



SD328B - Stepper Drive

Programming example
(without PLCopen Library)
Profibus with
Siemens S7-1200/1500



Profibus





Table of Contents

1	Introduction.....	3
1.1	Overview	3
1.2	Before You Begin	3
1.3	Start-Up and Test	5
1.4	Operations and Adjustments	6
2	Programming example overview	7
2.1	Required files	7
3	Hardware Configuration.....	8
3.1	Profibus Hardware configuration	8
3.2	HW Identifier.....	9
4	SD328B Function Blocks.....	10
4.1	Create Axis Reference Structure	10
4.2	SD3_Control.....	11
4.3	SD3_Homing	13
4.4	SD3_Jog.....	16
4.5	SD3_ProfilePosition.....	18
4.6	SD3_ProfileVelocity.....	20
4.7	SD3_Parameter_Read_Write.....	22



1 Introduction

1.1 Overview

This chapter gives the introduction.

Contents of this chapter

This chapter contains the following topics:

Topic	Page
Before you begin	03
Start-Up and Test	05
Operations and Adjustments	06

1.2 Before You Begin

General

The products specified in this document have been tested under actual service conditions. Of course, your specific application requirements may be different from those assumed for this and any related examples described herein. In that case, you will have to adapt the information provided in this and other related documents to your particular needs. To do so, you will need to consult the specific product documentation of the hardware and/or software components that you may add or substitute for any examples specified in this documentation. Pay particular attention and conform to any safety information, different electrical requirements and normative standards that would apply to your adaptation.

© 2014 Schneider Electric. All rights reserved

WARNING

REGULATORY INCOMPATIBILITY

Be sure that all equipment applied and systems designed comply with all applicable local, regional and national regulations and standards

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Please Note

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material. A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved. Failure to observe this information can result in injury or equipment damage.



Expert Support Machine Solution

The use and application of the information contained herein require expertise in the design and programming of automated control systems. Only the user or integrator can be aware of all the conditions and factors present during installation and setup, operation, and maintenance of the machine or process, and can therefore determine the automation and associated equipment and the related safeties and interlocks which can be effectively and properly used. When selecting automation and control equipment, and any other related equipment or software, for a particular application, the user or integrator must also consider any applicable local, regional or national standards and/or regulations.

Some of the major software functions and/or hardware components used in the proposed architectures and examples described in this document cannot be substituted without significantly compromising the performance of your application. Further, any such substitutions or alterations may completely invalidate any proposed architectures, descriptions, examples, instructions, wiring diagrams and/or compatibilities between the various hardware components and software functions specified herein and in related documentation. You must be aware of the consequences of any modifications, additions or substitutions.

A residual risk, as defined by EN/ISO 12100-1, Article 5, will remain if

- it is necessary to modify the recommended logic and if the added or modified components are not properly integrated in the control circuit.
- you do not follow the required standards applicable to the operation of the machine, or if the adjustments to and the maintenance of the machine are not properly made (it is essential to strictly follow the prescribed machine maintenance schedule).
- the devices connected to any safety outputs do not have mechanically-linked contacts.

CAUTION

EQUIPMENT INCOMPATIBILITY

Read and thoroughly understand all device and software documentation before attempting any component substitutions or other changes related to the application examples provided in the document

Failure to follow these instructions can result in injury, or equipment damage.



1.3 Start-Up and Test

Before using electrical control and automation equipment after design and installation, the application and associated functional safety system must be subjected to a start-up test by qualified personnel to verify correct operation of the equipment. It is important that arrangements for such testing be made and that enough time is allowed to perform complete and satisfactory testing.

CAUTION

EQUIPMENT OPERATION HAZARD

- Verify that all installation and set up procedures have been completed.
- Before operational tests are performed, remove all blocks or other temporary holding means used for shipment from all component devices
- Remove tools, meters, and debris from equipment.

Failure to follow these instructions can result in injury, or equipment damage.

Verify that the completed system, including the functional safety system, is free from all short circuits and grounds, except those grounds installed according to local regulations. If high-potential voltage testing is necessary, follow the recommendations in equipment documentation to help prevent injury or equipment damage.



1.4 Operations and Adjustments

General

Regardless of the care exercised in the design and manufacture of equipment or in the selection and ratings of components, there are hazards that can be encountered if such equipment is improperly installed and operated.

In some applications, such as packaging machinery, additional operator protection such as point-of-operation guarding must be provided. This is necessary if the hands and other parts of the body are free to enter the pinch points or other hazardous areas where serious injury can occur. Software products alone cannot protect an operator from injury. For this reason, the software cannot be substituted for or take the place of point-of-operation protection.

⚠ WARNING
UNGUARDED MACHINERY CAN CAUSE SERIOUS INJURY
<ul style="list-style-type: none">•Do not use this software and related automation equipment on equipment which does not have point-of -operation protection.•Do not reach into machinery during operation.
Failure to follow these instructions can result in death, serious injury, or equipment damage.

Ensure that appropriate safeties and mechanical/electrical interlocks related to point-of-operation protection have been installed and are operational before placing the equipment into service. All interlocks and safeties related to point-of-operation protection must be coordinated with the related automation equipment and software programming.

NOTE: Coordination of safeties and mechanical/electrical interlocks for point-ofoperation protection is outside the scope of the examples and implementations suggested herein. It is sometimes possible to adjust the equipment incorrectly and this produce unsatisfactory or unsafe operation. Always use the manufacturer instructions as a guide to functional adjustments. Personnel who have access to these adjustments must be familiar with the equipment manufacturer instructions and the machinery used with the electrical equipment. Only those operational adjustments actually required by the machine operator should be accessible to the operator. Access to other controls should be restricted to help prevent unauthorized changes in operating characteristics.



2 Programming example overview

The programming example contents function blocks to control a SD328B Stepper Drive via Profibus.

2.1 *Required files*

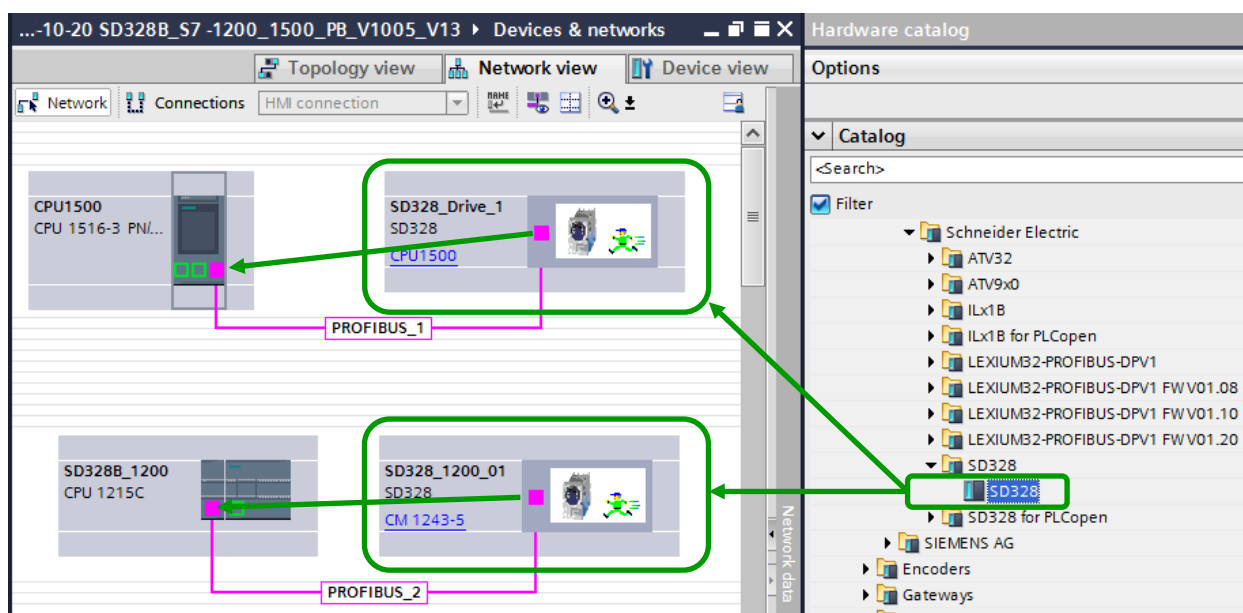
SD328B ProfibusGSD file “SE100A19.GSD”



3 Hardware Configuration

3.1 Profibus Hardware configuration

1. Insert the SD328B Profibus Device to the Network view
2. Link the SD328B PB interface to the PLC Master Interface



You have now linked the drive into the network as a Profibus Slave.



3.2 HW Identifier

The IO data is spitted in 2 channels.

Parameter channel and Process channel.

The HW Identifier of these both channels is used in the application to address the IO data.

Module	Rack	Slot	I address	Q address	Type
SD328_Drive_1	0	0			SD328
Parameterchannel	0	1	0...7	0...7	PKW Module PPO2 word
Processchannel	0	2	8...19	8...19	PKW Module PPO2 word

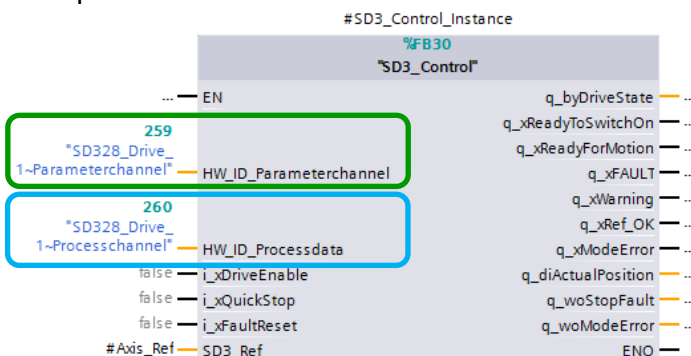
The 'Hardware identifier' field for the 'Parameterchannel' is set to 259.

Processchannel:

The 'Hardware identifier' field for the 'Processchannel' is set to 260.

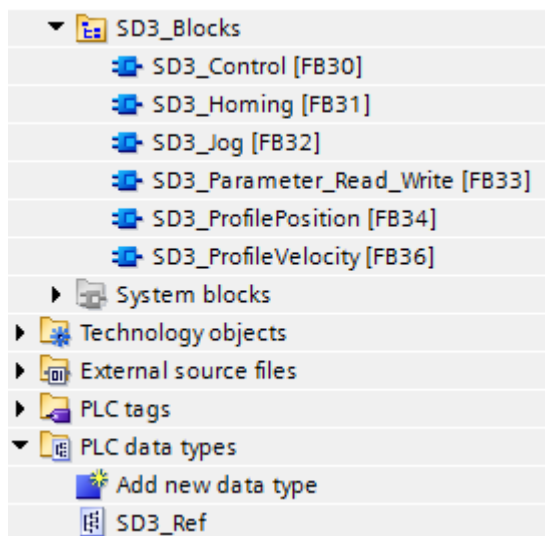
Note: The Hardware identifier is used in the application to address the IO data of the Parameter channel and the process data channel. The Hardware identifier must be declared at the input "HW_ID_Parameterchannel" and "HW_ID_Processdata" of the function block SD3_Control.

Example:





4 SD328B Function Blocks



4.1 Create Axis Reference Structure

Each SD328 needs an own axis reference structure of the PLC data type “SD3_Ref”. This structure can be defined in a global data block or in the static variables of a function block.

In this example you can see how to declare the axis reference structure in the stat variables of a function block.

FB_SD328B_Motion_Blocks							
	Name	Data type	Default value	Retain	Accessible f...	Visible in ...	Setpoint
1	▼ Input				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	HW_ID_Parameterchannel	HW_IO	16#0	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	HW_ID_Processdata	HW_IO	16#0	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	▼ Output				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	<Add new>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	▼ InOut				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	<Add new>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8	▼ Static				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9	▶ Axis_Ref	*SD3_Ref		Non-ret...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	▶ SD3_Control Instance	*SD3_Control			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

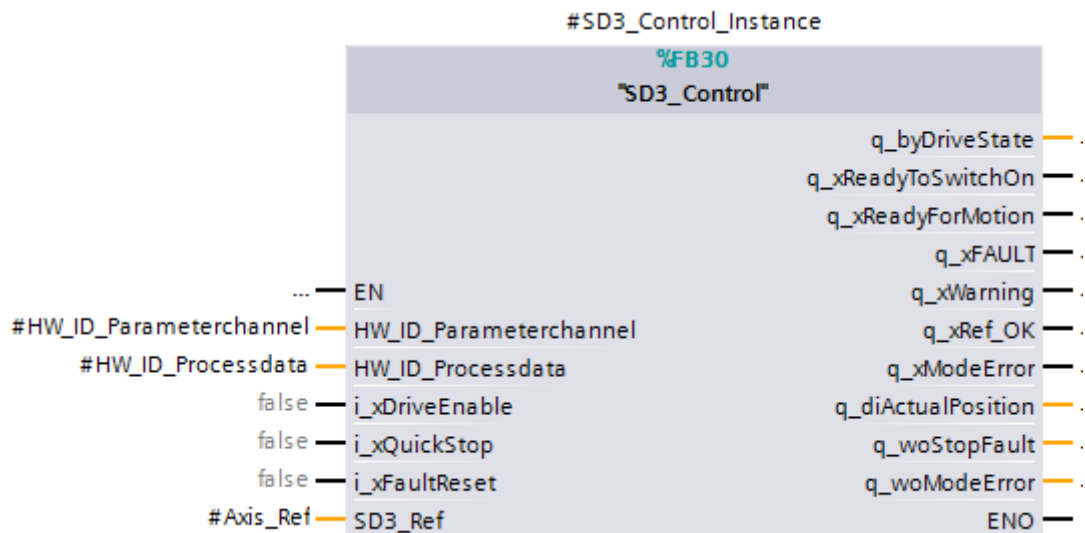


4.2 SD3_Control

Function description

- Read / write of SD328B Input / Output data. (FB must be called cyclic)
- Control the SD328B drive state (Enable power stage, fault reset, QuickStop)
- Monitor SD328B status (drive state, actual position, drive error ID, mode error ID)

Graphical representation



Input parameter description:

Parameter	Data type	Description
HW_ID_Parameterchannel	HW_IO	Hardware ID of Parameterchannel (see HW configuration)
HW_ID_Processdata	HW_IO	Hardware ID of Processchannel (see HW configuration)
i_xDriveEnable	BOOL	Enable power stage
i_xQuickStop	BOOL	Stop movement with QuickStop
i_xFaultReset	BOOL	Error reset

Input/Output parameter description:

Parameter	Data type	Description
SD3_Ref	SD3_Ref	Axis reference structure



Output parameter description:

Parameter	Data type	Description
q_byDriveState	Byte	Drive status 3 = disabled 4 = ready to switch on 6 = operation enable (ready for motion) 7 = QuickStop active 9 = fault active
q_xReadyToSwitchOn	BOOL	Power stage can be enabled (drive state = 4)
q_xReadyForMotion	BOOL	Power stage is enabled -> motion command can be executed (drive state = 6)
q_xFault	BOOL	Fault is active
q_xWarning	BOOL	Warning is active
q_xRef_OK	BOOL	Reference position is valid
q_xModeError	BOOL	Mode error occurred
q_diActualPosition	DINT	Actual motor position in USR units
q_woStopFault	BOOL	Error causing a stop (error classes 1 to 4) Number of the current error. Any consecutive errors do not overwrite this error number. Example: If a limit switch error reaction caused an overvoltage error, this parameter would contain the number of the limit switch error.
q_woModeError	BOOL	Error code for synchronous errors Usually, this is an error that was caused by the activation of an operating mode. The ModeError bit relates to MT-dependent parameters.



4.3 SD3_Homing

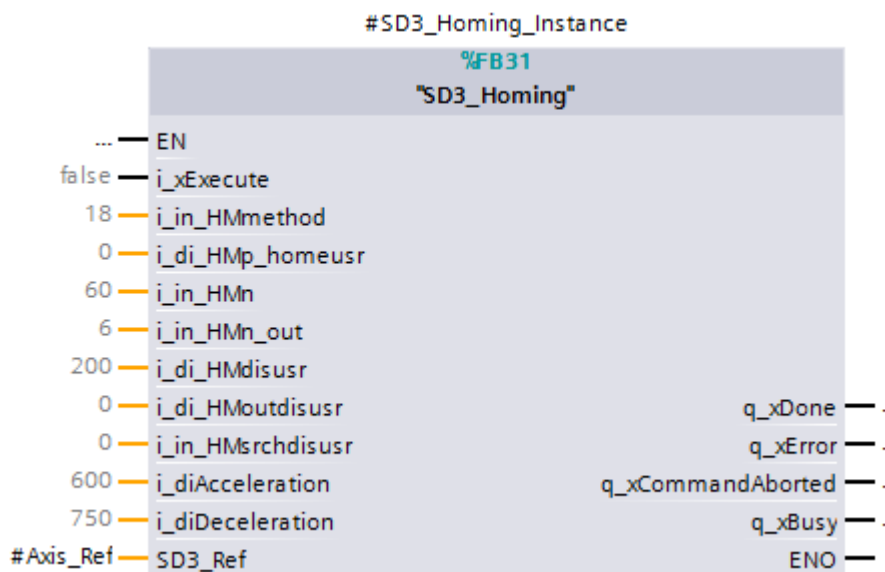
Function description

The operating mode Homing is used to define a reference point. The reference point establishes an absolute position reference between the motor position and a defined axis position. The reference point can be defined by means of a reference movement or by means of position setting.

- Reference movement: Movement to a limit switch, a reference switch or the index pulse of the motor encoder. When the position is reached, a position reference is automatically created. This position becomes the absolute user-defined position.
- Position setting: (homing method 35). The current motor position is set to a desired position value. The zero point is defined by the position value. Position setting is only possible when the motor is at a standstill.

The operating mode Homing must be completed without an error for the new reference point to be valid.

Graphical representation





Input parameter description:

Parameter	Data type	Description
i_xExecute	BOOL	Rising edge starts the homing
i_in_HM_method	INT	1 = LIMN with index pulse 2 = LIMP with index pulse 7 = REF+ with index pulse, beyond REF, in direction of LIMN 8 = REF+ with index pulse, within REF, in direction of LIMN 9 = REF+ with index pulse, within REF, in direction of LIMP 10 = REF+ with index pulse, beyond REF, in direction of LIMP 11 = REF- with index pulse, beyond REF, in direction of LIMN 12 = REF- with index pulse, within REF, in direction of LIMN 13 = REF- with index pulse, within REF, in direction of LIMP 14 = REF- with index pulse, beyond REF, in direction of LIMP 17 = LIMN 18 = LIMP 23 = REF+, beyond REF, in direction of LIMN 24 = REF+, within REF, in direction of LIMN 25 = REF+, within REF, in direction of LIMP 26 = REF+, beyond REF, in direction of LIMP 27 = REF-, beyond REF, in direction of LIMN 28 = REF-, within REF, in direction of LIMN 29 = REF-, within REF, in direction of LIMP 30 = REF-, beyond REF, in direction of LIMP 33 = on index pulse, in direction of LIMN 34 = on index pulse, in direction of LIMP 35 = Dimension setting without movement
i_di_HM_p_homeusr	DINT	Position is set as current motor position after successful reference movement [usr]. Value range: depends on scaling factor, initial value: 0.
i_in_HMn	INT	Speed for searching the limit or reference switch [rpm]. Drive stops when switching edge has been detected. Value range: 1...2147483647; Initial value: 60.
i_in_HMn_out	INT	Speed for clearance movement back to the switching edge [rpm]. Value range: 1...2147483647; Initial value: 6.
i_di_HMdisusr	DINT	Distance between the switching edge and the reference point [usr]. At end of movement, the drive moves back towards switching edge until the distance has been reached. Value range: 1...2147483647, initial value: 200.
i_di_HMoutdisusr	DINT	0: Clearing monitor switched off. >0: Run-off [usr], i.e. max. travel distance when searching for the switching edge. If the switching edge is not found in this distance, the reference movement is interrupted with an error. Value range: 0...2147483647, initial value: 0.
i_in_HMsrchdisusr	DINT	Maximum search distance after overtravel of switch 0: Search distance monitoring disabled >0: Search distance The switch must be activated again within this search distance, otherwise the reference movement is canceled. Changed settings become active the next time the motor moves.
i_diAcceleration	DINT	Value for the acceleration ramp gradient [min-1/s] Value range: 1...2147483647; Initial value: 600.
i_diDeceleration	DINT	Value for the deceleration ramp gradient [min-1/s] Value range: 1...2147483647; Initial value: 750.



Input/Output parameter description:

Parameter	Data type	Description
SD3_Ref	SD3_Ref	Axis reference structure

Output parameter description:

Parameter	Data type	Description
q_xDone	BOOL	Homing is finished without error
q_xError	BOOL	Homing is finished with error
q_xCommandAborted	BOOL	Homing aborted
q_xBusy	BOOL	Homing is active



4.4 SD3_Jog

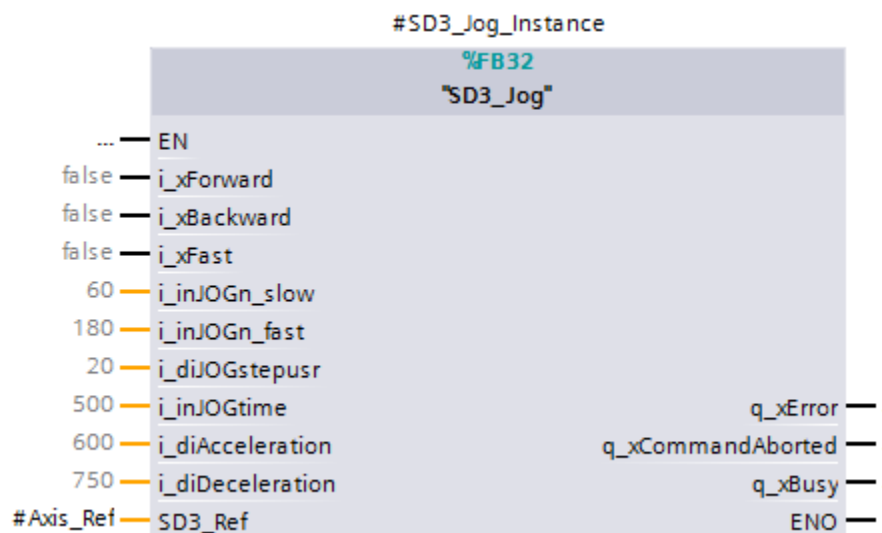
Function description

In the operating mode Jog, a movement is made from the actual motor position in the desired direction. The velocity can be set. As long as the signal for the direction is available, a continuous movement is made in the desired direction.

If movements in positive and negative directions are requested at the same time, there is no motor movement.

The function block starts the operating mode Jog. TRUE at the input Forward or the input Backward starts the jog movement.

Graphical representation



Input parameter description:

Parameter	Data type	Description
i_xForward	BOOL	JOG movement in positive direction
i_xBackward	BOOL	JOG movement in negative direction
i_xFast	BOOL	The velocity can be changed during the movement. FALSE: Movement at the velocity set in i_inJOGn_slow. TRUE: Movement at the velocity set in i_inJOGn_fast.
i_inJOGn_slow	INT	Velocity for slow movement
i_inJOGn_fast	INT	Velocity for fast movement
i_diJOGstepusr	DINT	Distance for step movement
i_inJOGtime	DINT	Wait time for step movement
i_diAcceleration	DINT	Value for the acceleration ramp gradient [min-1/s] Value range: 1...2147483647; Initial value: 600.
i_diDeceleration	DINT	Value for the deceleration ramp gradient [min-1/s] Value range: 1...2147483647; Initial value: 750.



Input/Output parameter description:

Parameter	Data type	Description
SD3_Ref	SD3_Ref	Axis reference structure

Output parameter description:

Parameter	Data type	Description
q_xError	BOOL	JOG movement is finished with error
q_xCommandAborted	BOOL	JOG movement aborted
q_xBusy	BOOL	JOG movement is active



4.5 SD3_ProfilePosition

Function description

The following settings can be made in the operating mode Profile Position:

- Target position
- Type of movement (relative movement or absolute movement)
- Target velocity
- Acceleration and deceleration ramps

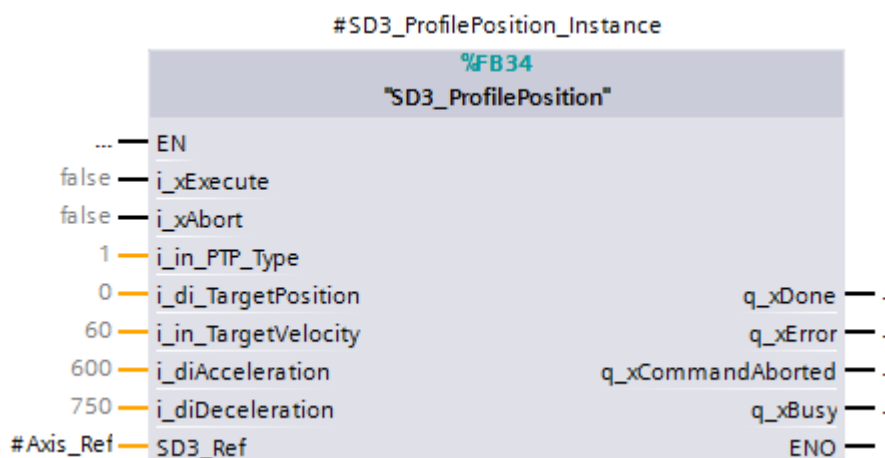
The movement to the target position is made on the basis of a motion profile. The motion profile is calculated by the profile generator in the drive. The calculation is performed on the basis of the actual position and the target position, the actual velocity and the target velocity and the acceleration and deceleration ramps.

In the operating mode Profile Position, absolute movements, relative movements and additive movements are possible.

- Absolute movement with reference to the zero point
- Relative movement with reference to the actual position
- Additive movement with reference to the previous target position

A zero point must be defined with the operating mode Homing prior to the first absolute movement.

Graphical representation



Input parameter description:

Parameter	Data type	Description
i_xExecute	BOOL	Rising edge starts the movement
i_xAbort	BOOL	Stops (aborts) an active movement
i_in_PTP_Type	INT	Type of movement 1 = Absolute movement 2 = Additive movement 3 = Relative movement
i_diTargetPosition	DINT	Target Position for absolute movement or target distance for additive/relative movement In USR units
i_in_TargetVelocity	INT	Target velocity in rpm



i_diAcceleration	DINT	Value for the acceleration ramp gradient [min-1/s] Value range: 1...2147483647; Initial value: 600.
i_diDeceleration	DINT	Value for the deceleration ramp gradient [min-1/s] Value range: 1...2147483647; Initial value: 750.

Input/Output parameter description:

Parameter	Data type	Description
SD3_Ref	SD3_Ref	Axis reference structure

Output parameter description:

Parameter	Data type	Description
q_xDone	BOOL	Movement is finished without error
q_xError	BOOL	Movement is finished with error
q_xCommandAborted	BOOL	Movement aborted
q_xBusy	BOOL	Movement is active

Note:

New target position can be started with a new rising edge on “i_xExecute” during an active movement.

Changing target velocity value during an active movement takes effect immediately.



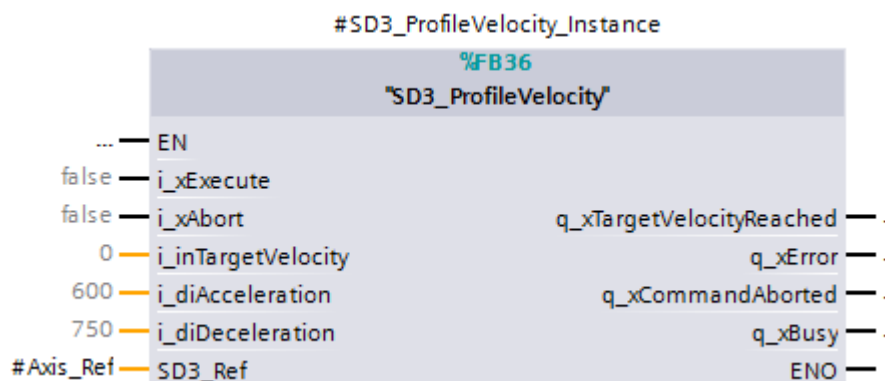
4.6 SD3_ProfileVelocity

Function description

You can set a target velocity in the operating mode Profile Velocity. The movement is performed with this target velocity in the operating mode Profile Velocity. The movement continues until a new target velocity is set or until the operating mode is aborted.

Transitions between two target velocities are performed on the basis of a motion profile. The motion profile is determined by the profile generator in the drive on the basis of the actual velocity, the target velocity and the acceleration and deceleration ramps.

Graphical representation



Input parameter description:

Parameter	Data type	Description
i_xExecute	BOOL	Rising edge starts the movement
i_xAbort	BOOL	Stops (aborts) an active movement
i_inTargetVelocity	INT	Target velocity in rpm
i_diAcceleration	DINT	Value for the acceleration ramp gradient [min-1/s] Value range: 1...2147483647; Initial value: 600.
i_diDeceleration	DINT	Value for the deceleration ramp gradient [min-1/s] Value range: 1...2147483647; Initial value: 750.

Input/Output parameter description:

Parameter	Data type	Description
SD3_Ref	SD3_Ref	Axis reference structure



Output parameter description:

Parameter	Data type	Description
q_xTargetVelocityReached	BOOL	Target velocity is reached
q_xError	BOOL	Movement is finished with error
q_xCommandAborted	BOOL	Movement aborted
q_xBusy	BOOL	Movement is active

Note:

Changing target velocity value during an active movement takes effect immediately.

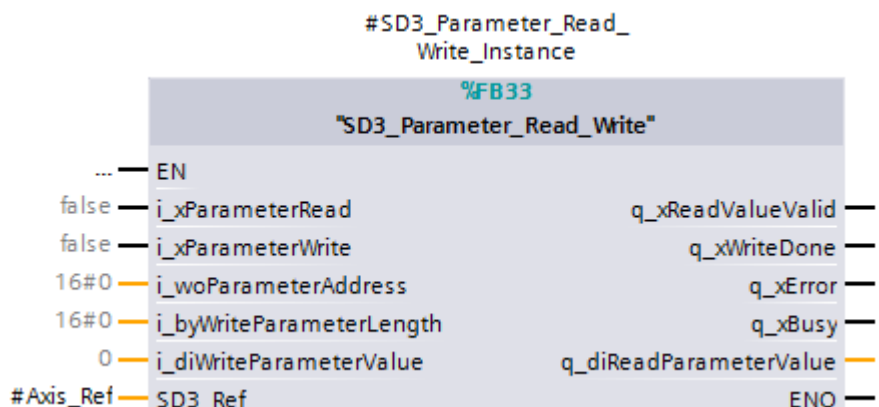


4.7 SD3_Parameter_Read_Write

Function description

The following functions block allows to read or write drive parameters. See the product manual for a description of the parameters.

Graphical representation



Input parameter description:

Parameter	Data type	Description
i_xParameterRead	BOOL	Reading a parameter
i_xParameterWrite	BOOL	Writing a parameter
i_woParameterAddress	WORD	Parameter address (see SD328B manual)
i_byWriteParameterLength	BYTE	Size in bytes of the parameter which will be written.
i_diWriteParameterValue	DINT	Write parameter value

Input/Output parameter description:

Parameter	Data type	Description
SD3_Ref	SD3_Ref	Axis reference structure

Output parameter description:

Parameter	Data type	Description
q_xReadValueValid	BOOL	When reading a parameter a new value is available on the output "q_diReadParameterValue"
q_xWriteDone	BOOL	Parameter value is written successfully
q_xError	BOOL	FB is finished with error
q_xBusy	BOOL	FB is active
q_diReadParameterValue	DINT	Parameter read value