SIEMENS
Ingenuity for life

Using the MC-PreServo and MC-PostServo Modules

SIMATIC S7-1500

https://support.industry.siemens.com/cs/ww/en/view/109741575

Siemens Industry Online Support

Warranty and Liability

Note

The Application Examples are not binding and do not claim to be complete regarding the circuits shown, equipping and any eventuality. The Application Examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for ensuring that the described products are used correctly. These Application Examples do not relieve you of the responsibility to use safe practices in application, installation, operation and maintenance. When using these Application Examples, you recognize that we cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these Application Examples at any time without prior notice.
If there are any deviations between the recommendations provided in these Application Examples and other Siemens publications – e.g. Catalogs – the contents of the other documents have priority.

We do not accept any liability for the information contained in this document.
Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act ("Produkthaftungsgesetz"), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of a condition which goes to the root of the contract ("wesentliche Vertragspflichten"). The damages for a breach of a substantial contractual obligation are, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.
Any form of duplication or distribution of these Application Examples or excerpts hereof is prohibited without the expressed consent of the Siemens AG.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.
In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions only form one element of such a concept.
Customer is responsible to prevent unauthorized access to its plants, systems, machines and networks. Systems, machines and components should only be connected to the enterprise network or the internet if and to the extent necessary and with appropriate security measures (e.g. use of firewalls and network segmentation) in place.
Additionally, Siemens' guidance on appropriate security measures should be taken into account. For more information about industrial security, please visit http://www.siemens.com/industrialsecurity.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends to apply product updates as soon as available and to always use the latest product versions. Use of product versions that are no longer supported, and failure to apply latest updates may increase customer's exposure to cyber threats.
To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under http://www.siemens.com/industrialsecurity.

Table of Contents

Warranty and Liability ... 2

1 Task ... 4

2 Solution ... 5

2.1 Position detection with MC-PreServo ... 5
2.2 Adjusting the set point with MC-PostServo ... 5
2.3 Hardware and Software Components ... 6
2.3.1 Validity ... 6
2.3.2 Components used ... 6

3 Basics ... 7

3.1 Motion Control ... 7
3.2 MC-PreServo and MC-PostServo ... 8

4 Mode of Operation ... 9

4.1 General overview ... 9
4.2 Functionality of MC-PreServo ... 10
4.2.1 Standard telegram 81 ... 10
4.2.2 “SimpleEnc” block ... 11
4.2.3 “EncoderTel81” data block ... 14
4.3 Functionality of MC-PostServo ... 15
4.3.1 Characteristic curve of a hydraulic valve ... 15
4.3.2 Linearization of characteristic curve ... 16
4.3.3 “NonLin” block ... 17
4.3.4 “SetPoints” data block ... 19

5 Configuration and Settings ... 20

5.1 Configuring the S7-1500 station ... 20
5.2 Generating the S7 program ... 22
5.2.1 Adding and configuring the blocks ... 22
5.2.2 Configuring the technology object ... 24

6 Installation and Commissioning ... 26

6.1 Installing the hardware ... 26
6.2 IP addresses and device names ... 26
6.3 PG/PC settings ... 27
6.4 Downloading the project to the components ... 28

7 Operation ... 29

8 Links & Literature ... 30

9 History ... 30

1 Task

Introduction

SIMATIC S7-1500 CPUs support the connection of drives as a speed axis or positioning axis via PROFINET, PROFIBUS or an analog drive connection. TIA Portal V14 and higher additionally provides the possibility to access the interface between axis and drive or between axis and encoder via the user program.

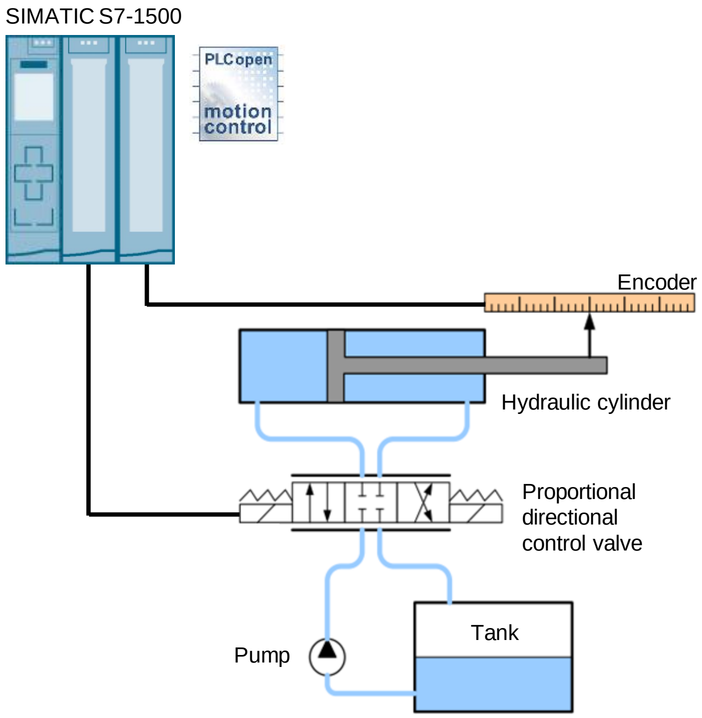
This application example shows the following:

- Using an analog encoder for a positioning axis.
- Installation of a linearization characteristic between axis and drive.

Overview of the automation task

Figure 1-1 provides an overview of the automation task.

Figure 1-1: Automation task



Requirements for the automation task

The automation fulfills the following requirements:

Table 1-1: Requirements of the automation task

Requirements	Explanation
Using a potentiometer as position encoder.	A potentiometer with an output voltage of 0 to 10 V is used as position encoder for the drive.
Presetting a speed set point with characteristic.	In the present application example, a hydraulic cylinder is operated as positioning axis. However, hydraulic valves do not have a linear control system behavior. Therefore, the speed set point needs to be adjusted for the valve.

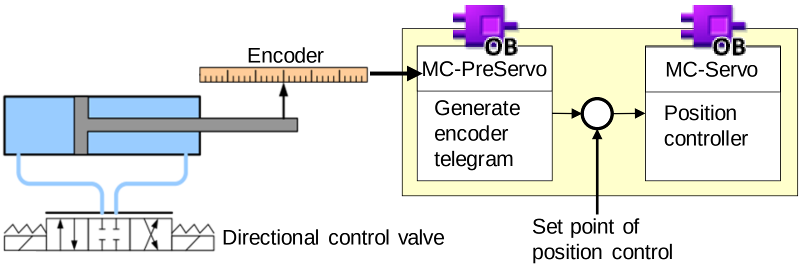
2 Solution

2.1 Position detection with MC-PreServo

The current position/current speed of the hydraulic cylinder is communicated to system control via a position measuring system (encoder). This actual value is read into the CPU via an analog input and – together with the set point value – it is used as input parameter of the position controller.

The position controller awaits the encoder data in a standardized telegram with a predefined structure. When the encoder value is read in via an analog input of the controller, this encoder telegram needs to be generated in the user program. This part of the program is called up with the organization block MC-PreServo.

Figure 2-1: Position detection with MC-PreServo



The functioning of the MC-PreServo organization block is described in chapter 4.2 .

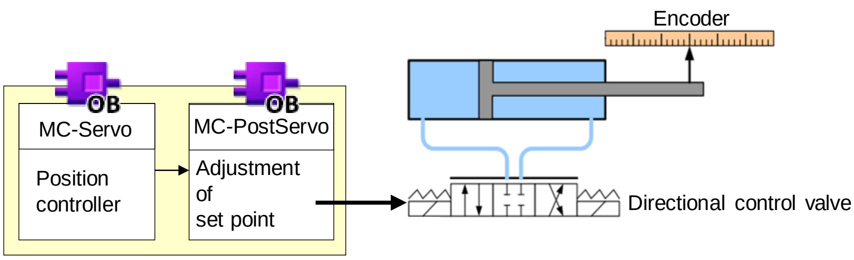
2.2 Adjusting the set point with MC-PostServo

Generally, a hydraulic axis consists of a hydraulic cylinder which is controlled via a proportional directional valve. The valve controls the flow or pressure of the hydraulic oil in the cylinder, which is provided via a pump. The valve is activated via an analog output of the CPU.

However, the position controller of the technology object is designed for linear control system behavior, such as applies in the case of electrical axes. The control system behavior of a hydraulic axis is, however, not linear. To optimally control a hydraulic axis via the “Positioning axis” technology object, you need to adjust the set point. This adjustment serves to transmit the linear control signal of the position controller to the non-linear behavior of the hydraulic axis.

Adjusting the set point to the non-linear control system behavior of the hydraulic axis is done via the function block “LonLin”. This block is part of the “Library with general functions” (LGF) and is called up in the organization block MC-PostServo.

Figure 2-2: Adjusting the set point with MC-PostServo



The functioning of the organization block MC-PostServo is described in chapter 4.3.

Topics not covered by this application

This application example does not include a description of the following points:

- SIMATIC S7-1500 Motion Control instructions
- Configuring a drive for controlling via digital and analog inputs and outputs

2.3 Hardware and Software Components

2.3.1 Validity

This application example is valid for the following products:

- STEP 7 Professional as of V14
- SIMATIC S7-1500 controllers as of firmware V2.0

2.3.2 Components used

Hardware components

Table 2-1: Hardware components

Component	Qty.	Article number	Note
SIMATIC CPU 1513-1 PN/DP	1	6ES7513-1AL01-2AB0	Alternatively, other comparable components can also be used.
SIMATIC S7-1500 Analog input module AI 8xU/I HS	1	6ES7531-7NF10-0AB0	
SIMATIC S7-1500 Analog output module AO 4xU/I ST	1	6ES7532-5HD00-0AB0	
Proportional 0-10V Directional valve	1	-	-
Potentiometer for position detection	1	-	-

Software components

Table 2-2: Software components

Component	Qty.	Article number	Note
STEP 7 (TIA Portal) Professional V14	1	6ES7822-1AA04-0YA5	-

Example files and projects

Table 2-3: Example files and projects

Component	Note
109741575_MC_PreServo_and_MC_PostServo_v10.zip	This zip file contains the STEP 7 project.
109741575_MC_PreServo_and_MC_PostServo_v10_de.pdf	This document.

3 Basics

3.1 Motion Control

S7-1500 Motion Control supports the controlled positioning and processing of axes and is an integral part of all S7-1500 as well as all S7-1500SP. The technology CPU S7-1500T additionally offers Motion Control functions.

The Motion Control function supports the following technology objects:

- Speed axis
- Positioning axis
- Synchronous axis
- External encoder and probe
- Cam and cam track
- Cam disc (for S7-1500T)

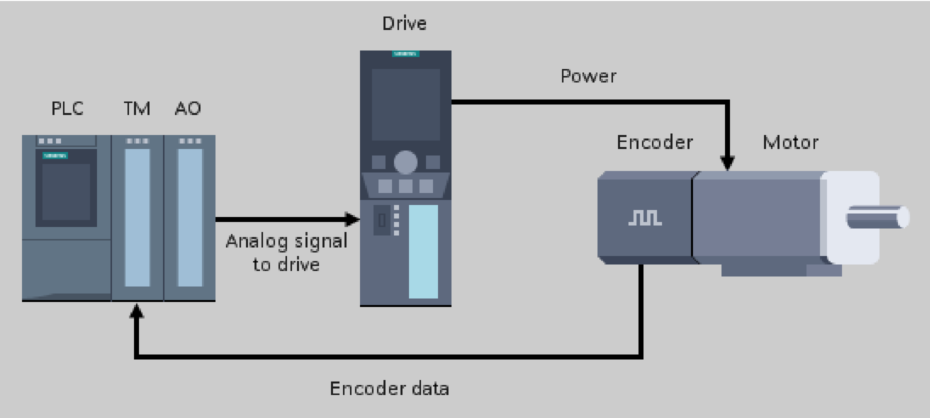
Drives with PROFI drive support and analog set point interface are controlled via standardized Motion Control instructions as per PLCopen.

The axis control panel and comprehensive online and diagnostic functions support the commissioning and optimization of drives.

In the present application example, a position-controlled operation of a hydraulic valve is realized via the "Positioning axis" technology object. In this, the valve receives the set point via the analog set point interface.

A potentiometer is used as position encoder, whose voltage value is read into the controller via an analog input module. This realizes a position detection which outputs the detected position as analog signal from 0 to 10 volt.

Figure 3-1: Configuring the technology object



Note

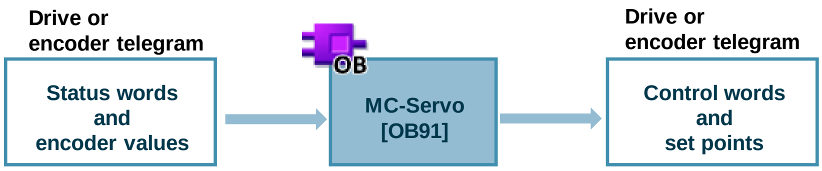
A detailed description of S7-1500 Motion Control can be found in the "SIMATIC S7-1500 Motion Control" function manual \3\.

https://support.industry.siemens.com/cs/de/en/view/109739589

3.2 MC-PreServo and MC-PostServo

At the start of the position controller of an axis (via the OB MC-Servo [OB 91]), the drive or encoder telegram is read. At the end of the position controller, the output range of the drive or encoder telegram is described.

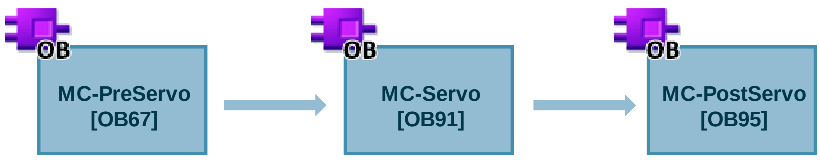
Figure 3-2: Position controller MC-Servo



The input range of the telegram (status words and encoder values) can be edited via the organization block MC-PreServo [OB 67] before it is processed by the MC-Servo. The MC-PreServo is called up by the MC-Servo.

The output range of the telegram (status words and encoder values) can be edited via the organization block MC-PostServo [OB 95] before it is output to the output module or the communication partner. The MC-PostServo is called up after the MC-Servo.

Figure 3-3: Call structure of organization blocks



The organization blocks MC-PreServo and MC-PostServo are programmable on the user side and need to be added via the “Add new block” command.

Note

A detailed description of the organization blocks MC-PreServo and MC-PostServo can be found in the “SIMATIC S7-1500 Motion Control“ function manual \3\.

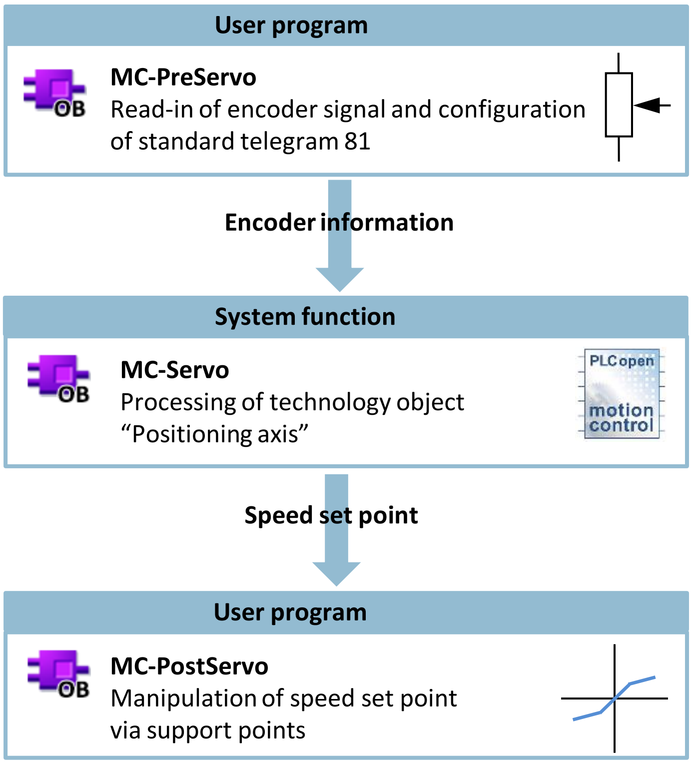
https://support.industry.siemens.com/cs/de/en/view/109739589

4 Mode of Operation

4.1 General overview

Figure 4-1 shows the call structure and the functions of the organization blocks (OBs).

Figure 4-1: General overview



Organization block MC-PreServo

In the present application example, a potentiometer is used as position encoder. The voltage of the potentiometer (0 to 10V) is read in through an analog input of the S7-1500 controller.

It now is MC-PreServo OB’s task to convert the encoder information into the structure that corresponds to the standard telegram 81. In this structure, the encoder information can then be passed on to the MC-Servo (see chapter 4.2).

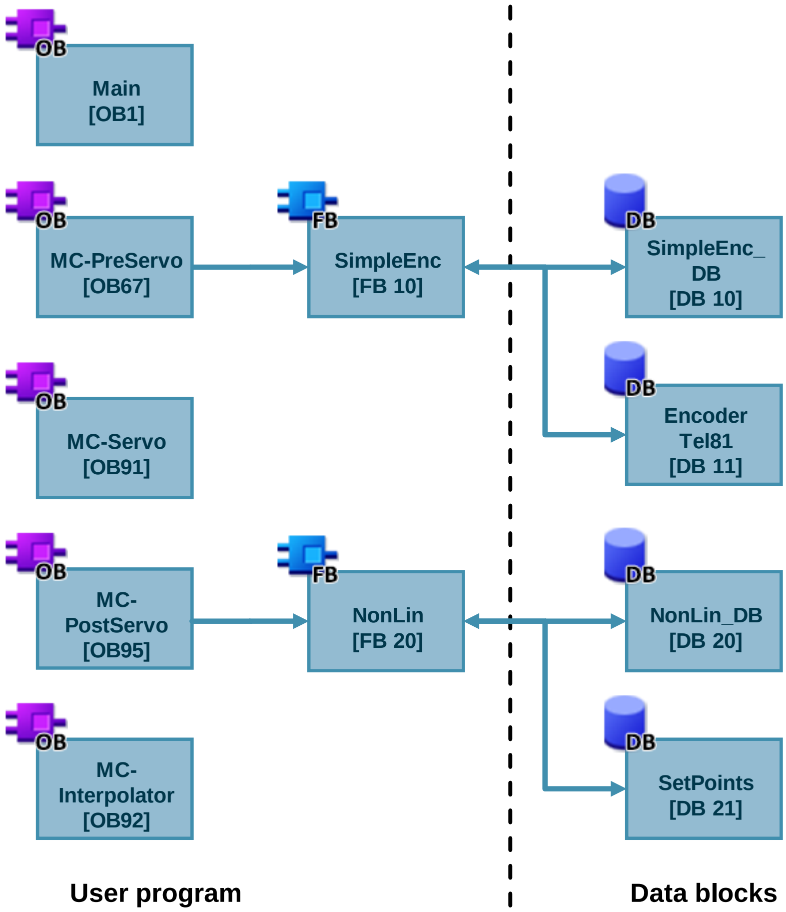
Organization block MC-PostServo

The position controller in the MC-Servo OB calculates a speed set point that is required for the positioning of the drive. In this application example, the speed set point is manipulated such that the course of the set point corresponds to a predefined characteristic (see chapter 4.3).

Program overview

The TIA program V14 contained in this application example has the following structure:

Figure 4-2: Program overview



4.2 Functionality of MC-PreServo

The task of the organization block MC-PreServo is to provide the encoder data in the structure of the standard telegram 81. In this application example, this task is fulfilled by the “SimpleEnc” function block.

Note A list of the status information of the MC-PreServo organization block can be found in the TIA Portal V14 Online Help.

4.2.1 Standard telegram 81

For the provision of the encoder data, the technology object uses the standard telegram 81. It contains all parameters that are required for the evaluation and control of a position encoder.

Table 4-1: Structure of standard telegram 81

PZD 1	PZD 2	PZD 3	PZD 4	PZD 5	PZD 6
STW2	G1_STW	-	-	-	-
ZSW2	G1_ZSW	G1_XIST1		G1_XIST2	

Table 4-2: Transmission telegram to the drive

Parameter	Meaning
STW2	Control word 2
G1_STW	Encoder 1 control word

Table 4-3: Receive telegram from the drive

Parameter	Meaning
ZSW2	Status word 2
G1_ZSW	Encoder 2 status word
G1_XIST1	Encoder 1 Actual position value 1
G1_XIST2	Encoder 1 Actual position value 2

4.2.2 “SimpleEnc” block

The “SimpleEnc” function block is called in the MC-PreServo OB and provides the encoder information for the position controller that is done in the MC-PreServo OB. In this, the encoder value is read in as 32-bit integer and output in the data structure of the standard telegram 81.

The “SimpleEnc” function block has the following tasks:

- Checking the read in encoder value for overflows.
- Detecting and providing the actual position value 1 (G1_XIST1).
- Generating and acknowledging sensor errors.
- Providing the actual position value 2 (G1_XIST2) upon request.

Checking the read in encoder value for overflows

The example described here shows how to integrate an analog encoder. These encoders operate within a limited value range. However, the “SimpleEnc” block is prepared to use other encoder types as well.

The position information could, for example, come from a 16-bit counter and the pulse generator connected to it could continually rotate in the same direction. In this case, the counter value would overflow after 2^{16} pulses and start from zero again. Using the position information from the encoder, the “SimpleEnc” FB generally calculates a continuously running 32-bit value (G1_XIST1) which counts upwards during forward rotation and downwards during backward rotation.

To be able to detect overflows of the encoder value, the “SimpleEnc” function block needs to be informed about the size (number of bits) of the read in encoder value. This is done via the “numberOfBits” block input.

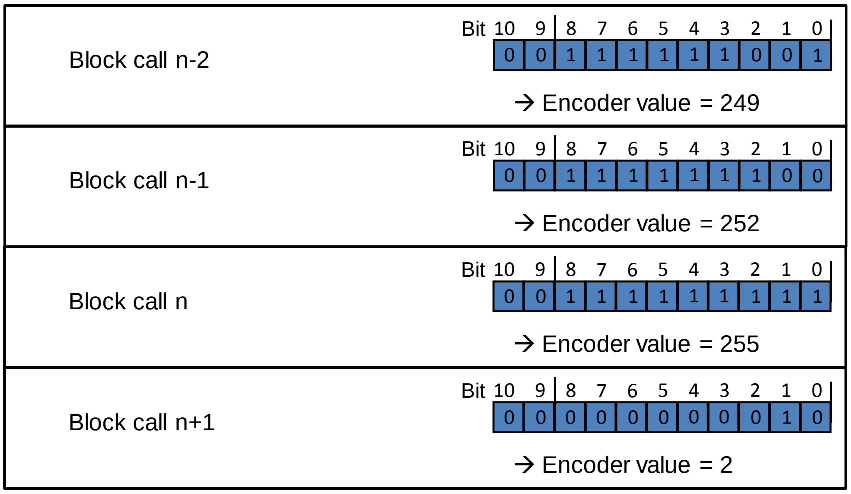
The size of the encoder value determines the maximum value that can be displayed. The “SimpleEnc” function block now detects an overflow of the encoder value, if the change in the encoder value between two calls exceeds half of the maximum value that can be displayed.

Example:

The encoder value has a size of 8 bit. The resulting maximum value that can be displayed is 255. Therefore, the value of 256 cannot be displayed.

The “SimpleEnc” function block now detects an overflow of the 8-bit encoder value, if the read in encoder value increases or decreases by more than the value of 128 between two calls.

Figure 4-3: 8-bit encoder value with overflow



Between the block calls n and n+1, the “SimpleEnc” function block now detects an overflow of the encoder value, since the encoder value has decreased by more than 128. In this case, the result is that the read in encoder value needs to be added up by the value of 256 in order to calculate the actual change in the position in the last cycle. This position change is added up to the actual value of G1_XIST1.

Detection and provision of actual position value 1 (G1_XIST1)

The encoder value is passed on to the “SimpleEnc” function block as 32-bit integer and then output by the block with the double word data type as actual position value 1 (G1_XIST1). In this, the block identifies the change in the encoder value at each call and adjusts the actual position value accordingly.

Generation and acknowledgment of sensor errors

If the encoder control word requests functions that are not supported, the “SimpleEnc” block generates a sensor error. An active sensor error is displayed in the encoder status word and the error code 16#0F01 is entered into actual position value 2 (G1_XIST2).

To generate and acknowledge encoder errors, the following entries of the encoder control word are evaluated or entries of the encoder status word are made.

Table 4-4: Control word G1_STW

<table>
<tr><th>Bit</th><th>Function</th><th>Comment</th></tr>
<tr><td>4</td><td rowspan="3">Start/stop/read selected function
(depends on the drive used)</td><td rowspan="4">These functions are not supported by the “SimpleEnc” block. A sensor error is generated when the signal status is TRUE</td></tr>
<tr><td>5</td></tr>
<tr><td>6</td></tr>
<tr><td>14</td><td>Request parking encoder</td></tr>
<tr><td>13</td><td>Request absolute value cyclically</td><td>If an incremental encoder value (“isAbsolute” input = FALSE) is used, the signal status TRUE causes a sensor error.</td></tr>
<tr><td>15</td><td>Acknowledge sensor errors</td><td>Acknowledgement of the sensor error</td></tr>
</table>

Table 4-5: Status word G1_ZSW

Bit	Function	Comment
15	Sensor error	Specifies a sensor error. The error code 16#0F01 is transferred to G1_XIST2.

Provision of actual position value 2 (G1_XIST2) upon request

If you use an absolute value encoder during the control start-up, the technology object requests the absolute position value. This value is entered into the actual position value 2 (G1_XIST2).

For the provision of the actual position value 2, the following entries of the encoder control word are evaluated or entries of the encoder status word are made.

Table 4-6: Control word G1_STW

Bit	Function	Comment
13	Request absolute value cyclically	Request cyclic transmission of absolute position value

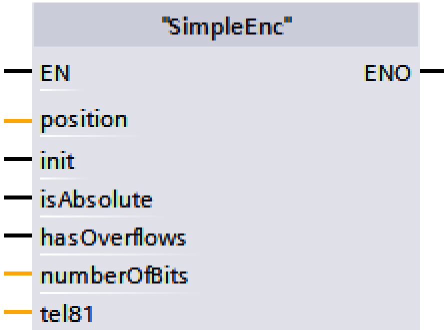
Table 4-7: Status word G1_ZSW

Bit	Function	Comment
13	Transfer absolute value cyclically	The absolute position value is transferred to G1_XIST2.

Block call

Figure 4-4 below shows the block call of the “SimpleEnc” function block in the MC-PreServo OB.

Figure 4-4: “SimpleEnc” block call



Block parameters

The parameters of the “SimpleEnc” function block are listed below.

Table 4-8: Input parameter

Parameter	Data type	Start value	Function
position	DINT	0	Reading in the encoder value
init	BOOL	FALSE	Is activated one-time at the first call up of the block.
isAbsolute	BOOL	FALSE	The value at the “position” input has the following statuses: FALSE = incremental value TRUE = absolute value
hasOverflows	BOOL	FALSE	Encoder value may have an overflow
numberOfBits	UINT	0	Number of bits of the encoder value

Table 4-9: In/out parameters

Parameter	Data type	Start value	Function
tel81	“PD_TEL81“	-	Data block for storing the encoder data (see chapter 4.2.3)

4.2.3 “EncoderTel81” data block

The “EncoderTel81” data block serves as interface between the “SimpleEnc” block and the processing of the technology object in the MC-Servo OB. In this, the standard telegram 81 is emulated in the “EncoderTel81” data block. That is why data block and standard telegram have the same structure.

Note

> The structure of the “EncoderTel81” data block needs to correspond to the structure of the standard telegram 81. Therefore, do not configure the “EncoderTel81” data block with an optimized block access.

Figure 4-5: “EncoderTel81” data block

EncoderTel81

	Name	Data type	Offset
1	Static		
2	Tel81	PD_TEL81	0.0
3	Input	PD_TEL81_IN	0.0
4	ZSW2_ENC	PD_ZSW2_ENC	0.0
5	G1_ZSW	PD_Gx_ZSW	2.0
6	G1_XIST1	DWord	4.0
7	G1_XIST2	DWord	8.0
8	Output	PD_TEL81_OUT	12.0
9	STW2_ENC	PD_STW2_ENC	12.0
10	G1_STW	PD_Gx_STW	14.0

The “EncoderTel81” data block is the data connection of the encoder and is thus configured in the technology object. This configuration is described in chapter 5.2.2.

4.3 Functionality of MC-PostServo

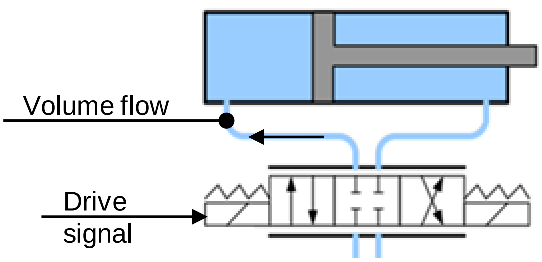
The MC-PostServo organization block (OB) is called up right after the MC-Servo OB. Thus, a program stored in the MC-PostServo OB offers the possibility to access the speed set point of the position controller. Position control is done in the “Positioning axis” technology object in the MC-Servo OB.

Note A list of the status information of the MC-PostServo organization block can be found in the TIA Portal V14 Online Help.

4.3.1 Characteristic curve of a hydraulic valve

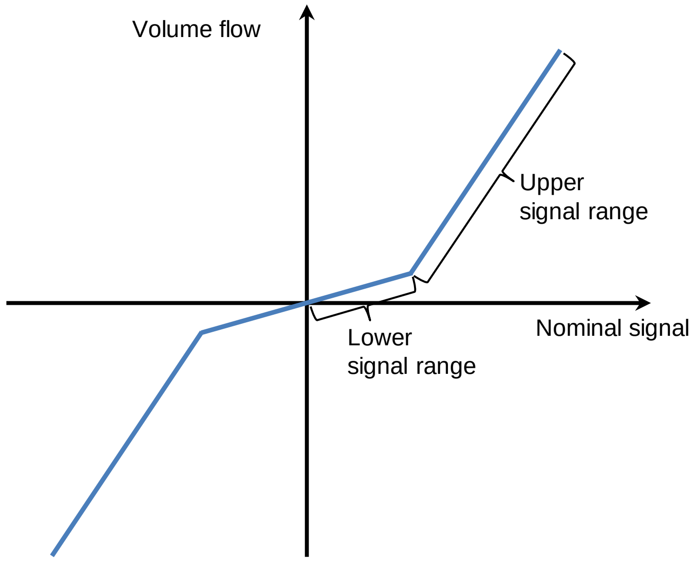
Hydraulic valves are a typical application regarding a change in the set point of the position controller. In this respect, there is a non-linear relationship between the volume flow and the control signal (set point).

Figure 4-6: Setpoint at a hydraulic valve



These valves with a bent characteristic fulfill the requirement for a greater resolution in the lower signal range and sufficient volume flow in the upper signal range.

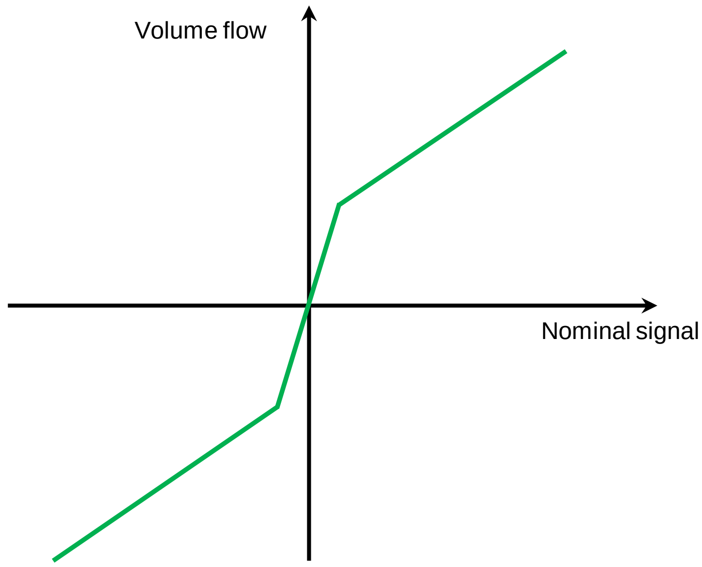
Figure 4-7: Pattern of a bent valve characteristic



4.3.2 Linearization of characteristic curve

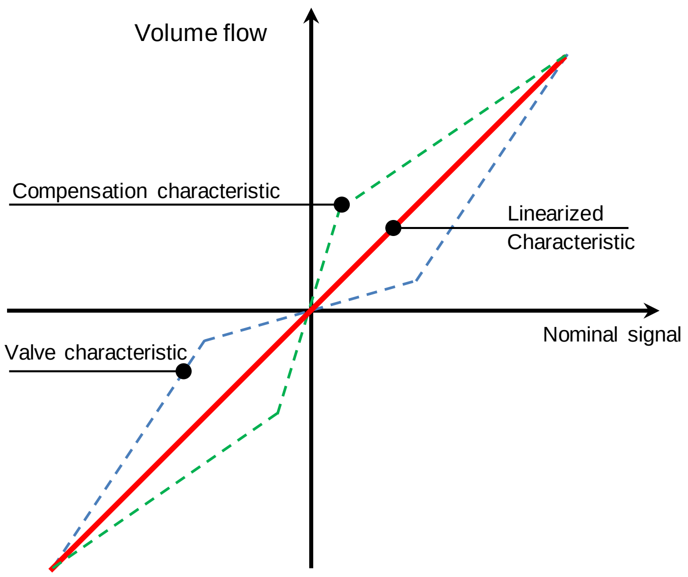
The position controller of the technology object is designed for linear control system behavior such as applies in the case of electrical axes. If a hydraulic valve is supposed to be controlled, you need to compensate the non-linear characteristic of the valves with an inverse characteristic for the set point. By this, a linearization of the characteristic curve is performed for the position control in the technology object. Such a characteristic, one that is inversely related to the curve of the valve characteristic, is called compensation characteristic.

Figure 4-8: Compensation characteristic



The curve of the compensation characteristic is specified with the “NonLin” function block in form of a support point table (see chapter 4.3.3).

Figure 4-9: linearized characteristic



By linearizing the control system behavior, the doubling of the set point generated by the position controller now results in a doubling of the speed of the hydraulic cylinder’s position change.

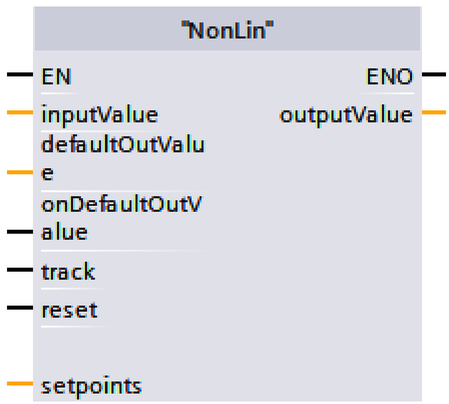
4.3.3 “NonLin” block

The “NonLin” block realizes a characteristic. This characteristic is specified via a support point table with a linear interpolation between the support points. In every cycle, a specified input value generates an output value on the basis of the characteristic from the support point table.

Block call

Figure 4-10 below shows the block call of the “NonLin” function block in the MC-PostServo OB.

Figure 4-10: “NonLin” call up



Block parameters

The parameters of the “NonLin” function block are listed below.

Table 4-10: Input parameter

Parameter	Data type	Start value	Function
inputValue	REAL	0.0	Input value for the calculation of the output value via the defined characteristic.
defaultOutValue	REAL	0.0	Default output value without the use of the characteristic.
onDefaultOutValue	BOOL	FALSE	Enabling the default output value
track	BOOL	FALSE	Direct forwarding of input value to output value without the use of the characteristic.
reset	BOOL	FALSE	Resetting the block.

Table 4-11: Output parameters

Parameter	Data type	Start value	Function
outputValue	REAL	0.0	Output value has been calculated via the defined characteristic on the basis of the input value.

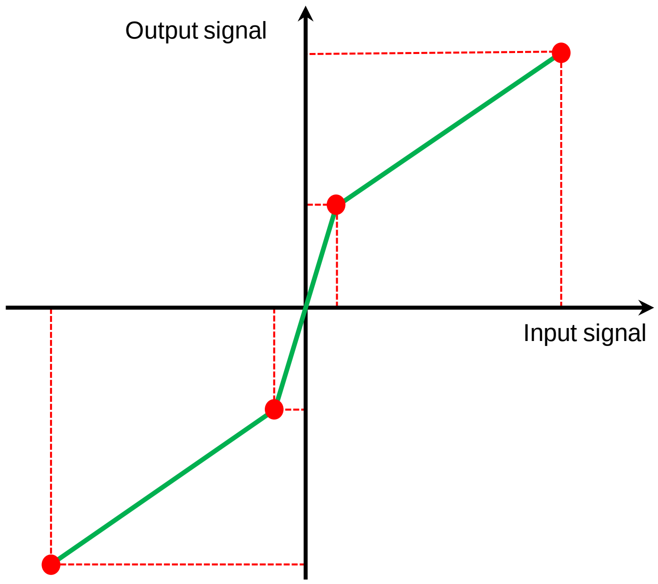
Table 4-12: In/out parameters

Parameter	Data type	Function
setpoints	typeNonLinSetpoint	Support point table for the definition of the character.

The “NonLin” block calculates an output value on the basis of the input value and the specified characteristic. In this, the following specifics are considered:

- As long as the “onDefaultOutValue” input is set, the value defined via the “defaultOutValue” parameter will be output as output value.
- As long as the “reset” input is set, the search index within the characteristic will be reset and the value of 0.0 will be output as output value.
- As long as the “track” input is set, the input value is directly output as output value, without taking into account the characteristic.
- On the basis of the input value, a characteristic value is calculated via the linearly interpolated support point table and output as output value.
  - If the input value is located between two support points within the support point table, the output value is calculated as intersection with the connection line between the previous and the next support point.
  - If the input value is located before the first support point (lowest value that has been defined in the support point table), the output value is calculated as intersection of the line that is formed by the first two support points of the support point table.
  - If the input value is located after the last support point (highest value that has been defined in the support point table), the output value is calculated as intersection of the line that is formed by the last two support points of the support point table.

Figure 4-11: Exemplary course of the output signal



The course of the output signal of the “NonLin” block corresponds to the compensation character (see chapter 4.3.2).

4.3.4 “SetPoints” data block

The support point table is realized by a tag of the Array data type. The type of the Array corresponds to the “typeNonLinSetpoint” PLC data type.

You can create the support point table in any global data block. The size of the Array depends on the number of support points.

Figure 4-12: “SetPoints” data block

SetPoints

	Name	Data type	Start value
1	Static		
2	NonLinSetpoint	Array[0..4] of "typeNonLinSetpoint"	
3	NonLinSetpoint[0]	"typeNonLinSetpoint"	
4	InputValue	Real	-2000.0
5	OutputValue	Real	-2200.0
6	NonLinSetpoint[1]	"typeNonLinSetpoint"	
7	InputValue	Real	-200.0
8	OutputValue	Real	-400.0
9	NonLinSetpoint[2]	"typeNonLinSetpoint"	
10	InputValue	Real	0.0
11	OutputValue	Real	0.0
12	NonLinSetpoint[3]	"typeNonLinSetpoint"	
13	InputValue	Real	200.0
14	OutputValue	Real	400.0
15	NonLinSetpoint[4]	"typeNonLinSetpoint"	
16	InputValue	Real	2000.0
17	OutputValue	Real	2200.0

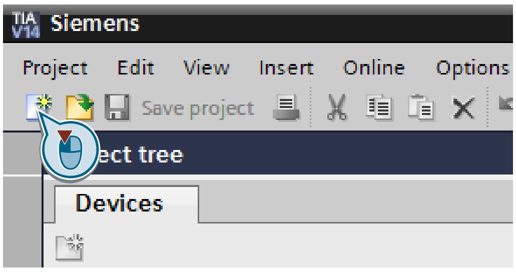
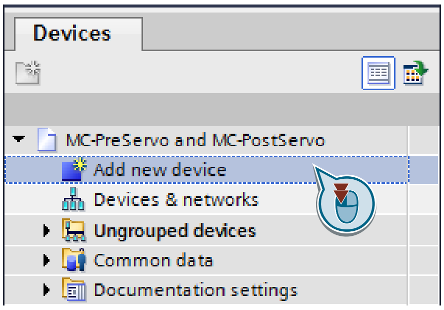
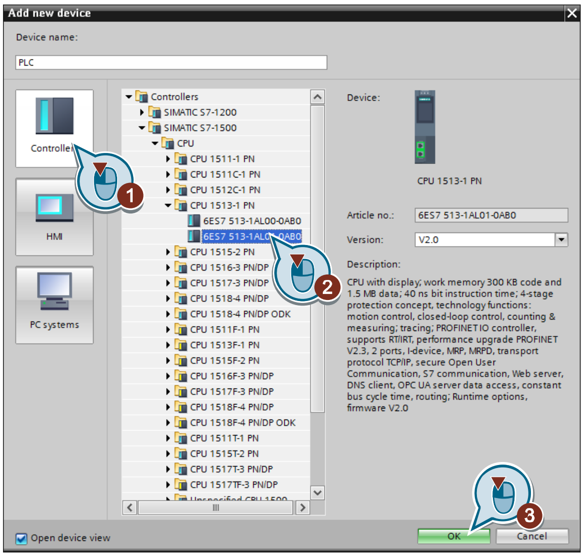
5 Configuration and Settings

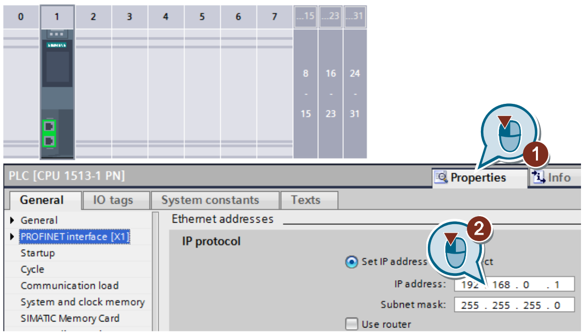
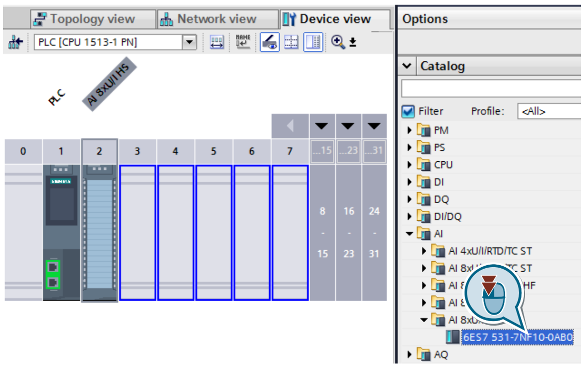
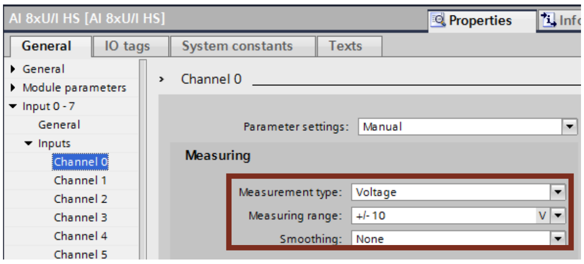
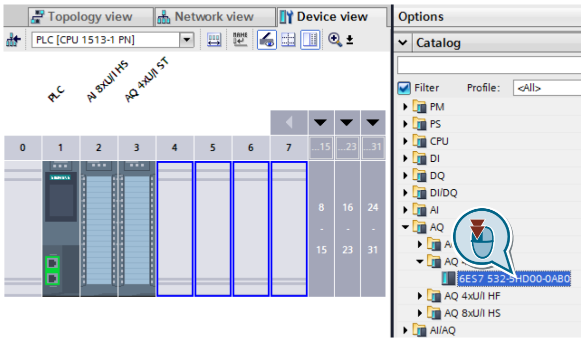
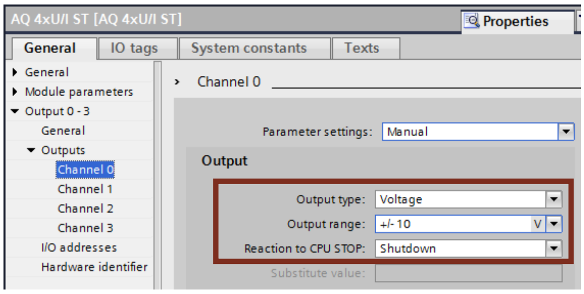
The following step tables show you how to configure the SIMATIC S7 CPU.
A requirement is that the software has been installed on your PG/PC according to Table 2-2.

Note The procedure described in the following step tables is one possibility to configure SIMATIC S7-1500 controller. TIA Portal offers several possible solutions that differ to a greater or lesser degree from the procedure shown here.

5.1 Configuring the S7-1500 station

Table 5-1: Configuring the controller

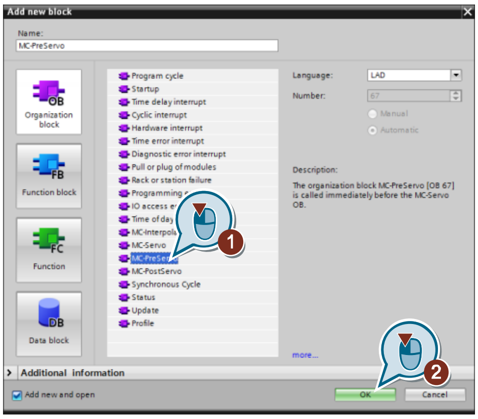
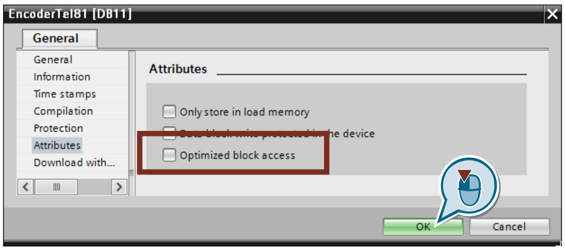
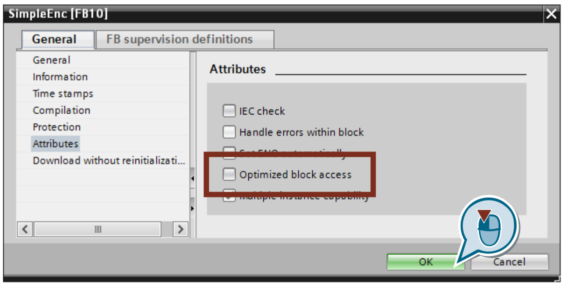
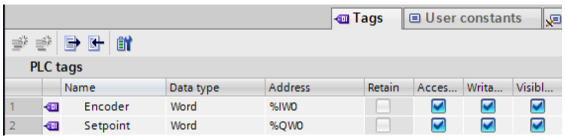
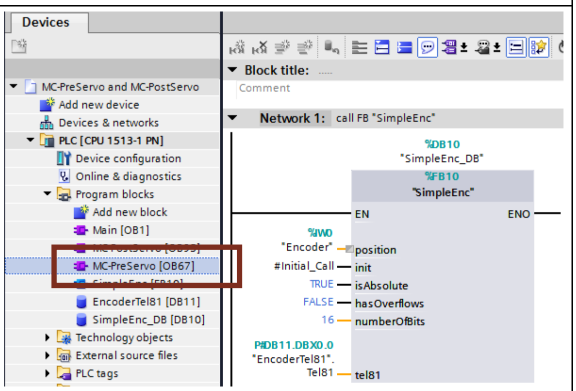
No.	Action	Remark
1.	Open TIA Portal and create a new project.	
2.	Double-click on “Add new device“.	
3.	Add your desired controller: 1. Select “Controller”. 2. Select the desired CPU. Make sure that the version of the selected CPU is V2.0 or higher. 3. Then click on “OK”.	

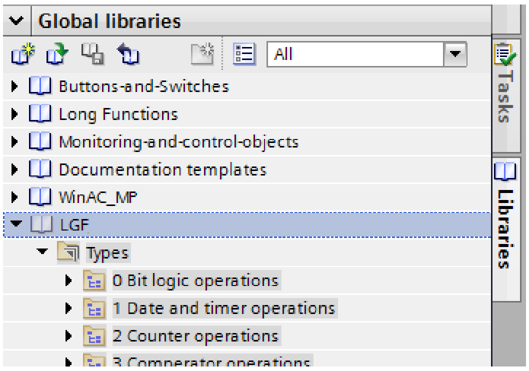
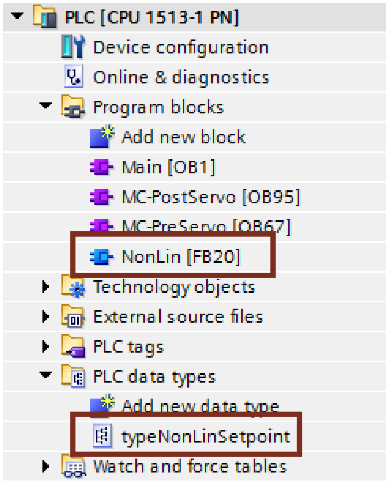
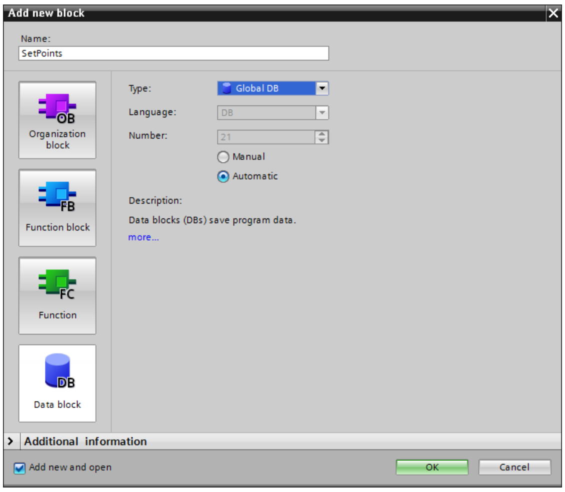
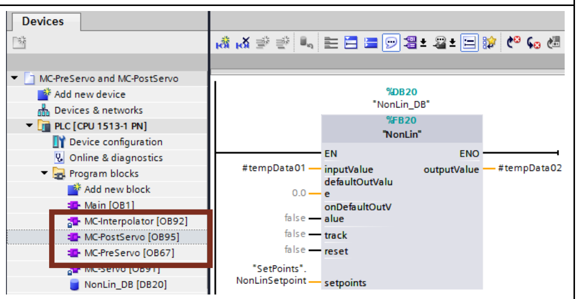
No.	Action	Remark
4.	Open the device configuration of the CPU and configure the PROFINET interface. 1. Open the “Properties” of the CPU. 2. Select an “Ethernet address”.	
5.	In the “Device view”, add the analog input module used.	
6.	Under the properties of the module, configure a channel for the voltage measurement.	
7.	Then add the analog output module used.	
8.	Under the properties of the module, configure a channel for a voltage output of +/- 10V.	
9.	Save the configuration.	-

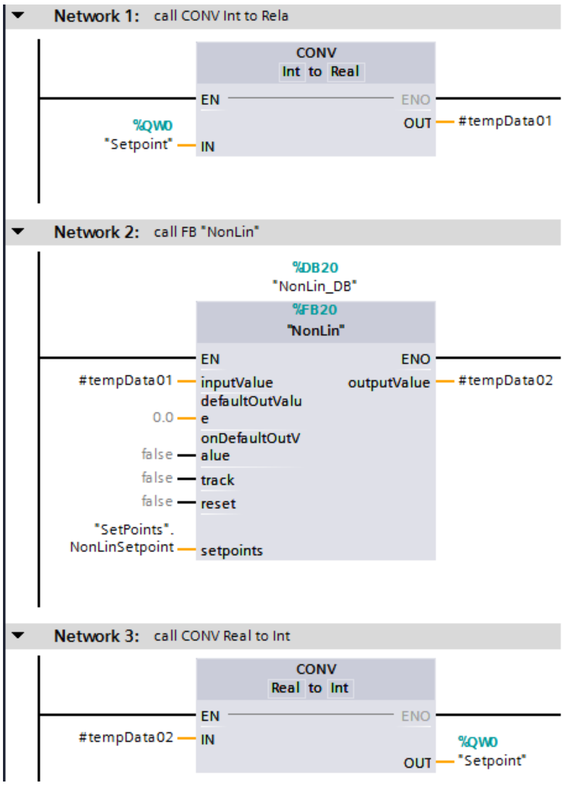
5.2 Generating the S7 program

5.2.1 Adding and configuring the blocks

Table 5-2: Configuring the S7 program

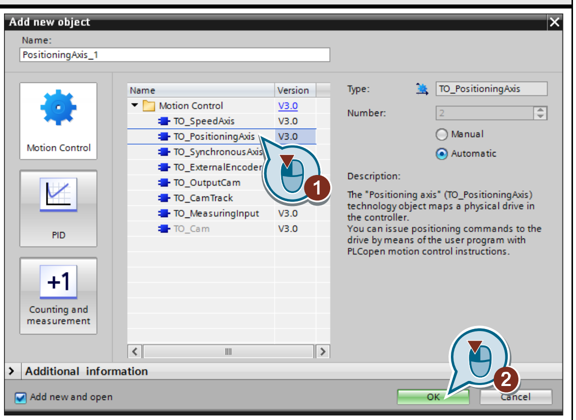
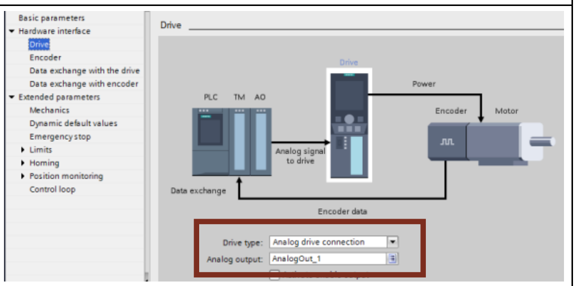
No.	Action	Remark
1.	Add the MC-PreServo and MC-PostServo organization blocks into the controller program. 1. Select the according organization block. 2. Confirm with "OK".	
2.	Add a data block to the program and structure it as described in chapter 4.2.3 . Deactivate the attribute "Optimized block access" in the properties of the data block.	
3.	Add a function block to the program and configure this block as described in chapter 4.2.2 (SimpleEnc). Deactivate the attribute "Optimized block access" in the properties of the function block.	
4.	Add the tags for the Encoder and the Setpoint.	
5.	Call up the function block created in step 3 in the MC-PreServo OB.	

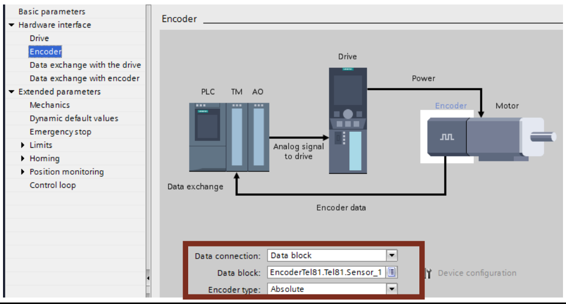
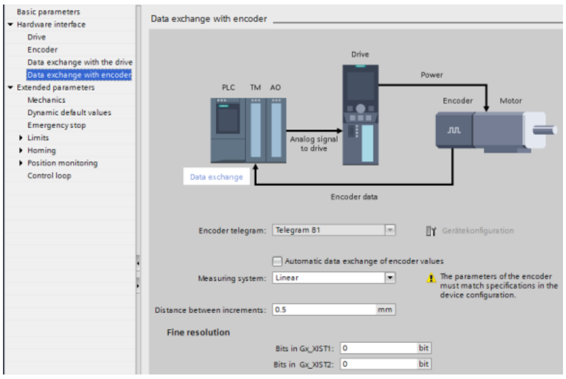
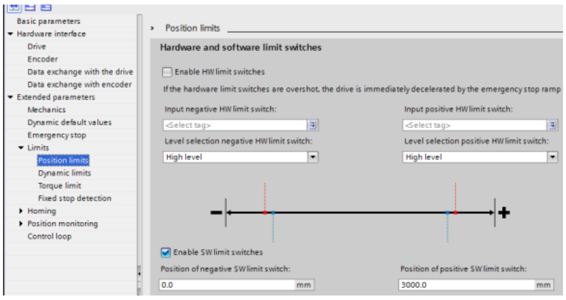
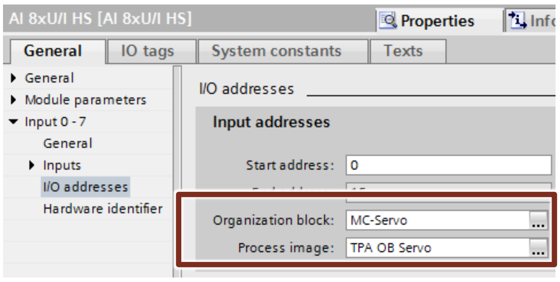
No.	Action	Remark
6.	Open the library with general functions (LGF) which is available as download under \5\ .	
7.	Add the “NonLin” function block and the “typeNonLinSetpoint” PLC data type from the LGF library to the program. Configure the block as described in chapter 4.3.3	
8.	Add a global data block to the program. Configure the data block as support point table as shown in chapter 4.3.4 .	
9.	Call up the “NonLin” function block in the MC-PostServo OB.	

No.	Action	Remark
10.	The “NonLin” block works with tags from the REAL data type. To be able to change the drive’s speed set point (specified via analog output with INT data type) with the block, you need to convert the set point into the REAL data type.	
11.	Save the configuration.	-

5.2.2 Configuring the technology object

Table 5-3: Configuring the technology object

No.	Action	Remark
1.	Add the “TO_PositioningAxis” to the controller project. 1. Select the “Positioning axis” technology object. 2. Then click on “OK”.	
2.	For the basic parameters of the drive, enter “Analog drive connection” as “Drive type”. Select the “Analog output” entry for the analog output of the controller with which the speed set point is to be output to the drive.	

No.	Action	Remark
3.	When configuring the encoder, you need to select the entry “data block” as “Data connection”. Enter the data block into this entry, in which the standard telegram 81 is emulated. As “Encoder type”, enter “Absolute”.	
4.	In “Data exchange with encoder”, you need to configure a linear measuring system. The value zero applies for the fine resolution.	
5.	To prevent mechanical damage to the drive system, you can configure “Position limits”.	
6.	After setting the technology objects, the analog input module and output module must be assigning to the MC-Servo.	

Note A description on how to configure the „Positioning axis“ technology object can be found in the function manual “S7-1500 Motion Control in TIA Portal“ \3\.

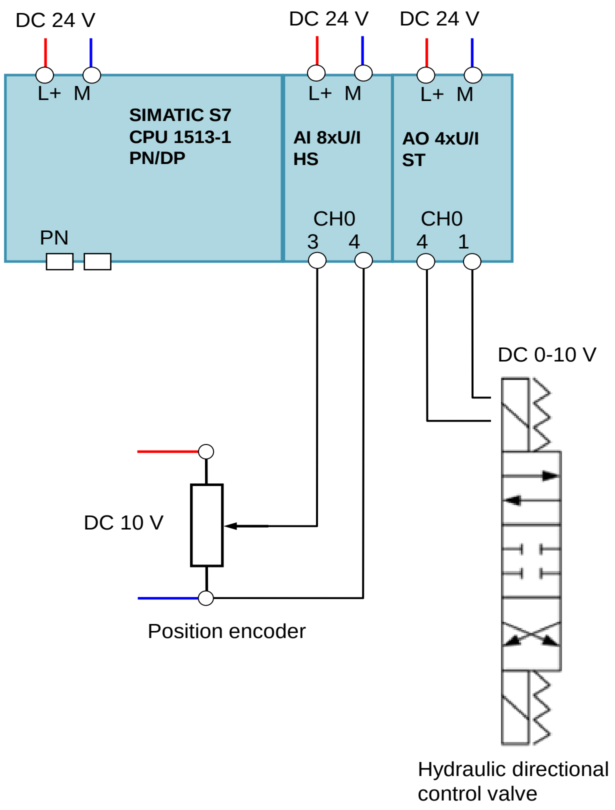
https://support.industry.siemens.com/cs/de/en/view/109739589

6 Installation and Commissioning

6.1 Installing the hardware

Figure 6-1 shows the hardware setup of the application.

Figure 6-1: Hardware setup



6.2 IP addresses and device names

Table 6-1: IP addresses and device names

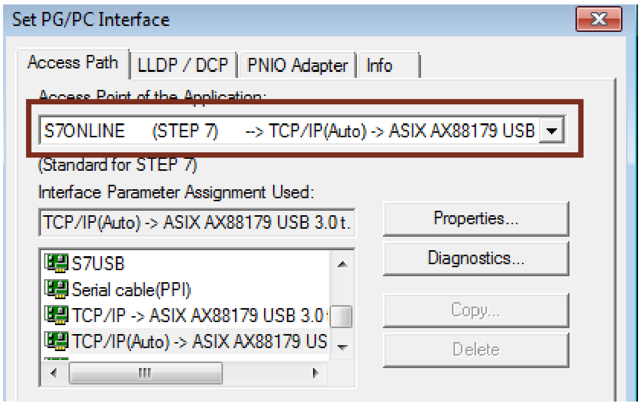
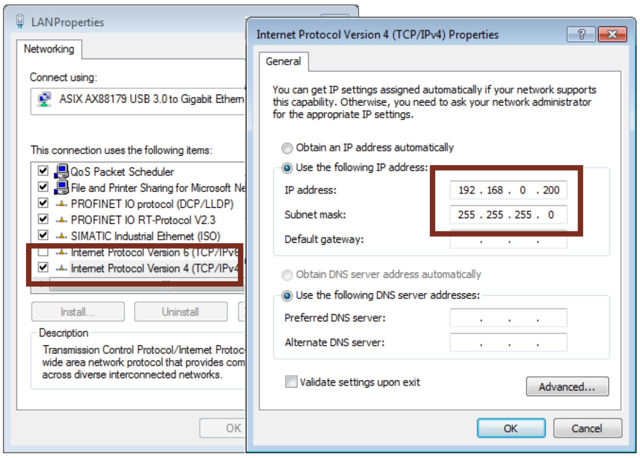
Components	Device name	IP addresses
SIMATIC S7-1513-1 PN/DP	PLC	192.168.0.1
PG/PC	-	192.168.0.200

The network mask is always 255.255.255.0 and no router is used.

6.3 PG/PC settings

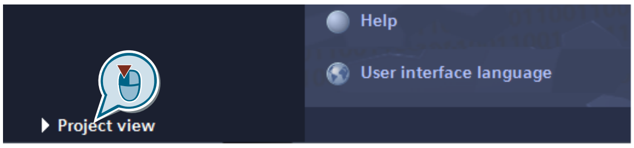
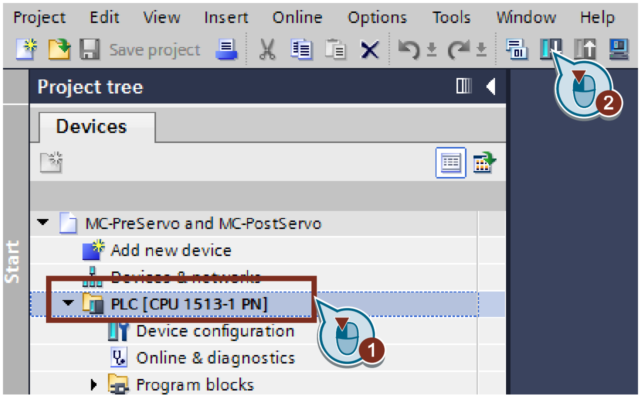
To create a connection between the components of the application example and your development system (PG/PC), you need to assign a fixed IP address to the network card used in the PG/PC. These IP addresses are set in the control panel of the PG/PC.

Table 6-2: PG/PC settings

No.	Action	Remark
1.	In the Control Panel, set PG/PC Interface. Select “S7ONLINE (STEP 7)” as the application's access point and “TCP/IP -> Network card used” as interface parameterization.	
2.	Assign a free, fixed IP address 192.168.0.x to the network card used (e.g. x = 200) and assign the subnet mask 255.255.255.0. With these settings, you can reach the components of the application example.	

6.4 Downloading the project to the components

Table 6-3: Downloading to the components

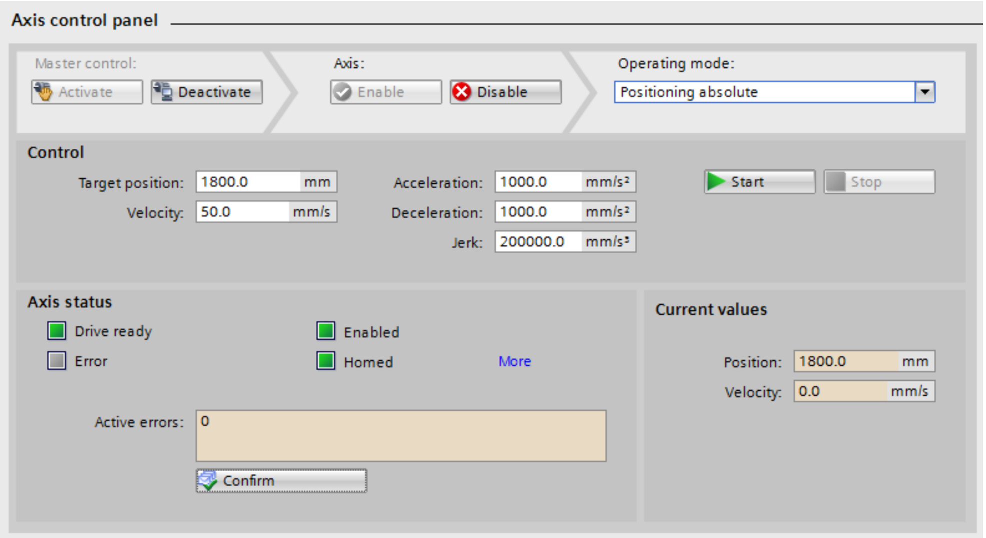
No.	Action	Remark
1.	Retrieve the project contained in the zip file “109741575_MC_PreServo_and_MC_PostServo_v10.zip” to a local directory.	-
2.	Double-click on the ap14-file in the project folder just retrieved in order to open the project in TIA Portal.	-
3.	If TIA Portal opens in the Portal view, go to the bottom left to switch to the Project view.	
4.	Load the program of the SIMATIC controller: 1. Select the S7 controller in the project tree. 2. Load the project into the controller.	

7 Operation

WARNING	**Make sure that no persons or system components are endangered by the moving drive.** Take appropriate measures to prevent the drive from exceeding technical or mechanical limits, if required.

The axis control panel is available for operating the axis: It is a clearly structured user interface for referencing and positioning the drive.

Figure 7-1: Axis control panel



There is no user program required for operating the axis control panel. Via the axis control panel, you maintain control over a technology object and control the axis' movements.

Note

A description on the function and structure of the axis control panel can be found in the function manual "S7-1500 Motion Control in TIA Portal" \3\.

https://support.industry.siemens.com/cs/de/en/view/109739589

8 Links & Literature

Table 8-1: Reference

	Topic
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Download page of the entry https://support.industry.siemens.com/cs/ww/en/view/109741575
\3\	"SIMATIC S7-1500 S7-1500 Motion Control V3.0 in TIA Portal V14" Function Manual https://support.industry.siemens.com/cs/de/en/view/109739589
\4\	"SIMATIC S7-1500 S7-1500T Motion Control V3.0 in TIA Portal V14" Function Manual https://support.industry.siemens.com/cs/de/en/view/109481326
\5\	Library with general functions (LGF) for STEP 7 (TIA Portal) https://support.industry.siemens.com/cs/de/en/view/109479728

9 History

Table 9-1: History

Version	Date	Modifications
V1.0	02/2017	First version
V 1.0.1	04/2018	Extension the table 5-3