittiert werden.

Anlauf:

Im Anlauf-OB (OB 100) werden die Steuersignale, die z. B. über die Variablen Tabelle beeinflusst werden können, zurückgesetzt.

Bedienung "Betriebsart Tippen"

Die Parameter des SFB DIGITAL können Sie über die vorbereitete Variablen Tabelle VAT_GETST_D ändern.

Setzen und aktivieren Sie folgende Eingangsparameter des SFB:

Parameter	Einstellung	Bedeutung
MODE_IN	1	Betriebsart 1 = Tippen
DRV_EN	1	Antriebsfreigabe einschalten
SPEED	0/1	Geschwindigkeit 0 = Schleichgang, 1 = Eilgang

Zum Starten des Antriebs mit unterschiedlichen Richtungen müssen Sie entweder den Eingangsparameter **DIR_P** oder **DIR_M** setzen und aktivieren.

Parameter	Einstellung	Bedeutung	Einstellung	Bedeutung
DIR_P	1	Tippen in Richtung Plus	0	Tippen in Richtung Minus
DIR_M	0		1	

Rückmeldungen:

Parameter	Wert	Bedeutung
ST_ENBLD	1	Startfreigabe
MODE_OUT	1	Aktive/eingestellte Betriebsart "Tippen"
WORKING	1	Fahrt läuft
ACT_POS	xx xxx xxx	Aktueller Lageistwert
ERROR	1	Fehler beim Starten/Fortsetzen einer Fahrt
STATUS	xxxx	Fehlernummer
ERR	xxxx	Prozessfehler

Bedienung "Betriebsart Referenzpunktfahren"

Setzen und aktivieren Sie folgende Eingangsparameter des SFB:

Parameter	Einstellung	Bedeutung
MODE_IN	3	Betriebsart 3 = Referenzpunktfahren
DRV_EN	1	Antriebsfreigabe einschalten
SPEED	0/1	Startgeschwindigkeit 0 = Schleichgang, 1 = Eilgang

Zum Starten des Antriebs mit unterschiedlichen Richtungen müssen Sie entweder den Eingangsparameter **DIR_P** oder **DIR_M** setzen und aktivieren.

Parameter	Einstellung	Bedeutung	Einstellung	Bedeutung
DIR_P	1	Referenzpunktfahrt in Richtung Plus	0	Referenzpunktfahrt in Richtung Minus
DIR_M	0		1	

Rückmeldungen:

Parameter	Wert	Bedeutung
ST_ENBLD	1	Startfreigabe
MODE_OUT	3	Aktive/eingestellte Betriebsart "Referenzpunktfahren"
WORKING	1	Fahrt läuft
SYNC	1	Achse ist synchronisiert
ACT_POS	xx xxx xxx	Aktueller Lageistwert
ERROR	1	Fehler beim Starten/Fortsetzen einer Fahrt
STATUS	xxxx	Fehlernummer
ERR	xxxx	Prozessfehler

Bedienung "Betriebsart Schrittmäßfahrt relativ"

Setzen und aktivieren Sie folgende Eingangsparameter des SFB:

Parameter	Einstellung	Bedeutung
MODE_IN	4	Betriebsart 4 = Schrittmäßfahrt relativ
DRV_EN	1	Antriebsfreigabe einschalten
TARGET	xx xxx xxx	Ziel/Wegstück in Impulse
SPEED	0/1	Geschwindigkeit 0 = Schleichgang, 1 = Eilgang

Zum Starten des Antriebs mit unterschiedlichen Richtungen müssen Sie entweder den Eingangsparameter **DIR_P** oder **DIR_M** setzen und aktivieren.

Parameter	Einstellung	Bedeutung	Einstellung	Bedeutung
DIR_P	1	Schrittmäßfahrt rel. in Richtung Plus	0	Schrittmäßfahrt rel.
DIR_M	0		1	in Richtung Minus

Rückmeldungen:

Parameter	Wert	Bedeutung
ST_ENBLD	1	Startfreigabe
MODE_OUT	4	Aktive/eingestellte Betriebsart "Schrittmäßfahrt relativ"
WORKING	1	Fahrt läuft
POS_RCD	1	Position erreicht
ACT_POS	xx xxx xxx	Aktueller Lageistwert
ERROR	1	Fehler beim Starten/Fortsetzen einer Fahrt
STATUS	xxxx	Fehlernummer
ERR	xxxx	Prozessfehler

Bedienung "Betriebsart Schrittmäßfahrt absolut"

Voraussetzung:

Die Achse ist synchronisiert (**SYNC** = TRUE)

Setzen und aktivieren Sie folgende Eingangsparameter des SFB:

Parameter	Einstellung	Bedeutung
MODE_IN	5	Betriebsart 5 = Schrittmäßfahrt absolut
DRV_EN	1	Antriebsfreigabe einschalten
TARGET	xx xxx xxx	Ziel/Wegstück in Impulse
SPEED	0/1	Geschwindigkeit 0 = Schleichgang, 1 = Eilgang

Linearachse

Zum Starten des Antriebs müssen Sie den Eingangsparameter **START** setzen und aktivieren.

Parameter	Einstellung	Bedeutung
START	1	Schrittmaßfahrt absolut

Rundachse

Zum Starten des Antriebs müssen Sie einen der Eingangsparameter **START**, **DIR_P** oder **DIR_M** setzen und aktivieren.

Parameter	Einstellung	Bedeutung
START	1	Schrittmaßfahrt absolut auf "kürzestem Weg" zum Ziel
DIR_P	1	Schrittmaßfahrt absolut in "positiver Richtung" zum Ziel
DIR_M	1	Schrittmaßfahrt absolut in "negativer Richtung" zum Ziel

Rückmeldungen:

Parameter	Wert	Bedeutung
ST_ENBLD	1	Startfreigabe
MODE_OUT	5	Aktive/eingestellte Betriebsart "Schrittmaßfahrt absolut"
WORKING	1	Fahrt läuft
POS_RCD	1	Position erreicht
ACT_POS	xx xxx xxx	Aktueller Lageistwert
ERROR	1	Fehler beim Starten/Fortsetzen einer Fahrt
STATUS	xxxx	Fehlernummer
ERR	xxxx	Prozessfehler

1.3.2 Digital 2 Betriebsarten

Ziel

In diesem Beispiel wird gezeigt, wie Sie den SFB DIGITAL versorgen müssen, um die einzelnen Betriebsarten zu realisieren.

Wenn Sie in Ihrer Anlage eine Achse einsetzen, erzeugen Sie die Steuersignale für Tippen, Referenzpunktfahren, Schrittmaßfahren meist durch eine Verknüpfungslogik oder eine Schrittkette. Diese Steuersignale können Sie in diesem Beispiel direkt zur Ansteuerung einer Fahrt benutzen.

Um eine Fahrt auszuführen brauchen Sie nur **ein Bit** anzusteuern.

Sie sehen auch, wie die Parametrierung der Aufträge "Bezugspunkt setzen" und "Längenmessung" realisiert wird.

Voraussetzungen:

- Sie haben die Parametrierung mit Hilfe der Parametriermasken durchgeführt.
- Sie haben im Baustein COMPLETE RESTART (OB 100) die Werte für Um- und Abschalt Differenz angepasst.
- Es liegt kein Prozessfehler an (ERR=0). Anliegende Prozessfehler müssen mit **ERR_A = 1** quittiert werden.
- Für die Betriebsart "Schrittmaßfahren absolut" muss die Achse synchronisiert sein.
- Sie haben für die Funktion "Längenmessung" die Art der Flanke mit dem Parameter "Längenmessung" über die Parametriermasken eingestellt.

Programmaufbau

Das Programm Betriebsarten (MODE_D) hat folgenden Aufbau:

- Reset der Startsignale
- Betriebsartenparameter für Referenzpunktfahren
- Betriebsartenparameter für Tippen
- Betriebsartenparameter für Schrittmaßfahrt relativ
- Betriebsartenparameter für Schrittmaßfahrt absolut
- Auftrag Bezugspunkt setzen
- Aufruf des SFB DIGITAL

Bedienung "Betriebsarten"

Tragen Sie die gewünschten Positionswerte der Schrittmaßfahrten in die Variablentabelle VAT_MODE_D ein. Setzen und aktivieren Sie den Parameter S_DRV_EN.

Parameter	Einstellung	Bedeutung
Pos1a	xx xxx xxx	Sollwert: Positionswert 1 absolut
Pos2a	xx xxx xxx	Sollwert: Positionswert 2 absolut
Pos1r	xx xxx xxx	Sollwert: Positionswert 1 relativ
Pos1a	xx xxx xxx	Sollwert: Positionswert 2 relativ
Pos_REFP	xx xxx xxx	Sollwert: Bezugspunkt
S_DRV_EN	1	Steuersignal: Antriebsfreigabe

Durch Setzen und Aktivieren **eines** der folgenden Steuersignale starten Sie die entsprechende Funktion:

Parameter	Einstellung	Bedeutung
S_REF	1	Start Referenzpunktfahrt
S_DIR_PF	1	Start Tippen vorwärts schnell
S_DIR_MF	1	Start Tippen rückwärts schnell
S_DIR_PS	1	Start Tippen vorwärts langsam
S_DIR_MS	1	Start Tippen rückwärts langsam
S_POS1a	1	Start Pos 1 anfahren (absolut)
S_POS2a	1	Start Pos 2 anfahren (absolut)
S_POS1rP	1	Start Pos 1 anfahren (relativ vorwärts)
S_POS1rM	1	Start Pos 2 anfahren (relativ rückwärts)
S_REFP_EN	1	Auftrag Bezugspunkt setzen

Rückmeldungen:

Parameter	Wert	Bedeutung
ST_ENBLD	1	Startfreigabe
MODE_OUT	x	Aktive/eingestellte Betriebsart
WORKING	1	Fahrt läuft
POS_RCD	1	Position erreicht
SYNC	1	Achse synchronisiert
ACT_POS	xx xxx xxx	Aktueller Lageistwert
ERROR	1	Fehler beim Starten/Fortsetzen einer Fahrt
STATUS	xxxx	Fehlernummer
ERR	xxxx	Prozessfehler

Längenmessung

Die Funktion "Längenmessung" wird direkt mit dem Eingang (DI + 0.3) "Längenmessung" gesteuert.

Rückmeldungen:

Parameter	Wert	Bedeutung
MSR_DONE	1	Längenmessung beendet
BEG_VAL	1	Lageistwert Längenmessung Beginn
END_VAL	1	Lageistwert Längenmessung Ende
LEN_VAL	1	Gemessene Länge

1.3.3 Digital 3 Pufferspeicher

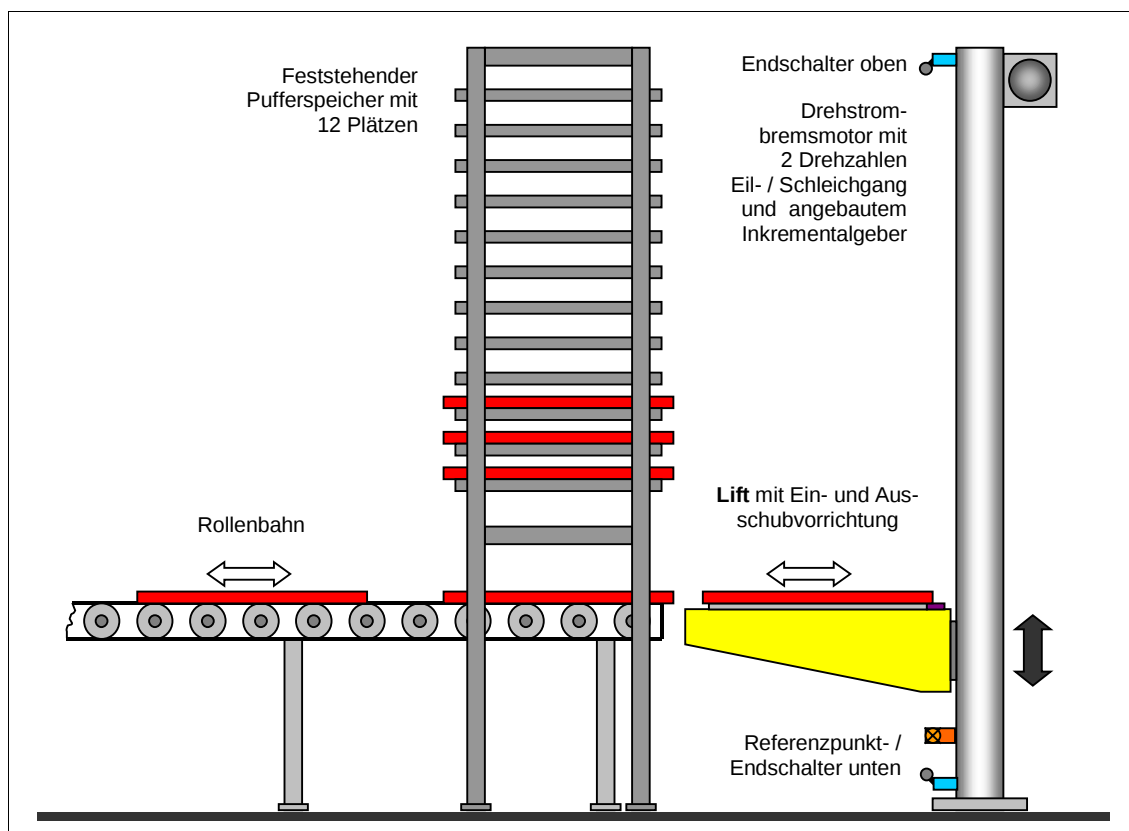
Ziel

Sie sehen mit dem vorliegenden Beispiel eine Anwendung für die Technologie **"Positionieren mit Digitalausgängen"**.

Hier werden die Steuersignale für den Automatikbetrieb durch eine Schrittkette erzeugt. Handfunktionen wie Tippen oder Schrittmaßfahren sind ebenfalls möglich.

Aufgabenstellung

An einer Anlage werden die Werkzeugplatten in einem Pufferspeicher zwischengespeichert.



Der Lift hat die Aufgabe die vorbereiteten Werkzeugplatten von der Rollenbahn abzuholen und im Pufferspeicher abzulegen.

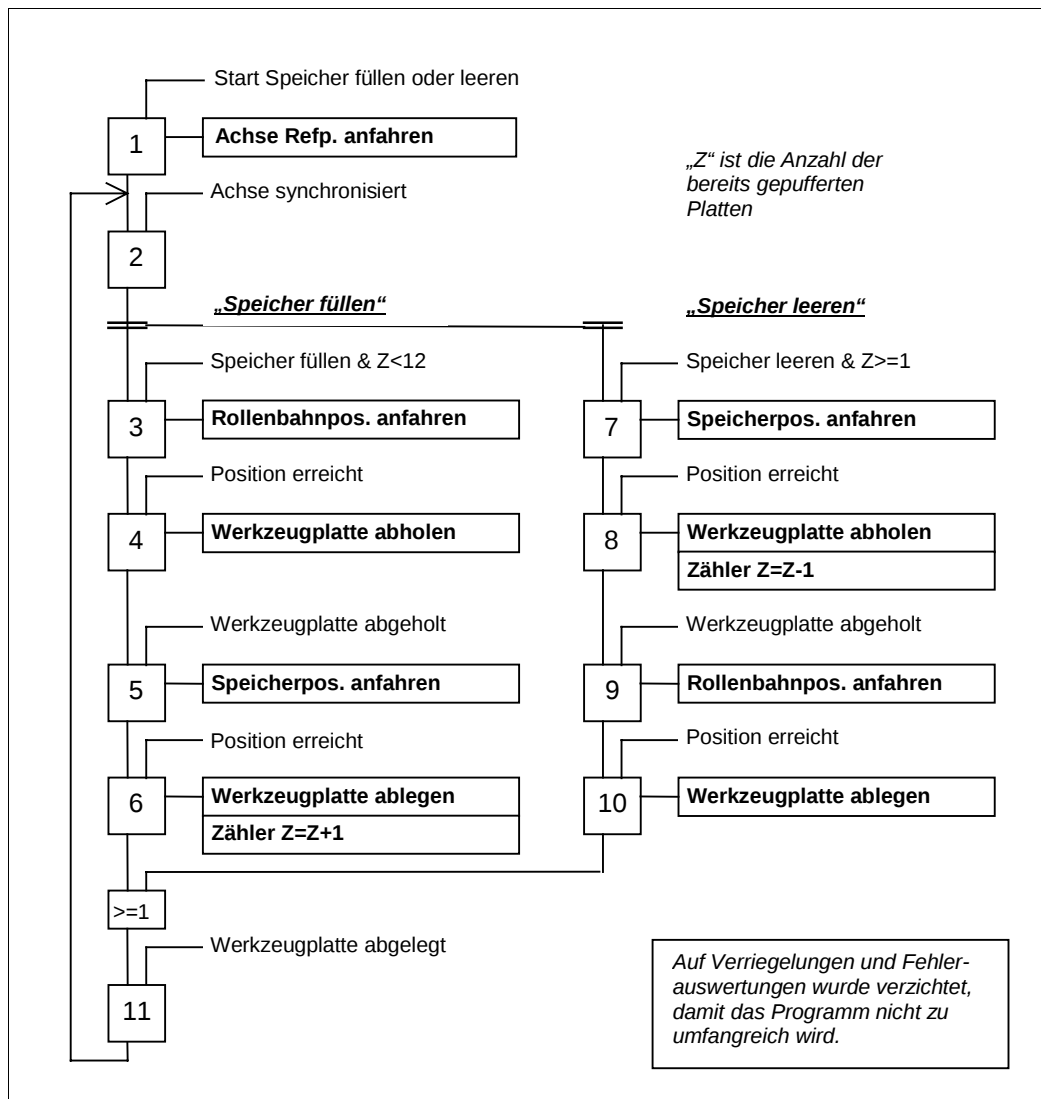
Bei Anforderung der Presse werden die Werkzeugplatten vom Speicher abgeholt und über die Rollenbahn zur Presse transportiert.

Der Lift wird mit einem **Drehstrombremsmotor** mit Eil-/Schleichgang angetrieben und der Istwert über einen Inkrementalgeber erfasst.

Der Referenzpunktschalter ist unten angebracht. Damit wird der Lift synchronisiert.

Ablaufdiagramm

Der Start erfolgt über die Befehle "Speicher Füllen" oder "Speicher Leeren".



Beim ersten Einschalten wird der Lift synchronisiert. Danach wird der Programmzweig "Speicher füllen" oder "Speicher leeren" durchlaufen. Beim "Speicher füllen" wird zunächst die Rollenbahnposition angefahren. Danach wird mit einem Greifersystem die Werkzeugplatte in den Lift gezogen. Nun kann die Speicherposition angefahren und die Werkzeugplatte ausgeschoben werden. Die abgelegten Werkzeugplatten im Speicher werden mit einem Zähler gezählt. "Speicher leeren" wird mit ähnlichem Ablauf durchfahren (siehe Diagramm).

Umrechnung des Weges von mm in Impulse (Weginkremente)

Der Inkrementalgeber liefert 2500 Inkremente pro Umdrehung und ist nach dem Getriebe angebaut. Der Lift bewegt sich bei einer Umdrehung 100 mm.

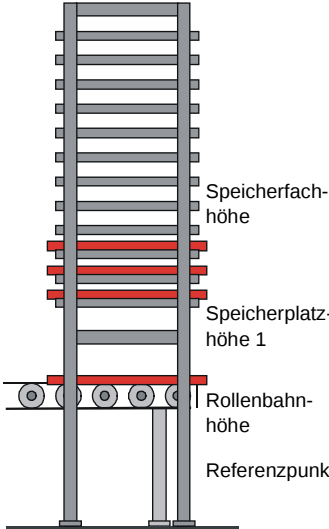
$$100 \text{ mm} / 2500 \text{ Inkremente} = 0,040 \text{ mm}$$

$$1 \text{ Inkrement} = 4 \text{ Impulse}$$

$$0,040 \text{ mm} / 4 = 0,010 \text{ mm}$$

Ein Impuls entspricht somit einer Wegstrecke von 0,010 mm.

Zielpositionen



Es werden folgende Positionssollwerte benötigt:

Zielpositionen	Umrechnung in Impulse (Weginkremente)
Speicherhöhe 12	$\text{Speicherhöhe 1} + (12-1) \cdot \text{Speicherfachhöhe}$ $1100 \text{ mm} + (12-1) \cdot 100 \text{ mm} = 2200 \text{ mm}$ $2200 \text{ mm} / 0,01 \text{ mm/Imp.} = \mathbf{220\ 000}$ Impulse
Speicherhöhe 2..11	siehe Berechnung Speicherhöhe 12
Speicherhöhe 1	$1100 \text{ mm} / 0,01 \text{ mm/Imp.} = \mathbf{110\ 000}$ Impulse
Rollenbahnhöhe	$600 \text{ mm} / 0,01 \text{ mm/Imp.} = \mathbf{60\ 000}$ Impulse
Referenzpunkt	$300 \text{ mm} / 0,01 \text{ mm/Imp.} = \mathbf{30\ 000}$ Impulse
Speicherfachhöhe	$100 \text{ mm} / 0,01 \text{ mm/Imp.} = \mathbf{10\ 000}$ Impulse

Parametrieren mit STEP 7

Mit STEP 7 parametrieren Sie die CPU wie folgt:

Parameter	Erläuterung
Technologieart	Positionieren mit Digitalausgängen
Ansteuerart	1
Zielbereich	100 Impulse (1 mm / 0.01 mm/Impuls = 100 Impulse)
Überwachungszeit	2 000 ms
Achsart	Linearachse
Software-Endschalter Anfang	25 000 Impulse Arbeitsbereichsgrenze Anfang (250 mm / 0,01 mm/Imp. = 25 000 Impulse)

Parameter	Erläuterung
Software-Endschalter Ende	225 000 Impulse Arbeitsbereichsgrenze Ende (2250 mm / 0,01 mm/Impuls = 225 000 Impulse)
Referenzpunktcoordinate	30 000 Impulse
Referenzpunktlage zu Referenzpunktschalter	In Richtung Plus
Inkremente pro Geberumdrehung	2 500
Zählrichtung	Normal (Spurwechsel Spur A vor B = Istwert positiv)

Um-/Abschaltdifferenz

Die **Umschaltdifferenz** definiert den Umschaltpunkt, an dem der Antrieb von Eilgang auf Schleichgang umschaltet. Der Antrieb muss sicher die Schleichgangsgeschwindigkeit erreichen.

Diese Wegstrecke ist im Beispiel mit 25 mm festgelegt.

$25 \text{ mm} / 0,010 \text{ mm/Impuls} = \mathbf{2500}$ Impulse

Die **Abschaltdifferenz** ist die Wegstrecke vom Abschaltpunkt bis zum Ziel. Der Antrieb wird im Abschaltpunkt abgeschaltet.

Damit die Zielposition so genau wie möglich getroffen wird, müssen Sie so vorgehen:

1. Geben Sie über das Anwenderprogramm eine Abschaltdifferenz $\geq 1/2$ Zielbereich vor.
2. Fahren Sie den Lift in der Betriebsart "Schrittmaßfahrt absolut" einmal auf eine Position.
3. Ermitteln Sie die Differenz zwischen tatsächlich erreichter Zielposition und vorgegebener Zielposition.
4. Addieren Sie diese Differenz zur Abschaltdifferenz.

Ermittelt wurde ein Nachlauf von 250 Impulsen für aufwärts und abwärts.

Im Anwenderprogramm tragen Sie ein:

- Umschaltdifferenz + = **2500** Impulse
- Umschaltdifferenz - = **2500** Impulse
- Abschaltdifferenz + = **250** Impulse
- Abschaltdifferenz - = **250** Impulse

Bedienung "Pufferspeicher"

Tragen Sie die gewünschten Positionswerte der Schrittmaßfahrten in die Variablen-tabelle VAT_STORAGE ein. Setzen und aktivieren Sie den Parameter S_DRV_EN.

Parameter	Einstellung	Bedeutung
PosCon	xx xxx xxx	Sollwert: Positionswert Rollenbahnhöhe
PosS1	xx xxx xxx	Sollwert: Positionswert Speicherplatz 1
PosSH	xx xxx xxx	Sollwert: Positionswert Speicherfachhöhe (Relativmaß)
S_DRV_EN	1	Steuersignal: Antriebsfreigabe

Automatikbetrieb können Sie mit einem der folgenden Signale starten:

Parameter	Einstellung	Bedeutung
S_Fill	1	Start Automatikbetrieb Speicher füllen
S_Empty	1	Start Automatikbetrieb Speicher leeren

Die folgenden Steuersignale können Sie setzen und aktivieren, wenn der Automatikbetrieb nicht gestartet ist. (S_Fill & S_Empty = 0 → Handbetrieb)

Parameter	Einstellung	Bedeutung
S_REF	1	Start Referenzpunktfahrt
S_DIR_PF	1	Start Tippen vorwärts schnell
S_DIR_MF	1	Start Tippen rückwärts schnell
S_DIR_PS	1	Start Tippen vorwärts langsam
S_DIR_MS	1	Start Tippen rückwärts langsam
S_PosCon	1	Start Pos. Rollenbahn anfahren (absolut)
S_PosS1	1	Start Pos. Speicherplatz 1 anfahren (absolut)
S_PosS_P	1	Start Pos. Speicherplatz +1 anfahren (relativ)
S_PosS_M	1	Start Pos. Speicherplatz -1 anfahren (relativ)

Rückmeldungen:

Parameter	Wert	Bedeutung
ST_ENBLD	1	Startfreigabe
MODE_OUT	x	Aktive/eingestellte Betriebsart
WORKING	1	Fahrt läuft
POS_RCD	1	Position erreicht
SYNC	1	Achse synchronisiert
ACT_POS	xx xxx xxx	Aktueller Lageistwert
ERROR	1	Fehler beim Starten/Fortsetzen einer Fahrt
STATUS	xxxx	Fehlernummer
ERR	xxxx	Prozessfehler

2 Beispielprogramme für die Technologie "Zählen, Frequenzmessen und Pulsweitenmodulation"

2.1 Einführung

Übersicht

In diesem Kapitel finden Sie Anwendungsbeispiele zur Technologie Zählen.

Die Beispiele befinden sich auf der Ihrer Dokumentation beigefügten CD. Falls Ihnen die CD nicht vorliegt, können Sie die Beispiele auch über das Internet beziehen. Das Projekt besteht aus mehreren kommentierten S7-Programmen verschiedener Komplexität und Zielrichtung.

Die Installation der Beispiele ist in der Liesmich.wri der CD beschrieben. Nach der Installation befinden sich die Beispiele im Katalog

...\STEP7\EXAMPLES\ZDt26_02_TF_____31xC_Cnt

Voraussetzungen

- Sie haben eine S7-Station, bestehend aus einer Stromversorgungsbaugruppe sowie einer CPU 31xC aufgebaut und verdrahtet.
- Auf Ihrem PG ist STEP 7 (>= V5.1 + Service Pack 2) korrekt installiert.
- Das PG ist an die CPU angeschlossen.

Vorbereiten der Beispiele

1. Öffnen Sie das Beispielprojekt **ZDt26_02_TF_____31xC_Cnt** im Ordner **...\STEP7\EXAMPLES** mit dem SIMATIC Manager und kopieren Sie es unter einem geeigneten Namen in Ihr Projektverzeichnis (**Datei > Speichern unter**). Nutzen Sie die Darstellung **Ansicht > Detail** zur Anzeige der kompletten Informationen.
2. Fügen Sie in Ihr Projekt eine Station entsprechend Ihrem Hardware-Aufbau ein.
3. Wählen Sie ein Beispielprogramm aus und kopieren Sie das S7-Programm in diese Station.
4. Konfigurieren Sie die Hardware mit **HW Konfig** und wählen Sie dabei das TF-Submodul **Zählen** aus.

5. Parametrieren Sie dieses Submodul anhand der Beispielparameter.
6. Speichern Sie die HW-Konfiguration und laden Sie diese in die CPU.
7. Laden Sie den Bausteinbehälter in die CPU.

Code der Beispiele

Die Beispiele sind in AWL/FUP geschrieben. Sie können sie direkt über den KOP/AWL/ FUP-Editor anschauen.

Wählen Sie in diesem Editor über **Ansicht > Anzeigen mit** "Symbolischer Darstellung", "Symbolauswahl" und "Kommentar". Wenn Sie genügend Platz auf dem Bildschirm haben, können Sie sich auch noch die "Symbolinformation" anzeigen lassen.

Anwenden eines Beispiels

In den Beispielprogrammen sind Variablentabellen (VAT) vorhanden, mit denen Sie die Werte ansehen und verändern können.

Weiterverwenden eines Beispiels

Sie können den Code der Beispiele direkt als Anwenderprogramm weiterverwenden.

Der Code der Beispiele ist jedoch nicht optimiert und auch nicht für alle Möglichkeiten ausgelegt. Fehlerauswertungen sind in den Beispielprogrammen nicht ausführlich ausprogrammiert, um die Programme nicht zu umfangreich werden zu lassen.

2.2 Zählen

Für die Technologie Zählen finden Sie zwei Beispielprogramme:

- **Zählen 1 Erste Schritte**
In diesem Beispiel wird der Systemfunktionsbaustein (SFB) COUNT aufgerufen und mit Parametern versorgt. Sie lernen vor allem die Parametrierung und die Funktionen des SFB COUNT kennen.
- **Zählen 2 Abfüllstation mit automatischer Behältergrößenerkennung**
Das Beispiel "Zählen 2 Abfüllstation" ist eine typische Anwendung für die Technologie Zählen und besteht aus einer Verknüpfungslogik für den Automatikablauf und den Aufrufen der SFB.

2.2.1 Zählen 1 Erste Schritte

Ziel

Mit diesem Beispiel nehmen Sie die Technologie "Zählen" in Betrieb.

Durch Anlegen der entsprechenden Parameter am SFB können Sie folgende Funktionen testen:

- Starten/Stoppen des Zählers
- Auslesen von Statusbits
- Auslesen des aktuellen Zählerstands
- Schreiben der internen Zählregister
- Freigabe/Steuern des Ausgangs DO
- Latchfunktion

Bausteinstruktur

Das Beispiel "Zählen 1 Erste Schritte" enthält folgende Bausteine:

Baustein	Name (in der Symbolleiste)	Beschreibung
OB 1	CYCLE_EXC	Zyklisches Programm
FB 11	GETST_C	Beispiel 1: Erste Schritte COUNT
DB 11	DI_GETST_C	Instanz-DB zu GETST_C
SFB 47	COUNT	Systemfunktionsbaustein COUNT
DB 16	DI_COUNT	Instanz-DB zu SFB COUNT

Im Baustein GETST_C wird der Systemfunktionsbaustein COUNT aufgerufen und mit Parametern versorgt.

Die Parameter des SFB COUNT können Sie über die vorbereitete Variablentabelle VAT_GETST_C ändern.

Anschlüsse:

Folgende Eingangssignale der CPU 31xC müssen Sie verdrahten:

Anschluss CPU 312C: X1	Name/ Adresse	Funktion im Beispiel
2	DI+0.0	Impulseingang
3	DI+0.1	Richtung
4	DI+0.2	Hardwaretor
8	DI+0.6	Latcheingang
12	2 M	Bezugspotential der Versorgungsspannung
13	1 L+	Versorgungsspannung 24 V DC
16	DO+0.2	Simulation: Impulseingang → mit DI+0.0 verbinden
17	DO+0.3	Simulation: Richtungsbit → mit DI+0.1 verbinden
18	DO+0.4	Simulation: Hardwaretor → mit DI+0.2 verbinden
19	DO+0.5	Simulation: Latcheingang → mit DI+0.6 verbinden
20	1 M	Bezugspotential der Versorgungsspannung

Anschluss CPU 313C-2 DP/PtP: X1; CPU 313C, 314C-2 DP/PtP: X2	Name/ Adresse	Funktion im Beispiel
1	1 L+	Versorgungsspannung 24 V DC
2	DI+0.0	Impulseingang
3	DI+0.1	Richtung
4	DI+0.2	Hardwaretor
16	DI+1.4	Latcheingang
20	1 M	Bezugspotential der Versorgungsspannung
21	2 L+	Versorgungsspannung 24 V DC
24	DO+0.2	Simulation: Impulseingang → mit DI+0.0 verbinden
25	DO+0.3	Simulation: Richtungsbit → mit DI+0.1 verbinden
26	DO+0.4	Simulation: Hardwaretor → mit DI+0.2 verbinden
27	DO+0.5	Simulation: Latcheingang → mit DI+1.4 verbinden
30	2 M	Bezugspotential der Versorgungsspannung

Parameter

Über die Hardwarekonfiguration parametrieren Sie die CPU 31xC wie folgt:

Parameter	Eingabe
Kanal	0
Betriebsart	Endlos Zählen
Torfunktion	Zählvorgang abbrechen
Signalauswertung	Impuls/Richtung
Funktion des Eingangs	Hardwaretor verwenden
Verhalten des Ausgangs DO	Ausgang schaltet bei Zählerstand $\geq$ Vergleichswert

Bedienung "Starten/Stoppen des Zählers"

"Starten des Zählers mit dem Hardwaretor"

- Erzeugen Sie Zählimpulse für den "Zähleingang" (Simulation: T_PULSE)
Über die Belegung der Variablen S_IMP_H in der VAT wählen Sie die Quelle für die Zählimpulse:
 - S_IMP_H = 0: Als Impulsquelle dient ein programmierter SW-Taktgeber, dessen Frequenz über die Variable T_PULSE (Impulsdauer) einstellbar ist. Der Takt wird über einen Digitalausgang auf den Impulseingang geführt.
 - S_IMP_H = 1: Die Zählimpulse werden von Hand durch das Setzen und Rücksetzen der Variablen S_IMP_T in der VAT vorgegeben.
- Setzen und aktivieren Sie folgenden SFB-Parameter:
SW_GATE = 1: Softwaretor zum Starten/Stoppen des Zählers
- Setzen Sie den Eingang "Hardwaretor" (Simulation: S_HWT = 1)

Parameter	Einstellung	Bedeutung
SW_GATE	1	Softwaretor öffnen
T_PULSE	S5T#500 ms	Impulslänge (z. B. 500 ms -> 1Hz)
S_IMP_H	0	Simulation: Impulse 0 = Auto, 1 = Hand
S_IMP_T	0-1	Simulation: Impulse manuell vorgeben
S_HWT	1	Simulation: Hardwaretor

"Starten des Zählers mit dem Softwaretor SW-GATE"

- Erzeugen Sie Zählimpulse für den "Zähleingang" (Simulation: T_PULSE)
- Setzen Sie den Eingang "Hardwaretor" (Simulation: S_HWT = 1)
- Setzen und aktivieren Sie folgenden SFB-Parameter:
SW_GATE = 1: Softwaretor zum Starten/Stoppen des Zählers

Parameter	Einstellung	Bedeutung
T_PULSE	S5T#500 ms	Impulslänge (z. B. 500 ms -> 1Hz)
S_IMP_H	0	Simulation: Impulse 0 = Auto, 1 = Hand
S_IMP_T	0-1	Simulation: Impulse manuell vorgeben
S_HWT	1	Simulation: Hardwaretor
SW_GATE	1	Softwaretor öffnen

"Stoppen des Zählers"

Sie können den Zähler stoppen, indem Sie das Eingangssignal "Hardwaretor" wegnehmen oder den SFB-Parameter "Softwaretor" zurücksetzen (SW_GATE = 0).

Folgende Ausgangsparameter können Sie beobachten:

Parameter	Wert	Bedeutung
STS_GATE	1	Status: Tor (UND-Verknüpfung Hard- und Softwaretor)
STS_STRT	1	Status: Starteingang (Hardwaretor)
STS_C_UP	1	Status: Richtung vorwärts
STS_C_DN	0	Status: Richtung rückwärts
COUNTVAL	xxxx	Aktueller Zählerstand

Bedienung "Laden des Zählers"

Falls Sie den Zähler z. B. auf 20 einstellen wollen, setzen und aktivieren Sie folgende SFB-Parameter:

Parameter	Einstellung	Bedeutung
JOB_ID	1	Auftragsnummer "Zähler direkt schreiben"
JOB_VAL	20	Zählwert eintragen, der übertragen werden soll
JOB_REQ	1	Auftragsanstoß setzen (steigende Flanke wird ausgewertet)

Folgende Ausgangsparameter können Sie beobachten:

Parameter	Wert	Bedeutung
COUNTVAL	20	Aktueller Zählerstand
JOB_DONE	1	Status: Neuer Auftrag kann gestartet werden
JOB_ERR	0	Status: Auftrag fehlerhaft
JOB_STAT	0	Status: Auftrags-Fehlernummer

Bedienung "Laden des Vergleichswertes"

Wenn Sie den Vergleichswert z. B. auf 100 einstellen wollen, setzen und aktivieren Sie folgende SFB-Parameter:

Parameter	Einstellung	Bedeutung
JOB_ID	4	Auftragsnummer "Vergleichswert schreiben"
JOB_VAL	100	Vergleichswert eintragen, der übertragen werden soll
JOB_REQ	1	Auftragsanstoß setzen (steigende Flanke wird ausgewertet)

Folgende Ausgangsparameter können Sie beobachten:

Parameter	Wert	Bedeutung
JOB_DONE	1	Status: Neuer Auftrag kann gestartet werden
JOB_ERR	0	Status: Auftrag fehlerhaft
JOB_STAT	0	Status: Auftrags-Fehlernummer

Bedienung "Freigabe und Steuern des Ausgangs DO"

Um den Ausgang einschalten zu können, muss der SFB-Parameter CTRL_DO gesetzt werden.

Parameter	Einstellung	Bedeutung
CTRL_DO	1	Freigabe Ausgang

Es gibt zwei Möglichkeiten:

1. Steuern des Ausgangs DO durch den Vergleich

Wenn Sie den Vergleichswert z. B. auf 100 eingestellt haben, und den Zähler nochmals starten, können Sie den Status des Ausgangs DO beobachten.

Parameter	Wert	Bedeutung
COUNTVAL	xxxx	Aktueller Zählerstand
STS_DO	x	Status: Ausgang Da das Verhalten des Ausgangs "Zählerstand >= Vergleichswert" eingestellt ist, schaltet der Ausgang DO, wenn der aktuelle Zählerstand >= 100 ist.

2. Steuern des Ausgangs DO gleichzeitig zu dem Vergleich

Mit dem SFB-Parameter **SET_DO** haben Sie die Möglichkeit die Wirkung der Vergleichsfunktionen über Ihr Steuerungsprogramm zu simulieren:

- Mit der **positiven Flanke** von **SET_DO** wird der Ausgang gesetzt.
- Eine **negative Flanke** von **SET_DO** setzt den Ausgang wieder zurück.

Beachten Sie, dass die Vergleiche weiterhin aktiv sind und den Ausgang mit jedem Zähltakt setzen bzw. rücksetzen können.

Bedienung "LATCH-Funktion"

Über eine positive Flanke am LATCH-Eingang können Sie den augenblicklichen Zählerstand abspeichern.

Führen Sie dazu folgende Schritte durch:

1. Starten Sie den Zähler indem Sie das Hard- und Softwaretor schließen und erzeugen Sie Zählimpulse für den "Zähleingang" (Simulation: PULSE_SEL).

Parameter	Einstellung	Bedeutung
SW_GATE	1	Softwaretor schließen
HWT	1	Simulation: Hardwaretor
T_PULSE	S5T#500 ms	Impulslänge (z. B. 500 ms -> 1 Hz)

2. Erzeugen Sie eine Flanke am Latcheingang (Simulation: LATCH).

Parameter	Einstellung	Bedeutung
S_LATCH	1	Simulation: Latcheingang

Folgende Ausgangsparameter können Sie beobachten:

Parameter	Wert	Bedeutung
STS_LATCH	1	Status Latcheingang
LATCHVAL	xxxx	gelesener Latchwert
COUNTVAL	xxxx	Aktueller Zählerstand

2.2.2 Zählen 2 Abfüllstation

Ziel:

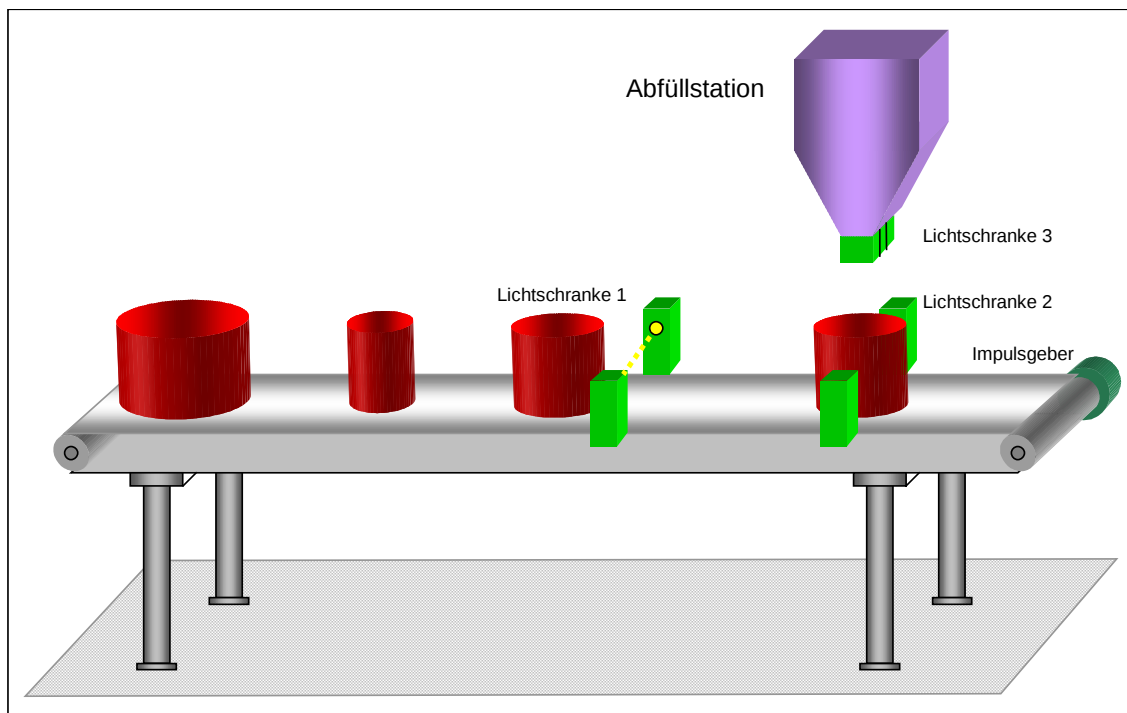
Mit dem vorliegenden Beispiel soll eine Anwendung für die Technologie **Zählen** gezeigt werden.

Aufgabenstellung:

Bei dem Beispiel handelt es sich um eine Abfüllstation mit automatischer Behältergrößenerkennung und Positionierung. In dieser Anlage sollen Behälter unterschiedlicher Größe befüllt werden.

Auf einem Förderband werden Behälter unterschiedlicher Größe transportiert. Die Behälter werden durch die Lichtschranke 1 erkannt. Der Behälterdurchmesser wird durch das Zählen der Impulse ermittelt.

Durch den ermittelten Behälterdurchmesser lässt sich das Volumen berechnen und damit die Abfüllmenge der Abfüllstation einstellen.



Ablauf

- Das Förderband wird gestartet, wenn kein Behälter in der Abfüllstation steht oder wenn das Abfüllen beendet ist.
- **Zähler 0** ermittelt den Behälterdurchmesser; **Lichtschranke 1** steuert das Hardwaretor des Zählers.
Verlässt der Behälter die **Lichtschranke 1**, so wird der Zählstand gespeichert. Je nach Durchmesser des Behälters wird der Sollwert für die Abfüllstation berechnet und der Abfüllstation (Zähler 1) mit Auftrag 4 (Vergleichswert laden) vorgegeben.
- Ist der Behälter durch die **Lichtschranke 2** positioniert, wird das Abfüllen gestartet. Mit **Zähler 1** werden die abgefüllten Teile gezählt und das Dosieren beendet.

Umrechnung der Abfüllmenge

Das Volumen errechnet sich aus $d^2 * \pi/4 * h$. Die Höhe des Behälters wird mit 100 mm angenommen. Pro 10 000 mm³ wird ein Teil abgefüllt.

Beispiel: Behälterdurchmesser 150 mm.

$$\begin{aligned}\text{Abfüllmenge} &= d^2 * \pi/4 * h/10\,000 \\ &= 150\text{mm} * 150\text{mm} * (\pi/4) * 100\text{ mm}/10\,000\text{ Teile/mm}^3 \\ &= \underline{176\text{ Teile}}\end{aligned}$$

Bausteinstruktur

Das Beispiel 2 "Abfüllstation" enthält folgende Bausteine:

Baustein	Name (in der Symbolleiste)	Beschreibung
OB1	CYCLE_EXC	Zyklisches Programm
FB12	FILL	Beispiel 2: Abfüllstation
DB12	DI_FILL	Instanz-DB zu FILL
SFB47	COUNT	Systemfunktionsbaustein COUNT
DB16	DI_COUNT	Instanz-DB zu SFB COUNT (Behälterdurchm.)
DB17	DI_COUNT2	Instanz-DB zu SFB COUNT (Abfüllstation)

Anschlüsse

Nachfolgend sind die Anschlüsse des Beispiels aufgelistet:

Anschluss CPU 312C: X1	Name/ Adresse	Funktion im Beispiel
Anschlüsse Zähler 0 "Behälterdurchmesser"		
2	DI+0.0	<u>Zählimpulse</u> Impulsgeber Bandtransport
4	DI+0.2	<u>Hardwaretor</u> Lichtschanke 1: Behälterdurchmesser ermitteln
Anschlüsse Zähler 1 "Abfüllen"		
5	DI+0.3	<u>Zählimpulse</u> Lichtschanke 3: abgefüllte Zeile zählen
15	DO+0.1	<u>Ausgang DO</u> Abfüllventil
Allgemein		
16	DO+0.2	Simulation: Impulse Bandtransport und Abfüllen → DO+0.2 mit DI+0.0 und DI+0.3 verbinden
18	DO+0.4	Simulation: HWT0 Behälterdurchmesser → DO+0.4 mit DI+0.2 verbinden
12	2 M	Bezugspotential der Versorgungsspannung
13	1 L+	Versorgungsspannung 24 V DC
20	1 M	Bezugspotential der Versorgungsspannung

Anschluss CPU 313C-2 DP/PtP: X1; CPU 313C, 314C-2 DP/PtP: X2	Name/ Adresse	Funktion im Beispiel
Anschlüsse Zähler 0 "Behälterdurchmesser"		
2	DI+0.0	<u>Zählimpulse</u> Impulsgeber Bandtransport
4	DI+0.2	<u>Hardwaretor</u> Lichtschanke 1: Behälterdurchmesser ermitteln
Anschlüsse Zähler 1 "Abfüllen"		
5	DI+0.3	<u>Zählimpulse</u> Lichtschanke 3: abgefüllte Zeile zählen
23	DO+0.1	<u>Ausgang DO</u> Abfüllventil
Allgemein		
24	DO+0.2	Simulation: Impulse Bandtransport und Abfüllen → DO+0.2 mit DI+0.0 und DI+0.3 verbinden
26	DO+0.4	Simulation: HWT0 Behälterdurchmesser → DO+0.4 mit DI+0.2 verbinden
1	1 L+	Versorgungsspannung 24 V DC

Anschluss CPU 313C-2 DP/PtP: X1; CPU 313C, 314C-2 DP/PtP: X2	Name/ Adresse	Funktion im Beispiel
20	1 M	Bezugspotential der Versorgungsspannung
21	2 L+	Versorgungsspannung 24 V DC
30	2 M	Bezugspotential der Versorgungsspannung

Parametrierung

Es werden zwei schnelle Zähler benutzt, die Sie über die Hardwarekonfiguration parametrieren.

1. Parametrierung Zähler 0 (Behälterdurchmesser):

Parameter	Eingabe
Kanal	0
Betriebsart	Endlos Zählen
Torfunktion	Zählvorgang abbrechen
Signalauswertung	Impuls/Richtung
Funktion des Eingangs	Hardwaretor verwenden
Verhalten des Ausgangs DO	Kein Vergleich

2. Parametrierung Zähler 1 (Abfüllen):

Parameter	Eingabe
Kanal	1
Betriebsart	Endlos Zählen
Torfunktion	Zählvorgang abbrechen
Signalauswertung	Impuls/Richtung
Funktion des Eingangs	Hardwaretor nicht verwendet
Verhalten des Ausgangs DO	Ausgang schaltet bei Zählerstand <= Vergleichswert

Bedienung "Abfüllstation"

Folgende Variablentabelle wurde für den Test der Abfüllstation vorbereitet:

Symbol	Symbolkommentar
S_START	Steuersignal: Automatik Start
S_LS1	Steuersignal: Lichtschranke 1 (Behälterdurchmesser)
S_LS2	Steuersignal: Lichtschranke 2 (Behälter in Abfüllposition)
T_PULSE	Simulation: Impulse
DIAM	Behälterdurchmesser (diameter)
FILL_SETP	Abfüll-Sollwert (fill set point)
FILL_VAL	Abfüll-Istwert (fill actual value)
CONV_ON	Band EIN
C_POS_RCD	Band Position erreicht
FILL_STRT	Abfüllen start
FILL_WORK	Abfüllen läuft
FILL_DONE	Abfüllen beendet

1. Erzeugen Sie Impulse für den Zähl Eingang (z. B. $T_PULSE = S5T\#50\text{ ms} \rightarrow 10\text{ Hz}$).
2. Starten Sie den Ablauf durch Setzen des Parameters **S_START = 1**. Wenn kein Behälter in der Abfüllstation steht oder das Abfüllen beendet ist, wird das Förderband gestartet.
3. Nun können Sie den Eingang **S_LS1 = 1** setzen (Lichtschranke 1: Behälterdurchmesser). Dabei können Sie erkennen, dass der Zähler 0 die Breite des Behälters ermittelt.
4. Mit dem Rücksetzen von **S_LS1 = 0** wird durch die negative Flanke die Abfüllmenge berechnet und als Vergleichswert dem Zähler 1 vorgegeben.
5. Wenn Sie den Eingang **S_LS2 = 1** setzen (Lichtschranke 2: Behälter in Position), wird das Förderband gestoppt und das Abfüllen gestartet.
 Wurde die berechnete Menge abgefüllt, so schaltet das Abfüllventil wieder aus und meldet Abfüllen beendet. Dadurch schaltet sich das Förderband wieder ein.
6. Nun können Sie den Eingang **S_LS2 = 0** wieder rücksetzen und die Schritte 2 bis 5 wiederholen.

2.3 Frequenzmessen

Für die Technologie "Frequenzmessen" finden Sie zwei Beispielprogramme:

1. **Freq 1 Erste Schritte:**

In diesem Beispiel wird der Systemfunktionsbaustein (SFB) FREQUENC aufgerufen und mit Parametern versorgt. Sie lernen vor allem die Parametrierung und die Funktionen des SFB FREQUENC kennen.

2. **Freq 2 Drehzahl:**

Das Beispiel 2 "Drehzahlüberwachung ..." ist eine typische Anwendung für die Technologie "Frequenz messen" und besteht aus einer Verknüpfungslogik, den Grenzwertberechnungen und dem Aufruf der SFBs.

2.3.1 Frequenz 1 Erste Schritte

Ziel

Mit diesem Beispiel nehmen Sie die Technologie "Frequenzmessen" in Betrieb.

Sie haben die Möglichkeit, folgende Funktionen durch Anlegen der entsprechenden Parameter am SFB zu testen:

- Starten/Stoppen der Frequenzmessung
- Auslesen von Statusbits
- Auslesen der aktuellen Frequenz und des Zählerstandes
- Schreiben der internen Register

Baueinstruktur:

Das Beispiel "Freq 1 Erste Schritte" enthält folgende Bausteine:

Baustein	Name (in der Symbolleiste)	Beschreibung
OB 1	CYCLE_EXC	Zyklisches Programm
FB 21	GETST_F	Beispiel 1: Erste Schritte FREQUENCY
DB 21	DI_GETST_F	Instanz-DB zu GETST_F
SFB 48	FREQUENC	Systemfunktionsbaustein FREQUENC
DB 26	DI_FREQ	Instanz-DB zu SFB FREQUENC

Im Baustein GETST_F wird der Systemfunktionsbaustein FREQUENC aufgerufen und mit Parametern versorgt.

Die Parameter des SFB FREQUENC können Sie über die vorbereitete Variablen-tabelle VAT_GETST_F ändern.

Anschlüsse:

Folgende Anschlüsse der CPU 31xC müssen Sie verdrahten:

Anschluss CPU 312C: X1	Name/ Adresse	Funktion im Beispiel
2	DI+0.0	Impulseingang: Die steigenden Flanken des Mess-Signals werden ausgewertet.
3	DI+0.1	Richtung
12	2 M	Bezugspotential der Versorgungsspannung
13	1 L+	Versorgungsspannung 24 V DC
14	DO+0.0	Ausgang DO: Dieser Ausgang schaltet, wenn die Frequenz außerhalb den Grenzen liegt.
16	DO+0.2	Simulation: Impulsausgang → mit DI+0.0 verbinden Das Simulationsprogramm erzeugt Impulse, und gibt sie über diesen Ausgang aus.
17	DO+0.3	Simulation: Richtungsbit → mit DI+0.1 verbinden
20	1 M	Bezugspotential der Versorgungsspannung

Anschluss CPU 313C-2 DP/PtP: X1; CPU 313C, 314C-2 DP/PtP: X2	Name/ Adresse	Funktion im Beispiel
1	1 L+	Versorgungsspannung 24 V DC
2	DI+0.0	Impulseingang: Die steigenden Flanken des Mess-Signals werden ausgewertet.
3	DI+0.1	Richtung
20	1 M	Bezugspotential der Versorgungsspannung
21	2 L+	Versorgungsspannung 24 V DC
22	DO+0.0	Ausgang DO: Dieser Ausgang schaltet, wenn die Frequenz außerhalb den Grenzen liegt.
24	DO+0.2	Simulation: Impulsausgang → mit DI+0.0 verbinden Das Simulationsprogramm erzeugt Impulse, und gibt sie über diesen Ausgang aus.
25	DO+0.3	Simulation: Richtungsbit → mit DI+0.1 verbinden
30	2 M	Bezugspotential der Versorgungsspannung

Parameter

Über die Hardwarekonfiguration parametrieren Sie die CPU 31xC wie folgt:

Parameter	Eingabe
Kanal	0
Betriebsart	Frequenzmessen
Integrationszeit	5000 ms
Untergrenze	0 mHz
Obergrenze	60 000 000 mHz (CPU 314) 30 000 000 mHz (CPU 313) 10 000 000 mHz (CPU 312)
Messwert ausgeben	Direkt
Signalauswertung	Impuls/Richtung
Funktion des Eingangs	Kein Hardwaretor verwendet
Verhalten des Ausgangs	Außerhalb der Grenzen

Bedienung "Starten/Stoppen der Frequenzmessung"

"Starten der Frequenzmessung mit dem Softwaretor SW-GATE"

1. Erzeugen Sie Zählimpulse für den "Impulseingang" (DI+0.0)

Über die Belegung der Variablen S_IMP_H in der VAT wählen Sie die Quelle für die Zählimpulse:

- S_IMP_H = 0: Als Impulsquelle dient ein programmierter SW-Taktgeber, dessen Frequenz über die Variable T_PULSE (Impulsdauer) einstellbar ist. Der Takt wird über einen Digitalausgang auf den Impulseingang geführt.
- S_IMP_H = 1: Die Zählimpulse werden von Hand durch das Setzen und Rücksetzen der Variablen S_IMP_T in der VAT vorgegeben.

2. Setzen und aktivieren Sie folgenden SFB-Parameter:

Parameter	Einstellung	Bedeutung
SW_GATE	1	Softwaretor zum Starten der Frequenzmessung
T_PULSE	S5T#250 ms	Impulslänge (z. B. 250 ms -> 2 Hz)
S_IMP_H	0	Simulation: Impulse 0 = Auto, 1 = Hand
S_IMP_T	0-1	Simulation: Impulse manuell vorgeben

"Stoppen der Frequenzmessung"

Die Frequenzmessung kann gestoppt werden, indem Sie den SFB-Parameter "Softwaretor schließen" (SW_GATE = 0) zurücksetzen.

Folgende Ausgangsparameter können Sie beobachten:

Parameter	Wert	Bedeutung
STS_GATE	1	Status: Softwaretor
STS_C_UP	1	Status: Richtung vorwärts
STS_C_DN	0	Status: Richtung rückwärts
MEAS_VAL	xxxx	Aktuelle Frequenz
COUNTVAL	xxxx	Aktueller Zählerstand
STS_UFLW	x	Status Unterlauf
STS_OFLW	x	Status Überlauf

Bedienung "Laden der Untergrenze"

Falls Sie die Untergrenze z. B. auf 9000 mHz einstellen wollen, setzen und aktivieren Sie folgende SFB-Parameter:

Parameter	Einstellung	Bedeutung
JOB_ID	1	Auftragsnummer "Untergrenze schreiben"
JOB_VAL	9000	Wert für Untergrenze eintragen [mHz]
JOB_REQ	1	Auftragsanstoß setzen (steigende Flanke wird ausgewertet)

Folgende Ausgangsparameter können Sie beobachten:

Parameter	Wert	Bedeutung
JOB_DONE	1	Status: Neuer Auftrag kann gestartet werden
JOB_ERR	0	Status: Auftrag fehlerhaft
JOB_STAT	0	Status: Auftrags-Fehlernummer

Bedienung "Laden der Obergrenze"

Falls Sie die Obergrenze z. B. auf 11000 mHz einstellen wollen, setzen und aktivieren Sie folgende SFB-Parameter:

Parameter	Einstellung	Bedeutung
JOB_ID	2	Auftragsnummer "Obergrenze schreiben"
JOB_VAL	11000	Wert für Obergrenze eintragen [mHz]
JOB_REQ	1	Auftragsanstoß setzen (steigende Flanke wird ausgewertet)

Folgende Ausgangsparameter können Sie beobachten:

Parameter	Wert	Bedeutung
JOB_DONE	1	Status: Neuer Auftrag kann gestartet werden
JOB_ERR	0	Status: Auftrag fehlerhaft
JOB_STAT	0	Status: Auftrags-Fehlernummer

Bedienung "Laden der Integrationszeit"

Falls Sie die Integrationszeit z. B. auf 100 ms einstellen wollen, setzen und aktivieren Sie folgende SFB-Parameter:

Parameter	Einstellung	Bedeutung
JOB_ID	4	Auftragsnummer "Integrationszeit schreiben"
JOB_VAL	100	Wert für Integrationszeit eintragen [ms]
JOB_REQ	1	Auftragsanstoß setzen (steigende Flanke wird ausgewertet)

Folgende Ausgangsparameter können Sie beobachten:

Parameter	Wert	Bedeutung
JOB_DONE	1	Status: Neuer Auftrag kann gestartet werden
JOB_ERR	0	Status: Auftrag fehlerhaft
JOB_STAT	0	Status: Auftrags-Fehlernummer

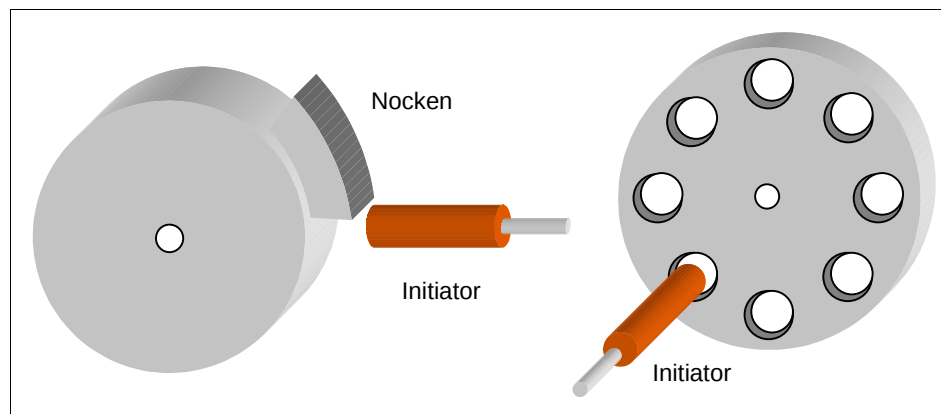
2.3.2 Frequenz 2 Drehzahlüberwachung mit variabler Sollwertvorgabe

Ziel:

Mit dem vorliegenden Beispiel soll eine Anwendung für die Technologie **Frequenzmessen** gezeigt werden.

Aufgabenstellung:

In einer Anlage soll die Drehzahl einer Welle überwacht werden. Die Drehzahl wird mit einem Initiator gemessen und auf Einhaltung eines variablen Drehzahlbereiches überwacht. Beim Verlassen des zulässigen Drehzahlbereiches wird eine Meldung ausgegeben.



Die Impulse können z. B. über eine Lochscheibe oder einen Schaltnocken erzeugt werden.

Ablauf

Mit dem **Start**-Signal wird die Frequenzmessung gestartet. Die Drehzahl (**SPEED**), die Toleranz (**TOLERANCE**) und die Anzahl der Impulse/Umdrehung (**PULSE**) können vorgegeben werden.

Durch die eingehenden Impulse wird die Istdrehzahl (**SPEED_VAL**) gebildet und mit den errechneten Grenzwerten verglichen. Wird die Untergrenze unterschritten wird das Bit **STATUS_U** gesetzt, bei Überschreitung der Obergrenze das Bit **STATUS_O**.

Berechnung der Unter- und Obergrenze

Die Unter-/Obergrenze kann nach folgender Formel errechnet werden.

$$\text{Untergrenze} = \frac{\text{speed}}{60} \times \frac{(100 - \text{tolerance})}{100} \times \text{pulse} \times 1000 \text{ mHz}$$

$$\text{Obergrenze} = \frac{\text{speed}}{60} \times \frac{(100 + \text{tolerance})}{100} \times \text{pulse} \times 1000 \text{ mHz}$$

Beispiel: Drehzahlsollwert = 600 1/min
 Toleranz = 5 %
 Impulse/Umdrehung = 8 (z. B. Lochscheibe mit 8 Löchern)

Vergleichswert	Drehzahl	Frequenz	Vergleichswert Unter-/Obergrenze
Untergrenze	600 1/min - 5% = 570 1/min	570 1/min / 60 = 9,5 Hz	9,5 Hz x 8 (Impulse/Umdrehung) = 76 Hz Untergrenze (Angabe in mHz): 76000 mHz
Obergrenze	600 1/min + 5% = 630 1/min	630 1/min / 60 = 10,5 Hz	10,5 Hz x 8 (Impulse/Umdrehung) = 84 Hz Obergrenze (Angabe in mHz): 84000 mHz

Bausteinstruktur:

Das Beispiel "Freq 2 Drehzahl" enthält folgende Bausteine:

Baustein	Name (in der Symbolleiste)	Beschreibung
OB 1	CYCLE_EXC	Zyklisches Programm
OB 100	COMPLETE RESTART	Neustart
FB 22	SPEED	Beispiel 2: Drehzahlüberwachung
DB 22	DI_SPEED	Instanz-DB zu SPEED
SFB 48	FREQUENC	Systemfunktionsbaustein FREQUENC
DB 26	DI_FREQ	Instanz-DB zu SFB FREQUENC

Anschlüsse

Nachfolgend sind die Anschlüsse des Beispiels aufgelistet:

Anschluss CPU 312C: X1	Name/ Adresse	Funktion im Beispiel
2	DI+0.0	Impulseingang Die steigenden Flanken des Messsignals werden ausgewertet.
12	2 M	Bezugspotential der Versorgungsspannung
13	1 L+	Versorgungsspannung 24 V DC

Anschluss CPU 312C: X1	Name/ Adresse	Funktion im Beispiel
14	DO+0.0	Ausgang DO Dieser Ausgang schaltet, wenn die Drehzahl außerhalb der Grenzen liegt.
16	DO+0.2	Simulation: Impulsausgang → mit DI+0.0 verbinden Das Simulationsprogramm erzeugt Impulse, und gibt sie über diesen Ausgang aus.
20	1 M	Bezugspotential der Versorgungsspannung

Anschluss CPU 313C-2 DP/PtP: X1; CPU 313C, 314C-2 DP/PtP: X2	Name/ Adresse	Funktion im Beispiel
1	1 L+	Versorgungsspannung 24 V DC
2	DI+0.0	Impulseingang Die steigenden Flanken des Messsignals messen.
20	1 M	Bezugspotential der Versorgungsspannung
21	2 L+	Versorgungsspannung 24 V DC
22	DO+0.0	Ausgang DO Dieser Ausgang schaltet, wenn die Drehzahl außerhalb der Grenzen liegt.
24	DO+0.2	Simulation: Impulsausgang → mit DI+0.0 verbinden Das Simulationsprogramm erzeugt Impulse, und gibt sie über diesen Ausgang aus.
30	2 M	Bezugspotential der Versorgungsspannung

Parametrieren

Über die Hardwarekonfiguration parametrieren Sie die CPU 31xC wie folgt:

Parameter	Eingabe
Kanal	0
Betriebsart	Frequenzmessen
Integrationszeit	1000 ms
Untergrenze	0 mHz
Obergrenze	60 000 000 mHz (CPU 314) 30 000 000 mHz (CPU 313) 10 000 000 mHz (CPU 312)
Messwert ausgeben	Direkt
Signalauswertung	Impuls/Richtung
Funktion des Eingangs	Kein Hardwaretor verwendet
Verhalten des Ausganges	Außerhalb der Grenzen

Bedienung Drehzahlüberwachung

Folgende Variablentabelle wurde vorbereitet:

Symbol	Symbolkommentar
T_PULSE	Simulation: Impulszeit für Frequenzerzeugung
START	Steuersignal: Frequenzmessung EIN
RES_STS	Steuersignal: Reset Status
SPEED	Sollwert: Drehzahlvorgabe in U/min.
TOLERANCE	Sollwert: Toleranz in %
PULSE	Sollwert: Impulse/Umdrehung
SPEED_VAL	Istwert: Aktuelle Drehzahl
STATUS_U	Status: Untergrenze unterschritten
STATUS_O	Status: Obergrenze überschritten
MEAS_VAL	SFB-Parameter: Aktueller Messwert
COUNTVAL	SFB-Parameter: Gelesener Zählerstand
UFLW_VAL	berechnete Untergrenze aktuell
OFLW_VAL	berechnete Obergrenze aktuell
UFLW_LOAD	berechnete Untergrenze geladener Wert
OFLW_LOAD	berechnete Obergrenze geladener Wert

Geben Sie folgende Werte vor:

Parameter	Einstellung	Bedeutung
SPEED	300	Sollwert: Drehzahlvorgabe in U/min.
TOLERANCE	5	Sollwert: Toleranz in %
PULSE	1	Sollwert: Impulse/Umdrehung

Das Programm errechnet und überträgt folgende Grenzwerte:

- Untergrenze UFLW_VAL = 4750 mHz
- Obergrenze OFLW_VAL = 5250 mHz

Legen Sie nun Impulse von 5 Hz (T_Pulse = 100 ms) am Messeingang an.

Alternativ können Sie diese Impulse auch über das Simulationsprogramm erzeugen und den Ausgang DO+0.2 mit dem Messeingang DI+0.0 verbinden. Um die Drehzahlüberwachung einzuschalten, müssen Sie den Parameter **START = 1** setzen und aktivieren. Sie können die aktuellen Istwerte, die berechneten Grenzwerte und die Status-Bits beobachten.

Ändern Sie nun den Drehzahlsollwert oder verändern Sie die Frequenz am Messeingang.

Die Status-Bits **STATUS_U** (Untergrenze unterschritten) oder **STATUS_O** (Obergrenze überschritten) melden, sobald die Grenzwerte nicht eingehalten sind. Auch der Ausgang DO meldet, wenn die Drehzahl außerhalb der Grenzen liegt. Die Status-Bits können mit **RES_STS = 1** zurückgesetzt werden.

2.4 Pulsweitenmodulation

Für die Technologie "Pulsweitenmodulation" finden Sie zwei Beispielprogramme:

1. PWM 1 Erste Schritte:
In diesem Beispiel wird der Systemfunktionsbaustein (SFB) PULSE aufgerufen und mit Parametern versorgt. Sie lernen vor allem die Parametrierung des SFBs PULSE kennen.
2. PWM 2 Aufheizen:
Das Beispiel 2 "Aufheizen einer Flüssigkeit" ist eine Anwendung für die Technologie "PWM" und besteht aus den Sollwertberechnungen und dem Aufruf des SFBs.

2.4.1 PWM 1 Erste Schritte

Ziel

Mit diesem Beispiel nehmen Sie die Technologie "PMW" in Betrieb.

Sie lernen vor allem die Parametrierung des Systemfunktionsbausteins (SFB) PULSE kennen.

Sie haben die Möglichkeit, folgende Funktionen durch Anlegen der entsprechenden Parameter am SFB zu testen:

- Starten/Stoppen der Pulsweitenmodulation
- Freigabe/Steuern des Ausgang DO
- Auslesen von Statusbits
- Eingeben des Ausgabewertes
- Schreiben der internen Register

Bausteinstruktur:

Das Beispiel "PWM 1 Erste Schritte" enthält folgende Bausteine:

Baustein	Name (in der Symbolleiste)	Beschreibung
OB 1	CYCLE_EXC	Organisationsbaustein
FC 31	GETST_P	Beispiel 1: Erste Schritte PMW
SFB 49	PULSE	Systemfunktionsbaustein PULSE
DB 36	DI_PULSE	Instanz-DB zu SFB PULSE

Im Baustein GETST_P wird der Systemfunktionsbaustein PULSE aufgerufen und mit Parametern versorgt. Die Parameter des SFBs PULSE können Sie über die vorbereitete Variablentabelle VAT_GETST_P ändern.

Anschlüsse

Nachfolgend sind die Anschlüsse des Beispiels aufgelistet:

Anschluss CPU 312C: X1	Eingang/ Ausgang	Funktion im Beispiel
12	2 M	Bezugspotential der Versorgungsspannung
13	1 L+	Versorgungsspannung 24 V DC
14	DO+0.0	<u>Ausgang DO</u> 24 V-Steuersignal (z. B. für Schütz oder TRIAC)
20	1 M	Bezugspotential der Versorgungsspannung

Anschluss CPU 313C-2 DP/PtP: X1; CPU 313C, 314C-2 DP/PtP: X2	Eingang/ Ausgang	Funktion im Beispiel
1	1 L+	Versorgungsspannung 24 V DC
20	1 M	Bezugspotential der Versorgungsspannung
21	2 L+	Versorgungsspannung 24 V DC
22	DO+0.0	<u>Ausgang DO</u> 24 V-Steuersignal (z. B. für Schütz oder TRIAC)
30	2 M	Bezugspotential der Versorgungsspannung

Parameter:

Über die Hardwarekonfiguration parametrieren Sie die CPU 31xC wie folgt:

Parameter	Eingabe
Kanal	0
Betriebsart	Pulsweitenmodulation
Ausgabeformat	Promille
Zeitbasis	Auflösung 1 ms dadurch können die Zeitwerte im Bereich von 1 ms bis 65,535 s eingestellt werden
Einschaltverzögerung	0 ms Zeit, die ab dem Start der Ausgabesequenz bis zur Ausgabe des Impulsfolge abläuft.
Periodendauer	20 000 ms Die Periodendauer sollte immer ein Vielfaches der Ansprechzeit Ihres am Digitalausgang DO angeschlossenen Stellglieds sein.

Parameter	Eingabe
Mindestimpulsdauer	500 ms Hier muss die Ansprechzeit des am Digitalausgang DO angeschlossenen Stellglieds eingetragen werden.
Funktion des Eingangs	Kein Hardwaretor verwenden

Bedienung "Starten/Stoppen der Pulsweitenmodulation"

"Starten der Ausgabesequenz"

Setzen und aktivieren Sie folgenden SFB-Parameter:

Parameter	Einstellung	Bedeutung
OUTP_VAL	250	Aktueller Ausgabewert (z. B. 250 %)
MAN_DO	0	Freigabe Ausgang (für Manual-Mode)
SW_EN	1	Starten der Ausgabesequenz

"Stoppen der Ausgabesequenz"

Sie können die Ausgabesequenz stoppen, indem Sie den SFB-Parameter "Software-Freigabe" (SW_EN = 0) zurücksetzen.

Folgende Ausgangsparameter können Sie beobachten:

Parameter	Wert	Bedeutung
STS_EN	1	Status: Freigabe
STS_STRT	1	Status: Starteingang (HW-Tor)
STS_DO	0/1	Status: Ausgang Da die Periodendauer auf 20 s eingestellt ist, wird eine Impulslänge von 5 s ausgegeben (250 % von 20 s) und eine Pause von 15 s.

Bedienung "Steuern des Ausgangs DO im Handbetrieb "

Setzen und aktivieren Sie folgende SFB-Parameter:

Parameter	Einstellung	Bedeutung
MAN_DO	1	Freigabe Ausgang (für Manual-Mode)
SET_DO	1	Steuern Ausgang

Folgende Ausgangsparameter können Sie beobachten:

Parameter	Wert	Bedeutung
STS_DO	1	Status: Ausgang Der Ausgang ist dauernd eingeschaltet, solange SET_DO = 1 ist..

Bedienung "Laden der Periodendauer"

Wenn Sie die Periodendauer z. B. auf 2000 ms einstellen wollen, setzen und aktivieren Sie folgende SFB-Parameter:

Parameter	Einstellung	Bedeutung
JOB_ID	1	Auftragsnummer "Periodendauer schreiben"
JOB_VAL	2000	Wert für Periodendauer eintragen [ms]
JOB_REQ	1	Auftragsanstoß setzen (steigende Flanke wird ausgewertet)

Folgende Ausgangsparameter können Sie beobachten:

Parameter	Wert	Bedeutung
JOB_DONE	1	Status: Neuer Auftrag kann gestartet werden
JOB_ERR	0	Status: Auftrag fehlerhaft
JOB_STAT	0	Status: Auftrags-Fehlernummer

Bedienung "Laden der Einschaltverzögerung"

Wenn Sie die Einschaltverzögerung z. B. auf 500 ms einstellen wollen, setzen und aktivieren Sie folgende SFB-Parameter:

Parameter	Einstellung	Bedeutung
JOB_ID	2	Auftragsnummer "Einschaltverzögerung schreiben"
JOB_VAL	500	Wert für Einschaltverzögerung eintragen [ms]
JOB_REQ	1	Auftragsanstoß setzen (steigende Flanke wird ausgewertet)

Folgende Ausgangsparameter können Sie beobachten:

Parameter	Wert	Bedeutung
JOB_DONE	1	Status: Neuer Auftrag kann gestartet werden
JOB_ERR	0	Status: Auftrag fehlerhaft
JOB_STAT	0	Status: Auftrags-Fehlernummer

Bedienung "Laden der Mindestimpulsdauer"

Wenn Sie die Mindestimpulsdauer z. B. auf 100 ms einstellen wollen, setzen und aktivieren Sie folgende SFB-Parameter:

Parameter	Einstellung	Bedeutung
JOB_ID	4	Auftragsnummer "Mindestimpulsdauer schreiben"
JOB_VAL	100	Wert für Mindestimpulsdauer eintragen [ms]
JOB_REQ	1	Auftragsanstoß setzen (steigende Flanke wird ausgewertet)

Folgende Ausgangsparameter können Sie beobachten:

Parameter	Wert	Bedeutung
JOB_DONE	1	Status: Neuer Auftrag kann gestartet werden
JOB_ERR	0	Status: Auftrag fehlerhaft
JOB_STAT	0	Status: Auftrags-Fehlernummer

2.4.2 PWM 2 Aufheizen einer Flüssigkeit

Ziel:

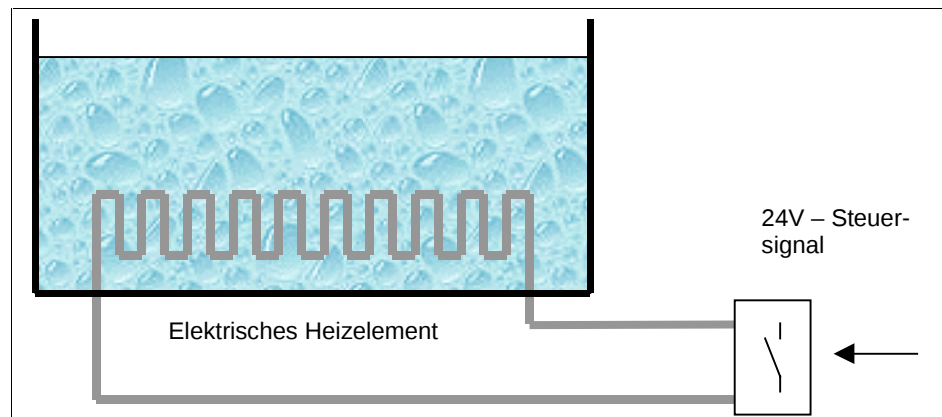
Mit dem vorliegenden Beispiel soll eine praxisrelevante Anwendung der Technologie **PWM** gezeigt werden.

Aufgabenstellung:

Mit einem elektrischen Heizelement wird eine Flüssigkeit erwärmt. Die dazu notwendige Energie wird über ein Schaltelement (z. B. Schütz) dem Heizelement zugeführt.

Die CPU 31xC erzeugt an ihrem Digitalausgang ein 24 V-Steuersignal für das Schaltelement. Die Temperatur des Heizelements ergibt sich aus der Ein-/Ausschaltdauer des 24 V-Steuersignals.

Je länger das 24 V-Steuersignal eingeschaltet ist, umso länger ist der Heizvorgang und dementsprechend größer die Temperaturerhöhung der Flüssigkeit.

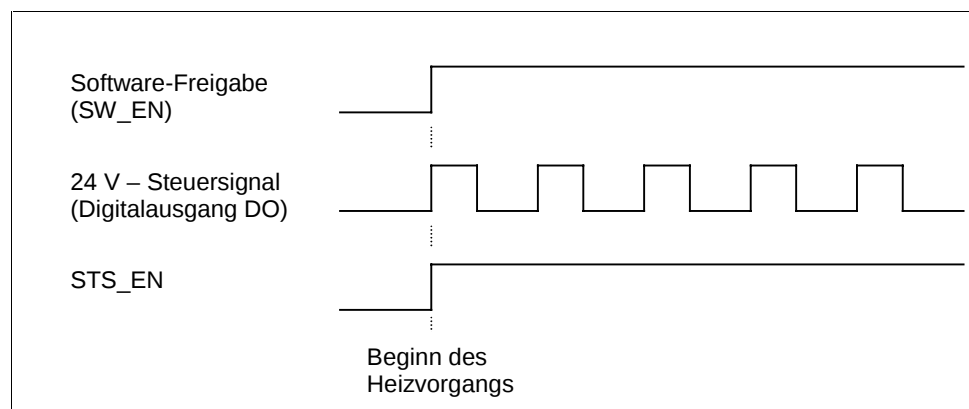


Voraussetzungen

- Das Heizelement hat nur zwei Schaltzustände: EIN oder AUS.
- Der effektive Heizstrom entspricht dem Verhältnis der Einschaltdauer/Ausschaltdauer des 24 V-Steuersignals.
- Die Mindestimpuls- bzw. Mindestpausendauer muss größer sein als die Reaktionszeiten von Schaltelement und Heizelement.

Ablauf

Starten des Heizvorgangs: Benutzen Sie zum Starten des Heizvorgangs die Software-Freigabe (SW_EN) Ihres Steuerungsprogramms.



Bausteinstruktur:

Das Beispiel "PWM 2 Aufheizen" enthält folgende Bausteine:

Baustein	Name (in der Symbolleiste)	Beschreibung
OB 1	CYCLE_EXC	Organisationsbaustein
FB 32	HEATING	Beispiel 2: Aufheizen einer Flüssigkeit
DB 32	DI_GETST_P	Instanz-DB zu HEATING
SFB 49	PULSE	Systemfunktionsbaustein PULSE
DB 36	DI_PULSE	Instanz-DB zu SFB PULSE

Anschlüsse

Nachfolgend sind die Anschlüsse des Beispiels aufgelistet:

Anschluss CPU 312C: X1	Eingang/ Ausgang	Funktion im Beispiel
12	2 M	Bezugspotential der Versorgungsspannung
13	1 L+	Versorgungsspannung 24 V DC
14	DO+0.0	<u>Ausgang DO</u> 24 V-Steuersignal (z. B. für Schütz oder TRIAC)
20	1 M	Bezugspotential der Versorgungsspannung

Anschluss CPU 313C-2 DP/PtP: X1; CPU 313C, 314C-2 DP/PtP: X2	Eingang/ Ausgang	Funktion im Beispiel
1	1 L+	Versorgungsspannung 24 V DC
20	1 M	Bezugspotential der Versorgungsspannung
21	2 L+	Versorgungsspannung 24 V DC
22	DO+0.0	<u>Ausgang DO</u> 24 V-Steuersignal (z. B. für Schütz oder TRIAC)
30	2 M	Bezugspotential der Versorgungsspannung

Parameter

Folgende Parameterangaben sind zum Heizen einer Flüssigkeit erforderlich:

Parameter	Eingabe
Kanal	0
Betriebsart	Pulsweitenmodulation
Ausgabeformat	Promille
Zeitbasis	Alle vorgegebenen Zeiten sind in der Auflösung von 1 ms angegeben
Einschaltverzögerung	0 ms Das 24 V-Steuersignal wird sofort mit SW_EN = 1 ausgegeben
Periodendauer	10 000 In der gewählten Zeitbasis sind dies 10 s
Funktion des Eingangs	Kein Hardwaretor verwendet
Mindestimpulsdauer	500 ms Mindestimpulsdauer In der gewählten Zeitbasis sind dies 500 ms; dies gilt auch für die Mindestimpulspause.

Bedienung "Aufheizen einer Flüssigkeit"

Sie können die folgende Variablentabelle VAT_HEATING verwenden.

Symbol	Symbolkommentar
START	Steuersignal: Heizen EIN
POWER	Sollwert: Heizleistung in %
STS_DO	Status Ausgang DO

Geben Sie folgenden Wert vor:

Parameter	Einstellung	Bedeutung
POWER	40	Sollwert: Heizleistung in %

Um das Aufheizen einzuschalten, müssen Sie den Parameter **START = 1** setzen und aktivieren.

Sie können den Status des Ausgangs DO beobachten. Bei einer Periodendauer von 10 s ist der Ausgang 4 s lang ein- und 6 s lang ausgeschaltet.

Ändern Sie nun den Sollwert der Heizleistung (POWER) und beobachten Sie den Ausgang DO.

3 Beispielprogramme für die Technologie "Punkt-zu-Punkt-Kopplung"

3.1 Einführung

Übersicht

In diesem Kapitel finden Sie Anwendungsbeispiele zur Technologie Punkt-zu-Punkt-Kopplung.

Die Beispiele befinden sich auf der Ihrer Dokumentation beigelegten CD. Falls Ihnen die CD nicht vorliegt, können Sie die Beispiele auch über das Internet beziehen. Das Projekt besteht aus mehreren kommentierten S7-Programmen verschiedener Komplexität und Zielrichtung.

Die Installation der Beispiele ist in der Liesmich.wri der CD beschrieben. Nach der Installation befinden sich die Beispiele im Katalog

`...\\STEP7\\EXAMPLES\\ZDt26_01_TF_____31xC_PtP`

Voraussetzungen

- Sie haben eine S7-Station, bestehend aus einer Stromversorgungsbaugruppe sowie einer CPU 313C-2 PtP oder CPU 314C-2 PtP aufgebaut und verdrahtet.
- Sie haben einen Kommunikationspartner (z. B. CP 341) oder Sie stecken einen "Kurzschlussstecker", d. h. die Sendeleitung wird auf die Empfangsleitung gebrückt.
- Auf Ihrem PG ist STEP 7 (>= V5.1 + Service Pack 2) korrekt installiert.
- Das PG ist an die CPU angeschlossen.

Vorbereiten der Beispiele

1. Öffnen Sie das Beispielprojekt **ZDt26_01_TF_____31xC_PtP** im Ordner **...\\STEP7\\EXAMPLES** mit dem SIMATIC Manager und kopieren Sie es unter einem geeigneten Namen in Ihr Projektverzeichnis (**Datei > Speichern unter**). Nutzen Sie die Darstellung **Ansicht > Detail** zur Anzeige der kompletten Informationen.
2. Fügen Sie in Ihr Projekt eine Station entsprechend Ihrem Hardware-Aufbau ein.
3. Wählen Sie ein Beispielprogramm aus und kopieren Sie das S7-Programm in diese Station.
4. Konfigurieren Sie die Hardware mit **HW Konfig** und wählen Sie dabei das TF-Submodul **PtP** aus.
5. Parametrieren Sie dieses Submodul anhand der Beispielparameter.
6. Speichern Sie die HW-Konfiguration und laden Sie diese in die CPU.
7. Laden Sie den Bausteinbehälter in die CPU.

Code der Beispiele

Die Beispiele sind in AWL/FUP geschrieben. Sie können sie direkt über den KOP/AWL/ FUP-Editor anschauen.

Wählen Sie in diesem Editor über **Ansicht > Anzeigen mit** "Symbolischer Darstellung", "Symbolauswahl" und "Kommentar". Wenn Sie genügend Platz auf dem Bildschirm haben, können Sie sich auch noch die "Symbolinformation" anzeigen lassen.

Anwenden eines Beispiels

In den Beispielprogrammen sind Variablentabellen (VAT) vorhanden, mit denen Sie die Werte ansehen und verändern können.

Weiterverwenden eines Beispiels

Sie können den Code der Beispiele direkt als Anwenderprogramm weiterverwenden.

Der Code der Beispiele ist jedoch nicht optimiert und auch nicht für alle Möglichkeiten ausgelegt. Fehlerauswertungen sind in den Beispielprogrammen nicht ausführlich ausprogrammiert, um die Programme nicht zu umfangreich werden zu lassen.

3.2 Punkt-zu-Punkt-Kopplung

Das Beispielprojekt besteht aus drei eigenständigen Komponenten:

1. **CPU 31xC ASCII:** Kopplung mit SEND und RCV für ASCII/3964(R)
2. **CPU 31xC ASCII BCC:** Kopplung für ASCII mit Blockcheck-Bildung
3. **CPU 31xC RK512:** Kopplung mit SEND und SERVE für RK512

Das Programmbeispiel "CPU 31xC ASCII" können Sie ohne Änderungen im Anwenderprogramm betreiben mit:

- ASCII-Treiber mit Endekriterium "Nach Ablauf der Zeichenverzugszeit"
- ASCII-Treiber mit Endekriterium "Nach Empfang einer festen Telegrammlänge"
- ASCII-Treiber mit Endekriterium "Nach Empfang der/des Endezeichen(s)". Im Anwenderprogramm müssen Sie zusätzlich noch die Endezeichen programmieren.
- Prozedur 3964(R)

Bausteinstruktur

In den folgenden Tabellen finden Sie die für die Beispielprogramme verwendeten Bausteine:

CPU 31xC ASCII:

Baustein	Symbol	Kommentar
OB 1	CYCLE	Zyklische Programmbearbeitung
OB 100	RESTART	Anlaufbearbeitung Neustart
FC 21	SEND	Daten senden
FC 22	RCV	Daten empfangen
DB 21	SEND IDB	Instanz-DB für SFB SEND_PTP
DB 22	RCV IDB	Instanz-DB für SFB RCV_PTP
DB 40	SEND WORK DB	Arbeits-DB für SFB SEND_PTP
DB 41	RCV WORK DB	Arbeits-DB für SFB RCV_PTP
DB 42	SEND SRC DB	Sende-Datenbaustein
DB 43	RCV DST DB	Empfangs-Datenbaustein
SFB 60	SEND_PTP	SFB für Daten senden
SFB 61	RCV_PTP	SFB für Daten empfangen

CPU 31xC ASCII BCC:

Baustein	Symbol	Kommentar
OB 1	CYCLE	Zyklische Programmbearbeitung
OB 100	RESTART	Anlaufbearbeitung Neustart
FC 21	SEND	Daten senden
FC 22	RECV	Daten empfangen
FC 23	GEN BCC	Erzeugt BCC für den Sender
FC 24	CHK BCC	Überprüft BCC beim Empfang
DB 21	SEND IDB	Instanz-DB für SFB SEND_PTP
DB 22	RECV IDB	Instanz-DB für SFB RCV_PTP
DB 40	SEND WORK DB	Arbeits-DB für SFB SEND_PTP
DB 41	RECV WORK DB	Arbeits-DB für SFB RCV_PTP
DB 42	SEND SRC DB	Sende-Datenbaustein
DB 43	RECV DST DB	Empfangs-Datenbaustein
SFB 60	SEND_PTP	SFB für Daten senden
SFB 61	RCV_PTP	SFB für Daten empfangen

CPU 31xC RK512:

Baustein	Symbol	Kommentar
OB 1	CYCLE	Zyklische Programmbearbeitung
OB 100	RESTART	Anlaufbearbeitung Neustart
FC 21	SEND	Daten senden
FC 22	SERVE	Daten empfangen
DB 21	SEND IDB	Instanz-DB für SFB SEND_RK
DB 22	SERVE IDB	Instanz-DB für SFB SERVE_RK
DB 40	SEND WORK DB	Arbeits-DB für SFB SEND_RK
DB 41	SERVE WORK DB	Arbeits-DB für SFB SERVE_RK
DB 42	SEND SRC DB	Sende-Datenbaustein
DB 43	SERVE DST DB	Empfangs-Datenbaustein
DB 80	SYNC DB	Synchronisations-Datenbaustein
SFB 63	SEND_RK	SFB für Daten senden mit RK 512
SFB 65	SERVE_RK	SFB für Daten empfangen mit RK 512

Fehlerverhalten

Ist im Anlauf ein Fehler aufgetreten, werden die zyklisch bearbeiteten Bausteinaufrufe mit Fehler beendet.

Bei einer Fehlermeldung wird der Parameterausgang ERROR der Bausteine gesetzt. Eine genauere Fehlerbeschreibung ist dann im Parameter STATUS der Bausteine hinterlegt.

Anlauf

Das Anlaufprogramm steht im OB 100. Im Anlauf werden die Steuerbits und die Zähler rückgesetzt.

3.2.1 CPU 31xC ASCII

Zyklisches Programm

Das zyklische Programm für die Kopplung mit SEND und RCV für ASCII/3964(R) steht im OB 1.

Im Beispiel arbeiten die Funktionsbausteine SFB 61 "RCV_PTP" und SFB 60 "SEND_PTP" mit den Funktionen FC 21 und FC 22 zusammen, sowie mit den Datenbausteinen DB 21 und DB 22 als Instanz-DBs und DB 42 und DB 43 als Sende- bzw. Empfangs-DB.

Die Parametrierung der Funktionsbausteine erfolgt im Beispiel teils durch Konstanten und teils durch symbolisch adressierte Aktualoperanden.

Datenübertragung

Die Datenübertragung findet statt von der CPU zu Ihrem Kommunikationspartner.

Beschreibung FC 21 (SEND)

- Programmteil "Flanke erzeugen für SEND_REQ":
Der "SEND_PTP" wird am Anfang einmal mit SEND_REQ = FALSE durchlaufen. Danach wird SEND_REQ auf TRUE gesetzt. Wenn am Steuerparameter SEND_REQ ein positiver Flankenwechsel erkannt wird, wird der "SEND_PTP" Auftrag gestartet.

Mit SEND_DONE = TRUE oder SEND_ERROR = TRUE wird SEND_REQ wieder auf FALSE gesetzt.
- Programmteil "Fertig ohne Fehler, SEND_DONE = TRUE":
Bei einem erfolgreichen Transfer wird am Parameterausgang des "SEND_PTP" der Parameter SEND_DONE auf TRUE gesetzt.
Um zeitlich nacheinander ablaufende Transfers zu unterscheiden, wird im Datenwort 0 des Quellbausteins DB 42 ein Sendezähler SEND_COUNTER_OK mitgeführt.
- Programmteil "Fertig mit Fehler, SEND_ERROR = TRUE":
Wird der "SEND_PTP" mit SEND_ERROR = TRUE durchlaufen, wird im Datenwort 2 der Fehlerzähler SEND_COUNTER_ERR hochgezählt. Außerdem wird der SEND_STATUS umkopiert, da er im nächsten Durchlauf mit 0 überschrieben wird und dann nicht mehr ausgelesen werden könnte.

Beschreibung FC 22 (RCV)

- Programmteil "Datenempfang freigeben":
Um Daten empfangen zu können, muss die Empfangsfreigabe RCV_EN_R am Baustein "RCV_PTP" auf TRUE gesetzt sein.
- Programmteil "Empfangen ohne Fehler, RCV_NDR = TRUE":
Wenn RCV_NDR gesetzt ist, sind neue Daten empfangen worden und es wird der Empfangszähler RCV_COUNTER_OK hochgezählt.
- Programmteil "Empfangen mit Fehler, RCV_ERROR = TRUE":
Bei fehlerhaftem Ablauf, d. h. wenn das Errorbit am Parameterausgang des "RCV_PTP" gesetzt ist, wird der Fehlerzähler RCV_COUNTER_ERR hochgezählt. Außerdem wird der RCV_STATUS umkopiert, da er im nächsten Durchlauf mit 0 überschrieben wird und dann nicht mehr ausgelesen werden könnte.

Alle relevanten Werte können zum Test in der VAT beobachtet werden.

3.2.2 CPU 31xC ASCII BCC

Das Programmbeispiel ist nur für den ASCII-Treiber sinnvoll.

Dieses Beispiel ist identisch zu dem Beispiel "CPU 31xC ASCII" aufgebaut, beinhaltet aber zusätzlich noch

- im Sendeteil die Bildung einer Blockcheck-Summe durch den Aufruf von FC 23 (GEN BCC) und
- im Empfangsteil die Überprüfung der Blockcheck-Summe durch den Aufruf von FC 24 (CHK BCC).

Beschreibung FC 23 (GEN BCC)

Mit dem Eingangsparameter "DB_NO" wird der DB angegeben, in dem die zu sendenden Daten abgelegt sind. Der Eingangsparameter "LEN" gibt die Länge der zu sendenden Daten an. Davon sind die letzten 2 Bytes für den Blockcheck (abgelegt als ASCII-Wert) frei zu halten.

Der Blockcheck erfolgt durch XOR-Verknüpfung aller zu sendenden Bytes (LEN -2). Die XOR-Verknüpfung erfolgt in der Schleife "LOOP". Das jeweilige Byte wird durch den Befehl L DBB [#d_loop_akt] geladen. Der Offset "#d_loop_akt" muss als Bit-Offset angegeben werden. Nach Ablauf der "LOOP"-Schleife steht der Blockcheck im Lowbyte der Variablen "#w_bcc_value". Der Blockcheck wird dann wie im unteren Beispiel beschrieben in zwei ASCII-Zeichen codiert.

Beispiel:

Es soll ein Datenblock von insgesamt 22 Bytes (20 Byte Daten + 2 Byte Blockcheck-Summe) Länge gesendet werden. In der Variablen #w_bcc_value steht das Ergebnis der XOR-Verknüpfung der 20 zu sendenden Datenbytes: 0025 Hex. Dieser Wert wird nun nach folgendem Muster in 2 ASCII-Zeichen gewandelt:

Bildung von BCC-HI ASCII-Zeichen "32" =2

#w_bcc_value	0025	
UW w#16#F0	00F0	//Unteres Halbbyte ausblenden
=	<u>0020</u>	
SLW 4	0200	// Um halbes Byte nach links schieben
OW w#16#3000	3000	// 3000 Hex aufodern
=	<u>3200</u>	// BCC-HI ASCII-Zeichen

Bildung von BCC-LO ASCII-Zeichen "35" =5

#w_bcc_value	0025	
UW w#16#F	000F	//Oberes Halbbyte ausblenden
=	<u>0005</u>	
OW w#16#30	0030	// 30 Hex aufodern
=	<u>0035</u>	// BCC-LO ASCII-Zeichen

Die BCC-LO und BCC-HI ASCII-Zeichen werden miteinander verodert und bilden den BCC 3235 Hex. Anschließend wird der BCC in Wort 20 des Datenbausteins geschrieben (L DBW [#d_loop_akt]).

Beschreibung FC 24 (CHK BCC)

Das Programm des FC24 ist beinahe identisch zum FC23. Der Unterschied besteht lediglich darin, dass am Ende der berechnete Blockcheck mit dem Blockcheck der empfangenen Daten verglichen und am Ausgangsparameter #RETVAl ausgegeben wird:

- 0 Blockcheck in Ordnung bzw.
- -1 Blockcheck falsch

3.2.3 CPU 31xC RK512

Zyklisches Programm

Das zyklische Programm für die Kopplung mit SEND und SERVE für RK512 steht im OB 1.

Im Beispiel arbeiten die Funktionsbausteine SFB 63 "SEND_RK" und SFB 65 "SERVE_RK" mit den Funktionen FC 21 und FC 22 zusammen, sowie mit den Datenbausteinen DB 21 und DB 22 als Instanz-DBs und DB 42 und DB 43 als Sende- bzw. Empfangs-DB. Die Parametrierung der Funktionsbausteine erfolgt teils durch Konstanten und teils durch symbolisch adressierte Aktualoperanden.

Datenübertragung

Die Datenübertragung findet von der CPU zu Ihrem Kommunikationspartner statt.

Beschreibung FC 21 (SEND)

- Programmteil "Flanke erzeugen für SEND_REQ":
Der "SEND_RK" wird am Anfang einmal mit SEND_REQ = FALSE durchlaufen. Danach wird SEND_REQ auf TRUE gesetzt. Wenn am Steuerparameter SEND_REQ eine positive Flanke erkannt wird, wird der "SEND_RK" Auftrag gestartet. Mit SEND_DONE = TRUE oder SEND_ERROR = TRUE wird SEND_REQ wieder auf FALSE gesetzt.
- Programmteil "Fertig ohne Fehler, SEND_DONE = TRUE":
Bei einem erfolgreichen Transfer wird am Parameterausgang des "SEND_RK" der Parameter SEND_DONE auf TRUE gesetzt. Um zeitlich nacheinander ablaufende Transfers zu unterscheiden, wird im Datenwort 0 des Quellbausteins DB 42 ein Sendezähler SEND_COUNTER_OK mitgeführt.
- Programmteil "Fertig mit Fehler, SEND_ERROR = TRUE":
Wird der "SEND_RK" mit SEND_ERROR = TRUE durchlaufen, wird im Datenwort 2 der Fehlerzähler SEND_COUNTER_ERR hochgezählt. Außerdem wird der SEND_STATUS umkopiert, da er im nächsten Durchlauf mit 0 überschrieben wird und dann nicht mehr ausgelesen werden könnte.

Beschreibung FC 22 (SERVE)

- Programmteil "Serverfunktion freigeben":
Um Daten empfangen zu können, muss die Freigabe SERVE_EN_R am Baustein "SERVE_RK" auf TRUE gesetzt sein.
- Programmteil "Serverfunktion ohne Fehler, SERVE_NDR = TRUE":
Wenn SERVE_NDR gesetzt ist, sind neue Daten empfangen worden und es wird der Empfangszähler SERVE_COUNTER_OK hochgezählt.
- Programmteil "Serverfunktion mit Fehler, SERVE_ERROR = TRUE":
Bei fehlerhaftem Ablauf, d. h. wenn das Errorbit am Parameterausgang des "SERVE_RK" gesetzt ist, wird der Fehlerzähler SERVE_COUNTER_ERR hochgezählt. Außerdem wird der SERVE_STATUS umkopiert, da er im nächsten Durchlauf mit 0 überschrieben wird und dann nicht mehr ausgelesen werden könnte. Alle relevanten Werte können zum Test in der VAT beobachtet werden.

4 Beispielprogramme für die Technologie "Regeln"

4.1 Einführung

Übersicht

In diesem Kapitel finden Sie Anwendungsbeispiele für die Technologie Regeln.

Die Beispiele befinden sich auf der Ihrer Dokumentation beigefügten CD. Falls Ihnen die CD nicht vorliegt, können Sie die Beispiele auch über das Internet beziehen. Das Projekt besteht aus mehreren kommentierten S7-Programmen verschiedener Komplexität und Zielrichtung.

Die Installation der Beispiele ist in der Liesmich.wri der CD beschrieben. Nach der Installation befinden sich die Beispiele im Katalog

...\\STEP7\\EXAMPLES\\ZDt26_04_TF____31xC_PID.

Voraussetzungen

- Sie haben eine S7-Station, bestehend aus einer Stromversorgungsbaugruppe sowie einer CPU 313C oder CPU 314C aufgebaut und verdrahtet.
- Auf Ihrem PG ist STEP 7 (>= V5.1 + Service Pack 2) korrekt installiert.
- Das PG ist an die CPU angeschlossen.

Vorbereiten der Beispiele

1. Öffnen Sie das Beispielprojekt **ZDt26_04_TF____31xC_PID** im Ordner **...\\STEP7\\EXAMPLES** mit dem SIMATIC Manager und kopieren Sie es unter einem geeigneten Namen in Ihr Projektverzeichnis (**Datei > Speichern unter**).
Nutzen Sie die Darstellung **Ansicht > Detail** zur Anzeige der kompletten Informationen.
2. Fügen Sie in Ihr Projekt eine Station entsprechend Ihrem Hardware-Aufbau ein.
3. Wählen Sie ein Beispielprogramm aus und kopieren Sie das Programm in diese Station.
4. Konfigurieren Sie die Hardware vollständig mit **HW Konfig**.
5. Speichern Sie die HW-Konfiguration und laden Sie diese in die CPU.
6. Laden Sie den Bausteinbehälter in die CPU.

Code der Beispiele

Die Beispiele sind in AWL/FUP geschrieben. Sie können sie direkt über den KOP/AWL/ FUP-Editor anschauen.

Wählen Sie in diesem Editor über **Ansicht > Anzeigen mit** "Symbolischer Darstellung", "Symbolauswahl" und "Kommentar". Wenn Sie genügend Platz auf dem Bildschirm haben, können Sie sich auch noch die "Symbolinformation" anzeigen lassen.

Anwenden eines Beispiels

In den Beispielprogrammen sind Variablentabellen (VAT) vorhanden, mit denen Sie die Werte ansehen und verändern können.

Weiterverwenden eines Beispiels

Sie können den Code der Beispiele direkt als Anwenderprogramm weiterverwenden.

Der Code der Beispiele ist jedoch nicht optimiert und auch nicht für alle Möglichkeiten ausgelegt. Fehlerauswertungen sind in den Beispielprogrammen nicht ausführlich ausprogrammiert, um die Programme nicht zu umfangreich werden zu lassen.

4.2 Regeln

Für die Technologie REGELN finden Sie drei S7-Programme:

1. **Regeln 1 CONT_S:**
Beispiel zum Schrittreger CONT_S mit Streckensimulation.
2. **Regeln 2 CONT_C:**
Beispiel zum kontinuierlichen Regler CONT_C mit Streckensimulation.
3. **Regeln 3 PULSEGEN:**
Beispiel zum kontinuierlichen Regler CONT_C mit nachgeschaltetem Impulsformer PULSEGEN und Streckensimulation.

4.2.1 Regeln 1 CONT_S Schrittreger mit Streckensimulation

Ziel

Mit diesem Beispiel nehmen Sie einen Schrittreger (Festwertregler) mit Streckensimulation in Betrieb.

Sie lernen vor allem die Parametrierung des Systemfunktionsbausteins (SFB) CONT_S kennen.

Vorraussetzung

Sie haben über die HW Konfig den Weckalarm für den OB35 auf 100 ms eingestellt.

Anwendung

Das Beispiel "Regeln 1 CONT_S" umfasst einen Schrittreger (CONT_S) in Verbindung mit einer simulierten Regelstrecke, die aus einem integrierenden Stellglied und einem nachgeschalteten Verzögerungsglied dritter Ordnung (PT3) besteht.

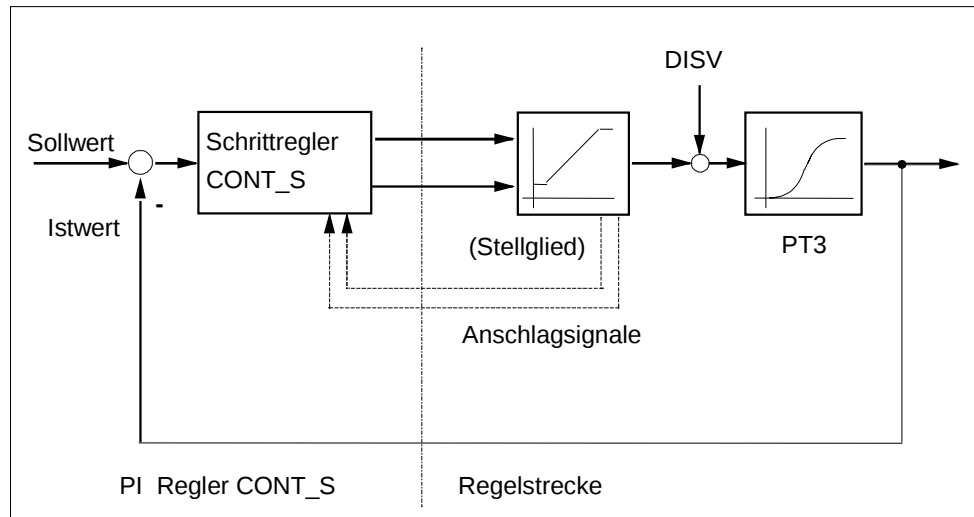
Mit Hilfe des Beispiels ist es möglich, auf einfache Weise einen Schrittreger zu generieren und diesen in allen Eigenschaften im Zusammenspiel mit einer typischen Streckenanordnung zu parametrieren und zu erproben.

Das Beispiel ermöglicht es, die Arbeitsweise und Konfiguration von Reglern mit diskontinuierlichem Ausgang, wie sie bei der Regelung von Strecken mit integrierenden Stellgliedern (z. B. Motor) sehr häufig eingesetzt werden, leicht zu verstehen. Es ist deshalb auch für Einführungs- bzw. Schulungszwecke anwendbar.

Durch entsprechende Wahl Ihrer Parameter nähern Sie die Strecke an die Eigenschaften des realen Prozesses an.

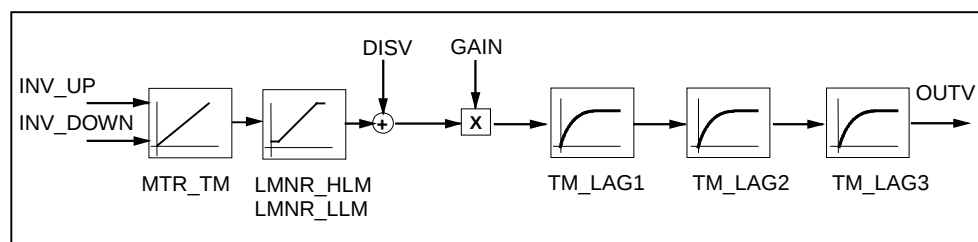
Funktionen

Das Beispiel 1 setzt sich im wesentlichen aus den beiden Funktionsbausteinen CONT_S und PROC_S zusammen. CONT_S verkörpert dabei den verwendeten Schrittregler, und PROC_S simuliert eine Regelstrecke mit den Funktionsgliedern "Ventil" und PT3. Dem Regler werden dabei neben der Regelgröße Informationen über erreichte Anschlagsignale übermittelt.



Das folgende Bild zeigt den Aufbau und die Parameter des Streckenbausteins PROC_S. Der Funktionsbaustein PROC_S bildet eine Reihenschaltung nach, die aus dem integrierenden Stellglied und drei Verzögerungsgliedern 1. Ordnung besteht. Zum Ausgangssignal des Stellgliedes wird immer die Störgröße **DISV** hinzuaddiert, so dass an dieser Stelle Streckenstörungen manuell aufgeschaltet werden können. Über den Faktor **GAIN** lässt sich die statische Streckenverstärkung bestimmen.

Der Parameter für die Motorstellzeit **MTR_TM** definiert die Zeit, welche das Stellglied für den Durchlauf von Anschlag zu Anschlag benötigt.



Bausteinstruktur

Das Beispiel 1 umfasst die Bausteine Neustart (OB100) und eine Weckalarmebene (OB35 mit 100 ms Zeittakt) in denen sowohl der Schrittreger CONT_S als auch die Streckensimulation PROC_S aufgerufen wird.

Baustein	Name (in der Symbolleiste)	Beschreibung
OB100	RESTART	Neustart-OB
OB35	CYC_INT5	Zeitgesteuerter OB: 100 ms
SFB42	CONT_S	Schrittreger
FB100	PROC_S	Strecke für Schrittreger
DB100	DI_PROC_S	Instanz-DB zu FB PROC_S
DB101	DI_CONT_S	Instanz-DB zu SFB CONT_S

Den beiden Funktionsbausteinen sind die Instanz-Datenbausteine DB100 für die Strecke und DB101 für den Regler zugeordnet.

Die Parameter des Reglerbausteins

Die Parameter des Reglerbausteins CONT_S und ihre Bedeutung sind im Handbuch beschrieben bzw. können Sie diese der SFB Online-Hilfe entnehmen.

Die Parameter des Streckenmodells

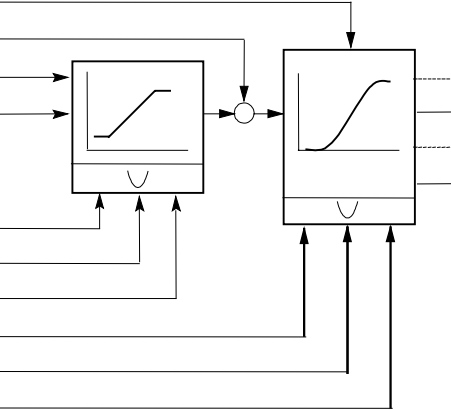
Die Parameter des Reglerbausteins CONT_S und ihre Bedeutung sind im Handbuch beschrieben bzw. können Sie der SFB Online-Hilfe entnehmen.

Die Parameter des Streckenbausteins PROC_S sind in der folgenden Tabelle aufgelistet:

Parameter	Typ	Wertebereich	Beschreibung
INV_UP	BOOL		Eingangssignal auf (mehr)
INV_DOWN	BOOL		Eingangssignal ab (weniger)
COM_RST	BOOL		Neustart
CYCLE	TIME	≥ 20 ms	Abtastzeit
DISV	REAL		Störgröße
GAIN	REAL		Streckenverstärkung
MTR_TM	TIME		Motorstellzeit
LMNR_HLM	REAL	LMNR_LLM ...100.0 [%]	obere Grenze der Stellungsrückmeldung
LMNR_LLM	REAL	-100.0... LMNR_HLM [%]	untere Grenze der Stellungsrückmeldung
TM_LAG1	TIME	$\geq$ CYCLE/2	Verzögerungszeit 1
TM_LAG2	TIME	$\geq$ CYCLE/2	Verzögerungszeit 2
TM_LAG3	TIME	$\geq$ CYCLE/2	Verzögerungszeit 3
OUTV	REAL		Ausgangsgröße
LMNR	REAL		Stellungsrückmeldung

Parameter	Typ	Wertebereich	Beschreibung
QLMNR_HS	BOOL		Stellglied am oberen Anschlag
QLMNR_LS	BOOL		Stellglied am unteren Anschlag

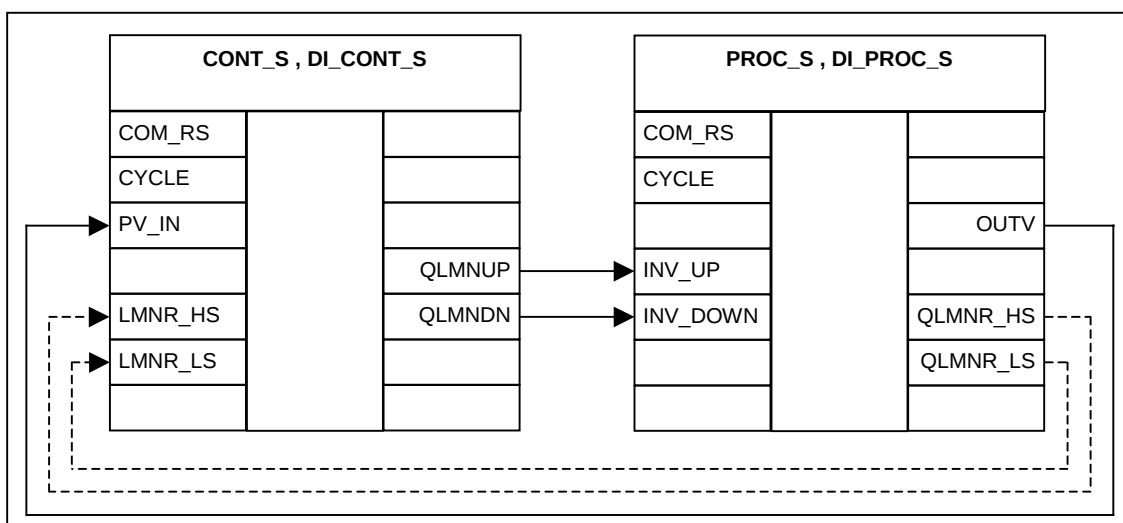
Nach **Neustart** werden die Ausgangsgröße OUTV sowie alle internen Speichergrößen auf Null gesetzt.

Eingangsparameter			PROC_S (FB100)	Ausgangsparameter			
Parameter	Typ	*)		Signal	Typ	*)	
COM_RST	BOOL	FALSE					
CYCLE	TIME	T#1s					
GAIN	REAL	0.0					
DISV	REAL	0.0					
INV_UP	BOOL	FALSE			QLMNR_HS	BOOL	FALSE
INV_DOWN	BOOL	FALSE			OUTV	REAL	0.0
					QLMNR_LS	BOOL	FALSE
					LMNR	REAL	0.0
LMNR_HLM	REAL	100.0					
LMNR_LLM	REAL	0.0					
MTR_TM	TIME	T#30s					
TM_LAG1	TIME	T#10s					
TM_LAG2	TIME	T#10s					
TM_LAG3	TIME	T#10s					

***) Vorbelegung bei Neuerstellung des Instanz-DB**

Verschaltung Schrittregler mit dem Streckenmodell

Wie der Schrittreger mit dem Streckenmodell zu einem Regelkreis verschaltet ist, geht aus dem folgenden Bild hervor:



Parameterisierung des Regelkreises

Anhand einer konkreten Parametrierung des Schrittreglers mit PI-Wirkung und eingestellter Totzone wird die Reaktion eines Regelkreises mit simulierter PT-Regelstrecke 3. Ordnung gezeigt. Die eingestellten Streckenparameter mit jeweils 10 s Verzögerungszeit bilden annähernd das Verhalten eines schnellen verfahrenstechnischen Prozesses (z. B. Temperatur- oder einer Füllstandsregelung) nach.

Einstellung einer der Verzögerungszeiten $TM_LAGx = 0$ s reduziert die Ordnung der Strecke um ein Grad.

Die folgenden Tabellen enthalten die aktuell eingestellten Werte der relevanten Parameter für Regler und Strecke:

Regler:			
Parameter	Typ	Parametrierung	Beschreibung
CYCLE	TIME	100 ms	Abtastzeit
GAIN	REAL	0.31	Proportionalbeiwert
TI	TIME	19.190 s	Integrationszeit
MTR_TM	TIME	20 s	Motorstellzeit
PULSE_TM	TIME	100 ms	Mindestimpulsdauer
BREAK_TM	TIME	100 ms	Mindestpausendauer
DEADB_W	REAL	0.5	Totzonbreite

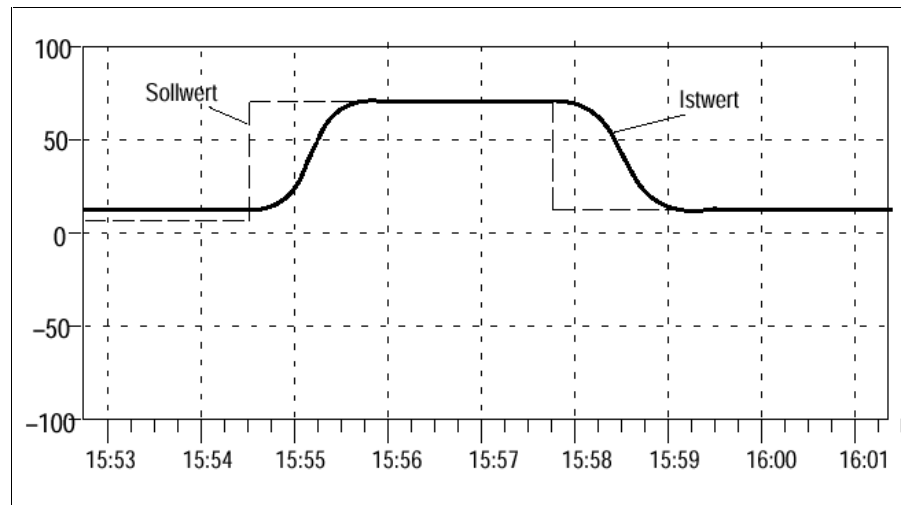
Regelstrecke:			
Parameter	Typ	Parametrierung	Beschreibung
GAIN	REAL	1.5	Streckenverstärkung
MTR_TM	TIME	20 s	Motorstellzeit
TM_LAG1	TIME	10 s	Verzögerungszeit 1
TM_LAG2	TIME	10 s	Verzögerungszeit 2
TM_LAG3	TIME	10 s	Verzögerungszeit 3

Sprungantwort beobachten

In der Variablentabelle können Sie über Setzen und Aktivieren eines neuen Sollwertes **SP_INT** das Verhalten des Ausgangs der Strecke **OUTV** (Istwert) beobachten.

Mit Hilfe des STEP 7-Werkzeugs **PID Control** können Sie die Verläufe der Reglergrößen auch grafisch verfolgen, wie im Bild unten dargestellt. Hierzu müssen Sie in **PID Control** Ihr Beispielprojekt öffnen und die Betriebsart "Online" anklicken. Wählen Sie im Ordner Bausteine den Instanz-DB 101 vom SFB 42 CONT_S an, und klicken Sie auf OK. Aktivieren Sie die Funktion **Kurvenschreiber**.

Das folgende Kurvendiagramm zeigt das Übergangs- und Einschwingverhalten des geschlossenen Regelkreises nach einer Sollwertänderung von 60 Prozent:



4.2.2 Regeln 2 CONT_C Kontinuierlicher Regler mit Streckensimulation

Ziel

Mit diesem Beispiel nehmen Sie einen Kontinuierlichen Regler in Betrieb.

Sie lernen vor allem die Parametrierung des Systemfunktionsbausteins (SFB) CONT_C kennen.

Vorraussetzung

Sie haben über die HW Konfig den Weckalarm für den OB35 auf 100 ms eingestellt.

Anwendung

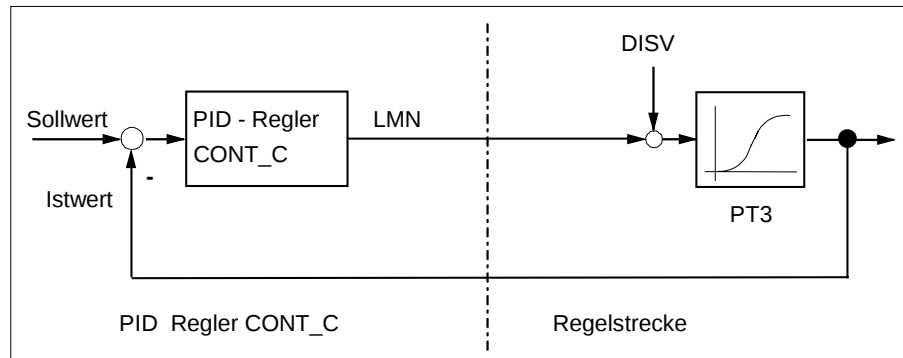
Das Beispiel 2 umfasst einen kontinuierlichen Regler (CONT_C) in Verbindung mit einer simulierten Regelstrecke, die aus einem Verzögerungsglied dritter Ordnung (PT3) besteht.

Mit Hilfe von Beispiel 2 ist es möglich, auf einfache Weise einen kontinuierlichen PID-Regler zu generieren und diesen in allen Eigenschaften im Zusammenspiel mit einer typischen Streckenanordnung zu parametrieren und zu erproben.

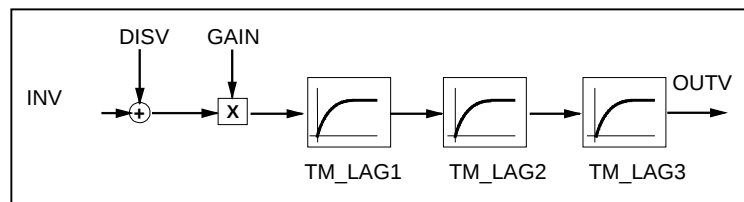
Das Beispiel ermöglicht es, die Arbeitsweise und Konfiguration von Reglern mit analogem Ausgangssignal, wie sie bei der Regelung von Strecken mit proportional wirkenden Stellgliedern eingesetzt werden, leicht zu verstehen. Es ist deshalb auch für Einführungs- bzw. Schulungszwecke anwendbar.

Funktionen

Das Beispiel 2 setzt sich im wesentlichen aus dem Systemfunktionsbaustein CONT_C und dem Funktionsbaustein PROC_C zusammen. CONT_C verkörpert dabei den verwendeten Regler, und PROC_C simuliert eine Regelstrecke mit Ausgleich dritter Ordnung.



Das folgende Bild zeigt den Aufbau und die Parameter des Streckenbausteins PROC_S. Der Systemfunktionsbaustein PROC_C bildet eine Reihenschaltung aus drei Verzögerungsgliedern 1. Ordnung nach. Zum Ausgangssignal des Stellgliedes wird immer die Störgröße **DISV** hinzuaddiert, so dass an dieser Stelle Streckenstörungen manuell aufgeschaltet werden können. Über den Faktor **GAIN** lässt sich die statische Streckenverstärkung bestimmen:



Bausteinstruktur

Das Beispiel 2 umfasst die Bausteine Neustart (OB100) und eine Weckalarmebene (OB35 mit 100 ms Zeittakt) in denen sowohl der Schrittreger CONT_C als auch die Streckensimulation PROC_C aufgerufen wird.

Baustein	Name (in der Symbolleiste)	Beschreibung
OB100	RESTART	Neustart-OB
OB35	CYC_INT5	Zeitgesteuerter OB: 100 ms
SFB41	CONT_C	Kontinuierlicher PID-Regler
FB100	PROC_C	Strecke für Kontinuierlichen Regler
DB100	DI_PROC_C	Instanz-DB zu FB PROC_C
DB101	DI_CONT_C	Instanz-DB zu SFB CONT_C

Den beiden Funktionsbausteinen sind die Instanz-Datenbausteine DB100 für die Strecke und DB101 für den Regler zugeordnet.

Die Parameter des Reglerbausteins

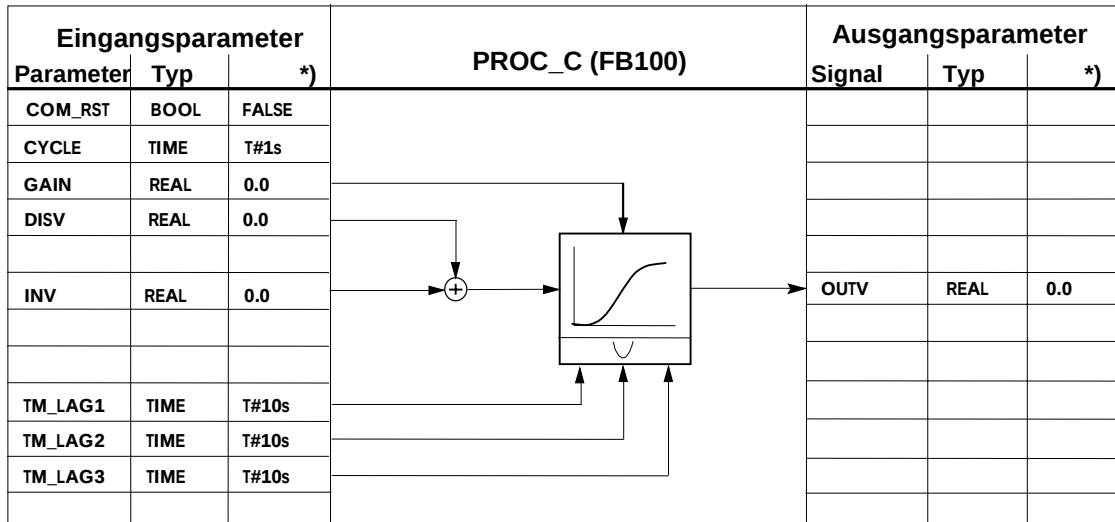
Die Parameter des Reglerbausteins CONT_C und ihre Bedeutung sind im Handbuch beschrieben bzw. können Sie der SFB Online-Hilfe entnehmen.

Die Parameter des Streckenmodells

Die Parameter des Streckenbausteins PROC_C sind in der folgenden Tabelle aufgelistet:

Parameter	Typ	Wertebereich	Beschreibung
INV	REAL		Einganggröße
COM_RST	BOOL		Neustart
CYCLE	TIME	≥ 1 ms	Abtastzeit
DISV	REAL		Störgröße
GAIN	REAL		Streckenverstärkung
TM_LAG1	TIME	$\geq \text{CYCLE}/2$	Verzögerungszeit 1
TM_LAG2	TIME	$\geq \text{CYCLE}/2$	Verzögerungszeit 2
TM_LAG3	TIME	$\geq \text{CYCLE}/2$	Verzögerungszeit 3
OUTV	REAL		Ausgangsgröße

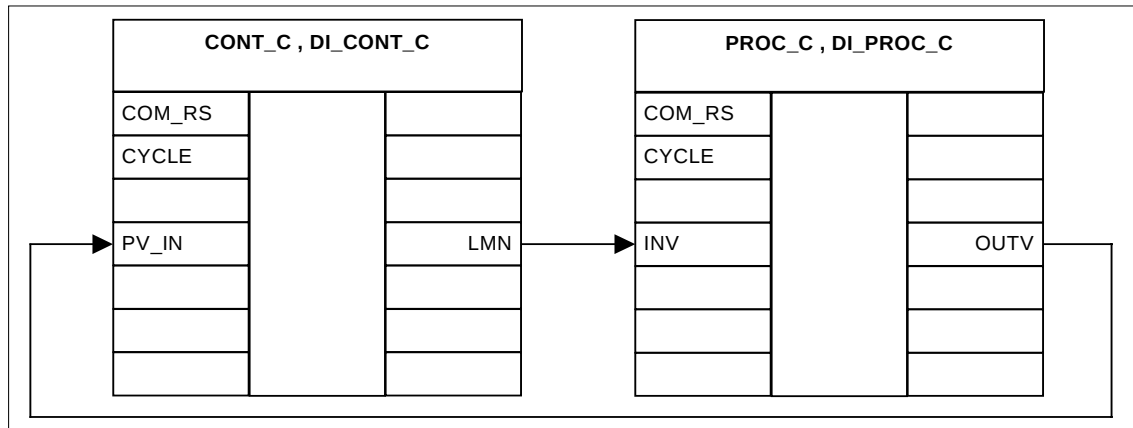
Nach **Neustart** werden die Ausgangsgröße OUTV sowie alle internen Speichergrößen auf Null gesetzt.



*) Vorbelegung bei Neuerstellung des Instanz-DB

Verschaltung kontinuierlicher Regler mit dem Streckenmodell

Wie der kontinuierliche Regler mit dem Streckenmodell zu einem Regelkreis verschaltet ist, geht aus dem folgenden Bild hervor:



Parameterisierung des Regelkreises

Anhand einer konkreten Parametrierung eines kontinuierlichen Reglers mit PID-Wirkung und eingestellter Totzone wird die Reaktion eines Regelkreises mit simulierter PT-Regelstrecke dritter Ordnung gezeigt. Die eingestellten Streckenparameter mit jeweils 10 s Verzögerungszeit bilden annähernd das Verhalten einer verfahrenstechnischen Strecke nach. Einstellung einer der Verzögerungszeiten $TM_LAGx = 0$ s reduziert die Ordnung der Strecke um ein Grad.

Die folgenden Tabellen enthalten die aktuell eingestellten Werte der relevanten Parameter für Regler und Strecke:

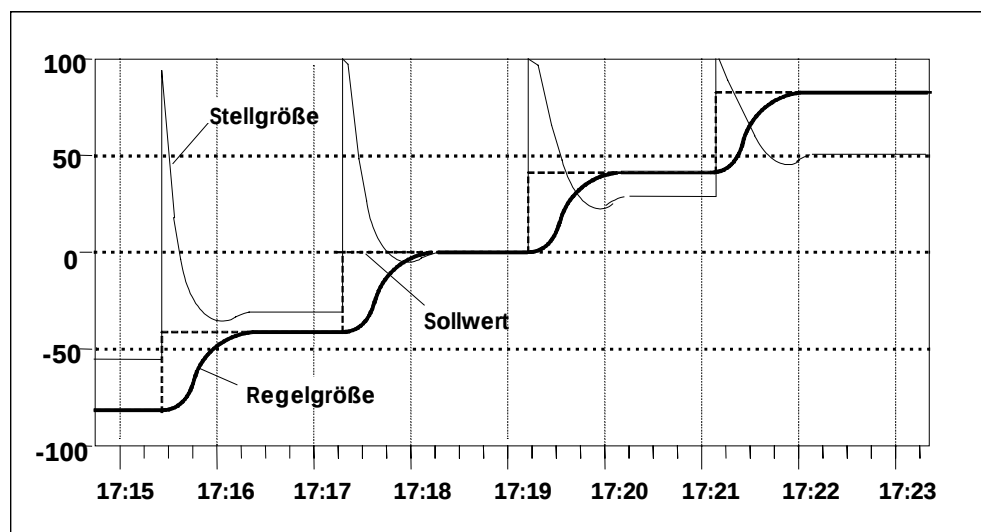
Regler:			
Parameter	Typ	Parametrierung	Beschreibung
CYCLE	TIME	100m s	Abtastzeit
GAIN	REAL	1.535	Proportionalbeiwert
TI	TIME	22.720 s	Integrationszeit
TD	TIME	5.974 s	Differenzierzeit
TM_LAG	TIME	1.195 s	Verzögerungszeit des D-Anteils
LMN_HLM	REAL	100.0	Stellwert obere Begrenzung
LMN_LLM	REAL	-100.0	Stellwert untere Begrenzung

Regelstrecke:			
Parameter	Typ	Parametrierung	Beschreibung
GAIN	REAL	1.5	Streckenverstärkung
TM_LAG1	TIME	10 s	Verzögerungszeit 1
TM_LAG2	TIME	10 s	Verzögerungszeit 2
TM_LAG3	TIME	10 s	Verzögerungszeit 3

Sprungantwort beobachten

In der Variablentabelle können Sie über Setzen und Aktivieren eines neuen Sollwertes **SP_INT** das Verhalten des Ausgangs der Strecke **OUTV** (Istwert) beobachten.

Mit Hilfe des STEP 7-Werkzeugs **PID Control** können Sie die Verläufe der Reglergrößen auch grafisch verfolgen, wie im Bild unten dargestellt. Hierzu müssen Sie in **PID Control** Ihr Beispielprojekt öffnen und die Betriebsart "Online" anklicken. Wählen Sie im Ordner Bausteine den Instanz-DB 101 vom SFB 41 CONT_C an, und klicken Sie auf OK. Aktivieren Sie die Funktion **Kurvenschreiber**. Das Kurvendiagramm zeigt das Übergangs- und Einschwingverhalten des geschlossenen Regelkreises nach einer Reihe von Sollwertänderungen von jeweils 20 Prozent.



4.2.3 Regeln 3 PULSEGEN Kontinuierlicher Regler CONT_C mit nachgeschaltetem Impulsformer PULSEGEN und Streckensimulation

Ziel

Mit diesem Beispiel nehmen Sie einen Kontinuierlichen Regler CONT_C mit nachgeschaltetem Impulsformer PULSEGEN in Betrieb.

Sie lernen vor allem die Parametrierung des Systemfunktionsbausteins (SFB) CONT_C und PULSEGEN kennen.

Vorraussetzung

Sie haben über die HW Konfig den Weckalarm für den OB35 auf 100 ms eingestellt.

Anwendung

Das Beispiel 3 umfasst einen kontinuierlichen Regler (CONT_C) mit nachgeschaltetem Impulsformer PULSEGEN und simulierter Regelstrecke, die aus einem Verzögerungsglied dritter Ordnung (PT3) besteht.

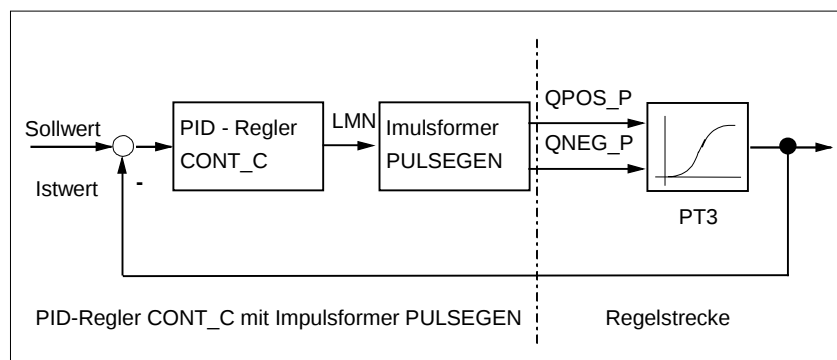
Mit Hilfe von Beispiels 3 ist es möglich, auf einfache Weise einen kontinuierlichen PID-Regler mit Impulsausgängen zu generieren und diesen in allen Eigenschaften im Zusammenspiel mit einer typischen Streckenanordnung zu parametrieren und zu erproben.

Dieses Beispiel ermöglicht es, die Arbeitsweise und Konfiguration von Reglern mit binären Impulsausgängen, wie sie bei der Regelung von Strecken mit proportional wirkenden Stellgliedern eingesetzt werden, leicht zu verstehen. Solche Regler kommen z. B. bei Temperaturstrecken mit elektrischer Heizung und Kühlung zum Einsatz. Es ist deshalb auch für Einführungs- bzw. Schulungszwecke anwendbar.

Funktionen

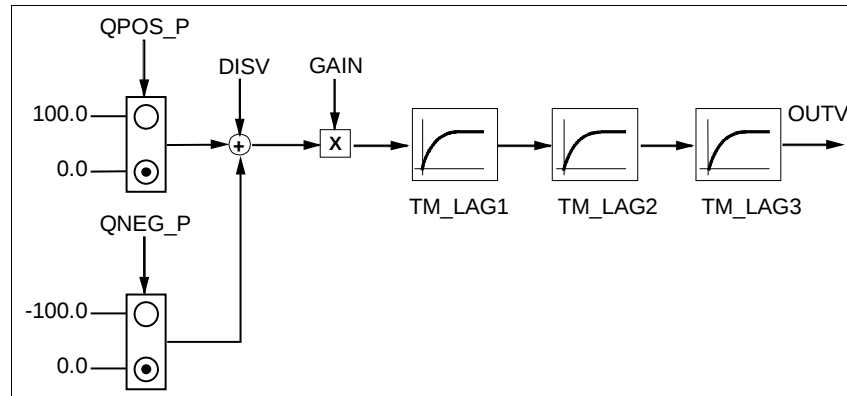
Das Beispiel 3 setzt sich im wesentlichen aus den Funktionsbausteinen CONT_C, PULSEGEN und PROC_P zusammen.

CONT_C verkörpert dabei den verwendeten Regler, PULSEGEN den Impulsformer und PROC_P simuliert eine Regelstrecke mit Ausgleich dritter Ordnung:



Das folgende Bild zeigt den Aufbau und die Parameter des Streckenbausteins PROC_P. Der Systemfunktionsbaustein PROC_P bildet eine Reihenschaltung aus drei Verzögerungsgliedern 1. Ordnung nach. Als Eingangssignal der Regelstrecke wirken nicht nur die Impulseingänge POS_P und NEG_P, sondern es wirkt als zusätzliches Eingangssignal die Störgröße **DISV**, so dass an dieser Stelle Streckenstörungen manuell aufgeschaltet werden können.

Über den Faktor **GAIN** lässt sich die statische Streckenverstärkung bestimmen:



Bausteinstruktur

Das Beispiel 3 besteht aus dem Funktionsbaustein PULS_CTR, der die Bausteine für den Regler enthält (CONT_C und PULSEGEN), und der simulierten Strecke PROC_P, sowie aus den Aufrufbausteinen für Neustart (OB 100) und einer Weckalarmebene (OB 35 mit 100 ms Zeittakt).

Der Bausteinaufruf ist so realisiert, dass CONT_C alle 2 s (= CYCLE * RED_FAC) und PULSEGEN alle 100 ms (= CYCLE) aufgerufen wird. Die Zykluszeit des OB 35 ist auf 100 ms eingestellt.

Bei Neustart (Warmstart) wird der Baustein PULS_CTR im OB100 aufgerufen und der Eingang COM_RST auf TRUE gesetzt.

Baustein	Name (in der Symbolleiste)	Beschreibung
OB 100	RESTART	Neustart-OB
OB 35	CYC_INT5	Zeitgesteuerter OB: 100 ms
SFB 41	CONT_C	Kontinuierlicher PID-Regler
SFB 43	PULSEGEN	Impulsformer
FB 100	PROC_P	Strecke für kontinuierlichen Regler mit Impulseingängen
DB 100	DI_PROC_P	Instanz-DB zu FB PROC_P
FB 101	PULS_CTR	Aufruf der SFBs CONT_C und PULSGEN
DB 101	DI_PULS_CTR	Instanz-DB zu FB PULS_CTR

Dem Funktionsbaustein PULS_CTR ist der Instanz-Datenbaustein DB 101, dem Funktionsbaustein PROC_P der DB 100 zugeordnet.

Die Parameter des Reglers

Die Parameter des Reglerbausteins CONT_C und des Impulsformers PULSEGEN sind im Handbuch beschrieben bzw. können Sie diese der SFB Online-Hilfe entnehmen.

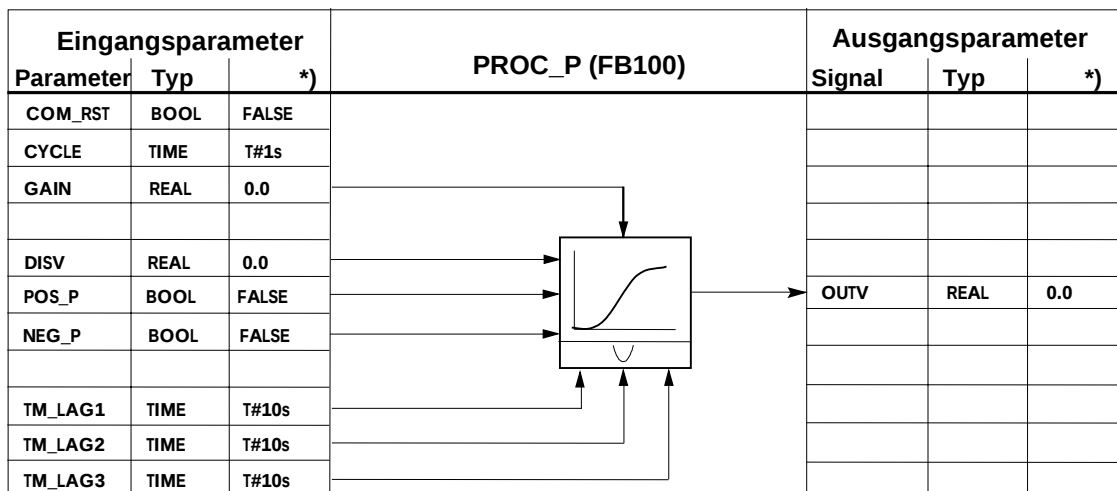
Die Parameter des Streckenmodells

Die Parameter des Reglerbausteins CONT_C und des Impulsformers PULSEGEN sind im Handbuch beschrieben bzw. können Sie der SFB Online-Hilfe entnehmen.

Die Parameter des Streckenbausteins PROC_P sind in der folgenden Tabelle aufgelistet:

Parameter	Typ	Wertebereich	Beschreibung
POS_P	BOOL		Positiver Impuls
NEG_P	BOOL		Negativer Impuls
COM_RST	BOOL		Neustart
CYCLE	TIME	$\geq 1 \text{ ms}$	Abtastzeit
DISV	REAL		Störgröße
GAIN	REAL		Streckenverstärkung
TM_LAG1	TIME	$\geq \text{CYCLE}/2$	Verzögerungszeit 1
TM_LAG2	TIME	$\geq \text{CYCLE}/2$	Verzögerungszeit 2
TM_LAG3	TIME	$\geq \text{CYCLE}/2$	Verzögerungszeit 3
OUTV	REAL		Ausgangsgröße

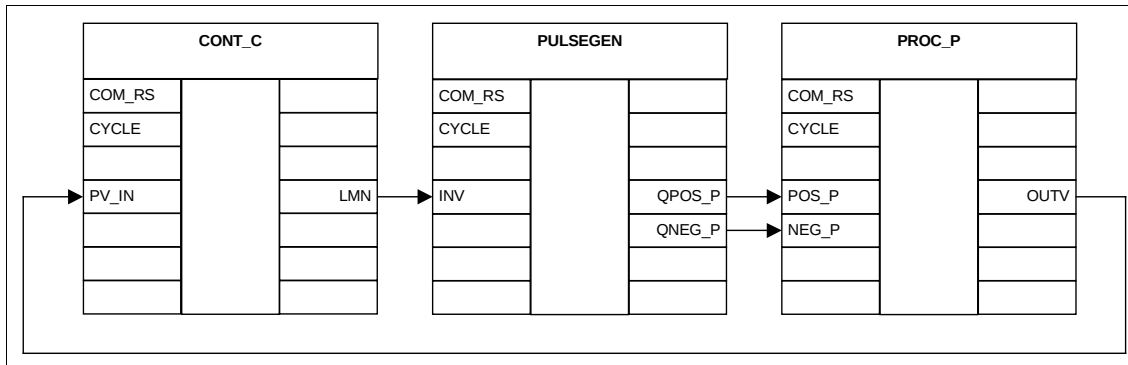
Nach **Neustart** werden die Ausgangsgröße OUTV sowie alle internen Speichergrößen auf Null gesetzt.



*) Vorbelegung bei Neuerstellung des Instanz-DB

Verschaltung kontinuierlicher Regler mit dem Streckenmodell

Wie der kontinuierliche Regler mit dem Streckenmodell zu einem Regelkreis verschaltet ist, geht aus dem folgenden Bild hervor:



Parameterisierung des Regelkreises

Anhand einer konkreten Parametrierung eines kontinuierlichen Reglers mit PID-Wirkung wird die Reaktion eines Regelkreises mit simulierter PT-Regelstrecke 3. Ordnung gezeigt. Die eingestellten Streckenparameter mit jeweils 10 s Verzögerungszeit realisieren eine schnellere Regelstrecke, wie es z. B. bei einer Temperaturregelung in der Praxis der Fall ist. Anhand der relativ schnellen Strecke lässt sich jedoch die Funktion des Reglers schneller testen. Durch Änderung der Verzögerungszeitkonstanten kann die Eigenschaft der simulierten Strecke jedoch einfach an eine reale Regelstrecke angenähert werden.

Die folgenden Tabellen enthalten die aktuell eingestellten Werte der relevanten Parameter für Regler, Impulsformer und Strecke:

Regler:			
Parameter	Typ	Wertebereich	Beschreibung
CYCLE	TIME	2 s	Abtastzeit des Reglers
GAIN	REAL	1.535	Proportionalbeiwert
TI	TIME	22.720 s	Integrationszeit
TD	TIME	5.974 s	Differenzierzeit
TM_LAG	TIME	1.195 s	Verzögerungszeit des D-Anteils
LMN_HLM	REAL	100.0	Stellwert obere Begrenzung
LMN_LLM	REAL	-100.0	Stellwert untere Begrenzung

Impulsformer			
Parameter	Typ	Wertebereich	Beschreibung
CYCLE	TIME	100 ms	Abtastzeit
P_B_TM	TIME	100 ms	Mindestimpuls- bzw. Mindestpausendauer

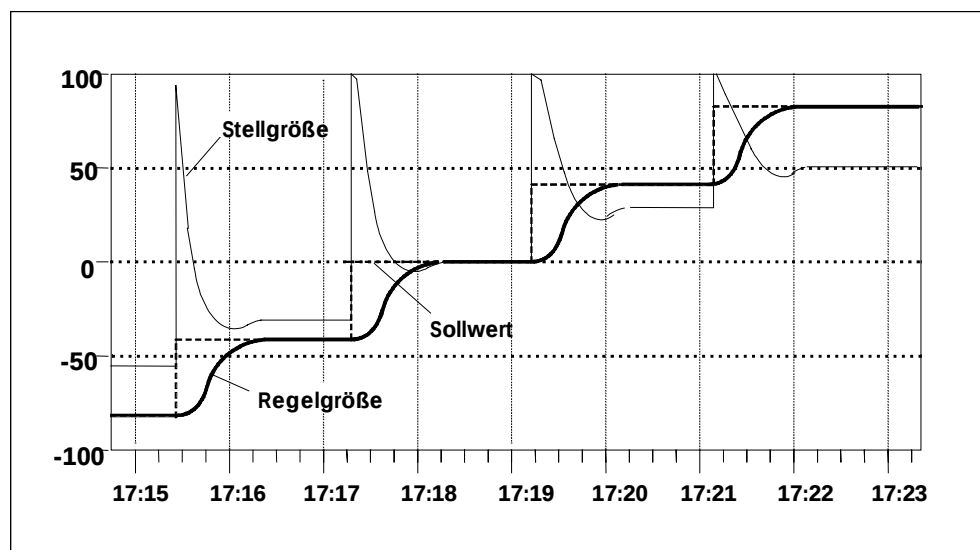
Regelstrecke:			
Parameter	Typ	Wertebereich	Beschreibung
GAIN	REAL	1.5	Streckenverstärkung
TM_LAG1	TIME	10 s	Verzögerungszeit 1
TM_LAG2	TIME	10 s	Verzögerungszeit 2
TM_LAG3	TIME	10 s	Verzögerungszeit 3

Sprungantwort beobachten

In der Variablentabelle können Sie über Setzen und Aktivieren eines neuen Sollwertes **SP_INT** das Verhalten des Ausgangs der Strecke **OUTV** (Istwert) beobachten.

Mit Hilfe des STEP 7-Werkzeugs **PID Control** können Sie die Verläufe der Reglergrößen auch grafisch verfolgen, wie im Bild unten dargestellt. Hierzu müssen Sie in **PID Control** Ihr Beispielprojekt öffnen und auf Betriebsart "Online" klicken. Wählen Sie im Ordner Bausteine den DB 101 an, klicken Sie auf OK und tragen Sie in der darauf erscheinenden Maske den Offset 16 Byte ein, ab dem die Instanzdaten des SFB 41 im DB 101 beginnen. Aktivieren Sie die Funktion **Kurvenschreiber**.

Das Kurvendigramm zeigt das Übergangs- und Einschwingverhalten des geschlossenen Regelkreises nach einer Reihe von Sollwertänderungen von jeweils 20 % des Messbereichs. Dabei ist die kontinuierliche Stellgröße des Reglers abgebildet, nicht die Impulsausgänge.



Index

A

Analog.....	1-3
Analog 1 Erste Schritte	1-4
Analog 2 Betriebsarten	1-8
Analog 3 Ladeportal.....	1-11

C

CPU 31xC ASCII.....	3-5
CPU 31xC ASCII BCC	3-6
CPU 31xC RK512.....	3-8

D

Digital.....	1-16
Digital 1 Erste Schritte	1-17
Digital 2 Betriebsarten.....	1-21
Digital 3 Pufferspeicher	1-24

F

Frequenz 1 Erste Schritte	2-14
Frequenz 2 Drehzahlüberwachung.....	2-19

Frequenzmessen.....	2-14
---------------------	------

P

Positionieren.....	1-1
Positionieren mit Analogausgang	1-3
Positionieren mit Digitalausgängen	1-16
Pulsweitenmodulation	2-23
Punkt-zu-Punkt-Kopplung.....	3-1
PWM 1 Erste Schritte	2-23
PWM 2 Aufheizen.....	2-27

R

Regeln	4-1
Regeln 1 CONT_S.....	4-3
Regeln 2 CONT_C	4-8
Regeln 3 PULSGEN	4-13

Z

Zählen	2-1
Zählen 1 Erste Schritte	2-3
Zählen 2 Abfüllstation	2-9

