

SIEMENS

SIMATIC

Sample Programs for Technological Functions

Contents

Sample Programs for "Positioning" Technology	1
Sample Programs for "Counting, Frequency Measurement and Pulse Width Modulation" Technology	2
Sample Programs for "Point-to-Point Connection" Technology	3
Sample Programs for "Controlling" Technology	4
Index	

This Manual is part of the documentation package
with order number:

6ES7398-8FA10-8BA0

Edition 10/2001

A5E00130043-01

Safety Guidelines

This manual contains notices intended to ensure personal safety, as well as to protect the products and connected equipment against damage. These notices are highlighted by the symbols shown below and graded according to severity by the following texts:



Danger

indicates that death, severe personal injury or substantial property damage will result if proper precautions are not taken.



Warning

indicates that death, severe personal injury or substantial property damage can result if proper precautions are not taken.



Caution

indicates that minor personal injury can result if proper precautions are not taken.

Caution

indicates that property damage can result if proper precautions are not taken.

Notice

draws your attention to particularly important information on the product, handling the product, or to a particular part of the documentation.

Qualified Personnel

Only **qualified personnel** should be allowed to install and work on this equipment. Qualified persons are defined as persons who are authorized to commission, to ground and to tag circuits, equipment, and systems in accordance with established safety practices and standards.

Correct Usage

Note the following:



Warning

This device and its components may only be used for the applications described in the catalog or the technical description, and only in connection with devices or components from other manufacturers which have been approved or recommended by Siemens.

This product can only function correctly and safely if it is transported, stored, set up, and installed correctly, and operated and maintained as recommended.

Trademarks

SIMATIC®, SIMATIC HMI® and SIMATIC NET® are registered trademarks of SIEMENS AG.

Third parties using for their own purposes any other names in this document which refer to trademarks might infringe upon the rights of the trademark owners.

Copyright © Siemens AG 2001 All rights reserved

The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or design, are reserved.

Disclaimer of Liability

We have checked the contents of this manual for agreement with the hardware and software described. Since deviations cannot be precluded entirely, we cannot guarantee full agreement. However, the data in this manual are reviewed regularly and any necessary corrections included in subsequent editions. Suggestions for improvement are welcomed.

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

Siemens Aktiengesellschaft

©Siemens AG 2001
Technical data subject to change.

A5E00130043



Contents

1	Sample Programs for "Positioning" Technology	
1.1	Introduction.....	1-1
1.2	Positioning With Analog Output	1-3
1.2.1	Analog 1 First steps	1-4
1.2.2	Analog 2 Operating Modes	1-8
1.2.3	Analog 3 Loading portal	1-11
1.3	Positioning With Digital Output.....	1-16
1.3.1	Digital 1 First steps.....	1-17
1.3.2	Digital 2 Operating modes.....	1-21
1.3.3	Digital 3 Buffer memory.....	1-24
2	Sample Programs for "Counting, Frequency Measurement and Pulse Width Modulation" Technology	
2.1	Introduction.....	2-1
2.2	Counting	2-3
2.2.1	Count 1 First steps	2-3
2.2.2	Count 2 Filling station.....	2-9
2.3	Frequency Measurement	2-14
2.3.1	Frequency 1, First Steps	2-14
2.3.2	Frequency 2, speed monitoring with variable setpoint.....	2-19
2.4	Pulse width modulation	2-24
2.4.1	PWM 1, First steps.....	2-24
2.4.2	PWM 2 Heating liquids.....	2-28
3	Sample Programs for "Point-to-Point Connection" Technology	
3.1	Introduction.....	3-1
3.2	Point-to-Point Communication	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	Sample Programs for "Controlling" Technology	
4.1	Introduction.....	4-1
4.2	Controlling	4-3
4.2.1	Controlling 1 CONT_S with distance simulation.	4-3
4.2.2	Controlling 2 CONT_C, Continuous controller CONT_C with distance simulation	4-8
4.2.3	Controlling 3 PULSEGEN Continuous controller CONT_C with downstream pulse generator PULSEGEN and control system simulation.....	4-12

Index

1 Sample Programs for "Positioning" Technology

1.1 Introduction

Overview

This chapter contains examples of positioning technology.

The CD included with your documentation also contains these samples. Download the examples from the Internet if you do not have this CD_ROM available. The project consists of several commented S7 programs of diverse complexity and aiming points.

The Readme.wri on the CD describes how to install the samples. After installation the examples are filed in the catalog

...\STEP7\EXAMPLES\ZEn26_03_TF_____31xC_Pos

Prerequisites

- An installed and wired S7 station, consisting of the power supply module and a CPU 314C.
- Correct installation of STEP 7 (>= V5.1 + Service Pack 2) on your PG.
- The PG is connected to the CPU.
- You have installed hardware limit switches and EMERGENCY-OFF switches for the protection of human health and material.

How to prepare the examples

1. Using SIMATIC Manager, open the sample project **ZEn26_03_TF_31xC_Pos** in your ...\\STEP7\\EXAMPLES folder. Copy it under a suitable filename to your project directory (**File > Save as**). Select **View > Details** to display the full information.
2. Configure a station in your project according to your hardware structure.
3. Select a sample program and copy the S7 program to this station.
4. Configure all your hardware in **HW Config**. Select the TF submodule **Positioning**.
5. Configure this submodule according to your system configuration.
6. Save your hardware configuration and download it to the CPU.
7. Download the block library to the CPU.

The sample code

The samples are written in STL/FDB. You can view them directly in your LAD/STL/LAD Editor.

In this Editor, select "Symbolic layout", "Symbol selection" and "Comment" via **View > View with**. You can also display "Symbol information", provided there is enough space available on your desktop.

Using the samples

You can use view and edit values, using the VATs included in the sample programs.

Simplify your control of the run signals (e. g. jogging) by connecting pushbuttons or switches to an input byte.

For example: L EB 0 // pushbutton/switch elements for axis control
 T DB6.DBB4 // control signals for sample no. 1

Subsequent use of a sample

Subsequently, you can use the sample code directly in a user program.

However, the sample code is not optimized or intended to cover all options. Error evaluation is not extensively programmed in the sample programs in order to restrict program size.

You can edit and reduce the scope of functions in your sample program "Operating modes", giving you a copy template as frame for a basic user program.

1.2 Positioning With Analog Output

Find a demo of this positioning mode in three commented S7 programs:

- **Analog 1 First steps**
- **Analog 2 Operating modes**
- **Analog 3 Loading portal**

In our sample "Analog 1 First steps" we only call and assign parameters to the SFB. You especially learn how to configure the functions of SFB ANALOG.

The sample "Analog 2 Operating modes" is based on the previous sample. Here, you learn how to configure the SFB for the various operating modes.

The example "Analog 3 Loading portal" is a typical application, consisting of the following segments:

- Sequential control for automatic processing,
- Operating mode section with position assignment/calculation,
- Call of the system function block (SFB),
- Evaluation of "Axis in Position" (POS_RCD).

Based on the previous samples "First steps" and "Operating modes", it shows how to create an S7 application for positioning tasks.

Block structure

The following blocks are prepared for ANALOG positioning mode:

Block	Name (in the toolbar)	Description
OB 1	CYCLE_EXC	Cyclic Program
OB 100	COMPLETE RESTART	Restart: Resets control signals
FC 1	GETST_A	Sample 1: First steps with ANALOG
FB 2	MODE_A	Analog 2: Operating mode
DB 2	DI_MODE_A	Sample 2: Instance-DB to MODE_A
FB 3	PORTAL	Sample 3: Loading portal
DB 3	DI_PORTAL	Sample 3: Instance-DB to PORTAL
SFB 44	ANALOG	SFB ANALOG
DB 6	DI_ANALOG	Instance-DB to SFB ANALOG

1.2.1 Analog 1 First steps

Target

In this sample, you commission the technology **"Positioning with Analog Output"**.

You learn how to configure the system function block (SFB) ANALOG.

By calling the respectively configured SFB, you can also test the following operating modes:

- Jogging
- Reference-point approach
- Relative incremental approach
- Absolute incremental approach

Requirements

- You have assigned the parameters in the parameter assignment screen forms.
- In the COMPLETE RESTART (OB 100) block, you have adapted the values for speed, acceleration, deceleration and switchover/switch-off difference.
- No hardware error is pending (ERR = 0). You must acknowledge pending hardware errors with **ERR_A = 1**.

Startup

The control signals you can influence in the VAT, for example, are reset in the startup OB.

Operation "Jogging mode"

You can edit the parameters of SFB ANALOG in VAT_GETST_A.

Set and enable the following SFB input parameters:

Parameters	Setting	Meaning
MODE_IN	1	Operating mode 1 = Jogging
DRV_EN	1	Switch on drive enable
SPEED	xx xxx xxx	Enter the speed in pulses/s

To determine the startup direction of the drive, set and enable input parameter **DIR_P** or **DIR_M**.

Parameters	Setting	Meaning	Setting	Meaning
DIR_P	1	Jogging in Plus direction	0	Jogging in
DIR_M	0		1	Minus direction

Feedback:

Parameters	Value	Meaning
ST_ENBLD	1	Start enabled
MODE_OUT	1	Enabled/configured operating mode "Jogging"
WORKING	1	Running
ACT_POS	xx xxx xxx	Actual position value
ERROR	1	Run start/resume error
STATUS	xxxx	Error ID
ERR	xxxx	Hardware error

Operation of "Reference point approach mode"

Set and enable the following SFB input parameters:

Parameters	Setting	Meaning
MODE_IN	3	Operating mode 3 = Reference-point approach
DRV_EN	1	Switch on drive enable
SPEED	xx xxx xxx	Startup speed in pulses/s

To determine the startup direction of the drive, set and enable input parameter **DIR_P** or **DIR_M**.

Parameters	Setting	Meaning	Setting	Meaning
DIR_P	1	Reference-point approach in plus direction	0	Reference-point
DIR_M	0		1	approach in minus direction

Feedback:

Parameters	Value	Meaning
ST_ENBLD	1	Start enable
MODE_OUT	3	Enabled/configured operating mode "Reference point approach"
WORKING	1	Running
SYNC	1	Axis is synchronized
ACT_POS	xx xxx xxx	Actual position value
ERROR	1	Run start/resume error
STATUS	xxxx	Error ID
ERR	xxxx	Hardware error

Operating mode "Relative incremental approach"

Set and enable the following SFB input parameters:

Parameters	Setting	Meaning
MODE_IN	4	Operating mode 4 = Relative incremental approach
DRV_EN	1	Switch on drive enable
TARGET	xx xxx xxx	Target/distance in pulses
SPEED	xx xxx xxx	Startup speed in pulses/s

To determine the startup direction of the drive, set and enable input parameter **DIR_P** or **DIR_M**.

Parameters	Setting	Meaning	Setting	Meaning
DIR_P	1	Relative incremental approach in plus direction	0	Relative incremental approach in minus direction
DIR_M	0		1	

Feedback:

Parameters	Value	Meaning
ST_ENBLD	1	Start enable
MODE_OUT	4	Enabled/configured operating mode "Relative incremental approach"
WORKING	1	Running
POS_RCD	1	Position reached
ACT_POS	xx xxx xxx	Actual position value
ERROR	1	Run start/resume error
STATUS	xxxx	Error ID
ERR	xxxx	Hardware error

Operating "Absolute incremental approach"

Prerequisites:

The axis is synchronized (SYNC = TRUE).

Set and enable the following SFB input parameters:

Parameters	Setting	Meaning
MODE_IN	5	Operating mode 5 = Absolute incremental approach
DRV_EN	1	Switch on drive enable
TARGET	xx xxx xxx	Target/distance in pulses
SPEED	xx xxx xxx	Startup speed in pulses/s

Linear axis

Set and enable input parameter **START** to start the drive.

Parameters	Setting	Meaning
START	1	Absolute incremental approach

Rotary axis

To start the drive, you must set and enable input parameter **START**, **DIR_P** or **DIR_M**.

Parameters	Setting	Meaning
START	1	Absolute incremental approach, "taking the shortest distance" to the target
DIR_P	1	Absolute incremental approach towards the target in "Plus direction"
DIR_M	1	Absolute incremental approach towards the target in "Minus direction"

Feedback:

Parameters	Value	Meaning
ST_ENBLD	1	Start enable
MODE_OUT	5	Enabled/configured operating mode "Absolute incremental approach"
WORKING	1	Running
POS_RCD	1	Position reached
ACT_POS	xx xxx xxx	Actual position value
ERROR	1	Run start/resume error
STATUS	xxxx	Error ID
ERR	xxxx	Hardware error

1.2.2 Analog 2 Operating Modes

Target

This sample shows you how to configure the SFB ANALOG for the various operating modes.

When using an axis in your system, you mainly use a link logic or sequential control to generate control signals for jogging, reference-point approach or incremental approach mode. For example, you could directly use these control signals for run control.

To execute a run, you need to set only **one bit**.

You are also shown how to configure the jobs "Set reference point" and "Length measurement".

Requirements

- You have configured your parameters in the parameter assignment screen forms.
- In the COMPLETE RESTART (OB 100) block you have adapted the values for speed, acceleration, deceleration and switchover/switch-off difference.
- No hardware error is pending (ERR = 0). You must acknowledge pending hardware errors with **ERR_A = 1**.
- The axis must be synchronized for operation in "Absolute incremental approach" mode.
- In the parameter assignment screen form, you have set the type of edge in the "Length measurement" function parameter.

Program structure

The Operating Mode program (MODE_A) has the following structure:

- Reset of the start signals
- Operating mode parameters for reference-point approach
- Operating mode parameters for jogging
- Operating mode parameters for relative incremental approach
- Operating mode parameters for absolute incremental approach
- Set job Set reference
- Call of SFB ANALOG

Operation of "Operating mode"

In VAT_MODE_A, specify the desired position values for incremental approach and customize your slow and rapid speed run values. Set and enable the parameter S_DRV_EN.

Parameters	Setting	Meaning
Pos1a	xx xxx xxx	Setpoint Position value 1, absolute
Pos2a	xx xxx xxx	Setpoint Position value 2, absolute
Pos1r	xx xxx xxx	Setpoint Position value 1, relative
Pos2r	xx xxx xxx	Setpoint Position value 2, relative
Speed_FAST	xx xxx xxx	Setpoint Rapid speed
Speed_SLOW	xx xxx xxx	Setpoint Slow speed
Pos_REFP	xx xxx xxx	Setpoint Reference point
S_DRV_EN	1	Control signal: Drive enable

Set and enable **one** of the following control signals in order to trigger the respective function:

Parameters	Setting	Meaning
S_REF	1	Start reference point approach
S_DIR_PF	1	Start jogging forward at rapid speed
S_DIR_MF	1	Start jogging backward at rapid speed
S_DIR_PS	1	Start jogging forward at slow speed
S_DIR_MS	1	Start jogging backward at slow speed
S_POS1a	1	Approach Start Pos 1 (absolute)
S_POS2a	1	Approach Start Pos 2 (absolute)
S_POS1rP	1	Approach Start Pos 1 (forward relative)
S_POS1rM	1	Approach Start Pos 2 (backward relative)
S_REFP_EN	1	Job = Set reference point

Feedback:

Parameters	Value	Meaning
ST_ENBLD	1	Start enabled
MODE_OUT	x	Enabled/configured operating mode
WORKING	1	Running
POS_RCD	1	Position reached
SYNC	1	Axis is synchronized
ACT_POS	xx xxx xxx	Actual position value
ERROR	1	Run start/resume error
STATUS	xxxx	Error ID
ERR	xxxx	Hardware error

Length measurement

The "Length measurement" function is controlled directly at input "Length measurement" (DI+0.3).

Feedback:

Parameters	Value	Meaning
MSR_DONE	1	End of length measurement
BEG_VAL	1	Actual position value at the start of length measurement
END_VAL	1	Actual position value at the end of length measurement
LEN_VAL	1	Measured length

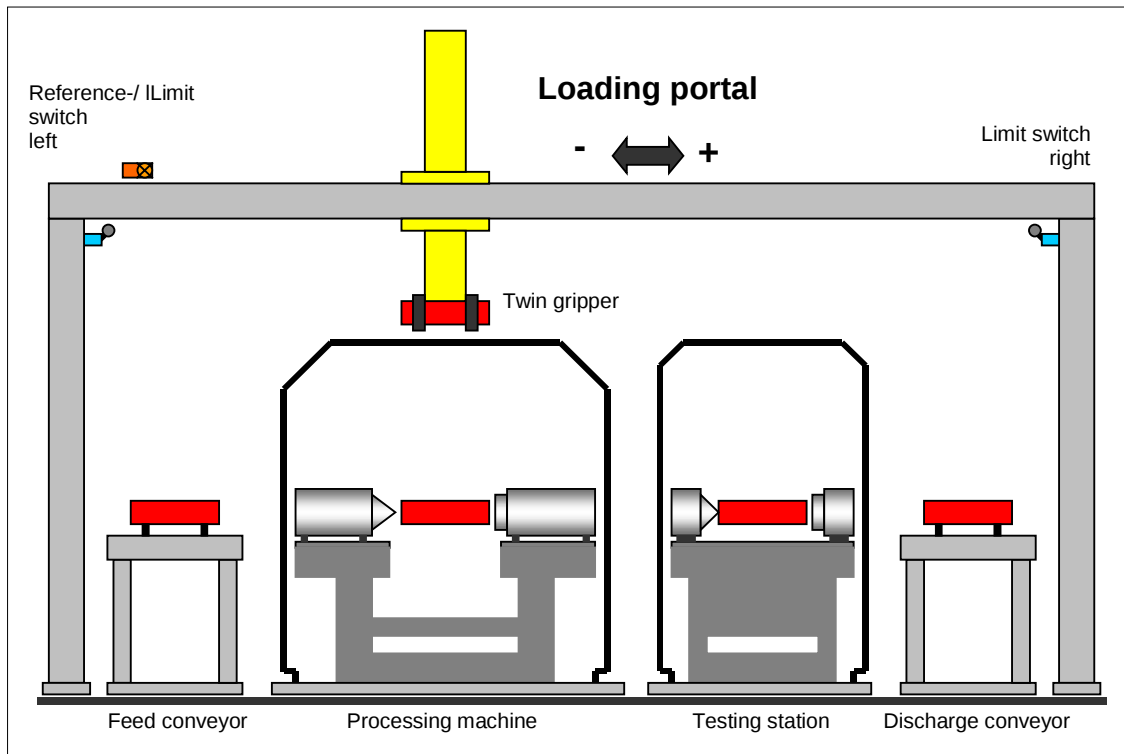
1.2.3 Analog 3 Loading portal

Target

This sample shows you an application for "**Positioning with Analog Output**" technology. The control signals for automatic mode are generated in a sequential control. Manual operation is also possible, e. g. jogging and incremental approach.

The task

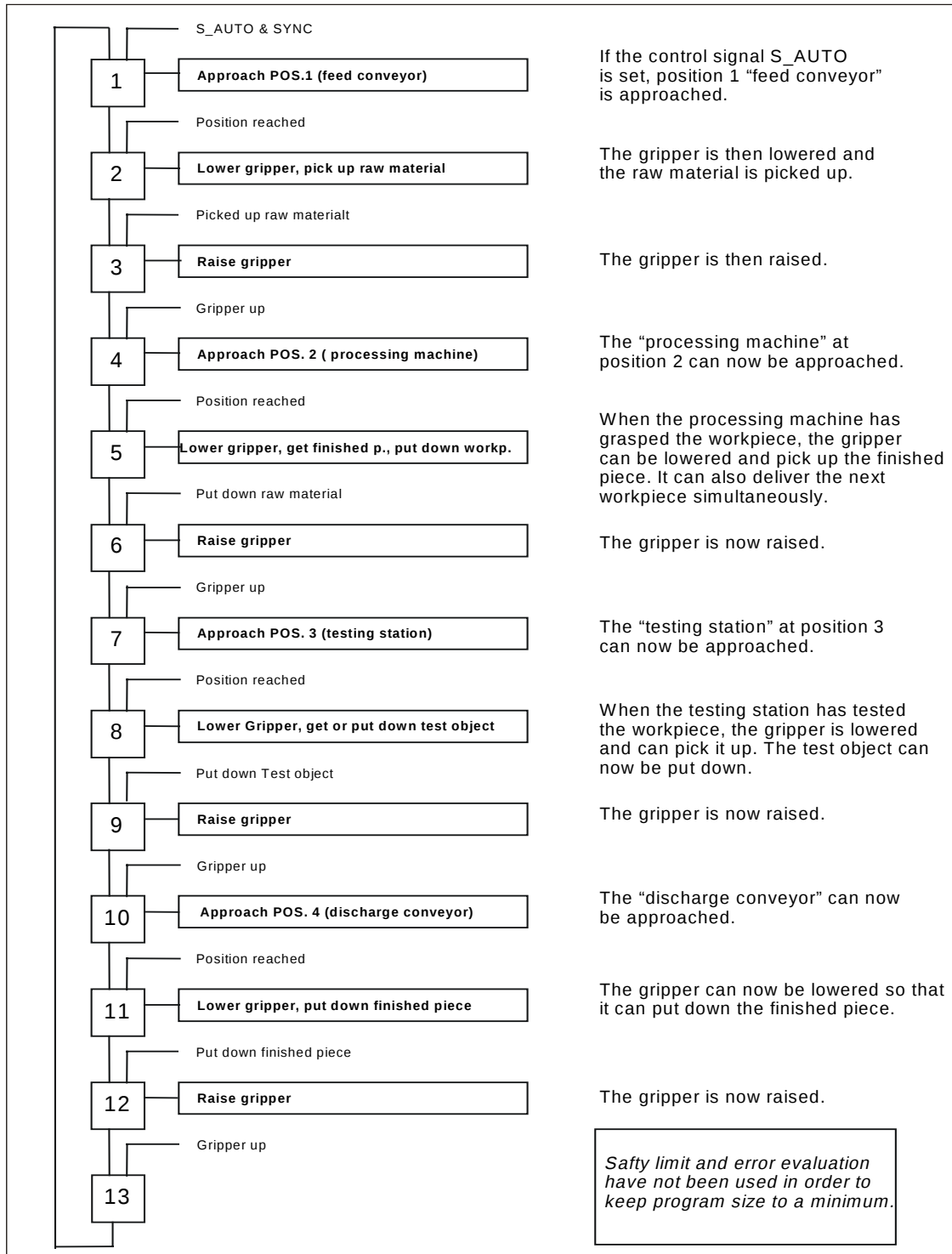
A loading portal's task is to pick up the material to be processed, to transport them to the processing machine and to the testing station and then to deliver them to the discharge conveyor.



The loading portal is equipped with a twin gripper (Gripper 1: raw material/ Gripper 2: finished part). A **servo-drive motor** moves the gripper horizontally, a pneumatic cylinder moves it vertically.

Flow chart

Reference point approach must be executed if the axis is not synchronized prior to start startup.



Conversion of the distance from mm to pulse (distance increments)

The incremental encoder supplies 2500 increment per revolution. It is installed downstream of the gears. The loading portal moves by 100 mm/revolution.

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

1 increment = 4 pulses

$0.040 \text{ mm} / 4 = 0.010 \text{ mm}$

Thus, one pulse is equivalent to a distance of 0.010 mm.

Target positions

The following position setpoints are required:

Target positions	Conversion to pulses (distance increments)
Position 1 Feed conveyor	$100 \text{ mm} / 0.01 \text{ mm/Imp.} = \mathbf{10\ 000}$ pulses
Position 2 processing machine	$1800 \text{ mm} / 0.01 \text{ mm/pulse} = \mathbf{180\ 000}$ pulses
Position 3 Testing station	$3000 \text{ mm} / 0.01 \text{ mm/Imp.} = \mathbf{300\ 000}$ pulses
Position 4 Discharge conveyor	$4500 \text{ mm} / 0.01 \text{ mm/Imp.} = \mathbf{450\ 000}$ pulses

Parameter assignment with STEP 7

Configure the CPU with STEP 7 as follows:

Parameters	Input
Type of technology	Positioning with analog output
Target range	100 pulses ($1 \text{ mm} / 0,01 \text{ mm/Imp.} = 100$)
Monitoring time	2000 ms
Maximum speed	10000 pulses/s
Creep/reference approach speed	100 pulses/s
Axis type	Linear axis
Software limit switch Start	-5000 pulses Operating range limit Start ($-50 \text{ mm} / 0.01 \text{ mm/Imp.} = -5000$ pulses)
Software limit switch End	500,000 pulses Operating range limit End ($5000 \text{ mm} / 0.01 \text{ mm/Imp.} = 500,000$ pulses)
Reference point to reference point switch	In plus direction
Increments per encoder revolution	2500
Counting direction	Standard (Track A transition before B = positive actual value)

Switchover/switch-off difference minus:

The **Changeover difference** defines the changeover position as of which the drive is operated at creep speed.

In our example we have determined a changeover difference of 50 mm.

50 mm / 0.01 mm/Imp. = **5000** pulses

The **Switch-off difference** defines the switch-off position to target distance. The drive is de-energized on reaching the cut-off position.

In order to reach the target position with maximum precision, proceed as follows:

1. In your user program, specify a switch-off difference $\geq 1/2$ target range.
2. Position the loader once in "Absolute incremental approach" mode.
3. Determine the difference between the actual value of the reached target position and the setpoint.
4. Add this difference to the switch-off difference.

The result was a left/right difference of 60 pulses.

Entries in the user program:

- Switchover difference + = **5000** pulses
- Switchover difference - = **5000** pulses
- Switch-off difference + = **60** pulses
- Switch-off difference - = **60** pulses

Note

If required, approach the target position from another direction and check the switch-off difference.

Operation of the "Loading portal"

In VAT_MODE_A, specify the desired position values for incremental approach and the slow/rapid run speed. Set and enable the parameter S_DRV_EN.

Parameters	Setting	Meaning
Pos1	xx xxx xxx	Setpoint Position value 1 (Feed conveyor)
Pos2	xx xxx xxx	Setpoint Position value 2 (Processing machine)
Pos3	xx xxx xxx	Setpoint Position value 3 (Testing station)
Pos4	xx xxx xxx	Setpoint Position value 4 (Discharge conveyor)
Speed_FAST	xx xxx xxx	Setpoint Rapid speed
Speed_SLOW	xx xxx xxx	Setpoint Slow speed
S_DRV_EN	1	Control signal: Drive enable

If the axis is not synchronized, switch the system to manual mode (S_AUTO = 0) and start reference-point approach.

Parameters	Setting	Meaning
S_AUTO	0	Manual mode = 0 / Auto mode = 1
S_REF	1	Start reference point approach

You can trigger auto mode using this signal:

Parameters	Setting	Meaning
S_AUTO	1	Auto mode

You can set and trigger the following signals, provided you have not started auto mode. (S_AUTO = 0 -> Manual mode)

Parameters	Setting	Meaning
S_REF	1	Start reference point approach
S_DIR_PF	1	Start jogging forward at rapid speed
S_DIR_MF	1	Start jogging backward at rapid speed
S_DIR_PS	1	Start jogging forward at slow speed
S_DIR_MS	1	Start jogging backward at slow speed
S_POS1	1	Approach Start Pos 1 (Feed conveyor)
S_POS2	1	Approach Start Pos 2 (Processing machine)
S_POS3	1	Approach Start Pos 3 (Testing station)
S_POS4	1	Approach Start Pos 4 (Discharge conveyor)

Feedback:

Parameters	Value	Meaning
ST_ENBLD	1	Start enable
MODE_OUT	x	Enabled/configured operating mode
WORKING	1	Running
POS_RCD	1	Position reached
SYNC	1	Axis is synchronized
ACT_POS	xx xxx xxx	Actual position value
ERROR	1	Run start/resume error
STATUS	xxxx	Error ID
ERR	xxxx	Hardware error

1.3 Positioning With Digital Output

Find examples of positioning mode in three commented S7 programs:

- **Digital 1 First steps**
- **Digital 2 Operating modes**
- **Digital 3 Buffer station**

In our sample "Digital 1 First steps" we only call and configure the SFB. You especially learn how to configure the functions of SFB DIGITAL.

The sample "Digital 2 Operating modes" is based on the previous sample. Here, you learn how to configure the SFB for the various operating modes.

The sample "Digital 3 Buffer station" is a typical application, consisting of the following segments:

- sequential control for automatic processing,
- operating mode section with position assignment/calculation,
- call of the system function block (SFB),
- evaluation of "Axis in position" (POS_RCD).

Also based on the previous samples "First steps" and "Operating modes", it shows how to create an S7 application for positioning tasks.

Block structure

The following blocks are prepared for STEP positioning mode:

Block	Name (in the toolbar)	Description
OB 1	CYCLE_EXC	Cyclic Program
OB 100	COMPLETE RESTART	Restart: Resets control signals
FC 1	GETST_D	Sample 1: First steps with DIGITAL
FB 2	MODE_D	Analog 2: Operating mode
DB 2	DI_MODE_D	Sample 2: Instance-DB to MODE_S
FB 3	STORAGE	Sample 3: Buffer station
DB 3	DI_STORAGE	Sample 3: Instance-DB to the buffer station
SFB 46	DIGITAL	System function block DIGITAL
DB 6	DI_DIGITAL	Instance-DB to SFB DIGITAL

1.3.1 Digital 1 First steps

Target

In this sample you commission the technology "**Positioning with Digital Output**".

You learn how to configure the system function block (SFB) DIGITAL.

By calling the respectively configured SFB, you can also test the following operating modes:

- Jog mode
- Reference-point approach
- Relative incremental approach
- Absolute incremental approach

Requirements

- You have assigned the parameters in the parameter assignment screen forms.
- You have adapted the switchover/switch-off difference values in the COMPLETE RESTART (OB 100) block.
- No hardware error is pending (ERR = 0). You must acknowledge pending hardware errors with **ERR_A = 1**.

Startup:

The control signals you can influence in the VAT, for example, are reset in the startup OB.

Operation "Jogging mode"

You can edit the parameters of SFB DIGITAL in VAT_GETST_A.

Set and enable the following SFB input parameters:

Parameters	Setting	Meaning
MODE_IN	1	Operating mode 1 = Jogging
DRV_EN	1	Switch on drive enable
SPEED	0/1	Speed 0 = creep speed, 1 = rapid speed

To determine the startup direction of the drive, enable input parameter **DIR_P** or **DIR_M**.

Parameters	Setting	Meaning	Setting	Meaning
DIR_P	1	Jogging in Plus direction	0	Jogging in
DIR_M	0		1	Minus direction

Feedback:

Parameters	Value	Meaning
ST_ENBLD	1	Start enable
MODE_OUT	1	Enabled/configured operating mode "Jogging"
WORKING	1	Running
ACT_POS	xx xxx xxx	Actual position value
ERROR	1	Run start/resume error
STATUS	xxxx	Error ID
ERR	xxxx	Hardware error

Operation "Reference point approach mode"

Set and enable the following SFB input parameters:

Parameters	Setting	Meaning
MODE_IN	3	Operating mode 3 = reference-point approach
DRV_EN	1	Switch on drive enable
SPEED	0/1	Startup speed 0 = creep speed, 1 = rapid speed

To determine the startup direction of the drive, set and enable input parameter **DIR_P** or **DIR_M**.

Parameters	Setting	Meaning	Setting	Meaning
DIR_P	1	Reference-point approach in plus direction	0	Reference-point approach in minus direction
DIR_M	0		1	

Feedback:

Parameters	Value	Meaning
ST_ENBLD	1	Start enable
MODE_OUT	3	Enabled/configured operating mode "Reference point approach"
WORKING	1	Running
SYNC	1	Axis is synchronized
ACT_POS	xx xxx xxx	Actual position value
ERROR	1	Run start/resume error
STATUS	xxxx	Error ID
ERR	xxxx	Hardware error

Operation of "Relative incremental approach"

Set and enable the following SFB input parameters:

Parameters	Setting	Meaning
MODE_IN	4	Operating mode 4 = Relative incremental approach
DRV_EN	1	Switch on drive enable
TARGET	xx xxx xxx	Target/distance in pulses
SPEED	0/1	Speed 0 = creep speed, 1 = rapid speed

To determine the startup direction of the drive, set and enable input parameter **DIR_P** or **DIR_M**.

Parameters	Setting	Meaning	Setting	Meaning
DIR_P	1	Relative incremental approach in plus direction	0	Relative incremental approach in minus direction
DIR_M	0		1	

Feedback:

Parameters	Value	Meaning
ST_ENBLD	1	Start enable
MODE_OUT	4	Enabled/configured operating mode "Relative incremental approach"
WORKING	1	Running
POS_RCD	1	Position reached
ACT_POS	xx xxx xxx	Actual position value
ERROR	1	Run start/resume error
STATUS	xxxx	Error ID
ERR	xxxx	Hardware error

Operation of "Absolute incremental approach"

Prerequisites:

The axis is synchronized (SYNC = TRUE).

Set and enable the following SFB input parameters:

Parameters	Setting	Meaning
MODE_IN	5	Operating mode 5 = Absolute incremental approach
DRV_EN	1	Switch on drive enable
TARGET	xx xxx xxx	Target/distance in pulses
SPEED	0/1	Speed 0 = creep speed, 1 = rapid speed

Linear axis

Set and enable input parameter **START** to start the drive.

Parameters	Setting	Meaning
START	1	Absolute incremental approach

Rotary axis

To start the drive, you must set and enable input parameter **START**, **DIR_P** or **DIR_M**.

Parameters	Setting	Meaning
START	1	Absolute incremental approach, taking the "shortest distance" to the target
DIR_P	1	Absolute incremental approach to the target in "positive direction"
DIR_M	1	Absolute incremental approach to the target in "negative direction"

Feedback:

Parameters	Value	Meaning
ST_ENBLD	1	Start enable
MODE_OUT	5	Enabled/configured operating mode "Absolute incremental approach"
WORKING	1	Running
POS_RCD	1	Position reached
ACT_POS	xx xxx xxx	Actual position value
ERROR	1	Run start/resume error
STATUS	xxxx	Error ID
ERR	xxxx	Hardware error

1.3.2 Digital 2 Operating modes

Target

This sample shows you how to configure the SFB DIGITAL for the various operating modes.

Most of the time you use a link logic or a sequential control to generate control signals for jogging, reference-point approach or incremental approach mode of an axis used in your system. For example, you could use these control signals for direct run control.

To execute a run, you need to set only **one bit**.

You can also see how to configure the jobs "Set reference point" and "Length measurement".

Requirements

- You have assigned the parameters in the parameter assignment screen forms.
- In the COMPLETE RESTART (OB 100) block you have adapted the switchover/switch-off difference values.
- No hardware error is pending (ERR = 0). You must acknowledge hardware errors with **ERR_A = 1**.
- The axis must be synchronized for operation in "Absolute incremental approach" mode.
- In the parameter assignment screen form, you have set the type of edge in the "Length measurement" function parameter.

Program structure

The Operating Mode program (MODE_A) has the following structure:

- Reset of the start signals
- Operating mode parameter for reference-point approach
- Operating mode parameter for jogging
- Operating mode parameter for relative incremental approach
- Operating mode parameter for absolute incremental approach
- Set job Set reference
- Call of SFB DIGITAL

Operation of "Operating modes"

In the variables table VAT_MODE_D, specify the desired position values for incremental approaches. Set and enable the parameter S_DRV_EN.

Parameters	Setting	Meaning
Pos1a	xx xxx xxx	Setpoint Position value 1, absolute
Pos2a	xx xxx xxx	Setpoint Position value 2, absolute
Pos1r	xx xxx xxx	Setpoint Position value 1, relative
Pos1a	xx xxx xxx	Setpoint Position value 2, relative
Pos_REFP	xx xxx xxx	Setpoint Reference point
S_DRV_EN	1	Control signal: Drive enable

Set and enable **one** of the following control signals in order to trigger the respective function:

Parameters	Setting	Meaning
S_REF	1	Start reference point approach
S_DIR_PF	1	Start jogging forward at rapid speed
S_DIR_MF	1	Start jogging backward at rapid speed
S_DIR_PS	1	Start jogging forward at slow speed
S_DIR_MS	1	Start jogging backward at slow speed
S_POS1a	1	Approach Start Pos 1 (absolute)
S_POS2a	1	Approach Start Pos 2 (absolute)
S_POS1rP	1	Approach Start Pos 1 (forward relative)
S_POS1rM	1	Approach Start Pos 2 (backward relative)
S_REFP_EN	1	Job = Set reference point

Feedback:

Parameters	Value	Meaning
ST_ENBLD	1	Start enable
MODE_OUT	x	Enabled/configured operating mode
WORKING	1	Running
POS_RCD	1	Position reached
SYNC	1	Axis is synchronized
ACT_POS	xx xxx xxx	Actual position value
ERROR	1	Run start/resume error
STATUS	xxxx	Error ID
ERR	xxxx	Hardware error

Length measurement

The "Length measurement" function is controlled directly at the "Length measurement" input (DI+0.3).

Feedback:

Parameters	Value	Meaning
MSR_DONE	1	End of length measurement
BEG_VAL	1	Actual position value at the start of length measurement
END_VAL	1	Actual position value at the end of length measurement
LEN_VAL	1	Measured length

1.3.3 Digital 3 Buffer memory

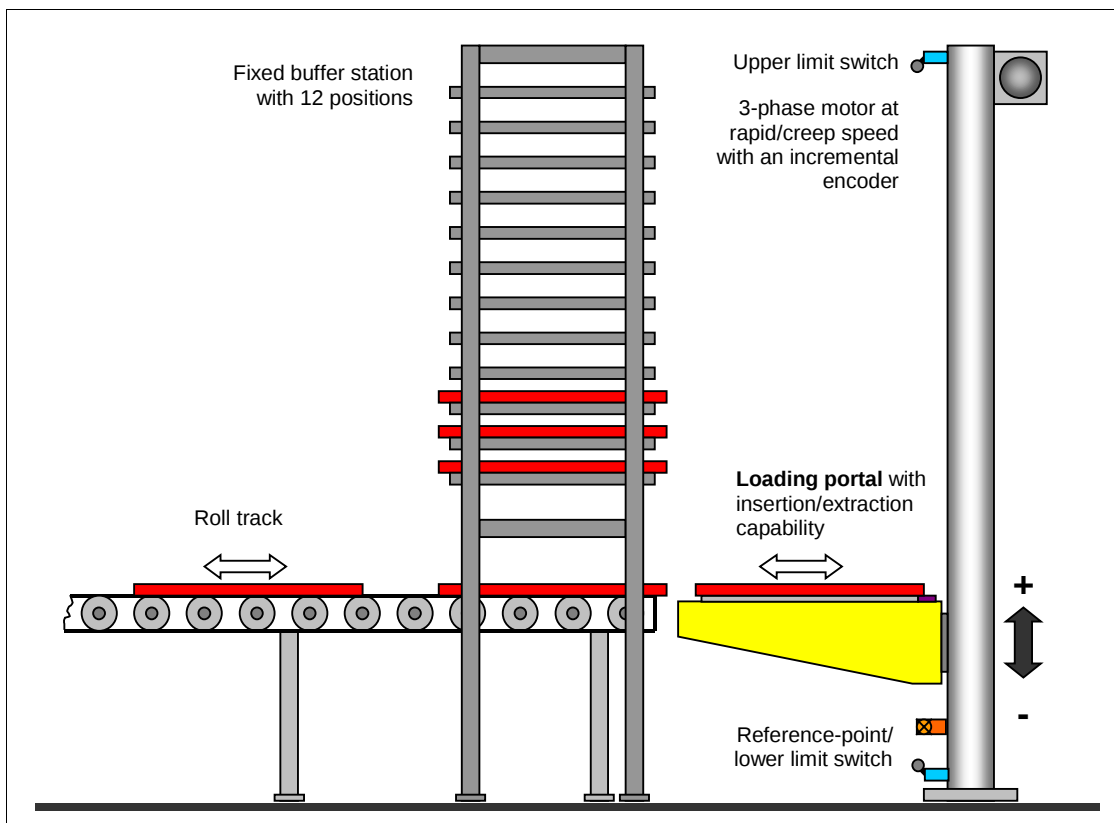
Target

This sample shows you an application for "**Positioning with Digital Output**" technology.

The control signals for automatic mode are generated in a sequential control. Manual functions are also possible, e. g. jogging and incremental approach.

The task

The tool plate values of a system are stored in an intermediate buffer station.



The loading portal fetches the prepared tool plates from the roll track and delivers them to the buffer station.

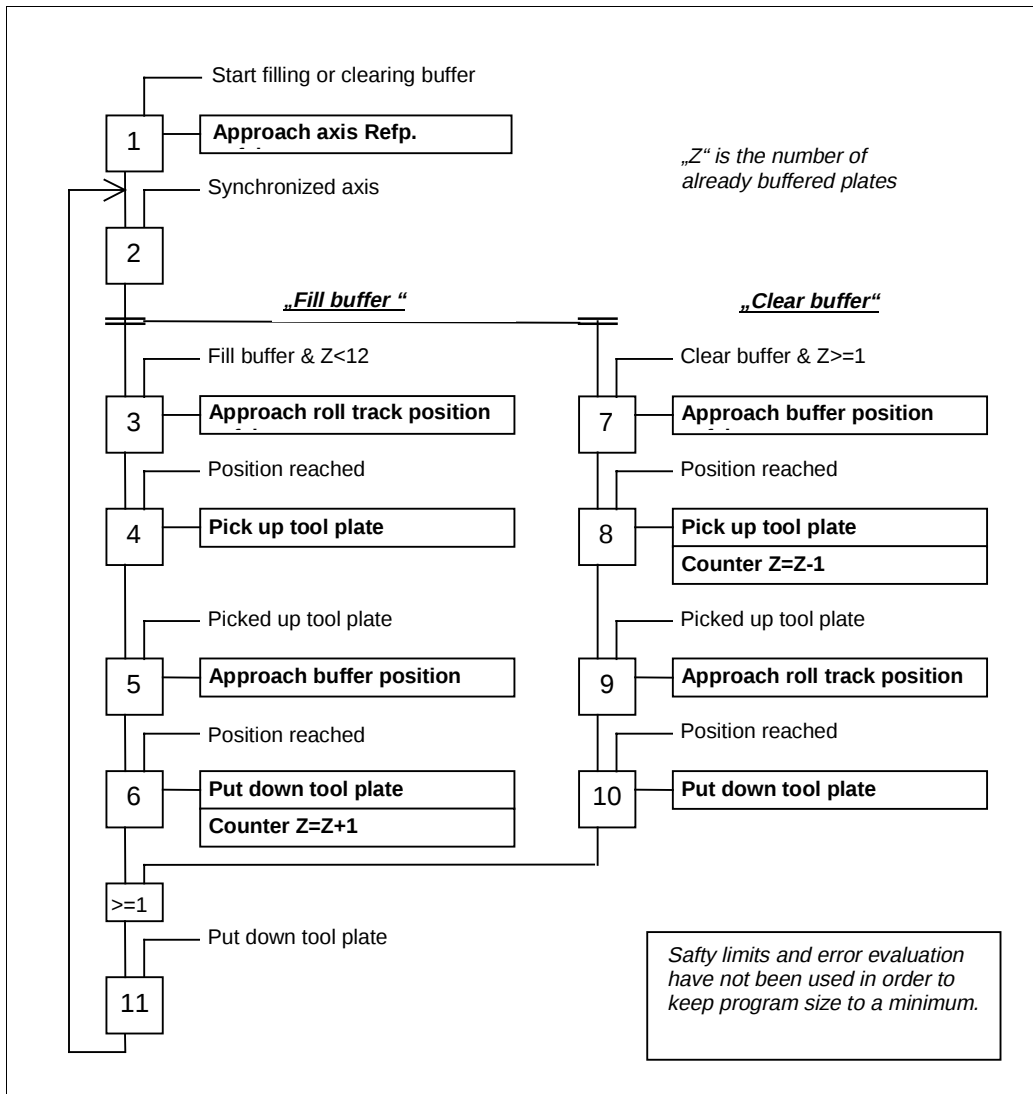
On request by the press, the tool plates are fetched from the buffer station and conveyed to the press along the roll track.

The loading portal is driven by a **3-phase motor** at rapid/creep speed. An incremental encoder supplies actual value feedback.

The reference-point switch installed at the bottom is used to synchronize the loading portal.

Flow chart

Start in operation per "Fill buffer" or "Clear buffer" instruction.



The loading portal is synchronized after it is initially switched on. Then, the program runs through the "Fill buffer" or "Clear buffer" branch. "Fill buffer" is initialized with an approach to the roll track position. Then, the gripper system pulls the tool plate into the loading portal. The buffer position can now be approached, the tool plates can be delivered. A counter counts the tool plates stored in the buffer station. The "Clear buffer" sequence is similar (see the diagram).

Conversion of the distance from mm to pulse (distance increments)

The incremental encoder supplies 2500 increment per revolution. It is installed downstream of the gears. The loading portal moves by 100 mm/revolution.

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

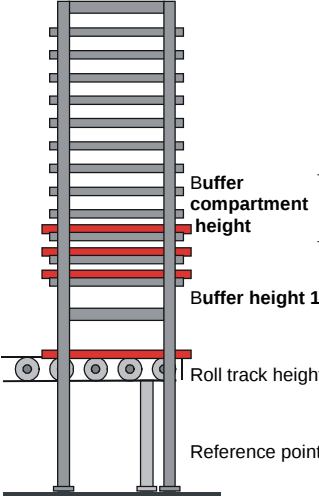
$$1 \text{ increment} = 4 \text{ pulses}$$

$$0.040 \text{ mm} / 4 = 0.010 \text{ mm}$$

Thus, one pulse is equivalent to a distance of 0.010 mm.

Target positions

The following position set points are necessary:



Target positions	Conversion in pulses (distance increments)
Buffer height 12	Buffer height 1 + (12-1) * Buffer comp. height 1100 mm + (12-1) * 100mm = 2200 mm 2200 mm : 0.01 mm/pul. = 220 000 Pulses
Buffer height 2..11	See calculation for buffer height 12
Buffer height 1	1100 mm : 0.01 mm/pul. = 110 000 Pulses
Roll track height	600 mm : 0.01 mm/pul. = 60 000 Pulses
Reference point	300 mm : 0.01 mm/pul. = 30 000 Pulses
Buffer comp. height	100 mm : 0.01 mm/pul. = 10 000 Pulses

Parameter assignment with STEP 7

Configure the CPU with STEP 7 as follows:

Parameters	Description
Type of technology	Positioning With Digital Outputs
Type of control	1
Target range	100 pulses (1 mm / 0.01 mm/Imp. = 100 pulses)
Monitoring time	2000 ms
Axis type	Linear axis
Software limit switch Start	25000 pulses Operating range limit Start (250 mm / 0.01 mm/Imp. = 25000 pulses)

Parameters	Description
Software limit switch End	250000 pulses Operating range limit End (2250 mm / 0.01 mm/Imp. = 225 000 pulses)
Reference point coordinate	30000 pulses
Reference point to reference point switch	In plus direction
Increments per encoder revolution	2500
Counting direction	Standard (Track A transition before B = positive actual value)

Switchover/switch-off difference minus:

The "Changeover difference plus" defines the changeover position as of which the drive is toggled from rapid to creep speed. It must be ensured that the drive reaches creep speed.

In our sample we have determined a distance of 60 mm.

60 mm / 0.01 mm/Imp. = **6000** pulses

The **Switch-off difference** defines the distance from the switch-off position to the target. The drive is de-energized on reaching the cut-off position.

In order to reach the target position with maximum precision, proceed as follows:

1. In your user program, specify a switch-off difference $\geq 1/2$ target range.
2. Position the loader once in "Absolute incremental approach" mode.
3. Determine the difference between the actually reached target position and the target position setpoint.
4. Add this difference to the switch-off difference.

We have determined a tracking of plus/minus 250 pulses.

In your user program, enter:

- Switchover difference + = **6000** pulses
- Switchover difference - = **6000** pulses
- Switch-off difference + = **250** pulses
- Switch-off difference - = **250** pulses

Operation of "Buffer station"

In the variables table VAT_STORAGE, specify the desired position values for incremental approaches. Set and enable the parameter S_DRV_EN.

Parameters	Setting	Meaning
PosCon	xx xxx xxx	Setpoint Position value roll track height
PosS1	xx xxx xxx	Setpoint Position value buffer 1
PosSH	xx xxx xxx	Setpoint Position value buffer compartment height (relative dimension)
S_DRV_EN	1	Control signal: Drive enable

You can trigger auto mode using one of the following signals:

Parameters	Setting	Meaning
S_Fill	1	Start auto mode for filling the buffer
S_Empty	1	Start "Clear buffer" auto mode

You can set and enable the following signals, provided you have not started auto mode. (S_Fill & S_Empty = 0 -> Manual mode)

Parameters	Setting	Meaning
S_REF	1	Start reference point approach
S_DIR_PF	1	Start jogging forward at rapid speed
S_DIR_MF	1	Start jogging backward at rapid speed
S_DIR_PS	1	Start jogging forward at slow speed
S_DIR_MS	1	Start jogging backward at slow speed
S_PosCon	1	Approach roll track Start Pos (absolute)
S_PosS1	1	Approach buffer 1 Start Pos. (absolute)
S_PosS_P	1	Approach buffer Start Pos. +1 (absolute)
S_PosS_M	1	Approach buffer Start Pos. -1 (absolute)

Feedback:

Parameters	Value	Meaning
ST_ENBLD	1	Start enable
MODE_OUT	x	Enabled/configured operating mode
WORKING	1	Running
POS_RCD	1	Position reached
SYNC	1	Axis is synchronized
ACT_POS	xx xxx xxx	Actual position value
ERROR	1	Run start/resume error
STATUS	xxxx	Error ID
ERR	xxxx	Hardware error

2 Sample Programs for "Counting, Frequency Measurement and Pulse Width Modulation" Technology

2.1 Introduction

Overview

In this chapter you will find sample applications of counting technology.

The CD included with your documentation also contains these samples. You can also download these samples from the Internet if you do not have this CD_ROM available. The project consists of several commented S7 programs of diverse complexity and aiming points.

The Readme.wri on the CD describes how to install the samples. After installation the examples are stored in the catalog

...\STEP7\EXAMPLES\ZEn26_02_TF____31xC_Cnt

Requirements

- An installed and wired S7 station consisting of the power supply module and a CPU 31C.
- STEP 7 (>= V5.1 + Service Pack 2) is installed correctly on your PG.
- The PG is connected to the CPU.

How to prepare the examples

1. Using SIMATIC Manager, open the sample project **ZEn26_02_TF____31xC_Cnt** in your ...\\STEP7\\EXAMPLES folder and copy it under a suitable filename to your project directory (**File > Save as**). Select **View > Detail** to display the full information.
2. Insert a station in your project according to your hardware structure.
3. Select a sample program and copy the S7 program to this station.
4. Configure all your hardware in **HW Config**. Then select the TF submodule **Count**.

5. Configure this submodule according to the sample parameters.
6. Save your hardware configuration and download it to the CPU.
7. Download the block library to the CPU.

The sample code

The samples are written in STL/FDB. You can view them directly in your LAD/STL/LAD Editor.

In this Editor, select "Symbolic layout", "Symbol selection" and "Comment" via **View > View with**. You can also display "Symbol information", provided your desktop offers sufficient space.

Using the sample

You can view and edit values, using the VATs included in the sample programs.

Subsequent use of a sample

Subsequently, you can use the sample code directly for a user program.

However, the sample code is not optimized or intended to cover all options. Error evaluation is not extensively programmed in the sample programs in order to restrict the size of the programs.

2.2 Counting

You will find two program samples of counting technology:

- **Count 1 First steps**
In this sample we call and assign parameters to the system function block (SFB) COUNT. You especially learn how to configure the functions of SFB ANALOG.
- **Count 2 Filling station with automatic recognition of the container size**
The "Count 2 Filling station" sample represents a typical application of counting technology, consisting of a link logic for automatic sequential control and calling the SFB.

2.2.1 Count 1 First steps

Target

In this sample you commission the "Counting" technology .

You can test the functions shown below after having enabled the respective SFB parameters:

- Start/Stop of the counter
- Reading the status bit
- Reading the actual counter value
- Writing the internal count registers
- Enabling/controlling of output DO
- Latch function

Block structure

The sample "Count 1 First steps" includes the following blocks:

Block	Name (in the toolbar)	Description
OB 1	CYCLE_EXC	Cyclic Program
FB 11	GETST_C	Sample 1: First steps with COUNT
DB 11	DI_GETST_C	Instance-DB to GETST_C
SFB 47	COUNT	System function block COUNT
DB 16	DI_COUNT	Instance-DB to SFB COUNT

The block GETST_C calls and supplies parameters to the system function block COUNT.

You can edit the parameters of SFB COUNT in the prepared variable table VAT_GETST_C.

Connections:

You must wire the following CPU 31xC input signals:

Connection CPU 312C: X1	Name/ Address	Sample functions
2	DI+0.0	Pulse input
3	DI+0.1	Direction
4	DI+0.2	Hardware gate
8	DI+0.6	Latch input
12	2 M	Reference potential of the supply voltage
13	1 L+	Supply voltage 24 V DC
16	DO+0.2	Simulation: connect pulse input -> to DI+0.0
17	DO+0.3	Simulation: connect direction bit -> to DI+0.1
18	DO+0.4	Simulation: connect the hardware gate -> to DI+0.2
19	DO+0.5	Simulation: connect the latch input -> to DI+0.6
20	1M	Reference potential of the supply voltage

Connection CPU 313C-2 DP/PtP: X1; CPU 313C, 314C-2 DP/PtP: X2	Name/ Address	Sample functions
1	1 L+	Supply voltage 24 V DC
2	DI+0.0	Pulse input
3	DI+0.1	Direction
4	DI+0.2	Hardware gate
16	DI+1.4	Latch input
20	1M	Reference potential of the supply voltage
21	2 L+	Supply voltage 24 V DC
24	DO+0.2	Simulation: connect pulse input -> to DI+0.0
25	DO+0.3	Simulation: connect direction bit -> to DI+0.1
26	DO+0.4	Simulation: connect the hardware gate -> to DI+0.2
27	DO+0.5	Simulation: connect the latch input -> to DI+1.4
30	2 M	Reference potential of the supply voltage

Parameters

Configure the CPU 31xC in HW Config as follows:

Parameters	Input
Channel	0
Operating mode	Infinite count
Gate function	Abort count
Signal evaluation	Pulse/Direction
Input function	Use the hardware gate
Reaction of output DO	The output switches when Counter value $\geq$ Comparison value

Operation of "Start/Stop counter"

"Starting the counter via Hardware Gate"

1. Generate a count pulse for the "Count input" (Simulation: T_PULSE)
In the VAT, select the count pulse source via variable S_IMP_H:
 - S_IMP_H = 0: the pulse source is a programmable software clock generator. Its frequency can be adjusted in the variable T_PULSE (pulse duration). A digital output supplies the clock signal to the pulse input.
 - S_IMP_H = 1: the count pulses are determined in the VAT per manual set/reset operation of the variable S_IMP_T.
2. Set and enable the following SFB input parameters:
SW_GATE = 1: software gate for Counter Start/Stop
3. Set the input "Hardware gate" (Simulation: S_HWT = 1)

Parameters	Setting	Meaning
SW_GATE	1	Open the software gate
T_PULSE	S5T#500 ms	Pulse duration (e. g. 500 ms -> 1 Hz)
S_IMP_H	0	Simulation: Pulses 0 = Auto, 1 = Manual
S_IMP_T	0-1	Simulation: Apply pulses manually
S_HWT	1	Simulation: Hardware gate

"Starting the counter via software gate SW-GATE"

1. Generate a count pulse for the "Count input" (Simulation: T_PULSE)
2. Set the input "Hardware gate" (Simulation: S_HWT = 1)
3. Set and enable the following SFB input parameters:
SW_GATE = 1: software gate for Counter Start/Stop

Parameters	Setting	Meaning
T_PULSE	S5T#500 ms	Pulse duration (e. g. 500 ms -> 1 Hz)
S_IMP_H	0	Simulation: Pulses 0 = Auto, 1 = Manual
S_IMP_T	0-1	Simulation: Apply pulses manually
S_HWT	1	Simulation: Hardware gate
SW_GATE	1	Open the software gate

"Stopping the counter"

You can stop the counter by clearing the "Hardware gate" input signal or resetting the SFB parameter "Software gate" (SW_GATE = 0).

Output parameters you can monitor:

Parameters	Value	Meaning
STS_GATE	1	Status: gate (logical link hardware gate AND software gate)
STS_STRT	1	Status: Start input (hardware gate)
STS_C_UP	1	Status: up count
STS_C_DN	0	Status: down count
COUNTVAL	xxxx	Actual counter value

Operation of "Load counter"

For example, if you want to load a value of 20 into the counter, set and enable the following SFB parameters:

Parameters	Setting	Meaning
JOB_ID	1	Job number "Write counter directly"
JOB_VAL	20	Enter the count value to be transferred
JOB_REQ	1	Job request (positive edge evaluation)

Output parameters you can monitor:

Parameters	Value	Meaning
COUNTVAL	20	Actual counter value
JOB_DONE	1	Status: New job can be started
JOB_ERR	0	Status: job error
JOB_STAT	0	Status: job error ID

Operation of "Load comparison value"

For example, if you want to set a comparison value of 100, set and enable the following SFB parameters:

Parameters	Setting	Meaning
JOB_ID	4	Job number "Write comparison value"
JOB_VAL	100	Enter the comparison value to be transferred
JOB_REQ	1	Job request (positive edge evaluation)

Output parameters you can monitor:

Parameters	Value	Meaning
JOB_DONE	1	Status: New job can be started
JOB_ERR	0	Status: job error
JOB_STAT	0	Status: job error ID

Operation of "Enable and control output DO"

To switch on the output, you must set the SFB parameter CTRL_DO.

Parameters	Setting	Meaning
CTRL_DO	1	Output enable

There are two options:

1. Controlling output DO via comparator

For example, you can monitor the status of output DO if you set the comparison value to 100 and then restart the counter.

Parameters	Value	Meaning
COUNTVAL	xxxx	Actual counter value
STS_DO	x	Status: Output Since the reaction of output DO is set to "Counter value >= Comparison value", it is switched on when the actual counter value >= 100.

2. Controlling output DO in synchronism to the comparator

The SFB parameter **SET_DO** gives you the option of simulating the effect of the comparison function in your control program:

- A **positive edge** at **SET_DO** sets the output.
- A **negative edge** at **SET_DO** resets the output.

Make sure the comparators remain active, enabling them to set or reset the output at every count pulse.

Operation of the "LATCH function"

You can save the actual counter value with a positive edge at the LATCH input.

Steps required:

1. Start the counter by closing the hardware and software gates and generate counting pulses for the "Count input" (Simulation: PULSE_SEL).

Parameters	Setting	Meaning
SW_GATE	1	Close the software gate
HWT	1	Simulation: Hardware gate
T_PULSE	S5T#500 ms	Pulse width (e. g. 500 ms -> 1 Hz)

2. Generate an edge at the latch input (Simulation: LATCH).

Parameters	Setting	Meaning
S_LATCH	1	Simulation: Latch input

Output parameters you can monitor:

Parameters	Value	Meaning
STS_LATCH	1	Latch input status
LATCHVAL	xxxx	fetches latch value
COUNTVAL	xxxx	Actual counter value

2.2.2 Count 2 Filling station

Target

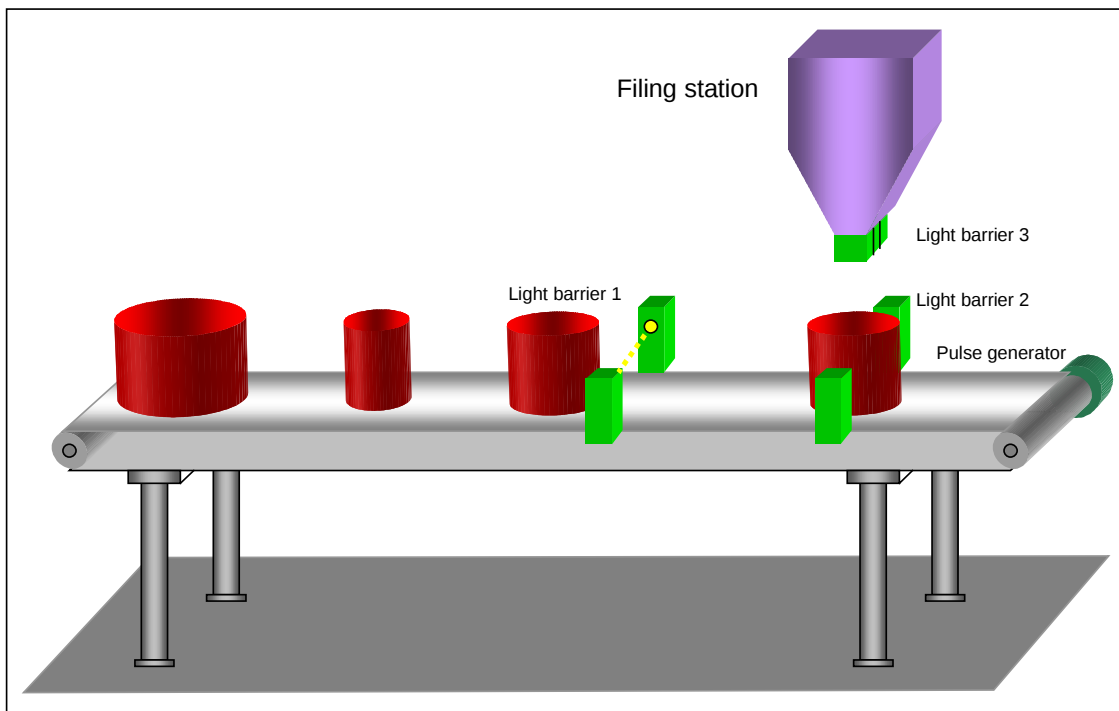
This sample application demonstrates **Counting** technology.

The task

The sample is an application for a filling station with automatic container size detection and positioning. The system fills containers of different size.

A conveyor belt feeds containers of a different size. Light barrier 1 detects the containers. The container diameter is determined by pulse count.

The container diameter can be used to calculate the volume and to adjust the station's filling quantity.



Sequence

- The conveyor belt starts when no container is present in the filling station or when filling has finished.
- **Counter 0** determines the container diameter; **light barrier 1** controls the hardware gate of the counter.
The counter value is saved when the container leaves **light barrier 1**.
Depending on the container diameter, the setpoint is calculated and set (counter 1) for the filling station with Job 4 (load comparison value).
- Filling starts after the container is positioned via **light barrier 2**.
Counter 1 counts the filled parts and ends dosing.

Conversion of the filling quantity

The volume is calculated of $d^2 * \pi/4 * h$. A container height of 100 mm is assumed.
One container is filled per 10000 mm³.

Example: Container diameter is 150 mm.

$$\begin{aligned}
 \text{Filling quantity} &= d^2 * \pi/4 * h/10\,000 \\
 &= 150\text{ mm} * 150\text{ mm} * (\pi/4) * 100\text{ mm}/10000\text{ parts/mm}^3 \\
 &= \underline{176\text{ parts}}
 \end{aligned}$$

Block structure

Sample 2 "Filling station" includes the following blocks:

Block	Name (in the toolbar)	Description
OB1	CYCLE_EXC	Cyclic Program
FB12	FILL	Sample 2: Filling station
DB12	DI_FILL	Instance-DB to FILL
SFB47	COUNT	System function block COUNT
DB16	DI_COUNT	Instance-DB to SFB COUNT (container diameter)
DB17	DI_COUNT2	Instance-DB to SFB COUNT (filling station)

Connections:

The connections for the sample are listed below:

Connection CPU 312C: X1	Name/ Address	Sample functions
		Connection of counter 0 "Container diameter"
2	DI+0.0	<u>Count pulses</u> Pulse generator belt conveyor
4	DI+0.2	<u>Hardware gate</u> Light barrier 1: determine container diameter
		Connection counter 1 "Filling"
5	DI+0.3	<u>Count pulses</u> Light barrier 3: count filled row
15	DO+0.1	<u>Output DO</u> Filling valve
		General
16	DO+0.2	Simulation: belt conveyor and filling pulses -> interconnect DO+0.2 with DI+0.0 and DI+0.3
18	DO+0.4	Simulation: HWT0 container diameter -> interconnect DO+0.4 and DI+0.2
12	2 M	Reference potential of the supply voltage
13	1 L+	Supply voltage 24 V DC
20	1M	Reference potential of the supply voltage

Connection CPU 313C-2 DP/PtP: X1; CPU 313C, 314C-2 DP/PtP: X2	Name/ Address	Sample functions
		Connection counter 0 "Container diameter"
2	DI+0.0	<u>Count pulses</u> Pulse generator belt conveyor
4	DI+0.2	<u>Hardware gate</u> Light barrier 1: determines the container diameter
		Connection counter 1 "Filling"
5	DI+0.3	<u>Count pulses</u> Light barrier 3: count the filled row
23	DO+0.1	<u>Output DO</u> Filling valve
		General
24	DO+0.2	Simulation: belt conveyor and filling pulses -> interconnect DO+0.2 with DI+0.0 and DI+0.3

Connection CPU 313C-2 DP/PtP: X1; CPU 313C, 314C-2 DP/PtP: X2	Name/ Address	Sample functions
26	DO+0.4	Simulation: HWT0 container diameter -> interconnect DO+0.4 and DI+0.2
1	1 L+	Supply voltage 24 V DC
20	1M	Reference potential of the supply voltage
21	2 L+	Supply voltage 24 V DC
30	2 M	Reference potential of the supply voltage

Parameter assignment

Two high-speed counters are used, configured by you in HW Config.

1. Configuration of counter 0 "Container diameter"

Parameters	Input
Channel	0
Operating mode	Infinite count
Gate function	Abort count
Signal evaluation	Pulse/Direction
Input function	Use hardware gate
Reaction of output DO	No comparison

2. Configuration of Counter 1 (filling):

Parameters	Input
Channel	1
Operating mode	Infinite count
Gate function	Abort count
Signal evaluation	Pulse/Direction
Input function	Hardware gate not used
Reaction of output DO	The output switches when Counter value >= Comparison value

Operation of the "Filling station"

The following VAT is prepared for testing the filling station:

Symbol	Symbol comment
S_START	Control signal: Auto Start
S_LS1	Control signal: Light barrier 1 (container diameter)
S_LS2	Control signal: Light barrier 2 (Container in filling position)
T_PULSE	Simulation: pulses
DIAM	Container DIAMeter
FILL_SETP	Setpoint fill
FILL_VAL	Actual value fill
CONV_ON	Conveyor ON
C_POS_RCD	Conveyor in position
FILL_STRT	Start filling
FILL_WORK	Filling in operation
FILL_DONE	Filling done

1. Generate a count pulse for the count input
(e. g. T_PULSE = S5T#50 ms -> 10 Hz).
2. Set parameter **S_START = 1** to start the sequence. The conveyor belt starts when no container is present in the filling station or when filling has finished.
3. You can now set input **S_LS1 = 1** (Light barrier 1: container diameter). Here you can see how counter 0 determines the container width.
4. A negative edge of the reset of **S_LS1 = 0** triggers calculation of the filling quantity and loads the comparison value into counter 1.
5. To stop the belt conveyor and start filling, set input **S_LS2 = 1** (Light barrier 2: Container in position).

When the calculated filling quantity is reached, the filling valve drops off and reports the end of the filling process. This again switches on the belt conveyor.

6. You can now reset input **S_LS2 = 0** again and repeat steps 2 to 5.

2.3 Frequency Measurement

We show you two samples of "Frequency measurement" technology:

1. Freq 1, First Steps, Counting, First Steps

In this sample we call and assign parameters to the system function block (SFB) FREQUENC. You especially learn how to configure the functions of SFB FREQUENC.

2. Freq 2, Rate of revolution:

Sample 2 of "Speed monitoring ..." represents a typical application of "Frequency measurement" technology, consisting of a link logic, a limit value calculation function and call of the SFBs.

2.3.1 Frequency 1, First Steps

Target

In this sample you commission "Frequency measurement" technology .

By calling the respectively configured SFB, you can also test the following operating modes:

- Start/stop of frequency measurement
- Reading the status bit
- Fetching the actual frequency and counter status
- Writing internal registers

Block structure

The sample "Freq 1, First steps" includes the following blocks:

Block	Name (in the toolbar)	Description
OB 1	CYCLE_EXC	Cyclic Program
FB 21	GETST_F	Sample 1: First steps with FREQUENCY
DB 21	DI_GETST_F	Instance-DB to GETST_F
SFB 48	FREQUENC	System function block FREQUENC
DB 26	DI_FREQ	Instance-DB to SFB FREQUENC

The block GETST_F calls and supplies parameters to SFB FREQUENC.

You can edit the parameters of SFB FREQUENC in VAT_GETST_F.

Connections:

You must wire the following CPU 31xC connections:

Connection CPU 312C: X1	Name/ Address	Sample functions
2	DI+0.0	Pulse input The positive edges of the measurement signal are evaluated.
3	DI+0.1	Direction
12	2 M	Reference potential of the supply voltage
13	1 L+	Supply voltage 24 V DC
14	DO+0.0	Output DO: This output is switched on if the frequency is out of range.
16	DO+0.2	Simulation: interconnect pulse output -> DI+0.0. Output for the pulses generated in the simulation program.
17	DO+0.3	Simulation: interconnect direction bit -> DI+0.1
20	1M	Reference potential of the supply voltage

Connection CPU 313C-2 DP/PtP: X1; CPU 313C, 314C-2 DP/PtP: X2	Name/ Address	Sample functions
1	1 L+	Supply voltage 24 V DC
2	DI+0.0	Pulse input The positive edges of the measurement signal are evaluated.
3	DI+0.1	Direction
20	1M	Reference potential of the supply voltage
21	2 L+	Supply voltage 24 V DC
22	DO+0.0	Output DO: This output is switched on when the frequency is out of range.
24	DO+0.2	Simulation: interconnect the pulse output -> to DI+0.0 Output for the pulses generated in the simulation program.
25	DO+0.3	Simulation: interconnect the direction bit -> DI+0.1
30	2 M	Reference potential of the supply voltage

Parameters

Configure the CPU 31xC in HW Config as follows:

Parameters	Input
Channel	0
Operating mode	Frequency measurement
Integration time	5000 ms
Lower limit	0 mHz
Upper limit	60,000,000 mHz (CPU 314) 30,000,000 mHz (CPU 313) 10,000,000 mHz (CPU 312)
Output measured value	Direct
Signal evaluation	Pulse/Direction
Input function	No hardware gate used
Output reaction	Out of range

Operation of "Start/Stop Frequency measurement"

"Triggering frequency measurement via SW-GATE"

1. Generate count pulses for the "Pulse input" (DI+0.0)

In the VAT, select the count pulse source in the variable S_IMP_H :

- S_IMP_H = 0: the pulse source is a programmable software clock generator. Its frequency can be adjusted in the variable T_PULSE (pulse duration). A digital output supplies the clock signal to the pulse input.
- S_IMP_H = 1: the count pulses are determined in the VAT per manual set/reset operation of the variable S_IMP_T.

2. Set and enable the following SFB input parameters:

Parameters	Setting	Meaning
SW_GATE	1	Software gate for triggering frequency measurement
T_PULSE	S5T#250 ms	Pulse width (e. g. 250 ms -> 2 Hz)
S_IMP_H	0	Simulation: Pulses 0 = Auto, 1 = Manual
S_IMP_T	0-1	Simulation: apply pulses manually

"Stopping frequency measurement"

Reset the SFB parameter "Close software gate" (SW_GATE = 0) to stop frequency measurement.

Output parameters you can monitor:

Parameters	Value	Meaning
STS_GATE	1	Status: Software gate
STS_C_UP	1	Status: up count
STS_C_DN	0	Status: down count
MEAS_VAL	xxxx	Actual frequency
COUNTVAL	xxxx	Actual counter value
STS_UFLW	x	Status of Underflow
STS_OFLW	x	Status of Overflow

Operation of "Load lower limit"

For example, to set a lower limit of 9000 mHz, set and enable the following SFB parameters:

Parameters	Setting	Meaning
JOB_ID	1	Job number "Write lower limit"
JOB_VAL	9000	Write lower limit [mHz]
JOB_REQ	1	Job request (positive edge evaluation)

Output parameters you can monitor:

Parameters	Value	Meaning
JOB_DONE	1	Status: new job can be started
JOB_ERR	0	Status: job error
JOB_STAT	0	Status: job error ID

Operation of "Load upper limit"

For example, to set an upper limit of 11000 mHz, set and enable the following SFB parameters:

Parameters	Setting	Meaning
JOB_ID	2	Job ID "Write upper limit"
JOB_VAL	11000	Write upper limit [mHz]
JOB_REQ	1	Job request (positive edge evaluation)

Output parameters you can monitor:

Parameters	Value	Meaning
JOB_DONE	1	Status: New job can be started
JOB_ERR	0	Status: job error
JOB_STAT	0	Status: job error ID

Operation of "Load Integration Time"

For example, if you want to specify an integration time of 100 ms, set and enable the following SFB parameters:

Parameters	Setting	Meaning
JOB_ID	4	Job ID "Write Integration Time"
JOB_VAL	100	Write Integration Time value [ms]
JOB_REQ	1	Job request (positive edge evaluation)

Output parameters you can monitor:

Parameters	Value	Meaning
JOB_DONE	1	Status: New job can be started
JOB_ERR	0	Status: job error
JOB_STAT	0	Status: job error ID

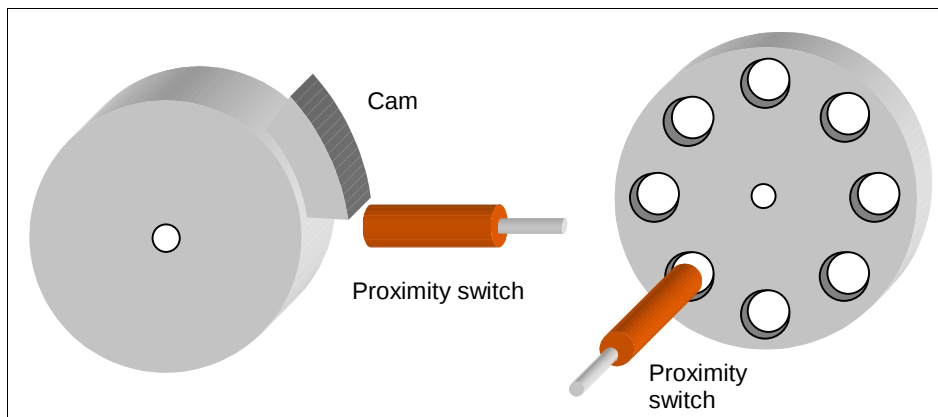
2.3.2 Frequency 2, speed monitoring with variable setpoint

Target

This sample application demonstrates **Frequency Measurement** technology.

The task

The rate of revolution of a drive shaft is to be monitored. The rate of revolution as well as consistency of a variable speed range is monitored via proximity switch. An error message is generated if the speed exceeds the permitted value.



Pulses, for example, can be generated via punched disk or control cam.

Sequence

The **Start** signal triggers frequency measurement. You can specify the speed (**SPEED**), tolerance (**TOLERANCE**) and the pulses/rev (**PULSE**).

The incoming pulses form the actual speed value (**SPEED_VAL**) and are compared with the calculated limit values. Bit **STATUS_U** is set if the value drops below the lower limit, bit **STATUS_O** is set if the value exceeds the upper limit.

Calculation of the upper and lower limit

The following equation can be used to calculate upper and lower limits.

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

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

Example: Speed setpoint = 600 1/min
Tolerance = 5 %
pulses/rev = 8 (e. g. punched disk with 8 holes)

Comparison value	RPM	Frequency	Comparison value upper/lower limit
Lower limit	600 1/min - 5% = 570 1/min	570 1/min / 60 = 9.5 Hz	9.5 Hz x 8 (pulses/rev) = 76 Hz Lower limit (in mHz): 76000 mHz
Upper limit	600 1/min + 5% = 630 1/min	630 1/min / 60 = 10.5 Hz	10.5 Hz x 8 (pulses/rev) = 84 Hz Lower limit (in mHz): 84000 mHz

Block structure

Sample 2 "Freq 2 Speed" includes the following blocks:

Block	Name (in the toolbar)	Description
OB 1	CYCLE_EXC	Cyclic Program
OB 100	COMPLETE RESTART	Restart:
FB 22	SPEED	Sample 2: speed monitoring
DB 22	DI_SPEED	Instance-DB to SPEED
SFB 48	FREQUENC	SFB FREQUENC
DB 26	DI_FREQ	Instance-DB to SFB FREQUENC

Connections:

The connections for the sample are listed below:

Connection CPU 312C: X1	Name/ Address	Sample functions
2	DI+0.0	Pulse input The positive edges of the measurement signal are evaluated.
12	2 M	Reference potential of the supply voltage
13	1 L+	Supply voltage 24 V DC
14	DO+0.0	Output DO This output is switched on if the frequency is out of range.
16	DO+0.2	Simulation: Connect pulse output -> to DI+0.0 Output for the pulses generated in the simulation program.
20	1M	Reference potential of the supply voltage

Connection CPU 313C-2 DP/PtP: X1; CPU 313C, 314C-2 DP/PtP: X2	Name/ Address	Sample functions
1	1 L+	Supply voltage 24 V DC
2	DI+0.0	Pulse input Measuring positive edges of the measurement signal.
20	1M	Reference potential of the supply voltage
21	2 L+	Supply voltage 24 V DC
22	DO+0.0	Output DO This output is switched on if the frequency is out of range.
24	DO+0.2	Simulation: Connect pulse output -> to DI +0.0 Output for the pulses generated in the simulation program.
30	2 M	Reference potential of the supply voltage

Parameter assignment

Configure the CPU 31xC in HW Config as follows:

Parameters	Input
Channel	0
Operating mode	Frequency measurement
Integration time	1000 ms
Lower limit	0 mHz
Upper limit	60,000,000 mHz (CPU 314) 30,000,000 mHz (CPU 313) 10,000,000 mHz (CPU 312)
Output measured value	Direct
Signal evaluation	Pulse/Direction
Input function	No hardware gate used
Output reaction	Out of range

Operation of speed monitoring

The following VAT was prepared:

Symbol	Symbol comment
T_PULSE	Simulation: period for frequency generation
START	Control signal: frequency measurement ON
RES_STS	Control signal: reset status
SPEED	Setpoint: speed setpoint in RPM.
TOLERANCE	Setpoint: tolerance in %
PULSE	Setpoint: pulses/rev
SPEED_VAL	Actual value: actual rate of revolution
STATUS_U	Status: underflow
STATUS_O	Status: overflow
MEAS_VAL	SFB parameters: actual measurement value
COUNTVAL	SFB parameters: fetched counter value
UFLW_VAL	Calculated actual underflow value
OFLW_VAL	Calculated actual overflow value
UFLW_LOAD	Load value of calculated underflow
OFLW_LOAD	Load value of calculated overflow

Specify the following values:

Parameters	Setting	Meaning
SPEED	300	Setpoint: speed setpoint in RPM.
TOLERANCE	5	Setpoint: tolerance in %
PULSE	1	Setpoint: pulses/rev

The program calculates and transfers the following limit values:

- Lower limit: UFLW_VAL = 4750 mHz
- Upper limit: OFLW_VAL = 5250 mHz

Now, generate 5 Hz pulses (T_Pulse = 100 ms) at the measuring input.

You can also choose to generate the pulses in your simulation program and interconnect output DO+0.2 and measuring input DI+0.0. Set parameter **START = 1** to trigger speed monitoring. You can monitor actual values, calculated limit values and the status bits.

Now, modify the speed setpoint or the frequency at the measuring input.

The status bits **STATUS_U** (Underflow) or **STATUS_O** (Overflow) generate an error message if the actual values are out of range. Output DO also generates an error message if the speed is out of range. You can reset the status bits with **RES_STS = 1**.

2.4 Pulse width modulation

Here, we show you two program samples of "Pulse Width Modulation" technology:

1. PWM 1, First steps:
In this sample we call and assign parameters to the system function block (SFB) PULSE. You especially learn how to configure the SFB PULSE.
2. PWM 2, Heating:
Sample 2 "Heating liquids" represents an application of "PWM" technology, consisting of setpoint calculation and call of the SFB.

2.4.1 PWM 1, First steps

Target

In this sample you commission "PWM" technology .

You learn especially how to configure the system function block (SFB) DIGITAL.

By calling the respectively configured SFB, you can also test the following operating modes:

- Start/Stop pulse width modulation
- Enabling/controlling of output DO
- Reading the status bit
- Writing the output value
- Writing internal registers

Block structure

The sample "PWM 1, First steps" includes the following blocks:

Block	Name (in the toolbar)	Description
OB 1	CYCLE_EXC	Organization Block
FC 31	GETST_P	Sample 1: First steps with PWM
SFB 49	PULSE	SFB PULSE
DB 36	DI_PULSE	Instance-DB to SFB PULSE

The block GETST_P calls and supplies parameters to SFB PULSE. You can edit the SFB PULSE parameters in VAT_GETST_P.

Connections:

The connections for the sample are listed below:

Connection CPU 312C: X1	I/O	Sample functions
12	2 M	Reference potential of the supply voltage
13	1 L+	Supply voltage 24 V DC
14	DO+0.0	Output DO: 24-V control signal (e. g. for contactor or TRIAC)
20	1M	Reference potential of the supply voltage

Connection CPU 313C-2 DP/PtP: X1; CPU 313C, 314C-2 DP/PtP: X2	I/O	Sample functions
1	1 L+	Supply voltage 24 V DC
20	1M	Reference potential of the supply voltage
21	2 L+	Supply voltage 24 V DC
22	DO+0.0	Output DO: 24-V control signal (e. g. for contactor or TRIAC)
30	2 M	Reference potential of the supply voltage

Parameters:

Configure the CPU 31xC in HW Config as follows:

Parameters	Input
Channel	0
Operating mode	Pulse width modulation
Output format	Per mil
A Timebase	Resolution of 1 ms. Permits a range of time values from 1 ms to 65,535 s
On delay	0 ms Delay time between start of the output sequence and pulse sequence output.
Period	20000 ms The period should always represent a multiple of the response time of the actuator you have connected to digital output DO.

Parameters	Input
Minimum pulse width	500 ms Here, you should enter the response time of the actuator connected to the digital output DO.
Input function	Do not use hardware gate

Operation of "Start/stop pulse width modulation"

"Start output sequence"

Set and enable the following SFB input parameters:

Parameters	Setting	Meaning
OUTP_VAL	250	Actual output value (e. g. 250 ‰)
MAN_DO	0	Output enable (for manual mode)
SW_EN	1	"Start output sequence"

"Stop output sequence"

You can stop the output sequence with a reset of SFB parameter "Software Enable" (SW_EN = 0).

Output parameters you can monitor:

Parameters	Value	Meaning
STS_EN	1	Status: Enable
STS_STRT	1	Status: Start input (HW gate)
STS_DO	0/1	Status: Output The set period of 20 s forces an output pulse/pause ratio of 5 s (250 ‰ of 20 s)/15 s.

Operation of "Manual control of output DO "

Set and enable the following SFB parameters:

Parameters	Setting	Meaning
MAN_DO	1	Output enable (for manual mode)
SET_DO	1	Output control

Output parameters you can monitor:

Parameters	Value	Meaning
STS_DO	1	Status: Output The output is switched on continuously if SET_DO = 1.

Operation of "Load Period"

For example, if you want to set a period of 2000 ms, set and enable the following SFB parameters:

Parameters	Setting	Meaning
JOB_ID	1	Job ID "Write Period"
JOB_VAL	2000	Write period value [ms]
JOB_REQ	1	Job request (positive edge evaluation)

Output parameters you can monitor:

Parameters	Value	Meaning
JOB_DONE	1	Status: New job can be started
JOB_ERR	0	Status: job error
JOB_STAT	0	Status: job error ID

Operation of "Load On Delay"

For example, if you want to set an on delay of 500 ms, set and enable the following SFB parameters:

Parameters	Setting	Meaning
JOB_ID	2	Job ID "Write on delay"
JOB_VAL	500	Write on delay value [ms]
JOB_REQ	1	Job request (positive edge evaluation)

Output parameters you can monitor:

Parameters	Value	Meaning
JOB_DONE	1	Status: New job can be started
JOB_ERR	0	Status: job error
JOB_STAT	0	Status: job error ID

Operation of "Load Minimum Pulse Width"

For example, if you want to set a minimum pulse width of 100 ms, set and enable the following SFB parameters:

Parameters	Setting	Meaning
JOB_ID	4	Job ID "Write Minimum Pulse Width"
JOB_VAL	100	Write minimum pulse width value [ms]
JOB_REQ	1	Job request (positive edge evaluation)

Output parameters you can monitor:

Parameters	Value	Meaning
JOB_DONE	1	Status: New job can be started
JOB_ERR	0	Status: job error
JOB_STAT	0	Status: job error ID

2.4.2 PWM 2 Heating liquids

Target

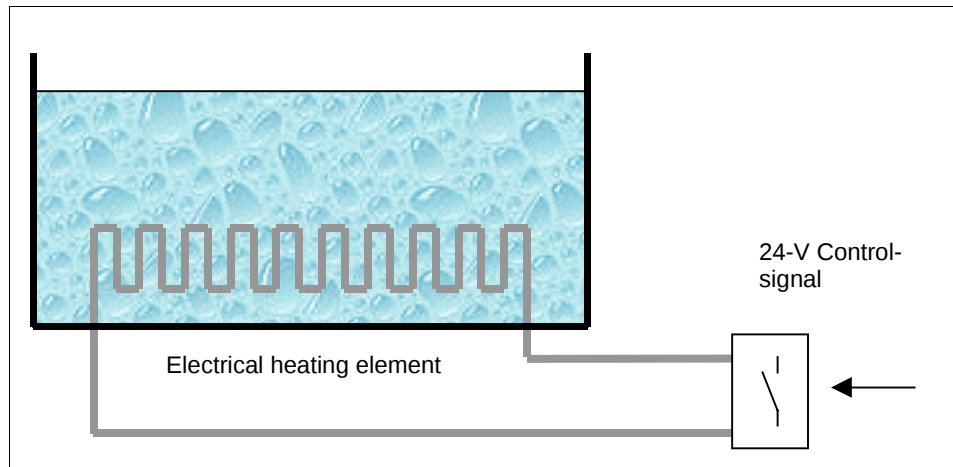
This practical sample application demonstrates **PWM** technology.

The task

An electrical heating element heats a liquid. The energy required by the heating element is applied with the help of a switching element (e. g. a contactor).

The CPU 31xC generates a 24-V control signal at the digital output for the switching element. The heating element temperature depends on the length of the on/off period of the 24-V control signal.

Longer on times of the 24-V control signal increase the heating period and, accordingly, liquid temperature.

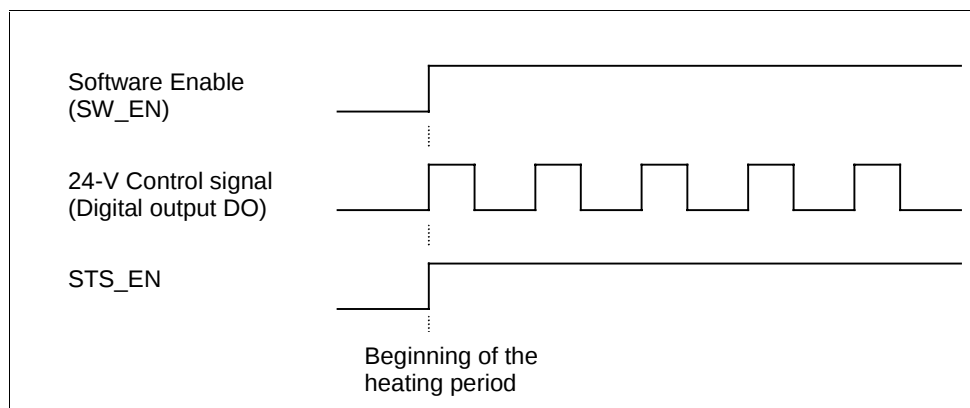


Requirements

- The heating element acquires only two switching states: ON or OFF.
- The effective heating current is proportional to the on/off period of the 24-V control signal.
- The minimum pulse/pause width must be higher than the response time of the switching and heating elements.

Sequence

Start heating: Set Software Enable (SW_EN) in your control program to start the heating cycle.



Block structure

The sample "PWM 2, Heating" includes the following blocks:

Block	Name (in the toolbar)	Description
OB 1	CYCLE_EXC	Organization Block
FB 32	HEATING	Sample 2: Heating liquids
DB 32	DI_GETST_P	Instance-DB to HEATING
SFB 49	PULSE	System function block PULSE
DB 36	DI_PULSE	Instance-DB to SFB PULSE

Connections:

The connections for the sample are listed below:

Connection CPU 312C: X1	I/O	Sample functions
12	2 M	Reference potential of the supply voltage
13	1 L+	Supply voltage 24 V DC
14	DO+0.0	Output DO: 24-V control signal (e. g. for contactor or TRIAC)
20	1M	Reference potential of the supply voltage

Connection CPU 313C-2 DP/PtP: X1; CPU 313C, 314C-2 DP/PtP: X2	I/O	Sample functions
1	1 L+	Supply voltage 24 V DC
20	1M	Reference potential of the supply voltage
21	2 L+	Supply voltage 24 V DC
22	DO+0.0	Output DO: 24-V control signal (e. g. for contactor or TRIAC)
30	2 M	Reference potential of the supply voltage

Parameters

The following parameters must be declared for the liquid heating process:

Parameters	Input
Channel	0
Operating mode	Pulse width modulation
Output format	Per mil
A Timebase	The resolution of all specified times is set to 1 ms
On delay	0 ms The 24-V control signal is output instantaneously when SW_EN = 1
Period	10000 In the selected timebase equivalent to 10 s
Input function	No hardware gate used
Minimum pulse width	500 ms Minimum pulse width In the selected timebase equivalent to 500 ms; this also applies to minimum pulse pause.

Operation of "Heating liquids"

You can use the following VAT (VAT_HEATING).

Symbol	Symbol comment
START	Control signal: Heating ON
POWER	Setpoint Heating power in %
STS_DO	Status output DO

Specify the following values:

Parameters	Setting	Meaning
POWER	40	Setpoint Heating power in %

Set parameter **START = 1** to trigger heating.

You can monitor the status of output DO. At a period of 10 s the output on time is 4 s and the off time is 6 s.

Now, modify the heating power setpoint (POWER) and monitor output DO.

3 Sample Programs for "Point-to-Point Connection" Technology

3.1 Introduction

Overview

In this chapter you can find application samples of PtP connection technology.

If the CD included with your documentation does not also contain these samples, you can download them from the Internet. The project consists of several commented S7 programs of diverse complexity and aiming points.

The Readme.wri on the CD describes how to install the samples. After installation the samples are filed in the catalog

...\\STEP7\\EXAMPLES\\ZEn26_01_TF____31xC_PtP.

Requirements

- You have installed and wired an S7 station, consisting of a power supply module and a CPU 313C-2 PtP or CPU 314C-2 PtP.
- You have a communication partner (e. g. CP 341) or you plug in a "Short-circuit plug", that is, the send line is bridged to the receive line.
- Correct installation of STEP 7 (>= V5.1 + Service Pack 2) on your PG.
- The PG is connected to the CPU.

How to prepare the examples

1. Using SIMATIC Manager, open the sample project **ZEn26_01_TF____31xC_PtP** in your ...\\STEP7\\EXAMPLES folder. Copy it under a suitable filename to your project directory (**File > Save as**). Select **View > Details** to display the full information.
2. Configure a station in your project according to your hardware structure.
3. Select a sample program and copy it to this station.
4. Configure all your hardware in **HW Config**. Then select the TF submodule **PtP**.
5. Configure this submodule according to the sample parameters.
6. Save your hardware configuration and download it to the CPU.
7. Download the block library to the CPU.

The sample code

The samples are written in STL/FDB. You can view them directly in your LAD/STL/LAD Editor.

In the Editor, select "Symbolic layout", "Symbol selection" and "Comment" via **View > View with**. You can also display "Symbol information", provided your desktop offers sufficient space.

Using the sample

You can view and edit values, using the variable tables (VAT) included in the sample programs.

Subsequent use of a sample

Subsequently, you can use the sample code directly as a user program.

However, the sample code is not optimized or intended to cover all options. Error evaluation is not extensively programmed in the sample programs in order to restrict the program size.

3.2 Point-to-Point Communication

The sample project consists of three independent components:

1. **CPU 31xC ASCII**: connection with SEND and RCV for ASCII/3964(R)
2. **CPU 31xC ASCII BCC**: connection for ASCII with BCC generation
3. **CPU 31xC RK512**: connection with SEND and SERVE for RK512

Without having to make changes, you can operate this program sample "CPU 31xC ASCII" in your user program with:

- ASCII driver with end criterion "On expiration of character delay time"
- ASCII driver with end criterion "After receiving fixed message frame length"
- ASCII driver with end criterion "After receiving the delimiter(s)".
Also program the delimiters in your user program.
- 3964(R) procedure

Block structure

The table below shows you the blocks used in the sample programs:

CPU 31xC ASCII:

Block	Symbol	Comment
OB 1	CYCLE	Cyclic program processing
OB 100	RESTART	Startup job Restart
FC 21	SEND	Send data
FC 22	RECV	Receive Data
DB 21	SEND IDB	Instance-DB to SFB SEND_PTP
DB 22	RECV IDB	Instance-DB to SFB RCV_PTP
DB 40	SEND WORK DB	Work-DB for SFB SEND_PTP
DB 41	RCV WORK DB	Work-DB for SFB RCV_PTP
DB 42	SEND SRC DB	Send DB
DB 43	RECV DST DB	Receive DB
SFB 60	SEND_PTP	Send data SFB
SFB 61	RCV_PTP	Receive data SFB

CPU 31xC ASCII BCC:

Block	Symbol	Comment
OB 1	CYCLE	Cyclic program processing
OB 100	RESTART	Startup job Restart
FC 21	SEND	Send data
FC 22	RECV	Receive Data
FC 23	GEN BCC	Generates BCC for the sender
FC 24	CHK BCC	Checks BCC when receiving
DB 21	SEND IDB	Instance-DB to SFB SEND_PTP
DB 22	RECV IDB	Instance-DB to SFB RCV_PTP
DB 40	SEND WORK DB	Work-DB for SFB SEND_PTP
DB 41	RECV WORK DB	Work-DB for SFB RCV_PTP
DB 42	SEND SRC DB	Send DB
DB 43	RECV DST DB	Receive DB
SFB 60	SEND_PTP	Send data SFB
SFB 61	RCV_PTP	Receive data SFB

CPU 31xC RK512:

Block	Symbol	Comment
OB 1	CYCLE	Cyclic program processing
OB 100	RESTART	Startup job Restart
FC 21	SEND	Send data
FC 22	SERVE	Receive Data
DB 21	SEND IDB	Instance-DB to SFB SEND_RK
DB 22	SERVE IDB	Instance-DB to SFB SEND_RK
DB 40	SEND WORK DB	Work-DB for SFB SEND_RK
DB 41	SERVE WORK DB	Work-DB for SFB SEND_RK
DB 42	SEND SRC DB	Send DB
DB 43	SERVE DST DB	Receive DB
DB 80	SYNC DB	Synchronization DB
SFB 63	SEND_RK	SFB sending data with RK 512
SFB 65	SERVE_RK	SFB receiving data with RK 512

Error response

If a startup error has occurred the periodically processed block calls are terminated with error.

If an error message occurs, parameter output ERROR of the blocks is set. A precise error description is then found in the block parameter STATUS.

Startup

OB 100 contains the startup program. Control bits and counters are reset at startup.

3.2.1 CPU 31xC ASCII

Cyclic Program

OB 1 contains the cyclic program for connections with SEND and RCV for ASCII/3964(R).

In the example SFB 61 "RCV_PTP" and SFB 60 "SEND_PTP" operate in correlation with the functions FC 21 and FC 22, as well as with the data blocks DB 21/DB 22 as instance-DBs and DB 42/DB 43 as send or receive DB.

The sample FBs are configured via constants and symbolically addressed operands.

Data transmission

Data are transferred from the CPU to your communication partner.

Description of FC 21 (SEND)

- Program section "Generate edge for SEND_REQ":
"SEND_PTP" is initially processed once with SEND_REQ = FALSE.
SEND_REQ is then set TRUE. The job "SEND_PTP" is started when a positive edge is detected at control parameter SEND_REQ.

SEND_DONE = TRUE or SEND_ERROR = TRUE resets SEND_REQ to FALSE.
- Program section "Done without error, SEND_DONE = TRUE":
After successful data transfer parameter SEND_DONE is set TRUE at the parameter output of "SEND_PTP".
In order to distinguish sequential transfers, data word 0 of source block DB 42 includes the send counter SEND_COUNTER_OK.
- Program section "Done with error, SEND_ERROR = TRUE":
If "SEND_PTP" is processed with SEND_ERROR = TRUE the error counter SEND_COUNTER_ERR is incremented in data word 2. SEND_STATUS is also copied to another location, because retrieval would be prevented since it is overwritten with 0 in the next cycle.

Description of FC 22 (RCV)

- Program section "Enable Receive Data":
Receive Data Enable (RCV_EN_R) at the block "RCV_PTP" must be set TRUE in order to receive data.
- Program section "No receive error, RCV_NDR = TRUE":
When RCV_NDR is set, new data were received and the receive counter RCV_COUNTER_OK is incremented.
- Program section "Receive Error, RCV_NDR = TRUE":
The error counter RCV_COUNTER_ERR is incremented if a process error occurs, that is, if the error bit at the parameter output of "RCV_PTP" is set. REC_STATUS is also copied to another location, because retrieval would be prevented since it is overwritten with 0 in the next cycle.

For testing you can monitor all relevant values in the VAT.

3.2.2 CPU 31xC ASCII BCC

This program sample only makes sense for the ASCII driver.

Its structure is widely identical to the "CPU 31xC ASCII" sample, however, it also includes

- BCC generation in the send section via call of FC 23 (GEN BCC) and a
- BCC check via call of FC 24 (CHK BCC) in the receive section.

Description of FC 23 (GEN BCC)

The input parameter "DB_NO" specifies the DB for storing send data. Input parameter "LEN" specifies the length of send data. The last two bytes must be reserved for BCC (stored in ASCII value).

The block check is carried out per logical XOR link of all send bytes (LEN -2). The XOR link is carried out in "LOOP". The L DBB [#d_loop_akt] instruction loads the respective byte. The offset "#d_loop_akt" must be specified as bit offset. After the "LOOP" cycle, the BCC is written to the low byte of variable "#w_bcc_value". As shown in the sample below, the BCC is coded in two ASCII characters.

Example:

A data block with a total length of 22 bytes (20 bytes data + 2 bytes BCC) is to be sent. The variable #w_bcc_value contains the result of the XOR link of the 20 send data bytes: 0025 Hex. This value is now converted into two ASCII characters according to the scheme shown below:

Generation of BCC-HI ASCII character "32" = 2

#w_bcc_value	0025	
UW w#16#F0	00F0	//Hide lower nibble
=	<u>0020</u>	
SLW 4	0200	// shift left by half byte
OW w#16#3000	3000	// OR link 3000 Hex
=	<u>3200</u>	// BCC-HI ASCII character

Generation of BCC-LO ASCII character "35" = 5

#w_bcc_value	0025	
UW w#16#F	000F	//Hide upper nibble
=	<u>0005</u>	
OW w#16#30	0030	// OR link 30 hex
=	<u>0035</u>	// BCC-LO ASCII character

The BCC-LO and BCC-HI ASCII characters are OR linked to form BCC 3235 Hex. Then, BCC is written to Word 20 of the DB (L DBW [#d_loop_akt]).

Description of FC 24 (CHK BCC)

The FC24 program is near identical to that of FC23. The only difference being, that at the end the calculated BCC is compared with the BCC of the received data and returned at output parameter #RETVAL:

- 0 BCC OK or
- -1 BCC false

3.2.3 CPU 31xC RK512

Cyclic Program

OB 1 contains the cyclic program for connections with SEND and SERVE for RH512.

In the example SFB 63 "RCV_PTP" and SFB 65 "SEND_PTP" operate in correlation with the functions FC 21 and FC 22, as well as with the data blocks DB 21/DB 22 as instance-DBs and DB 42/DB 43 as send or receive DB. The FBs of the sample are configured via constants and symbolically addressed operands.

Data transmission

Data are transferred from the CPU to your communication partner.

Description of FC 21 (SEND)

- Program section "Generate edge for SEND_REQ":
"SEND_RK" is initially processed once with SEND_REQ = FALSE. SEND_REQ is then set TRUE. The job "SEND_REQ" is started when a positive edge is detected at control parameter SEND_RK. SEND_DONE = TRUE or SEND_ERROR = TRUE resets SEND_REQ to FALSE.
- Program section "Done without error, SEND_DONE = TRUE":
After successful data transfer parameter SEND_DONE is set TRUE at the parameter output of "SEND_RK". In order to distinguish sequential transfers, data word 0 of source block DB 42 includes the send counter SEND_COUNTER_OK.
- Program section "Done with error, SEND_ERROR = TRUE":
If "SEND_RK" is processed with SEND_ERROR = TRUE the error counter SEND_COUNTER_ERR is incremented in data word 2. SEND_STATUS is also copied to another location, because retrieval would be prevented since it is overwritten with 0 in the next cycle.

Description of FC 22 (SERVE)

- Program section "Enable Server Function":
Receive Data Enable (RCV_EN_R) at the block "RCV_PTP" must be set TRUE in order to receive data.
- Program section "No Server Function Error, SERVE_NDR = TRUE":
When SERVE_NDR is set, new data were received and the receive counter RCV_COUNTER_OK is incremented.
- Program section "Server Function Error, SERVE_NDR = TRUE":
The error counter SERVE_COUNTER_ERR is incremented if a process error occurs, that is, if the error bit at the parameter output of "RCV_RK" is set. SERVE_STATUS is also copied to another location, because retrieval would be prevented since it is overwritten with 0 in the next cycle. For testing you can monitor all relevant values in the VAT.

4 Sample Programs for "Controlling" Technology

4.1 Introduction

Overview

In this chapter you can find samples of controlling technology.

If the CD included with your documentation does not contain these samples, you can also download them from the Internet. The project consists of several commented S7 programs of diverse complexity and aiming points.

The Readme.wri on the CD describes how to install the samples. After installation the samples are stored in the catalog

...\STEP7\EXAMPLES\ZEn26_04_TF____31xC_PID.

Requirements

- You have installed and wired an S7 station, consisting of a power supply module and a CPU 313CPtP or CPU 314CPtP.
- Correct installation of STEP 7 (>= V5.1 + Service Pack 2) on your PG.
- The PG is connected to the CPU.

How to prepare the examples

1. Using SIMATIC Manager, open the sample project **ZEn26_04_TF____31xC_PID** in your ...\\STEP7\\EXAMPLES folder. Copy it under a suitable filename to your project directory (**File > Save as**).
Select **View > Detail** to display the full information.
2. Configure a station in your project according to your hardware structure.
3. Select a sample program and copy it to this station.
4. Configure all your hardware in **HW Config**.
5. Save your hardware configuration and download it to the CPU.
6. Download the block library to the CPU.

The sample code

The samples are written in STL/FDB. You can view them directly in your LAD/STL/LAD Editor.

In the Editor, select "Symbolic layout", "Symbol selection" and "Comment" via **View > View with**. You can also display "Symbol information", provided your desktop offers sufficient space.

Using the sample

You can view and edit values, using the VATs included in the sample programs.

Subsequent use of a sample

Subsequently, you can use the sample code directly for a user program.

However, the sample code is not optimized or intended to cover all options. Error evaluation is not extensively programmed in the sample programs in order to restrict the size of the programs.

4.2 Controlling

Here you will find three S7 programs for CONTROLLING technology:

1. **Controlling 1 CONT_S:**
Example for the step controller CONT_S with distance simulation.
2. **Controlling 2 CONT_C:**
Example for the continuous controller CONT_C with distance simulation.
3. **Controlling 3 PULSEGEN:**
Example for the continuous controller CONT_C with downstream pulse generator PULSEGEN and distance simulation.

4.2.1 Controlling 1 CONT_S with distance simulation.

Target

In this sample you commission a step controller (set-value controller) with distance simulation.

You learn especially how to configure the system function block (SFB) CONT_S.

Prerequisites:

In HW Config you have set the watchdog interrupt for OB 35 to 100 ms.

Application

The "Controlling 1 CONT_S" sample includes a step controller (CONT_S) in combination with a simulated control system, consisting of an integrated controlling element and a downstream on delay element of the third order (PT3).

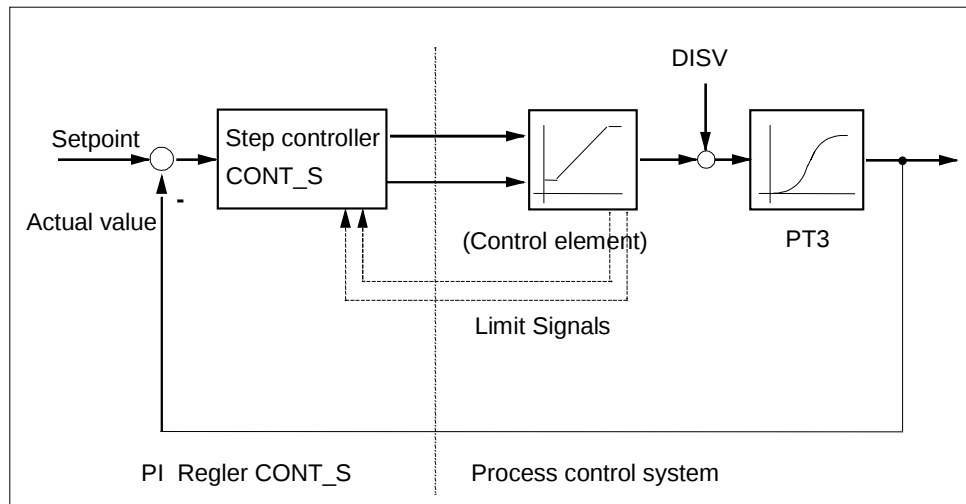
It is possible with the help of this sample to generate a stepping controller in a simple way and to configure and test all its properties for a standard controlling system.

The sample helps you to understand the way controllers with pulse output operate in frequently used control systems with integrated control elements (e. g. motors) and how they are configured. This example is therefore highly suitable for introduction and training purposes.

You can approximate the properties of your system controls to the physical process by selecting respective parameters.

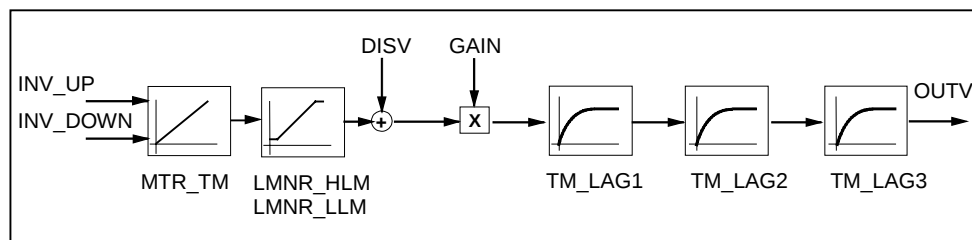
Functions

Sample 1 mainly consists of the two FBs **CONT_S** and **PROC_S**. **CONT_S** represents the used step controller, while **PROC_S** simulates a process control system with the operating elements "Valve" and **PT3**. Here, the controller receives information relating to reached limit signals and controlled variables.



The figure below shows the structure and parameters of the system function block **PROC_S**. The function block **PROC_S** forms an image of a series circuit, consisting of the integrated control element and three delay elements of the 1st order. The disturbance variable **DISV** is always added to the output signal of the control element. You can manually connect a disturbance variable at this point. Static gain of the control system is specified via **GAIN** factor.

The motor time parameter **MTR_TM** defines the time the control system requires for one point-to-point pass.



Block structure

Sample 1 includes the restart block (OB 100) and a watchdog interrupt OB (OB 35 with 100 ms clock cycle) which are used for calling the step controller CONT_S as well as the process control system simulation PROC_S.

Block	Name (in the toolbar)	Description
OB100	RESTART	Restart OB
OB35	CYC_INT5	Time-controlled OB: 100 ms
SFB42	CONT_S	Step controller
FB100	PROC_S	Process control system for step controllers
DB100	DI_PROC_S	Instance-DB to FB PROC_S
DB101	DI_CONT_S	Instance-DB to SFB CONT_S

The Instance-DBs DB 100 for the process control and DB101 for the controller are assigned to these two function blocks.

The system function block parameters

The parameters of the system function block CONT_S and their meaning are described in the manual. You can also refer to the Online Help for information on these SFBs.

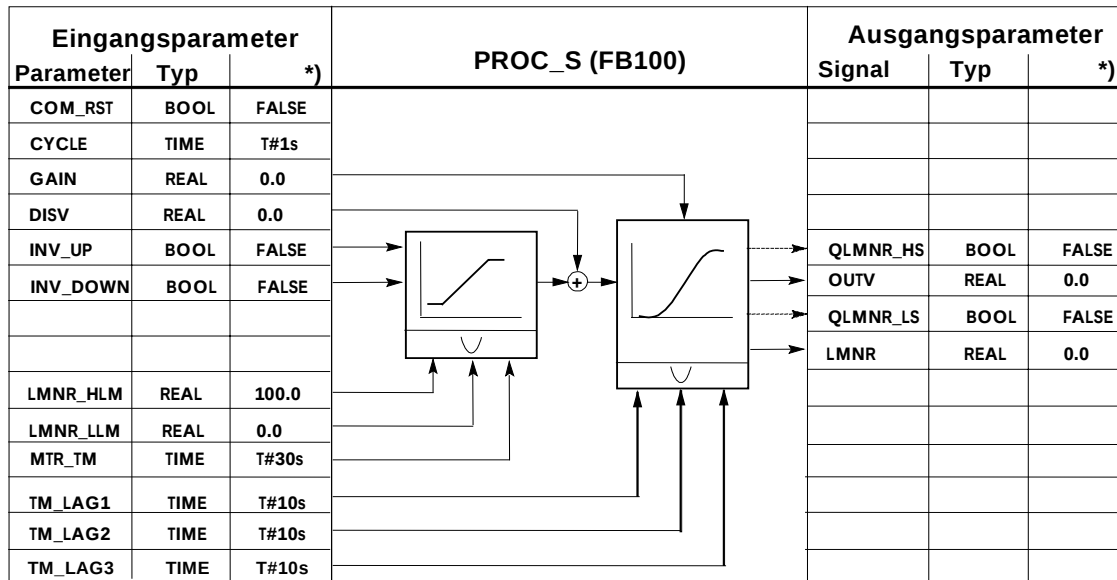
The parameters of the system control model

The parameters of the system function block CONT_S and their meaning are described in the manual. You can also refer to the SFB Online Help.

The table below lists the parameters of the system function block PROC_S:

Parameters	Type	Value Range	Description
INV_UP	BOOL		Input signal Up (more)
INV_DOWN	BOOL		Input signal down (less)
COM_RST	BOOL		Restart:
CYCLE	TIME	≥ 20 ms	Sampling rate
DISV	REAL		Disturbance variable
GAIN	REAL		Control system gain
MTR_TM	TIME		Motor actuating time
LMNR_HLM	REAL	LMNR_LLM ...100.0 [%]	Upper limit of position feedback
LMNR_LLM	REAL	-100.0... LMNR_HLM [%]	Lower limit of position feedback
TM_LAG1	TIME	$\geq \text{CYCLE}/2$	Delay time 1
TM_LAG2	TIME	$\geq \text{CYCLE}/2$	Delay time 2
TM_LAG3	TIME	$\geq \text{CYCLE}/2$	Delay time 3
OUTV	REAL		Output variable

Parameters	Type	Value Range	Description
LMNR	REAL		Position feedback
QLMNR_HS	BOOL		Control element at upper limit
QLMNR_LS	BOOL		Control element at lower limit

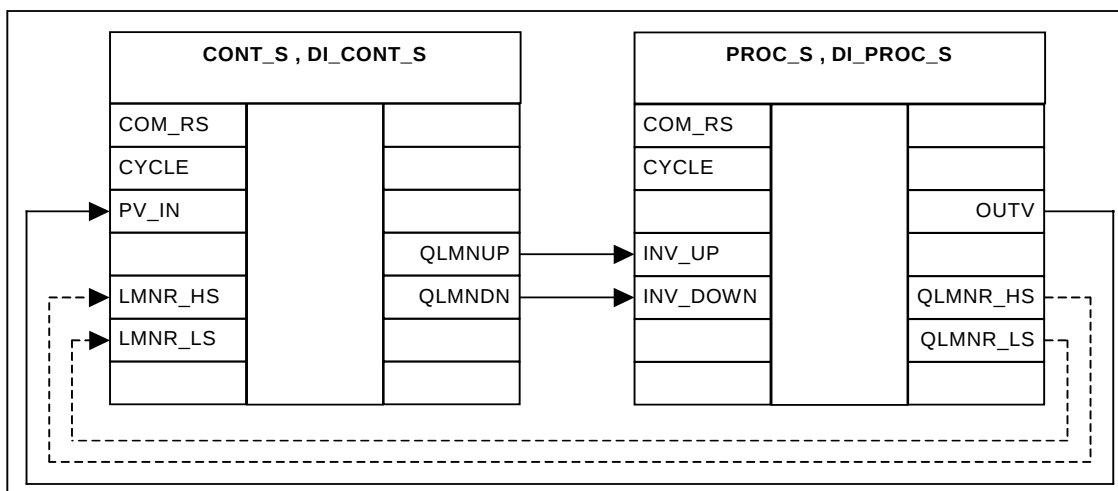


*) Vorbelegung bei Neuerstellung des Instanz-DB

The output variable OUTV and all internal ,memory variables are reset to zero after **Restart**.

Connecting the step controller to the process control system model

The figure below shows how a control circuit is formed with the step controller and the control system model:



Configuring the control circuit

This example of the configuration of a step controller with PI action and set dead-time demonstrates the reaction of a control circuit with a simulated PT-process control system of the 3rd order. The control parameters are delayed respectively by 10 s. They form an approximate image of a high-speed process control system (e. g. temperature or filling level control).

The order of the control by one degree if one of the delay time **TM_LAGx** is set to 0 s.

The tables below contain the actual configured values of the parameters relevant to the controller and control system.

Controller:			
Parameters	Type	Parameter assignment	Description
CYCLE	TIME	100 ms	Sampling rate
GAIN	REAL	0.31	Proportional gain
TI	TIME	19.190 s	Integration time
MTR_TM	TIME	20 s	Motor actuating time
PULSE_TM	TIME	100 ms	Minimum pulse width
BREAK_TM	TIME	100 ms	Minimum pulse width
DEADB_W	REAL	0.5	Dead-time width

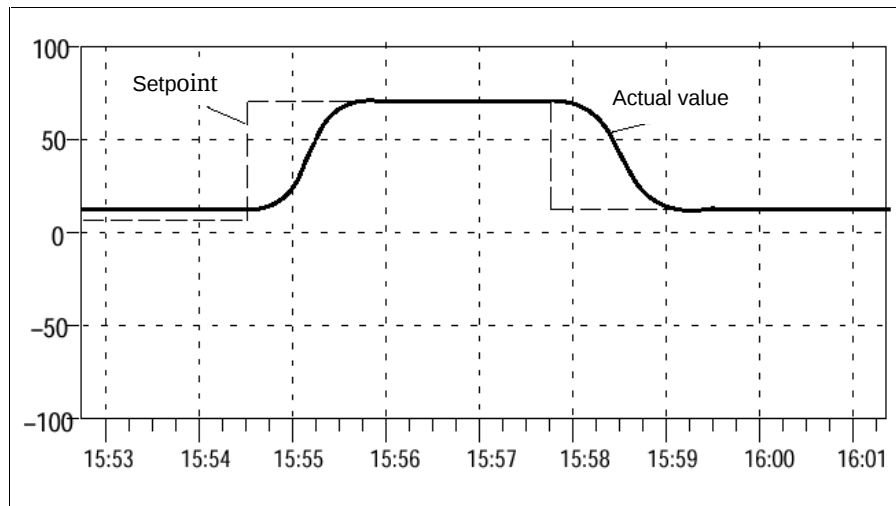
Control system:			
Parameters	Type	Parameter assignment	Description
GAIN	REAL	1.5	Control system gain
MTR_TM	TIME	20 s	Motor actuating time
TM_LAG1	TIME	10 s	Delay time 1
TM_LAG2	TIME	10 s	Delay time 2
TM_LAG3	TIME	10 s	Delay time 3

Monitoring step response

Set and enable a new setpoint **SP_INT** in the VAT to monitor the reaction of the control system output **OUTV** (actual value).

As shown in the figure below, you can also visualize the control variables with the help of the STEP 7 tool **PID Control**. Open your sample project in **PID Control** and click on "Online" mode. In the block folder, select the Instance-DB 101 of SFB 42 CONT_S. Confirm with OK and enable the **Curve Plotter** function.

The following graph illustrates the transient response of the closed control circuit after a setpoint modification of 60 %:



4.2.2 Controlling 2 CONT_C, Continuous controller CONT_C with distance simulation

Target

In this sample you commission a continuous controller.

You especially learn how to configure the system function block (SFB) CONT_C.

Prerequisites:

In HW Config you have set the watchdog interrupt for OB35 to 100 ms.

Application

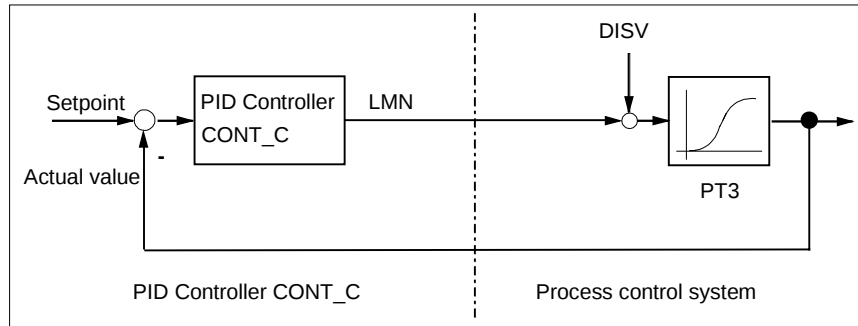
Sample 2 includes a continuous controller (CONT_C) in combination with a simulated control system that consists of a delay element of the third order (PT3).

Sample 2 helps you to generate a continuous PID controller in a simple way and to configure and test all its properties for a standard controlling system.

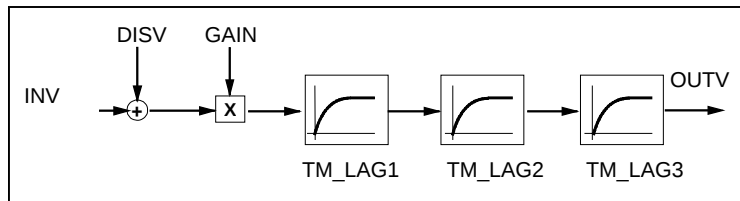
It also helps you to easily understand the way controllers with analog output operate and how they are configured, e. g. for controlling systems with proportional control elements. This example is therefore highly suitable for introduction and training purposes.

Functions

Sample 2 consists primarily of the SFB CONT_C and the function block PROC_C. CONT_C represents the used controller, PROC_C simulates a control system with compensation of the third order.



The figure below shows the structure and parameters of SFB PROC_S. SFB PROC_C forms an image of a series circuit consisting of three delay elements of the first order. The disturbance variable **DISV** is always added to the output signal of the control element. You can manually connect a disturbance variable at this point. Static gain of the control system is specified via **GAIN** factor:



Block structure

Sample 2 includes the restart block (OB 100) and a watchdog interrupt OB (OB 35 with 100 ms clock cycle), used for calling the step controller CONT_S as well as process control system simulation PROC_S.

Block	Name (in the toolbar)	Description
OB100	RESTART	Restart OB
OB35	CYC_INT5	Time-controlled OB: 100 ms
SFB41	CONT_C	Continuous PID controller
FB100	PROC_C	Control system for a continuous controllers
DB100	DI_PROC_C	Instance-DB to FB PROC_C
DB101	DI_CONT_C	Instance-DB to SFB CONT_C

The Instance-DBs DB100 for the control system and DB101 for the controller are assigned to these two function blocks.

The system function block parameters

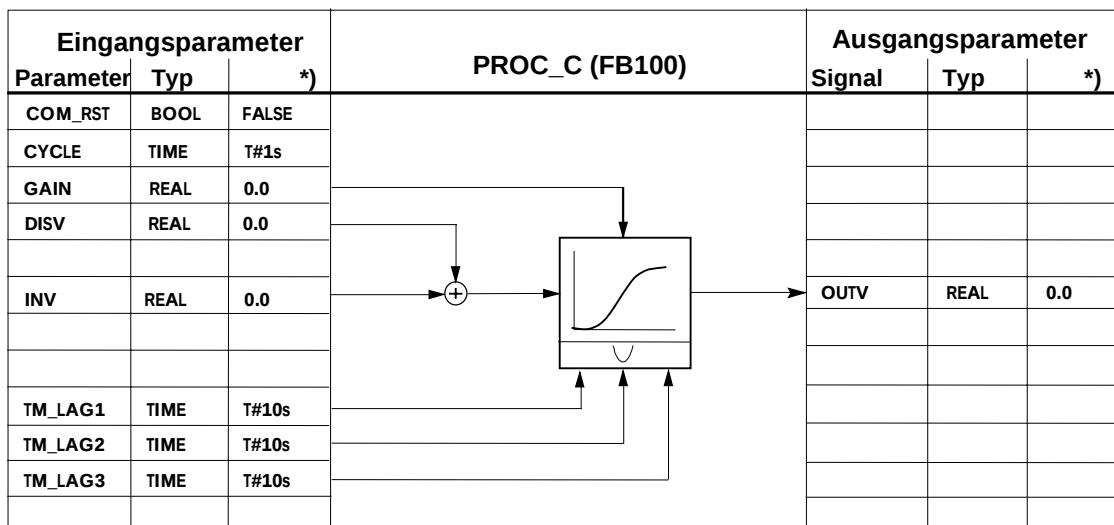
The parameters of SFB CONT_C and their meaning are described in the manual. You can also refer to the SFB Online Help.

The parameters of the system control model

The table below lists the parameters of the system function block PROC_C.

Parameters	Type	Value Range	Description
INV	REAL		Input variable
COM_RST	BOOL		Restart:
CYCLE	TIME	$\geq 1 \text{ ms}$	Sampling rate
DISV	REAL		Disturbance variable
GAIN	REAL		Control system gain
TM_LAG1	TIME	$\geq \text{CYCLE}/2$	Delay time 1
TM_LAG2	TIME	$\geq \text{CYCLE}/2$	Delay time 2
TM_LAG3	TIME	$\geq \text{CYCLE}/2$	Delay time 3
OUTV	REAL		Output variable

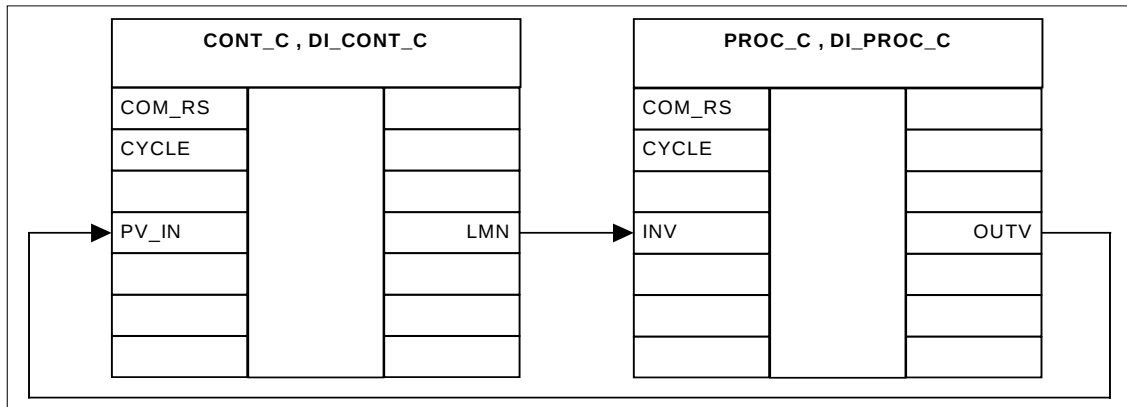
The output variable OUTV and all internal memory variables are reset to zero after **Restart**.



*) Vorbelegung bei Neuerstellung des Instanz-DB

Connecting continuous controllers to the control system model

The figure below shows how to interconnect the continuous controller and the control system model to form a control circuit.



Configuring the control circuit

This example of the configuration of a continuous controller with PID action and set dead-time demonstrates the reaction of a control circuit with simulated PT process control system of the third order. The control system parameters are delayed respectively by 10 s. They form an approximate image of a high-speed process control system. The order of the control system is reduced by one degree if one of the delay time TM_LAG_x is set to 0 s.

The tables below contain the actual configured values of the parameters which are relevant to the controller and the control system.

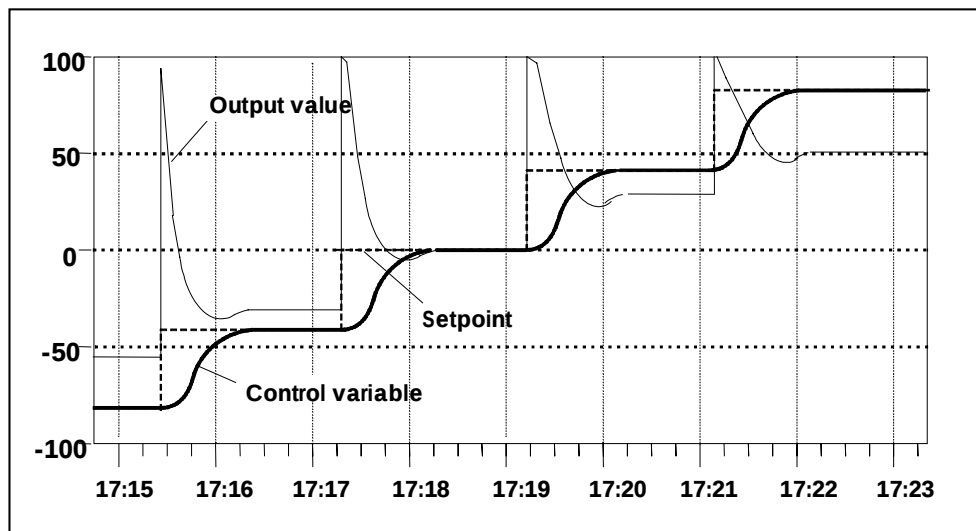
Controller			
Parameters	Type	Parameter assignment	Description
CYCLE	TIME	100 ms	Sampling rate
GAIN	REAL	1.535	Proportional gain
TI	TIME	22.720 s	Integration time
TD	TIME	5.974 s	Differentiation time
TM_LAG	TIME	1.195 s	Delay time of the D-portion
LMN_HLM	REAL	100.0	Control value upper limit
LMN_LLM	REAL	-100.0	Control value lower limit

Control system			
Parameters	Type	Parameter assignment	Description
GAIN	REAL	1.5	Control system gain
TM_LAG1	TIME	10 s	Delay time 1
TM_LAG2	TIME	10 s	Delay time 2
TM_LAG3	TIME	10 s	Delay time 3

Monitoring step response

Set and enable a new setpoint **SP_INT** in the VAT to monitor the reaction of the control system output **OUTV** (actual value).

As shown in the figure below, you can also visualize the control variables with the help of the STEP 7 tool **PID Control**. Open your sample project in **PID Control** and click on "Online" mode. In the block folder, select the Instance-DB 101 of SFB 41 CONT_C. Confirm with OK. Enable the **Curve Plotter** function. The following graph illustrates the transient response of the closed-loop control circuit after a setpoint modification of 20 %.



4.2.3 Controlling 3 PULSEGEN Continuous controller CONT_C with downstream pulse generator PULSEGEN and control system simulation.

Target

In this sample you commission a continuous controller CONT_C with downstream pulse generator PULSEGEN.

You especially learn how to configure the system function block (SFB) CONT_C and PULSEGEN.

Prerequisites:

In HW Config, you have set the watchdog interrupt for OB35 to 100 ms.

Application

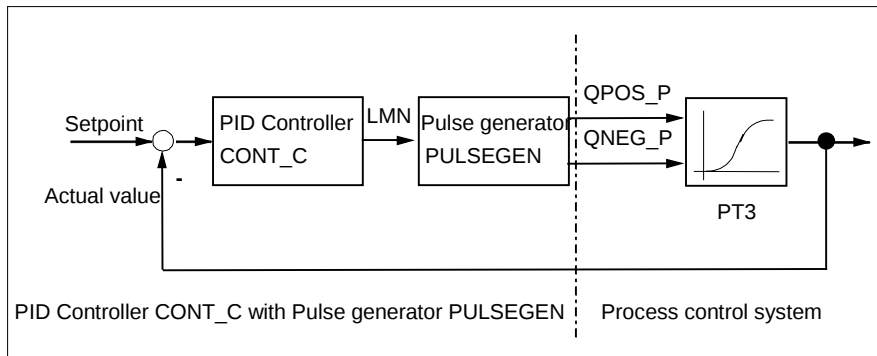
Sample 3 includes a continuous controller (CONT_C) in combination with a simulated control system that consists of a delay element of the third order (PT3).

Sample 3 helps you to generate a continuous PID controller in a simple way and to configure and test all its properties for a standard controlling system.

The sample also helps you to easily understand the way controllers with digital output operate and how they are configured, e. g. for controlling systems with proportional control elements. The controllers are used, for example, for temperature control of electrical heating and cooling systems. This example is therefore highly suitable for introduction and training purposes.

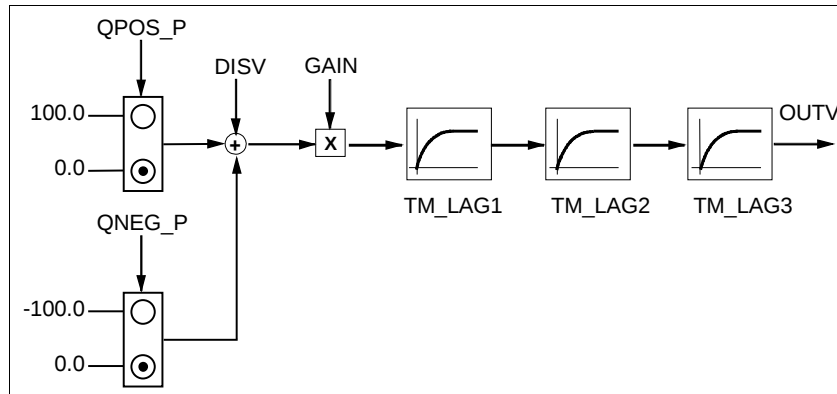
Functions

Sample 3 primarily consists of the two SFBs CONT_C, PULSEGEN and PROC_P. CONT_C represents the used controller, PULSEGEN the pulse generator and PROC_P simulates a control system with compensation of the third order:



The figure below shows the structure and parameters of SFB PROC_P. SFB PROC_P forms an image of a series circuit consisting of three delay elements of the first order. Not only the pulse inputs POS_P and NEG_P form the input signals for the control system, but also the disturbance variable **DISV**. At this point you can manually connect disturbance variables.

The static gain of the control system is specified via a **GAIN** factor.



Block structure

Sample 3 includes the function block PULS_CTR containing the controller blocks (CONT_C and PULSEGEN), the simulated control system PROC_P as well as the call blocks for restart (OB 100) and the watchdog interrupt OB (OB 35 with 100 ms clock cycle).

Block	Name (in the toolbar)	Description
OB100	RESTART	Restart OB
OB35	CYC_INT5	Time-controlled OB: 100 ms
SFB41	CONT_C	Continuous PID controller
SFB43	PULSEGEN	Pulse generator
FB100	PROC_P	Control system for continuous controllers with pulse inputs
DB100	DI_PROC_P	Instance-DB to FB PROC_P
FB101	PULS_CTR	Call of the SFBs CONT_C and PULSEGEN
DB101	DI_PULS_CTR	Instance-DB to FB PULS_CTR

SFB PULS_CTR is assigned to Instance-DB DB 101, SFB PROC_P to DB 100.

The controller parameters

The parameters of the system function block CONT_C and their meaning are described in the manual. You can also refer to the Online Help for information on these SFBs.

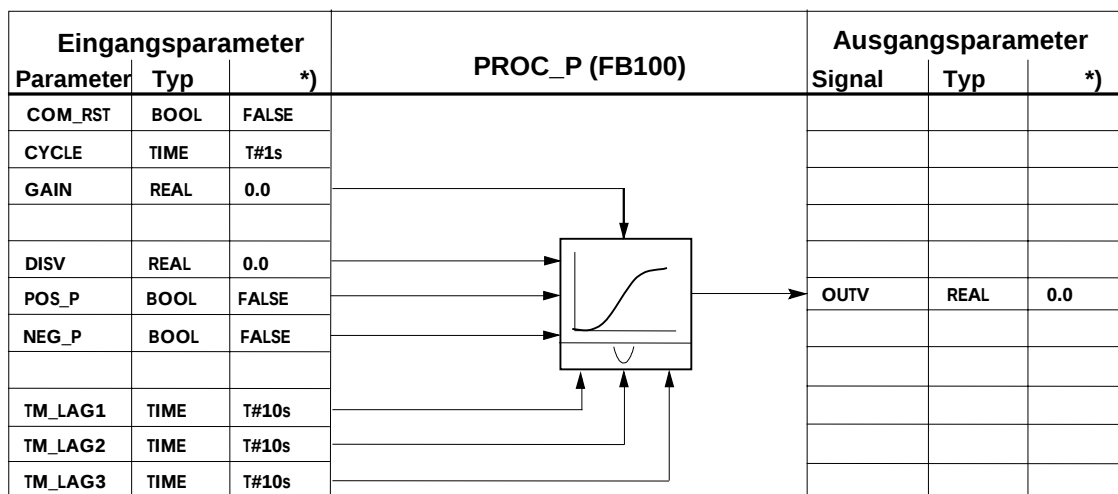
The parameters of the system control model

The parameters of SFB CONT_C and of the pulse generator PULSEGEN and their meaning are described in the manual. You can also refer to the Online Help for information on these SFBs.

The table below lists the parameters of SFB PROC_S:

Parameters	Type	Value Range	Description
POS_P	BOOL		Positive pulse
NEG_P	BOOL		Negative pulse
COM_RST	BOOL		Restart:
CYCLE	TIME	$\geq 1 \text{ ms}$	Sampling rate
DISV	REAL		Disturbance variable
GAIN	REAL		Control system gain
TM_LAG1	TIME	$\geq \text{CYCLE}/2$	Delay time 1
TM_LAG2	TIME	$\geq \text{CYCLE}/2$	Delay time 2
TM_LAG3	TIME	$\geq \text{CYCLE}/2$	Delay time 3
OUTV	REAL		Output variable

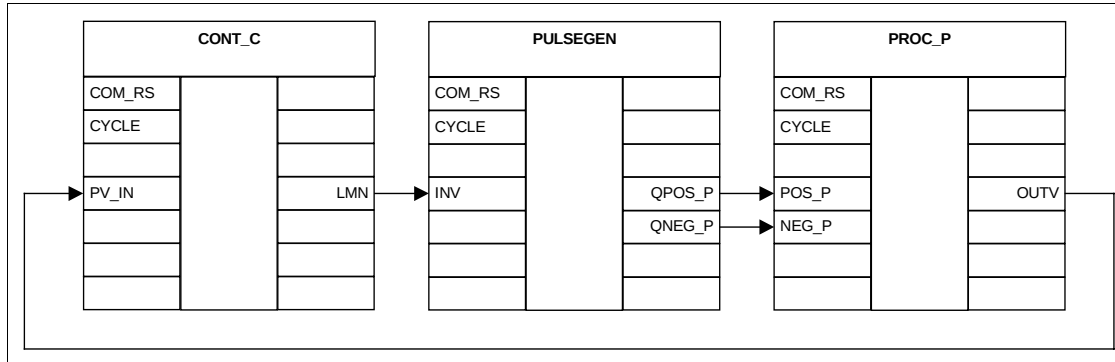
The output variable OUTV and all internal memory variables are reset to zero after **Restart**.



*) Vorbelegung bei Neuerstellung des Instanz-DB

Connecting continuous controllers to the control system model

The figure below shows how to interconnect a continuous controller and the control system model to form a control circuit.



Configuring the control circuit

Example 3 of the configuration of a step controller with PID action and set dead-time demonstrates the reaction of a control circuit with simulated PT process control system of the third order. The control parameters are delayed respectively by 10s. They form an approximate image of a fast process control system (e. g. as commonly used for temperature control). Due to the relatively fast performance of this control system, function testing of the controller is accelerated. By modifying its delay time constants, you can also easily approximate the simulated control system to the physical control system.

The tables below contain the actual configured values of the parameters relevant to the controller, the pulse generator and the control system.

Controller			
Parameters	Type	Value Range	Description
CYCLE	TIME	2 s	Sampling rate of the controller
GAIN	REAL	1.535	Proportional gain
TI	TIME	22.720 s	Integration time
TD	TIME	5.974 s	Differentiation time
TM_LAG	TIME	1.195 s	Delay time of the D-portion
LMN_HLM	REAL	100.0	Control value upper limit
LMN_LLM	REAL	-100.0	Control value lower limit

Pulse generator			
Parameters	Type	Value Range	Description
CYCLE	TIME	100 ms	Sampling rate
P_B_TM	TIME	100 ms	Minimum pulse/pause width

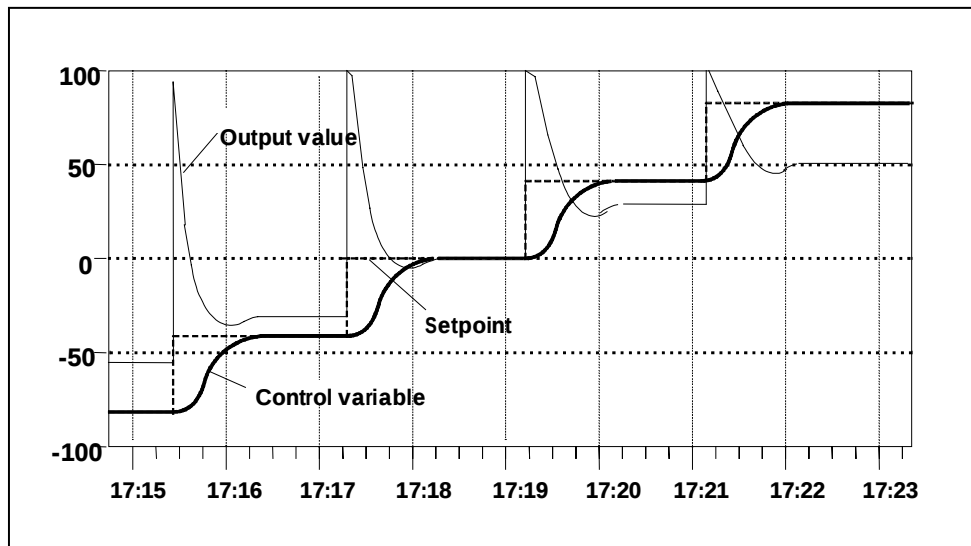
Control system			
Parameters	Type	Value Range	Description
GAIN	REAL	1.5	Control system gain
TM_LAG1	TIME	10 s	Delay time 1
TM_LAG2	TIME	10 s	Delay time 2
TM_LAG3	TIME	10 s	Delay time 3

Monitoring step response

Set and enable a new setpoint **SP_INT** in the VAT to monitor the reaction of the control system output **OUTV** (actual value).

As shown in the figure below, you can also visualize the control variables with the help of the STEP 7 tool **PID Control**. Open your sample project in **PID Control** and click on "Online" mode. Select DB 101 from the block folder and confirm with OK. In the pop-up screen form, enter a starting point offset of 16 byte in DB 101 for SFB 41 instance data. Enable the **Curve Plotter** function.

The following graph illustrates the transient response of the closed-loop control circuit after a number of setpoint modifications of 20 %. The graph does not show the pulse outputs, but rather the continuous output value of the controller.



Index

A

Analog	1-3
Analog 1 First steps.....	1-4
Analog 2 Operating modes.....	1-8
Analog 3 Loading portal	1-11

C

Controlling	4-1
Controlling 1 CONT_S.....	4-3
Controlling 2 CONT_C	4-8
Controlling 3 PULSGEN	4-12
Counter 2 Filling station.....	2-9
Counter 2 Heating	2-28
Counter 2 Speed monitoring	2-19
Counting	2-1
Counting 1 First steps	2-3
CPU 31xC ASCII	3-5
CPU 31xC ASCII BCC	3-6
CPU 31xC RK512	3-8

D

Digital.....	1-16
Digital 1 First steps	1-17
Digital 2 Operating modes	1-21
Digital 3 Loading portal	1-24

F

Frequency 1 First steps	2-14
Frequency Measurement.....	2-14

P

Point-to-Point Connection.....	3-1
Positioning	1-1
Positioning with Analog Output	1-3
Positioning with Digital Output	1-16
Pulse width modulation	2-24
PWM 1, First steps	2-24

