SIEMENS

SIMATIC

S7-300
CPU 31xC: Technological functions

Operating Instructions

This manual is part of the documentation package
with the order number: 6ES7398-8FA10-8BA0

03/2011
A5E00105484-05

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠ DANGER
indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠ WARNING
indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠ CAUTION
with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

CAUTION
without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

NOTICE
indicates that an unintended result or situation can occur if the corresponding information is not taken into account.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation for the specific task, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠ WARNING
Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be adhered to. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of the Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E00105484-05
Ⓟ 04/2011

Preface

Purpose of the Manual

This manual provides you with a complete overview of the integrated technological functions of the CPUs 31xC.

It is aimed at persons involved in implementing control tasks with technological functions based on SIMATIC automation systems.

Experience Required

To understand the manual, you should have general experience of automation engineering.

Scope of this Manual

This manual is valid for the following CPUs with the following hardware and software versions:

CPU	Convention: The following CPUs are described in this manual:	Order no.	as of Version	
			Firmware	Hardware
CPU 312C	CPU 31xC	6ES7312-5BF04-0AB0	V3.3	01
CPU 313C		6ES7313-5BG04-0AB0	V3.3	01
CPU 313C-2 PtP		6ES7313-6BG04-0AB0	V3.3	01
CPU 313C-2 DP		6ES7313-6CG04-0AB0	V3.3	01
CPU 314C-2 PtP		6ES7314-6BH04-0AB0	V3.3	01
CPU 314C-2 DP		6ES7314-6CH04-0AB0	V3.3	01
CPU 314C-2 PN/DP		6ES7314-6EH04-0AB0	V3.3	01

Note

This documentation package contains a description of all modules available at the time of publication.

We reserve the right to include separate up-to-date Product Information on new modules and new releases of existing modules.

Position in the overall documentation structure

The documentation listed below is part of the S7-300 documentation package.

You can also find this on the Internet and the corresponding entry ID.

Title of documentation	Description
Manual CPU 31xC and CPU 31x: Technical Specifications Entry ID: 12996906 (http://support.automation.siemens.com/WW/view/en/12996906)	Description of: • Operating and display elements • Communication • Memory concept • Cycle and response times • Technical specifications
Operating Instructions CPU 31xC and CPU 31x: Installation Entry ID: 13008499 (http://support.automation.siemens.com/WW/view/en/13008499)	Description of: • Configuring • Mounting • Wiring • Addressing • Commissioning • Maintenance and test functions • Diagnostics and troubleshooting
Operating Instructions CPU 31xC: Technological functions incl. CD Entry ID: 12429336 (http://support.automation.siemens.com/WW/view/en/12429336)	Description of the specific technological functions: • Positioning • Counting • Point-to-point connection • Controlling The CD contains examples for the technological functions.
Manual S7-300 Automation System: Module Data Entry ID: 8859629 (http://support.automation.siemens.com/WW/view/en/8859629)	Description of and technical specifications for the following modules: • Signal modules • Power supplies • Interface modules
List manual S7-300-CPU and ET 200-CPU instruction list Entry ID: 31977679 (http://support.automation.siemens.com/WW/view/en/31977679)	• List of the CPU instruction set and their execution times. • List of the executable blocks (OBs/SFCs/SFBs) and their execution times.

Additional information

You also require information from the following descriptions:

Title of documentation	Description
Getting Started S7-300 Automation System: Getting Started CPU 31x: Commissioning Entry ID: 15390497 (http://support.automation.siemens.com/WW/view/en/15390497)	Description of examples showing the various commissioning phases leading to a functional application.
Getting Started S7-300 Automation System: Getting Started CPU 31xC: Commissioning Entry ID: 48077635 (http://support.automation.siemens.com/WW/view/en/48077635)	Description of examples showing the various commissioning phases leading to a functional application.
Getting Started First steps in commissioning CPU 31xC: Positioning with analog output Entry ID: 48070939 (http://support.automation.siemens.com/WW/view/en/48070939)	Description of examples showing the various commissioning phases leading to a functional application.
Getting Started First steps in commissioning CPU 31xC: Positioning with digital output Entry ID: 48077520 (http://support.automation.siemens.com/WW/view/en/48077520)	Description of examples showing the various commissioning phases leading to a functional application.
Getting Started First steps in commissioning CPU 31xC: Counting Entry ID: 48064324 (http://support.automation.siemens.com/WW/view/en/48064324)	Description of examples showing the various commissioning phases leading to a functional application.
Getting Started First steps in commissioning CPU 31xC: Point-to-point connection Entry ID: 48064280 (http://support.automation.siemens.com/WW/view/en/48064280)	Description of examples showing the various commissioning phases leading to a functional application.
Getting Started First steps in commissioning CPU 31xC: Rules Entry ID: 48077500 (http://support.automation.siemens.com/WW/view/en/48077500)	Description of examples showing the various commissioning phases leading to a functional application.

Title of documentation	Description
Getting Started CPU315-2 PN/DP, 317-2 PN/DP, 319-3 PN/DP: Configuring the PROFINET interface Entry ID: 48080216 (http://support.automation.siemens.com/WW/view/en/48080216)	Description of examples showing the various commissioning phases leading to a functional application.
Getting Started CPU 317-2 PN/DP: Configuring an ET 200S as PROFINET IO device Entry ID: 19290251 (http://support.automation.siemens.com/WW/view/en/19290251)	Description of examples showing the various commissioning phases leading to a functional application.
Reference manual System and standard functions for S7-300/400, volume 1/2 Entry ID: 1214574 (http://support.automation.siemens.com/WW/view/en/1214574)	Overview of objects included in the operating systems for S7-300 and S7-400 CPUs: • OBs • SFCs • SFBs • IEC functions • Diagnostics data • System status list (SSL) • Events This manual is part of the STEP 7 reference information. You can also find the description in the STEP 7 Online Help.
Manual Programming with STEP 7 Entry ID: 18652056 (http://support.automation.siemens.com/WW/view/en/18652056)	This manual provides a complete overview of programming with STEP 7. This manual is part of the STEP 7 basic information. You can also find the description in the STEP 7 Online Help.
System manual PROFINET system description Entry ID: 19292127 (http://support.automation.siemens.com/WW/view/en/19292127)	Basic information on PROFINET: • Network components • Data exchange and communication • PROFINET IO • Component Based Automation • Application example of PROFINET IO and Component Based Automation
Programming manual From PROFIBUS DP to PROFINET IO Entry ID: 19289930 (http://support.automation.siemens.com/WW/view/en/19289930)	Guideline for the transition from PROFIBUS DP to PROFINET IO.

Title of documentation	Description
Manual SIMATIC NET: Twisted Pair and Fiber-Optic Networks Entry ID: 8763736 (http://support.automation.siemens.com/WW/view/en/8763736)	Description of: • Industrial Ethernet networks • Network configuration • Components • Guidelines for setting up networked automation systems in buildings, etc.
Configuring manual Configure SIMATIC iMap plants Entry ID: 22762190 (http://support.automation.siemens.com/WW/view/en/22762190)	Description of the SIMATIC iMap configuration software
Configuring manual SIMATIC iMap STEP 7 AddOn, create PROFINET components Entry ID: 22762278 (http://support.automation.siemens.com/WW/view/en/22762278)	Descriptions and instructions for creating PROFINET components with STEP 7 and for using SIMATIC devices in Component Based Automation.
Function manual Isochronous mode Entry ID: 15218045 (http://support.automation.siemens.com/WW/view/en/15218045)	Description of the system property "Isochronous mode"
System manual Communication with SIMATIC Entry ID: 1254686 (http://support.automation.siemens.com/WW/view/en/1254686)	Description of: • Principles • Services • Networks • Communication functions • Connecting PGs/OPs • Engineering and configuring in STEP 7

Service & Support on the Internet

Information on the following topics can be found on the Internet (http://www.siemens.com/automation/service):

- Contacts for SIMATIC (http://www.siemens.com/automation/partner)
- Contacts for SIMATIC NET (http://www.siemens.com/simatic-net)
- Training (http://www.sitrain.com)

Table of contents

1 Overview of the Technological Functions

Overview

The following technological functions are supported, depending on your type of CPU:

CPU	Positioning	Counting	Point-to-point communication	Controlling
CPU 312C	-	2 channels each for counting, frequency measurement (max. 10 kHz) or pulse width modulation (2.5 kHz)	-	-
CPU 313C	-	3 channels each for counting, frequency measurement (max. 30 kHz) or pulse width modulation (2.5 kHz)	-	Yes
CPU 313C-2 PtP	-	3 channels each for counting, frequency measurement (max. 30 kHz) or pulse width modulation (2.5 kHz)	ASCII (19.2 kbps Full Duplex, 38.4 kbps Half Duplex) 3964R (38.4 kbps)	Yes
CPU 313C-2 DP	-	3 channels each for counting, frequency measurement (max. 30 kHz) or pulse width modulation (2.5 kHz)	-	Yes
CPU 314C-2 PtP	1 channel with analog or digital output	4 channels[1] each for counting, frequency measurement (max. 60 kHz) or pulse width modulation (2.5 kHz)	ASCII (19.2 kbaud full duplex, 38.4 kbaud half duplex) 3964R (38.4 kbaud) RK512 (38.4 kbps)	Yes
CPU 314C-2 DP	1 channel with analog or digital output	4 channels[1] each for counting, frequency measurement (max. 60 kHz) or pulse width modulation (2.5 kHz)	-	Yes
CPU 314C-2 PN/DP	1 channel with analog or digital output	4 channels[1] each for counting, frequency measurement (max. 60 kHz) or pulse width modulation (2.5 kHz)	-	Yes
[1] Only two channels are available when a positioning channel is used				

Access to I/Os Used by Technological Functions

Inputs used by the technological functions can always be accessed via the input address of the digital input I/O.

Write access to outputs used by the technological functions is internally locked.

Positioning 2

2.1 Modes Supported by the Controlled Positioner

2.1.1 Positioning Control with Analog Output

Introduction

The CPU supports controlled positioning with analog output.

Properties

Controlled positioning with analog output has the following features:

- The drive is controlled via a permanently assigned **analog output**, either with a voltage of ±10 V (pin 16) or a current of ±20 mA (pin 17), or with a voltage of 0 to 10 V (pin 16) or a current of 0 to 20 mA (pin 17) and an additional 24 V digital output as a directional signal (X2, pin 29).
- Controlling a brake or enabling a drive via a permanently assigned 24 V digital output (X2, pin 28).
- Connectable are, for example, servo-drive motors via converter or asynchronous motors via frequency converter.
- A 24 V encoder is used for incremental position feedback.
- The run is performed with the help of a specifiable acceleration and deceleration ramp.
- First, the axis is accelerated up to a specified speed. At a defined distance from the target, the axis is decelerated to a lower speed (creep speed). The drive is de-energized at a specified distance just before the axis reaches the target. The CPU can monitor the target approach during the process.
- You specify the speed, deceleration and target approach differences in your parameters.

2.1.2 Controlled Positioning with Digital Outputs

Introduction

The CPU supports controlled positioning with digital outputs (rapid / creep speed control).

Properties

Controlled positioning with digital outputs (rapid / creep speed control) has the following features:

- The drive is controlled via four permanently assigned **24 V digital outputs**. These digital outputs control the direction and speed stages (rapid/creep speed), depending on the configured type of control.
- Connectable are multiple speed, pole-switching motors via contactor combinations or asynchronous motors via frequency converter with a fixed speed setting.
- A 24 V encoder is used for incremental position feedback.
- First, the target is approached with a specified (rapid) speed. Within a defined distance to the target speed is reduced to a lower (creep) speed. The drive is de-energized at a specified distance just before the axis reaches the target. The CPU can monitor the target approach during the process.
- You declare the speed, deceleration and target approach differences in your parameters.

2.2 Positioning Overview

Overview

- Number of axes
  - CPU 314C-2 DP, PN/DP, PtP: 1 Axis

Note

You now only have counting channels
available to you when using a positioning function (channel 2 and 3).

- Axis types
  - Linear axis
  - Rotary axis
- Typically used drives/motors
  - Asynchronous motors with pole-switching via contactor combination
  - Asynchronous motors with frequency converter
  - Servo-drive motors with converter
- Distance measuring systems:
  - Incremental encoders 24 V, asymmetric, two tracks with phase difference of 90 degrees (with or without zero mark)
- Monitoring functions (can be individually activated)
  - Missing pulse (zero mark)
  - Traversing range
  - Working range
  - Actual value
  - Target approach
  - Target range
- System of units
  - All values are specified in pulses.
- Project design
  - Via parameter assignment screens

2.3 Function scope

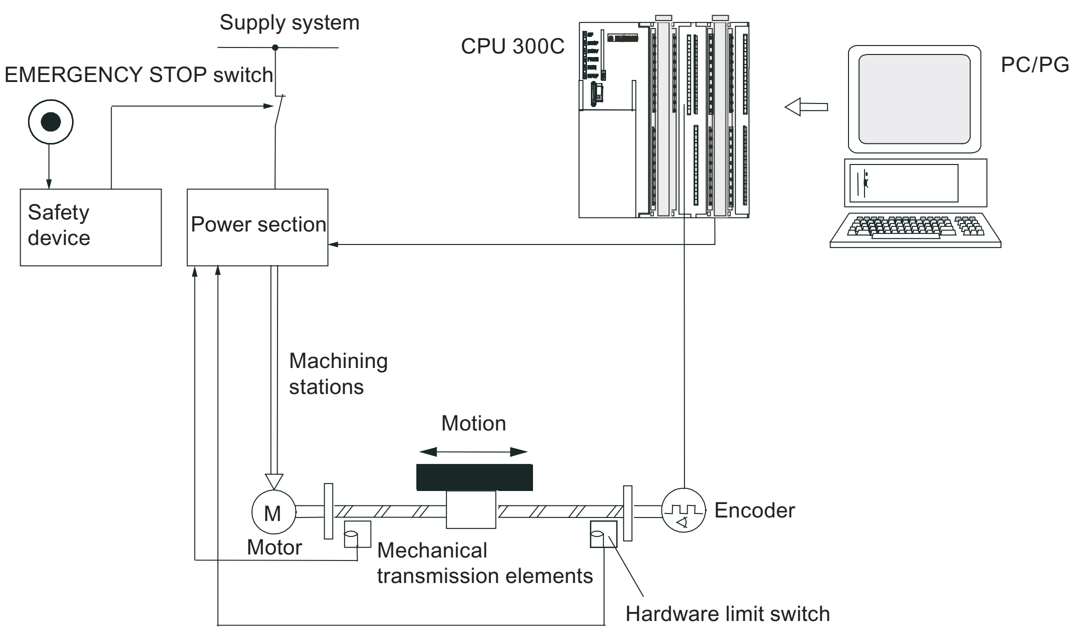
Overview

- **Operating modes:**
  - Jog mode
  - Reference Point Approach
  - Relative incremental approach
  - Absolute incremental approach
- **Additional functions:**
  - Set reference point
  - Deleting the distance-to-go
  - Length measurement

2.4 Components for Positioning Control

Basic Design

The view below shows the components for controlled positioning:



- The **CPU** uses the outputs to control the converter.
- The **converter** processes the positioning signal and controls the motor.
- When a **safety device** (Emergency-Off switch or hardware limit switch) is actuated, the converter switches off the motor.
- The **motor** is controlled through the converter and drives the axis.
- The **encoder** feeds back position and direction information.
- You can control rotary or linear axes as **mechanical transmission elements**.
- Use the **PG/PC**
  - to configure the CPU in the *parameter assignment* screens for the technological functions of the CPU.
  - to program CPU SFBs which you can implement directly in your user program.
  - atart CPU operation and test it with the help of the standard STEP 7 user interface (monitoring functions and variable table).

3 Positioning with Analog Output

3.1 Wiring

3.1.1 Important Safety Rules

Adherence to Safety Rules

DANGER

For the safety concept of the system it is imperative to install the switchgear mentioned below and to adapt them to your system:

- **The Emergency-Off switch.** You can use this to switch off the entire system.
- **Hardware limit switches** that have a direct effect on all drive power units.
- **Motor protection**

WARNING

Harm to health and damage to assets cannot be excluded if you do not switch off voltage:

If you wire the front plug of the CPU in live state you risk injury due to the influence of electrical current!

Always wire the CPU in off-voltage state!

Harm to health and damage to assets due to missing safety devices:

If no Emergency-Off Switch is installed damage can be caused by connected aggregates.

Install an Emergency-Off switch that enables you to switch off all connected drives.

Note

Direct connection of inductive loads (e.g. relays and contactors) is possible without auxiliary circuitry.

If it is possible to switch off SIMATIC output current circuits via additionally installed auxiliary contacts (e.g. relay contacts), you must install additional surge voltage suppression elements across the coils of the inductive loads.

3.1.2 Wiring Rules

Connecting Cables/Shielding

- The cables for the analog outputs and the 24 V encoder must be shielded.
- The cables for the digital I/O must be shielded if their length exceeds 100 m.
- The cable shielding must be terminated on both ends.
- Flexible cable, cross-section 0.25 mm to 1.5 mm^2.
- Cable sleeves are not required. Should you still decide to do so, use cable sleeves without insulating collar (DIN 46228, Shape A, short version).

Shielding termination element

You can use this shielding termination element for easy shielded-cable-to-ground connections – due to the direct contact of the shielding termination element to the profile rail.

Additional Information

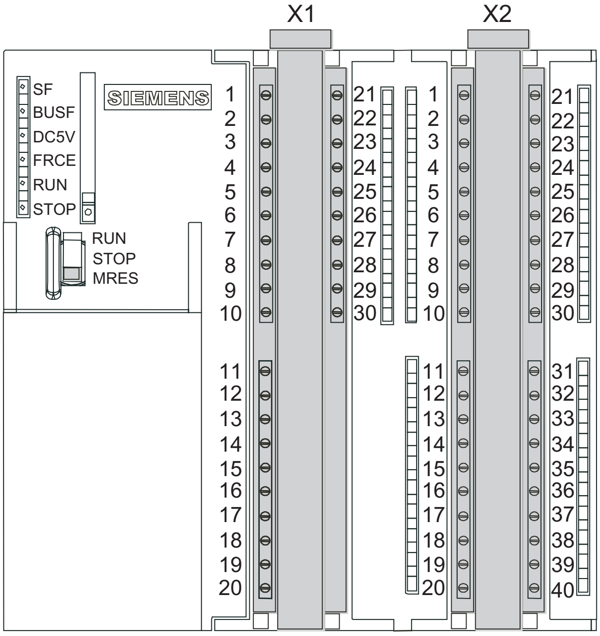
For additional information refer to the *CPU Data* manual and to the installation instructions for your CPU.

3.1.3 Terminals for Positioning with Analog Output

Introduction

Use the front connectors X1 and X2 of the CPU 314C-2 DP, PN/DP and PtP to connect the following components:

- 24 V encoder
- Length measurement switch
- Reference point switch
- Power section



Description of Pin Assignment

Then following connector pin assignments only refers to connections relevant to the positioning mode.

Note

Since they partially use the same inputs, you cannot use counters 0 and 1 when you utilize the positioning function.

Table 3- 1 Pin Assignment for Connector X1

Connection	Name/Address	Function
1	–	Not connected
2	AI 0 (V)	–
3	AI 0 (I)	–
4	AI 0 (C)	–
5	AI 1 (V)	–
6	AI 1 (I)	–
7	AI 1 (C)	–
8	AI 2 (V)	–
9	AI 2 (I)	–
10	AI 2 (C)	–
11	AI 3 (V)	–
12	AI 3 (I)	–
13	AI 3 (C)	–
14	AI R_P	–
15	AI R_N	–
16	AO 0 (V)	Voltage output of the power section
17	AO 0 (I)	Current output of the power section
18	AO 1 (V)	–
19	AO 1 (I)	–
20	Mana	Analog ground
21	–	Not connected
22	DI + 2.0	–
23	DI + 2.1	–
24	DI + 2.2	–
25	DI + 2.3	–
26	DI + 2.4	–
27	DI + 2.5	–
28	DI + 2.6	–
29	DI + 2.7	–
30	4 M	Chassis ground
V: Voltage input/output I: Current input/output C: Common input		

Table 3- 2 Pin Assignment for Connector X2

Connection	Name/Address	Function
1	1 L+	24 V power supply for the inputs
2	DI + 0.0	Encoder signal A
3	DI + 0.1	Encoder signal B
4	DI + 0.2	Encoder signal N
5	DI + 0.3	Length measurement
6	DI + 0.4	Reference point switch
7	DI + 0.5	–
8	DI + 0.6	–
9	DI + 0.7	–
10	–	Not connected
11	–	Not connected
12	DI + 1.0	–
13	DI + 1.1	–
14	DI + 1.2	–
15	DI + 1.3	–
16	DI + 1.4	–
17	DI + 1.5	–
18	DI + 1.6	–
19	DI + 1.7	–
20	1 M	Chassis ground
21	2 L+	24 V power supply for the outputs
22	DO + 0.0	–
23	DO + 0.1	–
24	DO + 0.2	–
25	DO + 0.3	–
26	DO + 0.4	–
27	DO + 0.5	–
28	DO + 0.6	CONV_EN: Enable power section
29	DO + 0.7	CONV_DIR: Direction signal*
30	2 M	Chassis ground
31	3 L+	24 V power supply for the outputs
32	DO + 1.0	–
33	DO + 1.1	–
34	DO + 1.2	–
35	DO + 1.3	–
36	DO + 1.4	–
37	DO + 1.5	–
38	DO + 1.6	–
39	DO + 1.7	–
40	3 M	Chassis ground
* This output is only used for control mode "Voltage 0 to 10 V or current 0 to 20 mA and direction signal".		

3.1.4 Connecting Components

Procedure

1. Switch off the power supply to all components.
2. Connect the power supply for the inputs and outputs:
   – 24 V at X2, pins 1, 21 and 31
   – Ground at X1, pin 30 and X2, pins 20, 30 and 40
3. Connect the 24 V encoder and switches to the 24 V power supply.
4. Connect the encoder signals and the required switches (X2, pins 2 to 6 and pin 20).

   You can connect bounce-free switches (24 V P-switching) or contactless sensors/BEROs (2 or 3-wire proximity switches) to the digital inputs "Length measurement" and "Reference point switch".
5. Connect the power section to the power supply.
6. Use shielded cables to connect the signal cables of the power section (X1, pin 16 or 17 and pin 20 and X2, pin 28).

   If you are controlling your power section with a voltage of 0 to 10 V (pin 16) or a current of 0 to 20 mA (pin 17) and an additional 24 V digital output for the direction signal, also connect the corresponding power section input with the 24 V digital output CONV_DIR (X2, pin 29).
7. Strip the insulation material on the shielded cables and bind the cable shield to the shield connection element. Use the shield terminal elements for this.

Note

The CPU does not detect the failure of a digital input. You can detect an encoder failure by activating the actual value monitoring (see Drive parameters (Page 32)).

Such a failure might have the following causes:

- Digital input failure
- Wire break
- Faulty encoder
- Faulty power section

3.2 Parameter configuration

3.2.1 Basics of Parameter Configuration

Basics

You can adapt the parameters for the positioning function to your specific application. You can assign the parameters with two parameter types:

- **Module parameters**

  These are basic settings that are specified once and no longer changed while the process is running. The parameters are described in this section.

  – You assign these parameters in the parameter assignment screens (in HW Config).

  – They are stored in the system memory of the CPU.

  – You cannot modify these parameters when the CPU is in RUN mode.

- **SFB parameters**

  Parameters that need to be changed during operation are located in the instance DB of the system function block (SFB). The SFB parameters are described in Section Positioning with Analog Output - Procedure (Page 42).

  – You assign these parameters offline in the DB Editor or online in the user program.

  – They are stored in the work memory of the CPU.

  – You can modify these parameters in the user program while the CPU is in RUN state.

Parameter assignment screens

You can assign the module parameters in the **parameter assignment screens**:

- General
- Addresses
- Basic parameters
- Drive
- Axis
- Encoder
- Diagnostics

The parameter assignment screens are self-explanatory. You can find the description of the parameters in the following sections and in the integrated help in the parameter assignment screens.

Note

You cannot assign parameters for the positioning technology if you have assigned channel 0 or channel 1 for the counting technology.

Note

You can only configure positioning with analog output mode after you have disabled output 0 in submodule AI5/AO2. In this case, you can no longer direct access this output via the user program.

3.2.2 Configuring Parameters Using the Parameter Assignment Screen

Requirements

Prerequisite for calling the parameter assignment screen is that you have created a project in which you can save your parameters.

Procedure

1. Start the SIMATIC Manager and call HW Config in your project.
2. Double-click on the submodule "AI 5/AO 2" of your CPU. Set the output status of analog output AO 0 to "disabled".
3. Double-click on the "Positioning" submodule of your CPU. The "Properties" dialog box opens.
4. Assign the parameters to the "Positioning" submodule and exit the parameter assignment screen with "OK".
5. Save your project in HW Config with "Station > Save and Compile".
6. Download the parameter data to your CPU in STOP mode with "PLC > Download to Module...". The data are now stored in the CPU's system data memory.
7. Switch the CPU to RUN mode.

Online Help

The online help in the parameter assignment screens offers you support when you assign parameters. You have the following options of calling the online help:

- Press the **F1** key in the respective views
- Click on the **Help button** in the various parameter assignment screens.

3.2.3 Basic parameters

Interrupt Selection Parameter

Parameter	Value range	Default
Interrupt selection	• None • Diagnostics	None

Here you can specify whether or not a diagnostic interrupt is to be triggered. The diagnostic interrupt is described in Section Configuring and Evaluating Diagnostic Interrupts (Page 80).

3.2.4 Drive parameters

Target Range Parameter

Parameter	Value range	Default
Target range	0 to 200,000,000 pulses The CPU rounds up odd values.	50

The target range is arranged symmetrically around the target.

When the value is 0, POS_RCD is not set to TRUE until the target has been overrun or reached to the accuracy of a pulse.

The target range is limited:

- to the rotary axis range of rotary axes
- to the working range of linear axes

Monitoring Time Parameter

Parameters	Value range	Default
Monitoring time	• 0 to 100 000 ms • 0 = No monitoring Rounded up by the CPU in 4 ms steps.	2000

The CPU uses this monitoring time to monitor

- actual value of the position
- target approach

Actual value and target approach monitoring is switched off when the value is set to "0".

Maximum Speed Parameter

Parameters	Value range	Default
Maximum speed	10 to 1,000,000 pulses/s	1000

This parameter is used for setting a proportional relationship between the level at the analog output and the speed. The maximum speed specified here is proportional to a level of 10 V or 20 mA at the analog output.

Creep/Reference Speed Parameter

Parameters	Value range	Default
Creep/ reference speed	10 up to the configured maximum speed	100

The speed is reduced to creep speed when the braking position is reached.

The speed is reduced to reference point approach speed when the drive reaches the reference point switch.

Off Delay Parameter

Parameters	Value range	Default
Off delay	0 to 100 000 ms Rounded up by the CPU in 4 ms steps.	1000

Off delay between the cancellation of a run and disabling of the converter (Digital output CONV_EN).

When controlling a brake via the digital output CONV_EN, you can use this delay to ensure that the axis is slow enough to allow the brake to absorb the kinetic energy.

Maximum Frequency Parameter: Position feedback

Parameters	Value range	Default
Max. frequency: Position feedback	• 60 kHz • 30 kHz • 10 kHz • 5 kHz • 2 kHz • 1 kHz	60 kHz

You can set the maximum frequency of the position feedback signals (encoder signals A, B, N) in fixed steps.

Maximum Frequency Parameter: Accompanying signals

Parameters	Value range	Default
Max. frequency: Accompanying signals	• 60 kHz • 30 kHz • 10 kHz • 5 kHz • 2 kHz • 1 kHz	10 kHz

You can set the maximum frequency of the length measurement and reference point switch signals in fixed steps.

Control Mode Parameter

Parameters	Value range	Default
Control mode	• Voltage ±10 V or current ±20 mA • Voltage 0 to 10 V or current 0 to 20 mA and directional signal	Voltage ±10 V or current ±20 mA

The control mode describes how a connected converter is controlled.

- Voltage ±10 V or current ±20 mA:

  A positive voltage or current is output for the run in a plus (forward) direction. A negative voltage or current is output for the run in a minus (backward) direction.

- Voltage 0 to 10 V or a current of 0 to 20 mA and a direction signal:

  A positive voltage or current is output for run in plus (forward) direction and the digital output CONV_DIR is switched off.

  A negative voltage or current is output run in minus (backward) direction and the digital output CONV_DIR is switched off.

Actual Value Parameter

Parameters	Value range	Default
Actual value monitoring	• Yes • No	Yes

The moving axis must cover a distance of least one pulse in specified direction within the monitoring time.

Actual value monitoring is switched on at the start of a run. It remains active until the cut-off position is reached.

Actual value monitoring is switched off when the monitoring time is set to "0".

The run is canceled when the monitoring facility responds.

The CPU does not detect the failure of a digital input. You can enable actual value monitoring for indirect detection of encoder or drive failure.

Target Approach Monitoring Parameter

Parameters	Value range	Default
Target approach monitoring	• Yes • No	No

The axis must reach the target range within the monitoring time after it has reached the cut-off position.

Target approach monitoring is switched off when the monitoring time is set to "0".

Target Range Monitoring Parameter

Parameters	Value range	Default
Target range monitoring	• Yes • No	No

After the target range has been reached, the drive is monitored to check whether it remains at the approached target position or drifts off.

An external error message is generated when the monitoring facility responds. This deactivates the monitoring. Monitoring is not switched on again until the start of a new run.

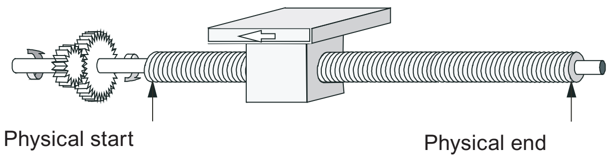
3.2.5 Axis parameters

Axis Type Parameter

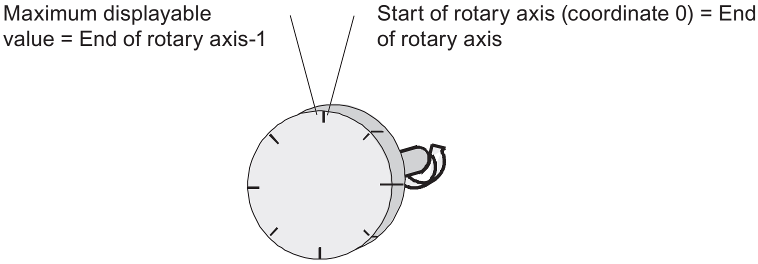
Parameter	Value range	Default
Axis type	• Linear axis • Rotary axis	Linear axis

You can control linear axes as well as rotary axes.

The maximum travel range of a **linear axis** is mechanically limited.



The **rotary axis** is not limited by mechanical stops.



Rotation of the rotary axis starts at the "Zero" coordinate and terminates at the coordinate "End of rotary axis – 1". The "Zero" coordinate is physically identical (= 0) to the "End of rotary axis". The actual position value display is toggled at this point. It is always displayed with a positive value.

Parameters for software limit switch start/end

Parameters	Value range	Default
Software limit switch start/ end	Software limit switch Start Software limit switch End -5×10^8 to $+5 \times 10^8$ pulses	-100 000 000 +100 000 000

Software limit switches are only used for linear axes.

These software limit switches limit the working range.

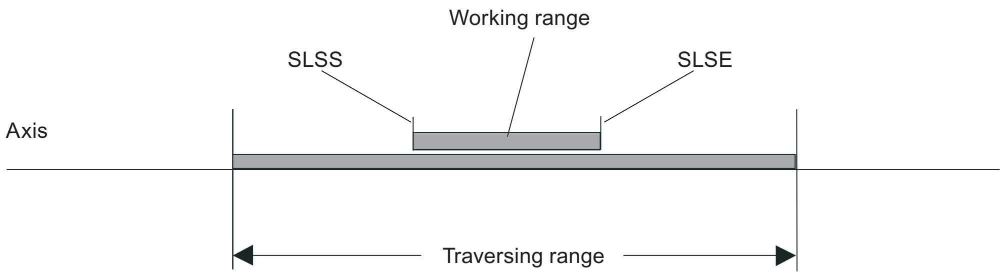
The software limit switches belong to the working range.

The software limit switches are monitored if the axis is synchronized and working range monitoring is switched on.

The axis is not initially synchronized after every STOP-RUN transition of the CPU.

The value of Software Limit Switch Start (SLSS) must always be less than the value of Software Limit Switch End (SLSE).

The working range must lie within the traversing range. This traversing range represents the value range the CPU can process.



SLSS = Software limit switch Start
SLSE = Software limit switch End

End of Rotary Axis Parameter

Parameter	Value range	Default
End of rotary axis	1 to 10^9 pulses	100 000

The value of "End of rotary axis" is theoretically the highest possible actual value. Its physical position is identical to the start of the rotary axis (= "0").

The highest displayed rotary axis value is "End of rotary axis – 1"

Example: End of rotary axis = 1,000

The display toggles:

- with positive rotary direction from 999 to 0
- with negative rotary direction from 0 to 999

Length Measurement and Reference Point Coordinate Parameters

Parameter	Value range	Default
Length measurement	• Off • Start/End at the positive edge DI • Start/End at the negative edge DI • Start with positive edge and end with negative edge • Start with negative edge and end with positive edge	Off
Reference point coordinate	-5×10^8 to $+5 \times 10^8$ pulses	0

After a STOP-RUN transition of the CPU, the actual value is set to the value of the reference point coordinate .

After a reference point approach, the reference point is assigned the value of the reference point coordinate.

The value of the reference point coordinate must lie within the working range (including the software limit switches) of the linear axis.

The value of the reference point coordinate of the rotary axis must lie within the range 0 to "End of rotary axis – 1".

Reference Point Location for Reference Point Switch Parameter

Parameter	Value range	Default
Reference point location for reference point switch	• Plus direction (actual values increase) • Minus direction (actual values decrease)	Plus direction

This parameter defines the reference point position with reference to the reference point switch.

Traversing Range Monitoring Parameter

Parameter	Value range	Default
Traversing range monitoring	Yes (set fixed)	Yes

Use traversing range monitoring to check whether the permitted traversing range of -5×10^8 to $+5 \times 10^8$is exceeded. This monitoring function cannot be switched off (switched on permanently in the "Monitoring" parameter).

Synchronization is canceled and the run is aborted when this monitoring responds.

Working Range Monitoring Parameter

Parameter	Value range	Default
Working range monitoring (only with linear axis)	• Yes • No	Yes

Here, you can specify whether to monitor the working range of the linear axis. In this case, the actual position value is monitored to check whether it is out of range of the software limit switches. This monitoring only affects a synchronized axis.

The coordinates of the software limit switches themselves belong to the working range.

The run is canceled when the monitoring responds.

3.2.6 Encoder parameters

Increments per Encoder Revolution Parameter

Parameter	Value range	Default
Increments per encoder revolution	1 to 2^{23} pulses	1000

The "Increments per encoder revolution" parameter specifies the increments per revolution output at the encoder. Refer to the description of your encoder for information on values.

The CPU evaluates the increments four times (one increment corresponds to four pulses, see Incremental encoders (Page 82)).

Count Direction Parameter

Parameter	Value range	Default
Count direction	• Normal • Inverted	Normal

Use the "Count direction" parameter to adapt the direction of path monitoring to the direction of movement of the linear axis. Also, take the rotary direction of all transmission elements into account (for example, couplings and gears).

- Normal = incrementing count pulses = ascending actual values
- Inverted = incrementing count pulses = descending actual values

Missing Pulse (Zero Mark) Monitoring Parameter

Parameter	Value range	Default
Missing pulse (zero mark) monitoring	• Yes • No	No

When zero mark monitoring is enabled, the CPU monitors consistency of the pulse difference between two successive zero mark signals (encoder signal N).

If you have configured an encoder whose pulses per revolution cannot be divided by 10 or 16, zero mark monitoring is automatically switched off, irrespective of the setting in the parameter assignment screen.

Note

The minimum pulse width of the zero mark signal is 8.33 μs (corresponds to the maximum frequency of 60 kHz).

If you are using an encoder whose zero mark signal is combined with encoder signals A and B using an "AND" operation, the pulse width is reduced by half to 25% of the period. This reduces the maximum frequency for zero mark monitoring to 30 kHz.

Not recognized is:

- Incorrectly assigned number of increments per encoder revolution.
- Failure of the zero mark signal.

Synchronization is canceled and the run is aborted when this monitoring responds.

3.2.7 Configuring the Diagnostics

Enabling diagnostic interrupt for monitoring

The responding monitoring feature can trigger a diagnostic interrupt.

Requirement: You have enabled the diagnostic interrupt in the "Basic Parameters" screen and activated the respective monitoring in the "Drive", "Axis" and "Encoder" screens.

Parameter	Value range	Default
Missing pulse (zero mark)	• Yes • No	No
Traversing range	• Yes • No	No
Working range (for linear axes)	• Yes • No	No
Actual value	• Yes • No	No
Target approach	• Yes • No	No
Target range	• Yes • No	No

3.3 Integration into the user program

Procedure

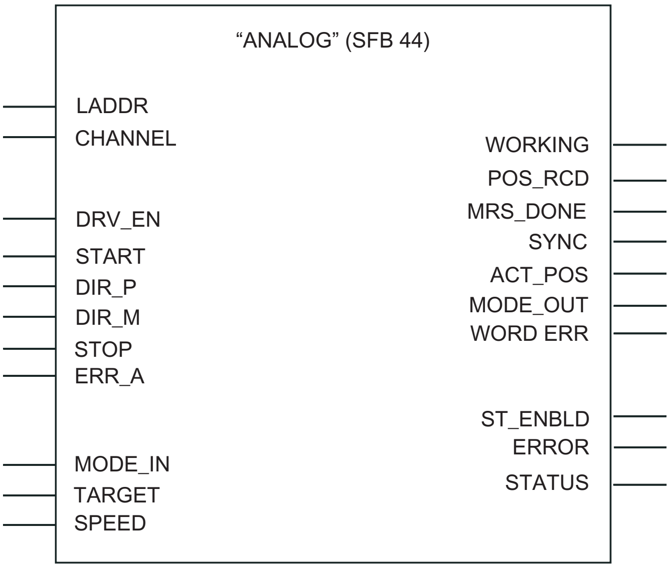
The positioning functions are controlled in your user program. To do this, call the system function block SFB ANALOG (SFB 44) . The SFB is found in the "Standard Library" under "System Function Blocks > Blocks".

The following sections help you to design a user program for your application.

Calling the SFB

Call the SFB with a corresponding instance DB.

Example: CALL SFB 44, DB20



Note

You must not call an SFB you have configured in your program in another program section under another priority class, because the SFB must not interrupt itself.

Example: It is not allowed to call the same SFB both in OB1 and in the interrupt OB.

Instance DB

The SFB parameters are stored in the instance DB. These parameter are described in Section Basic Configuration of the SFB ANALOG (SFB 44) (Page 48). You can access these parameters via

- DB number and absolute address in the DB
- DB number and symbolic address in the DB

The main parameters for the function are also interconnected to the block. You can assign the input parameters values directly at the SFB or you can evaluate the output parameters.

3.4 Functions for Positioning with Analog Output

3.4.1 Positioning with Analog Output - Procedure

Overview

A permanently assigned analog output (analog output 0) controls the drive with a voltage (voltage signal) of between ±10 V or 0 to 10 V with an additional digital output CONV_DIR or with a current (current signal) of ±20 mA or 0 to 20 mA with additional digital output CONV_DIR.

Position feedback is realized via an asymmetric 24 V incremental encoder that is equipped with two signals with a 90° phase shift.

Digital output CONV_EN is used to enable and switch off the power section and/or to control a brake.

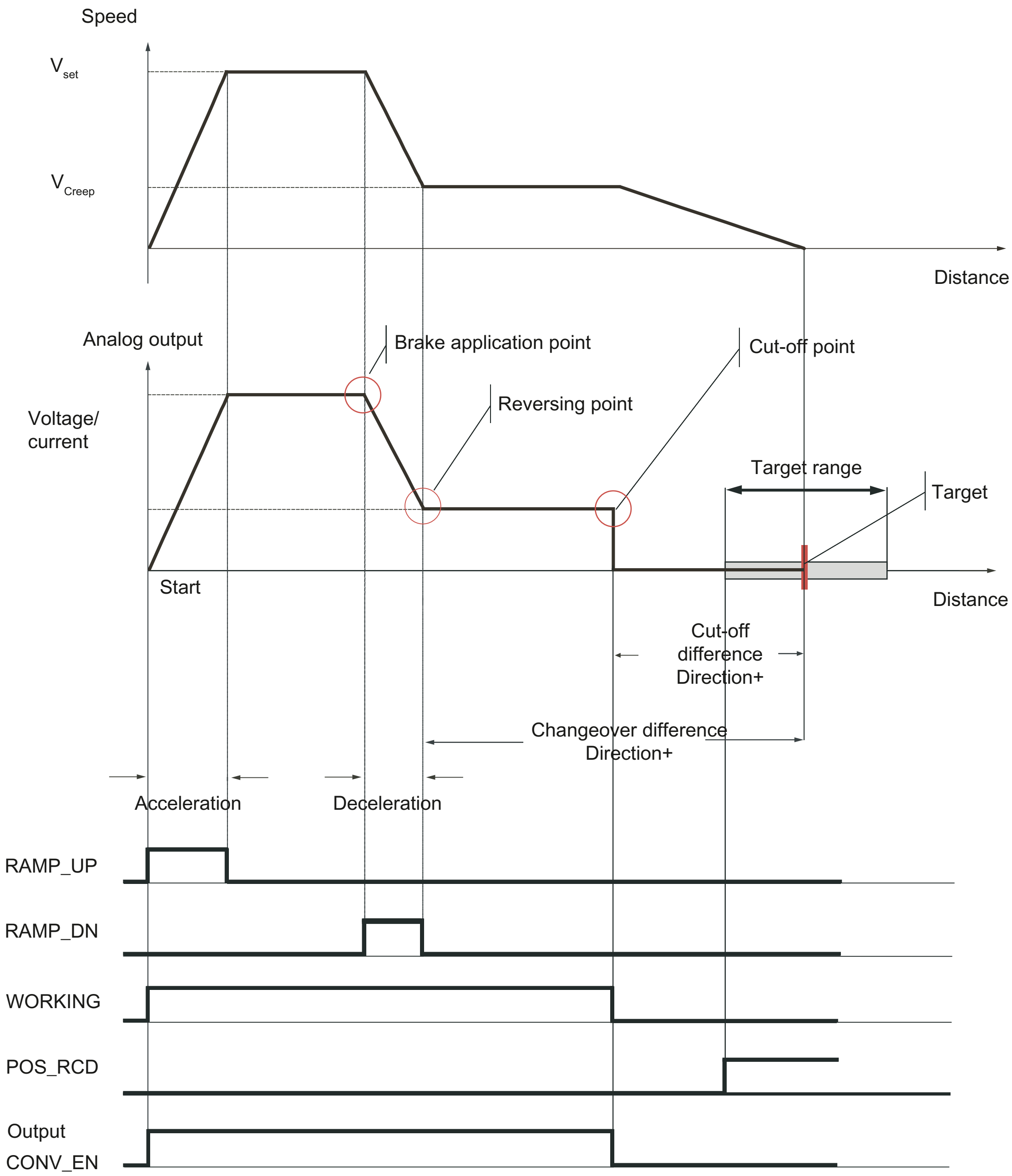
Starting a Run

Start the run with START, DIR_P or DIR_M, depending on the operating mode.

Positioning with Analog Output

The upper section of the figure below shows the run profile. We will simply assume a linear change of the actual speed across the traversing distance.

The lower section of the view shows the corresponding voltage/current profile at the analog output.



- Following the ramp-up phase (RAMP_UP), the target is initially approached with the speed V_{set}.
- At the **braking point** calculated by the CPU, a ramp-down (RAMP_DN) phase is activated up the reversing point.
- Immediately after the **reversing point** is reached, the run is continued at creep speed (V_{creep}).
- The drive is switched off at the **cut-off point**.
- The changeover point and cut-off point for every target to be approached is determined by the values of the **changeover difference** and **cut-off difference** you have specified in your parameters. The changeover difference and cut-off difference can be specified differently for forward (Plus direction) and reverse (Minus direction) run.
- The run is terminated (WORKING= FALSE) when the cut-off point is reached. At this point of time a new run can be started.
- The specified target is reached (POS_RCD = TRUE) when the actual position value has reached the **target area**. The "Position reached" signal is not reset if the actual position value drifts off the target area without a new run having been started.
- If the changeover difference is less than the cut-off difference, the speed is reduced along a linear ramp until the speed setpoint 0, starting at the braking point.

Enable power section (CONV_EN)

The digital output CONV_EN is used to enable and disable the power section or to control a brake. The output is set at the start and reset at the end of a run (at the cut-off point or when speed setpoint = 0).

When using the digital output to control a brake, you must take into account that the brake must absorb the kinetic energy of the drive at the time the output is reset (at the cut-off point or when the speed setpoint = 0).

Direction Signal (CONV_DIR)

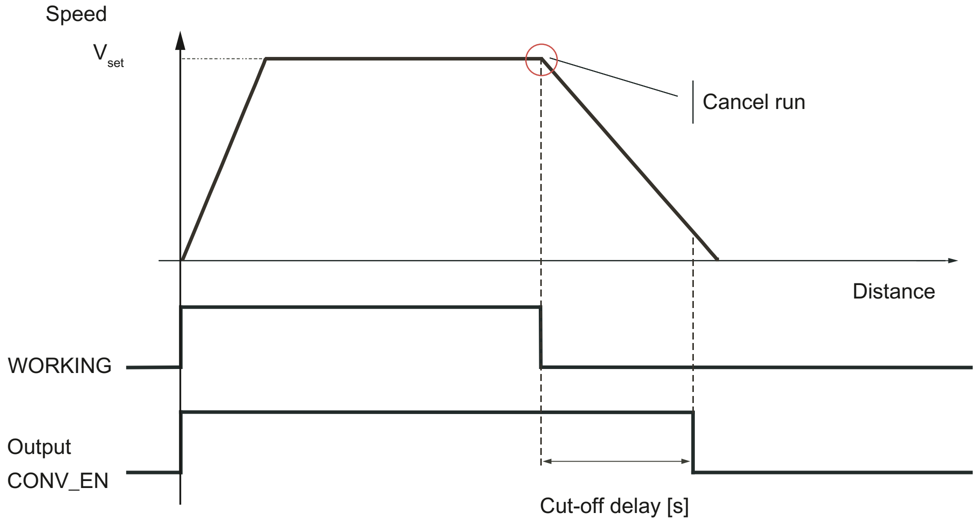
In control mode "Voltage 0 to 10 V or current 0 to 20 mA and direction signal", the 24 V digital output CONV_DIR supplies direction information.

- Run in plus direction (forward): DO switched off
- Run in minus direction (reversed): DO switched on

Off Delay when Canceling a Run

In the parameter assignment screens in the "Off delay" parameter, you can configure the delay time (only active when a run is aborted) to be active between the cancellation of a run and reset of the digital output CONV_EN.

This ensures that the axis is slow enough at the time the output is reset to enable the brake to absorb the kinetic energy of the axis.

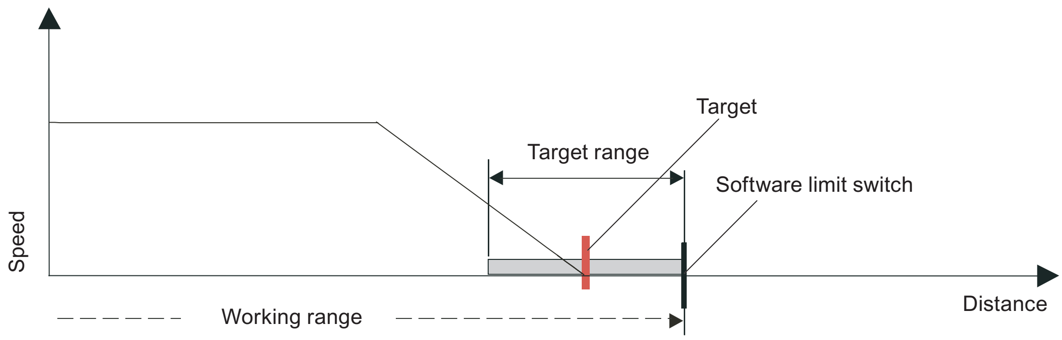


Working range

Determine the working range with the help of the software limit switch coordinates. A run may never exceed the working range of a synchronized linear axis.

You must always specify the run targets according to the working range.

After an axis has overrun the working range, you can only return it in jog mode.



Monitoring functions

The parameter assignment screens help you to enable various monitoring functions individually. When one of the monitoring functions responds, the run is canceled with external error (acknowledge with ERR_A).

Monitoring	Description
Missing pulse (zero mark)	When zero mark monitoring is enabled, the CPU checks the consistency of the pulse difference between two successive zero mark signals. If you have configured an encoder whose pulses per revolution cannot be divided by 10 or 16, zero mark monitoring is automatically switched off, irrespective of the setting in the parameter assignment screen. The minimum pulse width of the zero mark signal is 8.33 μs (corresponds to the maximum frequency of 60 kHz). If you are using an encoder whose zero mark signal is combined with encoder signals A and B using an "AND" operation, the pulse width is reduced by half to 25% of the period. This reduces the frequency for missing pulse monitoring to a maximum of 30 kHz. Not recognized is: • Incorrectly assigned number of increments per encoder revolution. • Failure of the zero mark signal. Response of the CPU to errors: Cancel synchronization, cancel the run.
Traversing range	The CPU uses traversing range monitoring to check whether the permitted traversing range of -5×10^8 to $+5 \times 10^8$ is exceeded. This monitoring function cannot be switched off (switched on permanently in the "Monitoring" parameter). Response of the CPU to errors: Cancel synchronization, cancel the run.
Working range	The CPU uses traversing range monitoring to check whether the actual value is out of range of the software limit switches. This facility cannot be switched on for monitoring rotary axis positioning. This monitoring only affects a synchronized axis. The coordinates of the software limit switches themselves belong to the working range. Response of the CPU to errors: The run is canceled.
Actual value	The moving axis must cover a distance of least one pulse in specified direction within the monitoring time. Actual value monitoring is switched on at the start of a run. It remains active until the cut-off point is reached. Actual value monitoring is switched off when the monitoring time is set to "0". The run is canceled when the monitoring responds. Response of the CPU to errors: The run is canceled.
Target approach	After it has reached cut-off point the axis must reach the target range within the monitoring time. Target approach monitoring is switched off when the monitoring time is set to "0". Response of the CPU to errors: The run is canceled.
Target range	After the target range has been reached, the CPU monitors the drive to check whether it stays at or drifts off the approached target position. An external error message is generated when the monitoring responds. Monitoring is switched off when you acknowledge the external error ERR_A (positive acknowledgment). Monitoring is not switched on again until the start of a new run. Response of the CPU to errors: The run is canceled.

Terminating a Run

There are three different ways to terminate a run:

- Target approach
- Deactivating
- Canceling

Target approach:

Target approach stands for automatic run termination when the specified target is reached.

In order to reach a specified target, target approach is carried out in the operating modes "Relative and absolute incremental approach".

Deactivating:

The drive is deactivated in the following cases:

- In all operating modes when STOP = TRUE (before the target is reached)
- In "Jog mode", when stopping and when run direction is reversed
- In "Reference point approach" mode, when the synchronization position is detected or at direction reversal

When deactivating, the speed is reduced over a linear ramp to the speed setpoint 0.

Canceling:

The run is terminated immediately, disregarding the changeover/cut-off difference. The analog output is switched directly to speed setpoint 0.

Cancellation is possible at any time or at standstill.

The run is canceled in the following cases:

- Canceling the drive enable signal (DRV_EN = FALSE)
- When the CPU switches to STOP mode
- When an external error occurs (Exception: monitoring of the target approach / target range)

Results of a cancellation:

- A current or stopped run is terminated immediately (WORKING = FALSE).
- The last target (LAST_TRG) is set to the actual value (ACT_POS).
- A distance-to-go is deleted, that is, "Relative incremental approach" can not be resumed.
- "Position reached" (POS_RCD) will not be set.
- The digital output CONV_EN (Enable power section) is reset with off delay.

3.4.2 Basic Configuration of the SFB ANALOG (SFB 44)

Overview of Basic Parameters:

The parameters which are identical for all operating modes are described in this section. Operating mode-specific parameters are described under the specific modes.

Assign the following SFB input parameters according to your application.

Input parameters

Parameter	Data type	Address (instance DB)	Description	Value range	Default
LADDR	WORD	0	Submodule I/O address you specified in "HW Config". If the input and output addresses are not the same, you must specify the lower one of both.	CPU-specific	310 hex
CHANNEL	INT	2	Channel number	0	0
STOP	BOOL	4.4	Stops the run STOP = TRUE can be used to stop/interrupt a run prematurely.	TRUE/FALSE	FALSE
ERR_A	BOOL	4.5	Group error acknowledgment ERR_A is used to acknowledge external errors (positive edge)	TRUE/FALSE	FALSE
SPEED	DINT	12	The axis is accelerated to $V_{setpoint}$. Speed change during run is not possible.	Creep speed of up to 1,000,000 pulses/s Up to the maximum speed	1000

Input parameters not assigned to the block (static local data)

Parameter	Data type	Address (instance DB)	Description	Value range	Default
ACCEL	DINT	30	Acceleration Change during run not possible.	1 to 100 000 Pulses/s^2	100
DECEL	DINT	34	Deceleration Change during run not possible.	1 to 100 000 Pulses/s^2	100
CHGDIFF_P	DINT	38	Changeover difference plus: "Changeover difference plus" defines the changeover point from which the drive continues its forward run with creep speed.	0 to $+10^8$ Pulses	1000
CUTOFF-DIFF_P	DINT	42	Cut-off difference plus: The "Cut-off difference plus" defines the cut-off point at which the drive is switched off when operating at creep speed in positive direction.	0 to $+10^8$ Pulses	100
CHGDIFF_M	DINT	46	Changeover difference minus: "Changeover difference minus" defines the changeover point from which the drive continues with a reverse run at creep speed.	0 to $+10^8$ Pulses	1000
CUTOFF-DIFF_M	DINT	50	Cut-off difference minus: The "Cut-off difference plus" defines the cut-off point at which the drive is switched off when operating at creep speed in negative direction.	0 to $+10^8$ Pulses	100

Rules for the Changeover/Cut-Off Difference

- Positive and negative values can differ.
- When the changeover difference is less than the cut-off difference, the drive is slowed down to speed setpoint 0 across a linear ramp, starting at the braking position.
- The cut-off difference must be greater or equal to half the target range.
- The changeover difference must be greater or equal to half the target range.
- Select a sufficient distance between the reversing point and the cut-off point to ensure that the drive's speed can be reduced to creep speed.
- Select a sufficient distance between the reversing point and the target to ensure the drive reaches the target area and comes to a standstill there.
- The distance to be traveled must be at least as high as the cut-off difference
- Changeover/cut-off difference are limited to 1/10 of the traversing range ($+10^8$).

Output parameters

Parameter	Data type	Address (instance DB)	Description	Value range	Default
WORKING	BOOL	16.0	Run is busy	TRUE/FALSE	FALSE
ACT_POS	DINT	18	Current actual position value	-5 x 10^8 to +5 x 10^8 pulses	0
MODE_OUT	INT	22	Active/set operating mode	0, 1, 3, 4, 5	0
ERR	WORD	24	External error: • Bit 2: missing pulse monitoring • Bit 11: traversing range monitoring (always 1) • Bit 12: Working range monitoring • Bit 13: actual value monitoring • Bit 14: target approach monitoring • Bit 15: target range monitoring • The other bits are reserved	Every bit 0 or 1	0
ST_ENBLD	BOOL	26.0	The CPU sets Start Enabled if all of the following conditions are met: • Faultless parameter assignment (PARA = TRUE) • No STOP pending (STOP = FALSE) • No external error has occurred (ERR = 0) • Drive Enable is set (DRV_EN = TRUE) • No positioning run active (WORKING = FALSE) Exception: Jog mode	TRUE/FALSE	TRUE
ERROR	BOOL	26.1	Run start/resume error	TRUE/FALSE	FALSE
STATUS	WORD	28	Error ID	0 to FFFF hex	0

Output parameters not assigned to the block (static local data)

Parameter	Data type	Address (instance DB)	Description	Value range	Default
PARA	BOOL	54.0	Axis is configured	TRUE/FALSE	FALSE
DIR	BOOL	54.1	Current/last sense of direction FALSE = Forward (plus direction) TRUE = Reverse (minus direction)	TRUE/FALSE	FALSE
CUTOFF	BOOL	54.2	Drive in cut-off range (from cut-off point to the start of the next run)	TRUE/FALSE	FALSE
CHGOVER	BOOL	54.3	Drive in changeover range (after reaching the reversing point up to the start of the next run)	TRUE/FALSE	FALSE
RAMP_DN	BOOL	54.4	Drive is ramped down (starting at the braking point, up to the reversing point)	TRUE/FALSE	FALSE
RAMP_UP	BOOL	54.5	Drive is ramped up (from start to reaching end speed)	TRUE/FALSE	FALSE
DIST_TO_GO	DINT	56	Actual distance to go	-5 x 10^8 to +5 x 10^8 pulses	0
LAST_TRG	DINT	60	Last/actual target • Absolute incremental approach: At run start LAST_TRG = actual absolute target (TARGET) • Relative incremental approach: The distance at run start is the distance specified in LAST_TRG = LAST_TRG of the previous run +/- (TARGET)	-5 x 10^8 to +5 x 10^8pulses	0

3.4.3 Jog Mode

Description

In "Jog" mode you can run the drive into plus or minus direction. A target is not specified.

Requirements

- You have assigned the module parameters via parameter assignment screens and downloaded them to the CPU (PARA = TRUE).
- You have assigned the basic parameters of the SFB as described.
- No external error ERR has occurred. You must acknowledge queued external errors with ERR_A (positive edge).
- Start is enabled (ST_ENBLD = TRUE).
- You can operate both a synchronized (SYNC = TRUE) and a non-synchronized (SYNC = FALSE) axis in jog mode.

Starting/Stopping the Run

You start the drive by setting control bit DIR_P or DIR_M.

- At every SFB call, these two control bits DIR_P and DIR_M are evaluated to check for logical level changes.
- If both control bits are FALSE the run is ramped down.
- If both control bits are TRUE the run is also ramped down.
- The axis moves in the corresponding direction when one of the control bits is set to TRUE.

Procedure

1. Configure the following SFB **input parameters** as specified in the "Setting" column:

Parameter	Data type	Address (instance DB)	Description	Value range	Default	Setting
DRV_EN	BOOL	4.0	Drive enable	TRUE/FALSE	FALSE	TRUE
DIR_P	BOOL	4.2	Jog mode, into plus direction (positive edge)	TRUE/FALSE	FALSE	DIR_P or DIR_M = TRUE
DIR_M	BOOL	4.3	Jog mode, into minus direction (positive edge)	TRUE/FALSE	FALSE	
MODE_IN	INT	6	Operating mode, 1 = Jog mode	0, 1, 3, 4, 5	1	1

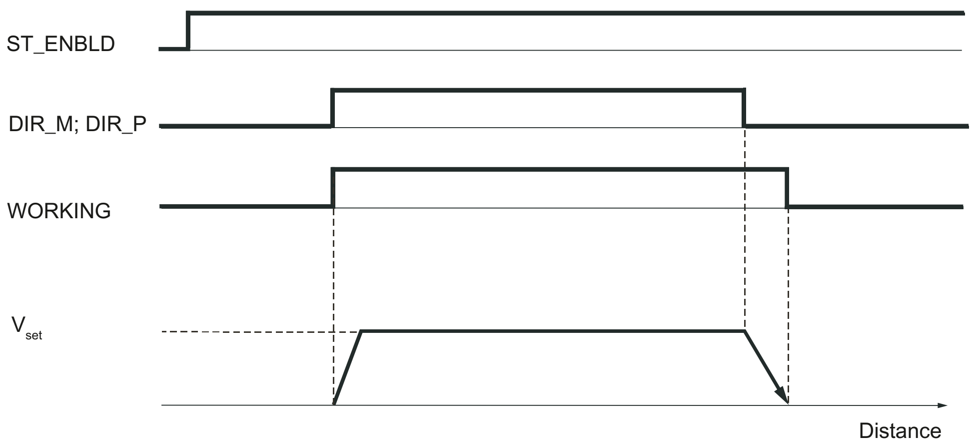
2. Call the SFB.

Result

The **output parameters** of the SFB provide the following information:

Parameters	Data type	Address (instance DB)	Description	Value range	Default
WORKING	BOOL	16.0	Run is busy	TRUE/FALSE	FALSE
ACT_POS	DINT	18	Current actual position value	-5×10^8 to $+5 \times 10^8$ pulses	0
MODE_OUT	INT	22	Active/set operating mode	0, 1, 3, 4, 5	0

- WORKING = TRUE is set immediately after the run has started. When you reset the direction bits DIR_P or DIR_M or set STOP = TRUE, the run is terminated (WORKING = FALSE).
- If an error occurred when the SFB call was interpreted, WORKING = FALSE and ERROR is set to TRUE. The precise error cause is then indicated with the STATUS parameter (see Error Lists (Page 84)).
- In jog mode, ST_ENBLD is always set to TRUE.
- "Position reached" (POS_RCD) will not be set.



3.4.4 Reference Point Approach

Description

After the CPU is switched on position value ACT_POS is not referenced to the mechanical position of the axis.

In order to assign a reproducible encoder value to the physical position, a reference (synchronization) must be established between the axis position and the encoder value. The synchronization is carried out by assigning a position value to a known position (reference point) of the axis.

Reference Point Switch and Reference Point

You require a reference point switch and a reference point at the axis to be able to perform a reference point approach.

- You require the **reference point switch** to ensure that the reference signal always has the same reference point (zero mark) and for changing over to reference point approach speed. You can use a BERO switch, for example. The signal length of the reference point switch must be high enough for the axis to reach reference point approach speed before it moves out of range of the reference point switch.
- **Reference point** is the next encoder zero mark after leaving the reference point switch. The axis is synchronized at the reference point and the feedback signal SYNC is set to TRUE. The reference point is assigned the reference coordinates you have specified via parameter assignment screens.

A reference point approach must always be started in the direction of the reference point switch. Otherwise, the axis travels into range limits because it is not synchronized and, thus, software limit switches do not exist.

By starting the reference point approach at the reference point switch you always ensure that the axis is starts into the direction of the switch (see Example 3).

Note

For rotary axes: Because of the required reproducibility of the reference point, the corresponding zero mark of the encoder must always be at the same physical position. Therefore, the "End of rotary axis" value and the number of "Increments per encoder revolution" must represent a proportional integral. Example: Four encoder revolutions are proportional to one revolution of the end of the rotary axis. In this case, the zero marks lie at 90, 180, 270 and 360 degrees.

The minimum pulse width of the zero mark signal is 8.33 μs (corresponds to the maximum frequency of 60 kHz).

When you are using an encoder whose zero mark signal is combined with encoder signals A and B using an "AND" operation, the pulse width is reduced by half to 25% of the period. This reduces the maximum counting frequency to 30 kHz during referencing.

Reference Point Position

With regard to the reference point position (**Zero mark signal**) during a reference point approach, you must distinguish between:

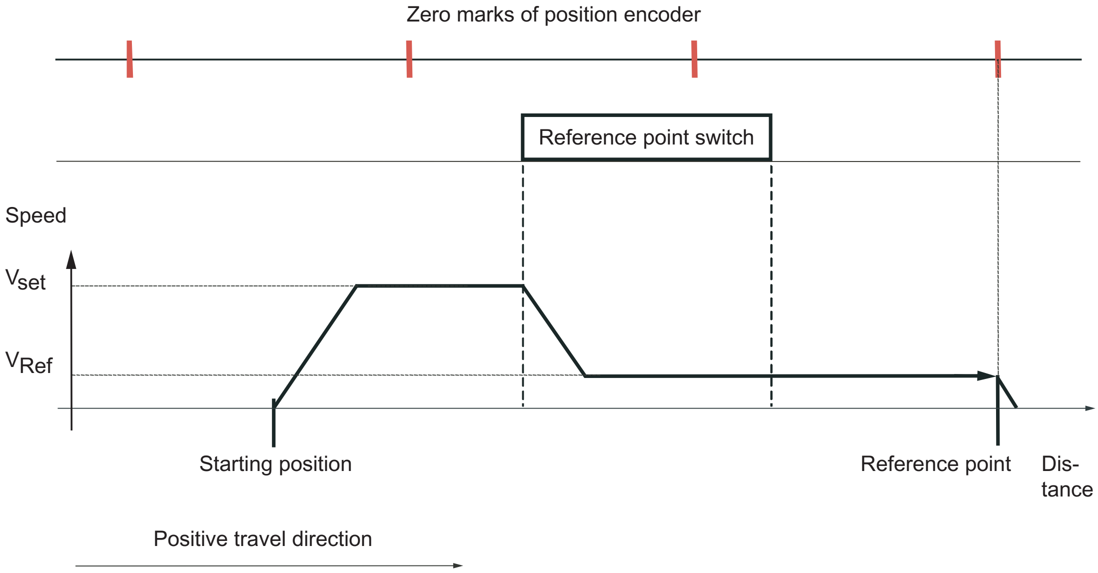
- The reference point position is referenced to the reference point switch in plus direction.
- The reference point position is referenced to the reference point switch in minus direction.

Make this setting via parameter assignment screens in the parameter "Reference point location for reference point switch".

The different situations for a reference point approach are determined by the direction of the run start and by the position of the reference point:

Example 1

- Start direction plus
- Reference point approach to reference point switch in plus direction



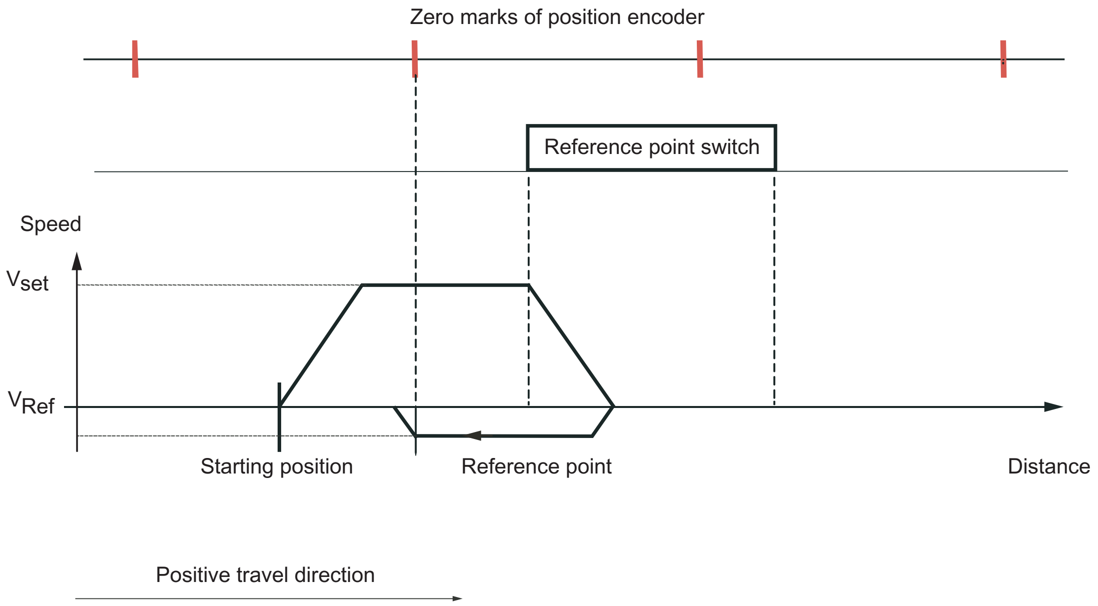
The run towards the reference point switch is executed at speed v_{set} specified in the SPEED parameter.

The speed is then reduced to reference run speed v_{Ref}.

After the reference point switch is exited, the speed is switched to zero at the next zero mark of the encoder.

Example 2

- Start direction plus
- Reference point approach to reference point switch in minus direction



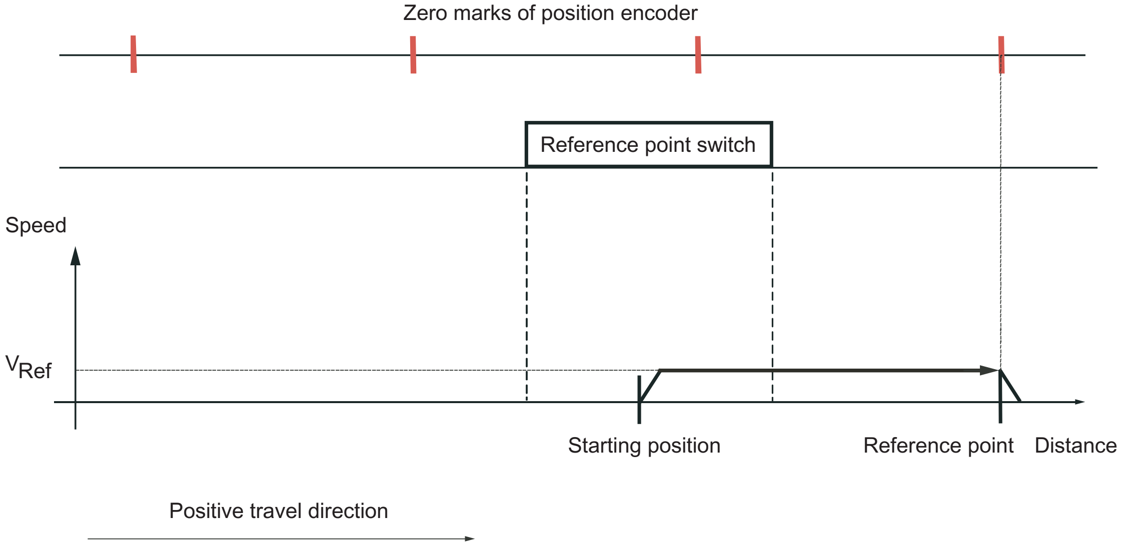
The run towards the reference point switch is executed at speed v_{set} specified in the SPEED parameter.

The speed is reduced to zero, the direction reversed, and motion resumed at reference speed v_{Ref}.

After the reference point switch is exited, the speed is switched to zero at the next zero mark of the encoder.

Example 3

- Start position is at the reference point switch
- Start direction minus
- Reference point approach to reference point switch in plus direction



Run is executed at reference run speed, V_{Ref}.

Run is executed towards the direction you have assigned in the parameter assignment screens with the "Reference point location for reference point switch" parameter, irrespective of the direction specified at the SFB.

After the reference point switch is exited, the speed is switched to zero at the next zero mark of the encoder.

3.4.5 Reference Point Approach – Procedure

Prerequisite for Reference Point Approach

- Encoder with zero mark or, when using an encoder without zero mark, a switch for the reference point signal.
- You have connected the reference point switch (connector X2, pin 6).
- You have assigned the module parameters via parameter assignment screens and downloaded them to the CPU (PARA = TRUE).
- You have assigned the basic parameters of the SFB as described.
- No external error ERR has occurred. You must acknowledge queued external errors with ERR_A (positive edge).
- Start is enabled (ST_ENBLD = TRUE).

Procedure

1. Assign the following **input parameters** of the SFB as specified in the "Setting" column:

Parameters	Data type	Address (instance DB)	Description	Value range	Default	Setting
DRV_EN	BOOL	4.0	Drive enable	TRUE/FALSE	FALSE	TRUE
DIR_P	BOOL	4.2	Reference point approach, plus direction (positive edge)	TRUE/FALSE	FALSE	DIR_P or DIR_M = TRUE
DIR_M	BOOL	4.3	Reference point approach, minus direction (positive edge)	TRUE/FALSE	FALSE	
MODE_IN	INT	6	Operating mode, 3 = reference point approach	0, 1, 3, 4, 5	1	3

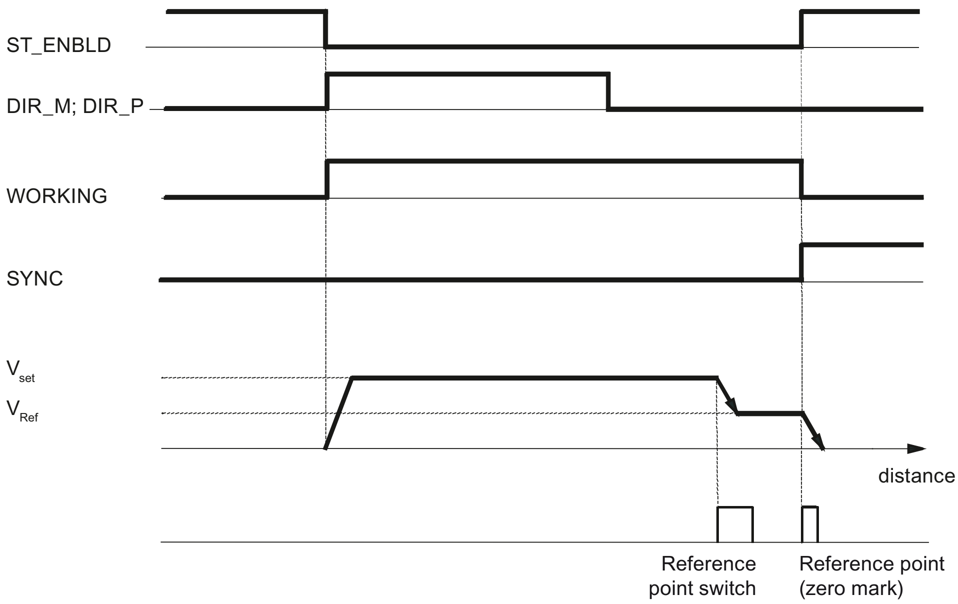
2. Call the SFB.

Result

The **output parameters** of the SFB provide the following information:

Parameters	Data type	Address (instance DB)	Description	Value range	Default
WORKING	BOOL	16.0	Run is busy	TRUE/FALSE	FALSE
SYNC	BOOL	16.3	SYNC = TRUE: Axis is synchronized	TRUE/FALSE	FALSE
ACT_POS	DINT	18	Actual position value	-5×10^8 to $+5 \times 10^8$pulses	0
MODE_OUT	INT	22	Enabled/set operating mode	0, 1, 3, 4, 5	0

- WORKING = TRUE is set and SYNC = FALSE immediately after the run has started. The status of WORKING is reset to FALSE after the reference point has been reached. SYNC = TRUE if executed without error.
- You must reset the direction bit (DIR_P or DIR_M) before you start the next run.
- If an error occurred when the SFB call was interpreted, WORKING = FALSE and ERROR is set to TRUE. The precise error cause is then indicated with the STATUS parameter.
- "Position reached" (POS_RCD) will not be set.



What the Operating Mode Affects

- A possibly existing synchronization is cleared (SYNC = FALSE) at the start of the reference point approach.
- At the positive edge of the reference point (zero mark) the actual position is set to the value of the reference point coordinate and the feedback signal SYNC is set.
- The working range is determined at the axis.
- All points within the working range maintain their original coordinates, but have new physical positions.

3.4.6 Relative incremental approach mode

Description

In "Relative incremental approach" mode, the drive moves into a specified direction by a relative distance, starting at the last target (LAST_TRG).

Starting point is not the actual position, but rather the last specified target (LAST_TRG). As a result, the positioning accuracies do not add up. The actual target is indicated at the parameter LAST_TRG after positioning has started.

Requirements

- You have assigned the module parameters via parameter assignment screens and downloaded them to the CPU (PARA = TRUE).
- You have assigned the basic parameters of the SFB as described in Section Basic Configuration of the SFB ANALOG (SFB 44) (Page 48).
- No external error ERR has occurred. You must acknowledge queued external errors with ERR_A (positive edge).
- Start is enabled (ST_ENBLD = TRUE).
- A "Relative incremental approach" is possible with a synchronized (SYNC = TRUE) and also with a non-synchronized (SYNC = FALSE) axis.

Specifying the Traversing Distance

When specifying the traversing distance of linear axes, note the following:

- The traversing distance must be greater than or equal to the cut-off difference.
- A new run is not started if the traversing distance is less than/equal to half of the target range. This mode is terminated immediately without an error.
- The target range must lie within the working range.

Procedure

1. Assign the following **input parameters** of the SFB as specified in the "Setting" column:

Parameters	Data type	Address (instance DB)	Description	Value range	Default	Setting
DRV_EN	BOOL	4.0	Drive enable	TRUE/FALSE	FALSE	TRUE
DIR_P	BOOL	4.2	Run in plus direction (positive edge)	TRUE/FALSE	FALSE	DIR_P or DIR_M = TRUE
DIR_M	BOOL	4.3	Run in minus direction (positive edge)	TRUE/FALSE	FALSE	
MODE_IN	INT	6	Operating mode, 4 = Relative incremental approach	0, 1, 3, 4, 5	1	4
TARGET	DINT	8	Distance in pulses (only positive values allowed)	0 to 10^9 Pulses	1000	xxxx

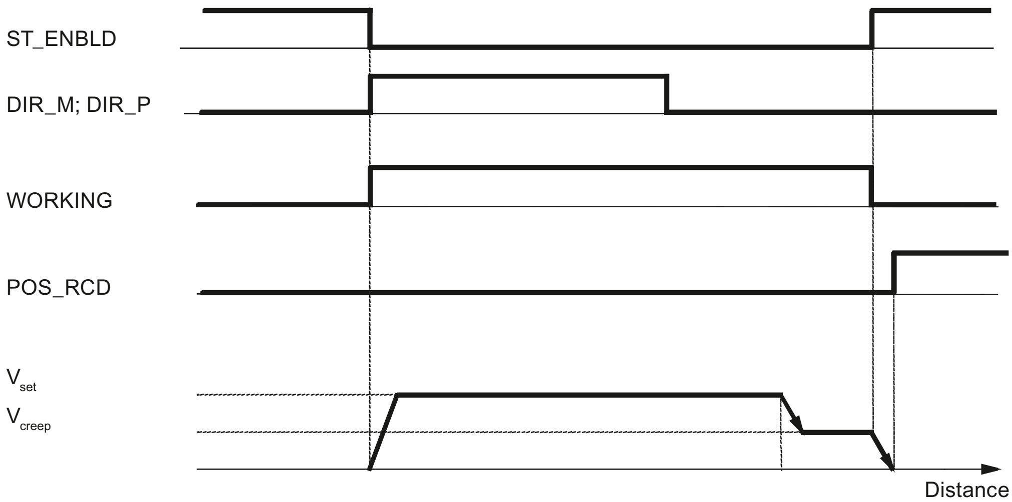
2. Call the SFB.

Result

The **output parameters** of the SFB provide the following information:

Parameters	Data type	Address (instance DB)	Description	Value range	Default
WORKING	BOOL	16.0	Run is busy	TRUE/FALSE	FALSE
POS_RCD	BOOL	16.1	Position reached	TRUE/FALSE	FALSE
ACT_POS	DINT	18	Actual position value	-5×10^8 to $+5 \times 10^8$ pulses	0
MODE_OUT	INT	22	Enabled/set operating mode	0, 1, 3, 4, 5	0

- WORKING = TRUE is set immediately after the run has started. WORKING is reset to FALSE at the cut-off point. POS_RCD is set to TRUE when the specified target is reached.
- You must reset the direction bit (DIR_P or DIR_M) before you start the next run.
- If an error occurred when the SFB call was interpreted, WORKING = FALSE and ERROR is set to TRUE. The precise error cause is then indicated with the STATUS parameter (see Section Error Lists (Page 84)).



Interrupting a Run and Target Range not Reached

When a run is stopped with STOP = TRUE and if the cut-off range has not been reached (distance to go is greater than the cut-off difference), you have the following options depending on the subsequent operating mode/job.

Option	Response
Continuing the run into the same direction	Run parameters will not be interpreted. The axis travels to the target point of the stopped run (LAST_TRG).
Continuing the run in opposite direction	Run parameters will not be interpreted. The axis moves to the starting point of the stopped run.
Starting a new "Absolute incremental approach"	The axis moves to the specified absolute target.
Job "Delete distance to go"	The distance to go (difference between target and actual value) will be deleted. The run parameters are interpreted again at the start of a subsequent "Relative incremental approach" and the axis moves to the current actual position value.

3.4.7 Absolute incremental approach mode

Description

In "Absolute incremental approach" mode you approach absolute target positions.

Requirements

- You have assigned the module parameters via parameter assignment screens and downloaded them to the CPU (PARA = TRUE).
- You have assigned the basic parameters of the SFB as described in Section Basic Configuration of the SFB ANALOG (SFB 44) (Page 48).
- No external error ERR has occurred. You must acknowledge queued external errors with ERR_A (positive edge).
- Start is enabled (ST_ENBLD = TRUE).
- The axis is synchronized (SYNC = TRUE).

Specifying the Target

Note the following when you specify the target:

- The traversing distance must be greater than or equal to the cut-off difference.
- A new run is not started if the traversing distance is less than/equal to half of the target range. This mode is terminated instantaneously and without error.
- The target range for a linear axis must lie within the working range, for a rotary axis it must lie between "0" and "End of rotary axis" -1.

Run Start

- The linear axis is always started with START = TRUE.
- You must specify the sense of direction of rotary axes:

  DIR_P = TRUE: Run in plus direction

  DIR_M = TRUE: Run in minus direction

  START = TRUE: The axis approaches the target along the shortest possible distance.

  The CPU calculates the sense of direction, taking into account the actual distance to go, the actual value and the target.

  Run is started in reverse direction, if the shortest distance is less than/equal to the cut-off difference and greater than/equal to half of the target range.

  If the travel difference is equal in both directions the axis moves in the plus direction.

Procedure

1. Assign the following **input parameters** of the SFB as specified in the "Setting" column:

Parameters	Data type	Address (instance DB)	Description	Value range	Default	Setting
DRV_EN	BOOL	4.0	Drive enable	TRUE/FALSE	FALSE	TRUE
START	BOOL	4.1	Run start (positive edge)	TRUE/FALSE	FALSE	START or DIR_P or DIR_M = TRUE
DIR_P	BOOL	4.2	Run in plus direction (positive edge)	TRUE/FALSE	FALSE	
DIR_M	BOOL	4.3	Run in minus direction (positive edge)	TRUE/FALSE	FALSE	
MODE_IN	INT	6	Operating mode, 5 = absolute incremental approach	0, 1, 3, 4, 5	1	5
TARGET	DINT	8	Target in pulses	Linear axis: -5×10^8 to $+5 \times 10^8$ Rotary axis: 0 to end of rotary axis - 1	1000	xxxx

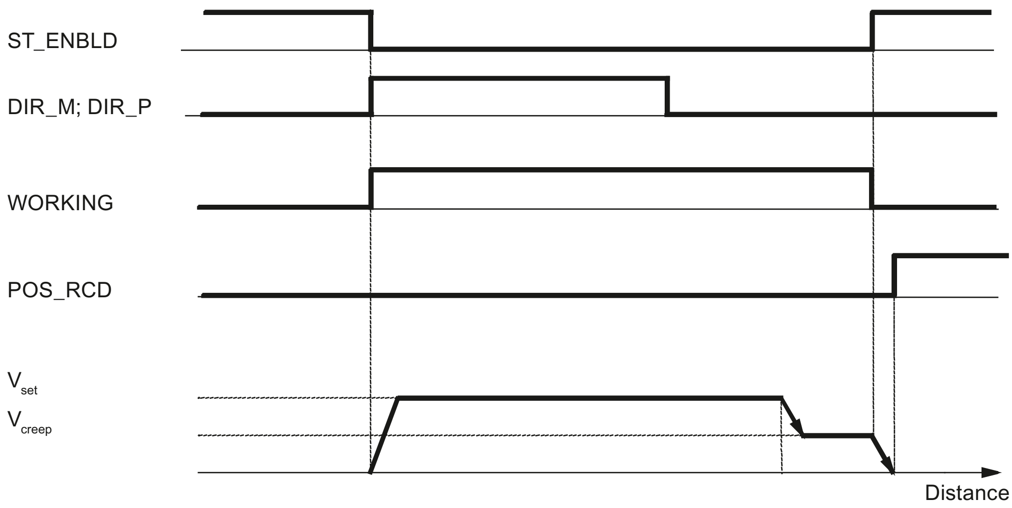
2. Call the SFB.

Result

The **output parameters** of the SFB provide the following information:

Parameters	Data type	Address (instance DB)	Description	Value range	Default
WORKING	BOOL	16.0	Run has started	TRUE/FALSE	FALSE
POS_RCD	BOOL	16.1	Position reached	TRUE/FALSE	FALSE
ACT_POS	DINT	18	Actual position value	-5×10^8 to $+5 \times 10^8$pulses	0
MODE_OUT	INT	22	Enabled/set operating mode	0, 1, 3, 4, 5	0

- WORKING = TRUE is set immediately after the run has started. WORKING is reset to FALSE at the cut-off point. POS_RCD = TRUE is set when the specified target is reached.
- You must reset the direction bit (DIR_P or DIR_M) before you start the next run.
- If an error occurred when the SFB call was interpreted, WORKING = FALSE and ERROR will be set to TRUE. The precise error cause is then indicated with the STATUS parameter (see Section Error Lists (Page 84)).



Interrupting a Run and Target Range not Reached

When a run is interrupted with STOP = TRUE and if the cut-off range has not been reached (distance to go is greater than the cut-off difference), you have the following options depending on the subsequent operating mode/job.

Option	Response
Start of a new "Absolute incremental approach"	The axis moves to the specified absolute target.
Continuing the run into the same direction with "Relative incremental approach" mode	Run parameters will not be interpreted. The axis performs the run to the target of the interrupted run (LAST_TRG).
Continuing the run into the reverse direction with "Relative incremental approach" mode	Run parameters will not be interpreted. The axis moves to the starting point of the interrupted run.
Job "Delete distance to go"	The distance to go (difference between target and actual value) will be deleted. The run parameters are interpreted again at the start of a subsequent "Relative incremental approach" and the axis moves to the current actual position value.

3.4.8 Specifying the Reference Point

Description

You can also use the "Set reference point" request to synchronize the axis without performing a reference point approach.

After the job has been executed, the actual position coordinate has the value you have assigned via the JOB_VAL parameter.

- Linear axis: The reference point coordinate must lie within the working range (including the software limit switches).
- Rotary axis: The reference point coordinate must lie within the range from 0 to "End of rotary axis" – 1.

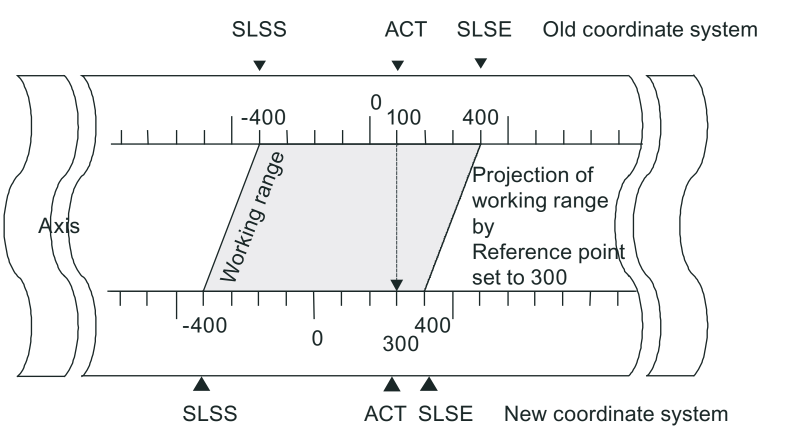
This does not change the reference point coordinates you have entered in the parameter assignment screens.

Example of Setting a Reference Point

The following is an example of how to set a reference point.

- The actual position value is 100. The software limit switches (SLSS, SLSE) are at the positions -400 and 400 (working range).
- The "Set reference point" request is executed with the value JOB_VAL = 300.

The actual value is then to coordinate 300. The software limit switches and the working range have the same coordinates as prior to the job. However, they are now physically shifted left by 200.



	SLS	ACT	SLE
Old coordinate system	-400	100	400
New coordinate system	-400	300	400

Requirements

- You have assigned the module parameters via parameter assignment screens and downloaded them to the CPU (PARA = TRUE).
- You have assigned the basic parameters of the SFB as described in Section Basic Configuration of the SFB ANALOG (SFB 44) (Page 48).
- The last job must be finished (JOB_DONE = TRUE).
- The last positioning operation must be ended (WORKING = FALSE).

Procedure

1. Assign the following **input parameters** (accessible via instance DB) as specified in the "Setting" column:

Parameter	Data type	Address (instance DB)	Description	Value range	Default	Setting
JOB_REQ	BOOL	76.0	Job trigger (positive edge)	TRUE/FALSE	FALSE	TRUE
JOB_ID	INT	78	Job, 1 = Set reference point	1, 2	0	1
JOB_VAL	DINT	82	Job parameters for the coordinates of the reference point	-5×10^8 to $+5 \times 10^8$pulses	0	xxxx

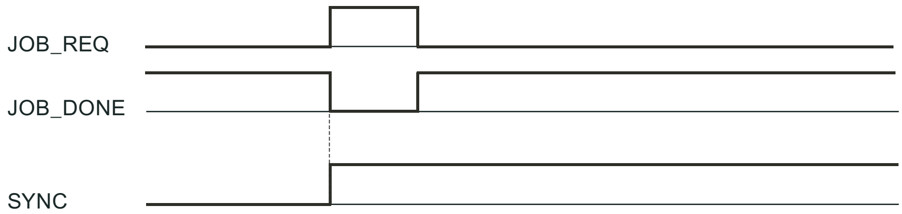
2. Call the SFB.

Result

The **output parameters** of SFB (JOB_DONE, JOB_ERR, JOB_STAT accessible via instance DB) provide the following information:

Parameters	Data type	Address (instance DB)	Description	Value range	Default
SYNC	BOOL	16.3	Axis is synchronized	TRUE/FALSE	FALSE
JOB_DONE	BOOL	76.1	New job can be started	TRUE/FALSE	TRUE
JOB_ERR	BOOL	76.2	Faulty job	TRUE/FALSE	FALSE
JOB_STAT	WORD	80	Job error number (see Section Error Lists (Page 84)).	0 to FFFF hex	0

- The job is immediately processed after the SFB is called. JOB_DONE is set to FALSE for the duration of one SFB cycle.
- You must reset the job request (JOB_REQ).
- SYNC = TRUE if the job was processed without error.
- JOB_ERR = TRUE if an error occurs. The precise error cause is then indicated in JOB_STAT.
- A new job can be started with JOB_DONE = TRUE.



Effects of the job

- The actual position value will be set to the value of the reference point coordinate and the status signal SYNC is set.
- The working range is shifted physically along the axis.
- All points within the working range maintain their original coordinates, but have new physical positions.

Simultaneous Call of a Job and a Positioning Operation

When a positioning operation and a job are initiated simultaneously, the job is executed first. Positioning is not executed if the job ends with an error.

A job initiated while a run is busy will be ended with an error.

3.4.9 Deleting the Distance-to-Go

Description

After a target run (absolute or relative incremental approach) the pending distance-to-go (DIST_TO_GO) can be deleted with the job.

Requirements

- You have assigned the module parameters via parameter assignment screens and downloaded them to the CPU (PARA = TRUE).
- You have assigned the basic parameters of the SFB as described in Section Basic Configuration of the SFB ANALOG (SFB 44) (Page 48).
- The last job must be finished (JOB_DONE = TRUE).
- The last positioning operation must be ended (WORKING = FALSE).

Procedure

- Assign the following **input parameters** (accessible via instance DB) as specified in the "Setting" column:

Parameter	Data type	Address (instance DB)	Description	Value range	Default	Setting
JOB_REQ	BOOL	76.0	Job trigger (positive edge)	TRUE/FALSE	FALSE	TRUE
JOB_ID	INT	78	Job, 2 = Delete distance-to-go	1, 2	0	2
JOB_VAL	DINT	82	None	-	0	Any

- Call the SFB.

  The **output parameters** of the SFB (accessible via instance DB) provide the following information:

Parameter	Data type	Address (instance DB)	Description	Value range	Default
JOB_DONE	BOOL	76.1	New job can be started	TRUE/FALSE	TRUE
JOB_ERR	BOOL	76.2	Faulty job	TRUE/FALSE	FALSE
JOB_STAT	WORD	80	Job error ID	0 to FFFF hex	0

- The job is immediately processed after the SFB is called. JOB_DONE is set to FALSE for the duration of one SFB cycle.
- You must reset the job request (JOB_REQ).
- JOB_ERR = TRUE if an error occurs. The precise error cause is then indicated in JOB_STAT.
- A new job can be started with JOB_DONE = TRUE.

Simultaneous Call of a Job and a Positioning Operation

When a positioning operation and a job are initiated simultaneously, the job is executed first. Positioning is not executed if the job ends with an error.

A job initiated while a run is busy will be ended with an error.

3.4.10 Length measurement

Description

With "Length measurement" you can determine the length of a work piece. The start and stop of length measurement is edge-triggered at the digital input "Length measurement".

At the SFB you are given the coordinates for length measurement start and end as well as the measured length.

With the help of the parameter assignment screens (Parameter "Length measurement") you can switch length measurement on and off and also determine the type of edge:

- Off
- Start/end at the positive edge
- Start/end at the negative edge
- Start with positive edge and end with negative edge
- Start with negative edge and end with positive edge

Requirements

- You have assigned the module parameters via parameter assignment screens and downloaded them to the CPU (PARA = TRUE).
- You have assigned the basic parameters of the SFB as described in Section Basic Configuration of the SFB ANALOG (SFB 44) (Page 48).
- You have connected a bounce-free switch to the digital input "Length measurement" (connector X2, pin 5).
- "Length measurement" is possible with synchronized (SYNC = TRUE) as well as non-synchronized (SYNC = FALSE) axis.

Procedure

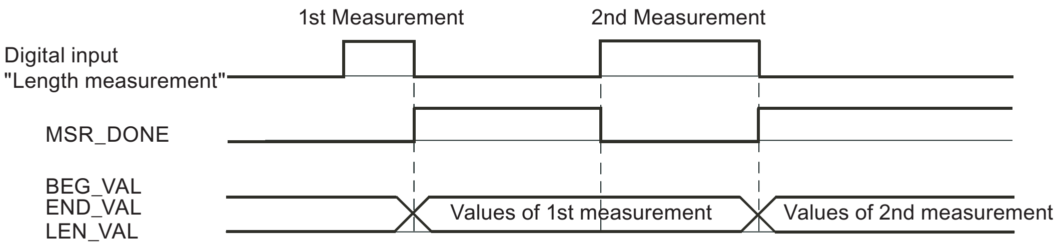
- An edge at the digital input starts length measurement.
- MSR_DONE is reset at the start of length measurement.
- MSR_DONE status is set to TRUE at the end of length measurement.
- The SFB then outputs the following values:
  - Start of length measurement: BEG_VAL
  - End of length measurement: END_VAL
  - Measured length: LEN_VAL

At the end of one length measurement until the end of the next length measurement the values are available at the block.

- The **output parameters** of the SFB (BEG_VAL, END_VAL, LEN_VAL are accessible via instance DB) provide following information:

Parameter	Data type	Address (instance DB)	Description	Value range	Default
MSR_DONE	BOOL	16.2	End of length measurement	TRUE/FALSE	FALSE
BEG_VAL	DINT	64	Actual position value, start of length measurement	-5 x 10^8 to +5 x 10^8pulses	0
END_VAL	DINT	68	Actual position value, end of length measurement	-5 x 10^8 to +5 x 10^8pulses	0
LEN_VAL	DINT	72	Measured length	0 to 10^9 pulses	0

The figure below shows the signal profile for a length measurement of the type: Start/End of length measurement at the positive/negative edge.



Note

When referencing during a length measurement; the change of the actual value is taken into account as follows:

Example: A length measurement is performed between two points with a distance of 100 pulses. When referencing during the length measurement, the coordinates are shifted by + 20. This results in a measured length of 120.

3.5 Adapting Parameters

3.5.1 Important Safety Rules

Important Note

Please note the following warning points:

WARNING

Harm to health or damage to assets is to be expected.

To avoid injury and damage to property, please note the following:

- Install an **Emergency-Off switch** in the area of the control system. This is the only possible way for you to ensure that the system can be safely switched off in case of control system failure.
- Install **Hardware limit switches** which affect the drive converters of all drives directly.
- Make sure that **no one has access to system areas** in which moving parts exist.
- **Parallel control and monitoring** via your program and STEP7 interface can cause conflicts which result in indefinite effects.

3.5.2 Determining the Module Parameters and Their Effects

Increments per encoder revolution

The **Increments per encoder revolution** parameter of the connected incremental encoder is found on its rating plate or in the specifications sheet. The technology evaluates the encoder signals in quadruple mode. Four pulses represent one encoder increment. All distance specifications are referenced to pulse units.

Maximum speed

You must calculate the **Maximum speed** parameter. Prerequisite is that you know the rated speed of the drive (with ±10 V at the analog output). You can find this information in the technical specification of your drive. If the encoder is mounted to the motor via a gear, you must take the gear ratio into account because the maximum speed is referenced to the encoder.

Maximum speed [pulses/s] =
Rated speed of the drive [rev/s] x gear ratio x
increments per encoder revolution [increments/revolution] x 4

Example:

Rated speed of the drive:	3000 [rpm]
Transmission ratio:	1 : 1 (no gear)
Increments per encoder revolution:	500 [increments/revolution]

3000 [rpm] = 50 [rev/s]
500 [increments/rev] = 2000 [pulses/rev]

$$\text{Maximum speed} = 50 \frac{\text{Revolutions}}{\text{s}} \times 1 \times 2000 \frac{\text{Pulses}}{\text{Revolution}} = 100000 \frac{\text{Pulses}}{\text{s}}$$

It is imperative to determine and specify the maximum speed correctly in order to achieve good and reproducible positioning results.

Creep Speed / Reference Speed

The Creep speed/reference speed parameter also refers to the encoder. Here, the specified maximum speed is converted into an analog voltage.

For example, if the maximum speed is 10000 pulses/s and the creep/ reference speed 1000 pulses/s, a voltage of 1 V will be output at the analog output.

The creep/reference speed must be high enough to keep the drive moving.

Monitoring Time

You must select a time of sufficient length in the Monitoring time parameter to ensure that the drive can supersede the startup holding torque of the axis within the specified time.

Example:
The drive starts moving at an analog voltage of 0.5 V.

Maximum speed:	10,000 [pulses/s] = 10 V
Acceleration:	1,000 [pulses/s^2]

⇒ Speed = 500 pulses/s = 0.5 V

⇒ T = speed / acceleration = 500 pulses/s / 1,000 pulses/s^2 = 0.5 s

i.e. the drive does not start moving until 0.5 s have expired. In this case the monitoring time must be set higher than 0.5 s.

The monitoring time is also used for target approach monitoring. This means the drive must reach the target range within this time after reaching the cut-off position.

Count direction

Use the Count direction parameter to adapt the direction of path monitoring to the direction of movement of the linear axis. Also take the direction of rotation of all transmission elements into account (for example couplings and gears).

- "Standard" means, the incrementing count pulses correspond to rising actual position values.
- "Inverted" means, the incrementing count pulses correspond to descending actual position values.

3.5.3 Effect of the SFB Parameters

ACCEL and DECEL

In the ACCEL (acceleration) and DECEL (deceleration) parameters, declare the acceleration/deceleration speed values for the drive.

Example:
At a wanted traversing speed of 10,000 pulses/s and acceleration of 1,000 pulses/s^2, it takes 10 s to reach the speed setpoint of 10,000 pulses/s.

CHGDIFF_P and CHGDIFF_M

The CHGDIFF_P (changeover difference in plus direction) and CHGDIFF_M (changeover difference in minus direction) parameters define the changeover position as of which the drive runs at creep speed.

If the difference is set too high, positioning is not optimized over time because creep speed runtime is unnecessarily extended .

CUTOFFDIFF_P and CUTOFFDIFF_M

The parameters CUTOFFDIFF_P (Cut-off difference in plus direction) and CUTOFFDIFF_M (Cut-off difference in minus direction) specify the pulses to go before the drive is switched off at the target approach.

Take into consideration that this distance varies according to the load on the drive.

If you select an insufficiently high difference of changeover/cut-off difference, the drive is switched off at a speed higher than the configured creep speed. The result is inexact positioning.

The difference of changeover/cut-off difference for the respective direction should at least be proportional to the distance the drive actually requires to reach creep speed. Here, the required traversing speed forms the base. Take the load on the drive into consideration.

3.5.4 Checking the Monitoring Time

Requirements

- Your system is wired correctly.
- The positioning submodule is configured, the parameters have been assigned and the project is loaded.
- For example, you have loaded the sample program "Analog 1, Getting Started" which is included in the scope of delivery.
- The CPU is in RUN.

Checklist

Step	What to do	✓
1	**Verify the wiring**	
	• Verify that the outputs are connected correctly (Analog output and Enable output "CONV_EN" for the converter)	❑
	• Verify that the encoder inputs are connected correctly	❑
2	**Check the axis motion**	
	• In "Jog" mode, move into plus and minus direction at creep speed (see module parameters). The actual sense of direction DIR must correspond with the specified direction. If this is not the case, change the module parameter "Count direction".	❑
3	**Synchronize the axis**	
	• Select the job "Set reference" (JOB_ID = 1). Enter the desired coordinate at the actual axis position as JOB_VAL (e. g. 0 pulses). Execute synchronization by setting JOB_REQ to TRUE. The coordinate you have specified is shown as actual position value and the synchronization bit SYNC is set. Evaluate (JOB_STAT) a reported error (JOB_ERR = TRUE). If required, correct your specified coordinate and repeat the job for setting the reference.	❑

Step	What to do	✓
4	**Check the changeover/cut-off differences**	
	• In "Absolute/relative incremental approach" mode, approach the specified target (TARGET) that is further away from the actual position than the configured changeover difference. Here, select a speed (SPEED) that is adapted to your application and higher than creep speed. Creep speed ≤ SPEED ≤ maximum speed.	❑
	• Note the individual positioning phases (acceleration, constant run, deceleration, target approach).	❑
	Increase the changeover difference to run the drive to the cut-off point at a clearly visible creep speed.	❑
	If the configured target range is not reached, reduce the cut-off difference and repeat the run until the target range is reached.	❑
	If the configured target range is overrun, increase the cut-off difference and repeat the run until the target range is not overrun anymore.	❑
	• Now optimize the changeover difference. Reduce the changeover difference without changing the cut-off difference and repeat the run. You can reduce the changeover difference to a point at which the drive just about moves at a hardly visible creep speed, that is, it has actually reached creep speed at the cut-off position and it is switched off there. Positioning accuracy remains unchanged as long as the drive is switched off at creep speed. A further reduction of the cut-off difference does not make sense.	
5	**Check the maximum speed (if poor positioning results)**	
	• In "Jogging" mode, move into plus and minus direction at maximum speed (see module parameters). Measure the frequency (e. g. using the counter submodule) of the encoder signal A or B in [1/s]. Multiply the measured frequency byte 4 and enter the product as maximum speed in the module parameters.	❑

3.6 Error Handling and Interrupts

3.6.1 Error Messages at the System Function Block (SFB)

Types of error

The SFB indicates the errors listed in the table below.

Except for system errors, all errors are specified in closer detail via an error number, which is available as an output parameter in the SFB.

Type of error	Errors are displayed in SFB parameters	The error number is displayed in the SFB parameters
Operating mode error	ERROR = TRUE	STATUS
Job Error	JOB_ERR = TRUE	JOB_STAT
External error	ERR > 0	ERR
System error	BIE = FALSE	–

Operating Mode Error (ERROR = TRUE)

This error occurs

- upon general parameter assignment errors at the SFB (e.g. use of incorrect SFB)
- at run start/resume. These errors occur during the interpretation of operating mode parameters.

When an error is detected, output parameter ERROR is set to TRUE.

The possible error numbers can be found in Section Error Lists (Page 84).

The parameter STATUS indicates the cause of error.

Job Error (JOB_ERR = TRUE)

Job errors can only occur during the interpretation/execution of a job.

When an error is detected, output parameter JOB_ERR is set to TRUE.

The possible error numbers can be found in Section Error Lists (Page 84).

The error cause is indicated at the JOB_STAT parameter.

External error (ERR)

The system monitors the run, traversing range and the connected I/O. Prerequisite is here that you have switched on monitoring in the "Drive", "Axis" and "Encoder" parameter assignment screens.

An external error is signaled when the monitoring responds.

External errors can occur at any time, regardless of the started functions.

You must acknowledge queued external errors with a positive edge on ERR_A.

External errors are indicated at the SFB parameter ERR (WORD) by setting a bit.

Monitoring	ERR	Bit in ERR-WORD
Missing pulse (zero mark)	0004 hex	2
Traversing range	0800 hex	11
Working range	1000 hex	12
Actual value	2000 hex	13
Target approach	4000 hex	14
Target range	8000 hex	15

The detection of an external error ("incoming" and "outgoing") can also trigger a diagnostic interrupt (see Section Configuring and Evaluating Diagnostic Interrupts (Page 80)).

System error

A system error is indicated with BIE = FALSE.

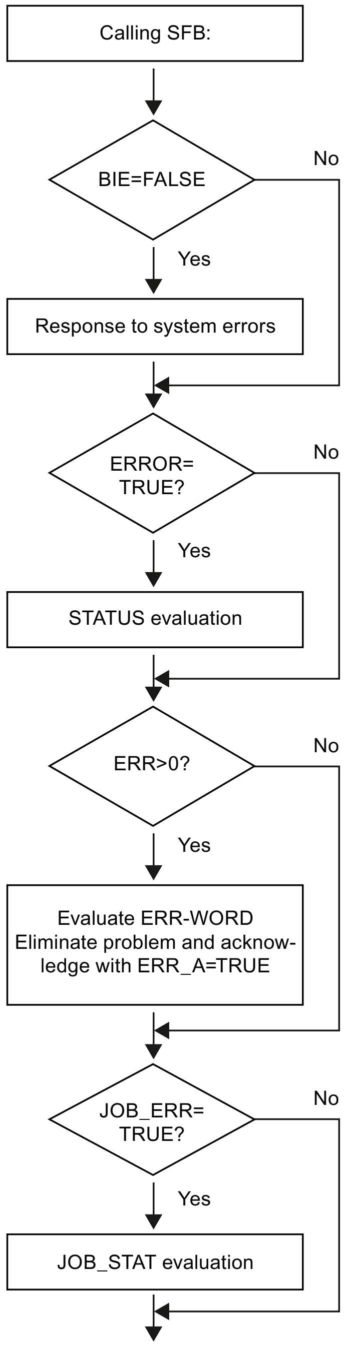
A system error is triggered by:

- Read/write access errors at the instance DB
- Multiple calls of the SFB

3.6.2 Error Evaluation in the User Program

Procedure

1. Call the error handling routine "Error evaluation" (see the view).
2. Query the specific error types in successive order.
3. If required, jump to the error reaction method that is specifically adapted to your application.
4. Error evaluation:



3.6.3 Configuring and Evaluating Diagnostic Interrupts

Basics

On occurrence of the following errors you can trigger a diagnostic interrupt:

- Parameter assignment error (Module data)
- External error (Monitoring)

The diagnostic interrupt is displayed in the event of incoming as well as outgoing errors.

In your user program, you can immediately respond to errors with the help of a diagnostic interrupt.

Procedure

1. Enable diagnostic interrupt in the "Basic parameters" dialog of the parameter assignment screens.
2. In the "Drive", "Axis" and "Encoder" parameter assignment screens, switch on the individual monitoring functions that should trigger a diagnostic interrupt when an error occurs.
3. In the parameter assignment screen "Diagnostics", enable diagnostic interrupts for each monitoring facility individually.
4. Incorporate the diagnostic interrupt OB (OB 82) in your user program.

Response to an Error with Diagnostic Interrupt

- Positioning is canceled.
- The CPU operating system calls OB82 in the user program.

> **Note**
>
> If the corresponding OB is not loaded the CPU switches to STOP when an interrupt is triggered.

- The CPU switches on the SF LED.
- The error is reported in the diagnostics buffer of the CPU as "incoming". An error is not indicated as "outgoing" until all pending errors are cleared.

How a Diagnostic Interrupt is Evaluated in the User Program

After a diagnostic interrupt is triggered, you can evaluate OB 82 to check which diagnostic interrupt is pending.

- If the module address of the "Positioning" submodule was entered in OB 82, byte 6 + 7 (OB 82_MDL_ADDR), the diagnostic interrupt was triggered by the positioning function of your CPU.
- Bit 0 of byte 8 in OB 82, bit 0 (Faulty module) is set as long as any errors are queued.
- In OB 82, bit 0 of byte 8 will be reset after all errors have been reported "outgoing".
- You can determine the precise error cause by evaluating data record 1, byte 8 and 9. To do this, you must call SFC 59 (read data record).
- Acknowledge the error with ERR_A.

Data record 1, byte 8	Description:	JOB_STAT	ERR
Bit 0	not used	–	–
Bit 1	not used	–	–
Bit 2	Missing pulse*	–	X
Bit 3	not used	–	–
Bit 4	not used	–	–
Bit 5	not used	–	–
Bit 6	not used	–	–
Bit 7	not used	–	–
* subsequent errors trigger an incoming and then automatically an outgoing interrupt.			

Data record 1, byte 9	Description:	JOB_STAT	ERR
Bit 0	Configuration error	X	–
Bit 1	not used	–	–
Bit 2	not used	–	–
Bit 3	Traversing range monitoring	X	X
Bit 4	Working range monitoring	X	X
Bit 5	Actual value monitoring*	X	X
Bit 6	Target approach monitoring*	X	X
Bit 7	Target range monitoring*	X	X
* subsequent errors trigger an incoming and then automatically an outgoing interrupt.			

3.7 Installation of Examples

Using Examples

The examples (program and description) are found on the CD-ROM included in your documentation. You can also download them from the Internet. The project consists of several commented S7 programs of various complexity and aim.

The Readme.wri on the CD describes how to install the samples. After they are installed, the examples are stored in the catalog ...\STEP7\EXAMPLES\ZDt26_03_TF_____31xC_Pos.

3.8 Specifications

3.8.1 Incremental encoders

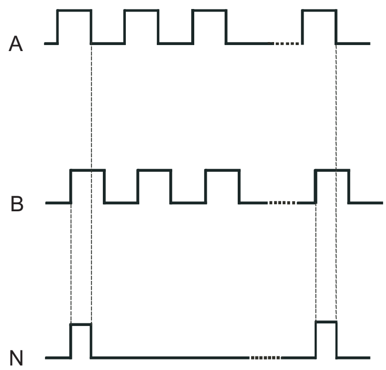
Connectable Incremental Encoders

Asymmetrical 24 V incremental encoders which have two pulse tracks with an electrical phase difference of 90° with or without zero mark are supported.

Encoder inputs	Pulse width min/ pulse pause min	Input frequency max.	Cable length max. (at max. input frequency)
Encoder signal A, B	8 μs	60 kHz	50 m
Encoder signal N (zero mark signal)	8 μs	60 kHz/30 kHz[1]	50 m
[1] If you are using an encoder whose zero mark signal is combined with encoder signals A and B using an "AND" operation, the pulse width is reduced by half to 25% of the period. In order to maintain the minimum pulse width, the maximum counting frequency must be reduced to 30 kHz.			

Signal Evaluation

The figure below shows the signal profile of encoders with asymmetric output signals:



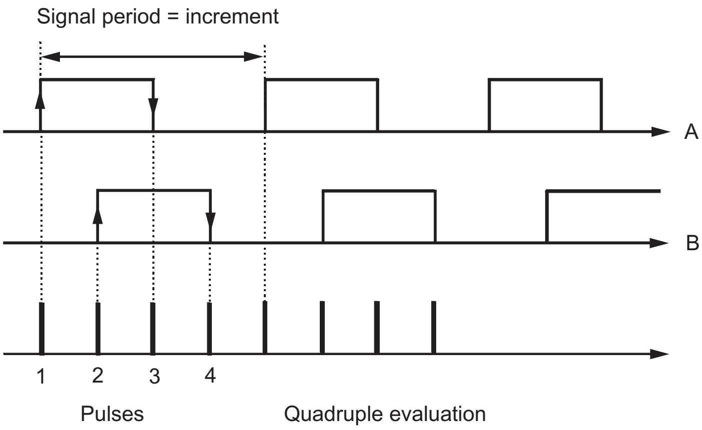
The CPU internally generates a logical AND link of the zero mark signal and the A and B track signals.

For referencing, the CPU uses the positive edge at the zero mark.

The CPU counts in positive direction if the signal A transition leads signal B.

Increments

An increment identifies a signal period of the two encoder track signals A and B. This value is specified on the rating plate of the encoder and/or in the specifications for the encoder.

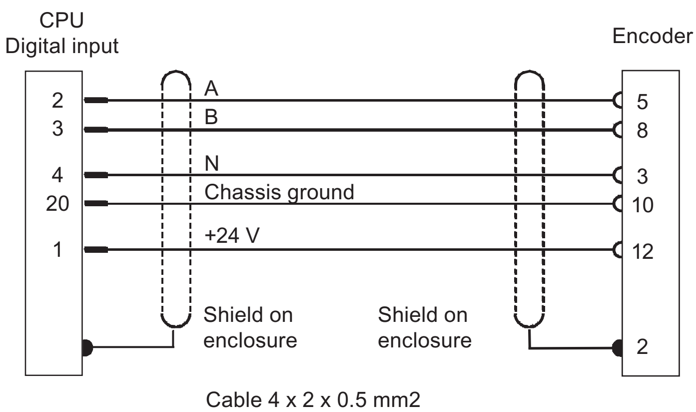


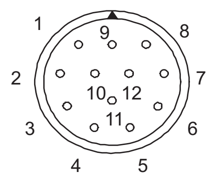
Pulses

The CPU evaluates all 4 edges of the track signals A and B (see the view) with every increment (quadruple evaluation). i.e. one encoder increment is proportional to four pulses.

Wiring Diagram of the Incremental Encoder Siemens 6FX 2001-4 (Up = 24 V;HTL)

The figure below shows the wiring diagram for the Siemens 6FX 2001-4xxxx (Up = 24 V; HTL) incremental encoder:





12-pin circular connector socket Siemens 6FX 2003-0CE12

Connection side (solder side)

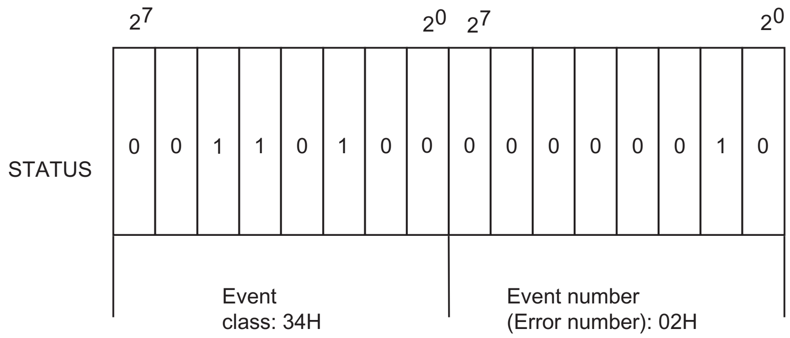
3.8.2 Error Lists

Basics

If an error occurs, an error ID is output at the SFB parameters STATUS or JOB_STAT. The error ID consists of an event class and number.

Example of an Error List

The view below shows the content of the STATUS parameter for the event "Incorrect target specified" (Event class: 34H, event number: 02H).



Error IDs at SFB Parameter "Status"

Event class 32 (20H): "SFB error"		
Event no.	**Event text**	**Remedy**
(20)02H	Incorrect SFB	Use SFB 44
(20)04H	Incorrect channel number (CHANNEL)	Set channel number "0"

Event class 48 (30H): "General run start error"		
Event no.	**Event text**	**Remedy**
(30)01H	Run job rejected because of faulty job in the same SFB call	Correct the respective JOB parameters
(30)02H	It is not allowed to modify MODE_IN while the drive is still in motion.	Wait until the current positioning operation has ended.
(30)03H	Unknown operating mode (MODE_IN)	The following is permitted: 1 (jog mode), 3 (reference point approach), 4 (relative incremental approach) and 5 (absolute incremental approach).
(30)04H	Start requests may only be set one at a time.	Permissible start requests are DIR_P, DIR_M or START
(30)05H	START is only allowed in "Absolute incremental approach" mode	Start the run with DIR_P or DIR_M
(30)06H	DIR_P or DIR_M is not allowed for a linear axis and in "Absolute incremental approach" mode	Start the run with START
(30)07H	Axis not synchronized	"Absolute incremental approach" is only possible if the axis is synchronized.
(30)08H	Moving out of working range	Run is only allowed into the direction of the working range.

Event class 49 (31H): "Run start error (Start enable)"		
Event no.	**Event text**	**Remedy**
(31)01H	Start not enabled because the axis is not configured.	Configure the "Position" submodule via HW Config
(31)02H	Start not enabled because drive enable is not set.	Set "Start enable" at the SFB (DRV_EN = TRUE)
(31)03H	Start not enabled because STOP is set.	Clear the STOP at the SFB (STOP = FALSE)
(31)04H	Start not enabled because the axis currently performs a positioning run (WORKING = TRUE).	Wait until the current positioning operation is terminated
(31)05H	Start not enabled because at least one pending error has not been acknowledged.	First, eliminate and acknowledge all external errors, then restart the run.

Event class 50 (32H): "Run start error (speed / acceleration)"		
Event no.	Event text	Remedy
(32)02H	Incorrect speed specification SPEED	The specified speed is outside the permissible range for the creep speed of up to 1000000 pulses/s, but does not exceed the assigned maximum speed.
(32)03H	Incorrect acceleration specification ACCEL	The specified acceleration value is outside the permissible range of 1 to 100000 pulses/s^2.
(32)04H	Incorrect deceleration specification DECEL	The specified deceleration value is outside the permissible range of 1 to 100000 pulses/s^2.
(32)06H	Incorrect speed specification SPEED	The specified speed value must be greater than/equal to the assigned reference frequency.

Event class 51 (33H): "Run start error (Changeover / cut-off differences)"		
Event no.	Event text	Remedy
(33)01H	Changeover/cut-off differences greater than 10^8 are not permitted	Specify a changeover/cut-off difference of maximum 10^8
(33)04H	Cut-off difference too small	The cut-off difference must have at least the length of half the target range.
(33)05H	Changeover difference too small	The changeover difference must have at least the length of half the target range.

Event class 52 (34H): "Run start error (default target/distance)"		
Event no.	Event text	Remedy
(34)01H	Default target out of working range	With a linear axis and absolute incremental approach, the default target must lie within the range of the software limit switch (inclusive).
(34)02H	Incorrect target specification	For the rotary axis, the specified target must be greater than 0 and less than the end of rotary axis.
(34)03H	Incorrect distance specification	With relative incremental approach the distance to be traversed must be positive.
(34)04H	Incorrect distance specification	The resulting absolute target coordinate must be greater than -5 x 10^8.
(34)05H	Incorrect distance specification	The resulting absolute target coordinate must be greater than 5 x 10^8.
(34)06H	Incorrect distance specification	The resulting absolute target coordinate must lie within the working range (+/-half of the target range)

Event class 53 (35H): "Run start error (traversing distance)"		
Event no.	Event	Remedy
(35)01H	Traversing distance too long	The target coordinate + actual distance to go must be greater than/equal to -5 x 10^8
(35)02H	Traversing distance too long	The target coordinate + actual distance to go must be less than/equal to 5 x 10^8
(35)03H	Traversing distance too short	The traversing distance in the plus direction must be greater than the specified cut-off difference for the plus direction
(35)04H	Traversing distance too short	The traversing distance in the minus direction must be greater than the specified cut-off difference for the minus direction
(35)05H	Traversing distance too short or the limit switch has already been overrun in plus direction	The last approachable target in the plus direction (working range/traversing range limits) is too close to the actual position
(35)06H	Traversing distance too short or the limit switch has already been overrun in minus direction	The last approachable target in the minus direction (working range/ traversing range limits) is too close to the actual position

Error numbers at SFB Parameter JOB_STAT

Event class 64 (40H): "General job execution error"		
Event no.	Event	Remedy
(40)01H	Axis not configured	Configure the "Position" submodule via HW Config.
(40)02H	Job not possible because positioning is still running	Jobs can only be executed if no positioning run is active. Wait until WORKING = FALSE and then repeat the job.
(40)04H	Unknown job	Check the Job ID and repeat the job.

Event class 65 (41H): "Error when executing the request to set the reference point"		
Event no.	Event	Remedy
(41)01H	Reference point coordinate out of working range	With a linear axis, the reference point coordinate must not exceed the working range limits.
(41)02H	Incorrect reference point coordinate	With a linear axis, the specified reference point coordinate + actual distance to go must still be greater than/equal to -5 x 10^8.
(41)03H	Incorrect reference point coordinate	With a linear axis, the specified reference point coordinate + actual distance to go must still be less than/equal to 5 x 10^8.
(41)04H	Incorrect reference point coordinate	With a linear axis, the specified reference point coordinate + actual difference to the starting point of the run must still be greater than/equal to -5 x 10^8.
(41)05H	Incorrect reference point coordinate	With a linear axis, the specified reference point coordinate + actual difference to the starting point of the run must still be less than/equal to 5 x 10^8 .
(41)06H	Reference point coordinate out of rotary axis range	With a rotary axis, the reference point coordinate must not be less than 0 and greater than/equal to the end of rotary axis.

External error (ERR)

External errors are indicated at the SFB parameter ERR (WORD) by setting a bit.

Monitoring	ERR	Bit in ERR-WORD
Missing pulse (zero mark)	0004 hex	2
Traversing range	0800 hex	11
Working range	1000 hex	12
Actual value	2000 hex	13
Target approach	4000 hex	14
Target range	8000 hex	15

3.8.3 Module Parameters of the Parameter Assignment Screen – Overview

Basic parameters

Parameters	Range of values	Default
Interrupt selection	• None • Diagnostics	None

Drive parameters

Parameters	Value range	Default
Target range	0 to 200,000,000 pulses The CPU rounds up odd values.	50
Monitoring time	• 0 to 100 000 ms • 0 = No monitoring Rounded up by the CPU in 4-ms steps.	2000
Maximum speed	10 to 1,000,000 pulses/s	1000
Creep/ Reference speed	10 up to the configured maximum speed	100
Off delay	0 to 100 000 ms Rounded to a 4 ms process cycle	1000
Max. frequency: Position feedback	60, 30, 10, 5, 2, 1 kHz	60 kHz
Max. frequency: Accompanying signals	60, 30, 10, 5, 2, 1 kHz	10 kHz
Control mode	• Voltage ±10 V or current ±20 mA • Voltage of 0 to 10 V or current of 0 to 20 mA and direction signal	Voltage ±10 V or current ±20 mA
Actual value monitoring	• Yes • No	Yes
Target approach monitoring	• Yes • No	No
Target range monitoring	• Yes • No	No

Axis Parameters

Parameters	Value range	Default
Axis type	• Linear axis • Rotary axis	Linear axis
Software limit switch Start / End	Software limit switch start software limit switch end -5×10^8 to $+5 \times 10^8$pulses	-100 000 000 +100 000 000
End of rotary axis	1 to 10^9 pulses	100 000
Length measurement	• Off • Start/End at the positive edge DI • Start/End at the negative edge DI • Start with positive and end with negative edge • Start with negative and end with positive edge	Off
Reference point coordinate	-5×10^8 to $+5 \times 10^8$pulses	0
Reference point location for reference point switch	• Plus direction (actual values increase) • Minus direction (actual values decrease)	Plus direction
Traversing range monitoring	Yes (set fixed)	Yes
Working range monitoring	• Yes • No	Yes

Encoder parameters

Parameters	Value range	Default
Increments per encoder revolution	1 to 2^{23} pulses	1000
Count direction	• Standard • Inverted	Normal
Missing pulse (zero mark) monitoring	• Yes • No	No

Diagnostic Parameters

Parameters	Value range	Default
Missing pulse (zero mark)	• Yes • No	No
Traversing range	• Yes • No	No
Working range (for linear axes)	• Yes • No	No
Actual value	• Yes • No	No
Target approach	• Yes • No	No
Target range	• Yes • No	No

3.8.4 Parameters for Instance DB of the SFB ANALOG (SFB44)

Overview

Parameters	Declaration	Data Type	Address (Instance DB)	Description	Value range	Default
LADDR	IN	WORD	0	Submodule I/O address you specified in "HW Config". If the I/O addresses are not equal you must specify the lower one of both.	CPU-specific	310 hex
CHANNEL	IN	INT	2	Channel number	0	0
DRV_EN	IN	BOOL	4.0	Drive enable	TRUE/FALSE	FALSE
START	IN	BOOL	4.1	Start run (positive edge)	TRUE/FALSE	FALSE
DIR_P	IN	BOOL	4.2	Run in plus direction (positive edge)	TRUE/FALSE	FALSE
DIR_M	IN	BOOL	4.3	Run in minus direction (positive edge)	TRUE/FALSE	FALSE
STOP	IN	BOOL	4.4	Stop run	TRUE/FALSE	FALSE
ERR_A	IN	BOOL	4.5	Group error acknowledgment ERR_A is used to acknowledge external errors (positive edge)	TRUE/FALSE	
MODE_IN	IN	INT	6	Operating mode	0, 1, 3, 4, 5	1
TARGET	IN	DINT	8	Relative incremental approach: Distance in pulses (only positive values allowed)	0 to 10^9	1000
				Absolute incremental approach: Target in pulses	Linear axis: -5×10^8 to $+5 \times 10^8$ Rotary axis: 0 to end of rotary axis - 1	
SPEED	IN	DINT	12	The axis is accelerated to "$v_{setpoint}$".	10 to 1,000,000 pulses/s Up to the configured maximum speed, at most	1000
WORKING	OUT	BOOL	16.0	Run is busy	TRUE/FALSE	FALSE
POS_RCD	OUT	BOOL	16.1	Position reached	TRUE/FALSE	FALSE
MSR_ DONE	OUT	BOOL	16.2	End of length measurement	TRUE/FALSE	FALSE

Parameters	Declaration	Data Type	Address (Instance DB)	Description	Value range	Default
SYNC	OUT	BOOL	16.3	Axis is synchronized	TRUE/FALSE	FALSE
ACT_POS	OUT	DINT	18	Actual position value	-5×10^8 to $+5 \times 10^8$pulses	0
MODE_ OUT	OUT	INT	22	Active/set operating mode	0, 1, 3, 4, 5	0
ERR	OUT	WORD	24	External error Bit 2: missing pulse monitoring Bit 11: traversing range monitoring (always 1) Bit 12: working range monitoring Bit 13: actual value monitoring Bit 14: target approach monitoring Bit 15: target range monitoring The other bits are reserved	Every bit 0 or 1	0
ST_ENBLD	OUT	BOOL	26.0	Start enable	TRUE/FALSE	TRUE
ERROR	OUT	BOOL	26.1	Run start/resume error	TRUE/FALSE	FALSE
STATUS	OUT	WORD	28.0	Error ID	0 to FFFF hex	0
ACCEL	STAT	DINT	30	Acceleration	1 to 100,000 pulses/s^2	100
DECEL	STAT	DINT	34	Deceleration	1 to 100 000 pulses/s^2	100
CHGDIFF_P	STAT	DINT	38	Changeover difference plus	0 to $+10^8$ Pulses	1000
CUTOFF-DIFF_P	STAT	DINT	42	Cut-off difference plus	0 to $+10^8$ Pulses	100
CHGDIFF_M	STAT	DINT	46	Changeover difference minus	0 to $+10^8$ Pulses	1000
CUTOFF-DIFF_M	STAT	DINT	50	Cut-off difference minus	0 to $+10^8$ Pulses	100
PARA	STAT	BOOL	54.0	Axis is configured	TRUE/FALSE	FALSE
DIR	STAT	BOOL	54.1	Current/last sense of direction FALSE = Forward (plus direction) TRUE = Reverse (minus direction)	TRUE/FALSE	FALSE
CUTOFF	STAT	BOOL	54.2	Drive in cut-off range (from cut-off position to the start of the next run)	TRUE/FALSE	FALSE

Parameters	Declaration	Data Type	Address (Instance DB)	Description	Value range	Default
CHGOVER	STAT	BOOL	54.3	Drive in changeover range (from reaching changeover position to the start of the next run)	TRUE/FALSE	FALSE
RAMP_DN	STAT	BOOL	54.4	Drive is ramped down (from braking point to changeover position)	TRUE/FALSE	FALSE
RAMP_UP	STAT	BOOL	54.5	Drive is ramped up (from start to reaching end speed)	TRUE/FALSE	FALSE
DIST_TO_ GO	STAT	DINT	56	Actual distance to go	-5×10^8 to $+5 \times 10^8$pulses	0
LAST_TRG	STAT	DINT	60	Last/actual target	-5×10^8 to $+5 \times 10^8$pulses	0
BEG_VAL	STAT	DINT	64	Actual position value, start of length measurement	-5×10^8 to $+5 \times 10^8$pulses	0
END_VAL	STAT	DINT	68	Actual position value, end of length measurement	-5×10^8 to $+5 \times 10^8$pulses	0
LEN_VAL	STAT	DINT	72	Measured length	0 to 10^9 pulses	0
JOB_REQ	STAT	BOOL	76.0	Initiates the job (positive edge)	TRUE/FALSE	FALSE
JOB_DONE	STAT	BOOL	76.1	New job can be started	TRUE/FALSE	TRUE
JOB_ERR	STAT	BOOL	76.2	Faulty job	TRUE/FALSE	FALSE
JOB_ID	STAT	INT	78	Job ID	1, 2	0
JOB_STAT	STAT	WORD	80	Job error ID	0 to FFFF hex	0
JOB_VAL	STAT	DINT	82	Job parameters for the coordinates of the reference point	-5×10^8 to $+5 \times 10^8$pulses	0

Positioning with digital outputs 4

4.1 Wiring

4.1.1 Important Safety Rules

Adherence to Safety Rules

DANGER

For the safety concept of the system it is imperative to install the switchgear mentioned below and to adapt them to your system:

- **Emergency-Off switch** which you can use to shut down the entire system.
- **Hardware limit switches** that have a direct effect on all drive power units.
- **Motor protection**

WARNING

Harm to health and damage to assets cannot be excluded if you do not switch off voltage:

If you wire the front plug of the CPU in live state you risk injury due to the influence of electrical current!

Always wire the CPU in off-voltage state!

Harm to health and damage to assets due to missing safety devices:

If no Emergency-Off Switch is installed damage can be caused by connected aggregates.

Install an Emergency-Off switch that enables you to switch off all connected drives.

Note

Direct connection of inductive loads (e.g. relays and contactors) is possible without auxiliary circuitry.
If it is possible to switch off SIMATIC output current circuits via additionally installed auxiliary contacts (e.g. relay contacts), you must install additional surge voltage suppression elements across the coils of the inductive loads.

4.1.2 Wiring Rules

Connecting Cables/Shielding

- The cables for the analog outputs and the 24 V encoder must be shielded.
- The cables for the digital I/O must be shielded if their length exceeds 100 m.
- The cable shielding must be terminated on both ends.
- Flexible cable, cross-section 0.25 mm to 1.5 mm^2.
- Cable sleeves are not required. Should you still decide to do so, use cable sleeves without insulating collar (DIN 46228, Shape A, short version).

Shielding termination element

You can use this shielding termination element for easy shielded-cable-to-ground connections – due to the direct contact of the shielding termination element to the profile rail.

Additional Information

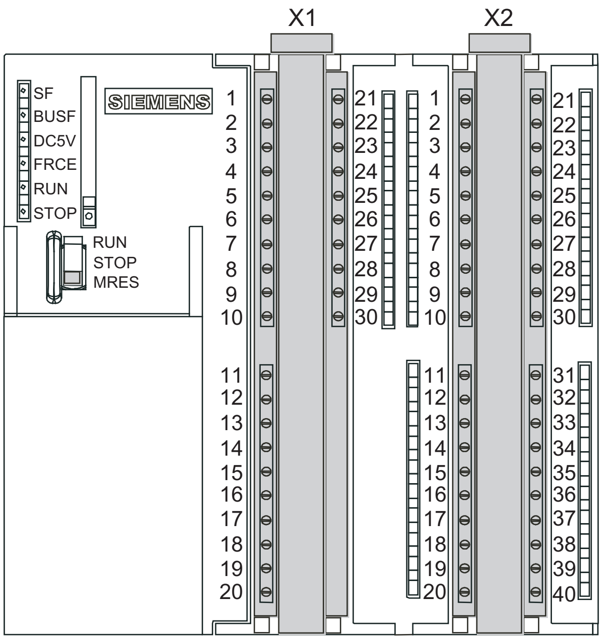
For additional information refer to the *CPU Data* manual and to the installation instructions for your CPU.

4.1.3 Terminals for Positioning with Digital Outputs

Overview

Use front connector X2 of the CPU 314C-2 DP, PN/DP, PtP to connect the following components:

- 24 V encoder
- Length measurement switches
- Reference point switch
- Converter (circuit-breaker)



Description of Pin Assignment

The following pin-out only relates to connections relevant to the position mode.

Note

Since they partially use the same inputs, you cannot use the counters 0 and 1 when you utilize the positioning function.

Table 4- 1 Pin Assignment for Connector X2

Connection	Name/ Address	Function
1	1 L+	24 V power supply for the inputs
2	DI + 0.0	Encoder signal A
3	DI + 0.1	Encoder signal B
4	DI + 0.2	Encoder signal N
5	DI + 0.3	Length measurement
6	DI + 0.4	Reference point switch
7	DI + 0.5	-
8	DI + 0.6	-
9	DI + 0.7	-
10	-	Not connected.
11	-	Not connected
12	DI + 1.0	-
13	DI + 1.1	-
14	DI + 1.2	-
15	DI + 1.3	-
16	DI + 1.4	-
17	DI + 1.5	-
18	DI + 1.6	-
19	DI + 1.7	-
20	1 M	Chassis ground
21	2 L+	24 V power supply for the outputs
22	DO + 0.0	-
23	DO + 0.1	-
24	DO + 0.2	-
25	DO + 0.3	-
26	DO + 0.4	-
27	DO + 0.5	-
28	DO + 0.6	-
29	DO + 0.7	-
30	2 M	Ground
31	3 L+	24 V power supply for the outputs
32	DO + 1.0	Digital output Q0
33	DO + 1.1	Digital output Q1
34	DO + 1.2	Digital output Q2
35	DO + 1.3	Digital output Q3
36	DO + 1.4	-
37	DO + 1.5	-
38	DO + 1.6	-
39	DO + 1.7	-
40	3 M	Ground

4.1.4 Connecting Components

Procedure

1. Switch off the power supply to all components.
2. Connect the voltage supply for the digital inputs and outputs:
   – 24 V at X2, pins 1, 21 and 31
   – Ground to X2, pins 20, 30 and 40
3. Connect the 24 V encoder and switches to the 24 V power supply.
4. Connect the encoder signals and the required switches (X2, pins 2 to 6 and pin 20). You can connect bounce-free switches (24 V P action) or non-contact sensors/BERO (2- or 3-wire proximity switches) to the digital inputs "Length measurement" and "Reference point switch".
5. Connect the power section to the power supply.
6. Connect the power section cables (X2, pins 32 to 35 and pin 40).
7. Strip the insulation material on the shielded cables and bind the cable shield to the shield connection element. Use the shield terminal elements for this.

Note

The CPU does not detect the failure of a digital input. You can detect an encoder failure by activating the actual value monitoring (see Drive parameters (Page 104)).

Such a failure might have the following causes:

- Digital input failure
- Wire break
- Faulty encoder
- Faulty power section

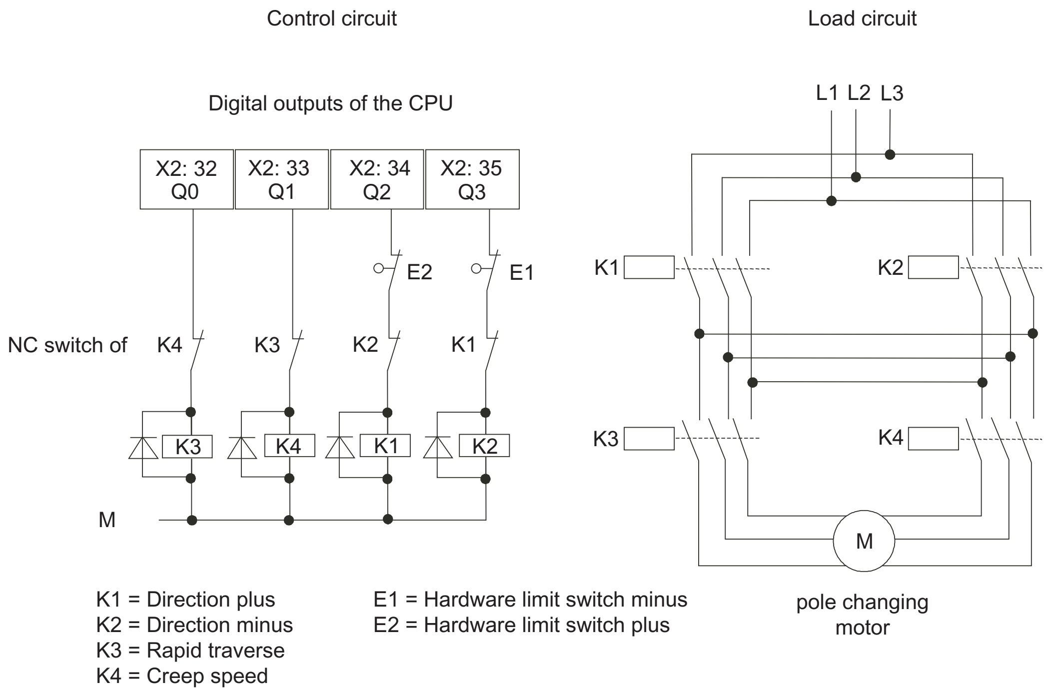
4.1.5 Circuit-Breaker for Digital Outputs

Description

CPU 314C-2 DP, PN/DP, PtP has 4 digital outputs for the positioning mode. The power section is controlled via the digital outputs. The function of the digital outputs depends on the control mode used (see Section Drive parameters (Page 104)). Select the control mode in the configuration software.

Output	Control mode			
	1	2	3	4
Q0	Rapid speed	Rapid/ Creep speed	Rapid speed	Rapid speed plus
Q1	Creep speed	Position reached	Creep speed	Creep speed plus
Q2	Run plus	Run plus	Run plus	Rapid speed minus
Q3	Run minus	Run minus	Run minus	Creep speed minus

The figure below shows you the control and power circuits of a power section. The functions of the digital outputs correspond with control mode 1.



Operating Principle of the Circuit-Breaker

The contactors K1 and K2 control the operating direction of the motor. They are interlocked by the NC contacts K2 and K1. The hardware limit switches E1 and E2 represent the minus/plus limit switches. The motor is switched off when it overruns one of these limit switches.

The contactors K3 (rapid speed) and K4 (creep speed) toggle the motor speed. They are interlocked by the NC contacts K4 and K3.

CAUTION
Damage to assets can occur: A short-circuit in the power network can occur if the power contactors are not interlocked. The power contactor interlock is shown in the figure above.

4.2 Parameter configuration

4.2.1 Basics of Parameter Configuration

Basics

You can adapt the parameters for the positioning function to your specific application. You can assign the parameters with two parameter types:

- Module parameters

  These are basic settings that are specified once and no longer changed while the process is running. These parameters are described in this section.

  – You assign these parameters in the parameter assignment screens (in HW Config).

  – They are stored in the system memory of the CPU.

  – You cannot modify these parameters when the CPU is in RUN mode.

- SFB parameters

  Parameters that need to be changed during operation are located in the instance DB of the system function block (SFB). The SFB parameters are described in Section Positioning with Digital Outputs (Rapid/Creep Speed) (Page 116).

  – You assign these parameters offline in the DB Editor or online in the user program.

  – They are stored in the work memory of the CPU.

  – You can modify these parameters in the user program while the CPU is in RUN state.

Parameter assignment screens

You can assign the module parameters in the **parameter assignment screens**:

- General
- Addresses
- Basic parameters
- Drive
- Axis
- Encoders
- Diagnostics

The parameter assignment screens are self-explanatory. You can find the description of the parameters in the following sections and in the integrated help for the parameter assignment screens.

Note

You cannot assign parameters for the positioning technology if you have assigned channel 0 or channel 1 for the counting technology.

4.2.2 Configuration with the Parameter Assignment Screen

Requirements

Prerequisite for calling the parameter assignment screen is that you have created a project in which you can save your parameters.

Procedure

1. Start the SIMATIC Manager and call HW Config in your project.
2. Double-click on the "Positioning" submodule of your CPU. The "Properties" dialog box opens.
3. Assign the parameters to the "Positioning" submodule and exit the parameter assignment screen with "OK".
4. Save your project in HW Config with "Station > Save and Compile".
5. Download the parameter data to your CPU in STOP mode with "PLC > Download to Module...". The data are now stored in the CPU's system data memory.
6. Switch the CPU to RUN mode.

Online Help

The online help in the parameter assignment screens offers you support when you assign parameters. You have the following options of calling the online help:

- Press the **F1** key in the respective views
- Click on the **Help button** in the various parameter assignment screens.

4.2.3 Basic parameters

Interrupt Selection Parameter

Parameter	Value range	Default
Interrupt selection	• None • Diagnostics	None

Here you can specify whether or not a diagnostic interrupt is to be triggered. The diagnostic interrupt is described in Section Configuring and Evaluating Diagnostic Interrupts (Page 152).

4.2.4 Drive parameters

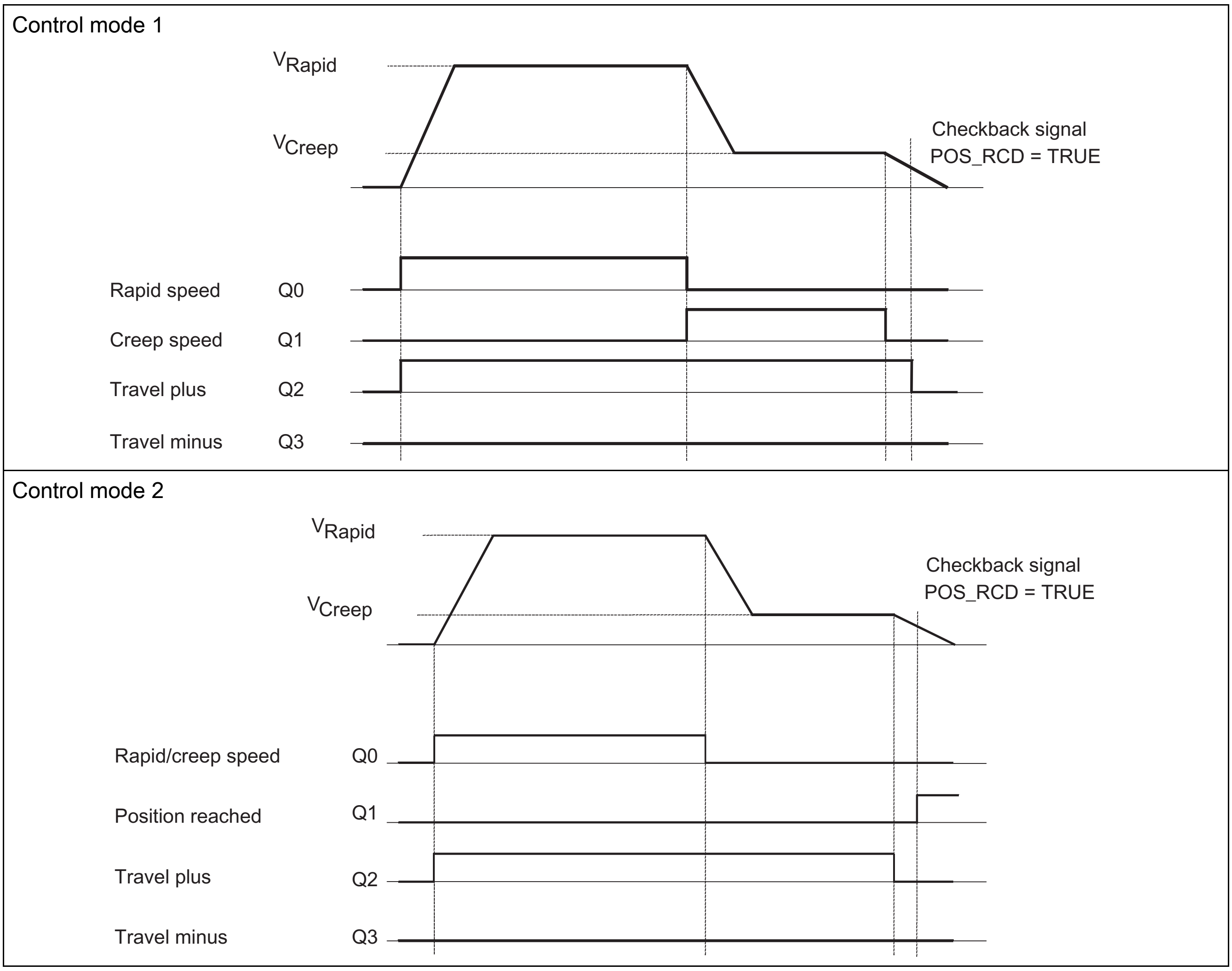
Control Mode Parameter

Parameters	Value range	Default
Control mode	1 - 4	1

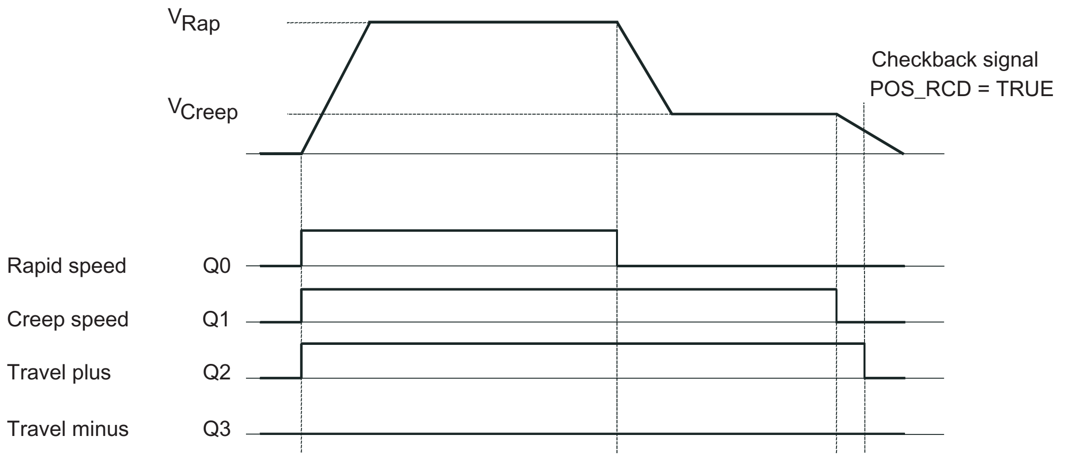
The control mode describes how the 4 digital outputs (Q0 to Q3) operate a connected motor via the converter control.

You can select 4 different control modes: The four control modes are shown in the following figure.

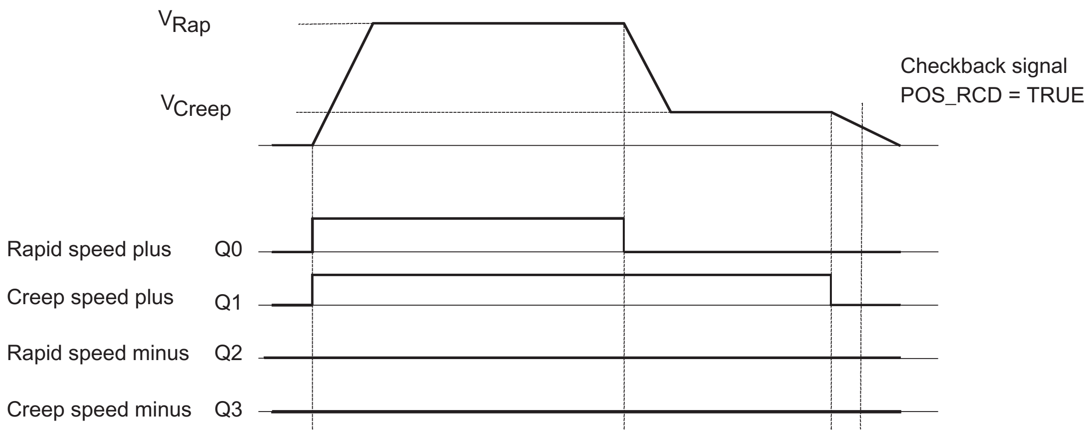
The following figures respectively show the approach in plus direction (POS_RDC = feedback signal).



Control mode 3



Control mode 4



Control mode 1

Control mode 2	Rapid speed		Creep speed		Position reached (POS_RCD)
	Plus direction	Minus direction	Plus direction	Minus direction	
Q0	1	1	0	0	-
Q1	0	0	1	1	-
Q2	1	0	1	0	-
Q3	0	1	0	1	-

Control mode 2

Control mode 1	Rapid speed		Creep speed		Position reached (POS_RCD)
	Plus direction	Minus direction	Plus direction	Minus direction	
Q0	1	1	0	0	0
Q1	0	0	0	0	1
Q2	1	0	1	0	0
Q3	0	1	0	1	0

Control mode 3

Control mode 3	Rapid speed		Creep speed		Position reached (POS_RCD)
	Plus direction	Minus direction	Plus direction	Minus direction	
Q0	1	1	0	0	-
Q1	1	1	1	1	-
Q2	1	0	1	0	-
Q3	0	1	0	1	-

Control mode 4

Control mode 4	Rapid speed		Creep speed		Position reached (POS_RCD)
	Plus direction	Minus direction	Plus direction	Minus direction	
Q0	1	0	0	0	-
Q1	1	0	1	0	-
Q2	0	1	0	0	-
Q3	0	1	0	1	-

Target Range Parameter

Parameters	Value range	Default
Target range	0 up to and including 200 000 000 pulses The CPU rounds up odd values.	50

The target range is arranged symmetrically around the target.

When the value is 0, POS_RCD is not set to TRUE until the target has been overrun or reached to the accuracy of a pulse.

The target range is limited:

- to the rotary axis range of rotary axes
- to the working range of linear axes

Monitoring Time Parameter

Parameters	Value range	Default
Monitoring time	• 0 to 100 000 ms • 0 = No monitoring Rounded up by the CPU in 4-ms steps.	2000

The CPU uses this monitoring time to monitor

- actual value of the position
- target approach

Actual value and target approach monitoring is switched off when the value is set to "0".

Actual Value Parameter

Parameters	Value range	Default
Actual value monitoring	• Yes • No	Yes

The moving axis must cover a distance of least one pulse in specified direction within the monitoring time.

Actual value monitoring is switched on at the start of a run. It remains active until the cut-off position is reached.

Actual value monitoring is switched off when the monitoring time is set to "0".

The run is canceled when the monitoring facility responds.

The CPU does not detect the failure of a digital input. You can enable actual value monitoring for indirect detection of encoder or drive failure.

Target Approach Monitoring Parameter

Parameters	Value range	Default
Target approach monitoring	• Yes • No	No

The axis must reach the target range within the monitoring time after it has reached the cut-off position.

Target approach monitoring is switched off when the monitoring time is set to "0".

Target Range Monitoring Parameter

Parameters	Value range	Default
Target range monitoring	• Yes • No	No

After the target range has been reached, the drive is monitored to check whether it remains at the approached target position or drifts off.

An external error message is generated when the monitoring facility responds. This deactivates the monitoring. Monitoring is not switched on again until the start of a new run.

Maximum Frequency Parameter: Position feedback

Parameters	Value range	Default
Max. frequency: Position feedback	• 60 kHz • 30 kHz • 10 kHz • 5 kHz • 2 kHz • 1 kHz	60 kHz

You can set the maximum frequency of the position feedback signals (encoder signals A, B, N) in fixed steps.

Maximum Frequency Parameter: Accompanying signals

Parameters	Value range	Default
Max. frequency: Accompanying signals	• 60 kHz • 30 kHz • 10 kHz • 5 kHz • 2 kHz • 1 kHz	10 kHz

You can set the maximum frequency of the length measurement and reference point switch signals in fixed steps.

4.2.5 Axis Parameters

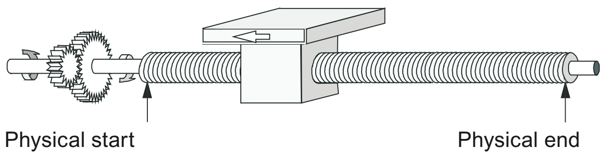
Axis Type Parameter

Parameter	Value range	Default
Axis type	• Linear axis • Rotary axis	Linear axis

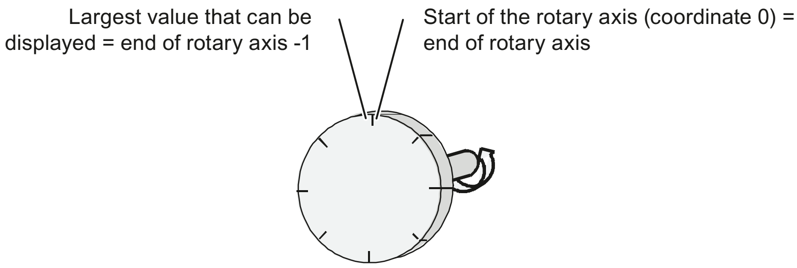
You can control linear axes as well as rotary axes.

Description

The maximum travel range of a **linear axis** is mechanically limited.



The **rotary axis** is not limited by mechanical stops.



Rotation of the rotary axis starts at the "Zero" coordinate and terminates at the coordinate "End of rotary axis -1". The "Zero" coordinate is physically identical (= 0) to the "End of rotary axis". The actual position value display is toggled at this point. It is always displayed with a positive value.

Parameters for software limit switch start/end

Parameters	Value range	Default
Software limit switch start/ end	Software limit switch Start Software limit switch End • -5×10^8 to $+5 \times 10^8$pulses	-100 000 000 +100 000 000

Software limit switches are only used for linear axes.

These software limit switches limit the working range.

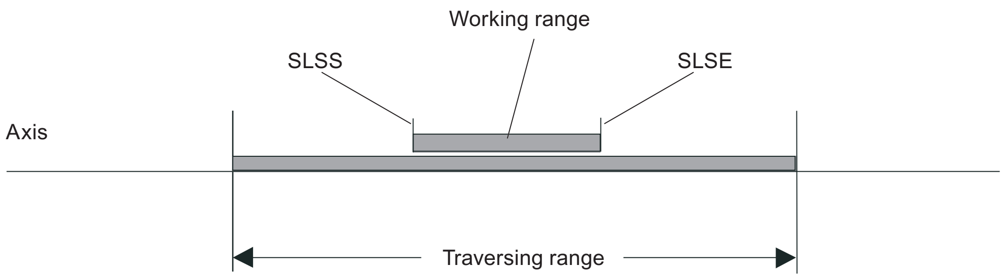
The software limit switches belong to the working range.

The software limit switches are monitored if the axis is synchronized and working range monitoring is switched on.

The axis is not initially synchronized after every STOP-RUN transition of the CPU.

The value of Software Limit Switch Start (SLSS) must always be less than the value of Software Limit Switch End (SLSE).

The working range must lie within the traversing range. This traversing range represents the value range the CPU can process.



SLSS = Software limit switch Start
SLSE = Software limit switch End

End of Rotary Axis Parameter

Parameter	Value range	Default
End of rotary axis	• 1 to 10^9 pulses	100 000

The value of "End of rotary axis" is theoretically the highest possible actual value. Its physical position is identical to the start of the rotary axis (= "0").

The highest rotary axis value displayed is "Rotary axis end -1".

Example: End of rotary axis = 1,000

The display toggles:

- with positive rotary direction from 999 to 0
- with negative rotary direction from 0 to 999

Length Measurement Parameters

Parameter	Value range	Default
Length measurement	• Off • Start/End at the positive edge DI • Start/End at the negative edge DI • Start with positive edge and end with negative edge • Start with negative edge and end with positive edge	Off

Reference Point Coordinate Parameters

Parameter	Value range	Default
Reference point coordinate	-5×10^8 to $+5 \times 10^8$pulses	0

After a STOP-RUN transition of the CPU, the actual value is set to the value of the reference point coordinate .

After a reference point approach, the reference point is assigned the value of the reference point coordinate.

The value of the reference point coordinate must lie within the working range (including the software limit switches) of the linear axis.

The value of the reference point coordinate of the rotary axis must lie within the range 0 to "End of rotary axis -1".

Reference Point Location for Reference Point Switch

Parameter	Value range	Default
Reference point location for reference point switch	• Plus direction (actual values increase) • Minus direction (actual values decrease)	Plus direction

This parameter defines the reference point position with reference to the reference point switch.

Traversing Range Monitoring Parameter

Parameter	Value range	Default
Traversing range monitoring	Yes (set fixed)	Yes

Use traversing range monitoring to check whether the permitted traversing range of -5×10^8 to $+5 \times 10^8$is exceeded. This monitoring feature cannot be switched off (switched on permanently in the "Monitoring" parameter).

Synchronization is canceled and the run is aborted when this monitoring responds.

Parameter Working range monitoring (only with linear axis)

Parameter	Value range	Default
Working range monitoring (only with linear axis)	• Yes • No	Yes

Here, you can specify whether to monitor the working range of the linear axis. In this case, the actual position value is monitored to check whether it is out of range of the software limit switches. This monitoring only affects a synchronized axis.

The coordinates of the software limit switches themselves belong to the working range.

The run is canceled when the monitoring responds.

4.2.6 Encoder Parameters

Overview

Parameter	Value range	Default
Increments per encoder revolution	1 to 2^{23} pulses	1000

The "Increments per encoder revolution" parameter specifies the increments per revolution output at the encoder. Refer to the description of your encoder for information on values.

The CPU evaluates the increments four times (one increment corresponds to four pulses, see Section Incremental encoders (Page 154)).

Count Direction Parameter

Parameter	Value range	Default
Count direction	• Normal • Inverted	Normal

Use the "Count direction" parameter to adapt the direction of path monitoring to the direction of movement of the linear axis. Also, take the rotary direction of all transmission elements into account (for example, couplings and gears).

- Standard = incrementing count pulses = ascending actual values
- Inverted = incrementing count pulses = descending actual values

Missing Pulse (Zero Mark) Monitoring Parameter

Parameter	Value range	Default
Missing pulse (zero mark) monitoring	• Yes • No	No

When zero mark monitoring is enabled, the CPU monitors consistency of the pulse difference between two successive zero mark signals (encoder signal N).

If you have configured an encoder whose pulses per revolution cannot be divided by 10 or 16, zero mark monitoring is automatically switched off, irrespective of the setting in the parameter assignment screen.

Note

The minimum pulse width of the zero mark signal is 8.33 µs (equal to a maximum frequency of 60 kHz).

When you are using an encoder whose zero mark signal is combined with encoder signals A and B using an "AND" operation, the pulse width is reduced by half to 25% of the period. This reduces the maximum frequency for zero mark monitoring to 30 kHz.

Not recognized is:

- Incorrectly assigned number of increments per encoder revolution.
- Failure of the zero mark signal.

Synchronization is canceled and the run is aborted when this monitoring responds.

4.2.7 Configuring the Diagnostics

Diagnostic Interrupts for Monitoring

The responding monitoring feature can trigger a diagnostic interrupt.

Diagnostic Interrupt Enable

Requirement: In the "Basic Parameters" screen, enable diagnostic interrupt and switch on the respective monitoring facility in the "Drive", "Axis" and "Encoder" screens.

Parameters	Value range	Default
Missing pulse (zero mark)	• Yes • No	No
Traversing range	• Yes • No	No
Working range (for linear axes)	• Yes • No	No
Actual value	• Yes • No	No
Target approach	• Yes • No	No
Target range	• Yes • No	No

4.3 Integration into the user program

Procedure

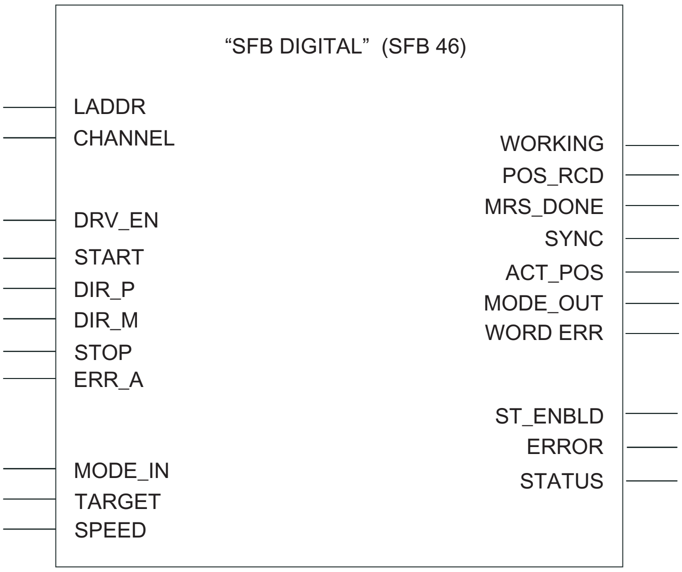
The positioning functions are controlled in your user program. Call the system function block **SFB DIGITAL (SFB 46)**. The SFB is found in the "Standard Library" under "System Function Blocks" > "Blocks".

The chapter below helps you to design a user program for your application.

Calling the SFB

Call the SFB with a corresponding instance DB.

Example: CALL SFB 46, DB22



Note

You must not call an SFB you have configured in your program in another program section under another priority class, because the SFB must not interrupt itself.

Example: It is not allowed to call an SFB both in OB1 and in the interrupt OB.

Instance DB

The SFB parameters are stored in the instance DB. These parameter are described in Section Positioning with Digital Outputs (Rapid/Creep Speed) (Page 116).

You can access these parameters via

- DB number and absolute address in the DB
- DB number and symbolic address in the DB

The essential function parameters are also assigned to the block. You can declare input parameters values directly at the SFB or you can evaluate the output parameters.

4.4 Functions for Positioning with Digital Outputs

4.4.1 Positioning with Digital Outputs (Rapid/Creep Speed)

Overview

Four permanently assigned 24 V digital outputs (Q0-Q3) control the drive. These digital outputs control the direction and speed stages (rapid/creep speed), depending on the configured type of control.

Position feedback is realized via an asymmetric 24 V incremental encoder that is equipped with two signal tracks with a 90° phase shift.

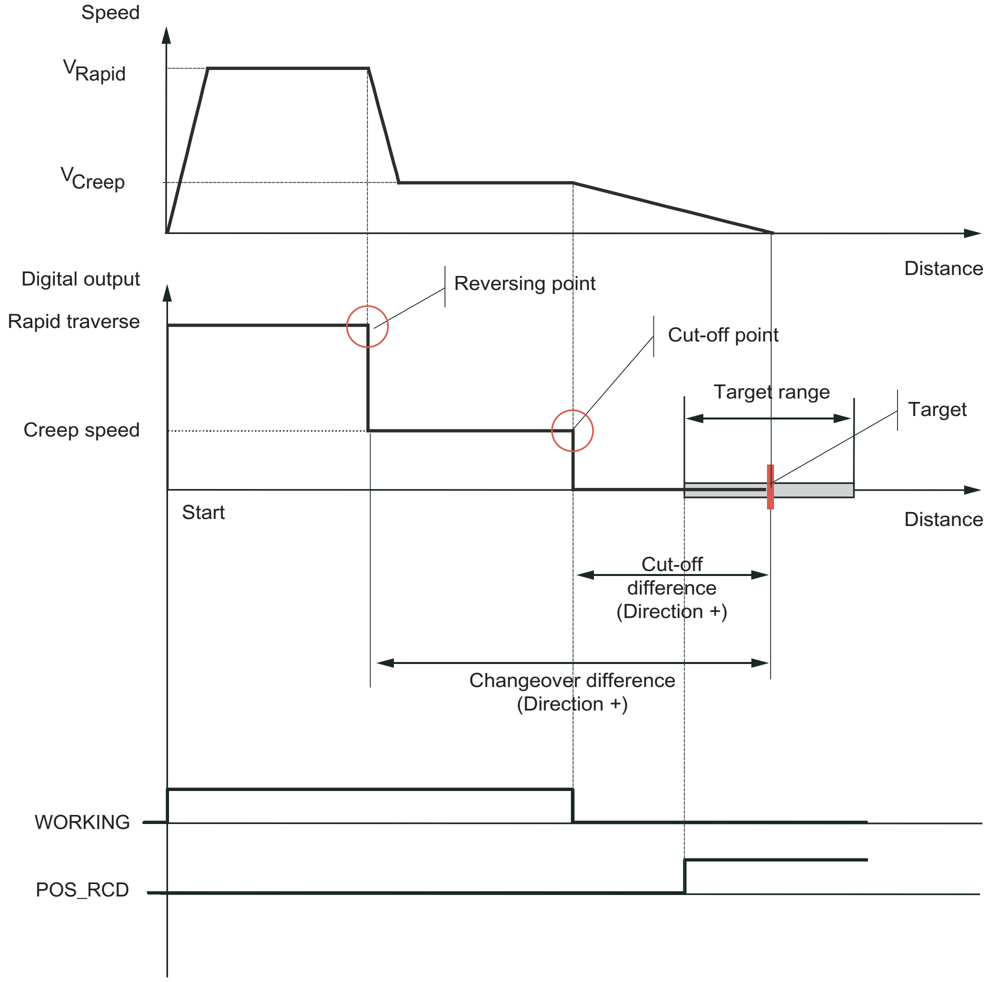
Starting a Run

Start the run with START, DIR_P or DIR_M, depending on the operating mode.

Positioning with digital outputs

The upper section of the figure below shows the run profile. We will simply assume a linear change of the actual speed across the traversing distance.

The lower part of the figure shows the corresponding profile of the digital outputs. Rapid speed and creep speed are determined by a combination of digital outputs 0 and 1 (see Section Drive parameters (Page 104).



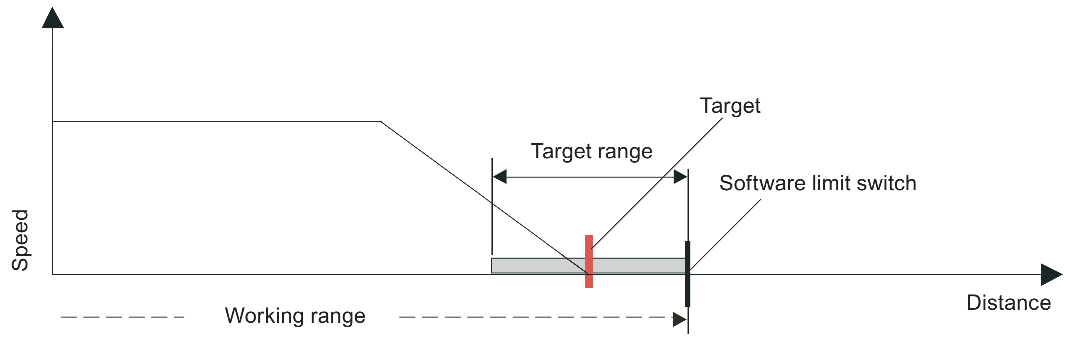
- First, the target is approached at V_{rapid} speed.
- At the **reversing point** the drive switches to creep speed V_{creep}.
- The drive is switched off at the **cut-off position.**
- The reversing point and cut-off point are determined for every target approach, using the values specified by you in the parameters **Changeover difference** and **cut-off difference**. The changeover difference and cut-off difference can be specified differently for forward (Plus direction) and reverse (Minus direction) run.
- The run is terminated (WORKING= FALSE) when the cut-off position is reached. At this point of time a new run can be started.
- The specified target is reached (POS_RCD = TRUE) when the actual position value has reached the **target area.** The "Position reached" signal is not reset if the actual position value drifts off the target area without a new run having been started.

Working range

Determine the working range with the help of the software limit switch coordinates. A run may never exceed the working range of a synchronized linear axis.

You must always specify the run targets according to the working range.

After an axis has overrun the working range, you can only return it in jog mode.



Monitoring functions

The parameter assignment screens help you to enable various monitoring functions individually. When one of the monitoring functions responds, the run is canceled with external error (acknowledge with ERR_A).

Monitoring	Description
Missing pulse (zero mark)	When zero mark monitoring is enabled, the CPU checks the consistency of the pulse difference between two successive zero mark signals. If you have configured an encoder whose pulses per revolution cannot be divided by 10 or 16, zero mark monitoring is automatically switched off, irrespective of the setting in the parameter assignment screen. The minimum pulse width of the zero mark signal is 8.33 μs (corresponds to the maximum frequency of 60 kHz). When you are using an encoder whose zero mark signal is combined with encoder signals A and B using an "AND" operation, the pulse width is reduced by half to 25% of the period. This reduces the maximum frequency for zero mark monitoring to 30 kHz. Not recognized is: • Incorrectly assigned number of increments per encoder revolution. • Failure of the zero mark signal. Response of the CPU to errors: Cancel synchronization, cancel the run.
Traversing range	The CPU uses traversing range monitoring to check whether the permitted traversing range of 5 x 10^8 to +5 x 10^8 is exceeded. This monitoring feature cannot be switched off (switched on permanently in the "Monitoring" parameter).

Monitoring	Description
Working range	The CPU uses traversing range monitoring to check whether the actual value is out of range of the software limit switches. This facility cannot be switched on for monitoring rotary axis positioning. This monitoring only affects a synchronized axis. The coordinates of the software limit switches themselves belong to the working range. Response of the CPU to errors: The run is canceled.
Actual value	Within the monitoring time the axis must travel into the specified direction by at least one pulse. Actual value monitoring is switched on at the start of a run and remains active until the cut-off position has been reached. Actual value monitoring is switched off when the monitoring time is set to "0". The run is canceled when the monitoring responds. Response of the CPU to errors: The run is canceled.
Target approach	The axis must reach the target range within the monitoring time after it has reached the cut-off position. Target approach monitoring is switched off when the set monitoring time is "0". Response of the CPU to errors: The run is canceled, the outputs are switched off.
Target range	After the target range has been reached, the CPU monitors the drive to check whether it stays at or drifts off the approached target position. An external error message is generated when the monitoring function responds. Monitoring is switched off when you acknowledge the external error with ERR_A (positive edge). Monitoring is not switched on again until the start of a new run. Response of the CPU to errors: The run is canceled.

Terminating a Run

There are three different ways to terminate a run:

- Target approach
- Deactivating
- Canceling

Target approach

Home run stands for automatic run termination when the specified target is reached.

In order to reach a specified target, home run is carried out in the operating modes "Relative and absolute incremental approach".

Deactivating

The drive is gradually shut off in the following cases:

- In all operating modes when STOP = TRUE (before the target is reached)
- In "Jog mode", when stopping and when run direction is reversed
- In "Reference point approach" mode, when the synchronization position is detected or at direction reversal

The sequences are analogous to target approach.

Canceling

The run is terminated immediately, disregarding the changeover/cut-off difference. All the relevant outputs of the control mode are deactivated immediately.

Cancellation is possible at any time or at standstill.

The run is canceled in the following cases:

- Canceling the drive enable signal (DRV_EN = FALSE)
- When the CPU switches to STOP mode
- When an external error occurs (Exception: monitoring of the target approach / target range)

Reactions:

- A current or interrupted run is terminated immediately (WORKING = FALSE).
- The last target (LAST_TRG) is set to the actual value (ACT_POS).
- A distance-to-go is deleted, that is, "Relative incremental approach" can not be resumed.
- "Position reached" (POS_RCD) will not be set.

4.4.2 Basic Configuration of SFB DIGITAL (SFB 46)

Overview of Basic Parameters

The parameters which are identical for all operating modes are described in this section. Operating mode-specific parameters are described under the specific modes.

Configure the following SFB input parameters according to your application.

Input parameters

Parameter	Data type	Address (instance DB)	Description	Value range	Default
LADDR	WORD	0	Submodule I/O address you specified in "HW Config". If the I and O addresses are not equal, the lesser of the two addresses must be specified.	CPU-specific	310 hex
CHANNEL	INT	2	Channel number	0	0
STOP	BOOL	4.4	STOP the run. STOP = TRUE can be used to stop/interrupt a run prematurely.	TRUE/FALSE	FALSE
ERR_A	BOOL	4.5	Group error acknowledgment External errors are acknowledged with ERR_A (positive edge).	TRUE/FALSE	FALSE
SPEED	BOOL	12.0	Two speed stages for rapid/creep speed TRUE = Rapid speed FALSE = Creep speed Speed change during run is not possible.	TRUE/FALSE	FALSE

Input Parameters not Assigned to the Block (Static Local Data)

Parameter	Data type	Address (instance DB)	Description	Value range	Default
CHGDIFF_P	DINT	28	Changeover difference plus: "Changeover difference plus" defines the changeover position at which the drive moving in positive direction is toggled from rapid to creep speed.	0 to $+10^8$pulses	1000
CUTOFF-DIFF_P	DINT	32	Cut-off difference plus: The "Cut-off difference plus" defines the cut-off position at which the drive is switched off when operating at creep speed in positive direction.	0 to $+10^8$pulses	100
CHGDIFF_M	DINT	36	Changeover difference minus: "Changeover difference minus" defines the changeover position at which the drive is toggled from rapid to creep speed when moving in negative direction.	0 to $+10^8$pulses	1000
CUTOFF-DIFF_M	DINT	40	Cut-off difference minus: The "Cut-off difference plus" defines the cut-off position at which the drive is switched off when operating at creep speed in negative direction.	0 to $+10^8$pulses	100

Rules for the Changeover/Cut-Off Difference

- Positive and negative values can differ.
- The changeover difference must be greater than/equal to the cut-off difference.
- The cut-off difference must be greater or equal to half the target range.
- Select a sufficient distance between the changeover position and the cut-off position to ensure that the drive's speed can be reduced to creep speed.
- Select a sufficient distance between the changeover position and the target to ensure the drive reaches the target area and comes to a standstill there.
- The distance to be traveled must be at least as high as the cut-off difference
- Changeover/cut-off difference are limited to 1/10 of the travel range ($+10^8$).

Output Parameters

Parameter	Data type	Address (instance DB)	Description	Value range	Default
WORKING	BOOL	14.0	Traverse running	TRUE/FALSE	FALSE
ACT_POS	DINT	16	Actual position value	-5×10^8 to $+5 \times 10^8$pulses	0
MODE_OUT	INT	20	Enabled/set operating mode	0, 1, 3, 4, 5	0
ERR	WORD	22	External error • Bit 2: missing pulse monitoring • Bit 11: traversing range monitoring (always 1) • Bit 12: Working range monitoring • Bit 13: actual value monitoring • Bit 14: target approach monitoring • Bit 15: target range monitoring • The other bits are reserved	Every bit 0 or 1	0
ST_ENBLD	BOOL	24.0	The CPU sets Start Enabled if all of the following conditions are met: • Faultless parameter assignment (PARA = TRUE) • No STOP pending (STOP = FALSE) • No external error has occurred (ERR = 0) • Drive Enable is set (DRV_EN = TRUE) • No positioning run active (WORKING = FALSE). Exception: Jog mode	TRUE/FALSE	TRUE
ERROR	BOOL	24.1	Run start/resume error	TRUE/FALSE	FALSE
STATUS	WORD	26	Error number	0 to FFFF hex	0

Output Parameters not Assigned to the Block (Static Local Data)

Parameter	Data type	Address (instance DB)	Description	Value range	Default
PARA	BOOL	44.0	Axis is configured	TRUE/FALSE	FALSE
DIR	BOOL	44.1	Current/last sense of direction FALSE = Forward (plus direction) TRUE = Reverse (minus direction)	TRUE/FALSE	FALSE
CUTOFF	BOOL	44.2	Drive in cut-off range (from cut-off position to the start of the next run)	TRUE/FALSE	FALSE
CHGOVER	BOOL	44.3	Drive in changeover range (from reaching changeover position to the start of the next run)	TRUE/FALSE	FALSE
DIST_TO_GO	DINT	46	Actual distance to go	-5 x 10^8 to +5 x 10^8pulses	0
LAST_TRG	DINT	50	Last/actual target • Absolute incremental approach: At run start LAST_TRG = actual absolute target (TARGET) • Relative incremental approach: The distance at run start is the distance specified in LAST_TRG = LAST_TRG of the previous run +/- (TARGET).	-5 x 10^8 to +5 x 10^8pulses	0

4.4.3 Jog Mode

Description

In "Jog" mode you can run the drive into plus or minus direction. A target is not specified.

Requirements

- You have assigned the module parameters via parameter assignment screens and downloaded them to the CPU (PARA = TRUE).
- You have assigned the basic parameters of the SFB as described in Section Basic Configuration of SFB DIGITAL (SFB 46) (Page 121).
- No external error ERR has occurred. You must acknowledge queued external errors with ERR_A (positive edge).
- Start enable ST_ENBLD = TRUE.
- You can operate both a synchronized (SYNC = TRUE) and a non-synchronized (SYNC = FALSE) axis in jog mode.

Starting/Stopping the Run

You start the drive by setting control bit DIR_P or DIR_M.

- At every SFB call, these two control bits DIR_P and DIR_M are evaluated to check for logical level changes.
- If both control bits are FALSE the run is ramped down.
- If both control bits are TRUE the run is also ramped down.
- The axis moves in the corresponding direction when one of the control bits is set to TRUE.

Procedure

1. Assign the following **input parameters** of the SFB as specified in the "Setting" column:

Parameter	Data type	Address (instance DB)	Description	Value range	Default	Setting
DRV_EN	BOOL	4.0	Drive enable	TRUE/FALSE	FALSE	TRUE
DIR_P	BOOL	4.2	Jog mode, into plus direction (positive edge)	TRUE/FALSE	FALSE	DIR_P or DIR_M = TRUE
DIR_M	BOOL	4.3	Jog mode, into minus direction (positive edge)	TRUE/FALSE	FALSE	
MODE_IN	INT	6	Operating mode, 1 = "Jog mode"	0, 1, 3, 4, 5	1	1

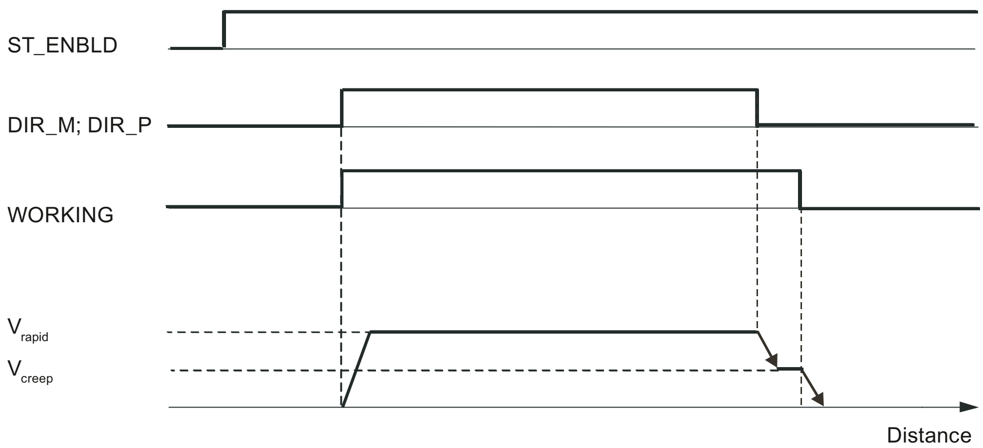
2. Call the SFB.

Result

The **output parameters** of the SFB provide the following information:

Parameters	Data type	Address (instance DB)	Description	Value range	Default
WORKING	BOOL	14.0	Traverse running	TRUE/FALSE	FALSE
ACT_POS	DINT	16	Current actual position value	-5×10^8 to $+5 \times 10^8$ pulses	0
MODE_OUT	INT	20	Active/set operating mode	0, 1, 3, 4, 5	0

- WORKING = TRUE is set immediately after the run has started. When you reset the direction bits DIR_P or DIR_M or set STOP = TRUE, the run is terminated (WORKING = FALSE).
- If an error occurred when the SFB call was interpreted, WORKING = FALSE and ERROR is set to TRUE. The precise error cause is then indicated with the STATUS parameter (see Section Error Lists (Page 156)).
- In jog mode, ST_ENBLD is always set to TRUE.
- "Position reached" (POS_RCD) will not be set.



4.4.4 Reference Point Approach

4.4.4.1 Reference Point Approach – Operation Principle

Description

After the CPU is switched on position value ACT_POS is not referenced to the mechanical position of the axis.

In order to assign a reproducible encoder value to the physical position, a reference (synchronization) must be established between the axis position and the encoder value. The synchronization is carried out by assigning a position value to a known position (reference point) of the axis.

Reference Point Switch and Reference Point

You require a reference point switch and a reference point at the axis to be able to perform a reference point approach.

- You require the **reference point switch** to ensure that the reference signal always has the same reference point (zero mark) and for changing over to reference point approach speed. You can use a BERO switch, for example. The signal length of the reference point switch must be high enough for the axis to reach reference point approach speed before it moves out of range of the reference point switch.
- **Reference point** is the next encoder zero mark after leaving the reference point switch. The axis is synchronized at the reference point and the feedback signal SYNC is set to TRUE. The reference point is assigned the reference coordinates you have specified via parameter assignment screens.

A reference point approach must always be started in the direction of the reference point switch. Otherwise, the axis travels into range limits because it is not synchronized and, thus, software limit switches do not exist.

By starting the reference point approach at the reference point switch you always ensure that the axis is starts into the direction of the switch (see Example 3).

Note

For rotary axes: Because of the required reproducibility of the reference point, the corresponding zero mark of the encoder must always be at the same physical position. Therefore, the "End of rotary axis" value and the number of "Increments per encoder revolution" must represent a proportional integral. Example: Four encoder revolutions are proportional to one revolution of the end of the rotary axis. In this case, the zero marks lie at 90, 180, 270 and 360 degrees.

The minimum pulse width of the zero mark signal is 8.33 μs (corresponds to the maximum frequency of 60 kHz).

When you are using an encoder whose zero mark signal is combined with encoder signals A and B using an "AND" operation, the pulse width is reduced by half to 25% of the period. This reduces the maximum counting frequency to 30 kHz during referencing.

Reference Point Position

With regard to the reference point position (**Zero mark signal**) during a reference point approach, you must distinguish between:

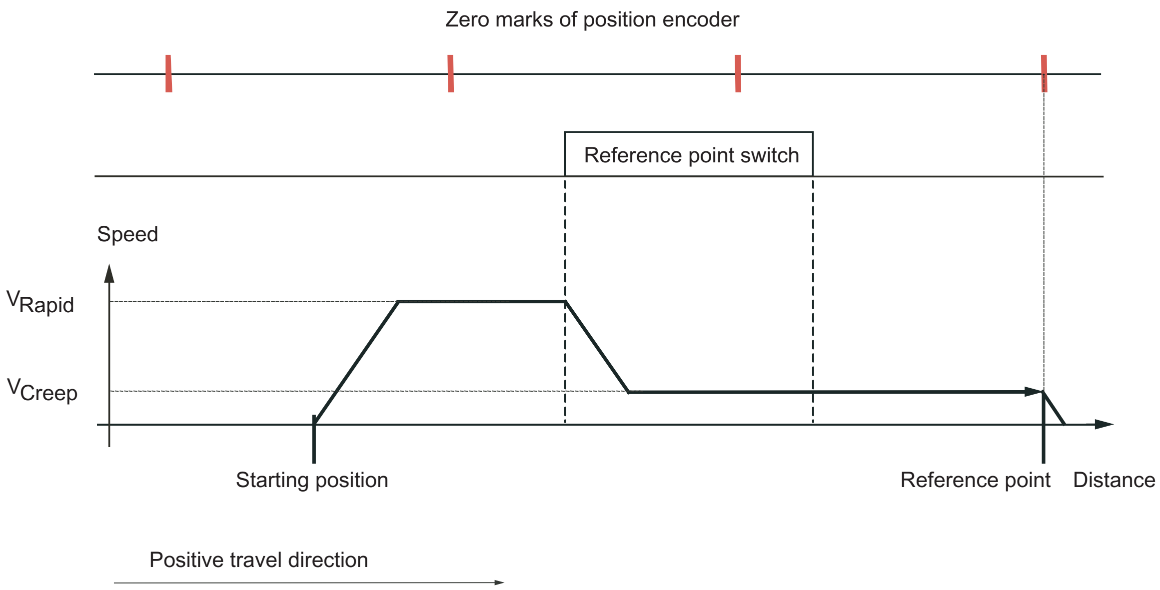
- The reference point position is referenced to the reference point switch in plus direction.
- The reference point position is referenced to the reference point switch in minus direction.

Make this setting via parameter assignment screens in the parameter "Reference point location for reference point switch".

The different situations for a reference point approach are determined by the direction of the run start and by the position of the reference point:

Example 1:

- Start direction plus
- Reference point location for reference point switch in plus direction



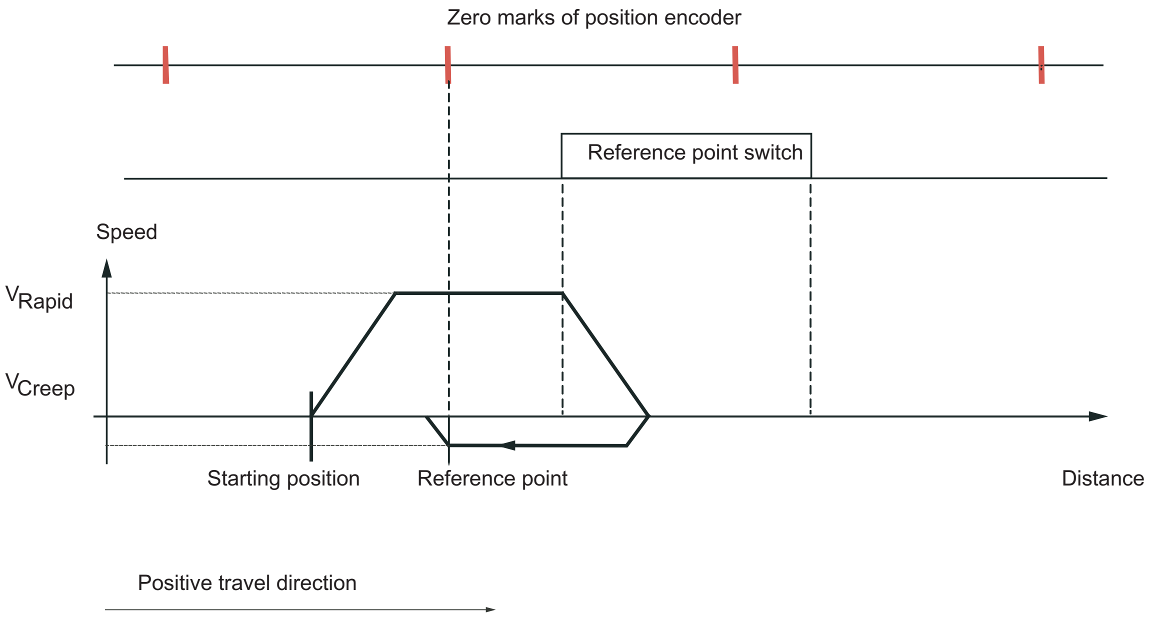
The reference point switch is approached at rapid traverse.

The drive is then toggled to creep speed.

After it has left the reference point switch, the drive is switched off at the next encoder zero mark.

Example 2:

- Start direction plus
- Reference point approach for reference point switch in minus direction



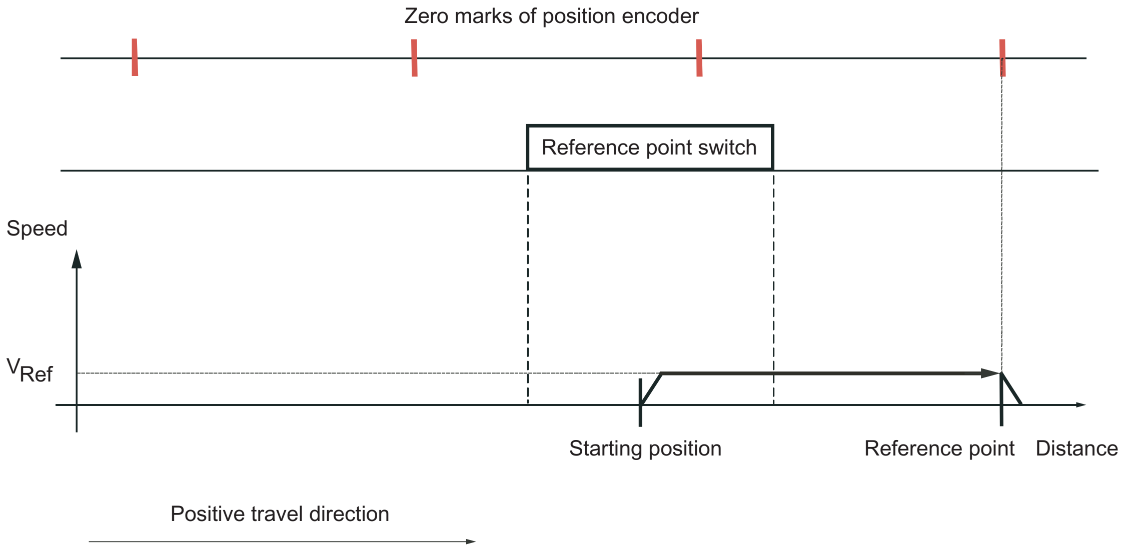
The reference point switch is approached at rapid traverse.

Then, the drive is switched to creep speed and the direction is reversed.

After it has left the reference point switch, the drive is switched off at the next encoder zero mark.

Example 3:

- Start position is at the reference point switch
- Start direction minus
- Reference point approach for reference point switch in plus direction



Carried out at creep speed.

Run is executed towards the direction you have assigned in the parameter assignment screens with the "Reference point location for reference point switch" parameter, irrespective of the direction specified at the SFB.

After it has left the reference point switch, the drive is switched off at the next encoder zero mark.

4.4.4.2 Reference Point Approach – Procedure

Prerequisite for Reference Point Approach

- Encoder with zero mark or, when using an encoder without zero mark, a switch for the reference point signal.
- You have connected the reference point switch (connector X2, pin 6).
- You have assigned the module parameters via parameter assignment screens and downloaded them to the CPU (PARA = TRUE).
- You have assigned the basic parameters of the SFB as described in Section Basic Configuration of SFB DIGITAL (SFB 46) (Page 121).
- No external error ERR has occurred. You must acknowledge queued external errors with ERR_A (positive edge).
- Start enable ST_ENBLD = TRUE.

Procedure

1. Assign the following **input parameters** of the SFB as specified in the "Setting" column:

Parameters	Data type	Address (instance DB)	Description	Value range	Default	Setting
DRV_EN	BOOL	4.0	Drive enable	TRUE/FALSE	FALSE	TRUE
DIR_P	BOOL	4.2	Reference point approach, plus direction (positive edge)	TRUE/FALSE	FALSE	DIR_P or DIR_M = TRUE
DIR_M	BOOL	4.3	Reference point approach, minus direction (positive edge)	TRUE/FALSE	FALSE	
MODE_IN	INT	6	Operating mode, 3 = "Reference point approach"	0, 1, 3, 4, 5	1	3

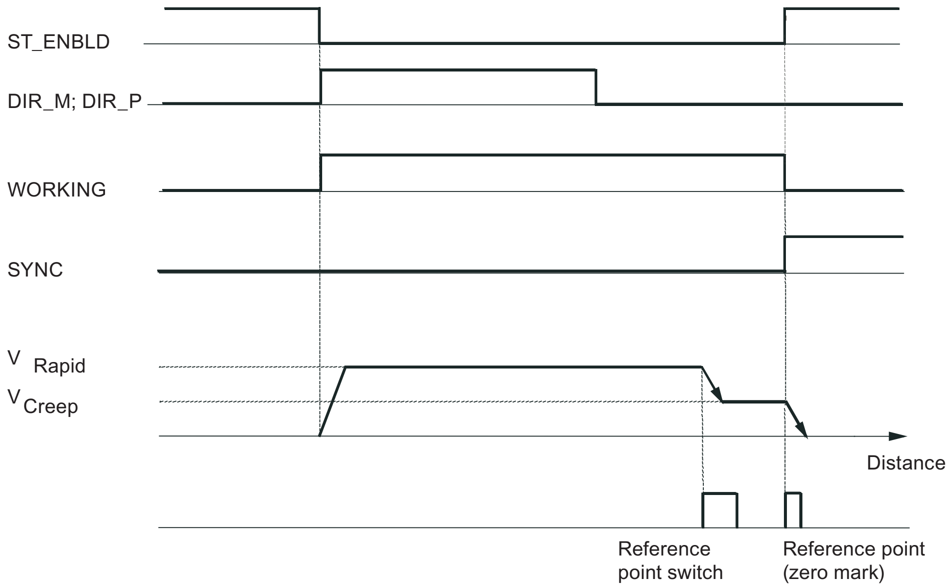
2. Call the SFB.

Result

The **output parameters** of the SFB provide the following information:

Parameters	Data type	Address (instance DB)	Description	Value range	Default
WORKING	BOOL	14.0	Run is busy	TRUE/FALSE	FALSE
SYNC	BOOL	14.3	SYNC = TRUE: Axis is synchronized	TRUE/FALSE	FALSE
ACT_POS	DINT	16	Actual position value	$-5x10^8$ to $+5x10^8$ pulses	0
MODE_OUT	INT	20	Enabled/set operating mode	0, 1, 3, 4, 5	0

- WORKING= TRUE is set and SYNC = FALSE immediately after the run has started. The status of WORKING is reset to FALSE after the reference point has been reached. SYNC = TRUE if executed without error.
- You must reset the direction bit (DIR_P or DIR_M) before you start the next run.
- If an error occurred when the SFB call was interpreted, WORKING = FALSE and ERROR is set to TRUE. The precise error cause is then indicated with the STATUS parameter (see Section Error Lists (Page 156)).
- "Position reached" (POS_RCD) will not be set.



What the Operating Mode Affects

- A possibly existing synchronization is cleared (SYNC = FALSE) at the start of the reference point approach.
- At the positive edge of the reference point (zero mark) the actual position is set to the value of the reference point coordinate and the feedback signal SYNC is set.
- The working range is determined at the axis.
- All points within the working range maintain their original coordinates, but have new physical positions.

4.4.5 Relative incremental approach mode

Description

In "Relative incremental approach" mode, the drive moves into a specified direction by a relative distance, starting at the last target (LAST_TRG).

Starting point is not the actual position, but rather the last specified target (LAST_TRG). This prevents cumulative positioning inaccuracy. The actual target is indicated at the parameter LAST_TRG after positioning has started.

Requirements

- You have assigned the module parameters via parameter assignment screens and downloaded them to the CPU (PARA = TRUE).
- You have assigned the basic parameters of the SFB as described in Section Basic Configuration of SFB DIGITAL (SFB 46) (Page 121).
- No external error ERR has occurred. You must acknowledge queued external errors with ERR_A (positive edge).
- Start enable ST_ENBLD = TRUE.
- A "Relative incremental approach" is possible with a synchronized (SYNC = TRUE) and also with a non-synchronized (SYNC = FALSE) axis.

Specifying the Traversing Distance

When specifying the traversing distance of linear axes, note the following:

- The traversing distance must be greater than or equal to the cut-off difference.
- A new run is not started if the traversing distance is less than/equal to half of the target range. The mode is terminated immediately without an error.
- The target range must lie within the working range.

Procedure

1. Assign the following **input parameters** of the SFB as specified in the "Setting" column:

Parameters	Data type	Address (instance DB)	Description	Value range	Default	Setting
DRV_EN	BOOL	4.0	Drive enable	TRUE/FALSE	FALSE	TRUE
DIR_P	BOOL	4.2	Run in plus direction (positive edge)	TRUE/FALSE	FALSE	DIR_P or DIR_M = TRUE
DIR_M	BOOL	4.3	Run in minus direction (positive edge)	TRUE/FALSE	FALSE	
MODE_IN	INT	6	Operating mode, 4 = "Relative incremental approach"	0, 1, 3, 4, 5	1	4
TARGET	DINT	8	Distance in pulses (only positive values allowed)	0 to 109	1000	xxxx

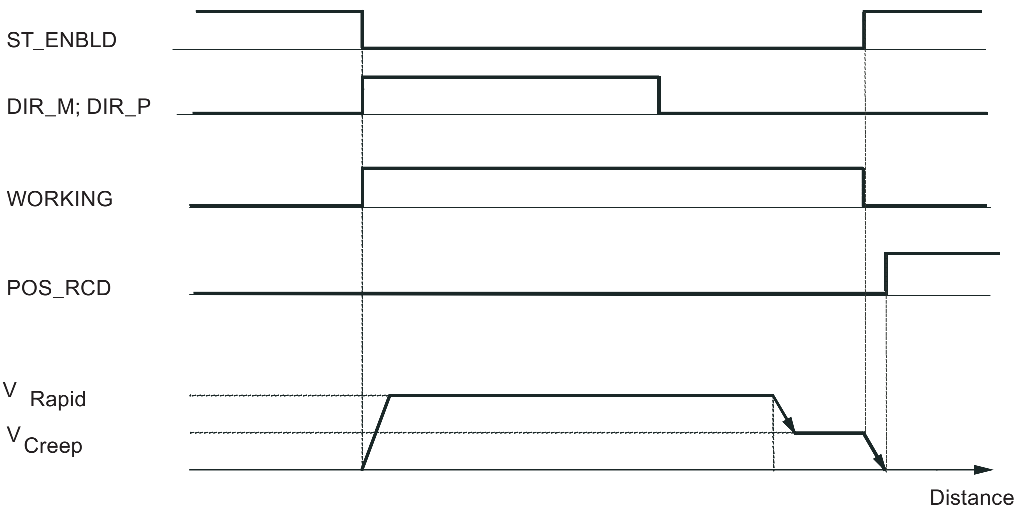
2. Call the SFB.

Result

The **output parameters** of the SFB provide the following information:

Parameters	Data type	Address (instance DB)	Description	Value range	Default
WORKING	BOOL	14.0	Run is busy	TRUE/FALSE	FALSE
POS_RCD	BOOL	14.1	Position reached	TRUE/FALSE	FALSE
ACT_POS	DINT	16	Actual position value	-5×10^8 to $+5 \times 10^8$pulses	0
MODE_OUT	INT	20	Enabled/set operating mode	0, 1, 3, 4, 5	0

- WORKING = TRUE is set immediately after the run has started. WORKING is reset to FALSE at the cut-off point. POS_RCD = TRUE is set when the specified target is reached.
- You must reset the direction bit (DIR_P or DIR_M) before you start the next run.
- If an error occurred when the SFB call was interpreted, WORKING = FALSE and ERROR is set to TRUE. The precise error cause is then indicated with the STATUS parameter (see Section Error Lists (Page 156)).



Stopping a run and target range not reached

When a run is stopped with STOP = TRUE and if the cut-off range has not been reached (distance to go is greater than the cut-off difference), you have the following options depending on the subsequent operating mode/job.

Option	Response
Continuing the run into the same direction	Run parameters will not be interpreted. The axis travels to the target point of the stopped run (LAST_TRG).
Continuing the run in opposite direction	Run parameters will not be interpreted. The axis moves to the starting point of the stopped run.
Starting a new "Absolute incremental approach"	The axis moves to the specified absolute target.
Job "Delete distance to go"	The distance to go (difference between target and actual value) will be deleted. The run parameters are interpreted again at the start of a "Relative incremental approach" and the axis moves to the current actual position value.

4.4.6 Absolute incremental approach mode

Description

In "Absolute incremental approach" mode you approach absolute target positions.

Requirements

- You have assigned the module parameters via parameter assignment screens and downloaded them to the CPU (PARA = TRUE).
- You have assigned the basic parameters of the SFB as described in Section Basic Configuration of SFB DIGITAL (SFB 46) (Page 121).
- No external error ERR has occurred. You must acknowledge queued external errors with ERR_A (positive edge).
- Start enable ST_ENBLD = TRUE.
- The axis is synchronized (SYNC = TRUE).

Specifying the Target

Note the following when you specify the target:

- The traversing distance must be greater than or equal to the cut-off difference.
- A new run is not started if the traversing distance is less than/equal to half of the target range. The mode is terminated immediately without an error.
- The target range for a linear axis must lie within the working range, for a rotary axis it must lie between "0" and "End of rotary axis" -1.

Run Start

- The linear axis is always started with START = TRUE.
- You must specify the sense of direction of rotary axes:
  - DIR_P = TRUE: Run in plus direction
  - DIR_M = TRUE: Run in minus direction
  - START = TRUE: The axis approaches the target along the shortest possible distance.

  The CPU calculates the sense of direction, taking into account the actual distance to go, the actual value and the target.
  Run is started in reverse direction, if the shortest distance is less than/equal to the cut-off difference and greater than/equal to half of the target range.
  If the travel difference is equal in both directions, the axis moves in the plus direction.

Procedure

1. Assign the following **input parameters** of the SFB as specified in the "Setting" column:

<table>
<tr><th>Parameters</th><th>Data type</th><th>Address (instance DB)</th><th>Description</th><th>Value range</th><th>Default</th><th>Setting</th></tr>
<tr><td>DRV_EN</td><td>BOOL</td><td>4.0</td><td>Drive enable</td><td>TRUE/FALSE</td><td>FALSE</td><td>TRUE</td></tr>
<tr><td>START</td><td>BOOL</td><td>4.1</td><td>Run start (positive edge)</td><td>TRUE/FALSE</td><td>FALSE</td><td rowspan="3">START or DIR_P or DIR_M = TRUE</td></tr>
<tr><td>DIR_P</td><td>BOOL</td><td>4.2</td><td>Run in plus direction (positive edge)</td><td>TRUE/FALSE</td><td>FALSE</td></tr>
<tr><td>DIR_M</td><td>BOOL</td><td>4.3</td><td>Run in minus direction (positive edge)</td><td>TRUE/FALSE</td><td>FALSE</td></tr>
<tr><td>MODE_IN</td><td>INT</td><td>6</td><td>Operating mode, 5 = "Absolute incremental approach"</td><td>0, 1, 3, 4, 5</td><td>1</td><td>5</td></tr>
<tr><td>TARGET</td><td>DINT</td><td>8</td><td>Target in pulses</td><td>Linear axis:
-5×10^8 to $+5 \times 10^8$
Rotary axis:
0 to end of rotary axis -1</td><td>1000</td><td>xxxx</td></tr>
</table>

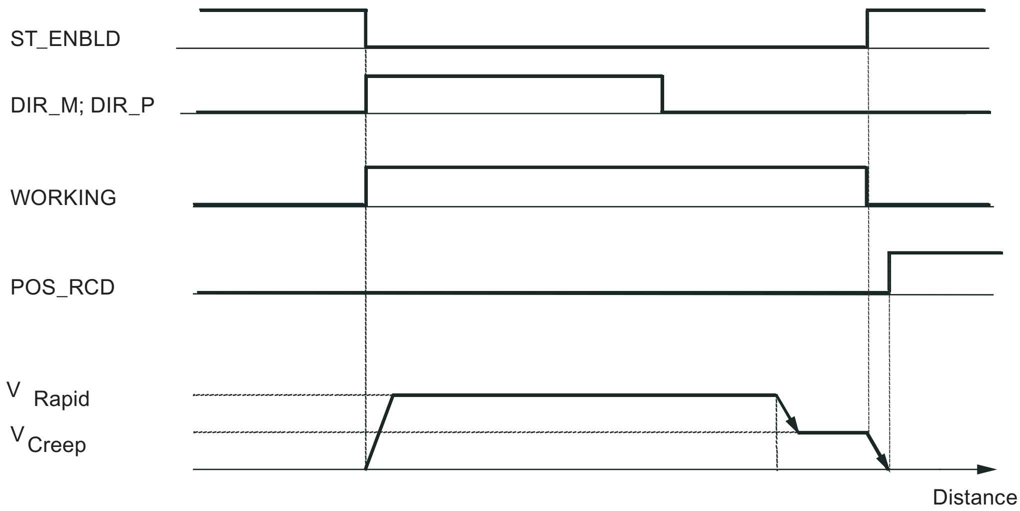
2. Call the SFB.

Result

The **output parameters** of the SFB provide the following information:

Parameters	Data type	Address (instance DB)	Description	Value range	Default
WORKING	BOOL	14.0	Run has started	TRUE/FALSE	FALSE
POS_RCD	BOOL	14.1	Position reached	TRUE/FALSE	FALSE
ACT_POS	DINT	16	Actual position value	-5×10^8 to $+5 \times 10^8$ pulses	0
MODE_OUT	INT	20	Enabled/set operating mode	0, 1, 3, 4, 5	0

- WORKING = TRUE is set immediately after the run has started. WORKING is reset to FALSE at the cut-off point. POS_RCD = TRUE is set when the specified target is reached.
- You must reset the direction bit (DIR_P or DIR_M) before you start the next run.
- If an error occurred when the SFB call was interpreted, WORKING = FALSE and ERROR will be set to TRUE. The precise error cause is then indicated with the STATUS parameter (see Section Error Lists (Page 84)).



Stopping a Run and Target Range not Reached

When a run is stopped with STOP = TRUE and if the cut-off range has not been reached (distance to go is greater than the cut-off difference), you have the following options depending on the subsequent operating mode/job.

Option	Response
Starting a new "Absolute incremental approach"	The axis moves to the specified absolute target.
Continuing the run into the same direction with "Relative incremental approach" mode	Run parameters will not be interpreted. The axis travels to the target point of the stopped run (LAST_TRG).
Continuing the run into the reverse direction with "Relative incremental approach" mode	Run parameters will not be interpreted. The axis moves to the starting point of the stopped run.
Job "Delete distance to go"	The distance to go (difference between target and actual value) will be deleted. The run parameters are interpreted again at the start of a "Relative incremental approach" and the axis moves to the current actual position value.

4.4.7 Specifying the Reference Point

Description

You can also use the "Set reference point" request to synchronize the axis without performing a reference point approach.

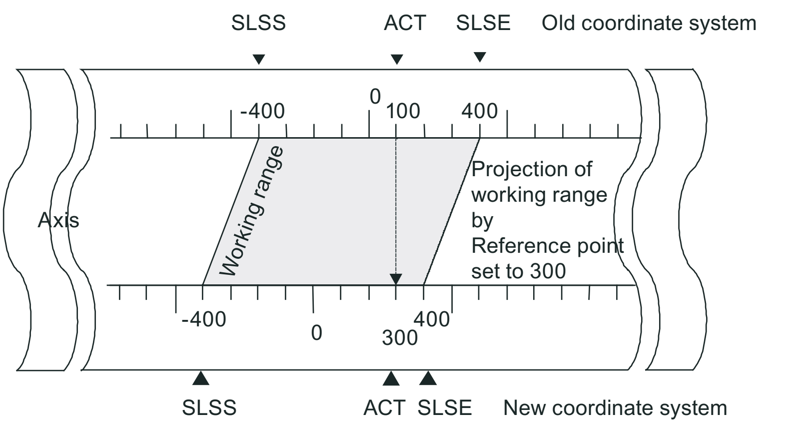
After the job has been executed, the actual position coordinate has the value you have assigned via the JOB_VAL parameter.

- Linear axis: The reference point coordinate must lie within the working range (including the software limit switches).
- Rotary axis: The reference point coordinate must lie within the range from 0 to "End of rotary axis" -1.

This does not change the reference point coordinates you have entered in the parameter assignment screens.

Example of Setting the Reference Point:

- The actual position value is 100. The software limit switches (SLSS, SLSE) are at the positions -400 and 400 (working range).
- The "Set reference point" request is executed with the value JOB_VAL = 300.
- The actual value is then to coordinate 300. The software limit switches and the working range have the same coordinates as prior to the job. However, they are now physically shifted left by 200.



	SLS	ACT	SLE
Old coordinate system	-400	100	400
New coordinate system	-400	300	400

Requirements

- You have assigned the module parameters via parameter assignment screens and downloaded them to the CPU (PARA = TRUE).
- You have assigned the basic parameters of the SFB as described in Section Basic Configuration of SFB DIGITAL (SFB 46) (Page 121).
- The last job must be finished (JOB_DONE = TRUE).
- The last positioning operation must be ended (WORKING = FALSE).

Procedure

1. Assign the following **input parameters** (accessible via instance DB) as specified in the "Setting" column:

Parameter	Data type	Address (instance DB)	Description	Value range	Default	Setting
JOB_REQ	BOOL	66.0	Job request (positive edge)	TRUE/FALSE	FALSE	TRUE
JOB_ID	INT	68	Job, 1 = "Set reference point"	1, 2	0	1
JOB_VAL	DINT	72	Job parameters for the coordinates of the reference point	-5×10^8 to $+5 \times 10^8$ pulses	0	xxxx

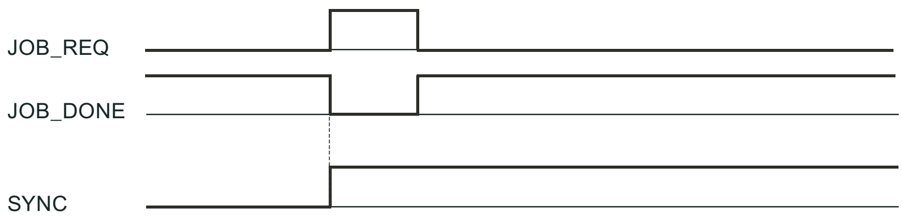
2. Call the SFB.

Result

The **output parameters** of SFB (JOB_DONE, JOB_ERR, JOB_STAT accessible via instance DB) provide the following information:

Parameter	Data type	Address (instance DB)	Description	Value range	Default
SYNC	BOOL	14.3	Axis is synchronized	TRUE/FALSE	FALSE
JOB_DONE	BOOL	66.1	New job can be started	TRUE/FALSE	TRUE
JOB_ERR	BOOL	66.2	Faulty job	TRUE/FALSE	FALSE
JOB_STAT	WORD	70	Job error number (see Section Error Lists (Page 156)).	0 to FFFF hex	0

- The job is immediately processed after the SFB is called. JOB_DONE is set to FALSE for the duration of one SFB cycle.
- You must reset the job request (JOB_REQ).
- SYNC = TRUE if the job was processed without error.
- JOB_ERR = TRUE if an error occurred. The precise error cause is then indicated in JOB_STAT.
- A new job can be started with JOB_DONE = TRUE.



Effects of the job

- The actual position value will be set to the value of the reference point coordinate and the status signal SYNC is set.
- The working range is shifted physically along the axis.
- All points within the working range maintain their original coordinates, but have new physical positions.

Simultaneous Call of a Job and a Positioning Operation

When a positioning operation and a job are initiated simultaneously, the job is executed first. Positioning is not executed if the job ends with an error.

A job initiated while a run is busy will be ended with an error.

4.4.8 Deleting the Distance-to-Go

Description

After a target run (absolute or relative incremental approach) the pending distance-to-go (DIST_TO_GO) can be deleted with the job.

Requirements

- You have assigned the module parameters via parameter assignment screens and downloaded them to the CPU (PARA = TRUE).
- You have assigned the basic parameters of the SFB as described in Section Basic Configuration of SFB DIGITAL (SFB 46) (Page 121).
- The last job must be finished (JOB_DONE = TRUE).
- The last positioning operation must be ended (WORKING = FALSE).

Procedure

1. Assign the following SFB **input parameters** (accessible via instance DB) as specified in the "Setting" column:

Parameter	Data type	Address (instance DB)	Description	Value range	Default	Setting
JOB_REQ	BOOL	66.0	Job trigger (positive edge)	TRUE/FALSE	FALSE	TRUE
JOB_ID	INT	68	Job, 2 = "Delete distance-to-go"	1, 2	0	2
JOB_VAL	DINT	72	None	-	0	Any

2. Call the SFB.

Result

The **output parameters** of the SFB (accessible via instance DB) provide the following information:

Parameter	Data type	Address (instance DB)	Description	Value range	Default
JOB_DONE	BOOL	66.1	New job can be started	TRUE/FALSE	TRUE
JOB_ERR	BOOL	66.2	Faulty job	TRUE/FALSE	FALSE
JOB_STAT	WORD	70	Job error number (see Section Error Lists (Page 156)).	0 to FFFF hex	0

- The job is processed immediately after the SFB is called.
  JOB_DONE is set to FALSE for the duration of one SFB cycle.
- You must reset the job request (JOB_REQ).
- JOB_ERR = TRUE if an error occurred. The precise error cause is then indicated in JOB_STAT.
- A new job can be started with JOB_DONE = TRUE.

Simultaneous Call of a Job and a Positioning Operation

When a positioning operation and a job are initiated simultaneously, the job is executed first. Positioning is not executed if the job ends with an error.

A job initiated while a run is busy will be ended with an error.

4.4.9 Length measurement

Description

With "Length measurement" you can determine the length of a work piece. The start and stop of length measurement is edge-triggered at the digital input "Length measurement".

At the SFB you are given the coordinates for length measurement start and end as well as the measured length.

With the help of the parameter assignment screens (Parameter "Length measurement") you can switch length measurement on and off and also determine the type of edge:

- Off
- Start/end at the positive edge
- Start/end at the negative edge
- Start with positive edge and end with negative edge
- Start with negative edge and end with positive edge

Requirements

- You have assigned the module parameters via parameter assignment screens and downloaded them to the CPU (PARA = TRUE).
- You have assigned the basic parameters of the SFB as described in Section Basic Configuration of SFB DIGITAL (SFB 46) (Page 121).
- You have connected a bounce-free switch to the digital input "Length measurement" (connector X2, pin 5).
- "Length measurement" is possible with synchronized (SYNC = TRUE) as well as non-synchronized (SYNC = FALSE) axis.

Procedure

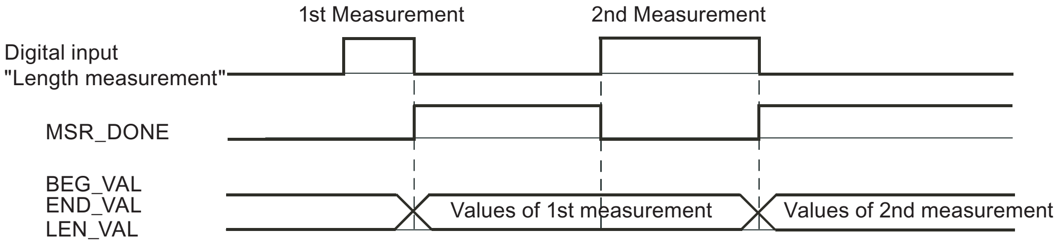
- An edge at the digital input starts length measurement.
- MSR_DONE is reset at the start of length measurement.
- MSR_DONE = TRUE is set at the end of the length measurement.
- The SFB then outputs the following values:
  - Start of length measurement: BEG_VAL
  - End of length measurement: END_VAL
  - Measured length: LEN_VAL

At the end of one length measurement until the end of the next length measurement the values are available at the block.

The **output parameters** of the SFB (BEG_VAL, END_VAL, LEN_VAL are accessible via instance DB) provide following information:

Parameter	Data type	Address (instance DB)	Description	Value range	Default
MSR_DONE	BOOL	14.2	End of length measurement	TRUE/FALSE	FALSE
BEG_VAL	DINT	54	Actual position value, start of length measurement	-5×10^8 to $+5 \times 10^8$pulses	0
END_VAL	DINT	58	Actual position value, end of length measurement	-5×10^8 to $+5 \times 10^8$pulses	0
LEN_VAL	DINT	62	Measured length	0 to 10^9 pulses	0

The figure below shows the signal profile for a length measurement of the type: Start/End of length measurement at the positive/negative edge.



Note

When referencing during a length measurement; the change of the actual value is taken into account as follows:

Example: A length measurement is performed between two points with a distance of 100 pulses. When referencing during the length measurement, the coordinates are shifted by + 20. This results in a measured length of 120.

4.5 Adapting Parameters

4.5.1 Important Safety Rules

Important Note

Please note the following warning points:

WARNING

Harm to health or damage to assets is to be expected.

To avoid injury and damage to property, please note the following:

- Install an **Emergency-Off switch** in the area of the control system. This is the only possible way for you to ensure that the system can be safely switched off in case of control system failure.
- Install **Hardware limit switches** which affect the drive converters of all drives directly.
- Make sure that **no one has access to system areas** in which moving parts exist.
- **Parallel control and monitoring** via your program and STEP 7 interface can cause conflicts, the effects of which are not clear.

4.5.2 Determining the Module Parameters and Their Effects

Increments per encoder revolution

The "**Increments per encoder revolution**" parameters of the connected incremental encoder are found on its rating plate or in the specifications sheet. The technology evaluates the encoder signals in quadruple mode. Four pulses represent one encoder increment. All distance specifications are referenced to pulse units.

Control mode

The parameter "**Control mode**" describes the signals of the four digital outputs that control the drive. You must specify this parameter according to your physical control circuit (circuit-breaker). A description of the control modes can be found in Section Drive parameters (Page 104).

Monitoring time

You must select a time of sufficient length in the "**Monitoring time**" parameter to ensure that the drive can supersede the startup holding torque of the axis within the specified time.

This monitoring time is also used to monitor target approach. That is, the drive must reach the target range within this time, after it has reached the cut-off point.

Count direction

Use the **"Count direction"** parameter to adapt the direction of path monitoring to the direction of movement of the linear axis. Also, take the rotary direction of all transmission elements into account (for example, couplings and gears).

- "Normal" means, the incrementing count pulses correspond to rising actual position values.
- "Inverted" means, the incrementing count pulses correspond to descending actual position values.

4.5.3 Effect of the SFB Parameters

CHGDIFF_P and CHGDIFF_M

The parameters "CHGDIFF_P" (Changeover difference, in plus direction) and "CHGDIFF_M" (Changeover difference, in minus direction) define the position at which the drive is toggled from rapid to creep speed.

If the difference is set too high, positioning is not optimized over time because creep speed runtime is unnecessarily extended .

CUTOFFDIFF_P and CUTOFFDIFF_M

The parameters "CUTOFFDIFF_P" (Cut-off difference in plus direction) and "CUTOFFDIFF_M" (Cut-off difference in minus direction) specify the number pulses to go before the drive is switched off at the target approach.

Take into consideration that this distance varies according to the load on the drive.

The drive is switched off at a speed higher than creep speed if the changeover/cut-off difference is too low. The result is inexact positioning.

The difference of changeover/cut-off difference for the respective direction should at least be proportional to the distance the drive actually requires to reach creep speed. Here you must also take into account the load on the drive.

4.5.4 Checking the Monitoring Time

Requirements

- Your system is wired correctly.
- The positioning submodule is configured, the parameters have been assigned and the project is loaded.
- For example, you have loaded the provided sample program "Digital 1, Getting started".
- The CPU is in RUN state

Checklist

Step	What to do	✓
1	Verify the wiring	
	• Verify correct wiring of the outputs. • Verify correct wiring of the encoder inputs.	❑ ❑
2	Check the axis motion	
	• "Jog" the drive at creep speed in plus or minus direction. The actual sense of direction DIR must correspond with the specified direction. If this is not the case, change the module parameter "Count direction".	❑
3	Synchronize the axis	
	• Select the job "Set reference" (JOB_ID = 1). Enter the desired coordinate at the actual axis position as JOB_VAL (e. g. 0 pulses). Execute synchronization by setting JOB_REQ to TRUE. The coordinate you have specified is shown as actual position value and the synchronization bit SYNC is set. Evaluate (JOB_STAT) a reported error (JOB_ERR = TRUE). If required, correct your specified coordinate and repeat the job for setting the reference.	❑

Step	What to do	✓
4	**Check the changeover/cut-off differences**	
	• Perform a "Relative or absolute incremental approach" run at rapid speed towards the specified target (TARGET) that is further away from the current position than specified in the changeover difference. • Note the individual positioning phases (acceleration, constant run, deceleration, target approach).	❑
	Increase the changeover difference until it can be seen that the drive moves towards cut-off position at creep speed.	❑
	If the configured target range is not reached, reduce the cut-off difference and repeat the run until the target range is reached.	❑
	If the configured target range is overrun, increase the cut-off difference and repeat the run until the target range is not overrun anymore.	❑
	• Now optimize the changeover difference.	❑
	Reduce the changeover difference without changing the cut-off difference and repeat the run. You can reduce the changeover difference to a value at which you can hardly notice the drive moving at creep speed, that is, it is definitely moving at creep speed when reaching the cut-off point. Positioning accuracy remains unchanged as long as the drive is switched off at creep speed. A further reduction of the cut-off difference does not make sense.	

4.6 Error Handling and Interrupts

4.6.1 Error Messages at the System Function Block (SFB)

Overview

The SFB indicates the errors listed in the table below.

Except for system errors, all errors are specified in closer detail via an error number, which is available as an output parameter in the SFB.

Type of error	Errors are displayed via SFB parameters	The error number is displayed in the SFB parameters
Operating mode error	ERROR = TRUE	STATUS
Job error	JOB_ERR = TRUE	JOB_STAT
External error	ERR > 0	ERR
System error	BIE = FALSE	-

Operating Mode Error (ERROR = TRUE)

This error occurs

- upon general parameter assignment errors at the SFB (e.g. use of incorrect SFB)
- at run start/resume These errors occur during the interpretation of operating mode parameters.

When an error is detected, output parameter ERROR is set to TRUE.

The **STATUS** parameter indicates the cause of error. The possible error numbers can be found in Section Error Lists (Page 156).

Job Error (JOB_ERR = TRUE)

Job errors can only occur during the interpretation/execution of a job.

When an error is detected, output parameter JOB_ERR is set to TRUE.

The error cause is indicated at the **JOB_STAT** parameter. The possible error numbers can be found in section .

External Error (ERR)

The system monitors the run, the traversing range, and the connected I/O. Prerequisite is here that you have enabled monitoring in the "Drive", "Axis" and "Encoder" parameter assignment screens.

An external error is signaled when the monitoring responds.

External errors can occur at any time, regardless of the started functions.

You must acknowledge external errors with ERR_A (positive edge).

External errors are indicated at the SFB parameter ERR (WORD) by setting a bit.

Monitoring	ERR	Bit in ERR-WORD
Missing pulse (zero mark)	0004 hex	2
Traversing range	0800 hex	11
Working range	1000 hex	12
Actual value	2000 hex	13
Target approach	4000 hex	14
Target range	8000 hex	15

The detection of an external error ("incoming" and "outgoing") can also trigger a diagnostic interrupt (see Section Configuring and Evaluating Diagnostic Interrupts (Page 152)).

System error

A system error is indicated with BIE = FALSE.

A system error is triggered by:

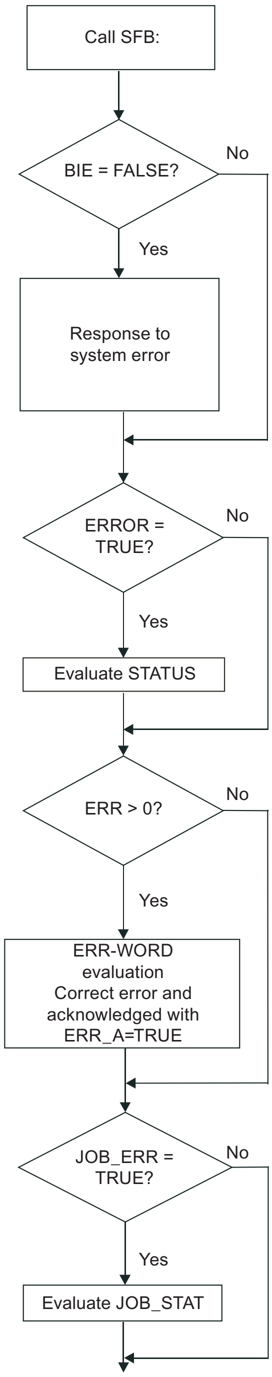
- Read/write access errors at the instance DB
- Multiple calls of the SFB

4.6.2 Error Evaluation in the User Program

Procedure

1. Call the error handling routine "Error evaluation" (see the view).
2. Query the specific error types in successive order.
3. If required, jump to the error reaction method that is specifically adapted to your application.

Error evaluation:



4.6.3 Configuring and Evaluating Diagnostic Interrupts

Basics

On occurrence of the following errors you can trigger a diagnostic interrupt:

- Parameter assignment error (Module data)
- External error (Monitoring)

The diagnostic interrupt is displayed in the event of incoming as well as outgoing errors.

In your user program, you can immediately respond to errors with the help of a diagnostic interrupt.

Procedure

1. Enable diagnostic interrupt in the "Basic parameters" dialog of the parameter assignment screens.
2. In the "Drive", "Axis" and "Encoder" parameter assignment screens, switch on the individual monitoring functions that should trigger a diagnostic interrupt when an error occurs.
3. In the parameter assignment screen "Diagnostics", enable diagnostic interrupts for each monitoring facility individually.
4. Incorporate the diagnostic interrupt OB (OB 82) in your user program.

Response to an Error with Diagnostic Interrupt

- Positioning is canceled.
- The CPU operating system calls OB82 in the user program.

Note

If the corresponding OB is not loaded the CPU switches to STOP when an interrupt is triggered.

- The CPU switches on the SF LED.
- The error is reported in the diagnostics buffer of the CPU as "incoming". An error is not indicated as "outgoing" until all pending errors are cleared.

How a Diagnostic Interrupt is Evaluated in the User Program

After a diagnostic interrupt is triggered, you can evaluate OB 82 to check which diagnostic interrupt is pending.

- If the module address of the "Positioning" submodule was entered in OB 82, byte 6 + 7 (OB 82_MDL_ADDR), the diagnostic interrupt was triggered by the positioning function of your CPU.
- Bit 0 of byte 8 in OB 82, bit 0 (Faulty module) is set as long as any errors are queued.
- In OB 82, bit 0 of byte 8 will be reset after all errors have been reported "outgoing".
- You can determine the precise error cause by evaluating data record 1, byte 8 and 9. To do this, you must call SFC 59 (read data record).
- Acknowledge the error with ERR_A.

Data record 1, byte 8	Description:	JOB_STAT	ERR
Bit 0	not used	-	-
Bit 1	not used	-	-
Bit 2	Missing pulse*	-	X
Bit 3	not used	-	-
Bit 4	not used	-	-
Bit 5	not used	-	-
Bit 6	not used	-	-
Bit 7	not used	-	-

Data record 1, byte 9	Description:	JOB_STAT	ERR
Bit 0	Configuration error	X	-
Bit 1	not used	-	-
Bit 2	not used	-	-
Bit 3	Traversing range monitoring	X	X
Bit 4	Working range monitoring	X	X
Bit 5	Actual value monitoring*	X	X
Bit 6	Target approach monitoring*	X	X
Bit 7	Target range monitoring*	X	X
* subsequent errors trigger an incoming and then automatically an outgoing interrupt.			

4.7 Installation of Examples

Using Examples

The examples (program and description) are found on the CD-ROM included in your documentation. You can also download them from the Internet. The project consists of several commented S7 programs of various complexity and aim.

The Readme.wri on the CD describes how to install the samples. After installation the examples are stored in the catalog ...\STEP7\EXAMPLES\ZDt26_03_TF_____31xC_Pos.

4.8 Specifications

4.8.1 Incremental encoders

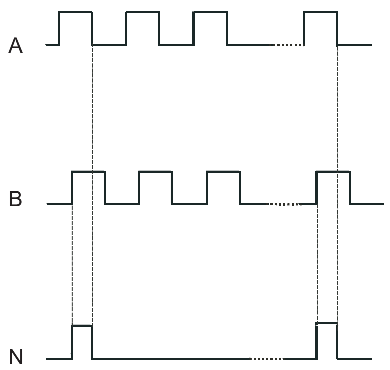
Connectable Incremental Encoders

Supported are asymmetrical 24 V incremental encoders which have two pulse tracks with an electrical phase difference of 90°, with/without zero mark.

Inputs for encoder connection	Pulse width min/ pulse pause min	Input frequency max.	Cable length max. (at max. input frequency)
Encoder signal A, B	8 µs	60 kHz	50 m
Encoder signal N (zero mark signal)	8 µs	60 kHz/30 kHz[1]	50 m
[1] If you are using an encoder whose zero mark signal is combined with encoder signals A and B using an "AND" operation, the pulse width is reduced by half to 25% of the period. In order to maintain the minimum pulse width, the maximum counting frequency must be reduced to 30 kHz.			

Signal Evaluation

The view shows the signal profile of encoders with asymmetrical output signals:



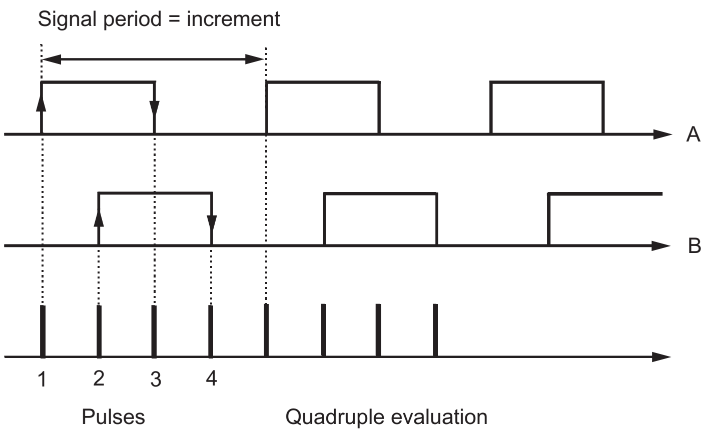
The CPU internally generates a logical AND link of the zero mark signal and the A and B track signals.

For referencing, the CPU uses the positive edge at the zero mark.

The CPU counts in positive direction if the signal A transition leads signal B.

Increments

An increment identifies a signal period of the two encoder track signals A and B. This value is specified on the rating plate of the encoder and/or in the technical specifications for the encoder.

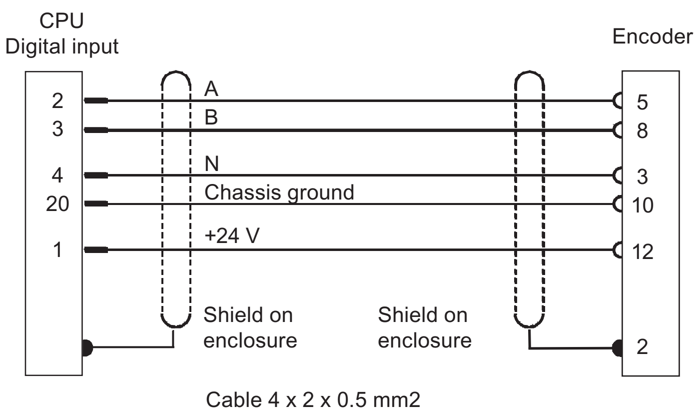


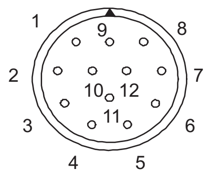
Pulses

The CPU evaluates all 4 edges of the track signals A and B (see the view) with every increment (quadruple evaluation). i.e. one encoder increment is proportional to four pulses.

Wiring Diagram of the Incremental Encoder Siemens 6FX 2001-4 (Up = 24 V;HTL)

The figure below shows the wiring diagram of the incremental encoder Siemens 6FX 2001-4xxxx (Up = 24 V; HTL):





12-pin circular connector socket Siemens 6FX 2003-0CE12

Connection side (solder side)

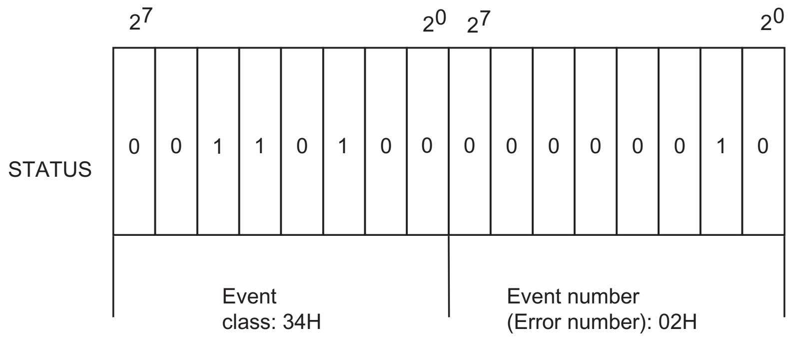
4.8.2 Error Lists

Basics

If an error occurs, an error ID is output at the SFB parameters STATUS or JOB_STAT. The error number consists of an event class and number.

Example

The view below shows the content of the STATUS parameter for the event "Incorrect target specified" (Event class: 34H, Event number 02H).



Error numbers at SFB Parameter "Status"

Event class 32 (20H): "SFB error"		
Event no.	**Event text**	**Remedy**
(20)01H	Incorrect SFB	Use SFB 46
(20)04H	Incorrect channel number (CHANNEL)	Set channel number "0"

Event class 48 (30H): "General run start error"		
Event no.	**Event text**	**Remedy**
(30)01H	Run job rejected because of faulty job in the same SFB call	Correct the respective JOB parameters
(30)02H	It is not allowed to modify MODE_IN while the drive is still in motion.	Wait until the current positioning operation has ended.
(30)03H	Unknown operating mode (MODE_IN)	Permitted is: 1 (Jog mode), 3 (reference point approach), 4 (relative incremental approach) and 5 (absolute incremental approach).
(30)04H	Start requests may only be set one at a time.	Permissible start requests are DIR_P, DIR_M or START
(30)05H	START is only allowed in "Absolute incremental approach" mode	Start the run with DIR_P or DIR_M
(30)06H	DIR_P or DIR_M is not allowed for a linear axis and in "Absolute incremental approach" mode	Start the run with START.
(30)07H	Axis not synchronized	"Absolute incremental approach" is only possible if the axis is synchronized.
(30)08H	Moving out of working range	Run is only allowed into the direction of the working range.

Event class 49 (31H): "Run start error (Start enable)"		
Event no.	**Event**	**Remedy**
(31)01H	Start not enabled because the axis is not configured.	Configure the "Position" submodule via HW Config
(31)02H	Start not enabled, because drive enable is not set	Set "Start enable" at the SFB (DRV_EN = TRUE)
(31)03H	Start not enabled because STOP is set.	Clear the STOP at the SFB (STOP = FALSE)
(31)04H	Start not enabled because the axis currently performs a positioning run (WORKING = TRUE).	Wait until the current positioning operation is terminated
(31)05H	Start not enabled because at least one pending error has not been acknowledged.	First, eliminate and acknowledge all external errors, then restart the run.

Event class 50 (32H): "Run start error (speed/acceleration)"		
Event no.	**Event**	**Remedy**
(32)01H	Incorrect speed specification SPEED	Only "Creep speed" (0) and "Rapid traverse" (1) is allowed when positioning with digital outputs.

Event class 51 (33H): "Run start error (Changeover/switch-off difference)"		
Event no.	Event	Remedy
(33)01H	Changeover/cut-off differences greater than 10^8 are not permitted	Specify a changeover/cut-off difference of maximum 10^8
(33)03H	A changeover difference less than the cut-off difference is not allowed	The changeover difference must be greater than/equal to the cut-off difference.
(33)04H	Cut-off difference too small	The cut-off difference must have at least the length of half the target range.

Event class 52 (34H): "Run start error (default target/distance)"		
Event no.	Event	Remedy
(34)01H	Default target out of working range	With a linear axis and absolute incremental approach the default target must lie within the range of the software limit switch (inclusive).
(34)02H	Incorrect target specification	For the rotary axis the specified target must be greater than 0 and less than the end of rotary axis.
(34)03H	Incorrect distance specification	With relative incremental approach the distance to be traversed must be positive.
(34)04H	Incorrect distance specification	The resulting absolute target coordinate must be greater than -5×10^8.
(34)05H	Incorrect distance specification	The resulting absolute target coordinate must be greater than 5×10^8.
(34)06H	Incorrect distance specification	The resulting absolute target coordinate must lie within the working range (+/-half of the target range)

Event class 53 (35H): "Run start error (traversing distance)"		
Event no.	Event	Remedy
(35)01H	Traversing distance too long	The target coordinate + actual distance to go must be greater than/equal to -5×10^8
(35)02H	Traversing distance too long	The target coordinate + actual distance to go must be less than/equal to 5×10^8
(35)03H	Traversing distance too short	The traversing distance into plus direction must be greater than the specified cut-off difference for the plus direction
(35)04H	Traversing distance too short	The traversing distance into minus direction must be greater than the specified cut-off difference for the minus direction
(35)05H	Traversing distance too short or the limit switch has already been overrun in plus direction	The last approachable target into plus direction (working range/traversing range limits) is too close to the actual position
(35)06H	Traversing distance too short or the limit switch has already been overrun in minus direction	The last approachable target into minus direction (working range/ traversing range limits) is too close to the actual position

Error numbers at the SFB Parameter "JOB_STAT"

Event class 64 (40H): "General job execution error"		
Event no.	**Event**	**Remedy**
(40)01H	Axis not configured	Configure the "Position" submodule via HW Config.
(40)02H	Job not possible because positioning is still running	Jobs can only be executed if no positioning run is active. Wait until WORKING = FALSE and then repeat the job.
(40)04H	Unknown job	Check the Job ID and repeat the job.

Event class 65 (41H): "Error when executing the request to set the reference point"		
Event no.	**Event**	**Remedy**
(41)01H	Reference point coordinate out of working range	With a linear axis the reference point coordinate must not exceed the working range limits.
(41)02H	Incorrect reference point coordinate	With a linear axis, the specified reference point coordinate + actual distance to go must still be greater than/equal to -5 x 10^8.
(41)03H	Incorrect reference point coordinate	With a linear axis, the specified reference point coordinate + actual distance to go must still be less than/equal to 5 x 10^8.
(41)04H	Incorrect reference point coordinate	With a linear axis, the specified reference point coordinate + actual difference to the starting point of the run must still be greater than/equal to -5 x 10^8.
(41)05H	Incorrect reference point coordinate	With a linear axis, the specified reference point coordinate + actual difference to the starting point of the run must still be less than/equal to 5 x 10^8 .
(41)06H	Reference point coordinate out of rotary axis range	With a rotary axis the reference point coordinate must not be less than 0 and greater than/equal to the end of rotary axis.

External Error (ERR)

External errors are indicated at the SFB parameter ERR (WORD) by setting a bit:

Monitoring	ERR	Bit in ERR-WORD
Missing pulse (zero mark)	0004 hex	2
Traversing range	0800 hex	11
Working range	1000 hex	12
Actual value	2000 hex	13
Target approach	4000 hex	14
Target range	8000 hex	15

4.8.3 Module Parameters of the Parameter Assignment Screen – Overview

Introduction

The following tables provide an overview of the module parameters that can be set in the parameter assignment screens.

Basic parameters

Parameter	Value range	Default
Interrupt selection	• None • Diagnostic	None

Drive parameters

Table 4- 2 Control Mode Parameter

Parameter	Value range	Default
Control mode	1 - 4	1

Table 4- 3 Meaning of the Control Modes

Output	Control mode			
	1	2	3	4
Q0	Rapid traverse	Rapid traverse / Creep speed	Rapid traverse	Rapid traverse plus
Q1	Creep speed	Position reached	Creep speed	Creep speed plus
Q2	Travel plus	Travel plus	Travel plus	Rapid traverse minus
Q3	Travel minus	Travel minus	Travel minus	Creep speed minus

Table 4- 4 Additional Drive Parameters

Parameter	Value range	Default
Target range	0 up to and including 200 000 000 pulses The CPU rounds up odd values.	50
Monitoring time	• 0 to 100 000 ms • 0 = No monitoring Rounded up by the CPU in 4-ms steps.	2000
Actual value monitoring	• Yes • No	Yes
Target approach monitoring	• Yes • No	No
Target range monitoring	• Yes • No	No
Max. frequency: Position feedback	60, 30, 10, 5, 2, 1 kHz	60 kHz
Max. frequency: Accompanying signals	60, 30, 10, 5, 2, 1 kHz	10 kHz

Axis Parameters

Parameter	Value range	Default
Axis type	• Linear axis • Rotary axis	Linear axis
Software limit switch start/ end	Software limit switch Start Software limit switch End -5×10^8 to $+5 \times 10^8$pulses	-100 000 000 +100 000 000
End of rotary axis	1 to 10^9 pulses	100 000
Length measurement	• Off • Start/End at the positive edge DI • Start/End at the negative edge DI • Start with positive edge and end with negative edge • Start with negative edge and end with positive edge	Off
Reference point coordinate	-5×10^8 to $+5 \times 10^8$pulses	0
Reference point location for reference point switch	• Plus direction (actual values increase) • Minus direction (actual values decrease)	Plus direction
Monitoring Traversing range	Yes (set fixed)	Yes
Monitoring Working range	• Yes • No	Yes

Encoder Parameters

Parameter	Value range	Default
Increments per encoder revolution	1 to 2^{23} pulses	1000
Count direction	• Normal • Inverted	Normal
Missing pulse (zero mark) monitoring	• Yes • No	No

Diagnostic parameters

Parameter	Value range	Default
Missing pulse (zero mark)	• Yes • No	No
Traversing range	• Yes • No	No
Working range (for linear axes)	• Yes • No	No
Actual value	• Yes • No	No
Target approach	• Yes • No	No
Target range	• Yes • No	No

4.8.4 Parameters for Instance DB of the SFB DIGITAL (SFB 46)

Overview

Parameters	Declaration	Data Type	Address (Instance DB)	Description	Range of values	Default
LADDR	IN	WORD	0	Submodule I/O address you specified in "HW Config". If the I/O addresses are not equal you must specify the lower one of both.	CPU-specific	310 hex
CHANNEL	IN	INT	2	Channel number	0	0
DRV_EN	IN	BOOL	4.0	Drive enable	TRUE/ FALSE	FALSE
START	IN	BOOL	4.1	Run start (positive edge)	TRUE/FALSE	FALSE
DIR_P	IN	BOOL	4.2	Run in plus direction (positive edge)	TRUE/FALSE	FALSE
DIR_M	IN	BOOL	4.3	Run in minus direction (positive edge)	TRUE/FALSE	FALSE
STOP	IN	BOOL	4.4	Stop run	TRUE/FALSE	FALSE
ERR_A	IN	BOOL	4.5	Group error acknowledgment ERR_A is used to acknowledge external errors (positive edge)	TRUE/FALSE	FALSE
MODE_IN	IN	INT	6	Operating mode	0, 1, 3, 4, 5	1
TARGET	IN	DINT	8	Relative incremental approach: Distance in pulses (only positive values allowed)	0 to 10^9	1000
				Absolute incremental approach: Target in pulses	Linear axis: -5×10^8 to $+5 \times 10^8$ Rotary axis: 0 to end of rotary axis -1	
SPEED	BOOL	DINT	12.0	Two speed stages for rapid/creep speed TRUE = Rapid speed FALSE = Creep speed	TRUE/FALSE	FALSE
WORKING	OUT	BOOL	14.0	Run is busy	TRUE/FALSE	FALSE
POS_RCD	OUT	BOOL	14.1	Position reached	TRUE/FALSE	FALSE
MSR_ DONE	OUT	BOOL	14.2	End of length measurement	TRUE/FALSE	FALSE

Parameters	Declaration	Data Type	Address (Instance DB)	Description	Range of values	Default
SYNC	OUT	BOOL	14.3	Axis is synchronized	TRUE/FALSE	FALSE
ACT_POS	OUT	DINT	16	Actual position value	-5×10^8 to $+5 \times 10^8$ pulses	0
MODE_OUT	OUT	INT	20	Enabled/set operating mode	0, 1, 3, 4, 5	0
ERR	OUT	WORD	22	External error Bit 2: missing pulse monitoring Bit 11: Traversing range monitoring (always 1) Bit 12: Working range monitoring Bit 13: actual value monitoring Bit 14: target approach monitoring Bit 15: Target range monitoring The other bits are reserved	Every bit 0 or 1	0
ST_ENBLD	OUT	BOOL	24.0	Start enable	TRUE/FALSE	TRUE
ERROR	OUT	BOOL	24.1	Run start error/ resume error	TRUE/FALSE	FALSE
STATUS	OUT	WORD	26.0	Error ID	0 to FFFF hex	0
CHGDIFF_P	STAT	DINT	28	Changeover difference plus	0 to $+10^8$ Pulses	1000
CUTOFF-DIFF_P	STAT	DINT	32	Cut-off difference plus	0 to $+10^8$ Pulses	100
CHGDIFF_M	STAT	DINT	36	Changeover difference minus	0 to $+10^8$ Pulses	1000
CUTOFF-DIFF_M	STAT	DINT	40	Cut-off difference minus	0 to $+10^8$ Pulses	100
PARA	STAT	BOOL	44.0	Axis is configured	TRUE/FALSE	FALSE
DIR	STAT	BOOL	44.1	Current/last sense of direction FALSE = Forward (plus direction) TRUE = Reverse (minus direction)	TRUE/FALSE	FALSE
CUTOFF	STAT	BOOL	44.2	Drive in cut-off range (from cut-off position to the start of the next run)	TRUE/FALSE	FALSE

Parameters	Declaration	Data Type	Address (Instance DB)	Description	Range of values	Default
CHGOVER	STAT	BOOL	44.3	Drive in changeover range (from reaching changeover position to the start of the next run)	TRUE/FALSE	FALSE
DIST_TO_GO	STAT	DINT	46	Actual distance to go	-5×10^8 to $+5 \times 10^8$ pulses	0
LAST_TRG	STAT	DINT	50	Last/actual target	-5×10^8 to $+5 \times 10^8$ pulses	0
BEG_VAL	STAT	DINT	54	Actual position value, start of length measurement	-5×10^8 to $+5 \times 10^8$ pulses	0
END_VAL	STAT	DINT	58	Actual position value, end of length measurement	-5×10^8 to $+5 \times 10^8$ pulses	0
LEN_VAL	STAT	DINT	62	Measured length	0 to 10^9 pulses	0
JOB_REQ	STAT	BOOL	66.0	Initiates the job (positive edge)	TRUE/FALSE	FALSE
JOB_DONE	STAT	BOOL	66.1	New job can be started	TRUE/FALSE	TRUE
JOB_ERR	STAT	BOOL	66.2	Faulty job	TRUE/FALSE	FALSE
JOB_ID	STAT	INT	68	Job ID	1, 2	0
JOB_STAT	STAT	WORD	70	Job error ID	0 to FFFF hex	0
JOB_VAL	STAT	DINT	72	Job parameters for the coordinates of the reference point	-5×10^8 to $+5 \times 10^8$ pulses	0

Counting, Frequency Measurement and Pulse Duration Modulation 5

5.1 Overviews

5.1.1 Operating Modes and Properties – Overview

Operating modes of the CPUs

- Counting
- Frequency counting
- Pulse-width modulation (output of a pulse train)

Overview of CPU Properties

- Number of channels
  - CPU 312C: 2 channels
  - CPU 313C: 3 channels
  - CPU 313C-2 DP, PtP: 3 channels
  - CPU 314C-2 DP, PN/DP, PtP: 4 channels

  Note

  You have only 2 channels
  (channels 2 and 3) available when using a positioning function.

- Counting frequency
  - CPU 312C: max. 10 kHz
  - CPU 313C: max. 30 kHz
  - CPU 313C-2 DP, PtP: max. 30 kHz
  - CPU 314C-2 DP, PN/DP, PtP: max. 60 kHz
- Signals the CPU counts

  24 V incremental encoder with two tracks (rotary encoders) with a 90° phase shift.[1]
  - 24 V pulse encoder with direction level
  - 24 V initiator (e.g. BERO or light barrier)
- Configuration
  - Via parameter assignment screens

[1] Four-fold evaluation of the two signals produces the internal 4-fold counting frequency.

5.1.2 Scope of Functions – Overview

Counting

- Counting modes
  - Count continuously
  - Count once
  - Count periodically
- Gate function

  For starting, stopping and interrupting the counting functions.
- Latch function

  You can use this function to save the current internal count value at the positive edge on the digital input.
- Comparator

  You can store a comparison value in the CPU. Depending on the count value and comparison value, you can activate a digital output or generate a hardware interrupt.
- Hysteresis

  You can specify a hysteresis for the digital output. This prevents the digital output from dithering at every small jitter of the encoder signal when the count value lies within the comparison value range.
- Hardware interrupts
- Period measurement

  You can measure the period of the counting signal up to a maximum counting frequency of 1 kHz.

Frequency Counting

- Gate function

  Use the gate function to start and stop the frequency measurement.
- High/low limit

  You can specify a high and low limit for frequency monitoring. You can activate a digital input and/or generate a hardware interrupt when one of the limits is reached.
- Hardware interrupts

Pulse-Width Modulation (PWM)

- Gate function

  Use the gate function to start and stop pulse-width modulation.
- Hardware interrupts

5.1.3 Components of a Counter Application – Overview

Overview

The counting functions (counting, frequency counting and pulse-width modulation) are integrated in the **CPU**. The **encoder,** or a bounce-free switch, supplies the count pulses.

Use the **PG/PC** to:

- Assign the CPU parameters using the parameter assignment screens for the technological functions of the CPU.
- Program CPU SFBs which you can integrate directly in your user program.
- Commission and test the CPU with the help of the standard STEP7 interface (monitoring functions and variable table).

5.2 Wiring

5.2.1 Wiring Rules

Connecting Cables/Shielding

- The encoder cables must be shielded.
- The cables for the digital I/O must be shielded if their length exceeds 100 m.
- The cable shielding must be terminated on both ends.
- Flexible cable, cross-section 0.25 mm to 1.5 mm^2.
- Cable sleeves are not required. Should you still decide to do so, use cable sleeves without insulating collar (DIN 46228, Shape A, short version).

Shielding termination element

You can use this shielding termination element for easy shielded-cable-to-ground connections – due to the direct contact of the shielding termination element to the profile rail.

WARNING
Harm to health and damage to assets cannot be excluded if you do not switch off voltage: If you wire the front plug on the live module you run the risk of injury as a result of electrical current! Always wire the module off-voltage state!

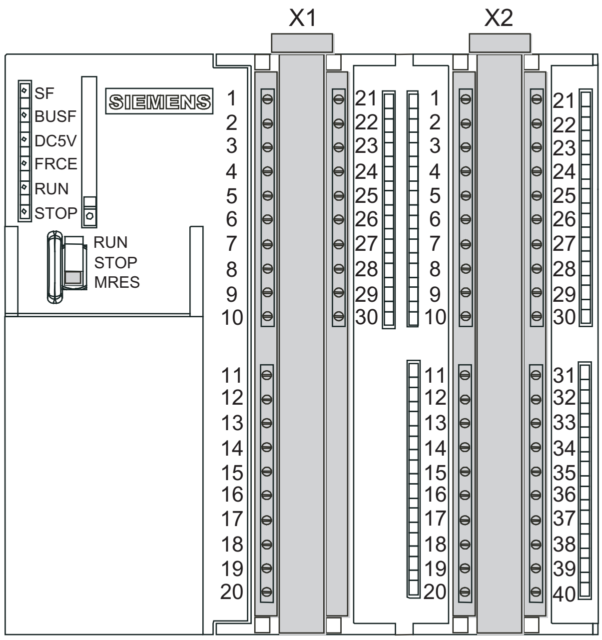
Additional Information

For additional information refer to the *CPU Data* manual and to the installation instructions for your CPU.

5.2.2 Pin Assignment

Basic Arrangement of the Connector

Based on the CPU 314C-2 DP, PN/DP, PtP, the view below shows the principal plug arrangement for CPUs with two connectors (X1 and X2):



Pin assignment

The following pin assignments only apply to connectors relevant for counting, frequency measurement and pulse width modulation.

Note

Since they require the same I/Os, you **cannot use the channels 0 and 1 anymore** if you utilize the positioning function.

WARNING

When operating with pulse width modulation, the corresponding channel input "TrackB/Direction" must either remain disconnected or be connected to logical "0".

Pin Assignment CPU 312C Connector X1

Connection	Name/ address	Counting	Frequency Measurement	Pulse Width Modulation
1	-	Not connected.		
2	DI + 0.0	Channel 0: Track A/Pulse	Channel 0: Track A/pulse	-
3	DI + 0.1	Channel 0: Track B/Direction	Channel 0: Track B/direction	0/do not use
4	DI + 0.2	Channel 0: Hardware Gate	Channel 0: Hardware gate	Channel 0: Hardware gate
5	DI + 0.3	Channel 1: Track A/pulse	Channel 1: Track A/pulse	-
6	DI + 0.4	Channel 1: Track B/direction	Channel 1: Track B/direction	0/do not use
7	DI + 0.5	Channel 1: Hardware gate	Channel 1: Hardware gate	Channel 1: Hardware gate
8	DI + 0.6	Channel 0: Latch	-	-
9	DI + 0.7	Channel 1: Latch	-	-
10	DI + 1.0	-		
11	DI + 1.1	-		
12	2 M	Chassis ground		
13	1 L+	24 V power supply for the outputs		
14	DO + 0.0	Channel 0: Output	Channel 0: Output	Channel 0: Output
15	DO + 0.1	Channel 1: Output	Channel 1: Output	Channel 1: Output
16	DO + 0.2	-		
17	DO + 0.3	-		
18	DO + 0.4	-		
19	DO + 0.5	-		
20	1 M	Ground		

Pin assignment CPU 313C (connector X2) or CPU 313C-2 DP, PtP (connector X1)

Connection	Name/ address	Counting	Frequency counting	Pulse-width modulation
1	1 L+	24 V power supply for the inputs		
2	DI + 0.0	Channel 0: Track A/pulse	Channel 0: Track A/pulse	-
3	DI + 0.1	Channel 0: Track B/direction	Channel 0: Track B/direction	0/do not use
4	DI + 0.2	Channel 0: Hardware gate	Channel 0: Hardware gate	Channel 0: Hardware gate
5	DI + 0.3	Channel 1: Track A/pulse	Channel 1: Track A/pulse	-
6	DI + 0.4	Channel 1: Track B/direction	Channel 1: Track B/direction	0/do not use
7	DI + 0.5	Channel 1: Hardware gate	Channel 1: Hardware gate	Channel 1: Hardware gate
8	DI + 0.6	Channel 2: Track A/pulse	Channel 2: Track A/pulse	-
9	DI + 0.7	Channel 2: Track B/direction	Channel 2: Track B/direction	0/do not use
10	-	Not connected		
11	-	Not connected		
12	DI + 1.0	Channel 2: Hardware gate	Channel 2: Hardware gate	Channel 2: Hardware gate
13	DI + 1.1	-	-	-
14	DI + 1.2	-	-	-
15	DI + 1.3	-	-	-
16	DI + 1.4	Channel 0: Latch	-	-
17	DI + 1.5	Channel 1: Latch	-	-
18	DI + 1.6	Channel 2: Latch	-	-
19	DI + 1.7	-	-	-
20	1 M	Ground		
21	2 L+	24 V power supply for the outputs		
22	DO + 0.0	Channel 0: Output	Channel 0: Output	Channel 0: Output
23	DO + 0.1	Channel 1: Output	Channel 1: Output	Channel 1: Output
24	DO + 0.2	Channel 2: Output	Channel 2: Output	Channel 2: Output
25	DO + 0.3	-		
26	DO + 0.4	-		
27	DO + 0.5	-		
28	DO + 0.6	-		
29	DO + 0.7	-		
30	2 M	Ground		
31	3 L+	24 V power supply for the outputs		
32	DO + 1.0	-		
33	DO + 1.1	-		
34	DO + 1.2	-		
35	DO + 1.3	-		
36	DO + 1.4	-		
37	DO + 1.5	-		
38	DO + 1.6	-		
39	DO + 1.7	-		
40	3 M	Ground		

Pin assignment CPU314C-2 DP PN/DP, PtP (connector X2)

Connection	Name/ address	Counting	Frequency measurement	Pulse-width modulation
1	1 L+	24-V power supply for the inputs		
2	DI + 0.0	Channel 0: Track A/pulse	Channel 0: Track A/pulse	-
3	DI + 0.1	Channel 0: Track B/direction	Channel 0: Track B/direction	0/do not use
4	DI + 0.2	Channel 0: Hardware gate	Channel 0: Hardware gate	Channel 0: Hardware gate
5	DI + 0.3	Channel 1: Track A/pulse	Channel 1: Track A/pulse	-
6	DI + 0.4	Channel 1: Track B/direction	Channel 1: Track B/direction	0/do not use
7	DI + 0.5	Channel 1: Hardware gate	Channel 1: Hardware gate	Channel 1: Hardware gate
8	DI + 0.6	Channel 2: Track A/pulse	Channel 2: Track A/pulse	-
9	DI + 0.7	Channel 2: Track B/direction	Channel 2: Track B/direction	0/do not use
10	-	Not connected		
11	-	Not connected		
12	DI + 1.0	Channel 2: Hardware gate	Channel 2: Hardware gate	Channel 2: Hardware gate
13	DI + 1.1	Channel 3: Track A/pulse	Channel 3: Track A/pulse	-
14	DI + 1.2	Channel 3: Track B/direction	Channel 3: Track B/direction	0/do not use
15	DI + 1.3	Channel 3: Hardware gate	Channel 3: Hardware gate	Channel 3: Hardware gate
16	DI + 1.4	Channel 0: Latch	-	-
17	DI + 1.5	Channel 1: Latch	-	-
18	DI + 1.6	Channel 2: Latch	-	-
19	DI + 1.7	Channel 3: Latch	-	-
20	1 M	Ground		
21	2 L+	24 V power supply for the outputs		
22	DO + 0.0	Channel 0: Output	Channel 0: Output	Channel 0: Output
23	DO + 0.1	Channel 1: Output	Channel 1: Output	Channel 1: Output
24	DO + 0.2	Channel 2: Output	Channel 2: Output	Channel 2: Output
25	DO + 0.3	Channel 3: Output	Channel 3: Output	Channel 3: Output
26	DO + 0.4	-		
27	DO + 0.5	-		
28	DO + 0.6	-		
29	DO + 0.7	-		
30	2 M	Ground		
31	3 L+	24 V power supply for the outputs		
32	DO + 1.0	-		
33	DO + 1.1	-		
34	DO + 1.2	-		
35	DO + 1.3	-		
36	DO + 1.4	-		
37	DO + 1.5	-		
38	DO + 1.6	-		
39	DO + 1.7	-		
40	3 M	Ground		

5.2.3 Connecting Components

Procedure

1. Switch off the power supply to all components.
2. Connect the power supply for the inputs and outputs:

   CPU 312C:

   – 24 V at X1, pin 13
   – Ground at X1, pins 12 and 20

   CPU 313C-2 DP, PtP:

   – 24 V at X1, pins 1 and 21
   – Ground at X1, pins 20 and 30

   CPU 313C, CPU 314C-2 DP, PN/DP, PtP:

   – 24 V at X2, pins 1 and 21
   – Ground at X2, pins 20 and 30
3. Connect the encoder and switches to the 24 V power supply.
4. Connect the encoder signals and the required switches. You can connect bounce-free switches (24 V P-switching) or sensors/BERO (2- or 3-wire proximity switches) to the digital inputs "Hardware gate" and "Latch".
5. Strip the insulation material on the shielded cables and bind the cable shield to the shield connection element. Use the shield terminal elements for this.

5.3 Parameter configuration

5.3.1 Configuration with the Parameter Assignment Screen

Basics

You adapt the counting function to your specific application by assigning parameters:

- You assign the parameters via the parameter assignment screens.
- They are stored in the system memory of the CPU.
- You can change some of the parameters while the CPU is in RUN mode, using the SFB job interface (see Section Controlling the Counter via the User Program (Page 193), Controlling the Frequency Counter via the User Program (Page 212), or Controlling Pulse Width Modulation via the User Program (Page 222)).

Parameter assignment screens

You can assign the module parameters in the **parameter assignment screens**:

- Basic parameters
- Continuous, once-only, and periodic counting
- Frequency counting
- Pulse-width modulation

Those parameter assignment screens are largely self-explanatory. You can find the description of the parameters in the following sections and in the integrated help for the parameter assignment screens.

Note

If you use channel 0 or channel 1, you can no longer use "Positioning" technology.

Requirements

Prerequisite for calling the parameter assignment screen is that you have created a project in which you can save your parameter assignment.

Procedure

1. Start the SIMATIC Manager and open HW Config in your project.
2. Double-click on the "Count" submodule of your CPU. The "Properties" dialog box opens.
3. Assign the parameters for the "Count" submodule and close the parameter assignment screen with "OK".
4. Save your project in HW Config with "Station > Save and Compile".
5. Download the parameter assignment data to the CPU when it is in STOP mode with "PLC > Download to Module...". The data are now stored in the CPU's system data memory.
6. Start the CPU.

Online Help

The online help in the parameter assignment screens offers you support when you assign parameters. You have the following options of calling the online help:

- Press the "F1" key in the respective views
- Click on the Help button in the various parameter assignment screens.

5.3.2 Basic parameters

Interrupt Selection Parameter

Parameters	Description	Value range	Default
Interrupt selection	Here you can select which interrupts should trigger the technological operation.	• None • Diagnostics • Process • Diagnostics and Process	None

5.3.3 Continuous, Single and Periodic Counting Parameters

Description of Parameters

Parameters	Description	Value range	Default
Main count direction	• None: No restriction of the counting range • Up: Restricts the counting range in the up direction. Counter starts at 0 or the load value and counts in the positive direction up to the assigned end value - 1. It then jumps back to the load value at the next positive encoder pulse. • Down: Restricts the counting range in the down direction. Counter starts at the assigned start value or the load value and counts in the negative direction to 1. It then jumps to start value at the next negative encoder pulse.	• None • Up (not with continuous counting) • Down (not with continuous counting)	None
End value/ Start value	• End value with main count direction up • Start value with main count direction down	2 to 2147483647 (2^{31} - 1)	2147483647 (2^{31} - 1)
Gate function	• Cancel the counting operation: The count restarts at the load value when the gate is closed and restarted. • Stop the counting operation: The count is stopped when the gate closes and resumed at the last actual count when the gate opens again.	• Cancel count • Stop count	Cancel count
Comparison value	The count value is compared with the comparison value. See also the "Characteristics of the Output" parameter:		0
	• No main count direction	-2^{31} to $+2^{31}-1$	
	• Main count direction up	-2^{31} to end value -1	
	• Main count direction down	1 to $+2^{31}$ - 1	
Hysteresis	The hysteresis is used to avoid frequent output switching actions if the count value lies within the range of the comparison value. 0 and 1 mean: Hysteresis switched off.	0 to 255	0
Max. frequency: Counting signals/ HW gate	You can set the maximum frequency of the track A/pulse, track B/direction and hardware gates signals in fixed steps. The maximum value is CPU-specific:		
	• CPU 312C	10, 5, 2, 1 kHz	10 kHz
	• CPU 313C, 313C-2 DP, PtP	30, 10, 5, 2, 1 kHz	30 kHz
	• CPU 314C-2 DP, PN/DP, PtP	60, 30, 10, 5, 2, 1 kHz	60 kHz

Parameters	Description	Value range	Default
Max. frequency: Latch	You can set the maximum frequency of the latch signal in fixed steps. The maximum value is CPU-specific:		
	• CPU 312C	10, 5, 2, 1 kHz	10 kHz
	• CPU 313C, 313C-2 DP, PtP	30, 10, 5, 2, 1 kHz	10 kHz
	• CPU 314C-2 DP, PN/DP, PtP	60, 30, 10, 5, 2, 1 kHz	10 kHz
Signal evaluation	• The counting and direction signals are connected to the input • A rotary encoder is connected to the input (single, double, or quadruple evaluation)	• Pulse/direction • Rotary encoder, single • Rotary encoder, double • Rotary encoder, quadruple	Pulse/ direction
HW gate	• Yes: Gate control via SW and HW gates. • No: Gate control via SW gate only.	• Yes • No	No
Inverted count direction	• Yes: Inverted "Direction" input signal. • No: "Direction" input signal is not inverted.	• Yes • No	No
Characteristics of the output	The output and the "Comparator" (STS_CMP) status bit are set, dependent on this parameter.	• No comparison • Count value ≥ comparison value • Count value ≤ comparison value • Pulse at comparison value	No comparison
Pulse duration	With the setting "Characteristics of the output: Pulse at comparison value", you can specify the pulse duration for the output signal. Only even values are possible.	0 to 510 ms	0
Assignment of input data	You can select whether the count value or the period can be read at a maximum counting frequency of 1 kHz in the input data (I data) of the "Count" submodule. If the maximum counting frequency is greater than 1 kHz, only "Count value" is possible.	Count value Period	Count value
Time base	You can specify whether the period is to be measured in units of 125 ns or 1 µs at a maximum counting frequency of 1 kHz. If the maximum counting frequency is greater than 1 kHz, the period is not measured.	125 ns 1 µs	125 ns
Hardware interrupt: HW gate opening	A hardware interrupt is generated when the hardware gate opens while the software gate is open.	• Yes • No	No
Hardware interrupt: HW gate closing	A hardware interrupt is generated when the hardware gate closes while the software gate is open.	• Yes • No	No

Parameters	Description	Value range	Default
Hardware interrupt: On reaching comparator	A hardware interrupt is generated on reaching the comparator.	• Yes • No	No
Hardware interrupt: Overflow	A hardware interrupt is generated on overflow (exceeding the high counting limit).	• Yes • No	No
Hardware interrupt: Underflow	A hardware interrupt is generated upon underflow (low counting limit violated).	• Yes • No	No
Hardware interrupt: Counting edge	You can select whether a hardware interrupt is generated on each counting edge at a maximum counting frequency of 1 kHz. This hardware interrupt cannot be selected if the maximum counting frequency is greater than 1 kHz. Generating a hardware interrupt on each counting edge results in high CPU utilization at higher counting frequencies. For this reason, you should only enable this hardware interrupt when the counting edges are at least 10 ms apart.	• Yes • No	No

5.3.4 Frequency measurement

Description of Parameters

Parameters	Description	Value range	Default
Integration time	Time window in which the incoming pulses are measured.	10 to 10,000 ms	100
Low limit	The measured value is compared with the low limit. The status bit "Underflow" (STS_UFLW) is set when the low limit is exceeded. The low limit must be less than the high limit.	CPU 312C: 0 to 9 999 999 mHz CPU 313C, CPU 313C-2 DP, PtP: 0 to 29 999 999 mHz CPU 314C-2 DP, PN/DP, PtP: 0 to 59 999 999 mHz	0
High limit	The measured value is compared with the high limit. The status bit "Overflow" (STS_OFLW) is set when the high limit is exceeded. The high limit must be greater than the low limit.	CPU 312C: 1 to 10 000 000 mHz CPU 313C, CPU 313C-2 DP, PtP: 1 to 30 000 000 mHz CPU 314C-2 DP, PN/DP, PtP: 1 to 60 000 000 mHz	CPU 312C: 10 000 000 mHz CPU 313C, CPU 313C-2 DP, PtP: 30 000 000 mHz CPU 314C-2 DP, PN/DP, PtP: 60 000 000 mHz

Parameters	Description	Value range	Default
Maximum counting frequency	You can set the maximum frequency of the track A/pulse, track B/direction and hardware gates signals in fixed steps. The maximum value is CPU-specific:		
	CPU 312C	10, 5, 2, 1 kHz	10 kHz
	CPU 313C, 313C-2 DP, PtP	30, 10, 5, 2, 1 kHz	30 kHz
	CPU 314C-2 DP, PN/DP, PtP	60, 30, 10, 5, 2, 1 kHz	60 kHz
Output measured value	if the period of the measured frequency exceeds the configured integration time: • With "direct" frequency mode, the value "0" is output at the end of the integration time. • With "averaged" frequency, the last value is distributed across the subsequent measuring intervals without an edge (f ≥ 1 mHz). This stretches the integration time. Here, the last measured value is divided by the number of measurement intervals without edge.	• Direct • Averaged	Direct
Signal evaluation	• The count and direction signals are connected to the input • A rotary transducer with single evaluation is connected	• Pulse/Direction • Rotary transducer, single	Pulse/direction
Inverted count direction	• Yes: Inverted "Direction" input signal. • No: "Direction" input signal is not inverted.	• Yes • No	No
HW gate	• Yes: Gate control via the SW and HW gates. The max. frequency of the HW gate signal is equivalent to the max. counting frequency set. • No: Gate control via SW gate only.	• Yes • No	No
Characteristics of the output	The measured value is compared with the high and low limits. The output is switched, depending on this parameter.	• No comparison • Out of limits • Below the low limit • Above the high limit	No comparison
Hardware interrupt: HW gate opening	A hardware interrupt is generated when the hardware gate opens while the software gate is open.	• Yes • No	No
Hardware interrupt: Closing the HW gate	A hardware interrupt is generated when the hardware gate closes while the software gate is open.	• Yes • No	No
Hardware interrupt: End of measurement	A hardware interrupt is generated at the end of measurement.	• Yes • No	No

Parameters	Description	Value range	Default
Hardware interrupt: Exceeding the low limit	A hardware interrupt is generated when the low limit is exceeded.	• Yes • No	No
Hardware interrupt: Exceeding the high limit	A hardware interrupt is generated when the high limit is exceeded.	• Yes • No	No

5.3.5 Pulse-width modulation

Description of parameters

Parameter	Description	Value range	Default
Output format	Output format	• Per mil • S7 analog value	Per mil
Time base	Time base for • On-delay • Period • Minimum pulse duration	• 0.1 ms • 1.0 ms	0.1 ms
On-delay	Time interval between the start of the output sequence and pulse output.	0 – 65535	0
Period	Defines the length of the output sequence in terms of pulse duration and interpulse period.	• Time base 0.1 ms: 4 to 65535 • Time base 1 ms: 1 to 65535	20 000
Minimum pulse duration	Output pulses/interpulse periods shorter than the minimum pulse duration are suppressed. With a time base of 1 ms and 0 value the minimum pulse duration is set internally to 0.2 ms.	• Time base 0.1 ms: 2 to period/2 • Time base 1 ms: 0 to period/2	2
HW gate	• Yes: Gate control via SW and HW gates. • No: Gate control via SW gate only.	• Yes • No	No
Filter frequency of HW gate	You can set the filter frequency of the hardware gate signal in fixed steps. The maximum value is CPU-specific:		
	CPU 312C	10, 5, 2, 1 kHz	10 kHz
	CPU 313C, 313C-2 DP, PtP	30, 10, 5, 2, 1 kHz	30 kHz
	CPU 314C-2 DP, PN/DP, PtP	60, 30, 10, 5, 2, 1 kHz	60 kHz
Hardware interrupt: HW gate opening	A hardware interrupt is generated when the hardware gate opens while the software gate is open.	• Yes • No	No

5.4 Implementing Functions in the User Program

Procedure

The functions are controlled in your user program. To do this, call the following system function blocks:

Function	SFB
Counting	SFB COUNT (SFB 47)
Frequency counting	SFB FREQUENC (SFB 48)
Pulse-width modulation	SFB PULSE (SFB 49)

The SFBs are found in the "Standard library" under "System Function Blocks".

The following sections help you to design a user program for your application.

You can read the actual count values in "Counting" mode and the actual frequency values in "Frequency counting" mode directly via the input address (I-address) set in the "Count" submodule.

Calling the SFB

Call the SFB with a corresponding instance DB.

Example: CALL SFB 47, DB30

Instance DB

The SFB parameters are stored in the instance DB. These parameter are described in Sections Controlling the Counter via the User Program (Page 193), Controlling the Frequency Counter via the User Program (Page 212), and Procedure for Pulse Width Modulation (Page 221).

You can access the parameters via:

- DB number and absolute address in the DB
- DB number and symbolic address in the DB

The main parameters for the function are also interconnected to the block. You can assign the input parameters directly at the SFB or you can evaluate the output parameters.

Note

For each channel you must always call the SFB with the same instance DB, because the instance DB contains the required states for the internal processes of the SFB.

Write access to the outputs of the instance DB is not allowed.

Program structure

The SFB must be called (e.g. OB1) cyclically.

Note

You must not call an SFB you have configured in your program in another program section with a different priority class, because the SFB must not interrupt itself.
Example: It is not allowed to call the same SFB both in OB1 and in the interrupt OB.

I/O Access

In counting and frequency counting modes, you can read the current count values/periods or frequency values (depending on the set mode) by accessing the I/O directly via the input address (I-address) of the "Count" submodule.

You have specified the input address of the submodule in "HW Config".

The submodules has an address area of 16 bytes.

I-address	Channel	Type	CPU	Comment	Value range
n + 0	0	DINT	312C 313C 313C-2 DP, PtP 314C-2 DP, PN/DP, PtP	Count value/ period	-2^{31} to $2^{31} - 1$
				Frequency value	0 to $2^{31} - 1$
n + 4	1	DINT	312C 313C 313C-2 DP, PtP 314C-2 DP, PN/DP, PtP	Count value/ period	-2^{31} to $2^{31} - 1$
				Frequency value	0 to $2^{31} - 1$
n + 8	2	DINT	313C 313C-2 DP, PtP 314C-2 DP, PN/DP, PtP	Count value/ period	-2^{31} to $2^{31} - 1$
				Frequency value	0 to $2^{31} - 1$
			312 C	Not assigned	0
n + 12	3	DINT	314C-2 DP, PN/DP, PtP	Count value/ period	-2^{31} to $2^{31} - 1$
				Frequency value	0 to $2^{31} - 1$
			312C 313C 313C-2 DP, PtP	Not assigned	0
n = input address of the "Count" submodule					

In the counting mode, you must specify whether you are reading the count value or the period. In the pulse-width modulation mode, the entire I/O area of the submodule is filled with "0" (as of the I-address).

Write access to the "Count" submodule (starting at the Q address) is not possible.

5.5 Counting – Description of Function

5.5.1 Basic Terms for Counting

Counting modes

The counting modes support your counting applications. Here, the counting signal is captured and evaluated by the CPU. You can count up and down.

You have the choice of the following operating modes:

- Count continuously, for example for position feedback with 24 V incremental encoders
- Count once, for example, for counting parts up to a maximum limit
- Count periodically, for example, for intermittent counting applications

You select the operating mode via the parameter assignment screens.

Maximum Counting Frequency

CPU 312C	CPU 313C, CPU 313C-2 DP, PtP	CPU 314C-2 DP, PN/DP, PtP
10 kHz	30 kHz	60 kHz

Period measurement

At a maximum counting frequency of 1 kHz, the time between two consecutive counting edges is always measured. You can read the measured period directly via the input data (I-data) of the "Count" submodule or by direct I/O access.

You must specify which input data are assigned. You can read either the count value or the period.

If the maximum counting frequency is greater than 1 kHz, the period is not measured and the value is 0.

After each parameter assignment, a measured period is available after the second counting edge; prior to that, the value is 0.

The period is available as an unsigned 32-bit value in the assigned time base.

Periods of up to a maximum of 4,294,967 s (49 d, 17 h, 2 min, 47 s) can be measured with a time base of 1 ms, and periods of up to 536 ms (8 min, 56 s) with a time base of 125 ns.

If the counting edges are further apart in time, the measured period is incorrect because an overflow has not been taken into account.

Count Value/Load Value

You can assign a default value to the counter.

This allows you to:

- Set the count value directly. The count value is then applied immediately.
- Set the load value. The load value is then applied as the new count value when specific events occur, depending on the set operating mode.

Main count direction

You can limit the counting range by specifying the main count direction. This specifies which counting limit applies as the start or end values in the operating modes "Count once" and "Count periodically".

You select the main count direction in the parameter assignment screens.

- No main count direction:

  The entire counting range is available to you with this setting:

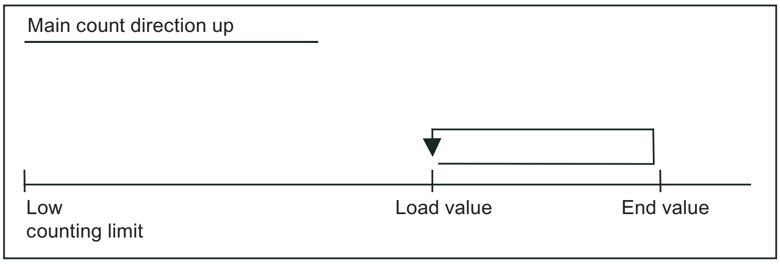
Low counting limit	–2 147 483 648	(-2^{31})
High counting limit	+2 147 483 647	(2^{31} - 1)

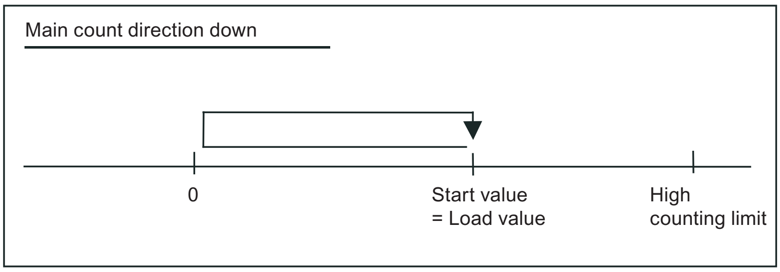
- Main count direction up:

  With main count direction up, you limit the counting range in the up direction. The counter starts at 0 or the load value and counts in the positive direction until it reaches the assigned end value -1. It then jumps back to the load value at the next positive encoder pulse.
- Main count direction down:

  With main count direction down, you limit the counting range in the down direction. The counter starts at the assigned start value or load value and counts in the negative direction until it reaches the value 1. It then jumps back to the start value at the next negative encoder pulse.

You specify the count direction independent of the "Main count direction" parameter. To do this, you either apply an appropriate direction signal or you assign the count direction during parameter assignment.





Starting/Stopping the Counter

Use the gate function to start, stop and interrupt the counting functions. For information on setting up the gate function, refer to Section Gate Function of the Counter (Page 201).

Overflow/Zero Crossing/Underflow

The overflow bit (STS_OFLW) is set when the high counting limit is exceeded.

The underflow bit (STS_UFLW) is set when the low counting limit is fallen below.

Zero crossing is indicated by setting the zero crossing bit (STS_ZP). This bit is only set when counting without a main count direction. The zero crossing is also indicated when the counter is set to 0 or is counting starting from load value = 0.

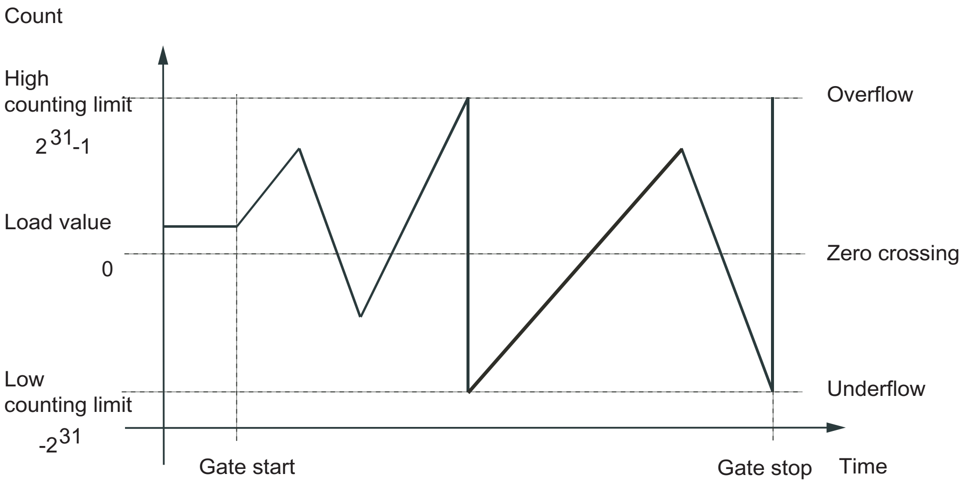
5.5.2 Count continuously

Description

In this operating mode, the CPU starts counting at 0 or load value.

- When the up-counter reaches the high limit, it jumps to the low limit at the next positive count pulse and resumes the count from there.
- When the down-counter reaches the low limit, it jumps to the high limit at the next negative count pulse and resumes the count from there.
- The count limits are set to maximum range.

	Valid value range	Default value
High count limit	+2147483647 (2^{31} - 1)	-
Low count limit	-2147483648 (-2^{31})	-
Count value	-2147483648 (-2^{31}) to +2147483647 (2^{31} - 1)	0
Load value	-2147483647 (-2^{31} + 1) to +2147483646 (2^{31} - 2)	0



5.5.3 Single-Cycle Count

Description

In this operating mode, the CPU counts once, depending on the assigned main count direction.

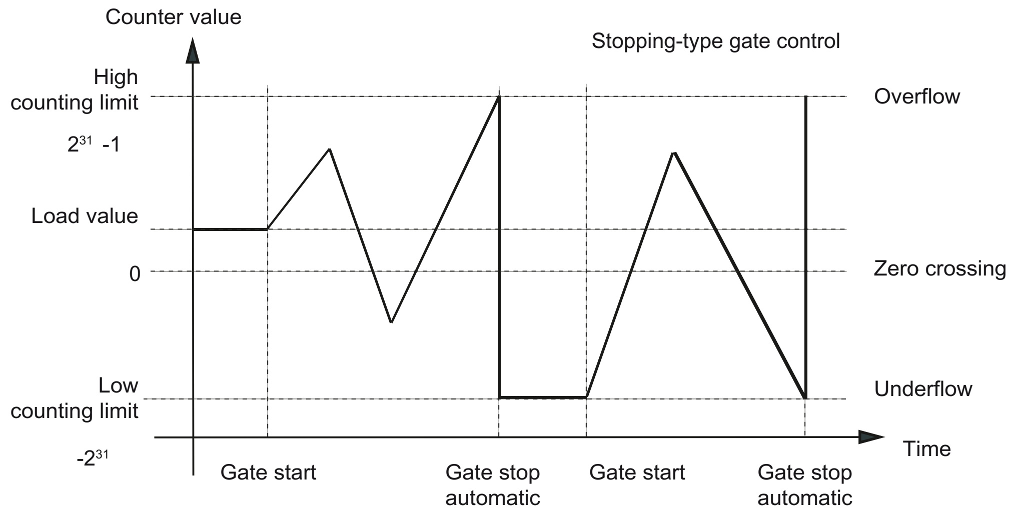
- **No main count direction:**
  - The CPU counts once starting from the load value.
  - The CPU counts up or down.
  - The counting limits are set to the maximum counting range.
  - On overflow or underflow at the counting limits, the counter jumps to the respective opposite counting limit, and the gate is closed automatically.

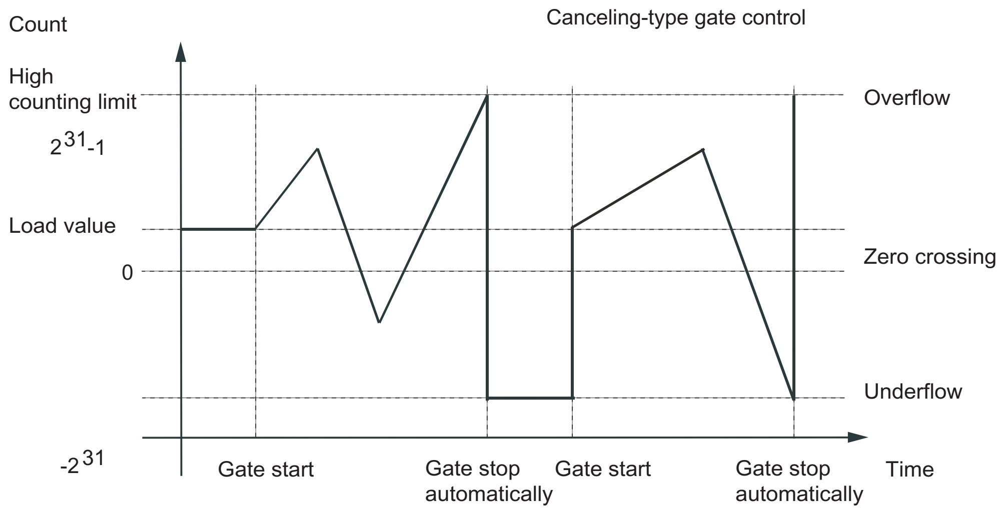
    To restart the count, you must generate a positive edge at the gate control.

    With stopping gate control, the count resumes at the current count.

    With canceling gate control, the count restarts from the load value.

	Valid value range	Default value
High count limit	+2147483647 (2^{31} - 1)	-
Low count limit	-2147483648 (-2^{31})	-
Count value	-2147483648 (-2^{31}) to +2147483647 (2^{31} - 1)	0
Load value	-2147483647 (-2^{31} + 1) to +2147483646 (2^{31} - 2)	0

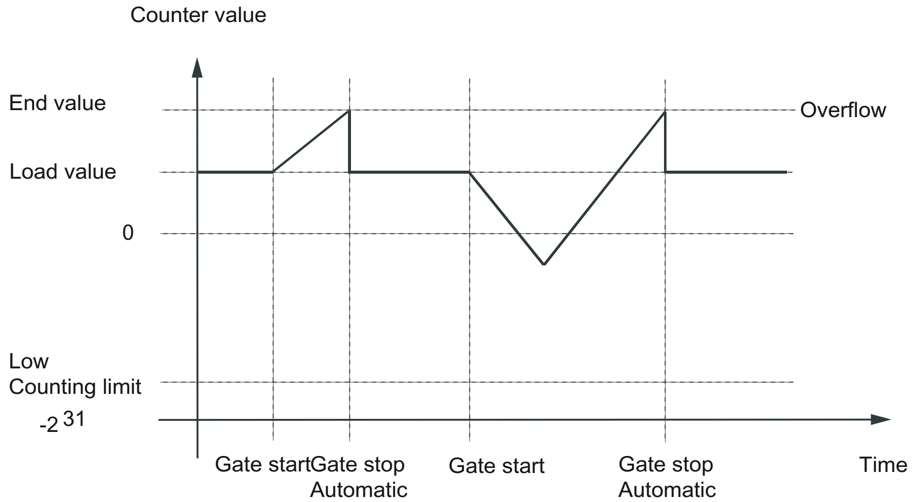




- **Main count direction up:**
  - The CPU starts counting at the load value.
  - The CPU counts up or down.
  - When the counter reaches the end value -1 in the positive direction, it jumps to the load value at the next positive counting pulse, and the gate is closed automatically.

    To restart the count, you must generate a positive edge at the gate control. The counter starts at the load value.
  - You can also count beyond the low counting limit. In this case, however, the count value and comparison results do not match. You should therefore avoid operation in this range.

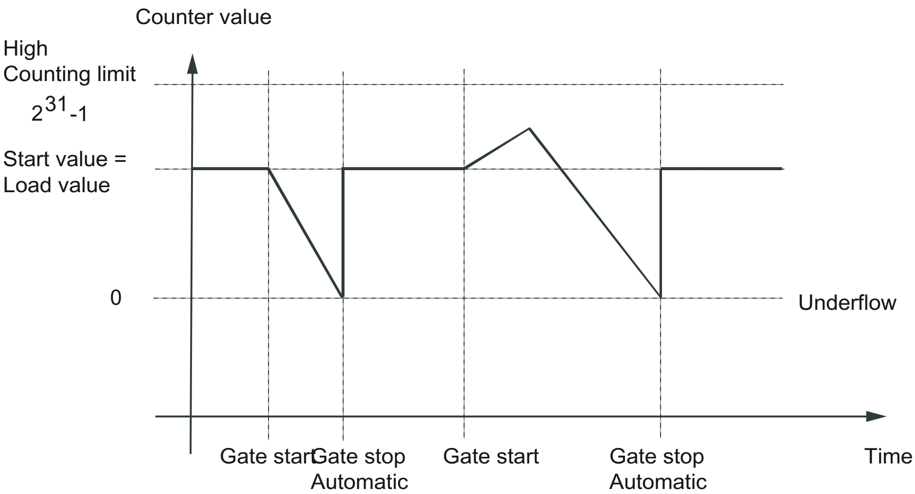
	Valid value range	Default value
End value	up to +2147483646 (2^{31} - 1)	Configurable
Low count limit	-2147483648 (-2^{31})	-
Count value	-2147483648 (-2^{31}) to end value - 1	0
Load value	-2147483648 (-2^{31}) to end value - 2	0



- **Main count direction down:**
  - The CPU starts counting at the load value.
  - The CPU counts up or down.
  - When the counter reaches the count value 1 in the negative direction, it jumps to the load value (start value) at the next negative count pulse, and the gate is closed automatically.

    To restart the count, you must generate a positive edge at the gate control (see Section Gate Function of the Counter (Page 201)). The counter starts at the load value.
  - You can also count beyond the high counting limit. In this case, however, the count value and comparison results do not match. You should therefore avoid operation in this range.

	Valid value range	Default value
Start value	up to +2147483647 (2^{31} - 1)	assignable
High count limit	+2147483647 (2^{31} - 1)	-
Count value	up to +2147483647 (2^{31} - 1)	Start value
Load value	up to +2147483647 (2^{31} - 1)	Start value



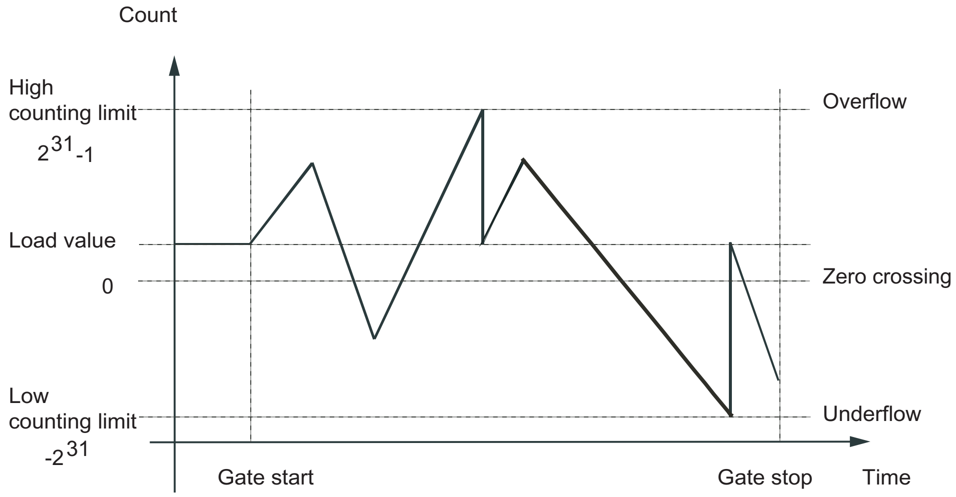
5.5.4 Periodic Count

Description

In this operating mode the CPU performs periodic counts, depending on the declared default direction of count.

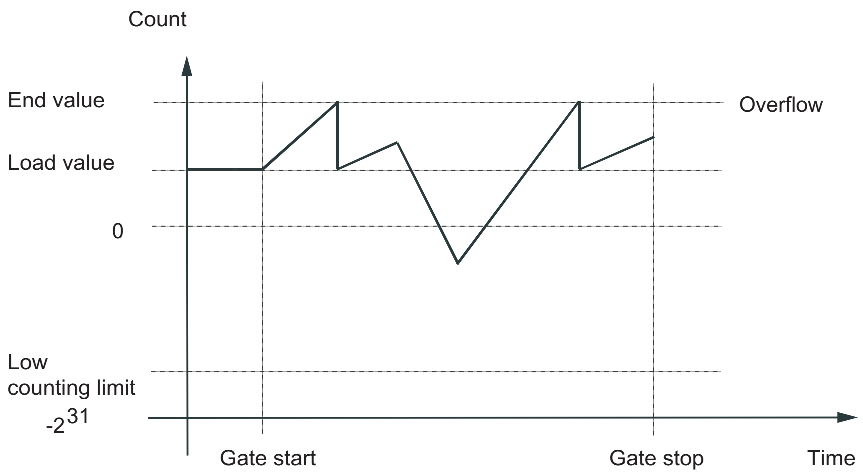
- **No default direction of count:**
  - The CPU starts counting at load value.
  - The CPU counts up or down.
  - On overflow or underflow at the respective count limit, the counter jumps to load value and resumes counting from there.
  - The count limits are fixed at the maximum count range.

	Valid value range	Default value
High count limit	+2147483647 (2^{31} - 1)	-
Low count limit	-2147483648 (-2^{31})	-
Count value	-2147483648 (-2^{31}) to +2147483647 (2^{31} - 1)	0
Load value	-2147483647 (-2^{31} + 1) to +2147483646 (2^{31} - 2)	0



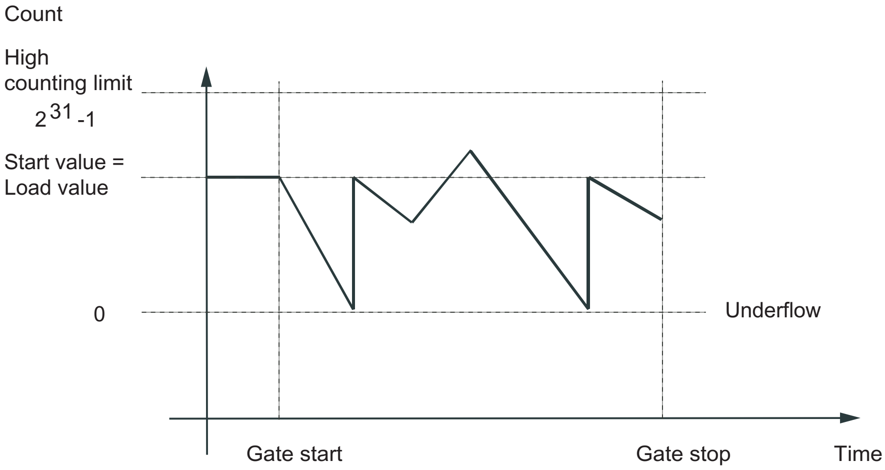
- **Up-count as default:**
  – The CPU starts counting at load value.
  – The CPU counts up or down.
  – When the counter reaches the end value -1, operating in positive direction, it jumps to load value at the next positive count pulse and resumes counting from there.
  – You can also exceed the lower count limit. In this case, however, the count value and compare results mismatch. You should therefore avoid operation in this range.

	Valid value range	Default value
End value	up to +2147483647 (2^{31} - 1)	assignable
Low count limit	-2147483648 (-2^{31})	-
Count value	-2147483648 (-2^{31}) to end value - 1	0
Load value	-2147483648 (-2^{31}) to end value - 2	0



- **Down-count as default:**
  - The CPU starts counting at load value.
  - The CPU counts up or down.
  - When the counter reaches the value 1, operating in negative direction, it jumps to load value (start value) at the next negative count pulse and resumes counting from there.
  - You can also exceed the high count limit. In this case, however, the count value and compare results mismatch. You should therefore avoid operation in this range.

	Valid value range	Default value
Start value	up to +2147483647 (2^{31} - 1)	assignable
High count limit	+2147483647 (2^{31} - 1)	-
Count value	up to +2147483647 (2^{31} - 1)	Start value
Load value	up to +2147483647 (2^{31} - 1)	Start value



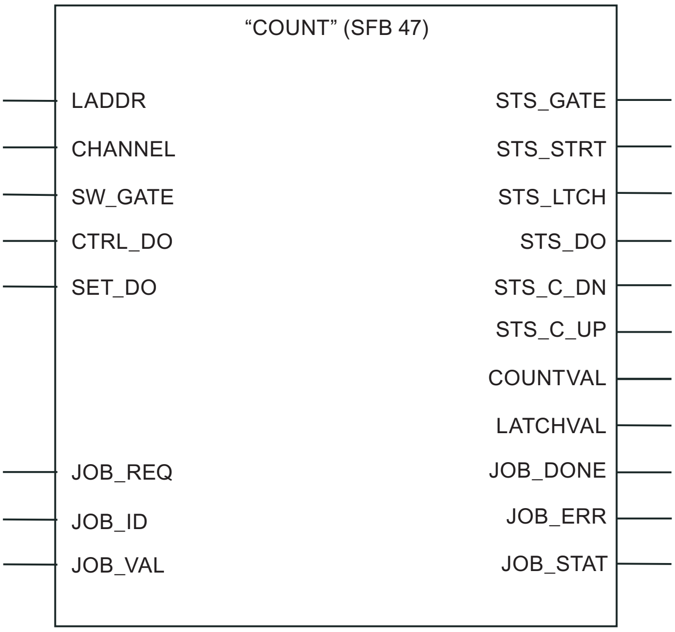
5.5.5 Controlling the Counter via the User Program

Description

To control the counter from the user program, use SFB COUNT (SFB 47).

The following functionalities are available to you:

- Start/stop the counter via software gate SW_GATE
- Enabling/controlling the output DO
- Reading out status bits
- Reading the current count value and the latch value
- Jobs for reading/writing the internal count registers
- Readout of current period (not interconnected to the block; available only in the instance DB)



Input parameters

Parameter	Data type	Address (instance DB)	Description	Value range	Default
LADDR	WORD	0	Submodule I/O address you specified in "HW Config". If the I/O addresses are not equal you must specify the lower one of both.	CPU-specific	300 hex
CHANNEL	INT	2	Channel number: CPU 312C CPU 313C, CPU 313C-2 DP, PtP CPU 314C-2 DP, PN/DP, PtP	 0 - 1 0 - 2 0 - 3	0
SW_GATE	BOOL	4.0	Software gate For counter start/stop	TRUE/FALSE	FALSE
CTRL_DO	BOOL	4.1	Enable output	TRUE/FALSE	FALSE
SET_DO	BOOL	4.2	Output control	TRUE/FALSE	FALSE

Note

If you have set the "Characteristics of the output" parameter to "No comparison" via the parameter assignment screen, the following applies:

- The output is switched as a standard output.
- The SFB input parameters CTRL_DO and SET_DO are ineffective.
- The status bits STS_DO and STS_CMP (status comparator in the IDB) remain reset.

Input parameters not interconnected to the block (static local data)

Parameters	Data type	Address (instance DB)	Description	Value range	Default
RES_STS	BOOL	32.2	Reset status bits Resets the status bits STS_CMP, STS_OFLW, STS_UFLW and STS_ZP. The SFB must be called twice to reset the status bits.	TRUE/FALSE	FALSE

Output parameters

Parameters	Data type	Address (instance DB)	Description	Value range	Default
STS_GATE	BOOL	12.0	Status of the internal gate	TRUE/FALSE	FALSE
STS_STRT	BOOL	12.1	Status of the hardware gate (Start input)	TRUE/FALSE	FALSE
STS_LTCH	BOOL	12.2	Status of the latch input	TRUE/FALSE	FALSE
STS_DO	BOOL	12.3	Output status	TRUE/FALSE	FALSE
STS_C_DN	BOOL	12.4	Status of the down-count. Always indicates the last count direction. After the first SFB call, STS_C_DN has the value FALSE.	TRUE/FALSE	FALSE
STS_C_UP	BOOL	12.5	Status of the up-count. Always indicates the last count direction. After the first SFB call, STS_C_UP has the value TRUE.	TRUE/FALSE	FALSE
COUNTVAL	DINT	14	Current count value	-2^{31} to 2^{31} - 1	0
LATCHVAL	DINT	18	Current latch value	-2^{31} to 2^{31} - 1	0

Output parameters not interconnected to the block (static local data)

Parameters	Data type	Address (instance DB)	Description	Value range	Default
STS_CMP	BOOL	26.3	Comparator status* Status bit STS_CMP indicates whether the comparison condition for the comparator is or was met. STS_CMP also indicates that the output was set (STS_DO = TRUE)	TRUE/FALSE	FALSE
STS_OFLW	BOOL	26.5	Overflow status*	TRUE/FALSE	FALSE
STS_UFLW	BOOL	26.6	Underflow status*	TRUE/FALSE	FALSE
STS_ZP	BOOL	26.7	Zero mark status* This bit is only set when counting without a main count direction. Indicates a zero crossing. This is also set when the counter is set to 0 or if it starts counting at load value = 0.	TRUE/FALSE	FALSE
* Reset with RES_STS					

5.5.6 Reading and Writing to the Request Interface for the Counter

Description

To read/write count registers, you can use the job interface.

Requirements

The last job must be finished (JOB_DONE = TRUE).

Procedure

1. Assign the following input parameters:

Parameters	Data type	Address (instance DB)	Description	Value range	Default
JOB_REQ	BOOL	4.3	Job request (positive edge)	TRUE/FALSE	FALSE
JOB_ID	WORD	6	Job number: • Job without function • Write count value • Write load value • Write comparison value • Write hysteresis • Write pulse duration • Read load value • Read comparison value • Read hysteresis • Read pulse duration	 00 hex 01 hex 02 hex 04 hex 08 hex 10 hex 82 hex 84 hex 88 hex 90 hex	0
JOB_VAL	DINT	8	Value for write jobs	-2^{31} to $+2^{31} - 1$	0

2. Call the SFB.

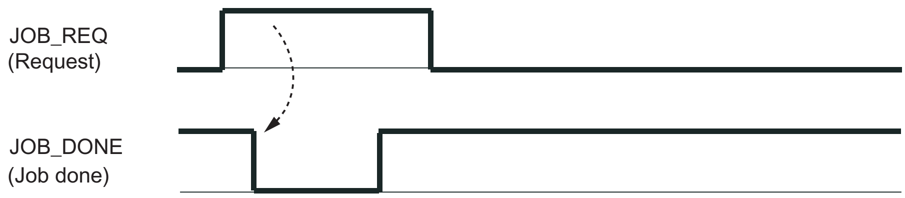
Result

The **output parameters** of the SFB provide the following information:

Parameters	Data type	Address (instance DB)	Description	Value range	Default
JOB_DONE	BOOL	22.0	New job can be started	TRUE/FALSE	TRUE
JOB_ERR	BOOL	22.1	Faulty job	TRUE/FALSE	FALSE
JOB_STAT	WORD	24	Job error number	0 to FFFF hex	0

- The job is immediately processed after the SFB is called. **JOB_DONE** is set to FALSE for the duration of one SFB cycle.
- **JOB_ERR** = TRUE if an error occurred. The precise error cause is displayed in **JOB_STAT**.
- A new job can be started with **JOB_DONE** = TRUE.
- Only for read jobs: Read the current value from the instance DB, parameter JOB_OVAL.

Parameters	Data type	Address (instance DB)	Description	Value range	Default
JOB_OVAL	DINT	28	Output value for read jobs	-2^{31} to $2^{31} - 1$	0



Permissible Value Range for JOB_VAL

Count continuously

Job	Valid value range
Write counter directly	-2147483647 ($-2^{31} + 1$) to +2147483646 ($2^{31} - 2$)
Write load value	-2147483647 ($-2^{31} + 1$) to +2147483646 ($2^{31} - 2$)
Write comparison value	-2147483648 (-2^{31}) to +2147483647 ($2^{31} - 1$)
Write hysteresis	0 to 255
Write pulse duration. Only even values allowed. Odd values are automatically rounded.	0 to 510 ms

Count once/Count periodically, no main count direction

Job	Valid value range
Write counter directly	-2147483647 ($-2^{31} + 1$) to +2147483646 ($2^{31} - 2$)
Write load value	-2147483647 ($-2^{31} + 1$) to +2147483646 ($2^{31} - 2$)
Write the comparison value	-2147483648 (-2^{31}) to +2147483647 ($2^{31} - 1$)
Write the hysteresis	0 to 255
Write pulse duration. Only even values allowed. Odd values are automatically rounded.	0 to 510 ms

Count once/Count periodically, main count direction up

Job	Valid value range
End value	up to +2147483646 ($2^{31} - 1$)
Write counter directly	-2147483648 (-2^{31}) to end value - 2
Write load value	-2147483648 (-2^{31}) to end value - 2
Write the comparison value	-2147483648 (-2^{31}) to end value - 1
Write the hysteresis	0 to 255
Write pulse duration. Only even values allowed. Odd values are automatically rounded.	0 to 510 ms

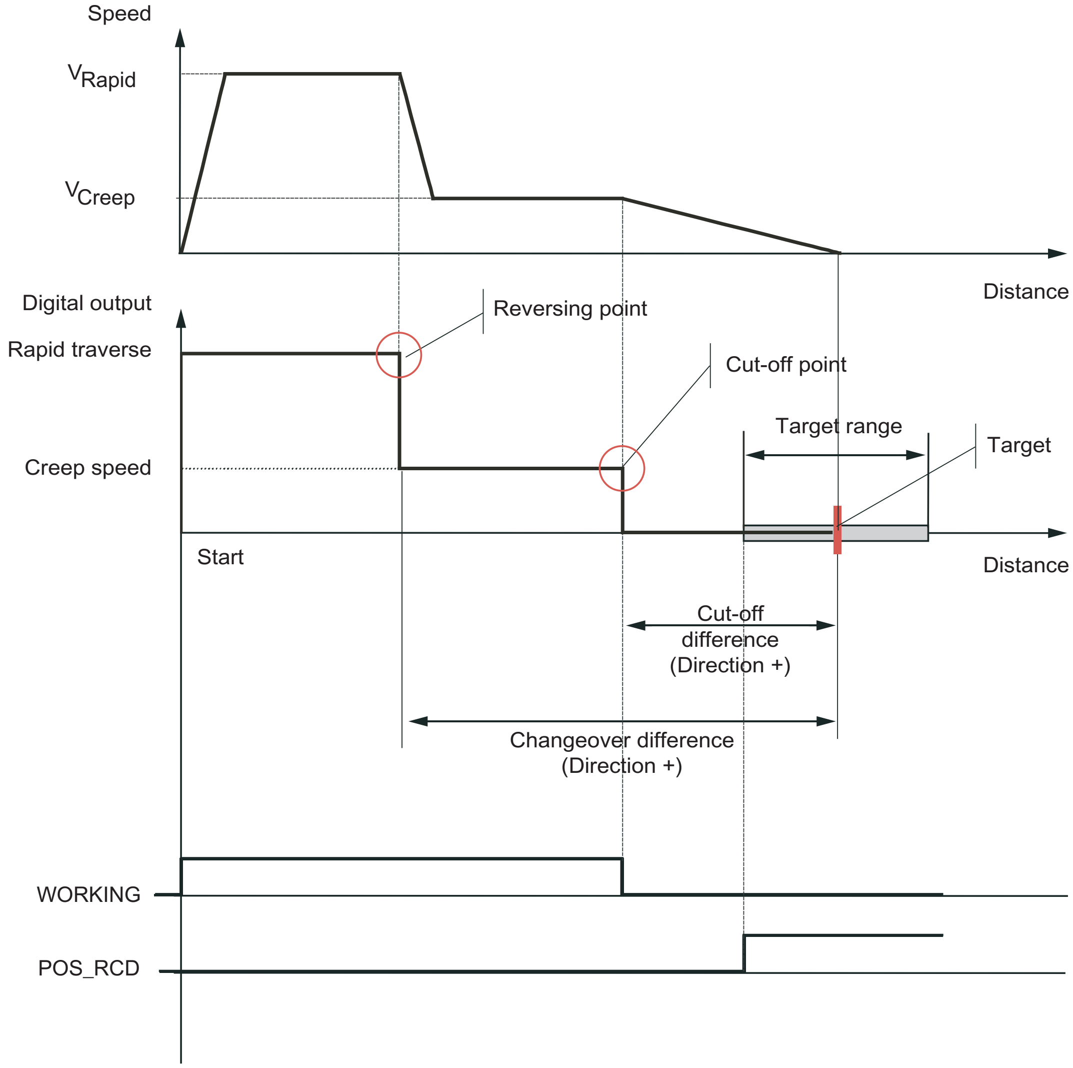
Count once/Count periodically, main count direction down

Job	Valid value range
Write counter directly	2 to +2147483647 ($2^{31} - 1$)
Write load value	2 to +2147483647 ($2^{31} - 1$)
Write the comparison value	1 to +2147483647 ($2^{31} - 1$)
Write the hysteresis	0 to 255
Write pulse duration. Only even values allowed. Odd values are automatically rounded.	0 to 510 ms

5.5.7 Counter FBs

Structure

The view below shows you the various function blocks as described in the following chapters:



5.5.8 Counter Inputs

Pulse/A

Here you connect the count signal or track A of the encoder. You can connect encoders with single, dual or quadruple evaluation.

Direction/B

Here you connect the direction signal or track B of the transducer. You can specify the inversion of the directional signal in your parameter assignment screen.

Note

The inputs are not monitored for missing pulses.

Latch

You can save the actual internal value by generating a positive edge at the digital input "Latch".

This gives you the option of event-dependent count value evaluation. You can output the actual latched value at SFB parameter **LATCHVAL** at every SFB call.

After a STOP-RUN transition of the CPU, LATCHVAL is reset to the start value of the counter.

Hardware Gate

You can start the counter via the digital input "Hardware gate".

5.5.9 Gate Function of the Counter

Basics

You have the choice of two gates for controlling your counter:

- A **software gate** (SW Gate) that is controlled via the user program.

  You can open the software gate with a positive edge at the SFB parameter **SW_GATE**. Reset this parameter to close it.
- A **hardware gate** (HW Gate). You can assign the hardware gate in the parameter assignment screens. The gate opens with a positive edge and closes with a negative edge at the digital input "Hardware gate".

Internal gate

The internal gate represents the logical AND link of the hardware and software gates. Count operation is only enabled if the hardware and software gates are open. The status bit **STS_GATE** (Status internal gate) displays this status.

The software gate configuration is relevant if you have not specifies the use of a hardware gate.

Count operation is initiated, interrupted, resumed and canceled via the internal gate.

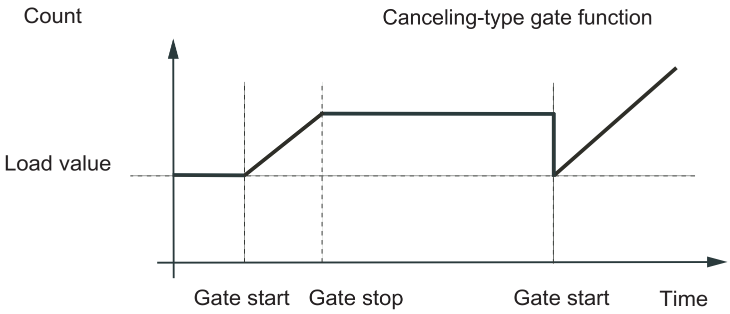
In single count mode, the internal gate is closed automatically on overflow/underflow at the count limits.

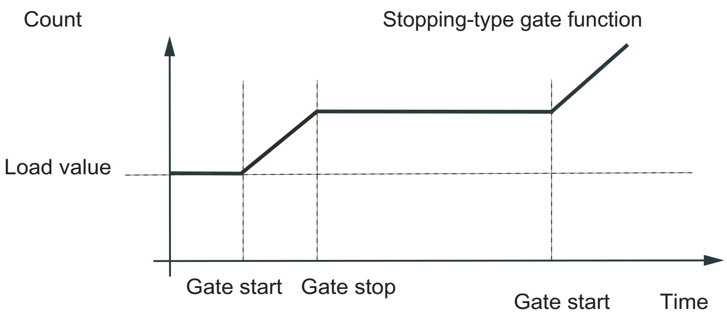
Canceling and Interrupting Gate Functions

In your parameter assignment for the gate functions, you can specify count canceling or count interrupting operation for the internal gate:

- When set to count canceling gate operation, the count restarts operation at the load value after the gate was closed and restarted.
- When set to count interrupting gate operation, the count is resumed at the last actual count value after the gate was closed.

The profiles below show you the response of count canceling and interrupting gate operations:





Controlling the gate via SW gate only

In the parameter assignment screens, specify in "Gate function" parameter how the CPU should respond when the SW gate opens:

- Gate control via SW gate, "Cancel count" configuration

Action	Response
SW gate 0 → 1	Start at load value

- Gate control via SW gate, "Interrupt count" configuration

Action	Response
SW gate 0 → 1	Resume at actual count value

Controlling the gate via SW and HW gates

In the parameter assignment screens, specify in the "Gate function" parameter how the CPU should respond when the SW and HW gates open:

- Controlling the gate via SW and HW gates, "Cancel count" configuration

Requirements	Action	Response
HW gate open	SW gate 0 → 1	Resume at current count value
SW gate open	HW gate 0 → 1	Start at load value

- Controlling the gate via SW and HW gates, "Interrupt count" configuration

Requirement	Action	Response
HW gate open	SW gate 0 → 1	Resume at current count value
SW gate open	HW gate 0 → 1	Resume at current count value

Gate Control via SW and HW Gates in "Single Count" Mode

After the internal gate was closed **automatically**, it can only be reopened by:

- a positive edge generated at the HW gate while the SW is open or
- a positive edge generated at the HW gate with subsequent opening of the SW gate.

5.5.10 Reactions of the Counter Output

Introduction

This section describes the response of the digital output.

Camparison Value

In the CPU, you can store a comparison value that is assigned to the digital output, to the status bit "Status Comparator" (STS_CMP) and to the hardware interrupt. You can enable the digital output depending on the count value and comparison value.

You can specify the comparison value in the parameter assignment screens. In your user program, you can read (JOB_ID = 84 hex) and write (JOB_ID = 04 hex) them via the request interface of the SFB.

Reaction of the Digital Output

You can specify the following response via the parameter assignment screen:

- No comparison
- Count value ≥ comparison value
- Count value ≤ comparison value
- Pulse at comparison value

No comparison

The output is switched as a standard output.

The SFB input parameters CTRL_DO and SET_DO are disabled.

The status bits STS_DO and STS_CMP (status comparator in the IDB) remain reset.

Count Value ≥ Comparison or Count Value ≤ Comparison Value

The comparator switches the output when the comparison value is reached.

Here, you must first set the control bit CTRL_DO.

Status bit STS_CMP shows the result of the compare operation. You can only reset this status bit after all compare conditions are no longer relevant.

Pulse at comparison value

When the count value reaches the comparison value, the comparator enables the output for the duration of the specified pulse period. If you have configured a default direction of count, the output is only enabled after the value specified for the default direction has been reached.

Here, you must first set the control bit CTRL_DO.

The status of bit STS_DO is always identical to the status of the digital output.

Status bit STS_CMP shows the result of the compare operation. You can only reset this status bit after the pulse period has expired.

Status Bit STS_CMP

The status bit STS_CMP indicates whether the corresponding output is enabled or was enabled. You must reset this status bit with RES_STS. If the output is still enabled, the corresponding bit is reset and immediately set again. This status bit is also set if the disabled output (CTRL_DO = FALSE) is enabled with SET_DO.

Note

You must call the SFB twice to reset the status bits with RES_STS.

Controlling the Outputs and the Comparators Simultaneously

If you have selected a comparison function for the output, you can control the output simultaneously with SET_DO (Condition: CTRL_DO=TRUE). Note the following rules:

- The comparator switches the output status from "0" to "1":
  The output can be reset to "0" by the comparator as well as with SET_DO = FALSE. Comparator operation is retriggered by every incoming count pulse. Thus, the output is set or reset according to the result of the comparator operation.
- The comparator switches with SET_DO=TRUE from "0" to "1":
  The output can be reset to "0" by the comparator only with SET_DO = FALSE.

Peculiarites of a "Pulse at Comparison Value" Configuration

Characteristics of the digital output

When the digital is set via control bit SET_DO, it is reset after the pulse period has expired.

- When the pulse period = 0 and the count is value out of range of the comparison value, the output cannot be controlled via SET_DO.
- When the pulse period = 0 and the count value = comparison value, the output can be controlled via SET_DO.

Pulse width

You can specify a pulse period for adaptation to the actuators you are using. The pulse period determines how long the output should be set. You can specify it in increments of 2 ms, from 0 to 510 ms Please note that the count pulses must be longer than the minimum switching time of the digital output.

If the pulse period = 0 the output is set until the compare condition no longer applies.

The start of the pulse period is triggered when the corresponding digital output is set. The pulse period inaccuracy is < 1 ms.

The pulse period is not retriggered if the comparison value is exceeded and reached again during the output of a pulse.

You can customize the pulse period in the parameter assignment screens and write (JOB_ID = 10 hex) and read (JOB_ID = 90 hex) access them in the user program via the SFB request interface.

A change of the pulse period during runtime is not applied until the next pulse.

5.5.11 Effect of Hysteresis on the Counter Modes

Description

The encoder might stop at a certain position and then "dither" around this position. In this state, the count will fluctuate around a particular value. For example, if a comparison value lies within this fluctuation range, the associated output would switch on and off with the rhythm of these fluctuations. The CPU is equipped with an assignable hysteresis to prevent this switching in case of small fluctuations.

You can select a range from 0 through 255. With settings 0 and 1, the hysteresis is disabled.

The hysteresis also acts on the zero crossing and the overflow/underflow.

You can specify the hysteresis in the parameter assignment screens and write (JOB_ID = 08 hex) or read (JOB_ID= 88 hex) the hysteresis in the user program via the job interface of the SFB.

Response to Changes

An active hysteresis remains active after a change. The new hysteresis range is applied when the next comparison value is reached.

Effect when "Counter value ≥ Comparison value" or "Counter value ≥ Comparison value"

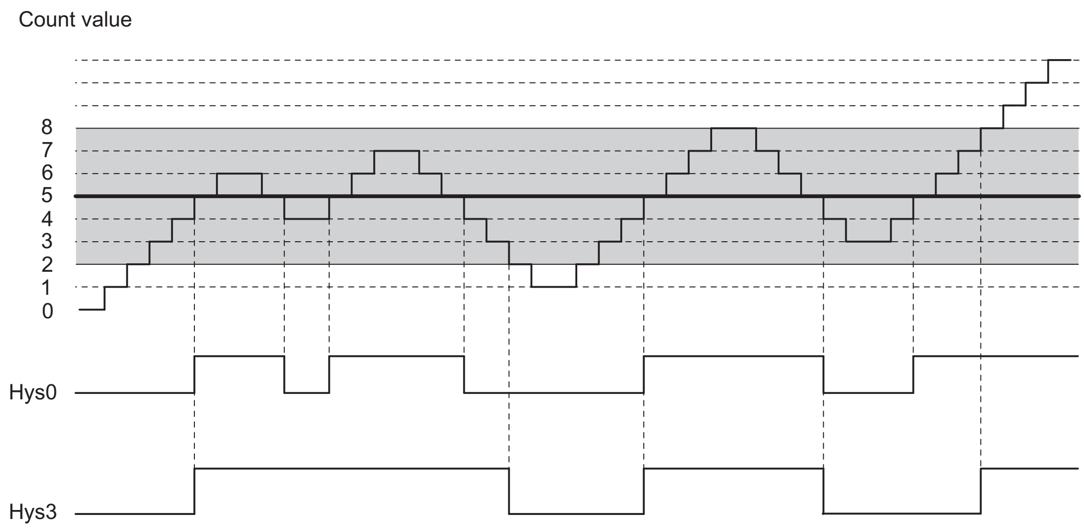
The example in the figure below demonstrates the hysteresis action. The diagram shows the differences in the output behavior when hysteresis values of 0 (= switched off) and 3 are assigned. The comparison value for this example = 5.

The counter is configured as follows:

- "Main count direction up"
- Output "Switch on if counter value ≥ comparison value"

The hysteresis is enabled when the comparison value is reached. When the hysteresis is active, the comparison result remains unchanged.

The hysteresis is disabled when the count value exits the hysteresis range. The comparator then resumes switching according to its comparison conditions.



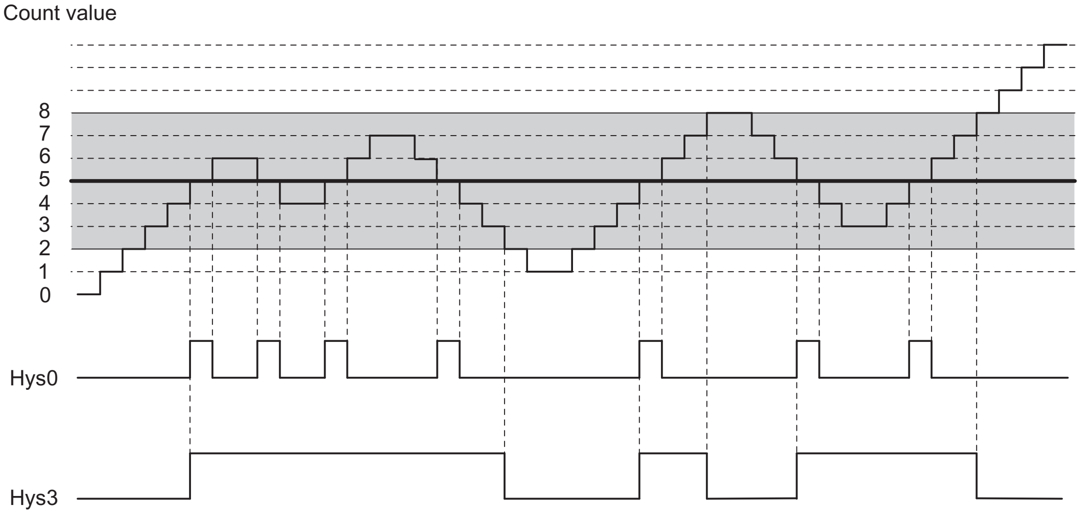
Effect when "Pulse at Comparison Value" and "Pulse Duration Equal to Zero"

The example in the figure below demonstrates the hysteresis action. The diagram shows the differences in the output behavior when hysteresis values of 0 (= switched off) and 3 are assigned. The comparison value for this example = 5.

The counter is configured as follows:

- "No main count direction"
- "Pulse on reaching the comparison value"
- "Pulse duration = 0"

The hysteresis is activated when the comparison conditions are reached. When the hysteresis is active, the comparison result remains unchanged. The hysteresis is no longer active when the count value exits the hysteresis range.



Effect when "Pulse at Comparison Value" and "Pulse Duration Unequal to Zero"

The example in the figure below demonstrates the hysteresis action. The diagram shows the differences in the output behavior when hysteresis values of 0 (= switched off) and 3 are assigned. The comparison value for this example = 5.

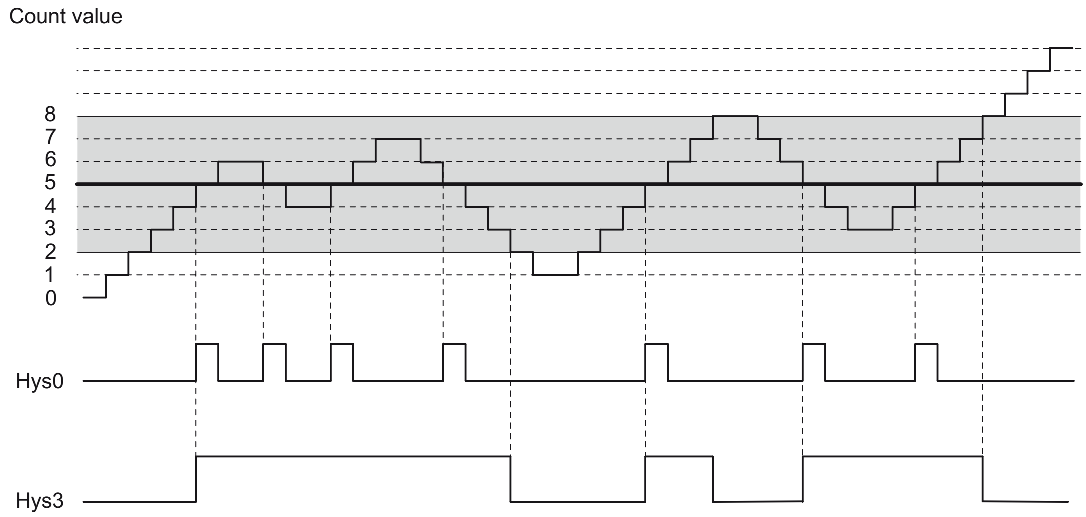
The counter is configured as follows:

- "No main count direction"
- "Pulse on reaching the comparison value"
- "Pulse duration > 0"

When the comparison conditions are reached, the hysteresis is activated and a pulse of the assigned duration is output.

The hysteresis is no longer active when the count value exits the hysteresis range.

The CPU memorizes the count direction when the hysteresis is active. A pulse is output if the count value exits the hysteresis range in the direction of the memorized count direction.



5.5.12 Hardware Interrupt during Count Operation

Setting a Hardware Interrupt

You enable hardware interrupts and specify the hardware interrupt triggering events in the parameter assignment screens:

- Opening of the HW gate while the SW gate is closed
- Closing of the HW gate while the SW gate is open
- Overflow (high counting limit exceeded)
- Underflow (low counting limit fallen below)
- Reaching the comparator (count value = comparison value)
- Counting edge occurred

Note

Triggering a hardware interrupt on each counting edge results in high CPU utilization at higher counting frequencies. If the hardware interrupts in the "Count" submodule occur faster than they can be processed in the hardware interrupt OB (OB 40), this produces the diagnostic "Hardware interrupt lost), provided the diagnostic interrupt is enabled.

High counting frequencies can cause the CPU utilization to become so high that your configured scan cycle monitoring time is exceeded or the CPU communication no longer responds or responds very slowly.

Therefore, we recommend that you do not enable the hardware interrupt for each counting edge unless it is certain that the individual edges are at least 10 ms apart.

5.6 Description of the Frequency Measurement Functions

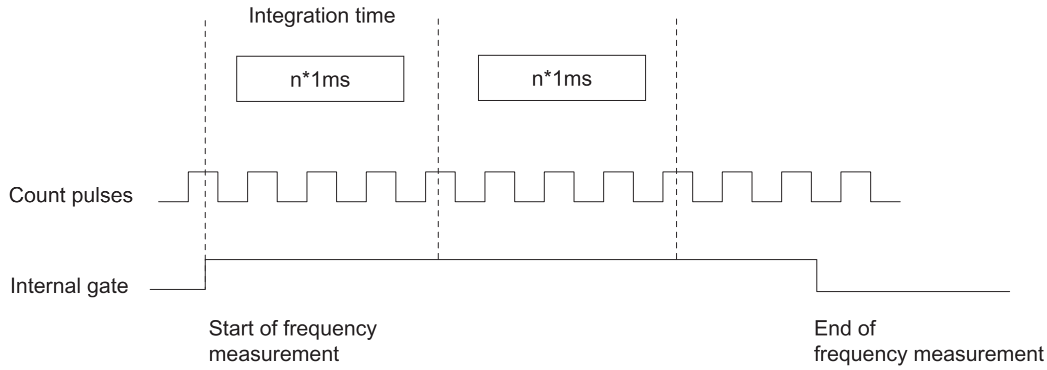
5.6.1 Frequency Measurement, Procedure

Basics

In this operating mode the CPU counts the incoming pulses during a specified integration time and outputs them as a frequency value.

You can set a value for the integration time between 10 ms through 10 000 ms, in increments of 1 ms. You can set the integration time in the parameter assignment screens or you can read and write them from your user program (see Section Controlling the Frequency Counter via the User Program (Page 212)).

The calculated frequency value is supplied in "mHz" units. You can read out this value in your user program with the SFB parameter **MEAS_VAL**. Bit **STS_CMP** is set when a new value is available (for a description of the SFB parameters, refer to Section Controlling the Frequency Counter via the User Program (Page 212)).



Measuring Procedure

The measurement is carried out within the specified integration time. The measurement value is updated after the integration time has expired.

If the period of the measured frequency exceeds the assigned integration time, a value of 0 or the average value is returned as the measured, depending on the parameter assignment.

The value -1 is returned up to the end of the first integration time.

Frequency range

CPU 312C	CPU 313C, CPU 313C-2 DP, PtP	CPU 314C-2 DP, PN/DP, PtP
0 to 10 kHz	0 to 30 kHz	0 to 60 kHz

Direction reversal

If a direction reversal occurs within an integration time, the measured value is indeterminate for this measuring period. You can react to possible process irregularities by evaluating the status bits STS_C_UP, STS_C_DN for direction (see Section Controlling the Frequency Counter via the User Program (Page 212)).

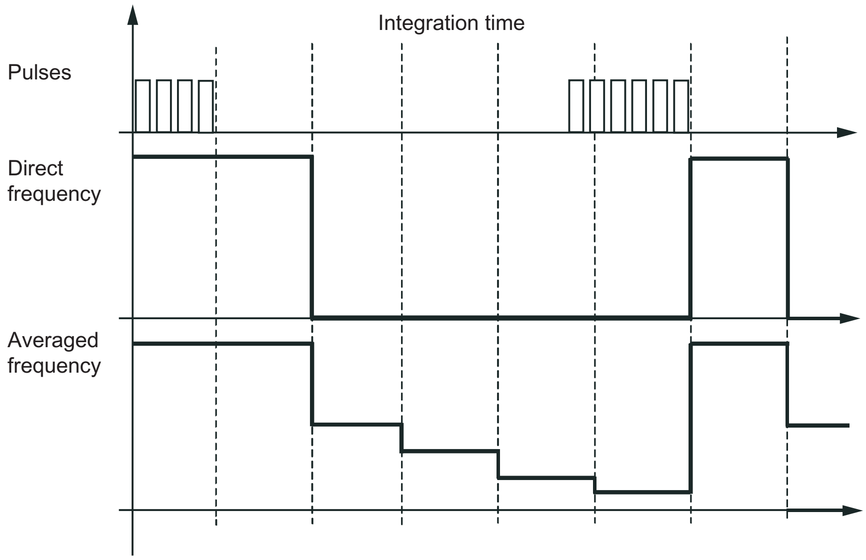
Direct / Averaged Frequency Value

The measured frequency is displayed at the end of the integration time (f ≥ 1 mHz).

if the period of the measured frequency exceeds the assigned integration time:

- With direct frequency, the value "0" is output at the end of the integration time.
- With averaged frequency, the last value is distributed across the subsequent measuring intervals without a positive edge (f ≥ 1 mHz). This prolongs the integration time. In this case, the last measured value is divided by the number of measuring intervals without a positive edge.

Example: If the last measured value was 12 000 mHz, the value 4000 mHz is output after three measuring intervals.



Possible measuring ranges with error indication

Integration time	f_{min}/abs. error	f_{max}/abs. error	f_{max}/abs. error	f_{max}/abs. error
10 s	0.25 Hz/1 mHz	10 kHz/2 Hz	30 kHz/5 Hz	60 kHz/10 Hz
1 s	2.5 Hz/1 mHz	10 kHz/2 Hz	30 kHz/5 Hz	60 kHz/10 Hz
0.1 s	25 Hz/2 mHz	10 kHz/2 Hz	30 kHz/5 Hz	60 kHz/10 Hz
0.01 s	250 Hz/100 mHz	10 kHz/6 Hz	30 kHz/10 Hz	60 kHz/20 Hz

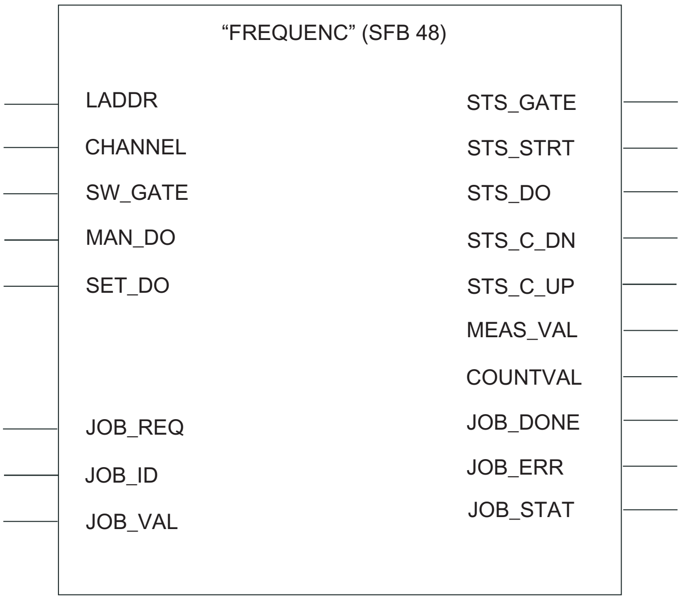
5.6.2 Controlling the Frequency Counter via the User Program

Scope of Functions for SFB FREQUENC

To control the frequency meter from your user program, you use SFB FREQUENC (SFB 48).

The following functionalities are available to you:

- Starting/stopping via software gate SW_GATE
- Enabling/controlling the output DO
- Reading out status bits
- Reading out the current measured value
- Jobs for reading and writing the internal frequency count registers



Input parameters

Parameters	Data type	Address (instance DB)	Description	Value range	Default
LADDR	WORD	0	Submodule I/O address you specified in "HW Config". If the I and O addresses are not equal, the lesser of the two addresses must be specified.	CPU-specific	300 hex
CHANNEL	INT	2	Channel number: CPU 312C CPU 313C CPU 313C-2 DP, PtP CPU 314C-2 DP, PN/DP, PtP	 0 - 1 0 - 2 0 - 3	0
SW_GATE	BOOL	4.0	Software gate For starting/stopping frequency measurement	TRUE/FALSE	FALSE
MAN_DO	BOOL	4.1	Enable manual control of the output	TRUE/FALSE	FALSE
SET_DO	BOOL	4.2	Control output	TRUE/FALSE	FALSE

Note

If you have set the "Characteristics of the output" parameter to "No comparison" via the parameter assignment screen, the following applies:

- The output is switched as a standard output.
- The SFB input parameters MAN_DO and SET_DO are ineffective.
- Status bit STS_DO remains reset.

Input parameters not interconnected to the block (static local data)

Parameters	Data type	Address (instance DB)	Description	Value range	Default
RES_STS	BOOL	32.2	Reset status bits Resets the status bits STS_CMP, STS_OFLW and STS_UFLW. The SFB must be called twice to reset the status bits.	TRUE/FALSE	FALSE

Output parameters

Parameters	Data type	Address (instance DB)	Description	Value range	Default
STS_GATE	BOOL	12.0	Internal gate status	TRUE/FALSE	FALSE
STS_STRT	BOOL	12.1	Status of the hardware gate (Start input)	TRUE/FALSE	FALSE
STS_DO	BOOL	12.2	Output status	TRUE/FALSE	FALSE
STS_C_DN	BOOL	12.3	Status of the down-count Always indicates the last count direction. The value of STS_C_DN is FALSE after the first call of the SFB.	TRUE/FALSE	FALSE
STS_C_UP	BOOL	12.4	Status of the up-count Always indicates the last count direction. After the first SFB call, STS_C_UP has the value TRUE.	TRUE/FALSE	FALSE
MEAS_VAL	DINT	14	Current frequency value	0 to 2^{31} - 1	0
COUNTVAL	DINT	18	Current count value Starts at 0 every time the internal gate opens.	-2^{31} to 2^{31} - 1	0

Output parameters not interconnected to the block (static local data)

Parameters	Data type	Address (instance DB)	Description	Value range	Default
STS_CMP	BOOL	26.3	End of measurement status* The measured value is updated after the integration time has expired. Here, the end of a measurement is signaled via status bit STS_CMP	TRUE/FALSE	FALSE
STS_OFLW	BOOL	26.5	Overflow status*	TRUE/FALSE	FALSE
STS_UFLW	BOOL	26.6	Underflow status*	TRUE/FALSE	FALSE
* Reset with RES_STS					

5.6.3 Reading and Writing to the Request Interface for Frequency Measurement

Introduction

You can use the job interface for reading and writing the frequency registers.

Requirements

The last job must be finished (JOB_DONE = TRUE)

Procedure

1. Assign the following input parameters:

Parameters	Data type	Address (instance DB)	Description	Value range	Default
JOB_REQ	BOOL	4.3	Job request (positive edge)	TRUE/FALSE	FALSE
JOB_ID	WORD	6	Job number:		0
			Job without function	00 hex	
			Write low limit	01 hex	
			Write high limit	02 hex	
			Write integration time	04 hex	
			Read low limit	81 hex	
			Read high limit	82 hex	
			Read integration time	84 hex	
JOB_VAL	DINT	8	Value for write jobs	-2^{31} to $+2^{31} - 1$	0

2. Call the SFB.

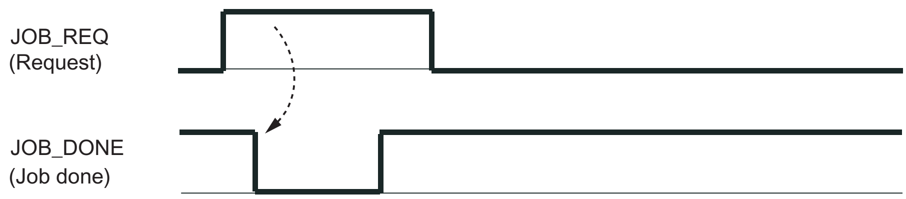
Result

The **output parameters** of the SFB provide the following information:

Parameters	Data type	Address (instance DB)	Description	Value range	Default
JOB_DONE	BOOL	22.0	New job can be started	TRUE/FALSE	TRUE
JOB_ERR	BOOL	22.1	Faulty job	TRUE/FALSE	FALSE
JOB_STAT	WORD	24	Job error number	0 to FFFF hex	0

- The job is processed immediately after the SFB is called. **JOB_DONE** is set to FALSE for the duration of one SFB cycle.
- **JOB_ERR** = TRUE if an error occurred. The precise error cause is displayed in **JOB_STAT**.
- A new job can be started with **JOB_DONE** = TRUE.
- Only for read jobs: Read the current value from the instance DB, parameter JOB_OVAL.

Parameters	Data type	Address (instance DB)	Description	Value range	Default
JOB_OVAL	DINT	28	Output value for read jobs	-2^{31} to $2^{31} - 1$	0



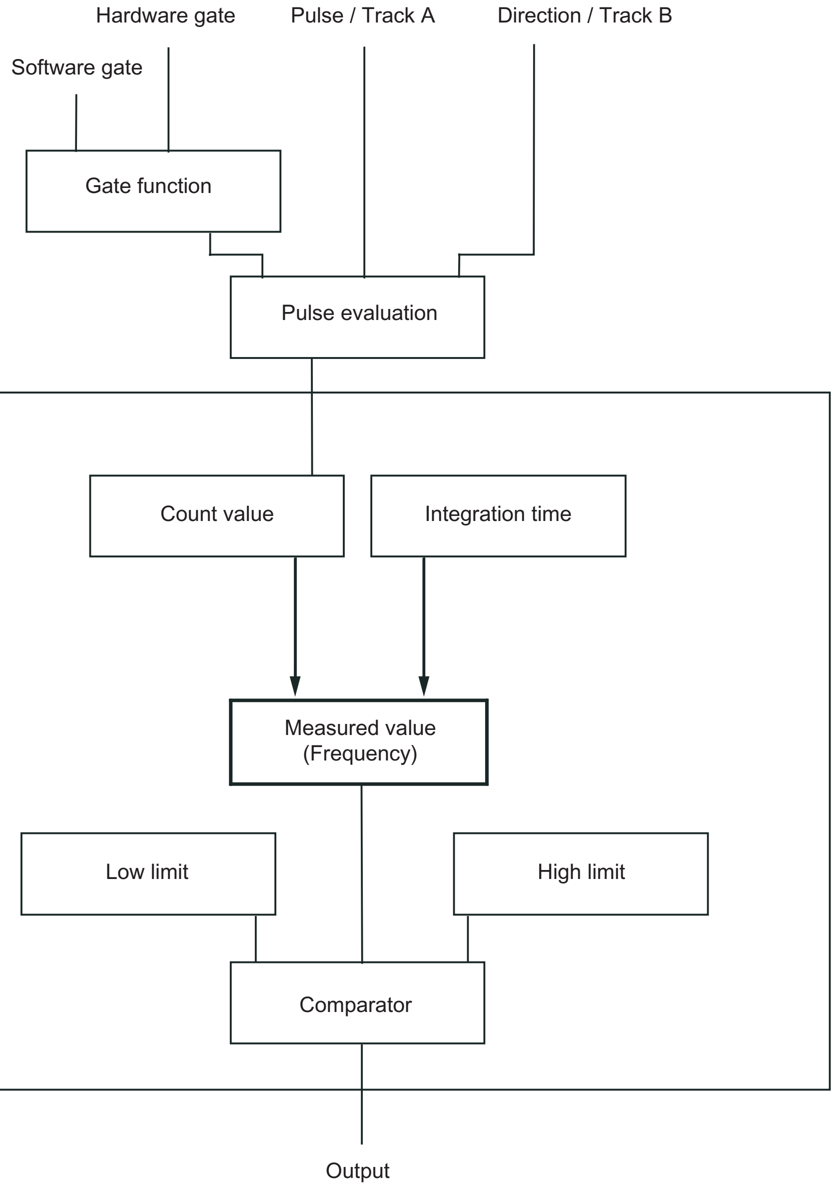
Permissible Value Range for JOB_VAL

Job	Valid value range
Write low limit The low limit must be less than the high limit.	• CPU 312C: 0 to 9 999 999 mHz • CPU 313C, CPU 313C-2 DP, PtP: 0 to 29 999 999 mHz • CPU 314C-2 DP, PN/DP, PtP: 0 to 59 999 999 mHz
Write high limit The high limit must be greater than the low limit.	• CPU 312C: 1 to 10 000 000 mHz • CPU 313C, CPU 313C-2 DP, PtP: 1 to 30 000 000 mHz • CPU 314C-2 DP, PN/DP, PtP: 1 to 60 000 000 mHz
Write integration time	• 10 to 10 000 ms

5.6.4 Function Blocks of the Frequency Counter

Structure

The view displays the various function blocks which are described in the following chapters:



5.6.5 Frequency Counter Inputs

Pulse/A

Here you connect the signal to be measured or track A of the encoder. You can connect encoders with single evaluation mode.

Direction/B

Here you connect the direction signal or track B of the transducer. You can specify the inversion of the directional signal in your parameter assignment screen.

Note

The inputs are not monitored for missing pulses.

Hardware Gate

You can use the digital input "Hardware gate" to control frequency measurements.

5.6.6 Gate Function of the Frequency Measurement

Basics

Two gates are available to you for frequency measurements:

- A **software gate** (SW Gate) that is controlled via the user program.

  You can open the software gate with a positive edge at the SFB parameter SW_GATE. Reset this parameter to close it.

- A **hardware gate** (HW Gate). You can assign the hardware gate in the parameter assignment screens. The gate opens with a positive edge and closes with a negative edge at the digital input "Hardware gate".

Internal gate

The internal gate represents a logical AND link of the HW gate and the SW gate. The measuring cycle is only active if both the HW gates AND the SW gates are open. The status bit **STS_GATE** (Status internal gate) displays this status. Only the SW gate setting is relevant if you have not configured a HW gate.

Controlling the gate via SW gate only

The measurement starts/stops when the HW gate opens/closes.

Controlling the gate via SW and HW gates

Measurement starts when both gates are open. The measurement stops when one of the gates closes.

5.6.7 Reactions of the Frequency Measurement Output

Low/High Limit

In the CPU, you can store a low limit and a high limit, that is assigned to the digital output and to the hardware interrupt. You can enable the digital output depending on the count value and the low/high limit.

You can set the limit values in the parameter assignment screen and write (JOB_ID=01/02 hex) and read (JOB_ID=81/82 hex) the limit values in the user program via the job interface of the SFB.

Characteristics of the Digital Output

You can specify the following behavior in the parameter assignment screens:

- No comparison
- Frequency outside the limits
- Frequency below the low limit
- Frequency above the high limit

No comparison

The output is switched as a standard output.

The SFB input parameters MAN_DO and SET_DO are ineffective.

Status bit STS_DO remains reset.

All other settings

You can control the output either manually or with the comparator:

- Manual control

  Set the SFB parameter MAN_DO to switch over to manual control mode. You can then control the output with SET_DO.

- Control via the comparator

  The comparator performs control when MAN_DO=FALSE.

  The comparator monitors the frequency for the high and low limit values.

  The comparator switches the output when the comparison condition is reached.

Bit STS_UFLW is set if the current frequency falls below the low limit.

Bit STS_OFLW is set if the current frequency exceeds the high limit.

You must use control bit RES_STS to reset these bits.

If the frequency is still or again outside the limits after the measured value is reset, the respective status bit is set once again.

Note

You must call the SFB twice to reset the status bits with RES_STS.

5.6.8 Frequency Measurement and Hardware Interrupt

Setting a Hardware Interrutpt

Enable hardware interrupts in the parameter assignment screens and specify the hardware interrupt triggering events:

- Opening of the HW gate while the SW gate is open.
- Closing of the HW gate while the SW gate is open
- Exceeding the high limit
- Exceeding the low limit
- End of measurement

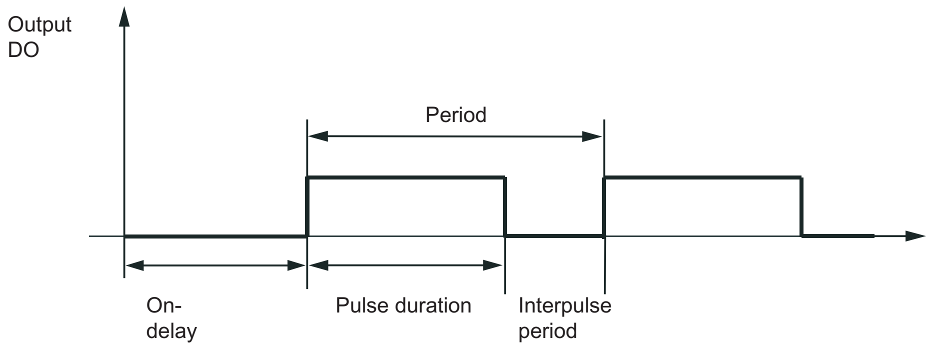
5.7 Description of the Pulse Width Modulation Functions

5.7.1 Procedure for Pulse Width Modulation

Basics

The CPU converts your specified output value (OUTP_VAL) into a pulse train with a respective pulse/pause ratio (Pulse width modulation). This pulse train is output at the digital output DO (output sequence) after the specified ON delay has expired.

Specifications of the Pulse Train	
Output frequency	0 to 2.5 kHz
Minimum pulse width	200 µs
Pulse pause accuracy Accuracy of the ON delay	±(Pulse duration x 100 ppm) ±100 µs ppm = Parts per million 0 to 250 µs The accuracy of the pulse pause can only be maintained if a maximum of one other parameter is changed in addition to the modify value during the same pulse width/pause. If several parameters are modified, the pulse width/pause may have a one-time longer or shorter length than the stated accuracy.



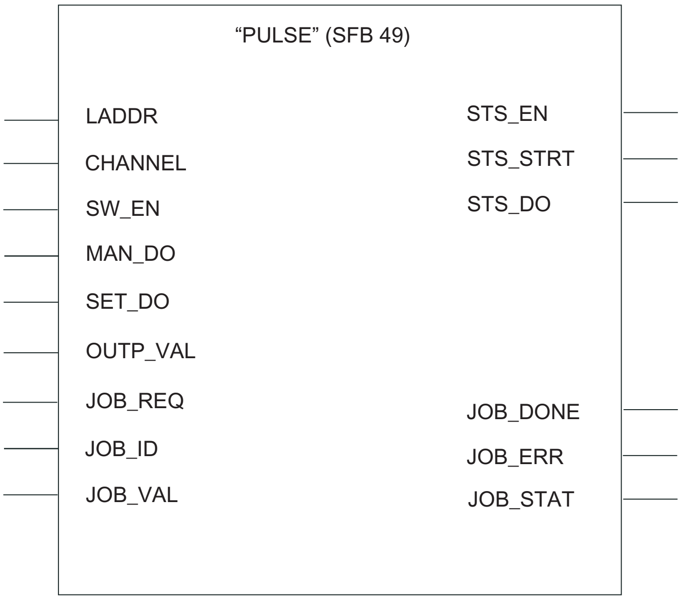
5.7.2 Controlling Pulse Width Modulation via the User Program

Control Functions

To control pulse width modulation in the user program, use SFB PULSE (SFB 49).

The following functionalities are available to you:

- Start/stop via software gate SW_EN
- Enabling/controlling of output DO
- Read status bits
- Input of the output value
- Request for read/write access to registers



Input parameters

Parameters	Data type	Address (instance DB)	Description	Value range	Default
LADDR	WORD	0	Submodule I/O address you specified in "HW Config". If the I and O addresses are not equal, the lesser of the two addresses must be specified.	CPU-specific	300 hex
CHANNEL	INT	2	Channel number: CPU 312C CPU 313C, CPU 313C-2 DP, PtP CPU 314C-2 DP, PN/DP, PtP	0 - 1 0 - 2 0 - 3	0
SW_EN	BOOL	4.0	Software gate Starts/stops data output	TRUE/FALSE	FALSE
MAN_DO	BOOL	4.1	Enables manual control of the output	TRUE/FALSE	FALSE
SET_DO	BOOL	4.2	Control output	TRUE/FALSE	FALSE
OUTP_VAL	INT	6.0	Specifying the output value default: • in per mil • as S7 analog value If you specify a value > 1,000 or 27,648, the CPU limits it to 1,000 or 27,648	0 to 1000 0 to 27648	0

Output parameters

Parameter	Data Type	Address (instance DB)	Description	Range of values	Default
STS_EN	BOOL	16.0	Enable status	TRUE/FALSE	FALSE
STS_STRT	BOOL	16.1	Status of the hardware gate (start input)	TRUE/FALSE	FALSE
STS_DO	BOOL	16.2	Output status	TRUE/FALSE	FALSE

5.7.3 Reading and Writing to the Request Interface for Pulse Width Modulation

Description

The job interface is available to you for reading and writing the registers.

Requirements

The last job must be finished (JOB_DONE = TRUE).

Procedure

1. Assign the following input parameters:

Parameters	Data type	Address (instance DB)	Description	Value range	Default
JOB_REQ	BOOL	8	Job trigger (positive edge)	TRUE/FALSE	FALSE
JOB_ID	WORD	10	Job number: • Job without function • Write period • Write on-delay • Write minimum pulse duration • Read period • Read on-delay • Read minimum pulse duration	 00 hex 01 hex 02 hex 04 hex 81 hex 82 hex 83 hex	0
JOB_VAL	DINT	12	Value for write jobs	-2^{31} to $+2^{31}-1$	0

2. Call the SFB.

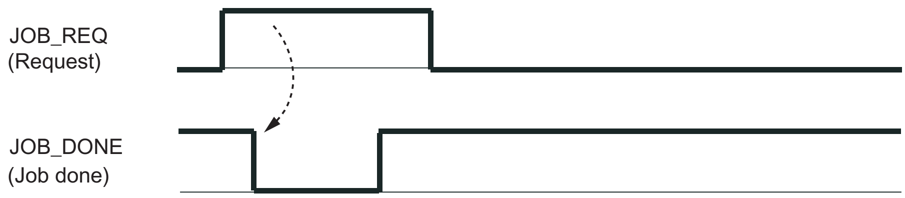
Result

The **output parameters** of the SFB provide the following information:

Parameters	Data type	Address (instance DB)	Description	Value range	Default
JOB_DONE	BOOL	16.3	New job can be started	TRUE/FALSE	TRUE
JOB_ERR	BOOL	16.4	Faulty job	TRUE/FALSE	FALSE
JOB_STAT	WORD	18	Job error number	0 to FFFF hex	0

- The job is processed immediately after the SFB is called. **JOB_DONE** is set to FALSE for the duration of one SFB cycle.
- **JOB_ERR** = TRUE if an error occurred. The precise error cause is displayed in **JOB_STAT**.
- A new job can be started with **JOB_DONE** = TRUE.
- Only for read jobs: Read the current value from the instance DB, parameter JOB_OVAL.

Parameters	Data type	Address (instance DB)	Description	Value range	Default
JOB_OVAL	DINT	20	Output value for read jobs	-2^{31} to $2^{31}-1$	0



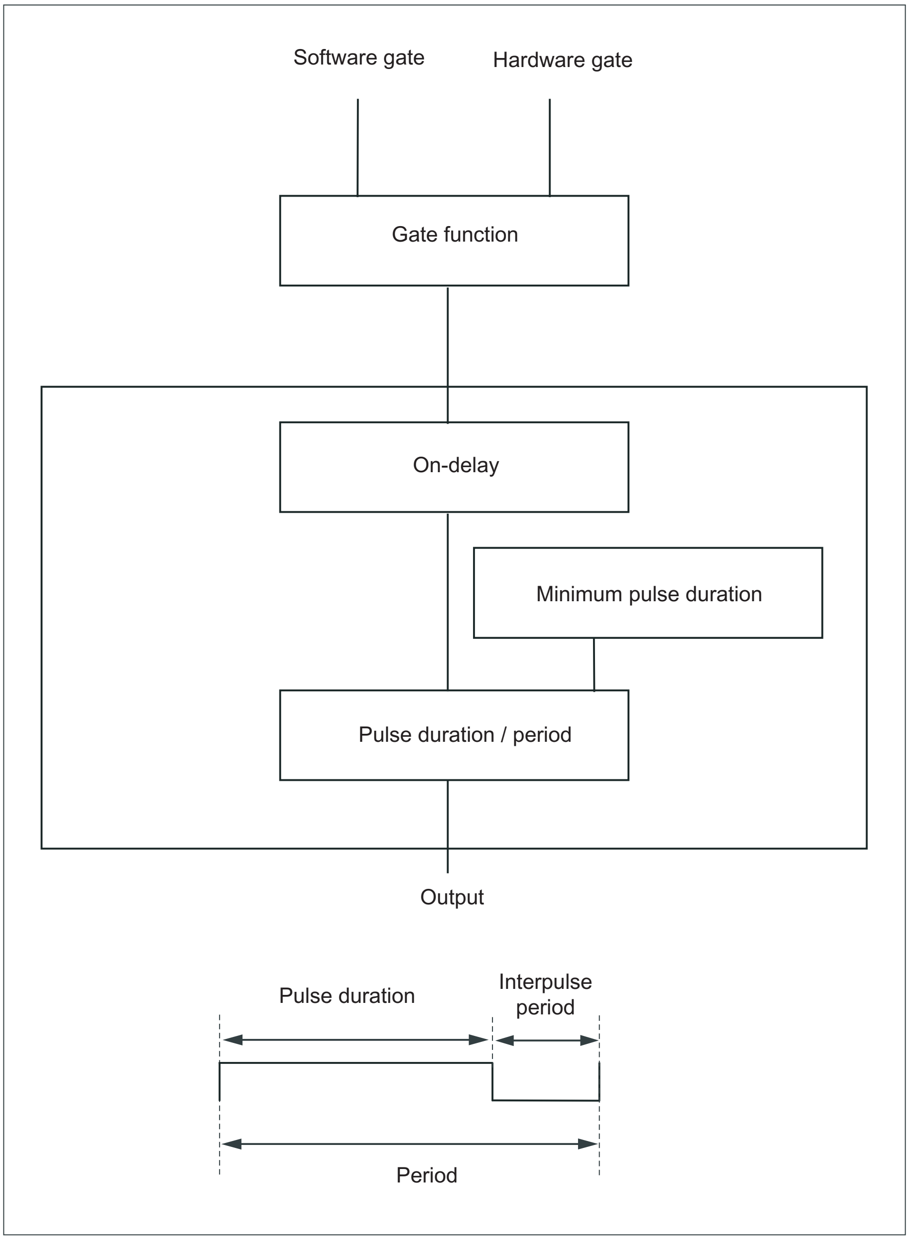
Permissible value range for JOB_VAL

Job	Valid value range	
Write period	• Time base 0.1 ms: • Time base 1 ms:	• 4 to 65535 • 1 to 65535
Write on-delay	• 0 to 65535	
Write minimum pulse duration	• Time base 0.1 ms: • Time base 1 ms:	• 2 to period/2 • 0 to period/2 (0 = 0.2 ms)

5.7.4 Function Blocks for Pulse Width Modulation

Structure

The view displays the various function blocks which are described in the following chapters:



5.7.5 Gate Function of Pulse Width Modulation

Basics

Two gates are available to you for pulse width modulation operations:

- A **software gate** (SW Gate) that is controlled via the user program.

  You can open the software gate at a positive edge at the SFB parameter SW_EN. Reset this parameter to close it.
- A **hardware gate** (HW Gate). You can assign the hardware gate in the parameter assignment screens. It is controlled via the digital input "Hardware gate".

Internal gate

The internal gate is used to start and stop pulse width modulation.

This internal gate represents a logical AND link of the HW and SW gates. The status bit **STS_EN** indicates the status of the internal gate.

After it is enabled, the on delay starts. The pulse train is output on expiration of the on delay time. This output sequence runs infinitely when the enable signal is set.

Controlling the gate via SW gate only

Pulse width modulation starts/stops when the SW gate opens/closes.

Controlling the gate via SW and HW gates

- A start of pulse width modulation is only possible if you open the SW gate in a first step and in the second step generate a positive edge at the HW gate input:

Requirements	Action
SW gate open	HW gate 0 → 1

- You can only stop pulse width modulation at the negative edge on the SW gate. The HW gate can have any state.

Requirement	Action
None, any HW gate state	SW gate 1 → 0

5.7.6 Setting the Parameters for the Pulse Train

Parameters, Their Settings and Control Possibilities

Parameter	Configurable in: Parameter assignment screen	Controlled via: SFB
Time base	Yes	-
Output format	yes	-
Output Value	-	Write
Period	yes	Read/Write
On-delay	yes	Read/write
Minimum pulse duration	yes	Read/write

Time base

Use the time base to specify the resolution and value range of the on delay, period and minimum pulse width.

Output format

In the output format parameter, declare the range of the output value:

Output format	Value range
Per mil	0 to 1000
S7 analog value	0 to 27648

Output value

Declare the output value as input parameter OUTP_VAL at the SFB.

The CPU uses this specified output value to calculate the pulse width:

Output format	Pulse width
Per mil	(Output value / 1000) x period
S7 analog value	(Output value / 27648) x period

If you change the output value during the pulse output sequence, the CPU immediately calculates the new period and switches the output accordingly. This can be used to extend or reduce the length of one period:

- If you change it when the signal is low, and the new output value is lower than the old one, the length of the period is extended in one cycle because the new Interpulse period is now longer.
- If you change it when the signal is low and the new output value is greater than the old one, the length of the period is reduced in one cycle because the new interpulse period is now shorter.
- If you change it when the signal is high, and the new output value is lower than the old one, the length of the period is extended in one cycle because the new low signal is now longer.
- If you change it when the signal is high, and the new output value is greater than the old one, the period remains constant.

Period

With the period you define the length of the output pulse/interpulse sequence.

Period = Time base x specified numeric value

The period must be at least twice the length of the minimum pulse width.

When you change the period while the pulse train is output, the CPU immediately calculates the new pulse/interpulse period and switches the output accordingly. This can be used to extend or reduce the length of one period:

- If you change it when the signal is low, and the new period is shorter than the previous one, the period generated is shorter than the previous and longer than the new one.
- If you change it when the signal is low, and the new period is longer than the previous one, the period generated is longer than the previous, but shorter than the new one.
- If you change it when the signal is high, and the new period is shorter than the previous one, a single period is generated that is shorter than the previous, but longer than the new one.
- If you change it when the signal is high, and the new period is longer than the previous one, a single period is generated that is longer than the previous, but shorter than the new one.

On-delay

Represents the time interval between the start of the output sequence and the output of the first pulse.

On delay = Time base x specified numeric value

If you change the on delay while it is active, the new setting is immediately applied:

- If the new on delay is shorter than the previous one an on delay can be generated once, shorter than the previous signal, but longer than the new one.
- The new on delay is used if it is longer than the previous delay time.

Minimum pulse duration

All output low/high signals shorter than the minimum pulse width are suppressed.

Minimum pulse width = Time base x specified numeric value

If you change the minimum pulse width while a code sequence is output the new value is immediately applied:

- The output is set to "1" when you apply a change while the signal is low and the interpulse width is shorter than the new minimum pulse width.
- The interpulse width is output when you apply a change while the signal is low and the width is greater than the new minimum pulse width.
- The output is set to "0" when you apply a change while the signal is high and the pulse width is shorter than the new minimum pulse width.
- The pulse is output when you apply a change while the signal is high and the pulse width is longer than the new minimum pulse width.

	Time base: 0.1 ms	Time base: 1 ms
Period	4 to 65535	1 to 65535
On-delay	0 to 65535	0 to 65535
Minimum pulse duration	2 to period/2	0 to period/2 (0 = 0.2 ms)

5.7.7 Reactions of the Pulse Width Modulation Output

Introduction

This section describes the characteristics of the digital output.

You can control the output manually, or you can use it to output the pulse train.

Manual Control

Set the SFB parameter MAN_DO to switch over to manual control mode. You can then control the output with SET_DO.

Output of the Pulse Train

MAN_DO=FALSE can be used to output the pulse train.

5.7.8 Pulse Width Modulation and Hardware Interrupt

Setting a Hardware Interrutpt

Enable hardware interrupts in the parameter assignment screens and specify the hardware interrupt triggering events:

- Opening of the HW gate while the SW gate is closed.

5.8 Error Handling and Interrupts

5.8.1 Error Display

Basics

Errors are indicated by

- Error Messages at the System Function Block (SFB)
- Diagnostic interrupt

You can trigger a hardware interrupt on specific events.

5.8.2 Error Messages at the System Function Block (SFB)

Overview

The SFB indicates the errors listed in the table below.

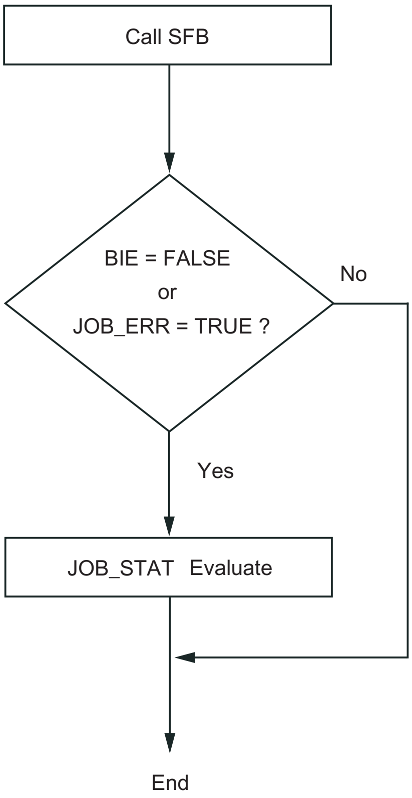
Type of error	Errors are displayed via SFB parameters	The error number is displayed in the SFB parameters
Job error	JOB_ERR = TRUE	JOB_STAT
System error	BIE = FALSE	JOB_STAT

Job errors occur during job interpretation/execution. The JOB_ERR parameter is set to TRUE when an error occurs.

A system error is triggered in the case of a basic parameter assignment error, e.g. "Incorrect operating mode". The system error is indicated with BIE = FALSE.

Parameter JOB_STAT describes the cause of error in closer detail. The possible error numbers are listed in Section Error Lists (Page 243).

Error evaluation



5.8.3 Configuring Diagnostic Interrupts

Using the Diagnostic Interrupt

When errors occur, for example

- parameter assignment errors (module data) and
- "Loss of hardware error signal"

you can trigger a diagnostic interrupt. The diagnostic interrupt is displayed with incoming and outgoing error events.

In your user program, you can immediately respond to errors with the help of a diagnostic interrupt.

Procedure

1. Enable the diagnostic interrupt in the parameter assignment screen. (Interrupt selection: Diagnostics or diagnostics/process)
2. Incorporate the diagnostic interrupt OB (OB 82) in your user program.

Response to an Error with Diagnostic Interrupt

- Current operation is not influenced by the diagnostic interrupt.
- The CPU operating system calls OB82 in the user program.

> **Note**
>
> If the corresponding OB is not loaded the CPU switches to STOP when an interrupt is triggered.

- The CPU switches on the SF LED.
- The error is reported in the diagnostics buffer of the CPU as "incoming". An error is not indicated as "outgoing" until all pending errors are cleared.

How a Diagnostic Interrupt is Evaluated in the User Program

After a diagnostic interrupt is triggered, you can evaluate OB 82 to check which diagnostic interrupt is pending.

- If the address of your submodule is entered in OB 82, byte 6 + 7 (OB 82_MDL_ADDR) the diagnostic interrupt was triggered by a counter in your CPU.
- Bit 0 of byte 8 in OB 82, bit 0 (Faulty module) is set as long as any errors are queued.
- Bit 0 of byte 8 in OB 82 is reset after all queued errors are reported as "outgoing".
- You can evaluate the bytes 8 and 11 to determine the cause of error.

OB 82, byte 8	Description:
Bit 0	Faulty module
Bit 1	-
Bit 2	-
Bit 3	-
Bit 4	-
Bit 5	-
Bit 6	-
Bit 7	Configuration error

OB 82, byte 11	Description:
Bit 0	-
Bit 1	-
Bit 2	-
Bit 3	-
Bit 4	-
Bit 5	-
Bit 6	Hardware interrupt loss
Bit 7	-

Hardware interrupt loss

With enabled hardware interrupt, the CPU reports the error "Lost hardware interrupt" if the interrupt triggering event occurs again before the previous interrupt was cleared.

5.8.4 Configuring the Hardware Interrupt

Using the Hardware Interrupt

You can trigger a hardware interrupt on specific events. With the help of the hardware interrupt, you can respond immediately to events in your user program.

Procedure

1. Enable hardware interrupts in the parameter assignment screen "Basic parameters". (Interrupt selection: Hardware or diagnostics/hardware)
2. In the respective "Count", "Frequency counting " and "Pulse-width modulation" parameter assignment screen, enable the hardware interrupt triggering events.
3. Integrate the hardware interrupt OB (OB 40) in your user program.

Response to a Hardware Interrupt

The CPU operating system calls OB 40 in the user program.

Note

If the corresponding OB is not loaded the CPU switches to STOP when an interrupt is triggered.

Evaluation of a Hardware Interrupt in the User Program

After a hardware interrupt is triggered, you can evaluate OB 40 to check which hardware interrupt is pending.

- If the address of your submodule is entered in OB 40, bytes 6 + 7 (OB 40_MDL_ADDR), the hardware interrupt was triggered by the counter in your CPU.
- To determine the precise cause, evaluate bytes 8 to 11 of DWord OB 40_POINT_ADDR.

Counting

OB 40, byte 8	Description:
Bit 0	Channel 0: HW gate opening
Bit 1	Channel 0: HW gate closing
Bit 2	Channel 0: Overflow/Underflow
Bit 3	Channel 0: Comparator has responded
Bit 4	Channel 1: HW gate opening
Bit 5	Channel 1: HW gate closing
Bit 6	Channel 1: Overflow/Underflow
Bit 7	Channel 1: Comparator has responded

OB 40, byte 9	Description:
Bit 0	Channel 2: HW gate opening
Bit 1	Channel 2: HW gate closing
Bit 2	Channel 2: Overflow/Underflow
Bit 3	Channel 2: Comparator has responded
Bit 4	Channel 3: HW gate opening
Bit 5	Channel 3: HW gate closing
Bit 6	Channel 3: Overflow/Underflow
Bit 7	Channel 3: Comparator has responded

OB 40, byte 10	Description:
Bit 0	Channel 0: Counting edge occurred
Bit 1	Channel 0: -
Bit 2	Channel 0: -
Bit 3	Channel 0: -
Bit 4	Channel 1: Counting edge occurred
Bit 5	Channel 1: -
Bit 6	Channel 1: -
Bit 7	Channel 1: -

OB 40, byte 11	Description:
Bit 0	Channel 2: Counting edge occurred
Bit 1	Channel 2: -
Bit 2	Channel 2: -
Bit 3	Channel 2: -
Bit 4	Channel 3: Counting edge occurred
Bit 5	Channel 3: -
Bit 6	Channel 3: -
Bit 7	Channel 3: -

Frequency Measurement

OB 40, byte 8	Description:
Bit 0	Channel 0: HW gate opening
Bit 1	Channel 0: HW gate closing
Bit 2	Channel 0: Violation of the high limit/low limit of the frequency
Bit 3	Channel 0: End of measurement
Bit 4	Channel 1: HW gate opening
Bit 5	Channel 1: HW gate closing
Bit 6	Channel 1: Violation of the high limit/low limit of the frequency
Bit 7	Channel 1: End of measurement

OB 40, byte 9	Description:
Bit 0	Channel 2: HW gate opening
Bit 1	Channel 2: HW gate closing
Bit 2	Channel 2: Violation of the high limit/low limit of the frequency
Bit 3	Channel 2: End of measurement
Bit 4	Channel 3: HW gate opening
Bit 5	Channel 3: HW gate closing
Bit 6	Channel 3: Violation of the high limit/low limit of the frequency
Bit 7	Channel 3: End of measurement

OB 40, Bytes 10 and 11: Not assigned

Pulse-width modulation

OB 40, byte 8	Description:
Bit 0	Channel 0: HW gate opening
Bit 1	Channel 0: -
Bit 2	Channel 0: -
Bit 3	Channel 0: -
Bit 4	Channel 1: HW gate opening
Bit 5	Channel 1: -
Bit 6	Channel 1: -
Bit 7	Channel 1: -

OB 40, byte 9	Description:
Bit 0	Channel 2: HW gate opening
Bit 1	Channel 2: -
Bit 2	Channel 2: -
Bit 3	Channel 2: -
Bit 4	Channel 3: HW gate opening
Bit 5	Channel 3: -
Bit 6	Channel 3: -
Bit 7	Channel 3: -

OB 40, Bytes 10 and 11: Not assigned

5.9 Installation of Examples

Using Examples

The examples (program and description) are found on the CD-ROM included in your documentation. You can also download them from the Internet. The project consists of several commented S7 programs of various complexity and aim.

The Readme.wri on the CD describes how to install the samples. After the installation the samples are stored in the catalog **...\STEP7\EXAMPLES\ZDt26_02_TF_____31xC_Cnt**.

5.10 Specifications

5.10.1 Functions

Counting

	CPU 312C	CPU 313C, CPU 313C-2 DP, PtP	CPU 314C-2 DP, PN/DP, PtP
Maximum frequency (Track A/Pulse, track B/direction, hardware gate and latch)	10 kHz	30 kHz	60 kHz
min. pulse width / min. interpulse width	48 μs	16 μs	8 μs
max. cable length (at max. count frequency)	100 m	100 m	50 m
Counting range	- 2 147 483 648 (-2^{31}) to + 2 147 483 647 (2^{31} - 1)		

Frequency counting

Table 5- 1 Frequency Range

	CPU 312C	CPU 313C, CPU 313C-2 DP, PtP	CPU 314C-2 DP, PN/DP, PtP
Frequency range (Track A/pulse, Track B/direction and hardware gate)	0 to 10 kHz	0 to 30 kHz	0 to 60 kHz
min. pulse width/min. pulse cause	48 μs	16 μs	8 μs
max. cable length (at max. count frequency)	100 m	100 m	50 m

Table 5- 2 Possible measuring ranges with error indication

Integration time	f_{min}/abs. error	f_{max}/abs. error	f_{max}/abs. error	f_{max}/abs. error
10 s	0.25 Hz/1 mHz	10 kHz/2 Hz	30 kHz/5 Hz	60 kHz/10 Hz
1 s	2.5 Hz/1 mHz	10 kHz/2 Hz	30 kHz/5 Hz	60 kHz/10 Hz
0.1 s	25 Hz/2 mHz	10 kHz/2 Hz	30 kHz/5 Hz	60 kHz/10 Hz
0.01 s	250 Hz/100 mHz	10 kHz/6 Hz	30 kHz/10 Hz	60 kHz/20 Hz

Pulse Width Modulation

Specifications of the Pulse Train	
Output frequency	0 to 2.5 kHz
Minimum pulse width	200 μs
Pulse pause accuracy Accuracy of the ON delay	±(Pulse duration x 100 ppm) ±100 μs ppm = Parts per million 0 to 250 μs The accuracy of the pulse pause can only be maintained if a maximum of one other parameter is changed in addition to the modify value during the same pulse width/pause. If several parameters are modified, the pulse width/pause may have a one-time longer or shorter length than the stated accuracy.

	CPU 312C	CPU 313C, CPU 313C-2 DP, PtP	CPU 314C-2 DP, PN/DP, PtP
Filter frequency (Hardware gate)	10 kHz	30 kHz	60 kHz
min. pulse width	48 μs	16 μs	8 μs
max. cable length	100 m	100 m	50 m

5.10.2 Incremental encoders

Connectable Incremental Encoders

Asymmetric 24 V incremental encoders with two pulses with an electrical phase difference of 90° are supported.

Signal Evaluation

Increments

An increment designates a signal period of the two encoder tracks A and B. This value is specified on the rating plate of the encoder and/or in the technical specifications for the encoder.

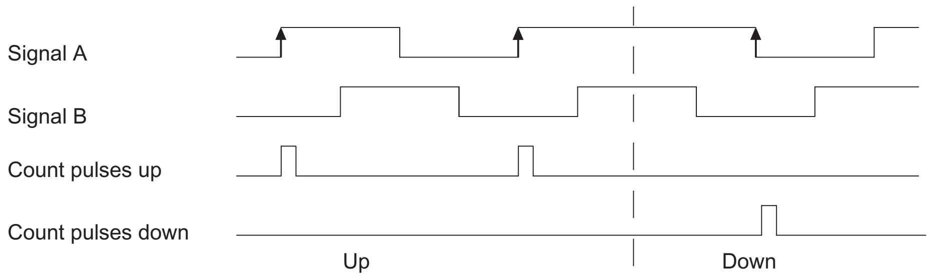
Edges at Tracks A and B

The CPU can count the edges of the tracks. Normally, the CPU evaluates only the edge at track A (single evaluation). You can,achieve a higher resolution with multiple evaluation. In the parameter assignment screens, you can specify single, double, or quadruple evaluation of the tracks.

Multiple evaluation is only possible for asymmetric 24 V incremental encoders equipped with two tracks A and B that have a 90° phase shift.

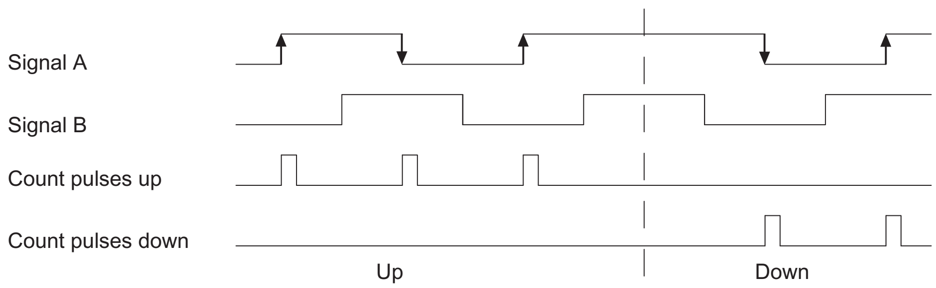
Single evaluation

In single evaluation mode, only one edge of track A is evaluated, that is, up-count pulses are measured on a positive edge on A and low level on B, and down-count pulses are measured on a positive edge on A and high level on B.



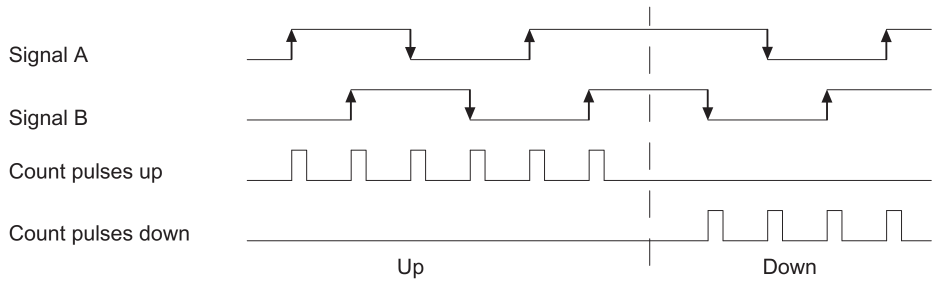
Double evaluation

In double evaluation mode, the positive and negative edge of track A are evaluated. The level of track B determines whether up- or down-count pulses are generated.



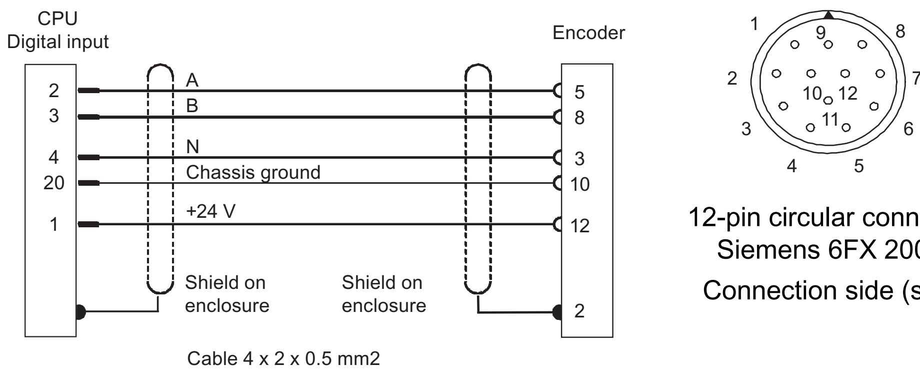
Quadruple evaluation

In quadruple evaluation mode, the positive and negative edges of tracks A and B are evaluated. The logical levels of tracks A and B determine whether up- or down-count pulses are generated.



Wiring diagram of the incremental encoder Siemens 6FX 2001-4 (Up = 24 V; HTL)

The figure below shows the wiring diagram for the incremental encoder Siemens 6FX 2001-4xxxx (Up = 24 V; HTL):



12-pin circular connector socket
Siemens 6FX 2003-0CE12
Connection side (solder side)

5.10.3 Error Lists

Basics

In the table below you can find the description of the error IDs for the SFB output JOB_STAT. The error ID consists of an event class and number.

Job Error

Event class 01 (01H): "Counting, parameter assignment errors in SFB parameters (SFB 47)"		
Event ID	Event text	Remedy
(01)21H	Compare value too low	Refer to the online help or user manual for information on valid values
(01)22H	Compare value too high	
(01)31H	Hysteresis too narrow	
(01)32H	Hysteresis too wide	
(01)41H	Pulse width too short	
(01)42H	Pulse width too long	
(01)51H	Load value too low	
(01)52H	Load value too high	
(01)61H	Count value too low	
(01)62H	Count value too high	
(01)FFH	Invalid job ID	

Event class 02 (02HH): "Measuring frequency, configuration error SFB parameter (SFB 48)"		
Event No.	Event Text	Remedy
(02)21H	Integration time too low	Refer to the online help or user manual for information on valid values
(02)22H	Integration time too high	
(02)31H	Frequency low limit too low	
(02)32H	Frequency low limit too high	
(02)41H	Frequency high limit too low	
(02)42H	Frequency high limit too high	
(02)FFH	Invalid job number	

Event class 04 (04H): "Pulse-width modulation, parameter assignment error in the SFB parameters (SFB 49)"		
Event No.	Event Text	Remedy
(04)11H	Period too short	Refer to the online help or user manual for information on valid values
(04)12H	Period too long	
(04)21H	On delay too short	
(04)22H	On delay too long	
(04)31H	Minimum pulse width too short	
(04)32H	Minimum pulse width too long	
(04)FFH	Invalid job number	

System error

Event class 128 (08H): "Configuration error in global SFB parameters"		
Event No.	Event Text	Remedy
(80)01H	Wrong operating mode or parameter assignment error.	In "HW Config", specify the correct operating mode or use an SFB that matches the set mode.
(80)09H	Invalid channel number	Valid channel numbers: • CPU 312C: 0-1 • CPU 313C, CPU 313C-2 DP, PtP: 0-2 • CPU 314C-2 DP, PN/DP, PtP: 0-3
On system error the BIE bit is set to FALSE		

5.10.4 Module Parameters (Overview)

Introduction

The following tables provide an overview of the module parameters that can be set in the parameter assignment screens.

Basic parameters

Parameter	Description	Value range	Default
Interrupt selection	Here you can select which interrupts the technological function should trigger.	• None • Diagnostic • Hardware • Diagnostic and hardware	None

Continuous, once-only, and periodic counting

Parameter	Description	Value range	Default
Main count direction	• None: No restriction of the counting range • Up: Restricts the counting range in the up direction. Counter starts at 0 or the load value and counts in the positive direction up to the assigned end value -1. It then jumps back to the load value at the next positive encoder pulse. • Down: Restricts the counting range in the down direction. Counter starts at the assigned start value or the load value and counts in the negative direction to 1. It then jumps to start value at the next negative encoder pulse.	• None • Up (not with continuous counting) • Down (not with continuous counting)	None
End value/	• End value with main count direction up	2 to 2147483647 (2^{31} - 1)	2147483647 (2^{31} - 1)
Start value	• Start value with main count direction down	2 to 2147483647 (2^{31} - 1)	2147483647 (2^{31} - 1)

Parameter	Description	Value range	Default
Gate function	• Cancel the counting operation: The count restarts at the load value when the gate is closed and restarted. • Stop the counting operation: The count is stopped when the gate closes and resumed at the last actual count when the gate opens again.	• Cancel count • Stop count	Cancel count
Comparison value	The count value is compared with the comparison value. See also the parameter "Characteristics of the Output" • No main count direction • Main count direction up • Main count direction down	-2^{31} to $+2^{31} - 1$ -2^{31} to end value - 1 1 to $+2^{31} - 1$	0
Hysteresis	The hysteresis is used to avoid frequent output switching actions if the count value lies within the range of the comparison value. 0 and 1 mean: Hysteresis switched off.	0 to 255	0
Max. frequency: Counting signals/ HW gate	CPU 312C	10, 5, 2, 1 kHz	10 kHz
	CPU 313C, 313C-2 DP, PtP	30, 10, 5, 2, 1 kHz	30 kHz
	CPU 314C-2 DP, PN/DP, PtP	60, 30, 10, 5, 2, 1 kHz	60 kHz
Max. frequency: Latch	CPU 312C	10, 5, 2, 1 kHz	10 kHz
	CPU 313C, 313C-2 DP, PtP	30, 10, 5, 2, 1 kHz	10 kHz
	CPU 314C-2 DP, PN/DP, PtP	60, 30, 10, 5, 2, 1 kHz	10 kHz
Signal evaluation	• The counting and direction signals are connected to the input • A rotary encoder is connected to the input (single, double, or quadruple evaluation)	• Pulse/direction • Rotary encoder, single • Rotary encoder, double • Rotary encoder, quadruple	Pulse/direction
HW gate	• Yes: Gate control via SW and HW gates • No: Gate control via SW gate only.	• Yes • No	No
Inverted count direction	• Yes: Inverted "Direction" input signal. • No: "Direction" input signal is not inverted.	• Yes • No	No

Parameter	Description	Value range	Default
Characteristics of the output	The output and the "Comparator" (STS_CMP) status bit are set, dependent on this parameter.	• No comparison • Count value ≥ comparison value • Count value ≤ comparison value • Pulse at comparison value	No comparison
Pulse duration	With the setting "Characteristics of the output: Pulse at comparison value", you can specify the pulse duration for the output signal. Only even values are possible.	0 to 510 ms	0
Assignment of input data	You can select whether the count value or the period can be read at a maximum counting frequency of 1 kHz in the input data (I data) of the "Count" submodule. If the maximum counting frequency is greater than 1 kHz, only "Count value" is possible.	Count value Period	Count value
Time base	You can specify whether the period is to be measured in units of 125 ns or 1 µs at a maximum counting frequency of 1 kHz. If the maximum counting frequency is greater than 1 kHz, the period is not measured.	125 ns 1 µs	125 ns
Hardware interrupt: HW gate opening	A hardware interrupt is generated when the hardware gate opens while the software gate is open.	• Yes • No	No
Hardware interrupt: HW gate closing	A hardware interrupt is generated when the hardware gate closes while the software gate is open.	• Yes • No	No
Hardware interrupt: On reaching the comparator	A hardware interrupt is generated on reaching the comparator.	• Yes • No	No
Hardware interrupt: Overflow	A hardware interrupt is generated on overflow (exceeding the high counting limit).	• Yes • No	No
Hardware interrupt: Underflow	A hardware interrupt is generated on underflow (low counting limit is fallen below).	• Yes • No	No
Hardware interrupt: Counting edge	You can select whether a hardware interrupt is generated on each counting edge at a maximum counting frequency of 1 kHz. This hardware interrupt cannot be selected if the maximum counting frequency is greater than 1 kHz. Generating a hardware interrupt on each counting edge results in high CPU utilization at higher counting frequencies.	• Yes • No	No

Frequency counting

Parameter	Description	Value range	Default
Integration time	Time window in which the incoming pulses are measured.	1 to 10 000 ms	100
Low limit	The measured value is compared with the low limit. The status bit "Underflow" (STS_UFLW) is set when the low limit is fallen below. The low limit must be less than the high limit.	CPU 312C: 0 to 9 999 999 mHz CPU 313C, CPU 313C-2 DP, PtP: 0 to 29 999 999 mHz CPU 314C-2 DP, PN/DP, PtP: 0 to 59 999 999 mHz	0
High limit	The measured value is compared to the high limit. The status bit "Overflow" (STS_OFLW) is set when the high limit is exceeded. The high limit must be greater than the low limit.	CPU 312C: 1 to 10 000 000 mHz CPU 313C, CPU 313C-2 DP, PtP: 1 to 30 000 000 mHz CPU 314C-2 DP, PN/DP, PtP: 1 to 60 000 000 mHz	CPU 312C: 10 000 000 mHz CPU 313C, CPU 313C-2 DP, PtP: 30 000 000 mHz CPU 314C-2 DP, PN/DP, PtP: 60 000 000 mHz
Maximum counting frequency	CPU 312C	10, 5, 2, 1 kHz	10 kHz
	CPU 313C, 313C-2 DP, PtP	30, 10, 5, 2, 1 kHz	30 kHz
	CPU 314C-2 DP, PN/DP, PtP	60, 30, 10, 5, 2, 1 kHz	60 kHz
Output measured value	If the period of the measured frequency exceeds the assigned integration time: • With "direct" frequency, the value "0" is output at the end of the integration time. • With "averaged" frequency, the last value is distributed across the subsequent measuring intervals without an edge (f ≥ 1 mHz). This prolongs the integration time. Here, the last measured value is divided by the number of measuring intervals without an edge.	• Direct • Averaged	Direct
Signal evaluation	• The count and direction signals are connected to the input • A rotary encoder with single evaluation is connected to the input	• Pulse/direction • Rotary encoder, single	Pulse/direction
Inverted count direction	• Yes: Inverted "Direction" input signal. • No: "Direction" input signal is not inverted.	• Yes • No	No

Parameter	Description	Value range	Default
HW gate	• Yes: Gate control via SW and HW gates. • No: Gate control via SW gate only.	• Yes • No	No
Characteristics of the output	The measured value is compared with the high and low limits. The output is switched, depending on this parameter.	• No comparison • Outside the limits • Below the low limit • Above the high limit	No comparison
Hardware interrupt: HW gate opening	A hardware interrupt is generated when the hardware gate opens while the software gate is open.	• Yes • No	No
Hardware interrupt: HW gate closing	A hardware interrupt is generated when the hardware gate closes while the software gate is open.	• Yes • No	No
Hardware interrupt: End of measurement	A hardware interrupt is generated at the end of measurement.	• Yes • No	No
Hardware interrupt: Low limit exceeded	A hardware interrupt is generated when the low limit is fallen below.	• Yes • No	No
Hardware interrupt: High limit exceeded	A hardware interrupt is generated when the high limit is exceeded.	• Yes • No	No

Pulse-Width Modulation

Parameter	Description	Value range	Default
Output format	Output format	• Per mil • S7 analog value	• Per mil
Time base	Time base for • On-delay • Period • Minimum pulse duration	• 0.1 ms • 1.0 ms	• 0.1 ms
On-delay	Time interval between the start of the output sequence and pulse output.	0 - 65535	0
Period	Defines the length of the output sequence in terms of pulse duration and interpulse period.	• Time base 0.1 ms: 4 to 65535 • Time base 1 ms: 1 to 65535	20 000
Minimum pulse duration	Output pulses/interpulse periods shorter than the minimum pulse duration are suppressed. With a time base of 1 ms and 0 value the minimum pulse duration is set internally to 0.2 ms.	• Time base 0.1 ms: 2 to period/2 • Time base 1 ms: 0 to period/2	2
HW gate	• Yes: Gate control via SW and HW gates. • No: Gate control via SW gate only	• Yes • No	No
Filter frequency: **HW gate**	CPU 312C	10, 5, 2, 1 kHz	10 kHz
	CPU 313C, 313C-2 DP, PtP	30, 10, 5, 2, 1 kHz	30 kHz
	CPU 314C-2 DP, PN/DP, PtP	60, 30, 10, 5, 2, 1 kHz	60 kHz
Hardware interrupt: **HW gate opening**	A hardware interrupt is generated when the hardware gate opens while the software gate is open.	• Yes • No	No

5.10.5 Instance DBs of the SFBs

Parameters of SFB 47 "COUNT"

Parameters	Declaration	Data type	Address (instance DB)	Description	Value range	Default
LADDR	IN	WORD	0	Submodule I/O address you specified in "HW Config". If the I/O addresses are not equal you must specify the lower one of both.	CPU-specific	300 hex
CHANNEL	IN	INT	2	Channel number:		0
				• CPU 312C	0 - 1	
				• CPU 313C, CPU 313C-2 DP, PtP	0 - 2	
				• CPU 314C-2 DP, PN/DP, PtP	0 - 3	
SW_GATE	IN	BOOL	4.0	Software gate For counter start/stop	TRUE/FALSE	FALSE
CTRL_DO	IN	BOOL	4.1	Enable output	TRUE/FALSE	FALSE
SET_DO	IN	BOOL	4.2	Control output	TRUE/FALSE	FALSE
JOB_REQ	IN	BOOL	4.3	Initiates the job (positive edge)	TRUE/FALSE	FALSE
JOB_ID	IN	WORD	6	Job number:		0
				• Job without function	00 hex	
				• Write the count value	01 hex	
				• Write load value	02 hex	
				• Write the comparison value	04 hex	
				• Write the hysteresis	08 hex	
				• Write the pulse width	10 hex	
				• Read load value	82 hex	
				• Read comparison value	84 hex	
				• Read hysteresis	88 hex	
				• Read pulse width	90 hex	
JOB_VAL	IN	DINT	8	Value for write jobs.	-2^{31} to $+2^{31} - 1$	0
STS_GATE	OUT	BOOL	12.0	Internal gate status	TRUE/FALSE	FALSE
STS_STRT	OUT	BOOL	12.1	Status of the hardware gate (Start input)	TRUE/FALSE	FALSE
STS_LTCH	OUT	BOOL	12.2	Status of the latch input	TRUE/FALSE	FALSE

Parameters	Declaration	Data type	Address (instance DB)	Description	Value range	Default
STS_DO	OUT	BOOL	12.3	Output status	TRUE/FALSE	FALSE
STS_C_DN	OUT	BOOL	12.4	Down direction status The last count direction is always displayed. The value of STS_C_DN is FALSE after the first call of the SFB.	TRUE/FALSE	FALSE
STS_C_UP	OUT	BOOL	12.5	Up direction status The last count direction is always displayed. The value of STS_C_UP is TRUE after the first call of the SFB.	TRUE/ FALSE	FALSE
COUNTVAL	OUT	DINT	14	Actual count value	-2^{31} to $2^{31} - 1$	0
LATCHVAL	OUT	DINT	18	Actual latch value	-2^{31} to $2^{31} - 1$	0
JOB_DONE	OUT	BOOL	22.0	New job can be started	TRUE/FALSE	TRUE
JOB_ERR	OUT	BOOL	22.1	Faulty job	TRUE/FALSE	FALSE
JOB_STAT	OUT	WORD	24	Job error number	0 to FFFF hex	0
STS_CMP	STAT	BOOL	26.3	Comparator status* Status bit STS_CMP indicates whether the comparison condition for the comparator is or was met. STS_CMP also indicates that the output was set (STS_DO = TRUE)	TRUE/FALSE	FALSE
STS_OFLW	STAT	BOOL	26.5	Overflow status*	TRUE/FALSE	FALSE
STS_UFLW	STAT	BOOL	26.6	Underflow status*	TRUE/FALSE	FALSE
STS_ZP	STAT	BOOL	26.7	Zero mark status* This bit is only set when counting without main counting direction. Indicates the zero mark. This is also set when the counter is set to 0 or if it starts counting at load value = 0.	TRUE/FALSE	FALSE
JOB_OVAL	STAT	DINT	28	Output value for read jobs.	-2^{31} to $2^{31} - 1$	0
RES_STS	STAT	BOOL	32.2	Reset status bits Resets the status bits STS_CMP, STS_OFLW, STS_UFLW and STS_ZP. The SFB must be called twice to reset the status bits.	TRUE/FALSE	FALSE
* Reset with RES_STS						

Parameters of SFB48 "FREQUENC"

Parameters	Declaration	Data type	Address (instance DB)	Description	Value range	Default
LADDR	IN	WORD	0	Submodule I/O address you specified in "HW Config". If the I and O addresses are not equal, the lesser of the two addresses must be specified.	CPU-specific	hex
CHANNEL	IN	INT	2	Channel number: • CPU 312C	0 - 1	0
				• CPU 313C, CPU 313C-2 DP, PtP	0 - 2	
				• CPU 314C-2 DP, PN/DP, PtP	0 - 3	
SW_GATE	IN	BOOL	4.0	Software gate For starting/stopping frequency measurements	TRUE/FALSE	FALSE
MAN_DO	IN	BOOL	4.1	Enable manual output control	TRUE/FALSE	FALSE
SET_DO	IN	BOOL	4.2	Control output	TRUE/FALSE	FALSE
JOB_REQ	IN	BOOL	4.3	Job trigger (positive edge)	TRUE/ FALSE	FALSE
JOB_ID	IN	WORD	6	Job number:		0
				• Job without function	00 hex	
				• Write the lower limit	01 hex	
				• Write the high limit	02 hex	
				• Write the integration time	04 hex	
				• Read low limit	81 hex	
				• Read high limit	82 hex	
				• Read integration time	84 hex	
JOB_VAL	IN	DINT	8	Value for write jobs	-2^{31} to $+2^{31} - 1$	0
STS_GATE	OUT	BOOL	12.0	Internal gate status	TRUE/FALSE	FALSE
STS_STRT	OUT	BOOL	12.1	Status of the hardware gate (Start input)	TRUE/FALSE	FALSE
STS_DO	OUT	BOOL	12.2	Output status	TRUE/FALSE	FALSE

Parameters	Declaration	Data type	Address (instance DB)	Description	Value range	Default
STS_C_DN	OUT	BOOL	12.3	Down direction status The last count direction is always displayed. The value of STS_C_DN is FALSE after the first call of the SFB.	TRUE/FALSE	FALSE
STS_C_UP	OUT	BOOL	12.4	Up direction status The last count direction is always displayed. The value of STS_C_UP is TRUE after the first call of the SFB.	TRUE/FALSE	FALSE
MEAS_VAL	OUT	DINT	14	Actual value of the frequency	to 2^{31} - 1	0
COUNTVAL	OUT	DINT	18	Actual count value Starts at 0 every time the internal gate opens.	-2^{31} to2^{31} - 1	0
JOB_DONE	OUT	BOOL	22.0	New job can be started	TRUE/FALSE	TRUE
JOB_ERR	OUT	BOOL	22.1	Faulty job	TRUE/FALSE	FALSE
JOB_STAT	OUT	WORD	24	Job error number	0 to FFFF hex	0
STS_CMP	STAT	BOOL	26.3	End of measurement status* The measurement value is updated on every expiration of the integration time. Here, the end of a measurement is reported via status bit STS_CMP	TRUEFALSE	FALSE
STS_OFLW	STAT	BOOL	26.5	Overflow status*	TRUE/FALSE	FALSE
STS_UFLW	STAT	BOOL	26.6	Underflow status*	TRUE/FALSE	FALSE
JOB_OVAL	STAT	DINT	28	Output value for read jobs	-2^{31} to 2^{31} - 1	0
RES_STS	STAT	BOOL	32.2	Reset status bits Resets the status bits STS_CMP, STS_OFLW and STS_UFLW. The SFB must be called twice to reset the status bits.	TRUE/FALSE	FALSE
* Reset with RES_STS						

Parameters of SFB 49 "PULSE"

Parameters	Declaration	Data type	Address (instance DB)	Description	Value range	Default
LADDR	IN	WORD	0	Submodule I/O address you specified in "HW Config". If the I and O addresses are not equal, the lesser of the two addresses must be specified.	CPU-specific	hex
CHANNEL	IN	INT	2	Channel number: • CPU 312C	0 - 3	0
				• CPU 313C, CPU 313C-2 DP, PtP	0 - 2	
				• CPU 314C-2 DP, PN/DP, PtP	0 - 3	
SW_EN	IN	BOOL	4.0	Software gate Starts/stops data output	TRUE/ FALSE	FALSE
MAN_DO	IN	BOOL	4.1	Enable manual output control	TRUE/FALSE	FALSE
SET_DO	IN	BOOL	4.2	Control output	TRUE/FALSE	FALSE
OUTP_VAL	IN	INT	6	Specifying the output value default: • in per mil • as S7 analog value If you specify a value > 1,000 or 27,648 the CPU limits it to 1,000 or 27,648	0 to 1000 0 to 27648	0
JOB_REQ	IN	BOOL	8.0	Job trigger (positive edge)	TRUE/FALSE	FALSE
JOB_ID	IN	WORD	10	Job number:		0
				• Job without function	00 hex	
				• Write the period length	01 hex	
				• Write the on delay	02 hex	
				• Write the minimum pulse width	04 hex	
				• Read the period length	81 hex	
				• Read Rise-time delay	82 hex	
				• Read the minimum pulse width	84 hex	
JOB_VAL	IN	DINT	12	Value for write jobs	-2^{31} to $+2^{31} - 1$	0

Parameters	Declaration	Data type	Address (instance DB)	Description	Value range	Default
STS_EN	OUT	BOOL	16.0	Enable status	TRUE/FALSE	FALSE
STS_STRT	OUT	BOOL	16.1	Status of the hardware gate (Start input)	TRUE/FALSE	FALSE
STS_DO	OUT	BOOL	16.2	Output status	TRUE/FALSE	FALSE
JOB_DONE	OUT	BOOL	16.3	New job can be started	TRUE/FALSE	TRUE
JOB_ERR	OUT	BOOL	16.4	Faulty job	TRUE/FALSE	FALSE
JOB_STAT	OUT	WORD	18	Job error number	0 to FFFF hex	0
JOB_OVAL	STAT	DINT	20	Output value for read jobs	-2^{31} to $2^{31} - 1$	0

Point-to-point communication 6

6.1 Overview

6.1.1 Product Description

Basics

You can use the serial interface with PtP communication to exchange data between programmable logic controllers, computers or simple devices. Communication between the partner devices runs on the basis of serial asynchronous transmission.

The integrated serial interface of the CPU 313/314C-2 PtP offers communication access via the X27 (RS422/485) interface.

The following protocols are available to you:

- CPU 313C-2 PtP: ASCII, 3964(R)
- CPU 314C-2 PtP: ASCII, 3964(R) and RK 512

You can configure the communication mode via the parameter assignment screen.

You can transfer a maximum of 1,024 bytes. Possible rates of transmission are 19.2 kbaud with full duplex and 38.4 kbaud with half duplex.

6.1.2 Communication Partner

Examples of Communication Partners

With the serial interface of the CPU it is possible to establish PtP connections to various Siemens modules and third-party products. Some examples are given below:

- SIMATIC S5, via 3964(R)/RK 512 with a corresponding interface module on S5 side
- Siemens BDE terminals of the ES2 family, via 3964(R) driver
- MOBY I (ASM 420/421, SIM), MOBY L (ASM 520) and logging station ES 030K, via 3964(R) driver
- SIMOVERT and SIMOREG (USS Protocol), via the ASCII driver (ET 200S SI RS 422/485) with respective protocol adaptation in a STEP 7 program
- PCs, via the 3964(R) protocol (development tools are available for programming on PC: PRODAVE DOS 64R (6ES5 897-2UD11) for MS-DOS, PRODAVE WIN 64R (6ES5 897-VD01) for Windows or ASCII-drivers
- Barcode readers, via the 3964(R) or ASCII drivers.
- PLCs of other manufacturers, via RK512, the 3964(R) or ASCII-driver
- Additional devices with simple protocol structures, via the ASCII driver with respective protocol adaptation.
- Other devices also equipped with 3964(R)/RK 512

6.1.3 Components for PtP Communication

Use of the Components

The protocols for the serial connection are integrated in the **CPU**. Your communication partner is connected via the serial interface.

You use a shielded cable for the **connecting cable**. The connecting cables for various communication partners are described in Section Cables (Page 347).

As **communication partners**, you can connect devices equipped with an RS422/485 interface that support the respective protocol.

Use the **PG/PC** to:

- Assign the CPU parameters using the parameter assignment screens for the technological functions of the CPU.
- Program CPU SFBs which you can integrate directly in your user program.
- Commission and test the CPU with the help of the standard STEP 7 interface (monitoring functions and variable table).

6.1.4 Properties of the (RS422/485) Interface

Definition

The X27 (RS422/485) interface represents a differential voltage interface for serial data transmission in compliance with the X27 standard.

- In RS422 mode, data are transmitted across a four-wire serial cable (four-wire operation). Two conductors (differential signal) are available for the send direction and two for the receive direction. This means you can send and receive data at the same time (Full Duplex operation).
- In RS485 mode data are transmitted across a two-wire serial cable (two-wire operation). The two wires (differential signal) are available alternately for the send and receive direction. This means you can either send or receive data at a given time (Half Duplex operation). After a send operation, the cable is immediately switched over to receive mode (transmitter is switched to high-impedance).

The operating mode is selected via the parameter assignment screen.

Properties

The X27 (RS422/485) interface has the following properties and meets the following requirements:

• Type	Differential voltage interface
• Front connector	15-pin Sub-D female connector with screw interlock
• Max. transmission rate	38.4 kbps (Half Duplex)
• Standard	DIN 66259 Part 1 and 3, EIA-RS 422/485, CCITT V.11

6.1.5 Serial Transmission of a Character

Basics

There are various networking alternatives for the transfer of data between two or more communication partners. The PtP-connection between two communication partners is the simplest way of information exchange. Data are transmitted serial in a PtP communication.

Serial Data Transmission

With serial transmission, the individual bits of each information byte are transmitted bit-wise in a fixed order. Data exchange with the communication partner is handled automatically via the serial interface. For this operation the CPU is equipped with three different drivers.

- ASCII driver
- 3964(R) procedure
- RK 512

Half Duplex/Full Duplex

For data transmission we differentiate between:

- Half Duplex (ASCII driver, 3964(R) protocol , RK 512)
  Data exchange alternates between the communication partners in both directions. Half Duplex means that either a transmit or a receive operation is carried out at one time. Single data flow control characters can here form the exception (e. g. XON/XOFF). These can also be transmitted/received during send/receive operation.
- Full Duplex (ASCII Driver)
  Data is exchanged simultaneously between the communication partners. Data can therefore be sent and received at one time. Every communication partner must be capable of operating a simultaneous transmitter and receiver unit.

RS 485 mode (2-wire) allows only Half Duplex operation via ASCII driver without flow control.

Asynchronous Data Transmission

Serial data are transmitted asynchronously. The so-called time base synchronism (a fixed timing code used in the transmission of a fixed character string) is only upheld during transmission of a character. Each character to be sent is preceded by a synchronization impulse, or start bit. The end of the character transmission is signaled by the stop bit.

Declarations

Apart from the start and stop bits, serial data transmission between the two communication partners requires additional declarations. These include:

- the transmission rate (baud rate)
- the character delay time and, if required, the acknowledgment delay time
- the parity
- the number of data bits
- the number of stop bits

Character frame

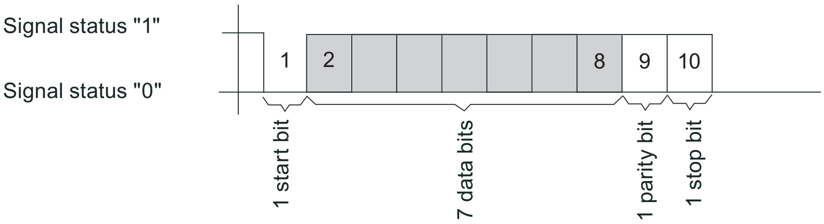
Data are transmitted in a character frame via serial interface. Two data formats are available for every character frame. Operation with 7 Data bits and no parity bit is not supported. You can configure the desired data transmission format in the parameter configuration tool.

Note

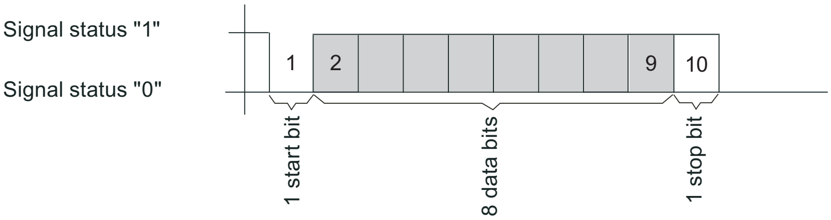
The character frame with 7 data bits and no parity is not supported.

The following exemplary view illustrates the two data formats of the 10-bit character frame:

7 data bits: 1 start bit, 7 data bits, 1 parity bit, 1 stop bit

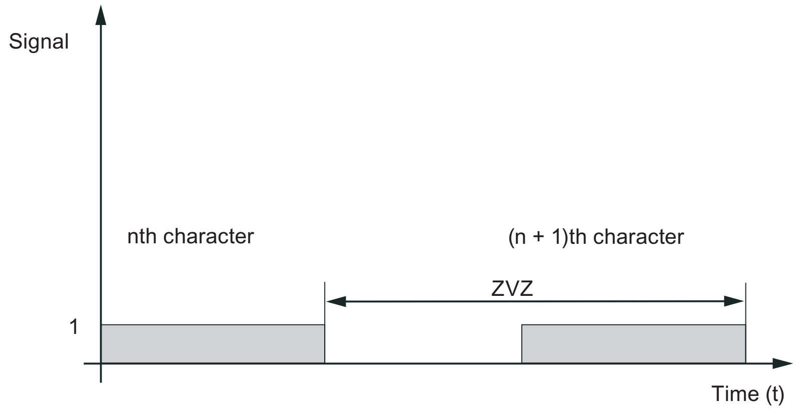


8 data bits: 1 start bit, 8 data bits, 1 stop bit



Character delay time

The following picture illustrates the maximum permissible time interval between two received characters of a message = character delay time:



6.2 Wiring

6.2.1 Wiring Rules

Connecting cable

- The cables must be shielded.
- The cable shields must be terminated on both ends.

Shield connection element

You can use the shield connection element to connect all shielded cables to ground via direct connection to mounting rail.

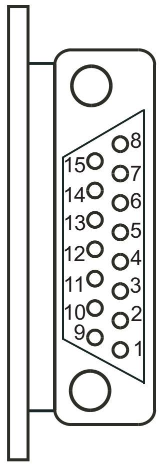
Additional Information

For additional information refer to the "CPU Data" Manual and to the Installation Manual for your CPU.

6.2.2 Connecting a Serial Cable

Pin Assignment

The table below shows the pin assignment of the 15-pin Sub-D socket on the front panel of the CPU.

<table>
<tr><th>Socket RS 422/485 (front view)</th><th>Pin</th><th>Designation</th><th>Input/
Output</th><th>Description</th></tr>
<tr><td rowspan="15">
</td><td>1</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>2</td><td>T (A) -</td><td>Output</td><td>Transmit data (Four-wire operation)</td></tr>
<tr><td>3</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>4</td><td>R (A) -
R (A)/T (A) -</td><td>Input
Input/output</td><td>Receive data (Four-wire operation)
Receive/transmit data (Two-wire operation)</td></tr>
<tr><td>5</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>6</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>7</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>8</td><td>GND</td><td>-</td><td>Functional ground (floating)</td></tr>
<tr><td>9</td><td>T (B) +</td><td>Output</td><td>Transmit data (Four-wire operation)</td></tr>
<tr><td>10</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>11</td><td>R (B) +
R (B)/T (B) +</td><td>Input
Input/output</td><td>Receive data (Four-wire operation)
Receive/transmit data (Two-wire operation)</td></tr>
<tr><td>12</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>13</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>14</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>15</td><td>-</td><td>-</td><td>-</td></tr>
</table>

When fabricating your own cables, take into account that you must always use shielded connector enclosures. Both ends of the cable shield must be connected to the connector enclosure and the shield cable over a large surface area.

CAUTION
Never connect the cable shield to ground, because this can destroy the interface.
GND (Pin 8) must always be connected on both sides to avoid destruction of the interface.

The connecting cables for various communication partners are described in Section Cables (Page 347).

6.3 Parameter configuration

6.3.1 Parameter types

Basics

In the parameter assignment screen, you can adapt the serial communication to your specific application. You can assign the parameters with two parameter types:

- **Module parameters**

  These are basic settings that are specified once and no longer changed while the process is running. These parameters are described in this section.

  – You assign the parameters via the parameter assignment screens.

  – They are stored in the system memory of the CPU.

  > **Note**
  >
  > It is not possible to change the parameters while the CPU is in RUN mode.

- **SFB parameters**

  Parameters that need to be changed during operation are located in the instance DB of the system function block (SFB). The SFB parameters are described in Section Communication Functions for ASCII/3964(R) – Basics (Page 281).

  – You assign these parameters offline in the DB Editor or online in the user program.

  – They are stored in the work memory of the CPU.

  – You can change these parameters from the user program while the CPU is in RUN mode.

6.3.2 Configuration with the Parameter Assignment Screen

Introduction

You can customize the protocol parameters with the help of the parameter assignment screens:

Those parameter assignment screens are widely self-explanatory. You can find the description of the parameters in the following chapters and in the help integrated in the parameter assignment screens.

Requirements

Prerequisite for calling the parameter assignment screen is that you have created a project in which you can save your parameters.

Procedure

1. Start the SIMATIC Manager and call HW Config in your project.
2. Double-click on the "PtP" submodule of your CPU. The "Properties" dialog box opens.
3. Edit the parameters for the "PtP" submodule and close the parameter assignment screen with "OK".
4. Save your project in HW Config with "Station > Save and Compile".
5. Download the parameter data to your CPU when it is in STOP mode with "PLC > Download to Module...". The data are now stored in the CPU's system data memory.
6. Start the CPU.

Online Help

You can find support in the integrated help of the parameter assignment screens when you configure parameters. You have the following option for calling the Integrated Help:

- Press the **F1** key in the respective views
- Click on the **Help button** in the various parameter assignment screens.

6.3.3 Basic parameters

Description

Parameter	Description	Value range	Default
Interrupt selection	Here you can specify whether or not a diagnostic interrupt is to be triggered.	• None • Diagnostics	None
Reaction to CPU Stop	This parameter influences the storing of received message frames in the receive buffer. The transmission process is canceled in both cases. The message frames stored up to this time are maintained in all cases. You can find information in closer detail in the following tables.	• Continue • STOP	Continue

The reaction to "CPU Stop" depends on whether an operation is carried out with or without data flow control.

Data flow control	Reaction to CPU Stop	Message frame just arriving	New message frames
None	Continue	Saving Discarded if buffer is full.	Saved until the buffer is full, then discarded.
	STOP	Discarded.	Discarded.
XON/XOFF	Continue	Saving Flow control is activated when buffer is full.	Saving Flow control is activated when buffer is full.
	STOP	No more data can be received because flow control is activated.	No more data can be received because flow control is activated.

6.3.4 Parameter Assignment Data for the ASCII Driver

Basics

In the parameter assignment screen, declare the parameters for the ASCII driver. Below you will find a detailed description of the parameters.

Note

The ASCII driver can be used for four-wire operation (RS 422) as well as for two-wire operation (RS 485).

Transmission

Parameters	Description	Value range	Default
Baud rate	Data transmission rate in bps (baud) * 38400 bps for half duplex only	• 300 • 600 • 1200 • 2400 • 4800 • 9600 • 19200 • 38400*	9600
Start bit	During transmission, a start bit is prefixed to each character to be sent.	1 (fixed value)	1
Data bits	Number of bits onto which a character is mapped.	• 7 • 8	8
Stop bits	During transmission, the stop bits are appended to every character to be sent; this signals the end of a character.	• 1 • 2	1
Parity	A sequence of information bits can be extended to include another bit, the parity bit. The addition of its value (0 or 1) brings the value of all the bits up to a defined status. Thus the data integrity is enhanced. If "none" is specified for parity, no parity bit is sent. It is not possible to specify "none" if 7 data bits are set.	• None • Odd • Even	Even

Parameters	Description	Value range	Default
Data flow control	Defines the method to be used for data flow control. Flow control is only possible in "Full Duplex (RS 422) four-wire PtP operation". You can avoid the loss of data transmitted by devices operating at different speeds by switching on software data flow control via XON/XOFF.	• None • XON/XOFF	None
XON character	Code for XON character The CPU transmits the XON character once it is set to the operating mode with flow control. The CPU transmits the XON character after the message frame has been fetched and after the receive buffer is ready to receive again.	• With 7 data bits: 0 to 7FH (Hex) • With 8 data bits: 0 to FFH (Hex)	11H = DC1
XOFF character	Code for XOFF character When the declared number of message frames, or 50 characters, are received before the receive buffer overflows (size of the receive buffer: 2048 bytes), the CPU transmits the XOFF character. If the communication partner nonetheless continues to transmit data, an error message is generated when the receive buffer overflows. Data received in the last message frame are discarded.	• With 7 data bits: 0 to 7FH (Hex) • With 8 data bits: 0 to FFH (Hex)	13H = DC3
Wait for XON after XOFF	The time the CPU has to wait for the XON character when transmitting. The CPU interrupts data transfer when it receives the XOFF character. If no XON is received within a specific configured time, the send operation is canceled and a corresponding error message is generated (0708_H) at the STATUS output of the SFBs.	20 to 65,530 ms in 10 ms increments	20000 ms

End-of-text character

<table>
<tr><th>Parameters</th><th>Description</th><th colspan="2">Value range</th><th>Default</th></tr>
<tr><td>End-of-message recognition for received frames</td><td>Defines which criterion signals the end of message frames.
• Upon expiration of character delay time:
The message frame has neither a fixed length nor a defined endoftext character; the end of the message is defined by a pause on the line (expiration of character delay time).
• Receiving a fixed number of characters:
The length of the received message frames is always identical.
• Receiving the end-of-text character(s):
The end of the message frame is marked by one or two defined endof-text characters.</td><td colspan="2">• on expiration of character delay time
• on receipt of a fixed number of characters
• on receipt of endoftext character</td><td>upon expiration of character delay time</td></tr>
<tr><td>Character delay</td><td>The character delay time defines the maximum permitted time between 2 consecutively received characters.</td><td colspan="2">1 to 65,535 ms
The shortest character delay time depends on the baud rate</td><td>4 ms</td></tr>
<tr><td>Monitoring time for missing end code</td><td>The character delay time is used as the monitoring time for missing end delimiter. This applies to the following settings for the end delimiter
• On receipt of a fixed character length
• Upon receipt of the end-of-text character(s)</td><td>Baud
300
600
1200
2400
4800
9600
19200
38400</td><td>Character delay time [ms]
130
65
32
16
8
4
2
1</td><td></td></tr>
<tr><td>Send pause between message frames for the length of monitoring time</td><td>The end criterion "After receiving a fixed number of characters" maintains a pause in transmission equal to the length of the monitoring time (for missing end delimiter) between two message frames to allow the partner to synchronize itself (recognition of the received message frame).</td><td colspan="2">• Yes
• No</td><td>Yes</td></tr>
<tr><td>Message frame length on receiving</td><td>The byte length of the message frame is specified for the end criterion "After receiving a fixed number of characters".</td><td colspan="2">1 to 1024 [bytes]</td><td>1024</td></tr>
</table>

Parameters	Description	Value range	Default
End-of-text character	One or two end-of-text characters can be used. As an option, one or two additional characters are received following the end-of-text character. You can for example use these characters to include a block check character (BCC) in transmission. The calculation at the transmitter and the evaluation of the block check character at the receiver must be done in the user program itself.	• 1 End-of-text character • 1 end-of-text character with 1 BCCs • 1 end-of-text character with 2 BCCs • 1. and 2nd end-of-text character • 1. and 2nd end-of-text character with 1 BCC • 1. and 2nd end-of-text character with 2 BCCs	1 End-of-text character
Endoftext character 1	First end code.	• With 7 data bits: 0 to 7FH (Hex) • With 8 data bits: 0 to FFH (Hex)	03H = ETX
Endoftext character 2	Second end code, if selected	• With 7 data bits: 0 to 7FH (Hex) • With 8 data bits: 0 to FFH (Hex)	0

Parameters	Description	Value range	Default
Transmitting with end-of-text character	You can include an end-of-text character in the transmission with the end criterion "On receiving the end-of-text character(s)". • Transmission including the end-of-text character: The end code must be included in the data to be sent. Data are only transferred up to the end delimiter, even if a greater data length is specified in the SFB. • Transmission up to the length specified in the block parameters: Data are transferred up to the length declared in the SFB parameters. The last character must be the end-of-text character. • Transmission up to the length specified in the block and automatic appending the end-of-text character: Data are transferred up to the length declared in the SFB parameters. The end-of-text character(s) is/are appended automatically; that is, the end delimiters must not be included in the data to be transferred. Depending on the number of end delimiters, 1 or 2 characters more than specified in the SFB (maximum 1024 bytes) are transferred to the partner.	• Transmission including the end-of-text character • Transmission up to the length specified in the block parameters • Transmission up to the length specified in the block parameters and automatically appending the end-of-text character	Transmission including the end-of-text character

Data Reception

Parameter	Description	Value range	Default
Clear receive buffer during startup	The receive buffer is cleared at power on or transition of the CPU from STOP to RUN.	• Yes • No	No
Prevent overwriting	You can use this parameter to prevent overwriting of data in a full receive buffer.	• Yes • No	Yes
Use entire buffer	You can use the entire receive buffer or specify the number of received frames you want to buffer. If you use the entire buffer of 2,048 bytes, the number of buffered received frames depends only on the length of the frames.	• Yes • No	Yes
Maximum number of buffered received frames	With the setting "Do not use entire buffer", you can specify the number of received frames you want to buffer in the receive buffer. If you assign "1", deactivate the "Prevent overwriting" parameter **and** read out received data periodically in the user program, a current message frame is always passed to the destination data block.	1 to 10	10

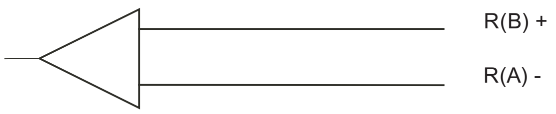
Signal Assignment for the X27 (RS 422/485) Interface

Parameters	Description	Value range	Default
Operating mode	Specifies whether the X27 (RS 422/485) is to be operated in Full Duplex (RS 422) or Half Duplex (RS 485) mode. • Full-duplex (RS 422) four-wire PtP operating mode Operating mode for four-wire PtP communication • Full-duplex (RS 422) four-wire operation, Multipoint Master Operating mode for connections capable of multipoint four-wire operation if the CPU is master. • Half Duplex (RS 485) two-wire operation PtP communication or connections capable of multipoint two-wire operation. The CPU can be master or slave.	• Full duplex (RS 422) four-wire PtP communication • Full Duplex (RS 422) four-wire operation, Multipoint Master • Half-duplex (RS 485) two-wire operation	Full duplex (RS 422) four-wire PtP communication

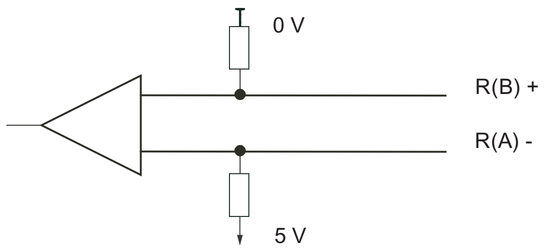
Parameters	Description	Value range	Default
Receive line default	• None: This setting only makes sense with bus capable special drivers. • Signal R(A) 5 V/Signal R(B) 0 V: Break detection is possible in this default state. (Not configurable with Full Duplex (RS422) four-wire Multipoint Master mode and Half Duplex (RS485) two-wire operation) • Signal R(A) 0 V/signal R(B) 5 V: This default state corresponds with the idle state (no transmitter active). Break detection is not possible in this default state.	• None • Signal R(A) 5 V/ signal R(B) 0 V (break detection) Signal R(A) 0 V/ signal R(B) 5 V	Depends on the set operating mode

The following view shows the wiring diagram of the receiver on an X27 (RS 422/485) interface:

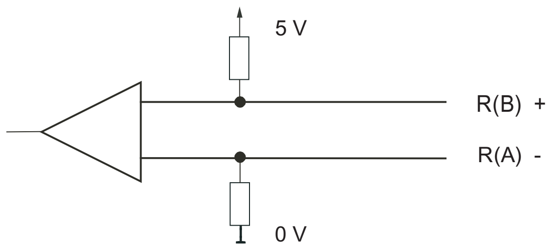
None



R(A) 5V / R(B) 0V



R(A) 0V / R(B) 5V



Topology for Using the CPU

In RS422 or RS485 operating mode the CPU can be used in various topologies.

Distinctions are made between connections with:

- two nodes (**PtP**) and
- multiple nodes (**multipoint**).

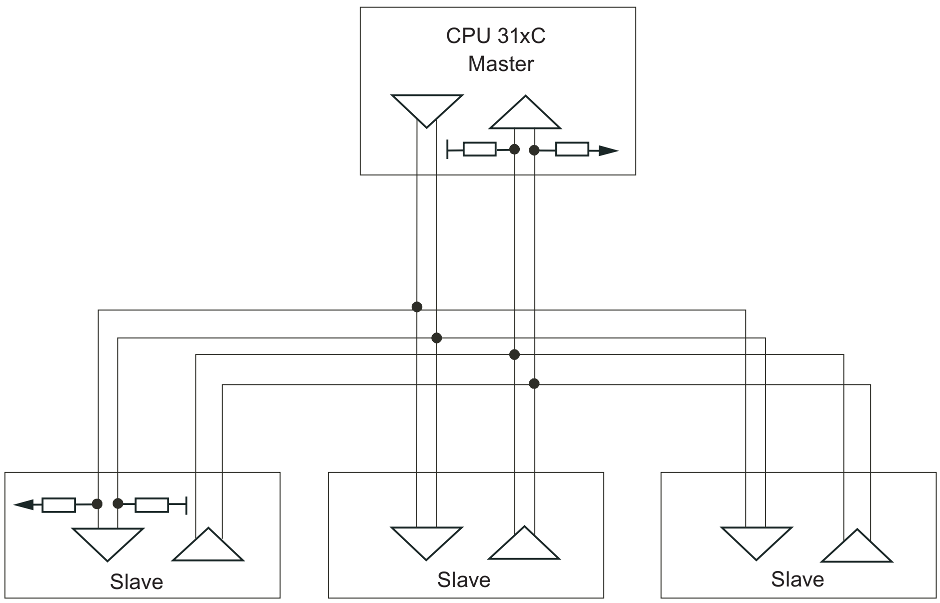
Here, it can be used as

- **Master** or
- **Slave** (only RS485 operation).

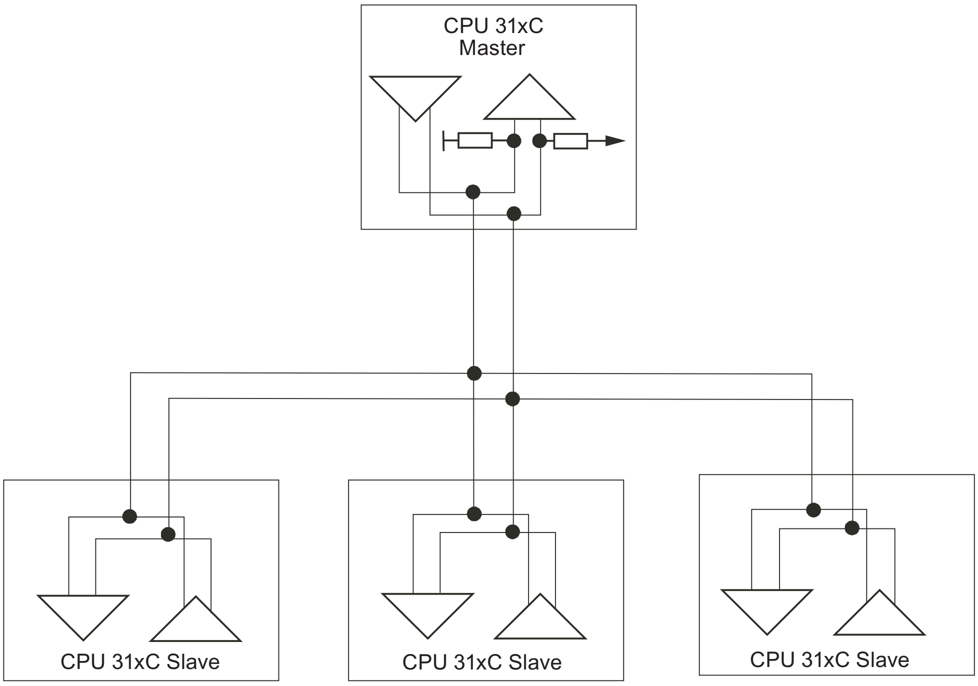
With a **master/slave topology**, there must be an appropriate message frame in the user program. Example: The master transmits a message frame containing an address information to all slaves. All the slaves listen in and compare the address with their own. If the address is the same, the addressed slave sends its answer.

The transmitters of all slaves must be able of switching to high-impedance.

- **With a master/slave topology in RS422 operation**
  - the **CPU can only be used in master** mode,
  - the master's transmitter is interconnected with the receivers of all the slaves.
  - the slaves' transmitter is interconnected with the master's receiver.
  - only the receivers of the master and slave are assigned a default setting. All the other slaves operate without default settings.



- In case of a topology with RS485 operation,
  - the send/receive cable pairs of all nodes are interconnected,
  - only the receiver of a node has a default setting. All the other modules operate without default settings.



The configuration required for the different topologies is carried out in the "Interface" dialog of the parameter assignment screen.

Note

If you edit the ASCII driver with or RS422 multipoint mode or with RS485 operation, you must always ensure in your user program that only one node is transmitting data. If data are transmitted simultaneously, the message frame is corrupted.

6.3.5 Parameter Assignment Data for the 3964(R) Procedure

Basics

In the parameter assignment screen, specify the parameters for the 3964(R) protocol. Below you will find a detailed description of the parameters.

Note

The 3964(R) protocol can only be used in four-wire operating mode (RS 422).

Transmission

Parameters	Description	Value range	Default
Baud rate	Data transmission rate in bps (baud)	• 300 • 600 • 1200 • 2400 • 4800 • 9600 • 19200 • 38400	9600
Start bit	During transmission, a start bit is prefixed to each character to be sent.	1 (fixed value)	1
Data bits	Number of bits onto which a character is mapped.	• 7 • 8	8
Stop bits	During transmission, the stop bits are appended to every character to be sent; this signals the end of a character.	• 1 • 2	1
Parity	A sequence of information bits can be extended to include another bit, the parity bit. The addition of its value (0 or 1) brings the value of all the bits up to a defined state. This increases the data reliability. If "none" is specified for parity, no parity bit is sent. **It is not possible to specify "none" if 7 data bits are set.**	• None • Odd • Even	Even
Priority	A partner has high priority if its send request takes precedence over the send request of the other partner. A partner has low priority if its send request must wait until the send request of the other partner has been dealt with. With the 3964(R) protocol you must assign different priorities for the two communication partners, that is, one partner is assigned high priority, the other low priority.	• Low • High	High

Parameter	Description	Default
Message frame parameter 3964(R) with standard values and block check	The protocol parameters are set to default values. The CPU stops receiving when it detects the character string DLE ETX BCC. It compares the received block check character BCC with the internally calculated longitudinal parity. If the block check character is correct and no other receive error has occurred, it transmits the DLE character (on error it transmits the NAK character to the communication partner).	3964(R) with standard values and block check: Character delay = 220 ms Acknowledgment delay = 2000 ms Connection retries = 6 Transmission retries = 6
Assignable message frame parameter 3964(R) with block check	The protocol parameters are freely programmable. The CPU stops receiving when it detects the character string DLE ETX BCC. It compares the received block check character BCC with the internally calculated longitudinal parity. If the block check character is correct and no other receive error has occurred, it transmits the DLE character (on error it transmits the NAK character to the communication partner).	
Message frame parameter 3964 with standard values and without block check	The protocol parameters are set to default values. When the CPU detects a DLE ETX string, it stops receiving and sends a DLE to the communication partner if the block was received without errors (or NAK with error).	
Assignable message frame parameter 3964 without block check	The protocol parameters are freely programmable. When the CPU detects a DLE ETX string, it stops receiving and sends a DLE to the communication partner if the block was received without errors (or NAK with error).	

Parameter	Description	Value range	Default
Character delay	The character delay defines the permitted maximum time interval between two received characters of a message frame.	20 ms to 65530 ms in 10 ms increments The shortest character delay depends on the baud rate: 300 bps: 60 ms 600 bps: 40 ms 1200 bps: 30 ms 2400 to 38400 bps: 20 ms	220 ms
Acknowledgment delay	The acknowledgment delay determines the maximum permitted time delay for the partner to send an acknowledgment when the connection is established (time between STX and acknowledgment DLE of the partner) or closed (time between DLE ETX (BCC) and the DLE acknowledgment of the partner).	20 ms to 65530 ms in 10 ms increments The shortest acknowledgment delay depends on the baud rate: 300 bps: 60 ms 600 bps: 40 ms 1200 bps: 30 ms 2400 to 38400 bps: 20 ms	2000 ms (550 ms with 3964 without block check)
Connection retries	The parameter defines the maximum number of attempts of the CPU to establish a connection.	1 to 255	6
Transmission retries	The parameter defines the maximum number of attempts to transfer a message frame (including the first one) in the event of errors.	1 to 255	6

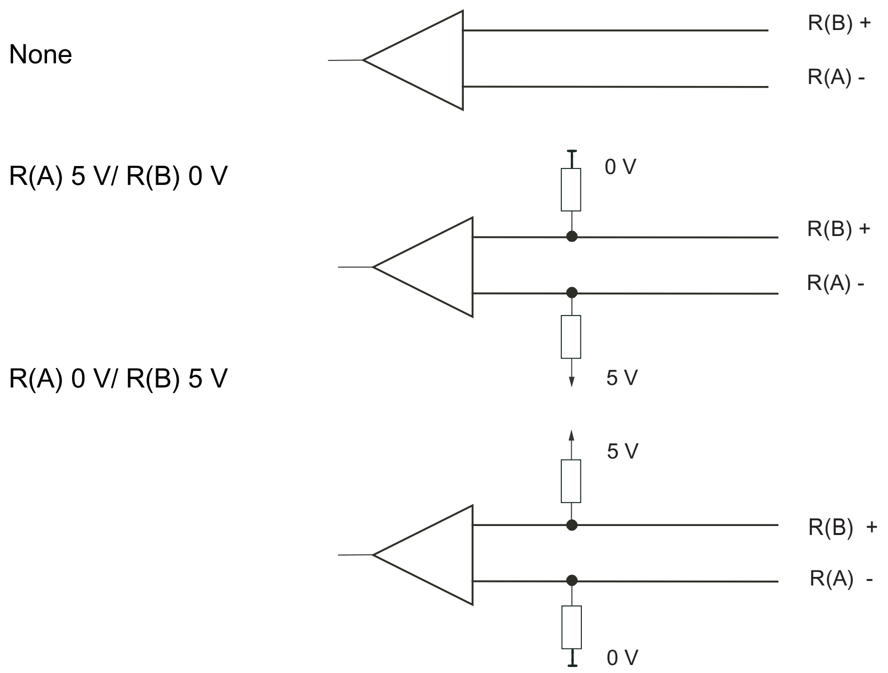
Data reception

Parameters	Description	Value range	Default
Clearing the receive buffer at startup	The receive buffer is cleared at power on or transition of the CPU from STOP to RUN.	• Yes • No	No
Prevent overwriting	You can use this parameter to prevent overwriting of data in a full receive buffer.	• Yes • No	Yes
Utilize the whole buffer	You can use the whole receive buffer or specify the number of received message frames you want to buffer. If you use the entire buffer of 2,048 bytes, the number of buffered received frames depends only on the length of the frames.	• Yes • No	Yes
Maximum number of buffered received frames	You can specify the number of received frames you want to buffer in the receive buffer with the setting "Do not use entire buffer". If you assign "1", deactivate the "Prevent overwriting" parameter **and** read out received data periodically in the user program, a current message frame will always be transferred to the destination data block.	1 to 10	10

Signal assignment for the X27 (RS 422/485) interface

Parameter	Description	Value range	Default
Receive line initial state	None: This setting only makes sense for drivers with bus capability.	None	R(A) 5 V/ R(B) 0 V
	R(A) 5 V/ R(B) 0 V: Break detection is possible in this default state.	R(A) 5V/R(B) 0 V	
	R(A) 0 V/ R(B) 5 V: Break detection is not possible in this default state.	R(A) 0V/R(B) 5 V	

The following figure shows the wiring of the receiver to the X27 (RS 422) interface:



6.3.6 Parameter Assignment Data for RK 512 Communication

Basics

The parameters are identical to those of the 3964(R) protocol, because 3964(R) represents a partial quantity of the RK 512 communication.

Exception:

- The number of bits per character with RK 512 communication is set fixed to 8.
- There is no receive buffer (no parameters for received data).

You must specify the parameters for the data target or source in the used system function blocks (SFBs).

6.4 Implementing the Connection in the User Program

Procedure

You control the serial connection with your user program. To do this, call the system function blocks (SFBs). The SFBs are found in the "Standard library" under "System Function Blocks".

The following sections help you to design a user program for your application.

Calling the SFB

Call the SFB with a corresponding instance DB.

Example: CALL SFB 60, DB20

Instance DB

All parameters required for the SFB are stored in the instance DB.

Note

In your user program, you must always call each SFB type (SEND, FETCH, RCV, ...) with the same instance DB, because the states required for the internal SFB processes are stored in this instance DB.

Access to the data in the instance DB is not permitted.

Program Structure

The SFB is processed asynchronously. To process the SFB completely, it must be called as frequently as necessary until it is ended with or without an error.

Note

You must not call an SFB you have configured in your program in another program section with a different priority class, because the SFB must not interrupt itself.

Example: It is not allowed to call an SFB both in OB1 and in the interrupt OB.

Classification of the SFB Parameters

The parameters of the SFBs can be split into four classes according to their function:

- **Control parameters** are used to activate a block.
- **Send parameters** point to the data areas that are to be transmitted to the remote partner.
- **Receive parameters** point to the data areas in which the data received from the remote partner are entered.
- **Status parameters** are used to monitor whether the block has completed its task without errors, or to analyze the errors that occurred. Status parameters are set for the duration of one call only.

6.5 Communication Functions

6.5.1 Communication Functions for ASCII/3964(R)

6.5.1.1 Communication Functions for ASCII/3964(R) – Basics

Overview

The following functions are provided for the ASCII and 3964 protocols.

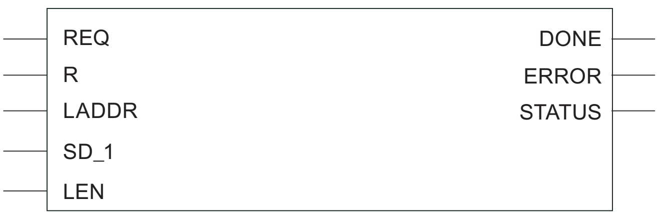
Block		Description
SFB 60	SEND_PTP	Send the whole or partial area of a data block to a communication partner.
SFB 61	RCV_PTP	Receive data from a communication partner and save it in a DB.
SFB 62	RES_RCVB	Reset the receive buffer of the CPU

6.5.1.2 Sending Data with SFB 60 "SEND_PTP"

Basics

With this SFB you transmit a data block from a DB.

“SEND_PTP”



The send process is activated after the block call and at a positive edge on the control input **REQ.** The area of the data to be transmitted is specified in **SD_1** (DB number and start address), and the data block length in **LEN**.

You must call the SFB with **R**(Reset) = FALSE to enable it to process the job. At a positive edge on control input R, the current transmission is canceled and the SFB is reset to the initial state. A canceled request is concluded with an error message (STATUS output).

You enter the submodule I/O address you specified in "HW Config" with **LADDR**.

Either **NDR** is set to TRUE if the request was closed without error, or **ERROR** is set to TRUE if the request was terminated with error.

If the request was run through with DONE = TRUE, it means that:

- When using the ASCII driver: Data were transmitted to the communication partner. However, it is not ensured that the data were received by the communication partner.
- When using the 3964(R) protocol: Data was transmitted to the communication partner, a positive acknowledgment was returned. However, it is not ensured that the data were also passed to the partner CPU.

STATUS displays the corresponding event number if an error or warning has occurred (see Section Error Messages (Page 351)). DONE or ERROR/STATUS is also output in the event of a RESET of the SFB (R = TRUE). The binary result BIE is reset if an error has occurred. If the block ends without error, the status of the binary result is TRUE.

Note

The SFB has no parameter check. If it is not programmed correctly the CPU might switch to STOP mode.

Parameter	Declaration	Data type	Description	Value range	Default
REQ	IN	BOOL	Control parameter "Request": Activates data exchange at a positive edge	TRUE/FALSE	FALSE
R	IN	BOOL	Control parameter "Reset": Cancels the request. Sending is blocked.	TRUE/FALSE	FALSE
LADDR	IN	WORD	Submodule I/O address you specified in "HW Config".	CPU-specific	3FF hex
DONE	OUT	BOOL	Status parameter (Set only for the duration of one call): • FALSE: The request has not yet been started or is still being executed. • TRUE: The request has been completed without error.	TRUE/FALSE	FALSE
ERROR	OUT	BOOL	Status parameter (set only for the duration of one call): Request completed with errors	TRUE/FALSE	FALSE
STATUS	OUT	WORD	Status parameter (Set only for the duration of one call. In order to display STATUS, you should therefore copy STATUS to a free data area): STATUS has the following significance, depending on the ERROR bit: • ERROR = FALSE: The STATUS value is: – 0000H: Neither warning nor error – <> 0000H: Warning, STATUS supplies detailed information • ERROR = TRUE: Error pending. STATUS supplies detailed information about the type of error (see Error Messages (Page 351)).	0 to FFFF hex	0

Parameter	Declaration	Data type	Description	Value range	Default
SD_1	IN_OUT	ANY	Send parameters: Here you specify: • The number of the DB from which the data are sent. • Data byte number as of which the data are to be sent. e.g.: DB 10 from byte 2 → DB10.DBB2	CPU-specific*	0
LEN	IN_OUT	INT	Here you specify the length of the data block to be transmitted, in bytes. (Length is specified here indirectly.)	1 to 1024	1
* Offset for parameter SD_1 may not be greater than 8190 for CPU 313C PtP or 314C PtP. An error message will be returned if this limit is violated.					

Data consistency

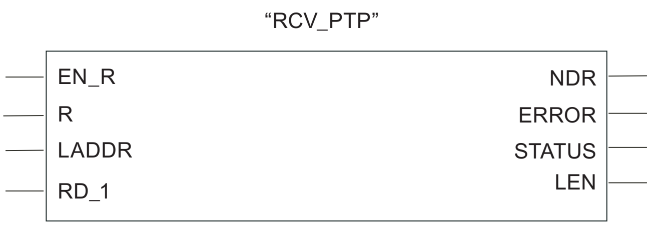
Data consistency is limited to 206 bytes. You must note the following points for the transmission of consistent data with a length of more than 206 bytes:

Do not write to the currently used section of the send area SD_1 until the transmission process is complete. This is the case when the status parameter DONE = TRUE.

6.5.1.3 Receiving Data with SFB 61 "RCV_PTP"

Basics

You receive data with this SFB and save them to a data block.



The block is ready to receive data after it is called with the value TRUE on control input EN_R. You can cancel a current transmission via signal status FALSE on parameter EN_R. A canceled request is concluded with an error message (STATUS output). Receiving is locked as long as the signal status at parameter EN_R is FALSE.

The receive area is specified in RD_1 (DB number and start address),and the data block length in LEN.

You must call the SFB with R(Reset) = FALSE to enable it to process the request. At a positive edge on control input R, a current transmission is canceled and the SFB is reset to the initial state. A canceled request is concluded with an error message (STATUS output).

You enter the submodule I/O address you have specified in "HW Config" in LADDR.

Either NDR is set to TRUE if the request was closed without error, or ERROR is set to TRUE if the request was terminated with error.

STATUS displays the corresponding event number if an error or warning has occurred (see Section Error Messages (Page 351)).
NDR or ERROR/STATUS are also output in the event of a RESET of SFB (R = TRUE) (Parameter LEN = 16#00).
The binary result BIE is reset if an error has occurred. If the block ends without error, the status of the binary result is TRUE.

Note

The SFB has no parameter check. If it is not programmed correctly the CPU might switch to STOP mode.

Parameter	Declaration	Data type	Description	Value range	Default
EN_R	IN	BOOL	Control parameter "Enable to receive": Receive enable	TRUE/FALSE	FALSE
R	IN	BOOL	Control parameter "Reset": Cancels the request	TRUE/FALSE	FALSE
LADDR	IN	WORD	Submodule I/O address you specified in "HW Config".	CPU-specific	3FF hex
NDR	OUT	BOOL	Status parameter "New data ready": Request completed without errors, data accepted • FALSE: The request has not yet been started or is still running. • TRUE: The request has been completed successfully.	TRUE/FALSE	FALSE
ERROR	OUT	BOOL	Status parameter (Set only for the duration of one call): Request completed with errors	TRUE/FALSE	FALSE
STATUS	OUT	WORD	Status parameter (Set only for the duration of one call. In order to display STATUS, you should therefore copy STATUS to a free data area): STATUS has the following significance, depending on the ERROR bit: • ERROR = FALSE: The STATUS value is: – 0000H: Neither warning nor error – <> 0000H: Warning, STATUS supplies detailed information • ERROR = TRUE: Error pending. STATUS supplies detailed information about the type of error (for error numbers, see Section Error Messages (Page 351)).	0 to FFFF hex	0

Parameter	Declaration	Data type	Description	Value range	Default
RD_1	IN_OUT	ANY	Receive parameters: Here you specify: • Number of the DB in which the received data are to be stored. • The data byte number as of which received data are to be stored. e.g.: DB 20 from byte 5 → DB20.DBB5	CPU-specific*	0
LEN	IN_OUT	INT	Output of the data length (number of bytes)	0 to 1024	0
* Offset for parameter RD_1 may not be greater than 8190 for CPU 313C PtP or 314C PtP. An error message will be returned if this limit is violated.					

Data consistency

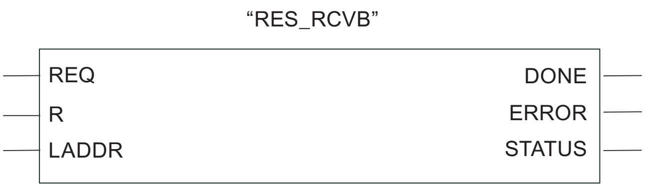
Data consistency is limited to 206 bytes. You must note the following points for the transmission of consistent data with a length of more than 206 bytes:

Do not access the receive DB unless all data have been received (NDR = TRUE). Then, lock the receive DB until (EN_R = FALSE) you have processed the data.

6.5.1.4 Clearing the Receive Buffer with SFB 62 "RES_RCVB"

Basics

With the SFB, you clear the entire receive buffer of the CPU. All stored message frames are deleted. A message frame coming in when "RES_RCVB" is being called is saved.



The request is activated after the block call and at a positive edge on control input **REQ**. The request can run across several calls (program cycles).

You must call the SFB with **R**(Reset) = FALSE to enable it to process the request. At a positive edge on control input R, the clearing process is canceled and the SFB is set to the initial state. A canceled request is concluded with an error message (STATUS output).

You enter the submodule I/O address you have specified in "HW Config" in **LADDR**.

Either **NDR** is set to TRUE if the request was closed without error, or **ERROR** is set to TRUE if the request was terminated with error.

STATUS displays the corresponding event number if an error or warning has occurred (see Section Error Messages (Page 351).

DONE or ERROR/STATUS is also output in the event of a RESET of the SFB (R = TRUE).

The binary result BIE is reset if an error has occurred. If the block ends without error, the status of the binary result is TRUE.

Note

The SFB has no parameter check. If it is not programmed correctly the CPU might switch to STOP mode.

Parameter	Declaration	Data type	Description	Value range	Default
REQ	IN	BOOL	Control parameter "Request": Activates the request at a positive edge	TRUE/FALSE	FALSE
R	IN	BOOL	Control parameter "Reset": Cancels the request.	TRUE/FALSE	FALSE
LADDR	IN	WORD	Submodule I/O address you specified in "HW Config".	CPU-specific	3FF hex
DONE	OUT	BOOL	Status parameter (set only for the duration of one call): • FALSE: The job has not yet been started or is still being executed. • TRUE: The job has been completed without errors.	TRUE/FALSE	FALSE
ERROR	OUT	BOOL	Status parameter (set only for the duration of one call): Job completed with error	TRUE/FALSE	FALSE
STATUS	OUT	WORD	Status parameter (set only for the duration of one call. In order to display STATUS, you should therefore copy STATUS to a free data area): STATUS has the following significance, depending on the ERROR bit: • ERROR = FALSE: The STATUS value is: – 0000H: Neither warning nor error – <> 0000H: Warning, STATUS supplies detailed information • ERROR = TRUE: Error pending. STATUS supplies detailed information about the type of error (for error numbers, see Section Error Messages (Page 351)).	0 to FFFF hex	0

6.5.2 RK 512 Communication Functions

6.5.2.1 Communication Functions for the RK 512 Computer Connection – Basics

Overview

The following functions are provided for the RK 512 protocol:

Block		Description
SFB 63	SEND_RK	Send the whole or part area of a data block to a communication partner.
SFB 64	FETCH_RK	Send the whole or partial area of a data block to a communication partner.
SFB 65	SERVE_RK	• Receive data from a communication partner and save it in a DB. • Provide data for a communication partner.

Parallel Processing of Requests

SEND/FETCH jobs must not be activated simultaneously in the user program i.e. a FETCH job cannot be started if a SEND job is not yet closed.

SYNC_DB

For the initialization at setup and for synchronizing operations between the SFBs, all SFBs you are using for RK512 communication require a common data area. You determine the DB number via the parameter SYNC_DB. The DB number must be identical for all SFBs in your user program. The DB must have a minimum length of 240 bytes.

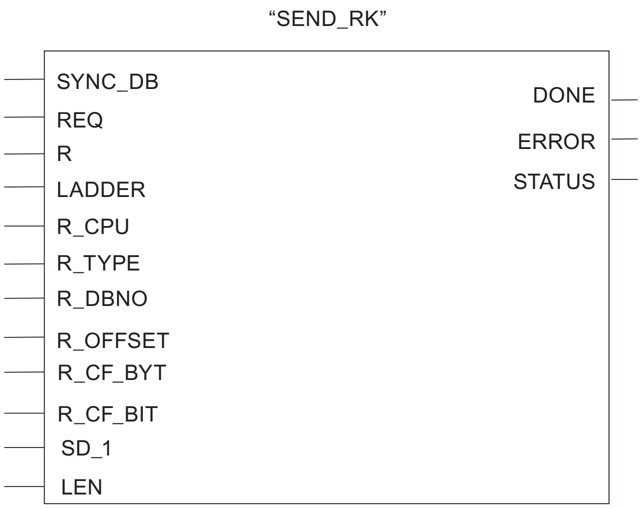
Interprocessor communication flag

The functionality of interprocessor communication flags known in SIMATIC S5 is supported by SFB "SERVE_RK" (SFB 65), in order to coordinate data processing in the CPU and asynchronous overwriting when receiving or providing data.

6.5.2.2 Sending data with SFB 63 " SEND_RK"

Basics

With this SFB you transmit a data block from a DB.



The send process is activated after the block call and at a positive edge on the control input REQ.

The area of the data to be transmitted is specified in SD_1 (DB number and start address), and the data block length in LEN.

In the SFB you also specify the receive area on the partner. This information is entered in the message frame header by the CPU and transferred to the partner.

The destination is specified by the CPU number R_CPU (only relevant for multiprocessor communication), the data type R_TYPE (data blocks (DB) and expanded data blocks (DX)), the data block number R_DBNO, and the offset R_OFFSET.

In R_CF_BYT and R_CF_BIT you specify the interprocessor communication flag byte and bit on the partner CPU.

In the SYNC_DB parameter, you specify the DB in which you want to store data common to all SFBs you are using for startup initialization and synchronization routines. The DB number must be identical for all SFBs in your user program.

You must call the SFB with R(Reset) = FALSE to enable it to process the job. At a positive edge on control input R, the current transmission is canceled and the SFB is set to the initial state. A canceled request is concluded with an error message (STATUS output).

You enter the submodule I/O address you have specified in "HW Config" in LADDR.

Either NDR is set to TRUE if the request was closed without error, or ERROR is set to TRUE if the request was terminated with error.

If the request was run through with DONE = TRUE, this means that the data were transmitted to the communication partner and positively acknowledged, and the data were passed to the partner CPU.

STATUS displays the corresponding event number if an error or warning has occurred (see Section Error Messages (Page 351)).

DONE or ERROR/STATUS is also output in the event of a RESET of the SFB (R = TRUE).

The binary result BIE is reset if an error has occurred. If the block ends without error, the status of the binary result is TRUE.

Note

The SFB has no parameter check. If it is not programmed correctly the CPU might switch to STOP mode.

Parameters	Declaration	Data type	Description	Value range	Default
SYNC_DB	IN	INT	Number of the DB in which the common data for the synchronization of the RK SFBs is stored (minimum length is 240 bytes).	CPU-specific, zero is not allowed.	0
REQ	IN	BOOL	Control parameter "Request": Activates the data exchange at a positive edge	TRUE/FALSE	FALSE
R	IN	BOOL	Control parameter "Reset": Cancels the request. Sending is blocked.	TRUE/FALSE	FALSE
LADDR	IN	WORD	Submodule I/O address you specified in "HW Config".	CPU-specific	3FF hex
R_CPU	IN	INT	Partner CPU number (only for multiprocessor mode)	0 to 4	1
R_TYPE	IN	CHAR	Address type on the partner CPU (capital letters only): • 'D' = Data block • 'X' = Expanded date block	'D', 'X'	'D'
R_DBNO	IN	INT	Data block number on partner CPU	0 to 255	0
R_OFFSET	IN	INT	Data byte number on partner CPU	0 to 510 (even values only)	0
R_CF_BYT	IN	INT	Interprocessor communication flag byte on partner CPU (255 means: no interprocessor communication flag)	0 to 255	255
R_CF_BIT	IN	INT	Interprocessor communication flag bit on partner CPU	0 to 7	0
DONE	OUT	BOOL	Status parameter (set only for the duration of one call): • FALSE: The job has not yet been started or is still being executed. • TRUE: The job has been completed without errors.	TRUE/FALSE	FALSE

Parameters	Declaration	Data type	Description	Value range	Default
ERROR	OUT	BOOL	Status parameter (set only for the duration of one call): Job completed with error	TRUE/FALSE	FALSE
STATUS	OUT	WORD	Status parameter (Set only for the duration of one call. In order to display STATUS, you should therefore copy STATUS to a free data area): STATUS has the following significance, depending on the ERROR bit: • ERROR = FALSE: The STATUS value is: – 0000H: Neither warning nor error – <> 0000H: Warning, STATUS supplies detailed information • ERROR = TRUE: Error pending. STATUS supplies detailed information about the type of error (for error numbers, see Section Error Messages (Page 351)).	0 to FFFF hex	0
SD_1	IN_OUT	ANY	Send parameters: Here you specify: • The number of the DB from which the data are sent. • The data byte number from which the data is to be sent. e.g.: DB 10 from byte 2 → DB10.DBB2	CPU-specific*	0
LEN	IN_OUT	INT	Here you specify the length of the data block to be transmitted, in bytes. (Length is specified here indirectly.)	1 to 1024	1
* Offset for parameter SD_1 may not be greater than 8190 for CPU 313C PtP or 314C PtP. An error message will be returned if this limit is violated.					

Data consistency

Data consistency is limited to 128 bytes. You must note the following points for the transmission of consistent data with a length of more than 128 bytes:

Do not write to the currently used section of the send area SD_1 until the transmission process is complete. This is the case when the status parameter DONE = TRUE.

Special Features for Sending Data

Note the following special features for "Sending Data":

- RK 512 allows only the transmission of an even data length. If you specify an odd data length (LEN), an additional "0" filler byte is appended to the transmitted data.
- RK 512 only allows you to specify an even offset. If you specify an odd offset, the data are stored on the partner as of the next smaller even offset.

Example: Offset is 7, data are written as of byte 6.

Information in the Message Frame Header

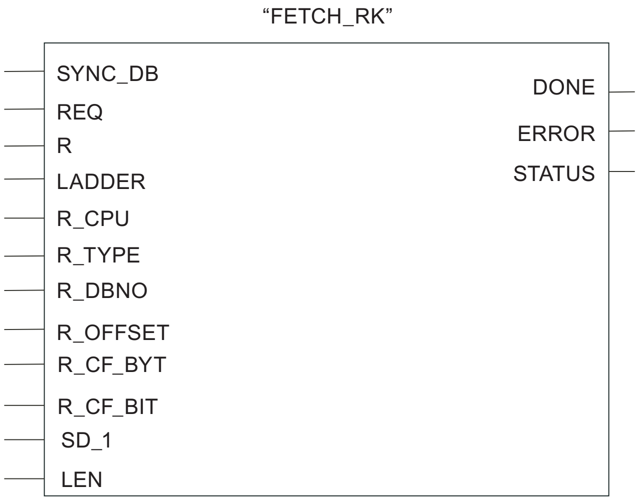
The following table shows the information in the RK 512 message frame header.

Source on your S7 automation system (local CPU)	To the target, partner CPU	Message frame header		
		Bytes 3/4: Command type	Bytes 5/6: Z-DBNR/Z offset	Bytes 7/8: Amount in
Data block	Data block	AD	DB/DW	Words
Data block	Expanded data block	AD	DB/DW	Words
Abbreviations: Z-DBNR: Destination data block number Z-Offset: Destination start address DW: Offset in words				

6.5.2.3 Fetching Data with SFB 64 " FETCH_RK"

Basics

With this SFB you fetch a data block from a partner and save the data to a DB.



The send process is activated after the block call and at a positive edge on the control input REQ.

The memory area in which fetched data are stored is specified in RD_1 (DB-number and start address), and the data block length in LEN.

In the SFB you also specify the partner area from which data are to be fetched. This information is entered in the RK 512 message frame header by the CPU and transferred to the partner (see Section Data Transmission with the RK 512 Computer Connection – Basics (Page 330)).

The area on the partner from which the first byte is to be fetched is specified by the CPU number R_CPU (only relevant for multiprocessor communication), the data type R_TYPE (data blocks, expanded data blocks, memory bits, inputs, outputs, counters, and timers), the data block number R_DBNO (only relevant for data blocks and expanded data blocks), and the offset parameter R_OFFSET.

The interprocessor communication flag byte and bit on the partner CPU is specified in R_CF_BYT and R_CF_BIT.

In the SYNC_DB parameter, you specify the DB in which you want to store data common to all SFBs you are using for startup initialization and synchronization routines. The DB number must be identical for all SFBs in your user program.

You must call the SFB with R(Reset) = FALSE to enable it to process the request. At a positive edge on control input R, the current transmission is canceled and the SFB is reset to the initial state. A canceled request is concluded with an error message (STATUS output).

You enter the submodule I/O address you specified in "HW Config" with LADDR.

DONE is set to TRUE if the job was closed without errors and ERROR to TRUE if the job was closed with errors.

STATUS displays the corresponding event number if an error or warning has occurred.

DONE or ERROR/STATUS is also output in the event of a RESET of the SFB (R = TRUE).

The binary result BIE is reset if an error has occurred. If the block ends without error, the status of the binary result is TRUE.

Note

The SFB has no parameter check. If it is not programmed correctly the CPU might switch to STOP mode.

You must program a "SERVE_RK" SFB on your CPU if data are to be fetched from there.

Parameters	Declaration	Data type	Description	Value range	Default
SYNC_DB	IN	INT	Number of the DB in which the common data for the synchronization of the RK SFBs are stored (minimum length is 240 bytes).	CPU-specific, zero is not allowed.	0
REQ	IN	BOOL	Control parameter "Request": Activates data exchange at a positive edge	TRUE/FALSE	FALSE
R	IN	BOOL	Control parameter "Reset": Cancels the request.	TRUE/FALSE	FALSE
LADDR	IN	WORD	Submodule I/O address you specified in "HW Config".	CPU-specific	3FF hex
R_CPU	IN	INT	CPU no. of the partner CPU (only for multiprocessor mode)	0 to 4	1
R_TYPE	IN	CHAR	Address type on the partner CPU (capital letters only!): • 'D' = Data block • 'X' = Expanded date block • 'M' = Memory bit • 'E' = Inputs • 'A' = Outputs • 'Z' = Counters • 'T' = Timers	'D', 'X', 'M', 'E', 'A', 'Z', 'T'	'D'
R_DBNO	IN	INT	Data block number on partner CPU	0 to 255	0
R_OFFSET	IN	INT	Data byte number on partner CPU	See the Table: "Parameters in the FB for data source (Partner CPU)"	0
R_CF_BYT	IN	INT	Interprocessor communication flag byte on partner CPU (255: means: no interprocessor communication flag)	0 to 255	255
R_CF_BIT	IN	INT	Interprocessor communication flag bit on partner CPU	0 to 7	0

Parameters	Declaration	Data type	Description	Value range	Default
DONE	OUT	BOOL	Status parameter (set only for the duration of one call): • FALSE: The job has not yet been started or is still being executed. • TRUE: The job has been completed without errors.	TRUE/FALSE	FALSE
ERROR	OUT	BOOL	Status parameter (set only for the duration of one call): Job completed with error	TRUE/FALSE	FALSE
STATUS	OUT	WORD	Status parameter (Set only for the duration of one call. In order to display STATUS, you should therefore copy STATUS to a free data area): STATUS has the following significance, depending on the ERROR bit: • ERROR = FALSE: The STATUS value is: – 0000H: Neither warning nor error – <> 0000H: Warning, STATUS supplies detailed information • ERROR = TRUE: Error pending. STATUS supplies detailed information relating to the type of error.	0 to FFFF hex	0
RD_1	IN_OUT	ANY	Receive parameters: Here you specify: • Number of the DB in which the fetched data are stored. • Data byte number as of which the fetched data are stored. e.g.: DB 10 from byte 2 → DB10.DBB2	CPU-specific*	0
LEN	IN_OUT	INT	Here you specify the byte length of the message frame to be fetched. (Length is specified here indirectly.) A length of two bytes must be specified per timer and counter.	1 to 1024	1
* Offset for parameter RD_1 may not be greater than 8190 for CPU 313C PtP or 314C PtP. An error message will be returned if this limit is violated.					

Data consistency

Data consistency is limited to 128 bytes. You must note the following points for the transmission of consistent data with a length of more than 128 bytes:

You must not write to the currently used section of the receive area RD_1 until the transmission process has ended. This is the case when the status parameter DONE = TRUE.

Special Features of (Expanded) Data Blocks

Note the following special features for "Fetching Data" from DBs and expanded DBs:

- RK 512 only allows you to fetch an even number of data. If you specify an odd number of data as length (LEN), an extra byte is always transmitted. However, the correct data length is entered in the destination DB.
- RK 512 only allows you to specify an even offset. If you specify an odd offset, the data are stored on the partner as of the next smaller even offset.

Example: Offset is 7, data are fetched as of byte 6.

Special Feature for Timers and Counters

When fetching timers and counters from the communication partner, you must remember that two bytes are fetched per timer and counter. For example, if you want to fetch 10 counters, you must declare a length of 20.

Parameters in the SFB for the Data Source (Partner CPU)

The following table lists the data types that can be transmitted.

Source on partner CPU	R_TYPE	R_DBNO	R_OFFSET** (in bytes)
Data block	'D'	0 - 255	0 - 510*
Expanded data block	'X'	0 - 255	0 - 510*
Memory bit	'M'	Irrelevant	0 - 255
Inputs	'E'	Irrelevant	0 - 255
Outputs	'A'	Irrelevant	0 - 255
Counters	'Z'	Irrelevant	0 - 255
Timers	'T'	Irrelevant	0 - 255
* Only even values make sense ** This value is specified by the partner CPU.			

Information in the Message Frame Header

The following table shows the information in the RK 512 message frame header (see also Section Data Transmission with the RK 512 Computer Connection – Basics (Page 330)).

Source on partner CPU	to target, your S7 automation system (local CPU)	Message frame header		
		Bytes 3/4: Command type	Bytes 5/6: Q-DBNR/ Q-Offset	Bytes 7/8: Amount in
Data block	Data block	ED	DB/DW	Words
Expanded data block	Data block	EX	DB/DW	Words
Memory bit	Data block	EM	Byte address	Bytes
Inputs	Data block	EE	Byte address	Bytes
Outputs	Data block	EA	Byte address	Bytes
Counters	Data block	EZ	Counter number	Words
Timers	Data block	ET	Timer number	Words
Abbreviations: Q-DBNR: Source data block number Q-Offset: Source start address				

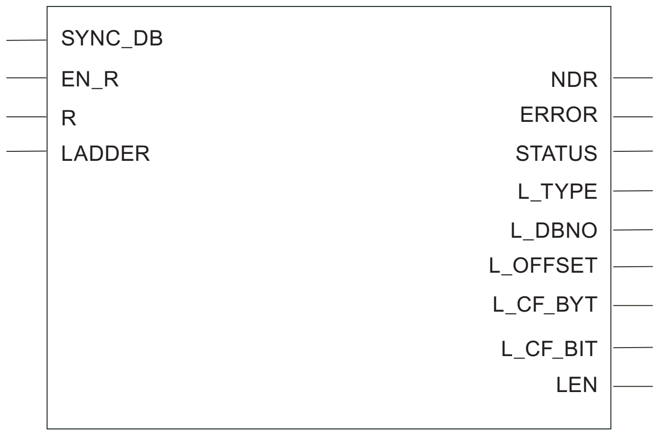
6.5.2.4 Receiving/Providing Data with SFB 65 "SERVE_RK"

Basics

Use this SFB for

- **Receiving data:** Data are stored in the data area specified by the partner in the RK 512 message frame header (see also Section Data Transmission with the RK 512 Computer Connection – Basics (Page 330)). The SFB must be called when the communication partner executes a "Send data" job (SEND job).
- **Providing data:**Data are fetched from the data area specified by the partner in the RK 512 message frame header (see also Section Data Transmission with the RK 512 Computer Connection – Basics (Page 330)). The SFB must be called when the communication partner executes a "Fetch data" job (FETCH job).

“SERVE_RK”



The block is ready after it is called with the value TRUE on control input EN_R. You can cancel a current transmission via signal status FALSE on parameter EN_R. A canceled request is concluded with an error message (STATUS output). Receiving is disabled as long as the signal status at parameter EN_R is FALSE.

In the SYNC_DB parameter, you specify the DB in which you want to store data common to all SFBs you are using for startup initialization and synchronization routines. The DB number must be identical for all SFBs in your user program.

You must call the SFB with R(Reset) = FALSE to enable it to process the job. At a positive edge on control input R, a current transmission is canceled and the SFB is reset to the initial state. A canceled request is concluded with an error message (STATUS output).

You enter the submodule I/O address you have specified in "HW Config" in LADDR.

NDR is set to TRUE if the job was closed without errors and ERROR to TRUE if the job was closed with errors.

If NDR = TRUE for one call, the block indicates where the data were stored or from where the data were fetched in the parameters L_TYPE, L_DBNO, and L_OFFSET. In addition, the parameters L_CF_BYT and L_CF_BIT and the length LEN of the corresponding request are shown for the duration of one call.

STATUS displays the corresponding event number if an error or warning has occurred (see Section Error Messages (Page 351).

NDR or ERROR/STATUS is also output in the event of a RESET of SFB (R = TRUE) (Parameter LEN = 16#00).

The binary result BIE is reset if an error has occurred. If the block ends without error, the status of the binary result is TRUE.

Note

The SFB has no parameter check. If it is not programmed correctly the CPU might switch to STOP mode.

Parameter	Declaration	Data type	Description	Value range	Default
SYNC_DB	IN	INT	Number of the DB in which the common data for the synchronization of the RK SFBs are stored (minimum length is 240 bytes).	CPU-specific, zero is not allowed.	0
EN_R	IN	BOOL	Control parameter "Enable to receive": Request enable	TRUE/FALSE	FALSE
R	IN	BOOL	Control parameter "Reset": Cancels the request.	TRUE/FALSE	FALSE
LADDR	IN	WORD	Submodule I/O address you specified in "HW Config".	CPU-specific	3FF hex
NDR	OUT	BOOL	Status parameter "New data ready" (Set only for the duration of one call): Request completed without errors • FALSE: The request has not yet been started or is still running. • TRUE: The request has been completed successfully.	TRUE/FALSE	FALSE
ERROR	OUT	BOOL	Status parameter (Set only for the duration of one call): Request completed with errors	TRUE/FALSE	FALSE
STATUS	OUT	WORD	Status parameter (Set only for the duration of one call. In order to display STATUS, you should therefore copy STATUS to a free data area): STATUS has the following significance, depending on the ERROR bit: • ERROR = FALSE: The STATUS value is: – 0000H: Neither warning nor error – <> 0000H: Warning, STATUS supplies detailed information • ERROR = TRUE: Error pending. STATUS supplies detailed information about the type of error (for error numbers, see Section Error Messages (Page 351)).	0 to FFFF hex	0

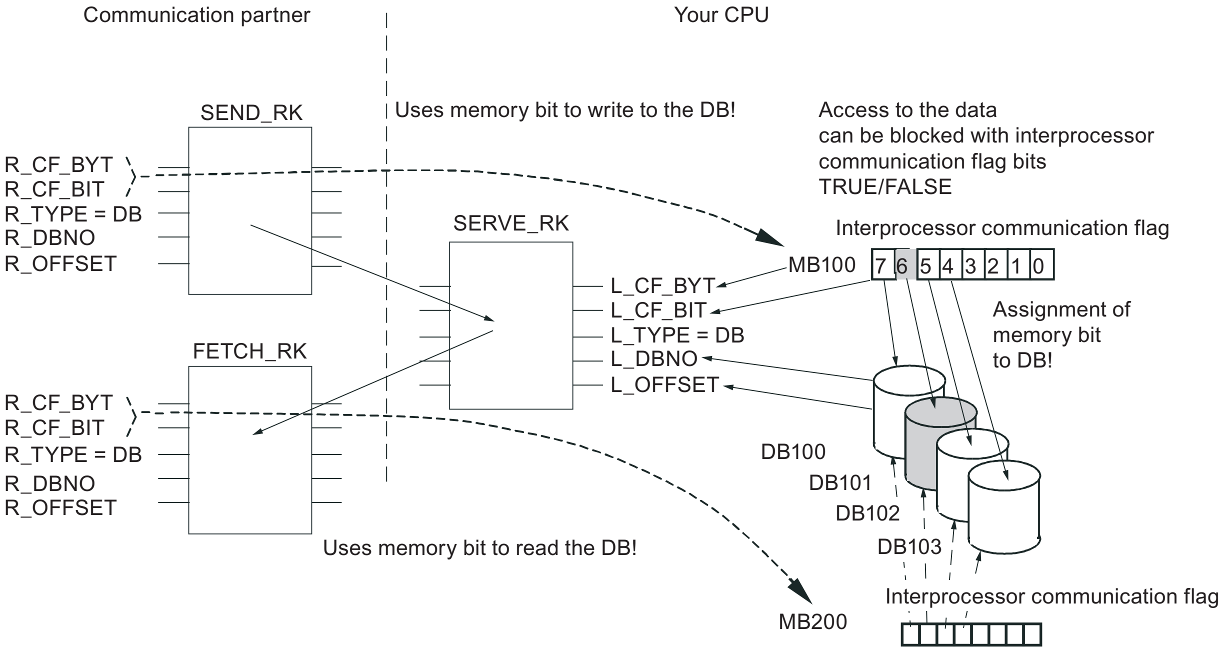
Parameter	Declaration	Data type	Description	Value range	Default
LEN	IN_OUT	INT	Message frame length, number in bytes (Set only for the duration of one call)	0 to 1024	0
L_TYPE	OUT	CHAR	(L_... parameters are set for the duration of one call only.) Receiving data: Type of target area on local CPU (capital letters only): 'D' = Data block	'D'	' '
			Providing data: Type of the source area on local CPU (capital letters only): • 'D' = Data block • 'M' = Memory bit • 'E' = Inputs • 'A' = Outputs • 'Z' = Counters • 'T' = Timers	'D', 'M', 'E', 'A', 'Z', 'T'	
L_DBNO	OUT	INT	Data block number on local CPU	CPU-specific	0
L_OFFSET	OUT	INT	Data byte number on local CPU	0-510	0
L_CF_BYT	OUT	INT	Interprocessor communication flag byte on local CPU (255: means: no interprocessor communication flag)	0 to 255	0
L_CF_BIT	OUT	INT	Interprocessor communication flag bit on local CPU	0 to 7	0

6.5.2.5 Example: Use of Interprocessor Communication Flags

Basics

You can block and enable SEND/FETCH requests of your communication partner via interprocessor communication flag. This prevents overwriting and reading of data which are not yet processed.

You can specify an interprocessor communication flag for every request.



6.5.2.6 Example: SEND_RK with Interprocessor Communication Flag

Procedure

In this example the communication partner transmits data to DB101 on your CPU.

1. On your CPU, set the interprocessor communication flag 100.6 to FALSE.
2. At your communication partner, specify the interprocessor communication flag 100.6 (parameter R_CF_BYT, R_CF_BIT) with the SEND request.

   The interprocessor communication flag is sent to your CPU in the RK 512 message frame header.

   Before it processes the request, your CPU checks the interprocessor communication flag specified in the RK 512 message frame header. The request is only processed if the interprocessor communication flag status on your CPU is FALSE. If the interprocessor communication flag status is TRUE the error message "32 hex" is returned to the communication partner in the response message frame.

   Once the data has been transferred to DB101, SFB SERVE sets the status of the interprocessor communication flag 100.6 on your CPU to TRUE and the flag byte and bit are output on SFB SERVE for the duration of one call (if NDR = TRUE).
3. You can recognize that the request is completed and that the transmitted data are ready for processing when you evaluate the interprocessor communication flag (interprocessor communication flag 100.6 = TRUE) in your user program.
4. After having edited the data in your user program, you must reset the interprocessor communication flag 100.6 to FALSE. This allows your communication partner to repeat execution of the request without error.

Data Consistency

Data consistency is limited to 128 bytes. You must note the following points for the transmission of consistent data with a length of more than 128 bytes:

Use the interprocessor communication flag function. Do not access the data until all data have been transmitted (evaluation of the interprocessor communication flag specified for this request; the interprocessor communication flag is active at the SFB for the duration of one call if NDR = TRUE). Do not reset the interprocessor communication flag status to FALSE until you have edited the data.

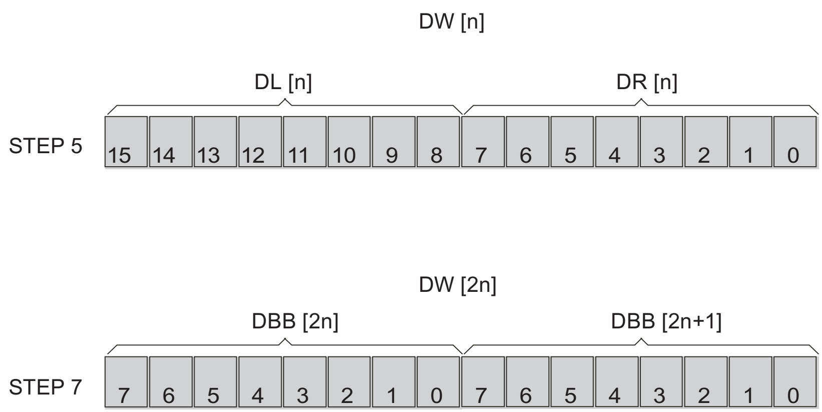
6.5.3 Information for programming of system function blocks

Introduction

This chapter addresses all users who upgrade SIMATIC S5 to SIMATIC S7. The following sections explain the points you have to take into account when programming FBs in STEP 7.

Addressing

Data operands in DBs are addressed byte-wise in STEP 7 (in contrast, in STEP 5 addressing by words. You therefore have to convert the addresses of the data operands.



Otherwise than in STEP 5, the data word address is duplicated in STEP 7. It is no longer divided into a low and a high data byte. The bits are always numbered from 0 through 7.

Examples

Conversion of STEP 5 data operands (left table column) into STEP 7 data operands (right table column).

STEP 5	STEP 7
DW10	DBW20
DL10	DBB20
DR10	DBB21
D10.0	DBX21.0
D10.8	DBX20.0
D255.7	DBX511.7

Assigning the Block Parameters

Direct/Indirect Parameter Assignment

Indirect parameter assignment as under STEP 5 (passing of the parameters in the currently opened DB) is not possible with STEP 7 blocks.

You can declare constants as well a variables in all block parameters. A differentiation between direct and indirect parameter assignment is therefore no longer required in STEP 7.

The parameter "LEN" for SFB 60, 63 and 64 forms an exception: parameters can only be assigned indirectly.

Example of "Direct Parameter Assignment"

Call of SFB 60 "SEND_PTP" according to "Direct parameter assignment":

STL

```
Network 1:
     CALL SFB 60, DB10
         REQ         := M 0.6                  //initiate SEND
         R           := M 5.0                  //initiate RESET
         LADDR       := +336                   //I/O address
         DONE        := M 26.0                 //End without error
         ERROR       := M 26.1                 //End with error
         STATUS      := MW 27                  //Status word
         SD 1        := P#DB11.DBX0.0          //Data block DB 11,
                                               //As of data byte DBB 0
         LEN         := DB10.DBW20             //Length is indirectly assigned parameters
```

Example of "Symbolic Addressing of the Current Operand"

Call of SFB 60 "SEND_PTP" with symbolic addressing of the current operands:

STL

```
Network 1:
     CALL SFB 60, DB10
         REQ         := SEND_REQ               //initiate SEND
         R           := SEND_R                 //initiate RESET
         LADDR       := BGADR                  //I/O address
         DONE        := SEND_DONE              //End without error
         ERROR       := SEND_ERROR             //End with error
         STATUS      := SEND_STATUS            //Status word
         SD_1        := QUELLZEIGER            //ANY pointer to target area
         LEN         := CPU_DB.SEND_LAE        //Message frame length
```

6.6 Commissioning

6.6.1 Commissioning the Interface Hardware

Procedure

You should test the connection if a communication cannot be established to the partner device after the configuration has been completed. Proceed as follows:

Step	What to do
1	Locating the cause of error:
	• Reversed polarity of the send/receive lines? • Are the default settings correct? Several default setups might have been carried defaults with different polarity. Some default settings are integrated in the device on a permanent basis. • Missing or wrong terminating resistors? • Reversal of high byte and low byte of the security word (e.g. CRC)?
2	Procedure:
	• First, check the cable connection with the help of the manual: – Assignment/polarity (see Section Connecting a Serial Cable (Page 263)) – Default settings (see Section Basic parameters (Page 266)) • The test per test setup
3	Prepare the test setup as simple as possible:
	• Interconnect only 2 nodes • Set RS485 mode (2-wire operation) if possible • Use a short connecting cable • Terminating resistors are not required for short distances • Test data transmission in both directions

Step	What to do
4	Check: • Case 1: the line polarity is definitely correct – Vary the default settings (all options) – Check the security word (e.g. CRC) • Case 2: the default settings are definitely correct – Cross-link the connections (Note: for RS422 cross-link both line pairs) – Check the security word (e.g. CRC) • Case 3: correct polarity or correct default setting not known – Cross-link the connections (Note: for RS422 cross-link both line pairs) – If not OK, change the default settings (all options) with an appropriate attempt to establish communication – If not OK, change the connections back, and change the default settings (all options) – Check the security word (e.g. CRC) • When you subsequently reassemble the system, do not forget to reconnect the terminating resistors you might have removed previously.
5	Additional tips: • If available, interconnect an interface tester (if required, an RS422/485 → V.24 converter) to the connecting line. • Check the signal level with a measuring device (measure signal to GND level on pin 8). • Some devices do not signal reception if data is being received but the CRC security word is not correct. • If required, replace the CPU to exclude an electrical fault.

6.7 Error Handling and Interrupts

6.7.1 Error locating and diagnostics

Diagnostics Possibilities

You can quickly locate pending errors by using the diagnostic functions. The following diagnostic options are available:

- Error Messages at the System Function Block (SFB)
- With RK512: Error IDs in the Response Message Frame
- Diagnostic interrupt

6.7.2 Error Messages at the System Function Block (SFB)

Basics

The parameter ERROR is set to TRUE if an error occurs. The STATUS parameter indicates the cause of error. The possible error messages are listed in Section Error Messages (Page 351).

Note

An error message is only output if the ERROR bit (job completed with error) is set. In all other cases the STATUS word is zero. If the ERROR bit is set you should therefore copy STATUS to a free data area to display the STATUS.

6.7.3 Error IDs in the Response Message Frame

Basics

If you are working with the RK 512 computer connection and a SEND/FETCH message frame error occurs at the communication partner, the communication partner returns a response message frame with an error ID in the 4th byte.

Table of Error IDs

In the table below you can find the assignment of the error IDs in the response message frame (REATEL) to the event classes/numbers in STATUS of the communication partner. The error IDs in the response message frame are output as hexadecimal values.

REATEL	Error messages (Event class/Event number)
0AH	0905H
0CH	0301H, 0609H, 060AH, 0902H
10H	0301H, 0601H, 0604H
12H	0904H
14H	0903H
16H	0602H, 0603H, 090AH
2AH	090DH
32H	060FH, 0909H
34H	090CH
36H	060EH, 0908H

6.7.4 Configuring and Evaluating Diagnostic Interrupts

Basics

You can trigger a diagnostic interrupt in case of a wire break in the serial connection to the communication partner (080DH). The diagnostic interrupt is displayed with incoming and outgoing error events.

In your user program, you can immediately respond to errors with the help of a diagnostic interrupt.

Procedure

1. Enable diagnostic interrupt in the "Basic parameters" dialog of the parameter assignment screens.
2. Incorporate the diagnostic interrupt OB (OB 82) in your user program.

Response to an Error with Diagnostic Interrupt

- Current operation is not influenced by the diagnostic interrupt.
- The CPU operating system calls OB82 in the user program.

Note

If the corresponding OB is not loaded the CPU switches to STOP when an interrupt is triggered.

- The CPU switches on the SF LED.
- The error is displayed in the diagnostic buffer of the CPU as "incoming" and "outgoing" event.

How a Diagnostic Interrupt is Evaluated in the User Program

After a diagnostic interrupt is triggered, you can evaluate OB 82 to check which diagnostic interrupt is pending.

- If the address of your submodule is written to OB 82, byte 6 + 7 (OB 82_MDL_ADDR), the diagnostic interrupt was triggered by the PtP connection of your CPU.
- Bit 0 of byte 8 in OB 82, bit 0 (Faulty module) is set as long as any errors are queued.
- Bit 0 of byte 8 in OB 82 is reset after all queued errors are reported as "outgoing".
- In the event of a line break in the serial connection, the "Faulty module", "Line break", "External error" and "Communication error" bits in byte 8 are set simultaneously.

OB 82, byte 8	Description:
Bit 0	Faulty module
Bit 1	-
Bit 2	External error
Bit 3	-
Bit 4	-
Bit 5	Line break
Bit 6	-
Bit 7	-

OB 82, byte 10	Description:
Bit 0	-
Bit 1	Communication error
Bit 2	-
Bit 3	-
Bit 4	-
Bit 5	-
Bit 6	Hardware Interrupt Loss
Bit 7	-

6.8 Installation of Examples

Using Examples

The examples (program and description) are found on the CD-ROM included in your documentation. You can also download them from the Internet. The project consists of several commented S7 programs of various complexity and aim.

The Readme.wri on the CD describes how to install the samples. After their installation the samples are stored in the catalog ...\STEP7\EXAMPLES\ZDt26_01_TF_____31xC_PtP.

6.9 Protocol Description

6.9.1 Data Transfer with ASCII Driver

6.9.1.1 Data Transfer with ASCII Driver – Basics

Basics

The ASCII driver controls data transfer in PtP communications between the CPU and a communication partner.

This provides an open structure for the message frames, since the S7 user passes the complete send message frame to the PtP interface. For the receiving direction, the end criterion of a message frame must be declared in the parameters. The structure of the send message frames may differ from that of the receive message frames.

The ASCII driver allows an open structure of data (all printable ASCII characters as well as all other characters from 00 through FFh (with 8 data bit character frames) or from 00 through 7Fh (with 7 data bit character frames) to be sent and received.

Operation is possible with RS422 and RS485.

RS422 Operation

In RS422 mode, data are transmitted across a four-wire serial cable (four-wire operation). Two conductors (differential signal) are available for the send direction and two for the receive direction. This means you can send and receive data at the same time (Full Duplex operation).

RS485 Operation

In RS485 mode data are transmitted across a two-wire serial cable (two-wire operation). The two wires (differential signal) are available alternately for the send and receive direction. This means you can either send or receive data at a given time (Half Duplex operation). After a send operation, the cable is immediately switched over to receive mode (transmitter is switched to high-impedance). The maximum toggle time is 1 ms.

6.9.1.2 Sending Data with the ASCII Driver

Basics

In the "LEN" parameter, you specify the number of bytes for user data to be transmitted when you call the SFB for a send operation.

When you work with the **end criterion "Expiration of character delay time"**, the ASCII driver also generates a waiting time between two message frames during a send operation. You can always call the SFB. However, the ASCII driver does not start output of data until an interval greater than the configured character delay time has expired since the last transmission of a message frame.

If you work with the **end criterion "Fixed character length"**, data are transmitted in sending direction according to the data length declared in the SFB SEND_PTP at the parameter "LEN". In receiving direction, that is, in the receive DB, the data length is entered as you have entered them via parameter assignment screen at the receiver in the parameter "Fixed character length". The two parameter settings must be identical in order to ensure error-free data traffic. When sending data, a pause equal to the length of the monitoring time with missing end code is inserted between two message frames to allow partner synchronization (recognition of message frame start).

If some other synchronization method is used, the send pause can be deactivated by means of the parameter assignment screen.

If you work with the **end-of-text character criterion**, you have three options:

1. Transmission including the end-of-text character:
   The end code must be included in the data to be sent. Data are only transferred up to the end code, even if a greater data length is specified in the SFB.
2. Transmission up to the length specified in the SFB parameters:
   Data are transferred up to the length declared in the SFB parameters. The last character must be the end-of-text character.
3. Transfer of data up to the length configured in the SFB parameters and automatically appending the end code/s:
   Data are transferred up to the length declared in the SFB parameters. The end-of-text character(s) is/are appended automatically; that is, the end code must not be included in the data to be transferred. Depending on the number of end delimiters, 1 or 2 characters more than specified in the SFB (maximum 1024 bytes) are transferred to the partner.

Note

When configuring XON/XOFF flow control, user data must not contain any of the configured XON or XOFF characters. Default settings are DC1 = 11H for XON and DC3 = 13H for XOFF.

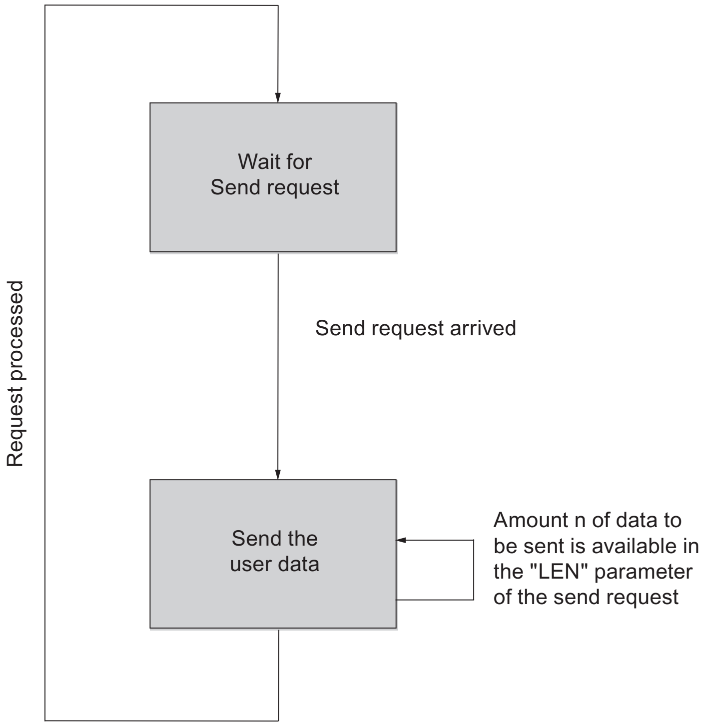
Transmission of Block Check Characters

If you want to ensure data integrity, using one (or two) block check character(s) (BCC), you must use the setting "Sending up to the length declared in the SFB parameters" in the end criterion "End-of-text characters". You can then append one (two) additional block check character(s) to the sent end-of-text character.

You must calculate the block check character directly in the user program.

Sending Data

The figure below illustrates a send operation:



6.9.1.3 Receiving Data Using the ASCII Driver

Basics

For data transmission with the ASCII driver you can choose between three different end criteria. The end criterion defines the point at which a message frame has been received completely. Configurable end criteria are:

- Expiration of character delay time:
  The message frame has neither a fixed length nor a defined endoftext character; the end of the message is defined by a pause on the line (expiration of character delay time).
- Receiving a fixed number of characters:
  The length of the received message frames is always identical.
- Receiving the end-of-text character(s):
  The end of the message frame is marked by one or two defined endoftext characters.

Code transparency

The code transparency of the procedure depends on the selection of the configured end criterion and flow control:

- With one(two) end character(s):
  Not code transparent
- End criterion character delay time or fixed character length:
  code transparent
- Code-transparent operation is not possible with XON/XOFF flow control operation.

"Code-transparent" means that user data can contain any character combinations, without recognition of the end criterion.

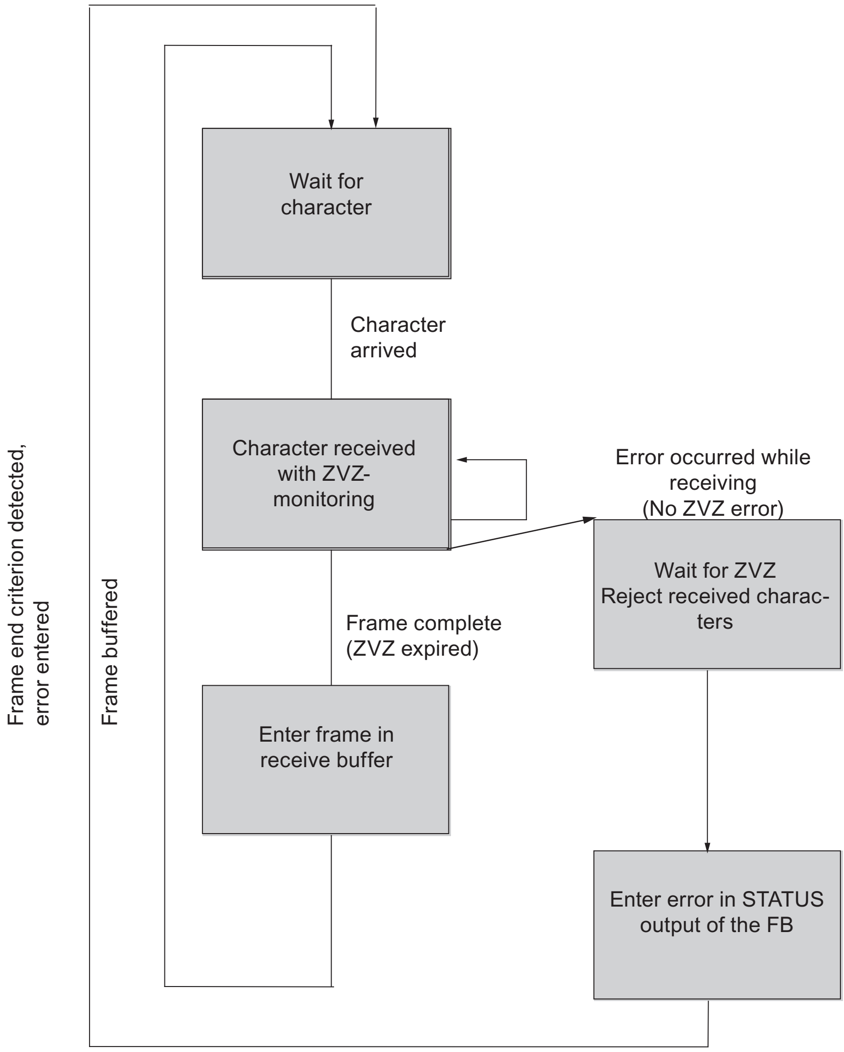
The End Criterion Expiration of the Character Delay Time

When receiving data, the end of the message frame is recognized on expiration of the character delay time. The received data are accepted by the CPU.

In this case, the character delay time must be configured to ensure its expiration between two consecutive message frames. However, it should be of sufficient length to exclude wrong identification of the end of the message frame whenever the communication partner performs a send pause within a message frame.

Procedure

The figure below illustrates a receive operation with the end criterion "On expiration of character delay time":



End Criterion Fixed Character Length

When receiving data, the end of the message frame is recognized after the declared number of characters have arrived. The received data are accepted by the CPU.

If the character delay time expires before the declared number of characters has been reached, the receive operation is closed. The character delay time is used in this situation as monitoring time. An error message is generated and the message frame fragment is discarded.

Special Considerations

If the character length does not conform with the configured fixed length, note the following:

- The length of the received characters is greater than the configured fixed length:

  All characters received after the configured fixed character length is reached are either:

  - Discarded if the monitoring time expires at the end of the message frame.
  - Merged with the next message frame if a new message frame is received before the monitoring time expires.

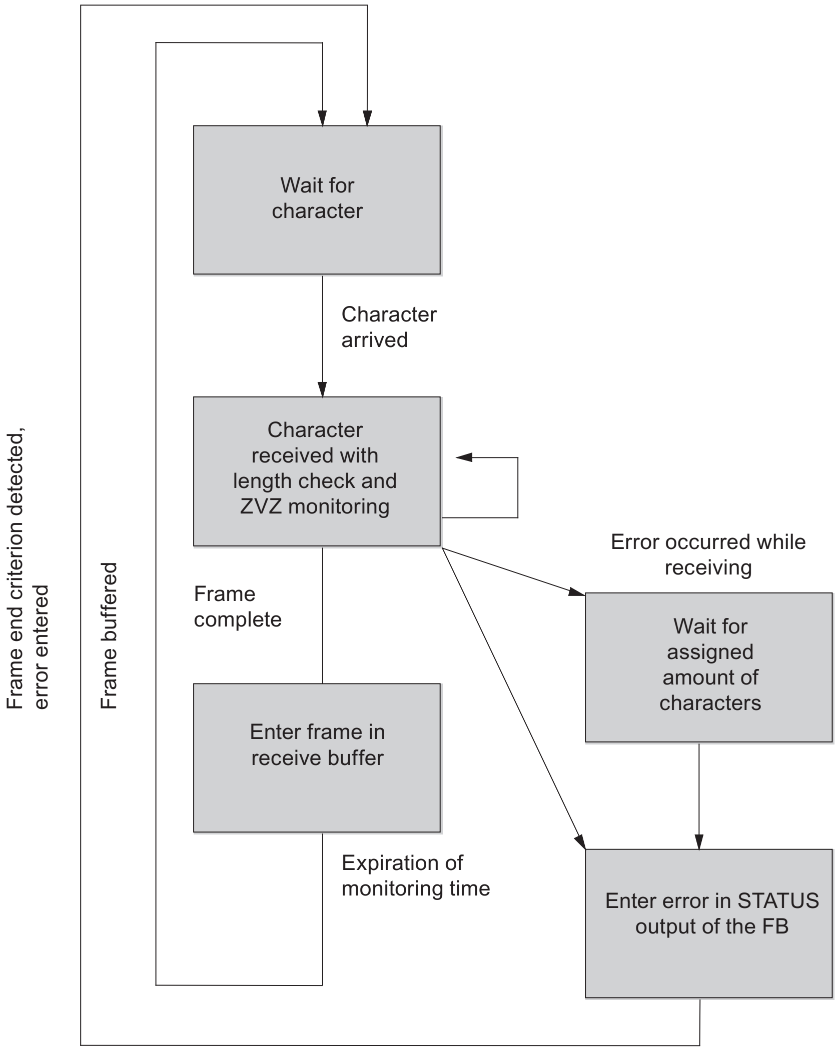
- The length of the received characters is less than the configured fixed length:

  The message frame is either:

  - Discarded if the monitoring time expires at the end of the message frame.
  - Merged with the next message frame if a new message frame is received before the monitoring time expires.

Procedure

The figure below illustrates a receive operation with the end criterion "Fixed character length":



End Criterion End-of-Text Character

When receiving data, the end of the message frame is recognized when the declared endof-text character(s) is(are) received. You have the following options:

- One endoftext character
- Two endoftext characters

The received data including the endoftext character(s) are accepted from the CPU.

If the end code is missing in the received data, the character delay time expires during reception and results in a termination of the frame. The character delay time is used in this situation as monitoring time. An error message is issued and the message frame fragment is discarded.

If you are working with endoftext characters, transmission is not code transparent. You must then make sure that the end code(s) is/are not included in the user data of the user.

Special considerations

If the last character in the received message frame is not the endoftext character, note the following points:

- End-of-text character elsewhere in the message frame:

  All characters, including the end-of-text character, are written to the DB. The characters appended to the end-of-text character are

  – Discarded if the monitoring time expires at the end of the message frame.

  – Merged with the next message frame if a new message frame is received before the monitoring time expires.

- End-of-text characters not included in message frame:

  The message frame is either:

  – Discarded if the monitoring time expires at the end of the message frame.

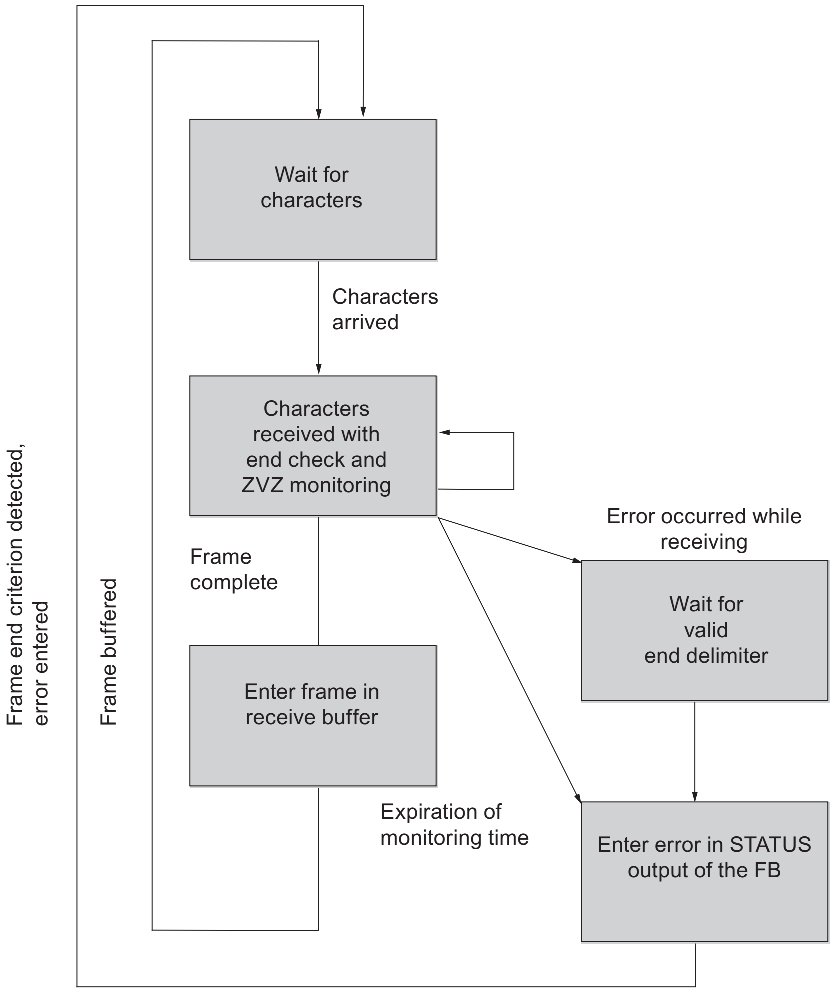
  – Merged with the next message frame if a new message frame is received before the monitoring time expires.

Receiving with Block Check Characters

In the parameter assignment screens, you can select operation with one (or two) block check character(s) (BCC), in addition to the end-of-text character. The characters (one or two) appended to the end-of-text character are also written to the receive DB.

You must calculate the block check character directly in the user program.

The figure below illustrates a receive operation with the end criterion "Endoftext character":



Receive Buffer of the CPU

The receive buffer is 2048 bytes. In you configuration, you can choose to prevent overwriting of data in the buffer. You can also specify the value range (1 to 10) for the number of buffered received frames or use the entire receive buffer.

You can clear the receive buffer at startup. You specify this either in the parameter assignment screen or by calling the SFB RES_RCV (see Section Clearing the Receive Buffer with SFB 62 "RES_RCVB" (Page 285)).

The receive buffer is a ring buffer:

- If multiple message frames are written to the receive buffer: Always the first received message frame is transmitted to the target block.
- If you always want to transfer the last received frame to the destination data block, you must assign the value "1" for the number of buffered frames and deactivate the overwrite protection.

Note

If continuous reading of received data is interrupted for a certain time in the user program and new received data are requested, it might happen that the old message frame is written to the target block prior to the latest message frame.

Old message frames are the en route frames between the CPU and the partner at the time of the interrupt, or the ones already received by the SFB.

6.9.1.4 Data Flow Control/Handshaking

Basics

Handshaking controls data flow between two communication partners. Handshaking ensures that data are not lost during the communication between devices operating at different speed. The CPU supports the software handshake via XON/XOFF.

Procedure

Data flow control is realized as follows:

1. Once it is configured for operation with flow control the CPU transmits the XON character.
2. When the declared number of message frames, or 50 characters, are received before the receive buffer overflows (size of the receive buffer: 2048 bytes), the CPU transmits the XOFF character. If the communication partner nonetheless continues to transmit data, an error message is generated when the receive buffer overflows. Data received in the last message frame are discarded.
3. The CPU transmits the XON character once a message frame has been fetched from the receive buffer and when the receive buffer is ready to receive data.
4. The CPU interrupts data transfer when it receives the XOFF character. If no XON is received within a specific configured time, the send operation is canceled and a corresponding error message is generated (0708H) at the STATUS output of the SFBs.

6.9.2 Data Transfer with the 3964(R) Procedure

6.9.2.1 Data Transfer with the 3964(R) Procedure – Basics

Basics

The 3964(R) procedure controls data transfer in a PtP communication between the CPU and a communication partner.

Control character

For data transfer, the 3964R) computer connection appends control characters to the user data. These control characters allow the communication partner to verify that the data have arrived complete and without errors.

The 3964(R) procedure analyzes the following control codes:

- STX: Start of Text, start of the string to be transmitted
- DLE: Data Link Escape (data transmission switchover) or positive acknowledgment
- ETX: End of Text, end of the string to be transmitted
- BCC: Block Check Character (only for 3964(R)); Block check character
- NAK: Negative Acknowledgement

Note

When the connection is being established and shut down, a DLE character transmitted as information character is transferred twice across the send line (DLE duplication) in order to distinguish it from the DLE control character. The receiver cancels this DLE doubling.

Priority

With the 3964(R) procedure, one communication partner must be assigned a higher priority and the other partner a lower priority. If both partners start to establish a connection at the same time, the partner with the lower priority will defer its send request.

Block Checksum

With the 3964(R) data transmission protocol, data integrity is increased by sending an additional block check character (BCC = Block Check Character).

Message frame:

STX	Data			DEL ETX		BCC
02H	30H	31H	32H	10H	03H	20H

30	=	0011	0000
31	=	0011	0001
XOR	=	0000	0001
32	=	0011	0010
XOR	=	0011	0011
10	=	0001	0000
XOR	=	0010	0011
03	=	0000	0011
XOR	=	0010	0000
BCC		2	0

The block checksum represents the even longitudinal parity (XOR linking of all data bytes) of a sent or received block. Its calculation begins with the first byte of user data (first byte of the message frame) after the connection setup, and ends after the DLE ETX code upon disconnection.

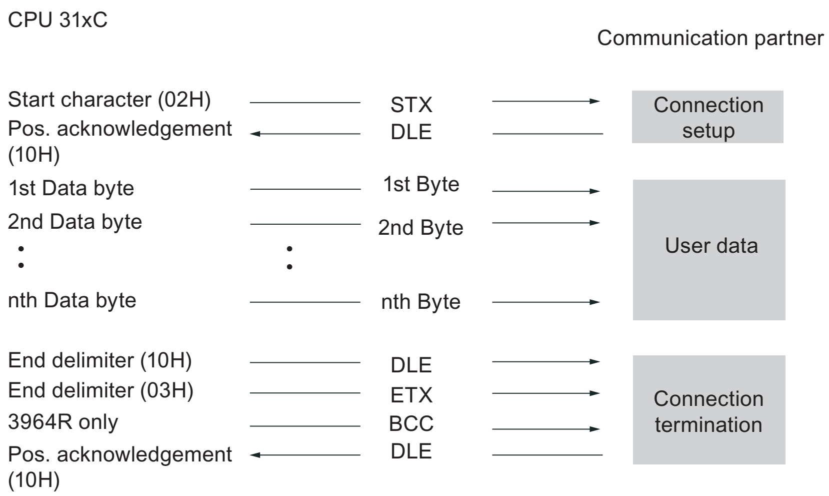
Note

With DLE duplication the DLE code is included twice in the BCC calculation.

6.9.2.2 Sending Data with 3964(R)

Procedure

The figure below illustrates the data flow when data are sent with the 3964(R) procedure:



Establishing a Connection during Data Transfer

To establish the connection, the 3964(R) procedure sends the control character STX. If the communication partner responds with the DLE character before the acknowledgment delay time (ADT) has expired, the procedure switches to send mode.

If the communication partner returns NAK or any other control code (except for DLE or STX), or if the acknowledgment delay time expires without a response, the procedure retries to connect. After the declared number of unsuccessful attempts to connect, the procedure aborts the attempt to set up a connection and transmits an NAK to the communication partner. The CPU reports the error to the SFB SEND_PTP (Output parameter STATUS).

Sending Data

After the connection is successfully established, the data are transferred to the communication partner, using the selected transmission parameters. The partner monitors the time intervals between the incoming characters. The interval between two characters must not exceed the character delay time.

If the communication partner transmits the NAK during a busy session, the procedure aborts the block and retries as described above, beginning with connection setup. If another character is transmitted, the procedure first waits for the character delay time to expire and then sends the NAK character to switch the communication partner to idle state. The procedure then restarts the send operation by establishing the connection STX.

Shutting down when Sending

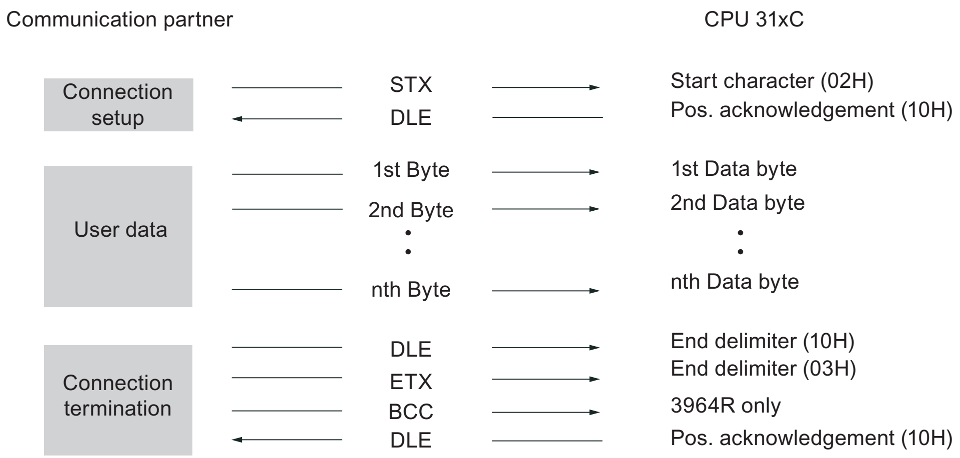
Once the contents of the buffer have been transmitted, the procedure appends the DLE ETX characters and, only in the case of 3964(R), the block checksum BCC as the end code, and then waits for an acknowledgment character. If the communication partner sends the DLE within the acknowledgment delay time, the data block has been received without errors. If the communication partner responds with NAK, any other code (except DLE), a corrupted character, or if the acknowledgment delay time expires without a response, the procedure restarts to send the data via connection setup STX.

After the configured attempts to send the data block, the procedure cancels the operation and transmits NAK to the communication partner. The error is displayed at SFB SEND_PTP (output parameter STATUS).

6.9.2.3 Receiving Data with 3964(R)

Procedure

The figure below illustrates the data flow when data are received with the 3964(R) procedure:



Note

When ready for operation, the 3964(R) procedure transmits the NAK character once to switch the communication partner to idle state.

Establishing a Connection when Receiving Data

In idle mode, when there is no send request to be processed, the procedure waits for the communication partner to establish the connection.

If no empty receive buffer is available when attempting to establish a connection via STX, a waiting time of 400 ms is started. If there is still no empty receive buffer after this time has expired, the STATUS output of the SFB displays the error. The procedure transmits an NAK character and returns to idle mode. Otherwise, the procedure transmits a DLE character and receives the data.

If the idle procedure receives any characters (except for STX or NAK), it waits for the character delay time to expire and then sends the NAK string. The error is displayed on the STATUS output of the SFB.

Receiving Data

After the connection is successfully established, incoming user data are written to the receive buffer. Of two consecutive DLE characters received only one is stored in the receive buffer.

After every received character, the next character is expected to be received within the character delay time. If this interval expires before another character is received, an NAK is sent to the communication partner. The system program reports the error to the SFB RCV_PTP (Output parameter STATUS).

If transmission errors occur during receiving (lost character, frame error, parity error, etc.), the procedure continues to receive data until the connection goes down. Then, an NAK is sent to the communication partner. A retry is then expected. If the block can still not be received without error after the number of retries declared in the static parameter record, or if a retry is not started by the communication partner within the specified block check time (corresponds with the acknowledgment delay time), the procedure cancels the receiving operation. The CPU reports the first faulty transmission and the final cancellation to the SFB RCV_PTP (Output parameter STATUS).

Closing the Connection when Receiving Data

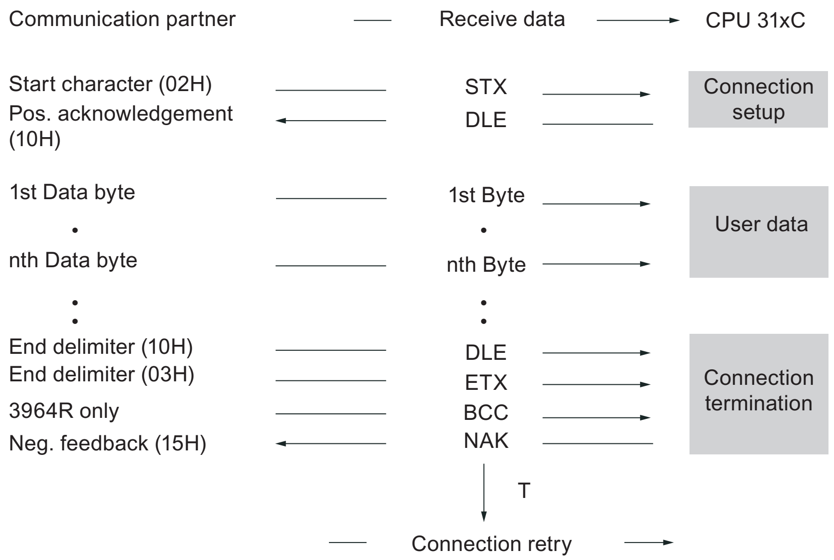
When the 3964 procedure detects the DLE ETX string, it cancels the receiving operation and transmits a DLE to the communication partner if the block was received without errors. If an error has occurred when receiving data it transmits an NAK. A retry is then expected.

The 3964(R) procedure stops receiving when it detects the character string DLE ETX BCC. It compares the received BCC with the internally calculated longitudinal parity. If the BCC is correct and no other receive errors have occurred, 3964(R) sends a DLE and returns to idle mode. If the BCC is faulty or a different receive error occurs, an NAK is sent to the communication partner. A retry is then expected.

6.9.2.4 Error Handling for Sending and Receiving with the 3964 (R) Procedure

Handling Corrupted Data

The view below shows how corrupted data are handled with 3964(R):

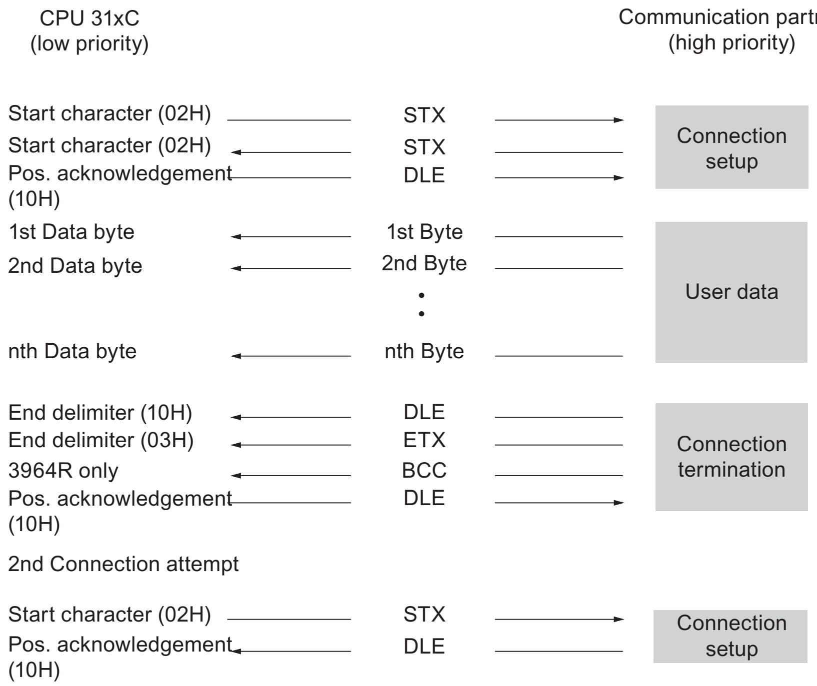


After receiving the DLE, ETC and BCC, the CPU compares the BCC of the communication partner with its internally calculated value. If the BCC is correct and if no other receive error has occurred the CPU responds with DLE.

Otherwise, it responds with NAK and a retry within the block check time. If the block cannot be received within the configured number of attempts, or if no further attempt is made within the block check time, it cancels the receive operation.

Initialization Conflict

The figure below illustrates data flow when an initialization conflict has occurred:



There is an initialization conflict if a device responds to the communication partner's send request (code STX) within the acknowledgment delay time (ADT) by sending the code STX instead of the acknowledgment DLE or NAK. Both devices request to send. The device with the lower priority defers its send request and responds with the code DLE. The device with the higher priority sends its data as described above. Once the connection is closed, the lower priority device can execute its send request.

To resolve initialization conflicts, you must configure different priorities for the communication partners.

Procedure Error

The procedure recognizes errors caused by faulty operation of the communication partner as well as errors caused by line disturbance.

In both cases, the procedure will retry to send/receive the data block correctly. If this is not possible within the maximum set number of transmission attempts (or if a new error status occurs), the procedure cancels the send or receive process. It generates the error ID of the first recognized error and then returns to idle mode. These error messages are displayed on the SFB STATUS output.

If the STATUS output of the SFB indicates a frequent repetitive send and receive attempt error, occasional disruptions of data transmission can be assumed. However, the large number of transmission attempts will compensate for this situation. In this case, you are advised to check the communication link for possible sources of interference, because frequent repetitions reduce user data transmission rates and transmission integrity. The disturbance could also be caused, however, by a malfunction on the part of the communication partner.

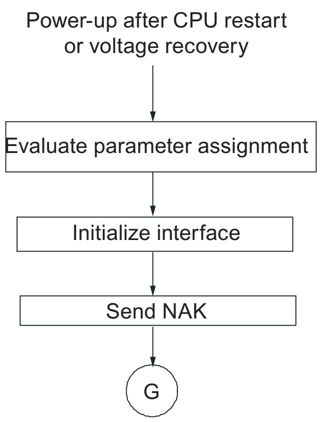
In the event of a BREAK on the receive line (receive line interrupted), an error message is displayed on the STATUS output of the SFB. No retry is started. BREAK status is automatically reset after the line is reconnected.

For all recognized transmission errors a unified error number is reported when a data block is received. However, the error is only reported following unsuccessful repetitions.

6.9.2.5 3964(R) Procedure Startup Sequence

Procedure

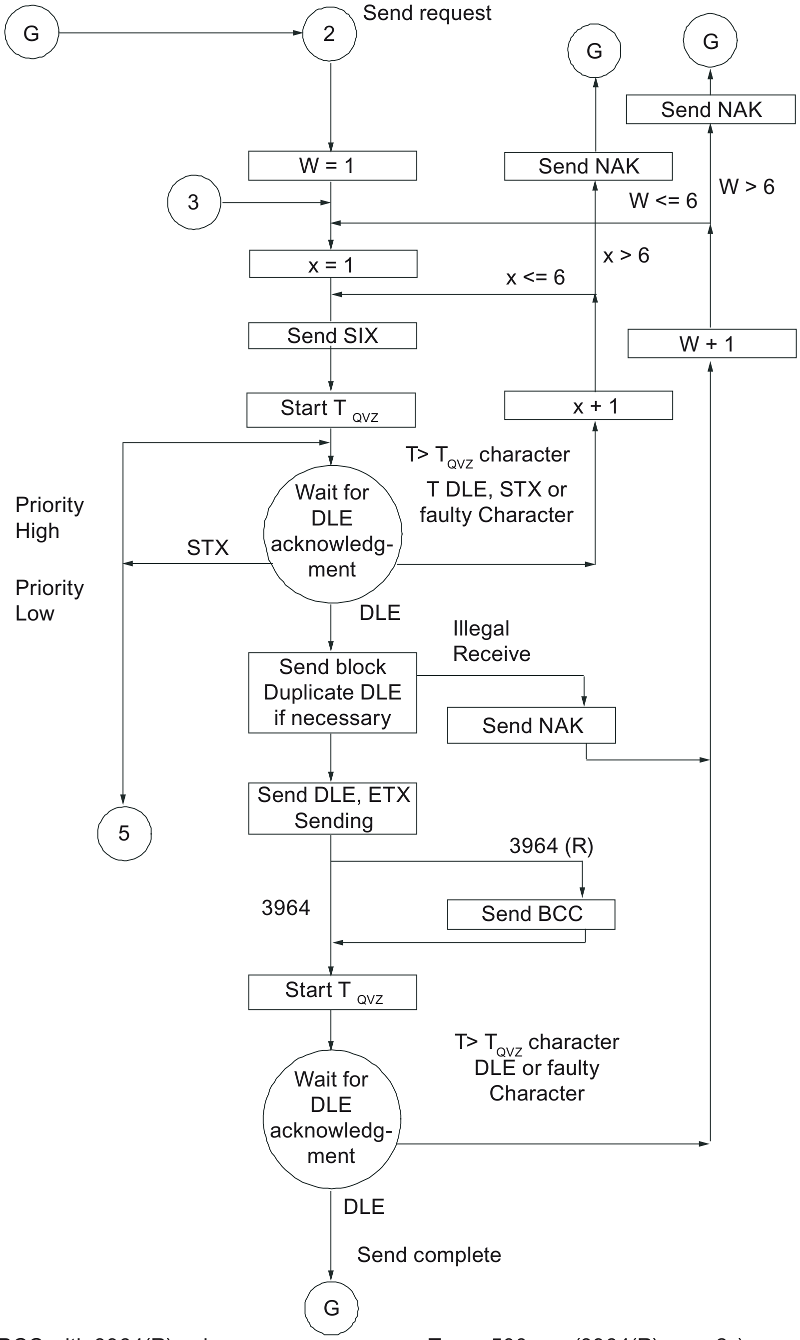
The picture below illustrates the startup sequence of the 3964(R) procedure:



6.9.2.6 Sending with 3964(R) Procedure

Procedure

The picture below illustrates the sending sequence with 3964(R) procedure:



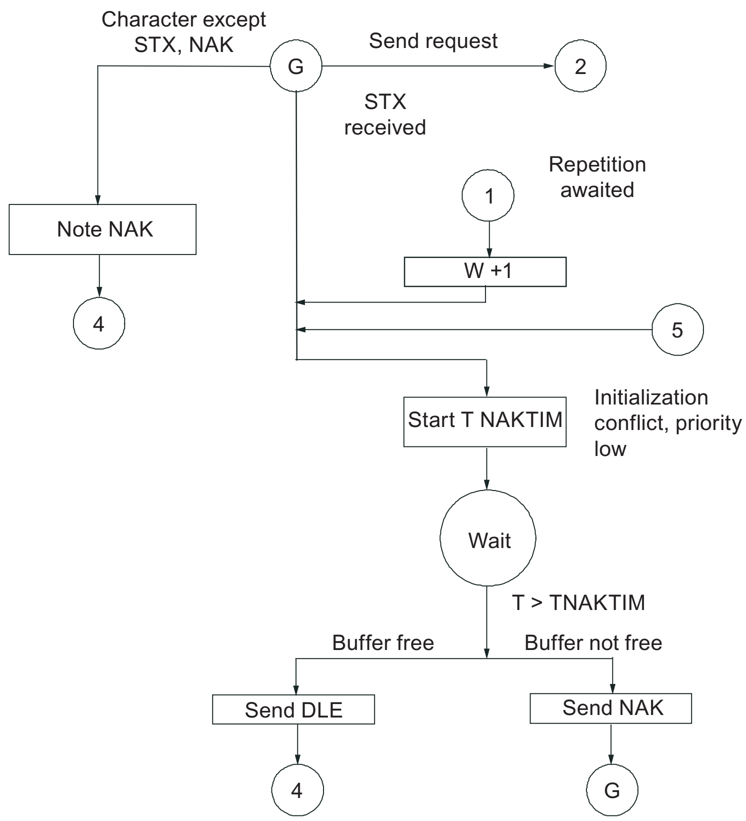
BCC with 3964(R) only
x =Counter connection retries
W= Counter transmission retries

T_{QVZ} = 500 ms (3964(R) $_{QVZ}$ = 2s)
Go immediately to initial state if wire break BREAK occurs

6.9.2.7 Receiving with 3964(R) procedure

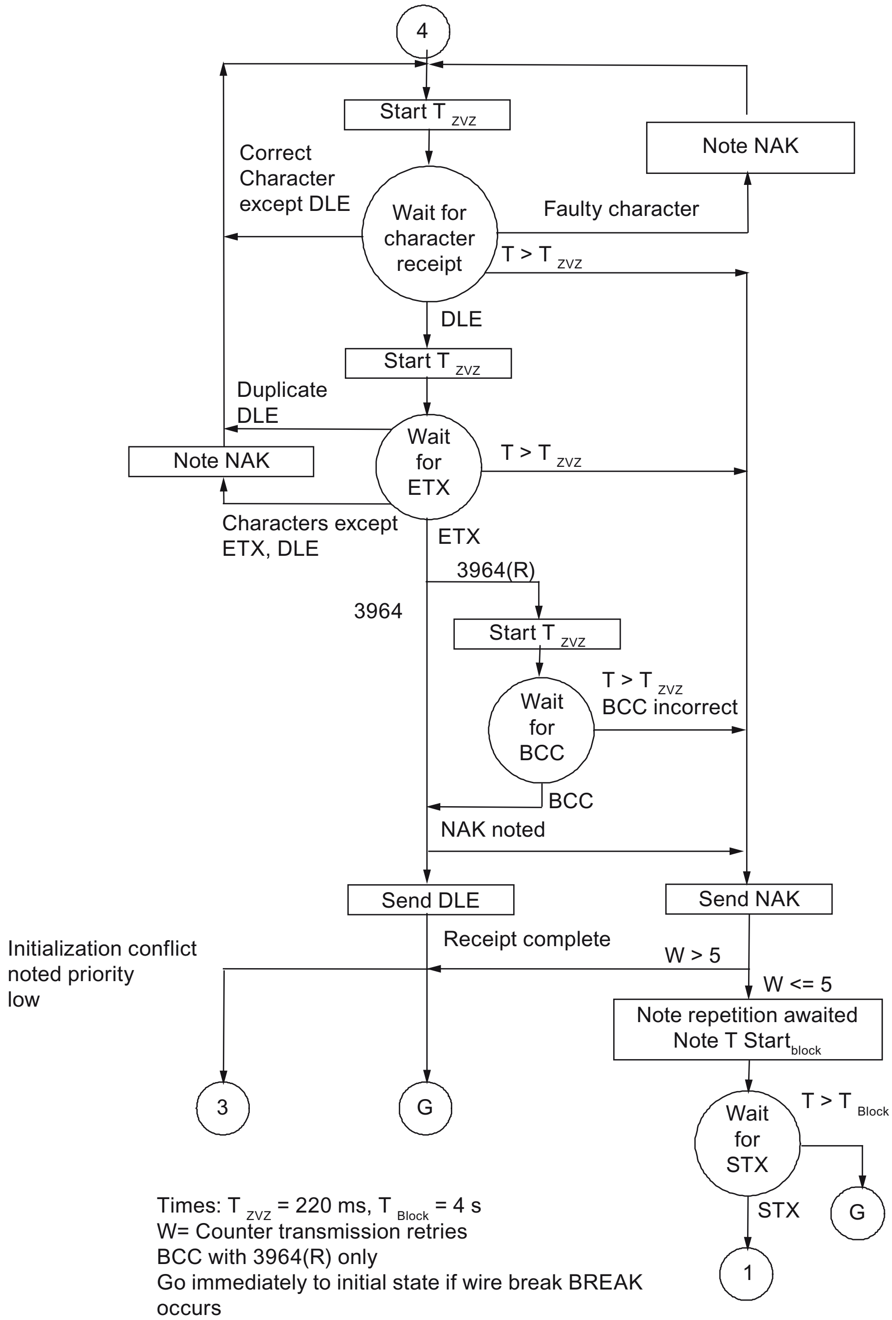
Receiving with 3964(R) procedure (part 1)

The picture below illustrates the receiving sequence with the 3964(R) procedure:



Receiving with 3964(R) procedure (part 2)

The picture below illustrates the receiving sequence with the 3964(R) procedure:



Receive buffer of the CPU

The receive buffer is 2048 bytes. In your parameter assignment, you can choose to prevent data in the buffer being overwritten. You can also specify the value range (1 to 10) for the number of buffered received frames or use the entire receive buffer.

You can clear the receive buffer upon startup. You specify this either in the parameter assignment screen or by calling the SFB RES_RCV (see section Clearing the Receive Buffer with SFB 62 "RES_RCVB" (Page 285)).

The receive buffer is a ring buffer:

- If multiple message frames are written to the receive buffer: The first message frame received is always transmitted to the target block.
- If you always want to transfer the last received frame to the destination data block, you must assign the value "1" for the number of buffered frames and deactivate the overwrite protection.

Note

If continuous reading of received data is interrupted for a while in the user program and new received data are requested, the latest message frame may not be entered in the target block until all old frames have been entered.

Old message frames are the frames in the process of being transferred between the CPU and the partner or already received by the SFB at the time of the interrupt.

6.9.3 Data Transmission with RK 512 Computer Connection

6.9.3.1 Data Transmission with the RK 512 Computer Connection – Basics

Introduction

RK 512 computer connection controls PtP data exchange between the CPU and a communication partner.

In contrast to 3964(R), the RK 512 computer connection offers higher data integrity and superior addressing options.

Response Message Frame

The RK 512 computer connection responds to every correctly received instruction frame with a response message frame to the CPU. This allows the sending station to verify that data were received without error by the CPU or that the data it requires are available on the CPU.

Command frame

Instruction frames are either SEND or FETCH message frames.

SEND Message Frame

When transmitting a SEND message frame, the CPU transmits an instruction frame including user data. The communication partner responds with a response message frame that contains no user data.

FETCH Message Frame

When transmitting a FETCH message frame, the CPU transmits an instruction frame that contains no user data. The communication partner responds with a response message frame that includes user data.

Sequential Message Frame

With SEND and FETCH message frames, sequential message frames are transmitted automatically if the data length exceeds 128 bytes.

Message Frame Header

With RK 512 computer connection, every message frame is initialized with a header. It can contain message frame IDs, information on data source and destination as well as an error number.

The table below shows the structure of the instruction frame header:

Byte	Description
1	Message frame ID in instruction frames (00H), for sequential instruction frames (FFH)
2	Message frame ID (00H)
3	• 'A' (41H): SEND request with target DB • 'O' (4FH): SEND request with target DX • 'E' (45H): FETCH request
4	Data to be transferred are fetched from a (only 'D' possible for sending): • 'D' (44H): Data block 'X' (58H) = expanded data block • 'E' (45H): Input bytes 'A' (41H) = Output bytes • 'M' (4DH): Memory bytes 'T' (54H) = Time cells • 'Z' (5AH): Counter cells
5	Data target with SEND job or data source with FETCH job, e. g. byte 5 = DB no., byte 6 = DW no.*
6	
7	High byte length: Length of data to be transmitted in bytes or word format depending on the type
8	Low byte length: Length of data to be transmitted in bytes or word format depending on the type
9	Byte number of the interprocessor communication flag; if you have not specified an interprocessor communication flag, the value FFh is entered here.

Byte	Description
10	• Bits 0 to 3: If you have not specified an interprocessor communication flag, the value FH is entered here. • Bits 4 to 7: CPU number (1 through 4); if you have not specified a CPU number (number 0), but rather an interprocessor communication flag, the value 0H is entered here; if you have not specified a CPU number or an interprocessor communication flag the value FH is entered here.
* RK 512 addressing describes the data source and destination with word limits. Conversion to byte address is executed automatically in SIMATIC S7.	

Bytes 3 and 4 contain ASCII characters.

The sequential instruction frame header consists only of the bytes 1 to 4.

Structure and Contents of the Response Message Frame

After the instruction frame is transmitted, RK 512 expects the response message frame of the communication partner within the monitoring time. The length of this monitoring time is 20 s.

The response message frame consists of 4 bytes and contains job processing information:

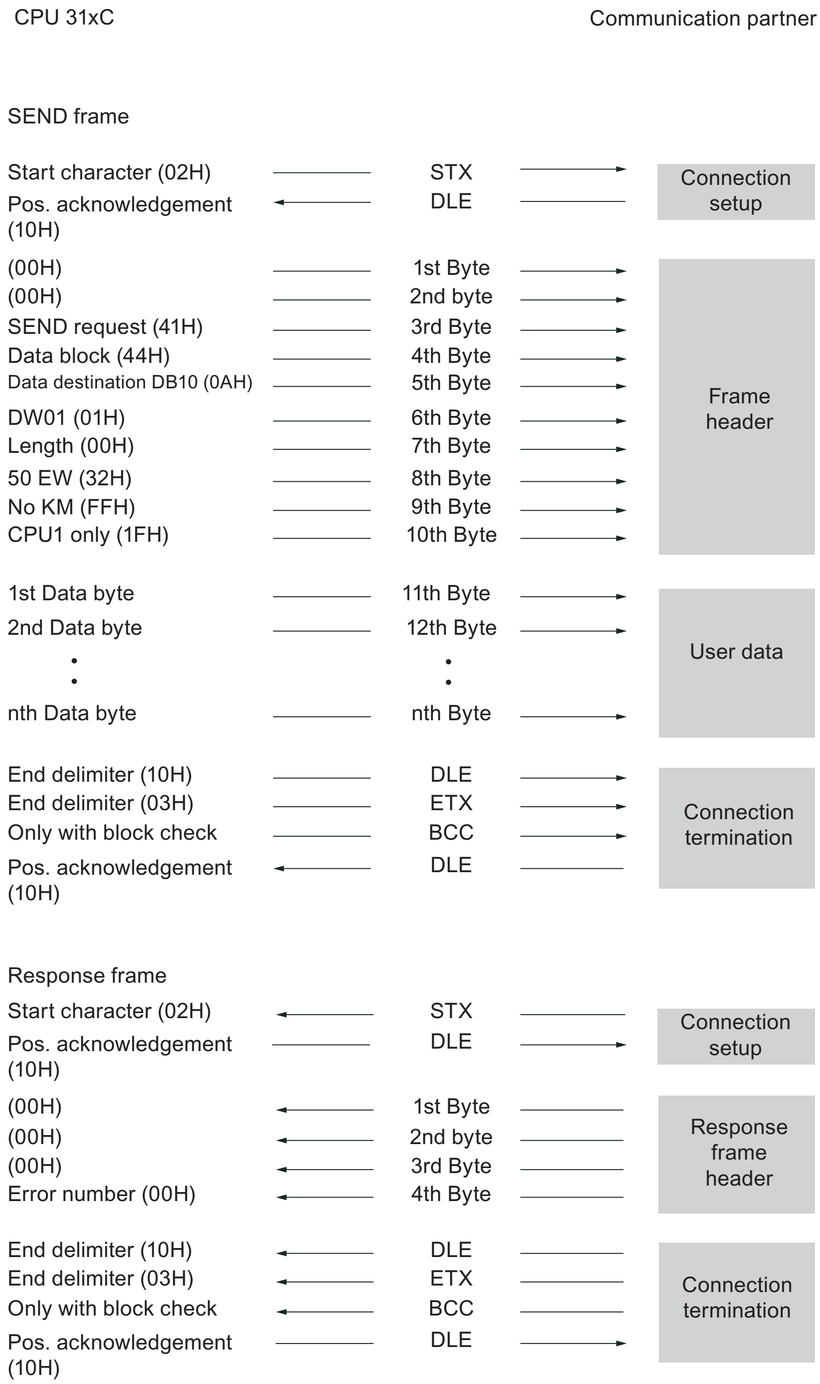
You can find structure and content of the response message frame in the table below.

Byte	Description
1	Message frame ID for response message frames (00H), for sequential response message frames (FFH)
2	Message frame ID (00H)
3	Assigned value 00H
4	Error number of the communication partner in the response message frame:* • 00H, if no error has occurred during transmission • > 00H error number
* An error number in the response message frame automatically generates an event ID at the STATUS output of the SFBs.	

6.9.3.2 Sending Data with RK 512

Procedure

The picture below shows the data transmission sequence when sending a response message frame using RK 512 computer connection:



Sending data

The SEND request is executed in the following sequence:

- Active partner

  Transmits a SEND message frame. Contains the message frame header and data.
- Passive partner

  Receives the message frame, verifies the header and the data and acknowledges with a response message frame after data have been written to the target block.
- Active partner

  Receives the response message frame.

  It sends a sequential SEND message frame if the user data length exceeds 128 bytes.
- Passive partner

  Receives the sequential SEND message frame, verifies the header and the data and acknowledges with a sequential response message frame after the data have been written to the target block.

Note

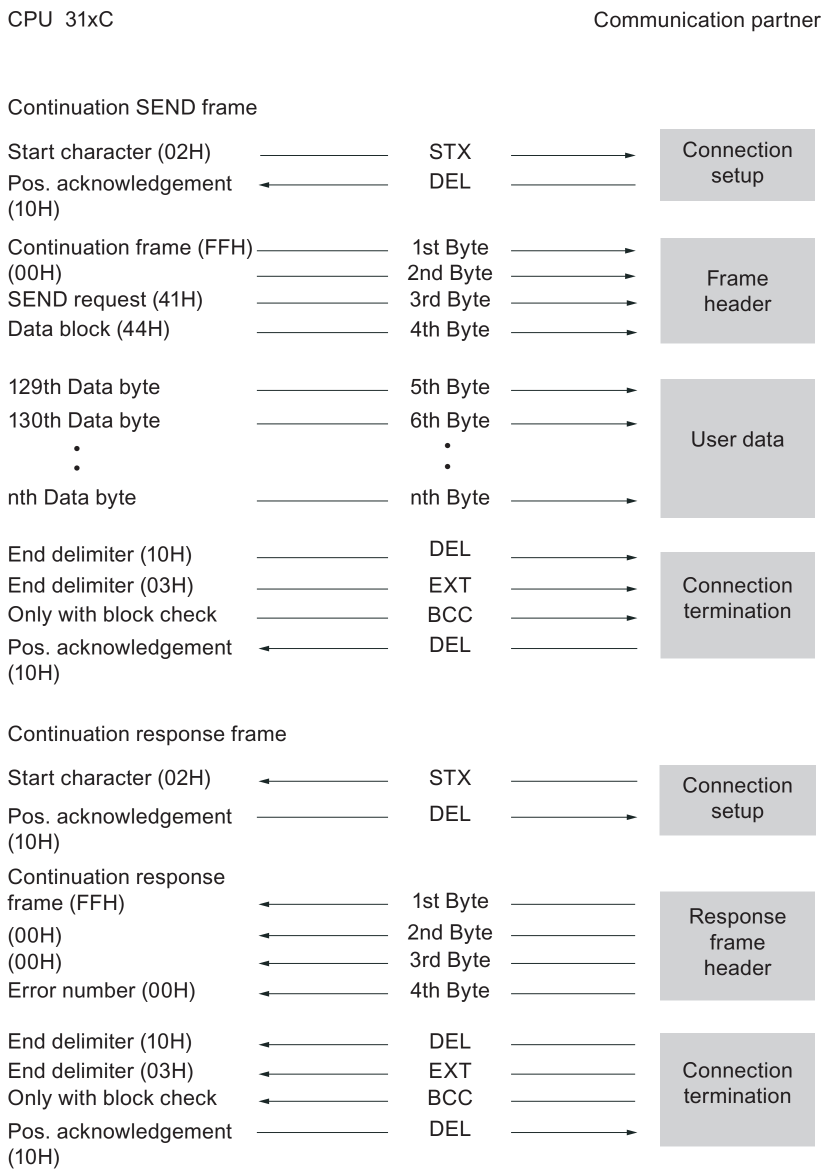
If the SEND message frame was not received without error by the CPU, or if an error has occurred in the message frame header, the communication partner enters an error number in byte 4 of the response message frame. In the case of protocol errors there will be no entry in the response message frame.

Sequential SEND Message Frames

A sequential SEND message frame is started if the data length exceeds 128 bytes. The sequence corresponds with that of the SEND message frame.

Bytes sent in excess of 128 bytes are automatically transmitted in one or multiple sequential message frames.

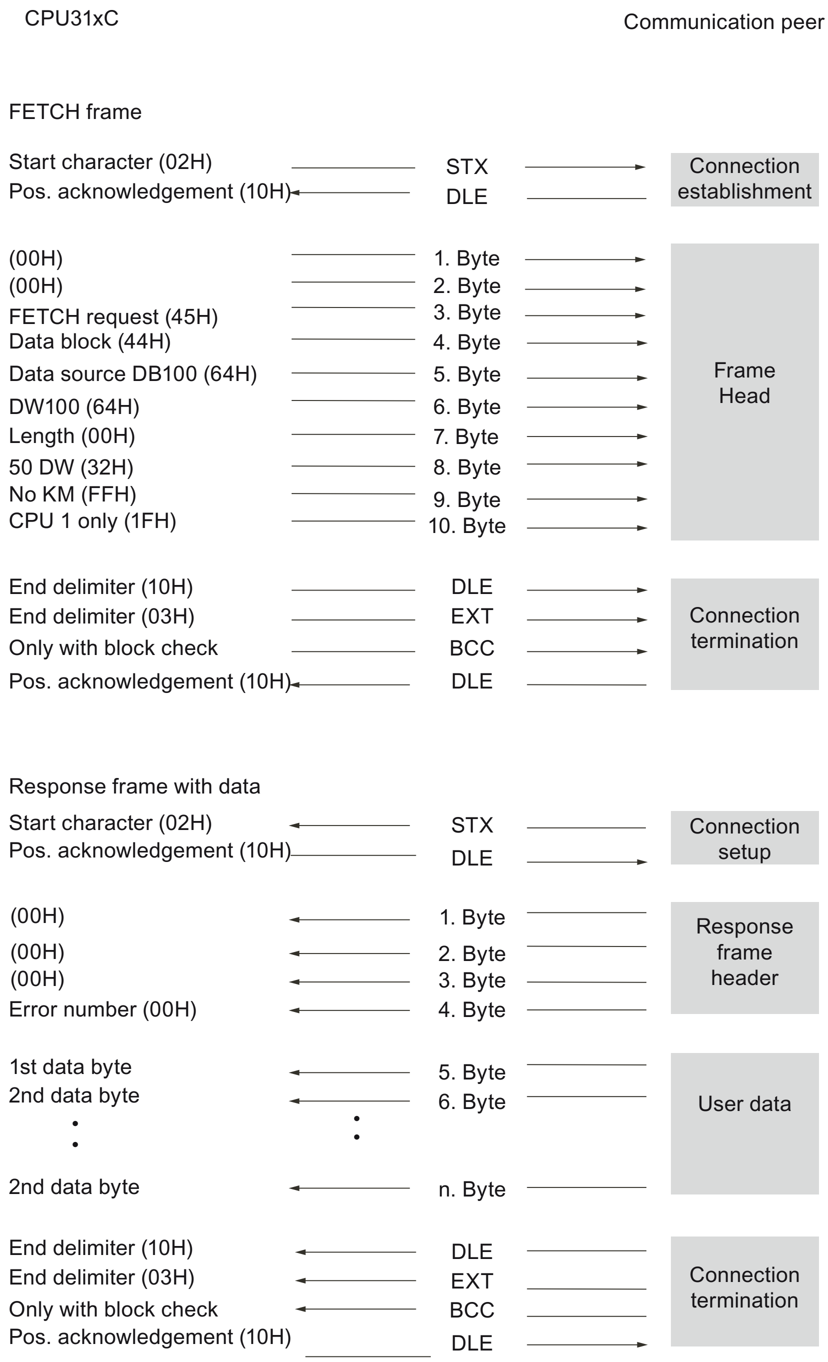
The picture below shows the data transmission sequence when sending a sequential SEND message frame with a sequential response message frame.



6.9.3.3 Fetching Data with RK 512

Procedure

The picture below shows the data transmission sequence when fetching data via response message frame with RK 512 computer connection.



Fetching Data

The FETCH request is executed in the following sequence:

1. **Active partner:**
   Transmits a SEND message frame. This contains the header.
2. **Passive partner:**
   Receives the message frame, verifies the header, fetches the data from the CPU, and acknowledges with a response message frame. This frame contains the data.
3. **Active partner:**
   Receives the response message frame.
4. It sends a sequential FETCH message frame if the user data length exceeds 128 bytes. This contains the header bytes 1 to 4.
5. **Passive partner:**
   Receives the sequential FETCH message frame, verifies the header, fetches the data from the CPU and acknowledges with a sequential response message frame that contains further data.

If there is an error number (not equal to 0) in the 4th byte, the response message frame does not contain any data.

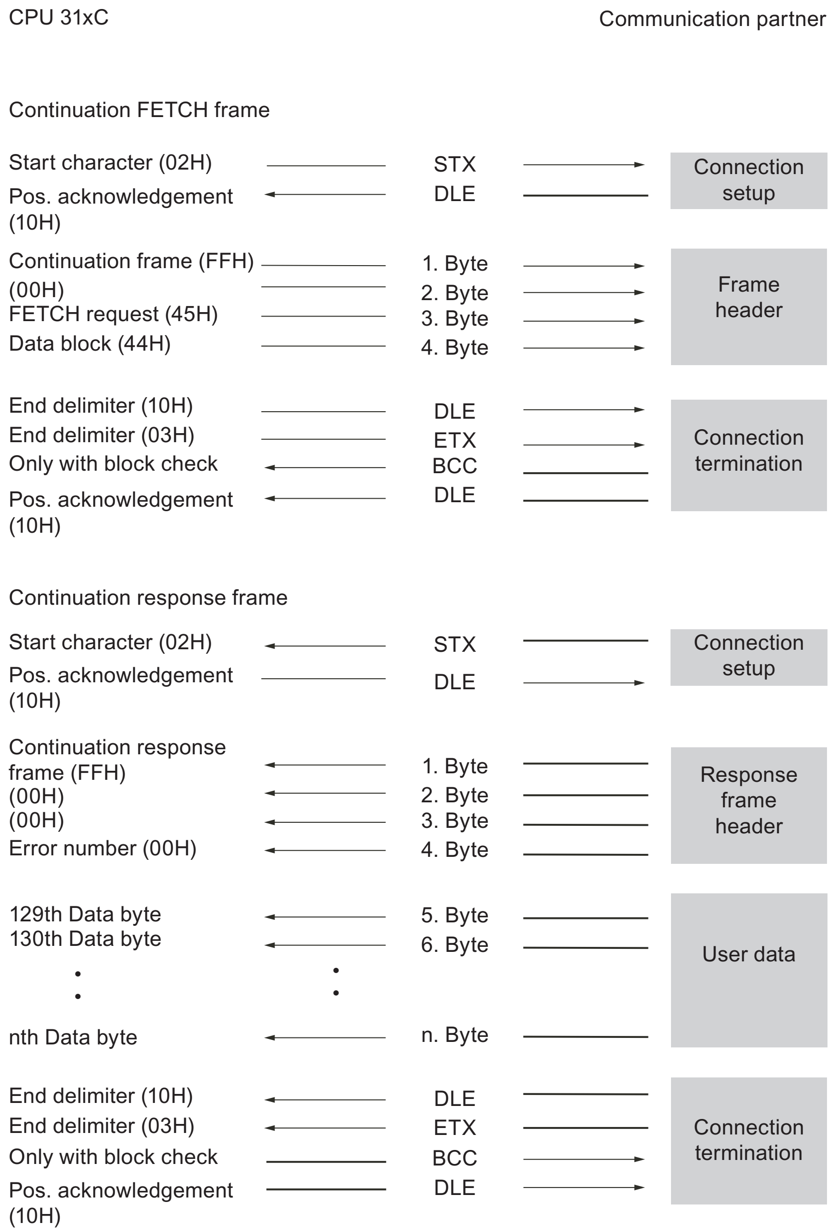
If more than 128 bytes are requested, the extra bytes are automatically fetched in one or more sequential message frame(s).

Note

If the FETCH message frame was not received without error by the CPU, or if an error has occurred in the message frame header, the communication partner enters an error number in byte 4 of the response message frame. In the case of protocol errors there will be no entry in the response message frame.

Sequential FETCH Message Frame

The picture below shows the data transmission sequence when fetching data with a sequential response message frame:

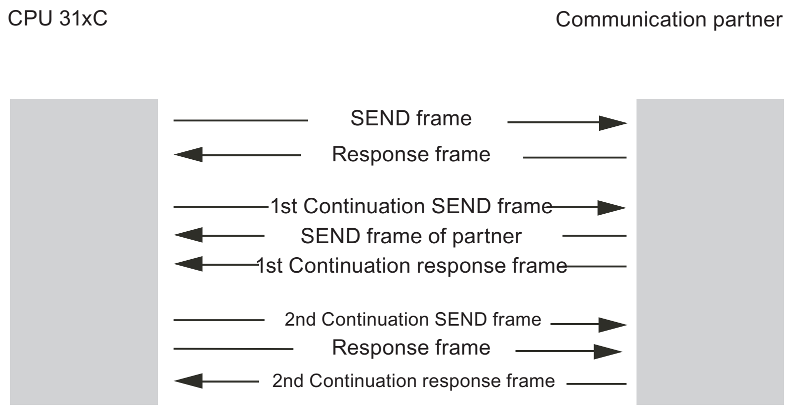


Pseudo Full Duplex Mode

Pseudo Full Duplex means: The partners can send instruction and response message frames at any time, even if the other partner is currently sending. The maximum nesting depth for instruction and response message frames is "1". Another instruction frame, therefore, cannot be processed until the previous one has been answered with a response message frame.

Under certain circumstances, if both partners request to send, it is possible to transmit a SEND message frame prior to the response message frame.

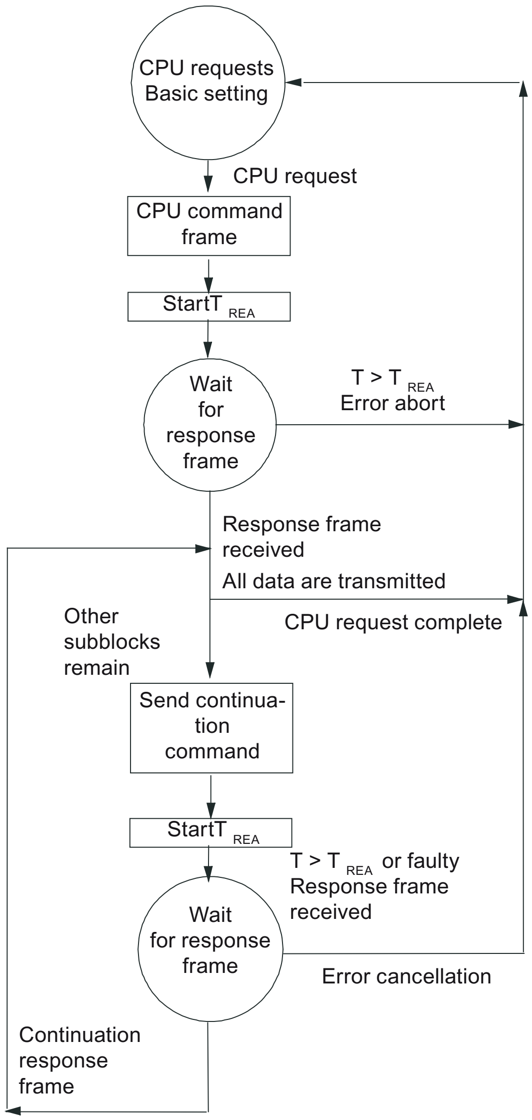
In the picture below, the sequential response message frame responding to the first SEND message frame is not sent until the partner has transmitted the SEND message frame:



6.9.3.4 RK 512 Sequence for Requests

RK 512 CPU Requests

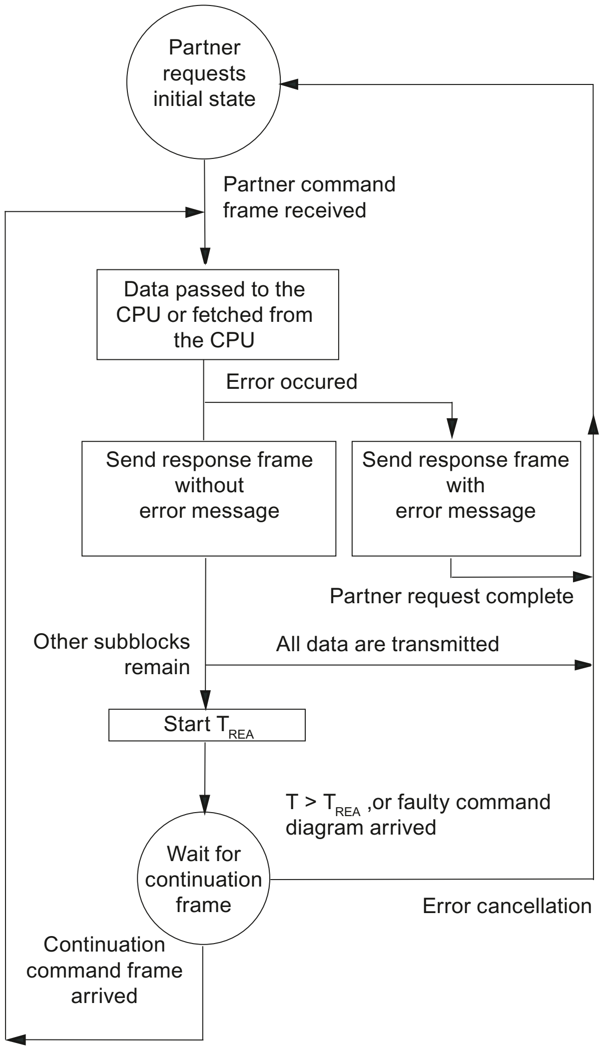
In the picture below you can find the sequences of RK 512 computer connection with CPU-requests:



Response frame monitoring time
T_{REA} = 10 s

RK 512 Partner Requests

In the picture below you can find the sequences of the RK 512 procedure initialized by partner requests:



Response frame monitoring time
T_{REA} = 10 s

6.10 Specifications

6.10.1 General Specifications

Overview

In the table below you can find general specifications.

For additional technical specifications relating to SIMATIC S7-300, refer to the *S7-300 Automation Systems, Module Data* reference manual and to the *S7-300 Automation System, Assembly* Installation Manual:

- E(lectro)M(agnetic)C(ompatibility)
- Transportation and Storage Conditions
- Mechanical and ambient climatic conditions
- Information on insulation testing, safety class and degrees of protection
- Approvals

Technical specifications	
Available protocol drivers	ASCII driver 3964(R) procedure RK 512
Transmission speed with 3964(R) procedure and RK 512	300, 600, 1200, 2400, 4800, 9600, 19200, 38400 baud
Transmission speed with ASCII driver	300, 600, 1200, 2400, 4800, 9600, 19200, 38400 (Half Duplex)
Character frame	• Bits per character (7 or 8). Only 8 characters with RK 512. • Start/stop bits (1 or 2) • Parity (no, even, odd); for 7 bits per character only "even" or "odd" configuration possible.

Specifications of the X27 (RS 422/485) Interface

The table below shows you the specifications of the X27 (RS 422/ 485) interface RS 422/485:

Technical specifications	
Interface	RS 422 or RS 485, 15-pin Sub-D socket
RS 422 signals RS 485 signals	TXD (A), RXD (A), TXD (B), RXD (B), GND R/T (A), R/T (B), GND Fully isolated from the S7 internal power supply (backplane bus) and the auxiliary 24 VDC supply
Max. transmission distance	1200 m
Max. transmission rate	38400 kbps

6.10.2 Specifications of the ASCII Driver

Overview

The table below shows you the specifications of the ASCII Driver:

ASCII driver	
Maximum message frame length	1024 Byte
Parameters	Configurable: • Transmittion rate: 300, 600, 1200, 2400, 4800, 9600, 19200 baud, 38400 baud (Half Duplex) • Character frame: 10, 11 or 12 bits • Character delay time: 1 ms to 65535 ms in 1 ms steps • Flow control: None, XON/XOFF • XON/XOFF characters (only with "Flow control" = "XON/XOFF") • Wait for XON after XOFF: 20 ms to 65530 ms in 10 ms steps • Number of message frames to be buffered: 1 to 10, utilizing the complete buffer • Prevent overwriting: Yes/No • end-of-message recognition for a received message frame: – on expiration of character delay time – on receipt of endoftext character(s) – on receipt of a fixed number of characters
ASCII driver end-of-message recognition on expiration of the character delay time	
Parameters	No further parameter assignment necessary. End-of-message is recognized when the declared character delay time expires.
ASCII driver with end-of-message recognition via configurable end-of-text characters	
Parameters	Configurable also: • Number of end-of-text characters: 1, 2 • Hex code for the first/second end code • Number of BCC characters: 1, 2
ASCII driver with end-of-message recognition via configured character length	
Parameters	Parameters can also be set for: • Character length: 1 to 1024 bytes

6.10.3 Specifications of the 3964(R) Procedure

Overview

The table below shows you the specifications of the 3964(R) procedure.

3964(R) procedure with default values	
Maximum message frame length	1024 Byte
Parameters	Configurable: • With/without block check character • Priority: low/high • Transmittion rate: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400 baud • Character frame: 10, 11 or 12 bits • Receive line default: none, R(A) 5V/R(B) 0V, R(A) 0V/R(B) 5V • Number of message frames to be buffered: 1 to 10, utilizing the complete buffer

Programmable with 3964(R) procedure	
Maximum message frame length	1024 Byte
Parameters	Configurable: • With/without block check character • Priority: low/high • Transmittion rate: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400 baud • Character frame: 10, 11 or 12 bits • Character delay time: 20 ms to 65530 ms in 10 ms steps • Acknowledgment delay time: 20 ms to 65530 ms in 10 ms steps • Number of attempts to connect: 1 to 255 • Number of attempts to transmit: 1 to 255 • Receive line default: none, R(A) 5V/R(B) 0V, R(A) 0V/R(B) 5V

6.10.4 Specifications of the RK 512 Computer Connection

Overview

The table below shows you the specifications of RK 512 computer connection:

RK 512 computer connection	
Maximum message frame length	1024 Byte
Parameters	Configurable: • Transmission rate: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400 baud • Character frame: 10, 11 or 12 bits • Character delay time: 20 ms to 65530 ms in 10 ms steps • Acknowledgment delay time: 20 ms to 65530 ms in 10 ms steps • Number of attempts to connect: 1 to 255 • Number of transmission attempts: 1 to 255 • Receive line default: – None – R(A) 5V/R(B) 0V – R(A) 0V/R(B) 5V

6.10.5 Minimum number of CPU cycles

Overview

The table below describes the minimum number of CPU cycles (SFB calls) required to process a request:

Block	Name	Number of CPU cycles for processing ...		
		Completion without error	Completion with error	RESET/RESTART
SFB 60	SEND_PTP	≥ 2	≥ 2	≥ 3
SFB 61	RCV_PTP	≥ 2	≥ 2	≥ 3
SFB 62	RES_RCVB	≥ 2	≥ 2	≥ 3
SFB 63	SEND_RK	≥ 2	≥ 2	≥ 3
SFB 64	FETCH_RK	≥ 2	≥ 2	≥ 3
SFB 65	SERVE_RK	≥ 2	≥ 2	≥ 3

6.10.6 Transmission Times

Overview

The tables below contain the measured transmission times, dependent on the communication protocol selected.

Two CPU 314C-2 PtP were interconnected for this measurement. Measured was the time expiring on the communication link between the appearance of the first character in the first message frame and the first character of the successive message frame.

With the ASCII driver measurement is based upon the fastest protocol version (end-of-message recognition with one end-of-text character and no software flow control).

Measurement was carried out with the respective default settings with the 3964(R) procedure and RK 512 computer connection, i.e. default values with BCC.

ASCII Driver (Transmission times in ms)

Transmission rate (Bd)/ for user data	38400	19200	9600	4800	2400	1200	600	300
1 Byte	5	6	7	9	13	23	41	78
10 Byte	7	11	17	28	51	97	190	376
20 Byte	11	17	28	51	97	190	374	744
50 Byte	19	34	62	120	236	465	927	1847
100 Byte	35	64	121	236	466	926	1846	3685
200 Byte	64	120	237	467	927	1845	3686	7363
500 Byte	154	298	586	1160	2309	4607	9204	13398
1000 Byte	305	591	1168	2316	4613	9210	18402	36788

3964(R) Procedure (Transmission Times in ms)

Transmission rate (Bd)/ for user data	38400	19200	9600	4800	2400	1200	600	300
1 Byte	8	11	14	22	38	71	137	267
10 Byte	11	16	25	43	80	154	302	601
20 Byte	14	22	36	66	126	246	487	966
50 Byte	23	38	71	136	264	522	1037	2071
100 Byte	38	68	130	250	494	982	1958	3907
200 Byte	67	126	246	482	956	1902	3798	7586
500 Byte	158	303	595	1175	2838	4664	9316	18620
1000 Byte	308	597	1177	2330	4642	9266	18515	37011

RK 512 Computer Connection (Transmission Times in ms)

Transmission rate (Bd)/ for user data	38400	19200	9600	4800	2400	1200	600	300
1 Byte	21	29	44	75	134	253	501	1002
10 Byte	33	42	63	101	180	337	667	1334
20 Byte	37	48	74	124	228	430	851	1701
50 Byte	48	71	112	199	368	709	1402	2804
100 Byte	70	105	178	321	605	1176	2323	4642
200 Byte	126	196	336	618	1173	2293	4543	9064
500 Byte	278	445	778	1450	2784	5450	10836	21608
1000 Byte	545	878	1554	2876	5534	10860	21571	43027

6.10.7 Cables

Introduction

When producing your own cables, take into account that you must always use shielded connector enclosures. A large-surface area of both sides of the cable shielding must be in contact with the connector' enclosure and the shielding contact.

CAUTION

Never connect the cable shielding to ground. Otherwise, the interface will be destroyed.

GND (Pin 8) must always be connected on both sides.

Otherwise, the interface might also be destroyed. The following pages show you some examples of cables for PtP connections between the CPU and S7 modules or SIMATIC S5.

Cable X 27/RS422 (CPU 31xC-CPU 31xC/CP 340/CP 341/CP 440/CP 441)

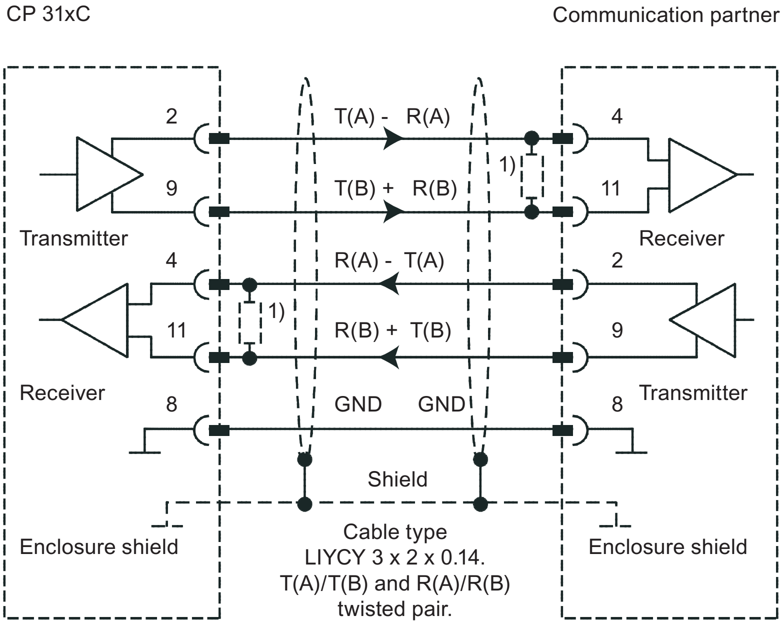
Cables are available in the default lengths: 5 m, 10 m and 50 m.

Type	Order no.
X27 (RS 422), 5 m	6ES7902-3AB00-0AA0
X27 (RS 422), 10 m	6ES7902-3AC00-0AA0
X27 (RS 422), 50 m	6ES7902-3AG00-0AA0

The picture below shows you the cable for RS422 operation between a CPU 31xC and CPU 31xC/CP 340/CP341/CP 440/CP 441.

You will require the following male connectors for the connecting cables:

- At CPU 31xC side: 15-pin Sub-D male connector with screw interlock
- At communication partner's side: 15-pin Sub-D male connector with screw interlock



1) To ensure interference-free data exchange with line lengths > 50 m, you must solder in a terminating resistance of approx. 330Ω at the receiver end.

Note

With the used type of cable the following lengths are possible:

- max. 1200 m at 19200 baud
- max. 500 m at 38400 baud

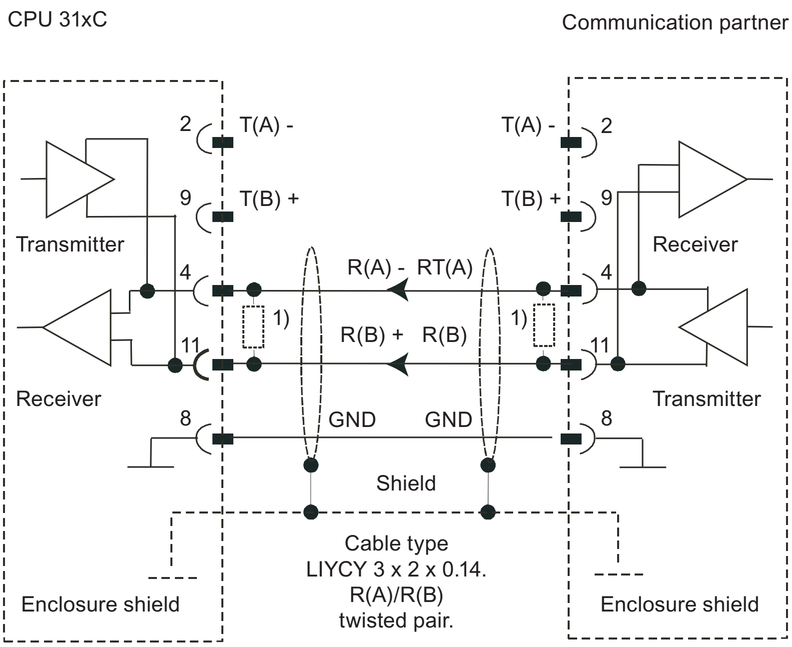
Cable X 27/RS485 (CPU 31xC – CPU 31xC/CP 340/CP 341/CP 440/CP 441)

The picture below shows you the cable for RS485 operation between a CPU 31xC and CPU 31xC/CP 340/CP 341/CP 440/CP 441.

Siemens does not offer a ready-made cable.

You will require the following male connectors for the connecting cables:

- At CPU 31xC end: 15-pin Sub-D male connector with screw interlock
- At communication partner end: 15-pin Sub-D male connector with screw interlock



[1)] To ensure interference-free data exchange with line lengths > 50 m, you must solder in a terminating resistance of approx. 330Ω at the receiver end.

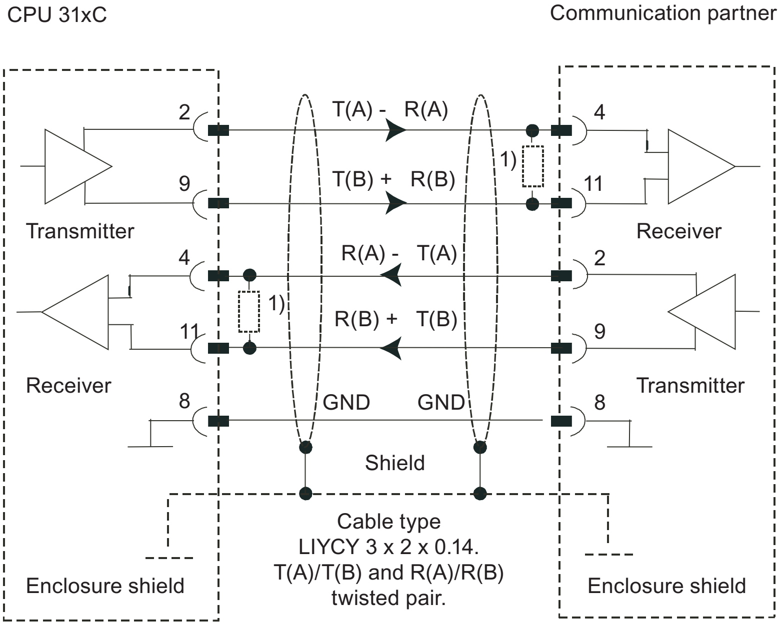
Cable X27/RS422 (CPU 31xC-CP 544, CP 524, CPU 928B, CPU 945, CPU 948)

The picture below shows you the cable for RS422 operation between a CPU 31xC and CP 544, CP 524, CPU 928B, CPU 945, CPU 948.

Siemens does not offer a ready-made cable.

You will require the following male connectors for the connecting cables:

- At CPU 31xC end: 15-pin Sub-D male connectors with screw interlock
- At communication partner end: 15-pin Sub-D male connector with screw interlock



1) To ensure interference-free data exchange with line lengths > 50 m, you must solder in a terminating resistance of approx. 330Ω at the receiver end.

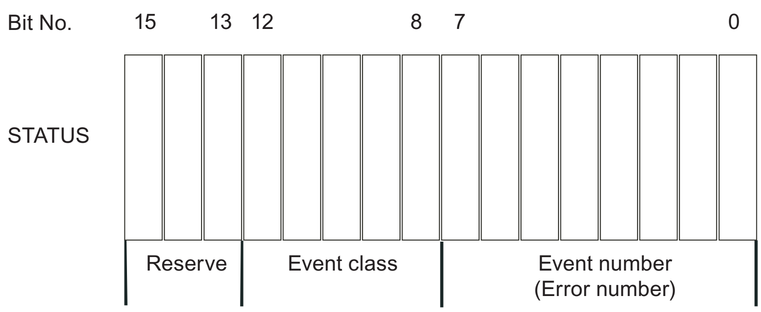
6.10.8 Error Messages

Basics

Every SFB is assigned a STATUS parameter for error diagnostics. The STATUS message IDs always have the same significance, independent on the used SFB.

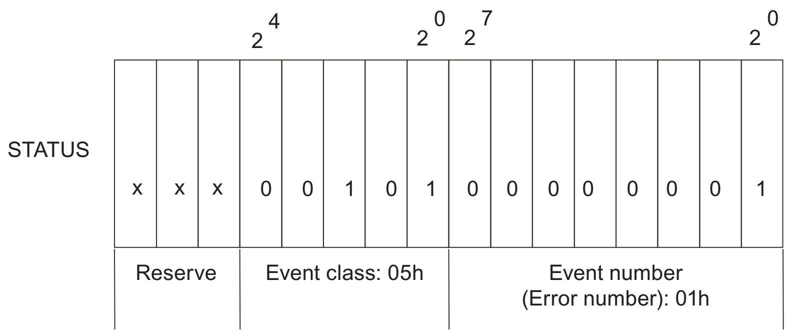
Numbering scheme event class/event number

The picture below shows the structure of the STATUS parameter:



Example

The picture below shows the content of the STATUS parameter for the event "Request canceled because of restart or reset" (Event class: 05H, event no. = 01H):



Event Classes

The tables below give you a description of the various event classes and IDs:

Event class 3(03H): "Error in the parameter assignment of the SFBs"		
Event no.	**Event**	**Remedy**
(03)01H	Illegal or missing source/target data type. Illegal range (start address, length). The DB is not permitted or does not exist (e.g. DB 0) or other illegal or missing source/target data type. Invalid byte or bit number of the interprocessor communication.	Check and, if required, correct the parameters. Partner supplying invalid parameters in message frame header. Check the parameters, create a block if required. See request tables for valid data types. Partner supplying wrong parameters in message frame header.
(03)03H	Access to area denied.	Check the parameters. Refer to the request tables for permitted start addresses and lengths. Otherwise, the partner supplies the wrong parameters in the message frame header.

Event class 5 (05H): "Job processing error"		
Event no.	**Event**	**Remedy**
(05)01H	Current request canceled due to cold restart or reset.	Repeat the canceled request. When you reassign interface parameters via the programming device, you should ensure that no more requests are active before you start the write operations.
(05)02H	Request not allowed in this operate state (e. g. device interface is not configured).	Configure the device interface.
(05)0EH	• Invalid frame length or • The declared end codes did not occur within the maximum permissible length.	• Message frame length is > 1024 bytes. Select a shorter message frame length. or • Add the end codes in the send buffer in an appropriate place.
(05)13H	Data type error (DB ...): • Data type unknown or not permitted (e.g. DE) • Mismatching source and target data types specified in the SFB.	See the job tables for permitted data types and combinations.
(05)15H	Incorrect bit number specified for coordination flag.	Permitted bit numbers: 0 to 7
(05)16H	Specified CPU number too high.	Permissible CPU no.: 0, 1, 2, 3 or 4
(05)17H	Transmission length > 1024 bytes is too high	Split up the request into multiple requests of shorter length.
(05)1DH	Send/receive request aborted due to • Communication block reset • New parameter assignment	Repeat the communication block call.
(05)22H	A new SEND request was started although the previous request is not yet completed.	Do not start the new SEND request until the previous request is closed with DONE or ERROR.

Event class 6 (06H): "Error when processing a partner request" only with RK512		
Event no.	**Event**	**Remedy**
(06)01H	Error in 1st command byte (not 00 or FFH)	Basic header structure error on partner. Check for malfunction on partner device, possibly by using an interface test device interconnected in the data link.
(06)02H	Error in 3rd command byte (not A, 0 or E)	Basic header structure error on partner. Check for malfunction on partner device, possibly by using an interface test device interconnected in the data link.
(06)03H	Error in 3rd command byte in the case of continuation message frames (command not as for 1st message frame)	Basic header structure error on partner. Check for malfunction on partner device, possibly by using an interface test device interconnected in the data link.
(06)04H	Error in 4th command byte (command letter incorrect)	Basic header structure error at the partner, or an illegal command combination was requested. Check the permissible commands. Check for malfunction on partner device, possibly by using an interface test device interconnected in the data link.
(06)06H	Error in 5th command byte (DB number not permissible)	Refer to the request tables for permissible DB numbers, start addresses or lengths.
(06)07H	Error in 5th or 6th command byte (start address too high)	Refer to the request tables for permissible DB numbers, start addresses or lengths.
(06)09H	Error in 9th and 10th command bytes (illegal coordination flag for this data type or bit number too high).	Basic header structure error on partner. Find out from the request tables when a coordination flag is permitted.
(06)0AH	Error in the 10th command byte (illegal CPU number)	Basic header structure error on partner.

Event class 7 (07H): "Send error"		
Event no.	**Event**	**Remedy**
(07)01H	With 3964(R) only: Send the first repetition: • An error was detected when transmitting the message frame, or • The partner requested a repetition by means of a negative acknowledgment code (NAK).	A repetition is not an error, but it can be an indication of transmission line disturbances or malfunction of the partner device. If the message frame still has not been transmitted after the maximum number of repetitions, an error number is output that describes the first error that occurred.
(07)02H	With 3964(R) only: Connect error: After STX was sent, NAK or any other code (except for DLE or STX) was received.	Check for malfunctioning of the partner device, possibly using an interface test device switched into the transmission line.
(07)03H	With 3964(R) only: • Acknowledgment delay time exceeded: • After STX was sent, partner did not respond within the acknowledgment delay time.	The partner device is too slow or not ready to receive, or, for example, there is a break in the send line. Check for malfunctioning of the partner device, possibly by using an interface test device interconnected in the data link.

Event class 7 (07H): "Send error"		
Event no.	**Event**	**Remedy**
(07)04H	With 3964(R) only: Cancellation at partner: During the current send operation, one or more characters were received from the partner.	Check whether the partner also indicates errors, perhaps because not all sent data arrived (e.g. break in the send line), fatal errors are pending, or the partner device has malfunctioned. Prove this, if required, using an interface tester interconnected in the data link.
(07)05H	With 3964(R) only: Negative acknowledgment when sending	Check whether the partner also indicates errors, perhaps because not all sent data was received, (e. g. break of the send line) or fatal errors are pending, or malfunctioning of the partner device. Prove this, if required, using an interface tester interconnected in the data link.
(07)06H	With 3964(R) only: End-of-connection error: • Partner rejected message frame at end of connection with NAK or a random string (except for DLE), or • Acknowledgment characters (DLE) received too early.	Check whether the partner also indicates errors, perhaps because not all sent data arrived (e.g. break in the send line), fatal errors are pending, or the partner device has malfunctioned. Prove this, if required, using an interface tester interconnected in the data link.
(07)07H	With 3964(R) only: Acknowledgment delay exceeded at end of connection or reply monitoring time exceeded after a send message frame: After connection release with DLE ETX, no response received from partner within acknowledgment delay time.	Partner device faulty or too slow. Prove this, if required, using an interface tester interconnected in the data link.
(07)08H	With ASCII driver only: Waiting time for XON has expired.	The communication partner has a fault, is too slow or is switched off-line. Check the communication partner or, if necessary, change the parameter assignment.
(07)09H	With 3964(R) only: Not possible to connect. Number of permitted requests to connect exceeded.	Check the interface cable or the transmission parameters. Also check whether the receive function between CPU and CP is configured correctly at the partner device.
(07)0AH	With 3964(R) only: Data could not be transmitted. The permitted number of attempts to transfer was exceeded.	Check the interface cable or the transmission parameters.
(07)0BH	With 3964(R) only: Initialization conflict cannot be resolved because both partners have high priority.	Change the parameter assignment.
(07)0CH	With 3964(R) only: Initialization conflict cannot be resolved because both partners have low priority.	Change the parameter assignment.

Event class 8 (08H): "Receive error"		
Event no.	**Event**	**Remedy**
(08)01H	Only for 3964(R): Expectation of the first repetition: An error was detected when the message frame was received. The CPU has requested a repetition from the partner via negative acknowledgment (NAK).	A repetition is not an error, but it can be an indication of transmission line disturbances or malfunction of the partner device. If the message frame still has not been transmitted after the maximum number of repetitions, an error number is output that describes the first error that occurred.
(08)02H	Only for 3964(R): Connect error: • In idle mode, one or more random codes (other than NAK or STX) were received, or • after an STX was received, partner sent more codes without waiting for response DLE. After partner power ON: • the CPU receives an undefined character while the partner is switched on.	Identify the malfunction on the partner device, possibly by using an interface test device interconnected in the data link.
(08)05H	Only for 3964(R): Logical error during receiving: After DLE was received, a further random code (other than DLE or ETX) was received.	Check whether the partner always duplicates the DLE in the message frame header and data string or the connection goes down via DLE ETX. Identify the malfunction on the partner device, possibly by using an interface test device interconnected in the data link.
(08)06H	Character delay time exceeded: • Two successive characters were not received within character delay time, or Only for 3964(R): • First character after sending of DLE when establishing the connection was not received within character delay time.	Partner device faulty or too slow. Prove this, if required, using an interface tester interconnected in the data link.
(08)07H	Illegal message frame length: A zero-length message frame was received.	Receipt of a zero-length message frame is not an error. Check why the communication partner is sending message frames without user data.
(08)08H	Only for 3964(R): Error in block check character (BCC): Internally calculated value of BCC does not match BCC received by partner at end of connection.	Check whether the connection is seriously disrupted; in this case you may also occasionally see error codes. Identify the malfunction on the partner device, possibly by using an interface test device interconnected in the data link.
(08)09H	Only for 3964(R): Waiting time for block repetition has expired	Declare a block waiting time at the communication partner identical to that in your module. Check for malfunctioning of the communication partner, possibly by using an interface tester interconnected in the data link.
(08)0AH	There is no free receive buffer available: No receive buffer space available for receiving data.	The SFB RCV must be called more frequently.

Event class 8 (08H): "Receive error"		
Event no.	**Event**	**Remedy**
(08)0CH	Transmission error: • A transmission error (parity/stop bit/overflow error) was detected. Only for 3964(R): • If a corrupted character is received in idle mode, the error is reported immediately so that disturbances on the transmission line can be detected early. Only for 3964(R): • If this happens during send or receive operations repetition is started.	Disturbances on the transmission line cause message frame repetitions, thus lowering user data throughput. The risk of undetected error increases. Change your system setup or cable wiring. Check the connecting cables of the communication partners or verify that the two devices have matching settings for the baud rate, parity and number of stop bits.
(08)0DH	BREAK: Receive line to partner is interrupted.	Reconnect or switch on partner.
(08)0EH	Receive buffer overflow with disabled flow control.	The receive SFB must be called more frequently in the user program, or communication must be configured with flow control.
(08)10H	Parity error	Check the connecting cables of the communication partners or verify that the two devices have matching settings for the baud rate, parity and number of stop bits.
(08)11H	Character frame error	Check the connecting cables of the communication partners or verify that the two devices have matching settings for the baud rate, parity and number of stop bits. Change your system setup or cable wiring.
(08)12H	With ASCII driver only: More characters were received after the CPU transmitted XOFF.	Reconfigure the communication partner or dispose of data more rapidly.
(08)14H	With ASCII driver only: A message frame or several message frames have got lost, because you have been working without flow control.	Work with flow control as much as you can. Use the entire receive buffer. In your basic parameters, set the "Reaction to CPU STOP" parameter to "Continue".
(08)16H	The length of a received message frame was longer than the maximum specified length.	Correction required at the partner.

Event class 9 (09H): "Response message frame received from interconnection partner with error or error message frame"		
Event no.	**Event**	**Remedy**
(09)02H	Only with RK 512: Memory access error on partner (memory does not exist) With SIMATIC S5 as partner: • Incorrect area at status word, or • Data area does not exist (except DB/DX), or • Data area too short (except DB/DX)	Check that the partner has the desired data area and that it is big enough, or check the parameters of the called system function block. Check the length specified in the system function block.
(09)03H	Only with RK 512: DB/DX access error at the partner (DB/DX does not exist or is too short) With SIMATIC S5 as partner: • DB/DX does not exist, or • DB/DX too short, or • Illegal DB/DX number. Permissible source range exceeded with FETCH request.	Check whether the desired data area • exists on the partner and whether it is of sufficient. • size, or check the parameters of the called SFB. • Check the length specified in the system function block.
(09)04H	Only with RK 512: Partner reports "Illegal request type".	Faulty partner behavior, because the CPU never outputs a system command.
(09)05H	Only with RK 512: Error at the partner or SIMATIC S5 as partner: • Illegal type of source/target or • memory error at the partner station or • CP/CPU communication error at the partner, or • the partner station is in STOP state	• Check whether the partner can transmit the requested data type. • Check the structure of the partner hardware. • Set the mode selector switch of the partner station to RUN.
(09)08H	Only with RK 512: Partner detects synchronous error: Message frame sequence error.	This error occurs at the restart of your own programmable controller or of the partner. This represents normal system startup behavior. You do not need to correct anything. The error is also conceivable during operation as a consequence of previous errors. Otherwise, you can assume a malfunction of the partner device.
(09)09H	Only with RK 512: DB/DX locked at the partner by coordination flag.	• In the partner program: Reset the coordination memory after the last transmission data was processed! • In the program: Repeat the request!
(09)0AH	Only with RK 512: Errors in message frame header that are detected by the partner: 3. command byte in the header is wrong	Check whether the error is the result of a disturbance or partner malfunction. Prove this using an interface test device interconnected in the data link.
(09)0CH	Only with RK 512: Partner detects incorrect message frame length (total length).	Check whether the error is the result of a disturbance or partner malfunction. Prove this using an interface test device interconnected in the data link.

Event class 9 (09H): "Response message frame received from interconnection partner with error or error message frame"		
Event no.	**Event**	**Remedy**
(09)0DH	Only with RK 512: Partner was not yet restarted.	Restart the partner station or set the mode selector switch to RUN.
(09)0EH	Only with RK 512: Unknown error number received in the response message frame.	Check whether the error is the result of a disturbance or partner malfunction. Prove this using an interface test device interconnected in the data link.

Event class 10 (0AH): "Errors in the response message frame of the partner that have been detected by the CPU"		
Event no.	**Event**	**Remedy**
(0A)02H	Only with RK 512: Error in the structure of the received response message frame (1st byte not 00H or FFH)	Check for malfunction of the partner device, possibly by using interface test device interconnected in the data link.
(0A)03H	Only with RK 512: Received response message frame has too many or too few data.	Check for malfunction of the partner device, possibly by using interface test device interconnected in the data link.
(0A)05H	Only with RK 512: No response message frame received from partner within monitoring time.	Is the partner a slow device? This error is also often displayed as a consequence of a previous error. For example, procedure receive errors (event class 8) can be displayed after a FETCH message frame was sent. Reason: The response message frame could not be received because of disturbances. The monitoring time expires. This error also possibly occurs if the partner is restarted before it could respond to the last received FETCH message frame.

Event class 11 (0BH): "Warnings"		
Event no.	**Event**	**Remedy**
(0B)01H	Receive buffer filled with more than 2/3 of its capacity	Call the receive block more often to avoid receive buffer overflow.

6.10.9 Parameters of the SFBs

Parameters of SFB 60 "SEND_PTP"

Parameter	Declaration	Data type	Description	Value range	Default
REQ	IN	BOOL	Initiates request at positive edge	TRUE/FALSE	FALSE
R	IN	BOOL	Cancels the request. Sending is locked.	TRUE/FALSE	FALSE
LADDR	IN	WORD	Submodule I/O address you specified in "HW Config".	CPU-specific	3FF hex
DONE	OUT	BOOL	Job completed without errors	TRUE/FALSE	FALSE
ERROR	OUT	BOOL	Job completed with errors	TRUE/FALSE	FALSE
STATUS	OUT	WORD	Error number	0 to FFFF hex	0
SD_1	IN_OUT	ANY	Send parameters: Here you specify: • The number of the DB from which the data are sent. • The data byte number, starting from which data will be transmitted. e.g.: DB 10 from byte 2 → DB10.DBB2	CPU-specific	0
LEN	IN_OUT	INT	Here you specify the byte length of the data block to be transmitted	1 to 1024	1

Parameters of the SFB 61 "RCV_PTP"

Parameter	Declaration	Data type	Description	Value range	Default
EN_R	IN	BOOL	Receive enable	TRUE/FALSE	FALSE
R	IN	BOOL	Cancels the request	TRUE/FALSE	FALSE
LADDR	IN	WORD	Submodule I/O address you specified in "HW Config".	CPU-specific	3FF hex
NDR	OUT	BOOL	Job completed without errors	TRUE/FALSE	FALSE
ERROR	OUT	BOOL	Job completed with errors	TRUE/FALSE	FALSE
STATUS	OUT	WORD	Error number	0 to FFFF hex	0
RD_1	IN_OUT	ANY	Receive parameters: Here you specify: • Number of the DB in which the received data are to be stored. • The data byte number as of which received data are to be stored. e. g.: DB 20 from byte 5 → DB20.DBB5	CPU-specific	0
LEN	IN_OUT	INT	Output of the data length (number of bytes)	0 to 1024	0

Parameters of SFB 62 "RES_RCVB"

Parameter	Declaration	Data type	Description	Value range	Default
REQ	IN	BOOL	Initiates job on positive edge	TRUE/FALSE	FALSE
R	IN	BOOL	Cancels the request	TRUE/FALSE	FALSE
LADDR	IN	WORD	Submodule I/O address you specified in "HW Config".	CPU-specific	3FF hex
DONE	OUT	BOOL	Job completed without errors	TRUE/FALSE	FALSE
ERROR	OUT	BOOL	Job completed with error	TRUE/FALSE	FALSE
STATUS	OUT	WORD	Error number	0 to FFFF hex	0

Parameters of SFB 63 "SEND_RK"

Parameters	Declaration	Data Type	Description	Range of values	Default
SYNC_DB	IN	INT	Number of the DB in which the common data for the synchronization of the RK SFBs is stored (minimum length is 240 bytes).	CPU-specific, zero is not allowed.	0
REQ	IN	BOOL	Initiates job on positive edge	TRUE/FALSE	FALSE
R	IN	BOOL	Job is aborted. Sending is blocked.	TRUE/FALSE	FALSE
LADDR	IN	WORD	Submodule I/O address you specified in "HW Config".	CPU-specific	3FF hex
R_CPU	IN	INT	CPU number of the partner CPU (only for multiprocessor operation)	0 to 4	1
R_TYPE	IN	CHAR	Address type at partner CPU • 'D' = Data block • 'X' = Expanded date block	'D', 'X'	'D'
R_DBNO	IN	INT	Data block number at partner CPU	0 to 255	0
R_OFFSET	IN	INT	Data byte number at partner CPU	0 to 510 (only even values)	0
R_CF_BYT	IN	INT	Interprocessor communication flag byte on partner CPU (255: Significance: without interprocessor communication flag)	0 to 255	255
R_CF_BIT	IN	INT	Interprocessor communication flag bit on the partner CPU	0 to 7	0
DONE	OUT	BOOL	Job completed without errors	TRUE/FALSE	FALSE
ERROR	OUT	BOOL	Job completed with error	TRUE/FALSE	FALSE
STATUS	OUT	WORD	Error number	0 to FFFF hex	0

Parameters	Declaration	Data Type	Description	Range of values	Default
SD_1	IN_OUT	ANY	Send parameters: Here you specify: • The number of the DB from which the data are sent. • The data byte number from which the data is to be sent. e.g.: DB 10 from byte 2 → DB 10.DBB2	CPU-specific	0
LEN	IN_OUT	INT	Here you specify the byte length of the data block to be transmitted	1 to 1024	1

Parameters of SFB 64 "FETCH_RK"

Parameters	Declaration	Data type	Description	Value range	Default
SYNC_DB	IN	INT	Number of the DB in which the common data for the synchronization of the RK SFBs is stored (minimum length is 240 bytes).	CPU-specific, zero is not allowed.	0
REQ	IN	BOOL	Initiates job on positive edge	TRUE/FALSE	FALSE
R	IN	BOOL	Job is aborted.	TRUE/FALSE	FALSE
LADDR	IN	WORD	Submodule I/O address you specified in "HW Config".	CPU-specific	3FF hex
R_CPU	IN	INT	CPU no. of the partner CPU (only for multiprocessor mode)	0 to 4	1
R_TYPE	IN	CHAR	Address type on the partner CPU • 'D' = Data block • 'X' = Expanded date block • 'M' = Memory bit • 'E' = Inputs • 'A' = Outputs • 'Z' = Counter • 'T' = Timer	'D', 'X', 'M', 'E', 'A', 'Z', 'T'	'D'
R_DBNO	IN	INT	Data block number on the partner CPU	0 to 255	0
R_OFFSET	IN	INT	Data byte number on the partner CPU	See the Table: "Parameter in the FB for data source (Partner CPU)"	0
R_CF_BYT	IN	INT	Interprocessor communication flag byte on partner CPU (255: means: no interprocessor communication flag)	0 to 255	255
R_CF_BIT	IN	INT	Interprocessor communication flag bit on partner CPU	0 to 7	0
DONE	OUT	BOOL	Job completed without errors	TRUE/FALSE	FALSE

Parameters	Declaration	Data type	Description	Value range	Default
ERROR	OUT	BOOL	Job completed with error	TRUE/FALSE	FALSE
STATUS	OUT	WORD	Error number	0 to FFFF hex	0
RD_1	IN_OUT	ANY	Receive parameter: Here you specify: • Number of the DB in which the fetched data are stored. • Data byte number as of which the fetched data are stored. e.g.: DB 10 from byte 2 → DB10.DBB2	CPU-specific	0
LEN	IN_OUT	INT	Here you specify the byte length of the message frame to be fetched. A length of two bytes must be declared per time and counter.	1 to 1024	1

Parameters of SFB 65 "SERVE_RK" for Receiving/Providing Data

Parameters	Declaration	Data type	Description	Value range	Default
SYNC_DB	IN	INT	Number of the DB in which the common data for the synchronization of the RK SFBs is stored (minimum length is 240 bytes).	CPU-specific, zero is not allowed.	0
EN_R	IN	BOOL	Request enable	TRUE/FALSE	FALSE
R	IN	BOOL	Cancels the request	TRUE/FALSE	FALSE
LADDR	IN	WORD	Submodule I/O address you specified in "HW Config".	CPU-specific	3FF hex
L_TYPE	OUT	CHAR	Receiving data: Type of target area on local CPU (capital letters only): • 'D' = Data block	'D'	' '
			Providing data: Type of the source area at local CPU (capital letters only): • 'D' = Data block • 'M' = Memory bit • 'E' = Inputs • 'A' = Outputs • 'Z' = Counters • 'T' = Timers	'D', 'M', 'E', 'A', 'Z', 'T'	
L_DBNO	OUT	INT	Data block number on local CPU (destination)	CPU-specific, zero is not allowed.	0
L_OFFSET	OUT	INT	Data byte number at local CPU (destination)	0-510	0

Parameters	Declaration	Data type	Description	Value range	Default
L_CF_BYT	OUT	INT	Interprocessor communication flag byte at local CPU (255: means: no interprocessor communication flag)	0 to 255	0
L_CF_BIT	OUT	INT	Interprocessor communication flag bit at local CPU	0 to 7	0
NDR	OUT	BOOL	Job completed without errors	TRUE/FALSE	FALSE
ERROR	OUT	BOOL	Job completed with errors	TRUE/FALSE	FALSE
STATUS	OUT	WORD	Error number	0 to FFFF hex	0
LEN	IN_OUT	INT	Message frame length in bytes	0 to 1024	0

Controlling 7

7.1 Overview

7.1.1 Concept of Integrated Controlling

Overview

The following SFBs are available to you for controlling with the CPU 313C, CPU 313C-2 DP, PtP and CPU 314C-2 DP, PN/DP, PTP:

- SFB 41 for continuous control (CONT_C)
- SFB 42 for step control (CONT_S)
- SFB 43 for pulse width modulation (PULSEGEN).

The SFBs are compatible to FBs 41 through 43. This software control block solution offers full controller functionality in every block. Data required for periodic calculation are stored in the assigned DBs (Instance DBs). This allows you multiple calls of the SFBs. SFB PULSEGEN in combination with SFB CONT_C used to obtain a controller with pulse output for proportional actuators (e. g. for heating and cooling aggregates).

Basic Functions

A controller created with the help of SFBs consists of a number of units you can configure. Functions for preparing setpoint and actual values as well as for post-processing the calculated manipulated variable are integrated, in addition to the actual controller with its PID algorithm.

Applications

Principally, the controls created with the two control blocks are neutral related to the field of application. The controlling efficiency, and therefore processing speed, depends exclusively on the performance of the CPU you are using. With the given CPU, a compromise must be found between the number of controllers and the processing frequency required for the controllers. The faster the connected control circuits are, that is, the more often the manipulated values have to be calculated per time unit, the lower the number of installable controllers. There are no restrictions with regard to the type of controllable processes. Slow (temperatures, filling levels etc.) as well as fast controlling systems (flow, speed etc.) can be controlled.

Control System Analysis

The static response (gain) and dynamic properties (lag, dead time, integration constant etc.) of the controlling system are a decisive factor for the layout and design of the controller and for the dimensions of its static (P action) and dynamic (I and D action) parameters.

Precise knowledge about the type and characteristics of the controlling system is therefore imperative.

The optional software package "PID Self Tuner" is available to facilitate control optimization.

Controller Selection

The properties of controlling systems are given by technical process/machine circumstances and can hardly be influenced. Therefore, to achieve good controlling results and to adapt it to the recovery behavior of the system you must select the most suitable type of system controller.

Creating the Controls

You can create the controls, ranging from the structure and parameter assignment to the time-oriented call by the system program, widely without programming effort. However, STEP 7 know-how is required.

Online Help

The STEP 7 Online Help also offers you information on the respective SFBs.

Additional Information

The integrated control is part of the standard control. For additional information on the topic 'standard controls' refer to:

- "Standard PID Control": SIMATIC S7 Manual and configuration package with ready-to-use controller structures and comfortable parameter assignment screens.
- "Modular PID Control": SIMATIC S7 Manual and configuration package including a flexible controller kit that is also suitable for complex tasks.
- "Controlling with SIMATIC", by Jürgen Müller: Practical manual for controlling with SIMATIC S7 and SIMATIC PCS7
- "PID Self Tuner". SIMATIC S7 Manual and software package for Online self-optimization of PID controllers
- FM 355/FM 455 is a self-sufficient back-up controller module that does not impose a load on the CPU.

7.1.2 Basics

Continuous/Switching Controller

The continuous controller outputs a linear (analog) value.

The switching controller outputs a binary (digital) value.

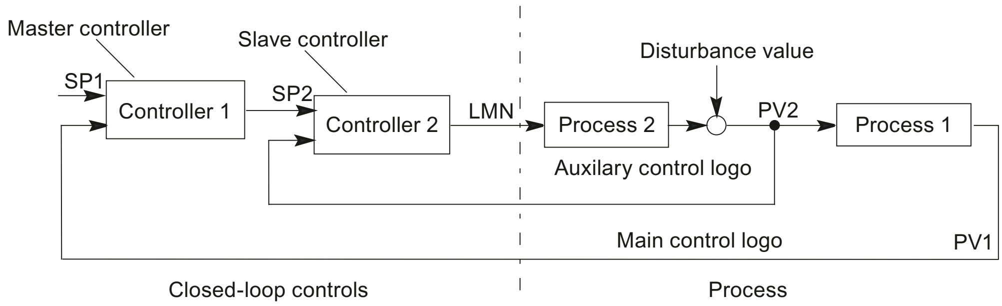
Set-value control

Set-value control is considered as control with a fixed, only occasionally modified reference variable. This regulates deviations in the process.

Cascade control

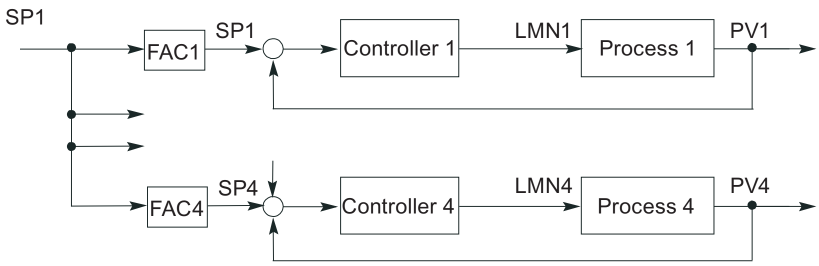
The cascade control represents a series connection of controllers. The first controller (master controller) determines the setpoint for the series (slave) controllers or influences their setpoints according to the actual error of the process variable.

The controlling performance of a cascade control can be improved with additional process variables. For this purpose, an auxiliary process variable PV2, acquired from a suitable point, is blended to the master process variable (output of master controller SP2). The master controller applies the process variable PV1 to the setpoint SP1 and adjusts SP2 in such a way that the target is reached as fast as possible and without overshooting.



Blending control

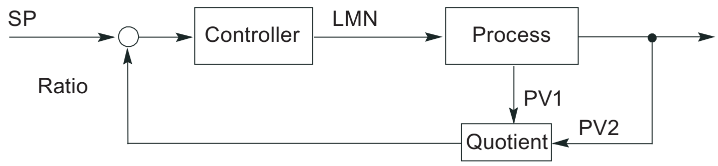
The blending control represents a control structure in which the setpoint for the total quantity SP is calculated as percentage for the desired quantities of the individual controlled components. Here, the sum of the blending factors FAC must be 1.



Ratio control

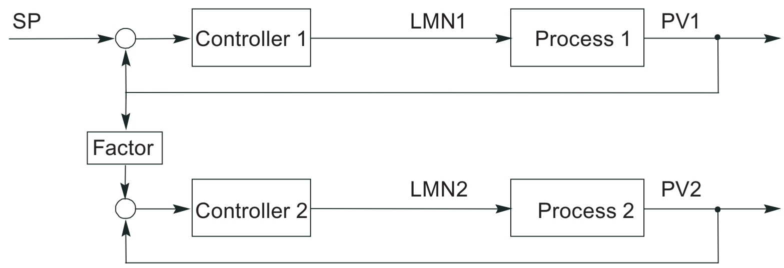
- Single Loop Ratio Controller

  A single loop ratio control, for example, is used in cases where the ratio between two process variables is more important than the absolute values of the process variable (e. g. speed control).



- Multiple Loop Ratio Controller

  With multiple loop ratio controlling the ratio between the two process variables PV1 and PV2 is held constant. Here, the setpoint for the 2nd control loop is calculated with the process value of the 1st control loop. With dynamic changes of the process variable x1 it is also ensured that the specified ratio is maintained.



Two-step control

A two-step control can only acquire two output states (e. g. On-Off). A typical control is pulse width modulation for a heating system via relay output.

Three-step control

The three-step control can acquire only three discrete output states. Here we must differentiate between pulse width modulation, e. g. heating and cooling (Heating-Off-Cooling) and step controlling with integrated actuators (e. g. Right-Stop-Left).

7.2 Wiring

7.2.1 Wiring Rules

Basics

Integrated I/Os do not exist for the controller. For the input and output you must use the free I/Os of the CPU or add-on I/O modules.

Connecting cables

- The cables for the digital I/O must be shielded if their length exceeds 100 m.
- The cable shielding must be terminated on both ends.
- Flexible cable, cross-sections 0.25 to 1.5 mm^2.
- Cable sleeves are not required. Should you still decide to do so, use cable sleeves without insulating collar (DIN 46228, Shape A, short version).

Shielding termination element

You can use the Shielding Termination Element to connect all shielded cables directly to ground via the profile rail.

WARNING

Risk of harm to human beings and damage to assets if the voltage is not switched off.

If you wire the front plug on the live module you run the risk of injury as a result of electrical current!

Always wire the module off-voltage state!

Additional Information

For additional information refer to the *CPU Data* manual and to the installation instructions for your CPU.

7.3 Parameter configuration

7.3.1 Configuring SFBs with Parameter Assignment Screens

Parameter assignment screens

You set the default parameters (instance DB) for SFBs 41, 42 and 43 **using the parameter assignment screens** "PID Control".

These parameter assignment screens are largely self-explanatory. You can find the description of the parameters in section Continuous Controlling with SFB 41 "CONT_C" (Page 372) and in the integrated help on parameter assignment screens.

Procedure

Requirement: The SFB has already been inserted in the S7 program with the instance DB. The SFBs are found in the "Standard library" under "System Function Blocks".

1. Start the parameter assignment screens with "SIMATIC/STEP7/PID Control Parameter Assignment".
2. Open your project with "File > Open" under PID-Control and select your instance DB.
3. Set your parameters.
4. Save the parameters (located in the instance DB) and load the program to your CPU.

Integrated help

The integrated help on parameter assignment screens offers you support when you assign parameters. You can call up the integrated help in one of the following ways:

- Using the menu command "Help > Help topics ..."
- By pressing **F1** in the respective areas

7.4 Implementing Controlling in the User Program

Overview

In the table below you can find an overview of the controlling functions of the module and of their assigned SFBs:

Function	SFB
Continuous controller	SFB CONT_C (SFB 41)
Step controller	SFB CONT_S (SFB 42)
Pulse width modulation	SFB PULSEGEN (SFB 43)

The SFBs are found in the "Standard library" under "System Function Blocks".

The following sections help you to design a user program for your application.

Calling the SFB

Call the SFB with a corresponding instance DB.

Example: CALL SFB 41, DB 30

Instance DB

The SFB parameters are stored in the instance DB. These parameter are described in Section Continuous Controlling with SFB 41 "CONT_C" (Page 372).

You can access the parameters via:

- DB number and offset address
- DB number and symbolic address in the DB

Program Structure

The SFBs must be called in the restart OBs and in the time-out interrupt OBs. Scheme:

OB100	Call SFB 41, 42, 43
OB35	Call FB 41, 42, 43

7.5 Description of Functions

7.5.1 Continuous Controlling with SFB 41 "CONT_C"

Introduction

The SFB "CONT_C" (*continuous controller*) serves for controlling technical processes with continuous I/O variables on SIMATIC S7 automation system. You can switch partial actions of the PID controller on or off via parameters, thus adapting it to the controlled system. You can easily do this by using the parameter assignment screen (Menu path: "Start > Simatic > STEP 7 > Assign PID control parameters"). The online electronic manual is found under "Start > Simatic > S7 Manuals > PID Control English".

Application

You can use the controller as a single PID fixed setpoint controller or in multiple control loops as cascade, blending or ratio controls. The functions of the controller are based on the PID control algorithm of the sampling controller with an analog output signal, if necessary including a pulse shaper stage to generate pulse-width modulated output signals for two or three step controllers with proportional actuators.

Description

Apart from the functions in the setpoint and process variable channels, the SFB/FB implements a complete PID controller with continuous manipulated variable output and the option of influencing the manipulated value manually.

Below you will find a detailed description of the subfunctions:

Setpoint operation
The setpoint is entered in floating-point format at the **SP_INT** input.

Process value channel
The actual value can be input in the peripheral (I/O) or floating-point format. The CRP_IN function converts the PV_PER I/O value to a floating-point format of -100 to +100 % in accordance with the following formula:

$$\text{Output of}\quad \text{CPR\_IN} = \quad \text{PV\_PER x}\ \frac{100}{27648}$$

The PV_NORM function standardizes the output of CRP_IN according to the following formula:

Output of PV_NORM = (output of CPR_IN) x PV_FAC + PV_OFF

PV_FAC has a default of 1 and PV_OFF a default of 0.

The variables PV_FAC and PV_OFF are the result of formula conversion as follows:

PV_OFF = (output of PV_NORM) - (output of CPR_IN) x PV_FAC

$$\text{PV\_FAC} = \frac{\text{(output of PV\_NORM) - PV\_OFF}}{\text{Output of CPR\_IN}}$$

Conversion to percentage value is not imperative. If the setpoint is to be determined physically, the actual value can also be converted to this physical value.

Calculating the negative deviation
The difference between setpoint and actual value forms the negative deviation. For the suppression of minor continuous oscillation as a result of manipulated variable quantization (e.g. for pulse width modulation with PULSEGEN), the error signal is applied to a dead band (DEADBAND). If DEADB_W = 0, the dead band is switched off.

PID algorithm
The PID algorithm operates as a position algorithm. The proportional, integral (INT), and derivative (DIF) actions are connected in parallel and can be activated or deactivated individually. This allows P, PI, PD, and PID controllers to be configured. However, stand-alone I-controllers or D controllers are also possible.

Manual mode
It is possible to switch over between manual and automatic mode. In the manual mode, the manipulated variable is corrected to a manually selected value.

The integrator (INT) is set internally to LMN-LMN_P-DISV and the derivative unit (DIF) to 0 and matched internally. This means that a switchover to the automatic mode does not cause any sudden change in the manipulated value.

Manipulated value processing
The manipulated value can be limited to a selected value using the LMNLIMIT function. Signaling bits indicate when a limit is exceeded by the input variable.

The LMN_NORM function normalizes the output of LMNLIMIT according to the following formula:

LMN = (output of LMNLIMIT) x LMN_FAC + LMN_OFF

Default for LMN_FAC is 1, for LMN_OFF it is 0.

The manipulated value is also available in peripheral (I/O) format. The CPR_OUT function converts the floating-point value LMN to a peripheral value according to the following formula:

$$\text{LMN\_PER} = \text{LMN} \times \frac{2764}{100}$$

Error Value Blending
Error Value Blending A disturbance variable can be fed forward at the DISV input.

Initialization

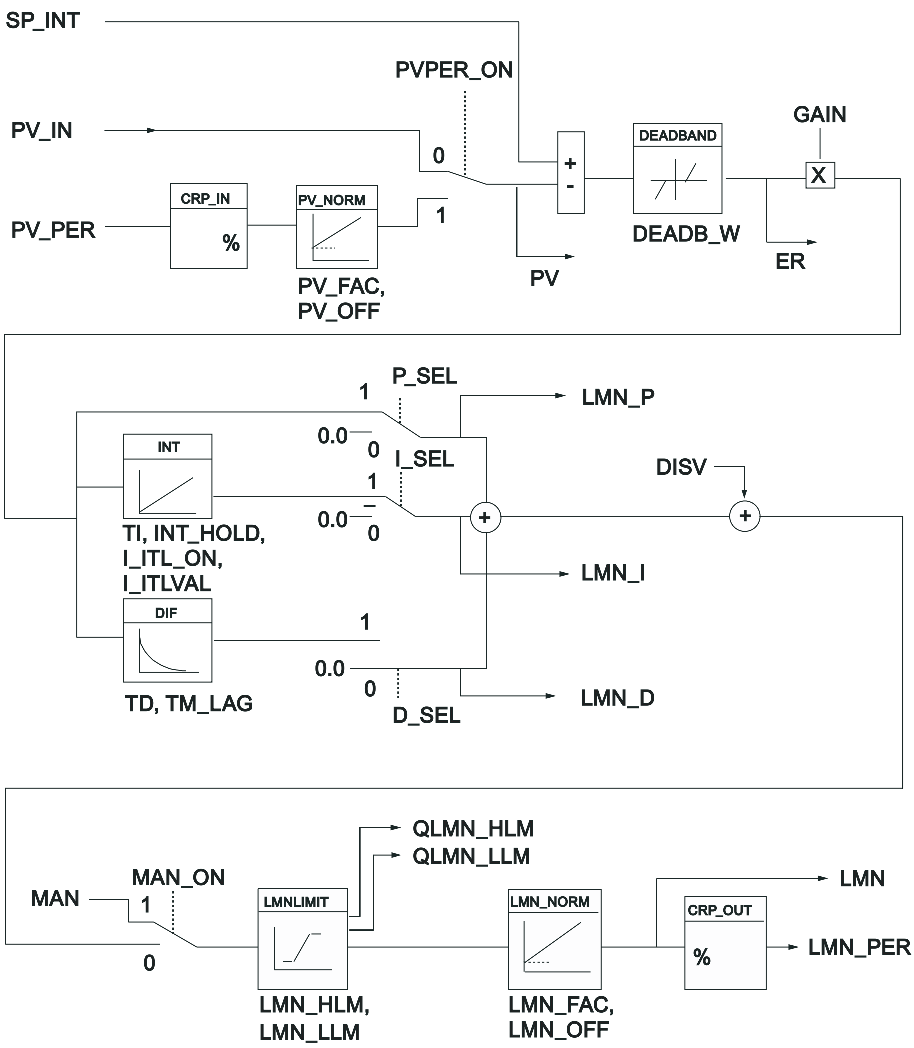
The SFB "CONT_C" has an initialization routine that is run through when the input parameter COM_RST = TRUE is set.

During initialization, the integrator is set internally to the initialization value I_ITVAL. When it is called in a cyclic interrupt priority class, it then continues to work starting at this value. All other outputs are set to their default values.

Error Information

A parameter check is carried out via the Parameter Assignment Tool.

Block Diagram CONT_C



Parameters of SFB 41

The table below contains the **input parameters** of SFB 41 "CONT_C":

Parameters	Data type	Address (Instance DB)	Description	Value range	Default
COM_RST	BOOL	0.0	COMPLETE RESTART The block has an initialization routine that is processed when the input COM_RST is set.	TRUE: restart FALSE: controller operation	FALSE
MAN_ON	BOOL	0.1	MANUAL VALUE ON If the input "manual value on" is set, the control loop is interrupted. A manual value is set as the manipulated value.		TRUE
PVPER_ON	BOOL	0.2	PROCESS VARIABLE PERIPHERY ON If the If the process variable is read from the I/Os, input PV_PER must be connected to the peripherals and input "PROCESS VARIABLE PERIPHERY ON" must be set.		FALSE
P_SEL	BOOL	0.3	PROPORTIONAL ACTION ON The PID algorithm allows On/Off switching of individual PID-actions. The P-action is on when the input "proportional action on" is set.		TRUE
I_SEL	BOOL	0.4	INTEGRAL ACTION ON The PID algorithm allows On/Off switching of individual PID-actions. The I-action is on when the input "integral action on" is set.		TRUE
INT_HOLD	BOOL	0.5	INTEGRAL ACTION HOLD The integrator output can be frozen. To do this, input "Integral Action Hold" must be set.		FALSE
I_ITL_ON	BOOL	0.6	INITIALIZATION OF THE INTEGRAL ACTION The output of the integrator can be set at input I_ITLVAL. To do this, input "Initialization of the integral-action" must be set.		FALSE
D_SEL	BOOL	0.7	DERIVATIVE ACTION ON The PID algorithm allows On/Off switching of individual PID-actions. The D-action is on when the input "derivative action on" is set.		FALSE

Parameters	Data type	Address (Instance DB)	Description	Value range	Default
CYCLE	TIME	2	SAMPLE TIME The time between the block calls must be constant. The "sampling time" input specifies the time between block calls.	≥20 ms	T#1 s
SP_INT	REAL	6	INTERNAL SETPOINT The "internal setpoint" input is used to specify a setpoint.	-100.0 ... 100.0 (%) or phys. value[1)]	0.0
PV_IN	REAL	10	PROCESS VARIABLE IN An initialization value can be set at the "Process variable in" input or an external process variable in floating point format can be connected.	-100.0 ... 100.0 (%) or phys. size[1)]	0.0
PV_PER	WORD	14	PROCESS VARIABLE PERIPHERY The process variable in the I/O format is connected to the controller at the "process variable peripheral" input.		W#16# 0000
MAN	REAL	16	MANUAL VALUE The "manual value" input is used to set a manual default value by means of operator control / monitoring functions.	-100.0 ... 100.0 (%) or phys. value[2)]	0.0
GAIN	REAL	20	PROPORTIONAL GAIN The "proportional gain" input sets the controller gain.	The sign specifies the controller's direction of action (e. g. negative gain for cooling operations)	2.0
TI	TIME	24	RESET TIME The "reset time" input determines the time response of the integrator.	≥ CYCLE	T#20 s
TD	TIME	28	DERIVATIVE TIME The "derivative time" input determines the time response of the derivative unit.	≥ CYCLE	T#10 s
TM_LAG	TIME	32	TIME LAG OF THE DERIVATIVE ACTION/delay time of the D-action The algorithm of the D-action includes a time lag which can be assigned to the "Time lag of the derivative-action" input.	≥ CYCLE/2 Recommended: 1/5 TD	T#2 s
DEADB_W	REAL	36	DEAD BAND WIDTH A dead band is applied to the error. The "dead band width" input determines the size of the dead band.	≥0.0 (%) or phys. size[1)]	0.0

Parameters	Data type	Address (Instance DB)	Description	Value range	Default
LMN_HLM	REAL	40	MANIPULATED VALUE HIGH LIMIT The manipulated value is always limited to an high and low limit. The "manipulated value high limit" input specifies the high limit.	LMN_LLM ... 100.0 (%) or phys. value[2)]	100.0
LMN_LLM	REAL	44	MANIPULATED VALUE LOW LIMIT The manipulated value is always limited to an high and low limit. The "manipulated value low limit" input specifies the low limit.	-100.0 ... LMN_HLM (%) or phys. value[2)]	0.0
PV_FAC	REAL	48	PROCESS VARIABLE FACTOR The "process variable factor" input is multiplied by the process variable. The input is used to adapt the process variable range.		1.0
PV_OFF	REAL	52	PROCESS VARIABLE OFFSET The "process variable offset" input is added to the process variable. The input is used to adapt the process variable range.		0.0
LMN_FAC	REAL	56	MANIPULATED VALUE FACTOR The "manipulated value factor" input is multiplied by the manipulated value. The input is used to adapt the manipulated value range.		1.0
LMN_OFF	REAL	60	MANIPULATED VALUE OFFSET The input "manipulated value offset" is added to the manipulated value. The input is used to adapt the manipulated variable range.		0.0
I_ITLVAL	REAL	64	INITIALIZATION VALUE OF THE INTEGRAL ACTION/ The output of the integrator can be set at input I_ITL_ON. The initialization value is indicated at the input "Initialization value of the integral-action."	-100.0 ... 100.0 (%) or phys. size[2)]	0.0
DISV	REAL	68	DISTURBANCE VARIABLE For feed forward control, the disturbance variable is connected to input "disturbance variable."	-100.0 ... 100.0 (%) or phys. size[2)]	0.0
[1)] Parameter in the setpoint and actual value operation with the same unit [2)] Parameter in the manipulated variable channel with the same unit					

The table below contains the **output parameters** of SFB 41 "CONT_C":

Parameter	Data type	Address (instance DB)	Description	Value range	Default
LMN	REAL	72	MANIPULATED VALUE The effective manipulated value is output in floating point format at the "manipulated value" output.		0.0
LMN_PER	WORD	76	MANIPULATED VALUE PERIPHERY The manipulated value in the I/O format is connected to the controller at the "manipulated value periphery" output.		W#16# 0000
QLMN_HLM	BOOL	78.0	HIGH LIMIT OF MANIPULATED VALUE REACHED The manipulated value is always limited to an high and low limit. The output "high limit of manipulated value reached" indicates that the high limit has been exceeded.		FALSE
QLMN_LLM	BOOL	78.1	LOW LIMIT OF MANIPULATED VALUE REACHED The manipulated value is always limited to an high and low limit. The output "low limit of manipulated value reached" indicates that the low limit has been exceeded.		FALSE
LMN_P	REAL	80	PROPORTIONALITY COMPONENT/ The "proportional component" output contains the proportional component of the manipulated variable.		0.0
LMN_I	REAL	84	INTEGRAL COMPONENT The "integral component" output contains the integral component of the manipulated value.		0.0
LMN_D	REAL	88	DERIVATIVE COMPONENT The "derivative action" output contains the derivative action of the manipulated variable.		0.0
PV	REAL	92	PROCESS VARIABLE The effective process variable is output at the "process variable" output.		0.0
ER	REAL	96	ERROR SIGNAL The effective error is output at the "error signal" output.		0.0

7.5.2 Step Control with SFB 42 "CONT_S"

Introduction

The SFB/FB "CONT_S" (step controller) is used on SIMATIC S7 programmable logic controllers to control technical processes with binary manipulated value output signals for integrating actuators. During parameter assignment, you can activate or deactivate subfunctions of the PI step controller to adapt the controller to the process. You can easily do this by using the parameter assignment screen (Menu path: "Start > Simatic > STEP 7 > Assign PID control parameters"). The online electronic manual is found under "Start > Simatic > S7 Manuals > PID Control English".

Application

You can use the controller as a PI fixed setpoint controller or in secondary control loops in cascade, blending or ratio controllers, however not as the primary controller. The functions of the controller are based on the PI control algorithm of the sampling controller supplemented by the functions for generating the binary output signal from the analog actuating signal.

The I action of the controller can be switched off with TI = T#0 ms. The block can therefore be used as P controller.

Since the controller works without any position feedback signal, the internally calculated manipulated variable will not exactly match the signal control element position. An adjustment is made if the manipulated variable (ER*GAIN) is negative. The controller then sets the output QLMNDN (manipulated value signal low) until LMNR_LS (low limit of the position feedback signal) is set.

The controller can also be used as a secondary actuator in a controller cascade. The setpoint input SP_INT is used to assign the control element position. In this case the actual value input and the parameter TI (integration time) must be set to zero. Fields of application include temperature control via a motor-operated valve flap. In this case, to close the valve completely, the manipulated variable (ER*GAIN) should have a negative setting.

Description

Apart from the functions in the process value channel, the SFB implements a complete PI controller with a digital manipulated value output and the option of influencing the manipulated value manually. The step controller operates without a position feedback signal. Limiting stop signal can be used to limit pulse output.

Below you will find a detailed description of the subfunctions:

Setpoint operation
The setpoint is entered in floating-point format at the **SP_INT** input.

Actual value operation
The process variable can be input in the peripheral (I/O) or floating-point format. The CRP_IN function converts the PV_PER I/O value to a floating-point format of -100 to +100 % according to the following formula:

$$\text{Output of CPR\_IN} = \text{PV\_PER} \quad x \quad \frac{100}{27648}$$

The PV_NORM function standardizes the output of CRP_IN according to the following formula:

Output of PV_NORM = (Output of CPR_IN) x PV_FAC + PV_OFF

PV_FAC has a default of 1 and PV_OFF a default of 0.

The variables PV_FAC and PV_OFF are the result of formula conversion as follows:

PV_OFF = (Output of PV_NORM) - (Output of CPR_IN) x PV_FAC

$$\text{PV\_FAC} = \frac{\text{(output of PV\_NORM) - PV\_OFF}}{\text{Output of CPR\_IN}}$$

Calculating the negative deviation
The difference between setpoint and actual value forms the negative deviation. To suppress a small constant oscillation due to the manipulated variable quantization (for example, due to a limited resolution of the manipulated value by the actuator valve), a dead band is applied to the error signal (DEADBAND). If DEADB_W = 0, the dead band is switched off.

PI step algorithm
The SFB operates without position feedback. The I action of the PI algorithm and the assumed position feedback signal are calculated in one integrator (INT) and compared with the remaining P action as a feedback value. The difference is applied to a three-step element (THREE_ST) and a pulse generator (PULSEOUT) that creates the pulses for the actuator. The switching frequency of the controller can be reduced by adapting the threshold on of the three-step element.

Error Value Blending
A disturbance variable can be fed forward at the DISV input.

Initialization

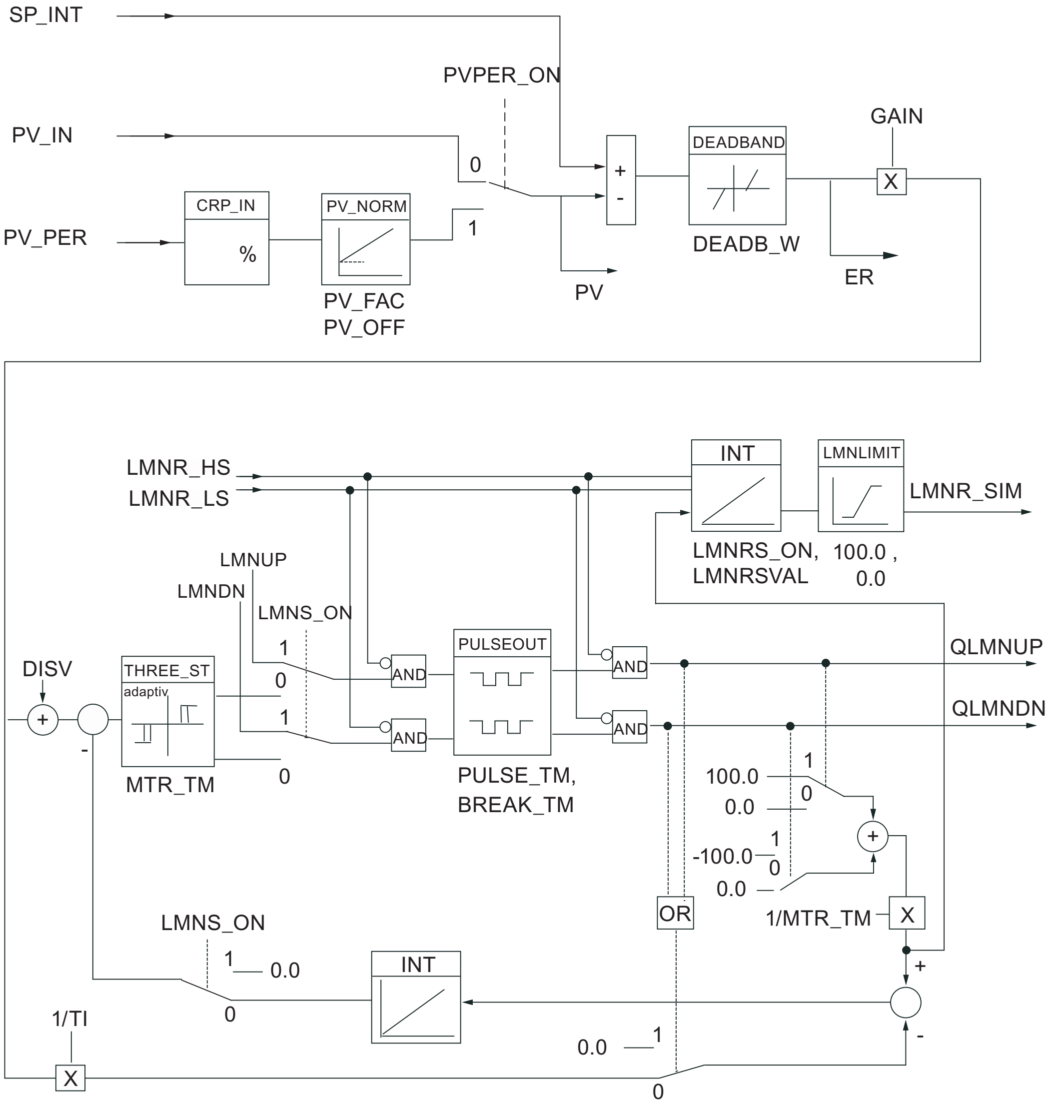
The SFB "CONT_S" has an initialization routine that is run through when the input parameter COM_RST = TRUE is set.

All other outputs are set to their default values.

Error Information

A parameter check is carried out via the Parameter Assignment Tool.

CONT_S Block Diagram



Parameters of SFB 42

The table below contains the **input parameters** of SFB 42 "CONT_S":

Parameters	Data type	Address (instance DB)	Description	Value range	Default
COM_RST	BOOL	0.0	COMPLETE RESTART The block has an initialization routine that is processed when the input COM_RST is set.	TRUE: restart FALSE: controller operation	FALSE
LMNR_HS	BOOL	0.1	HIGH LIMIT SIGNAL OF REPEATED MANIPULATED VALUE The signal "Control valve at high limit stop" is interconnected at input "High limit signal of repeated manipulated value". LMNR_HS = TRUE means: The control valve is at high limit stop.		FALSE
LMNR_LS	BOOL	0.2	LOW LIMIT SIGNAL OF REPEATED MANIPULATED VALUE The signal "Control valve at low limit stop" is interconnected at the input "Low limit signal of repeated manipulated value". LMNR_LS = TRUE means: The control valve is at low limit stop.		FALSE
LMNS_ON	BOOL	0.3	MANIPULATED SIGNALS ON The actuating signal processing is switched to manual at the "manual actuating signals on" input.		TRUE
LMNUP	BOOL	0.4	MANIPULATED SIGNALS UP With manual actuating value signals, the output signal QLMNUP is set at the input "actuating signals up."		FALSE
LMNDN	BOOL	0.5	MANIPULATED SIGNALS DOWN With manual actuating value signals, the output signal QLMNDN is set at the input "actuating signals down."		FALSE
PVPER_ON	BOOL	0.6	PROCESS VARIABLE PERIPHERY ON If the If the process variable is read from the I/Os, input PV_PER must be connected to the peripherals and input "PROCESS VARIABLE PERIPHERY ON" must be set.		FALSE
CYCLE	TIME	2	SAMPLE TIME The time between the block calls must be constant. The "sampling time" input specifies the time between block calls.	≥ 20ms	T#1 s
SP_INT	REAL	6	INTERNAL SETPOINT The "internal setpoint" input is used to specify a setpoint.	-100.0 ... 100.0 (%) or phys. size 1)	0.0
PV_IN	REAL	10	PROCESS VARIABLE IN An initialization value can be set at the "Process variable in" input or an external process variable in floating point format can be connected.	-100.0 ... 100.0 (%) or phys. size 1)	0.0

Parameters	Data type	Address (instance DB)	Description	Value range	Default
PV_PER	WORD	14	PROCESS VARIABLE PERIPHERY The process variable in the I/O format is connected to the controller at the "process variable peripheral" input.		W#16# 0000
GAIN	REAL	16	PROPORTIONAL GAIN The "proportional gain" input sets the controller gain.	The sign specifies the controller's direction of action (e. g. negative gain for cooling operations)	2.0
TI	TIME	20	RESET TIME The "reset time" input determines the time response of the integrator.	T#0 ms or ≥ CYCLE	T#20 s
DEADB_W	REAL	24	DEAD BAND WIDTH A dead band is applied to the error. The "dead band width" input determines the size of the dead band.	100.0 (%) or phys. size[1)]	1.0
PV_FAC	REAL	28	PROCESS VARIABLE FACTOR The "process variable factor" input is multiplied by the process variable. The input is used to adapt the process variable range.		1.0
PV_OFF	REAL	32	PROCESS VARIABLE OFFSET The "process variable offset" input is added to the process variable. The input is used to adapt the process variable range.		0.0
PULSE_TM	TIME	36	MINIMUM PULSE TIME A minimum pulse duration can be assigned with the parameter "minimum pulse time."	≥ CYCLE integral multiple of Cycle	T#3 s
BREAK_TM	TIME	40	MINIMUM BREAK TIME A minimum break duration can be assigned with the parameter "minimum break time."	≥ CYCLE integral multiple of Cycle	T#3 s
MTR_TM	TIME	44	MOTOR MANIPULATED VALUE The time required by the actuator to move from limit stop to limit stop is entered at the "motor actuating time" parameter.	≥ CYCLE	T#30 s
DISV	REAL	48	DISTURBANCE VARIABLE For disturbance compensation, the disturbance variable is connected to the input "disturbance variable."	-100.0 ... 100.0 (%) or phys. size[2)]	0.0

[1)] Parameter in the setpoint and process variable channel with the same unit

[2)] Parameter in the manipulated value channel with the same unit

The table below contains the **output parameters** of SFB 42 "CONT_S":

Parameters	Data type	Address (instance DB)	Description	Value range	Default
QLMNUP	BOOL	52.0	MANIPULATED SIGNALS UP If the output "actuating signal up" is set, the control valve is opened.		FALSE
QLMNDN	BOOL	52.1	MANIPULATED SIGNALS DOWN If the output "actuating signal down" is set, the control valve is opened.		FALSE
PV	REAL	54	PROCESS VARIABLE The effective process variable is output at the "process variable" output.		0.0
ER	REAL	58	ERROR SIGNAL The effective error is output at the "error signal" output.		0.0

7.5.3 Generating Pulses with SFB 43 "PULSEGEN"

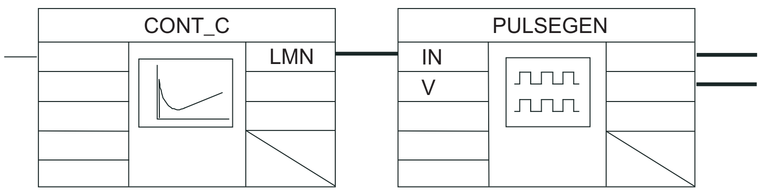
Introduction

SFB "PULSEGEN" (pulse generator) is used to structure a PID controller with pulse output for proportional actuators.

The online electronic manual is found under "Start > Simatic > S7 manuals > PID Control English".

Application

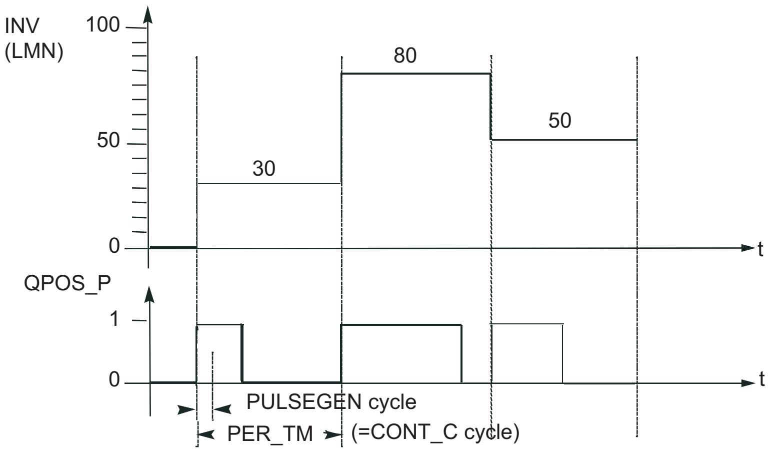
PID two-or three-step controllers with pulse width modulation can be configured using the SFB "PULSEGEN". This function is normally used in conjunction with the continuous controller "CONT_C".



Description

The function PULSEGEN transforms the input variable INV (= LMN of the PID controller) by modulating the pulse width into a pulse train with a constant period, corresponding to the cycle time at which the input variable is updated and which must be assigned in PER_TM.

The duration of a pulse per period is proportional to the input variable. The cycle configuration in PER_TM is not identical to the processing cycle of the SFB "PULSEGEN". Rather, a PER_TM cycle represents the sum of multiple processing cycles of the SFB "PULSEGEN". Here, the number of SFB "PULSEGEN" calls per PER_TM cycle represents a measure for the accuracy of the pulse width. The minimum manipulated value is here determined in the P_B_TM parameter.

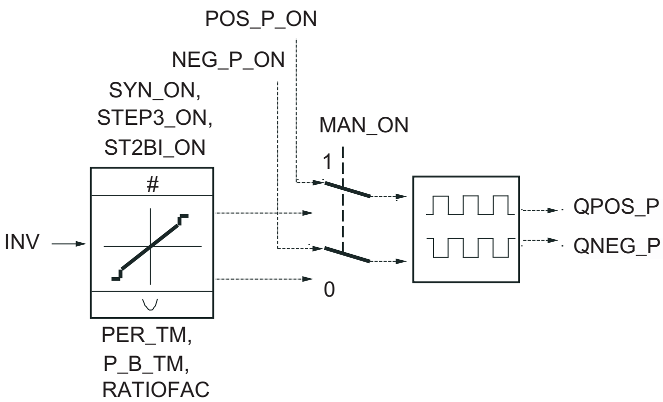


Pulse Width Modulation

An input variable of 30% and 10 SFB "PULSEGEN" calls per PER_TM therefore means:

- "One" at the QPOS output for the first three calls of SFB "PULSEGEN" (30 % of 10 calls)
- "Zero" at the QPOS output for seven further calls of SFB "PULSEGEN" (70 % of 10 calls)

Block Diagram



Accuracy of the Manipulated Value

In this example, the "Sampling rate" of 1:10 (CONT_C calls to PULSEGEN calls) reduces the accuracy of the manipulated value to 10 %, that is, default input values INV can only be imaged on output QPOS in a 10 % pitch.

The accuracy is increased as the number of SFB "PULSEGEN" calls per CONT_C call is increased.

For example, if PULSEGEN is called 100 times more frequently than CONT_C, the resolution of the manipulated value range will reach 1% (recommended resolution ≤ 5%).

Note

The call ratio must be programmed by the user.

Automatic Synchronization

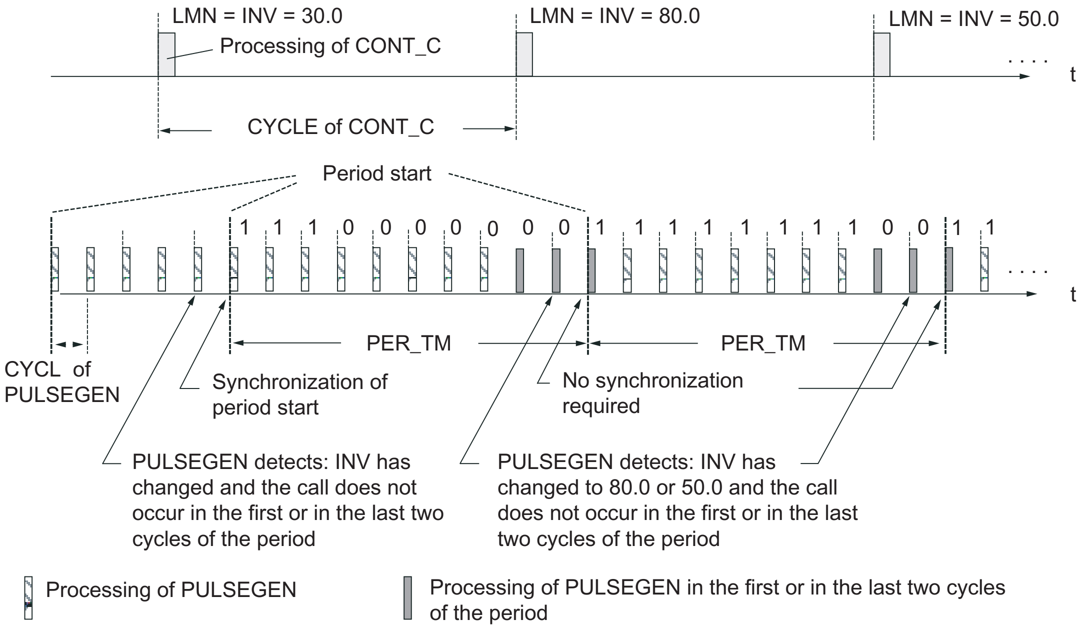
It is possible to synchronize the pulse output automatically with the block that updates the input variable INV (for example, CONT_C). This ensures that a change in the input variable is output as quickly as possible as a pulse.

The pulse generator evaluates the input value INV at intervals corresponding to the period PER_TM and converts the value into a pulse signal of corresponding length.

Since, however, INV is usually calculated in a slower cyclic interrupt class, the pulse generator should start the conversion of the discrete value into a pulse signal as soon as possible after the updating of INV.

To allow this, the block can synchronize the start of the period using the following procedure:

If INV changes and if the block call is not in the first or last two call cycles of a period, the synchronization is performed. The pulse duration is recalculated and in the next cycle is output with a new period.



Automatic synchronization can be switched off per "SYN_ON" (= FALSE) input.

Note

At the start of the new period an image of the old value of INV (that is, of LMN) is blended more or less inaccurately to the pulse signal.

Operating Modes

Depending on the parameters assigned to the pulse generator, PID controllers with a three-step output or with a bipolar or unipolar two-step output can be configured. The following table illustrates the setting of the switch combinations for the possible modes:

Operating mode	MAN_ON	Switch STEP3_ON	ST2BI_ON
Three-step control	FALSE	TRUE	Any
Two-step control with bipolar manipulated value range (-100% to 100 %)	FALSE	FALSE	TRUE
Two-step control with unipolar manipulated value range (0 % to 100 %)	FALSE	FALSE	FALSE
Manual mode	TRUE	Any	Any

Three-step control

"Three-step control" can generate three states for the control signal. The values of the binary output signals QPOS_P and QNEG_P are assigned to the states of the actuator. The table shows the example of a temperature control:

Output Signals	Heat	Actuator Off	Cool
QPOS_P	TRUE	FALSE	FALSE
QNEG_P	FALSE	FALSE	TRUE

Based on the input variable, a characteristic curve is used to calculate a pulse duration. The form of the characteristic curve is defined by the minimum pulse or minimum break time and the ratio factor.

The normal value for the ratio factor is 1.

The "doglegs" in the curves are caused by the minimum pulse or minimum break times.

Minimum pulse or minimum break time

A correctly assigned minimum pulse or minimum break time P_B_TM can prevent short on/off times that reduce the working life of switching elements and actuators.

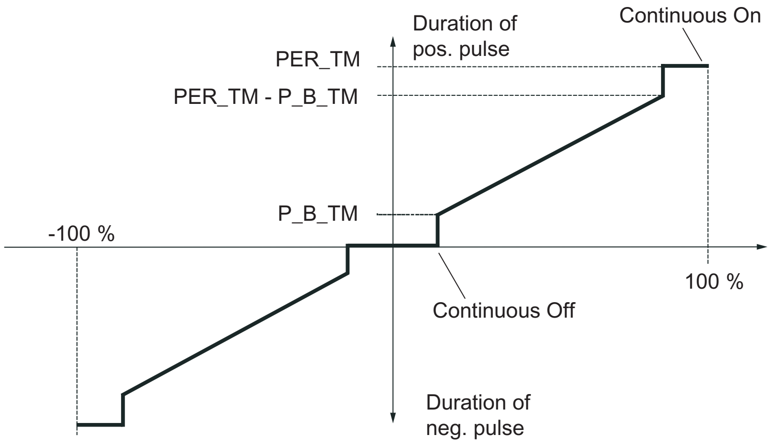
Note

Small absolute values at the input variable LMN that could otherwise generate a pulse duration shorter than P_B_TM are suppressed. Large input values that would generate a pulse duration of longer than (PER_TM-P_B_TM) are set to 100 % or -100 %.

The duration of the positive or negative pulses is calculated from the input variable (in %) multiplied by the period:

$$\text{Pulse duration} = \frac{\text{INV}}{100} \times \text{PER\_TM}$$

The following figure shows a symmetrical curve of a three-step controller (ratio factor = 1):



Using the ratio factor RATIOFAC, the ratio of the duration of positive to negative pulses can be changed. With thermal processes for example, this can be used to take into consideration the different time constants for heating and cooling actuators.

The ratio factor also influences the minimum pulse or minimum break time. A ratio factor < 1 means that the threshold value for negative pulses is multiplied by the ratio factor.

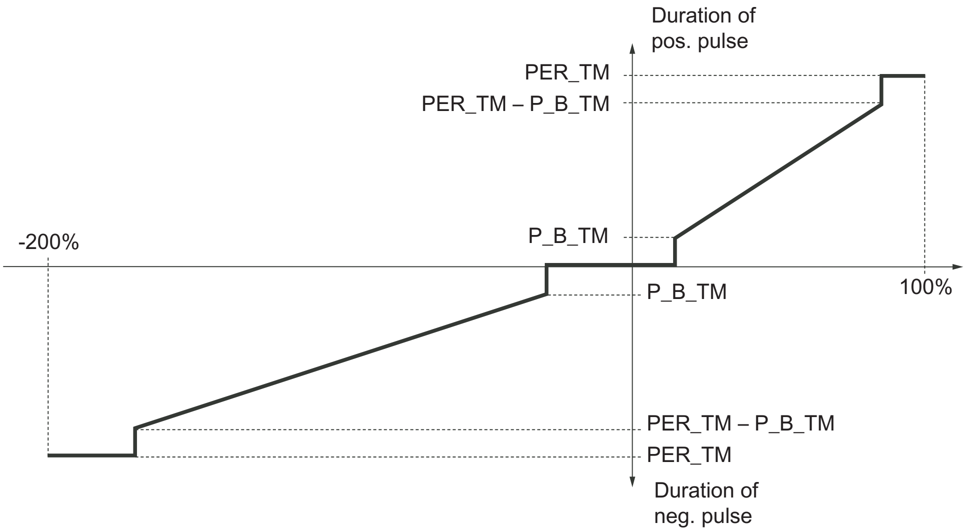
Ratio Factor < 1

The pulse period on the negative pulse output is reduced by the ratio factor that is calculated with the input value multiplied by the pulse period.

$$\text{Positive pulse duration} = \frac{\text{INV}}{100} \times \text{PER\_TM}$$

$$\text{Negative pulse duration} = \frac{\text{INV}}{100} \times \text{PER\_TM} \times \text{RATIOFAC}$$

The following figure shows the asymmetric curve of the three-step controller (ratio factor = 0.5):



Ratio factor > 1

The pulse duration on the positive pulse output calculated from the input variable times the period is reduced by the ratio factor.

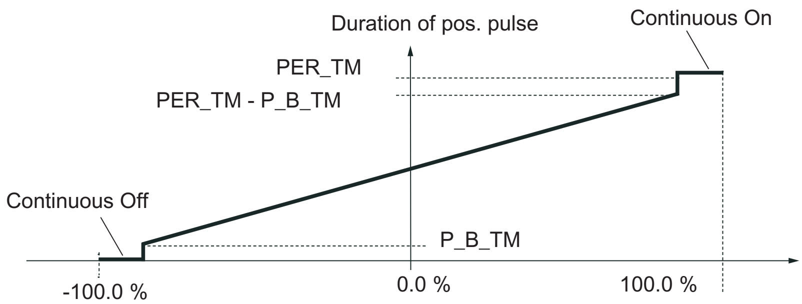
$$\text{Negative pulse duration} = \frac{INV}{100} \times PER\_TM$$

$$\text{Positive pulse duration} = \frac{INV}{100} \times \frac{PER\_T}{RATIOFAC}$$

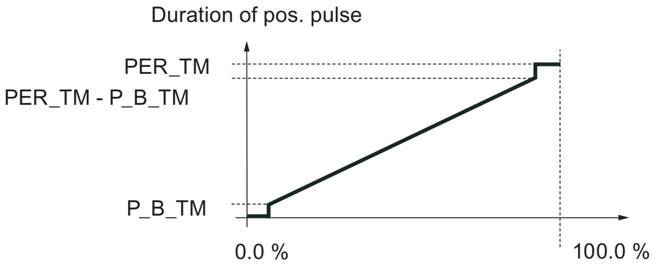
Two-step control

With the two-step control, only the positive pulse output QPOS_P of PULSEGEN is connected to the I/O actuator. Depending on the manipulated value range being used, the two-step controller has a bipolar or a unipolar manipulated value range.

Two-step control with bipolar manipulated value range (-100 % to 100 %):



Two-step control with unipolar manipulated value range (0 % to 100 %)



The negated output signal is available at QNEG_P if the connection of the two-step controller in the control loop requires a logically inverted binary signal for the actuating pulses.

Pulse	Actuator On	Actuator Off
QPOS_P	TRUE	FALSE
QNEG_P	FALSE	TRUE

Manual Mode with Two or Three-Step Control

In manual mode (MAN_ON = TRUE), the binary outputs of the three-step or two-step controller can be set using the signals POS_P_ON and NEG_P_ON regardless of INV.

	POS_P_ON	NEG_P_ON	QPOS_P	QNEG_P
Three-step control	FALSE	FALSE	FALSE	FALSE
	TRUE	FALSE	TRUE	FALSE
	FALSE	TRUE	FALSE	TRUE
	TRUE	TRUE	FALSE	FALSE
Two-step control	FALSE	Any	FALSE	TRUE
	TRUE	Any	TRUE	FALSE

Initialization

The SFB "PULSEGEN" has an initialization routine that is run through when the input parameter COM_RST = TRUE is set.

All the signal outputs are set to zero.

Error Information

A parameter check is carried out via the Parameter Assignment Tool.

Parameters of SFB 43

The table below contains the **input parameters** of SFB 43 "PULSEGEN":

Parameter	Data type	Address (instance DB)	Description	Value range	Default
INV	REAL	0	INPUT VARIABLE An analog manipulated value is assigned to the input parameter "input variable".		0.0
			• For three-step control with RATIOFAC <1:	-100/ RATIOFAC to 100 (%)	
			• For three-step control with RATIOFAC >1:	-100 to 100/ RATIOFAC (%)	
			• With bipolar two-step control:	-100 to 100 (%)	
			• With unipolar two-step control:	to 100 (%)	
PER_TM	TIME	4	PERIOD TIME The constant period of pulse width modulation is input with the "period time" input parameter. This corresponds with the sampling time of the controller. The ratio between the sampling time of the pulse generator and the sampling time of the controller determines the accuracy of pulse width modulation.	≥ 20*CYCLE of SFB 43 (corresponds to sampling time SFB 41)	T#1 s
P_B_TM	TIME	8	MINIMUM PULSE/BREAK TIME A minimum pulse or minimum break time can be assigned at the input parameters "minimum pulse or minimum break time".	≥ CYCLE	T#50 ms
RATIOFAC	REAL	12	RATIO FACTOR The input parameter "ratio factor" can be used to change the ratio of the duration of negative to positive pulses. In a thermal process, for example, this would allow different time constants for heating and cooling (e.g. process with electrical heating and water cooling) to be compensated.	10.0	1.0
STEP3_ON	BOOL	16.0	THREE STEP CONTROL ON The "three-step control on" input parameter activates this mode. In three-step control, both output signals are active.		TRUE

Parameter	Data type	Address (instance DB)	Description	Value range	Default
ST2BI_ON	BOOL	16.1	TWO STEP CONTROL FOR BIPOLAR MANIPULATED VALUE RANGE ON With the input parameter "two-step control for bipolar manipulated value range on" you can select between the modes control for bipolar manipulated value" and "two-step control for unipolar manipulated value range". Here, STEP3_ON = FALSE.		FALSE
MAN_ON	BOOL	16.2	MANUAL MODE ON By setting the input parameter "manual mode on," the output signals can be set manually.		FALSE
POS_P_ON	BOOL	16.3	POSITIVE MODE ON In the manual mode with three-step control, the output signal QPOS_P can be operated at the input parameter "Positive pulse On". In the manual mode with two-step control, QNEG_P is always set inversely to QPOS_P.		FALSE
NEG_P_ON	BOOL	16.4	NEGATIVE PULSE ON For manual mode three-step control, the output signal QNEG_P can be controlled at the input parameter "Negative pulse on". In the manual mode with two-step control, QNEG_P is always set inversely to QPOS_P.		FALSE
SYN_ON	BOOL	16.5	SYNCHRONIZATION ON By setting the input parameter "synchronization on," it is possible to synchronize automatically with the block that updates the input variable INV. This ensures that a changing input variable is output as a pulse as quickly as possible.	Condition: PER_TM = Sampling time of the SFB 41	TRUE
COM_RST	BOOL	16.6	COMPLETE RESTART The block has an initialization routine that is processed when the input COM_RST is set.	TRUE: restart FALSE: controller operation	FALSE
CYCLE	TIME	18	SAMPLE TIME The time between the block calls must be constant. The "sampling time" input specifies the time between block calls.	≥ 20 ms	T#10 ms

Note

The values of the input parameters are not limited in the block. There is no parameter check.

The table below contains the **output parameters** of SFB 43 "PULSEGEN":

Parameter	Data type	Address (instance DB)	Description	Value range	Default
QPOS_P	BOOL	22.0	OUTPUT POSITIVE PULSE The output parameter "output positive pulse" is set when a pulse is to be output. In three-step control, this is always the positive pulse. With two-step control QNEG_P is always set inversely to QPOS_P.		FALSE
QNEG_P	BOOL	22.1	OUTPUT NEGATIVE PULSE The output parameter "output negative pulse" is set when a pulse is to be output. In three-step control, this is always the negative pulse. With two-step control QNEG_P is always set inversely to QPOS_P.		FALSE

7.6 Diagnostics/Error Handling

Basics

A parameter check is carried out via the parameter assignment tool. "Senseless" parameters are not checked if parameters are changed in the user program. You will not receive an error information in such cases.

7.7 Installation of Examples

Using Examples

The examples (program and description) are found on the CD-ROM included in your documentation. You can also download them from the Internet. The project consists of several commented S7 programs of various complexity and aim.

The Readme.wri on the CD describes how to install the samples. After installation the samples can be found in the catalog ...\STEP7\EXAMPLES\ZDt26_04_TF_____31xC_PID.

Index

D

E

F

O

P

S