

SIEMENS

Ingenuity for life

Industry Online Support

Home

"LAxisBasics" Easy Control of the "Axis" TO in the SIMATIC S7-1500

SIMATIC S7-1500(T) / TIA Portal V15.1

<https://support.industry.siemens.com/cs/ww/en/view/109749348>

Siemens
Industry
Online
Support



Legal information

Use of application examples

Application examples illustrate the solution of automation tasks through an interaction of several components in the form of text, graphics and/or software modules. The application examples are a free service by Siemens AG and/or a subsidiary of Siemens AG ("Siemens"). They are non-binding and make no claim to completeness or functionality regarding configuration and equipment. The application examples merely offer help with typical tasks; they do not constitute customer-specific solutions. You yourself are responsible for the proper and safe operation of the products in accordance with applicable regulations and must also check the function of the respective application example and customize it for your system.

Siemens grants you the non-exclusive, non-sublicensable and non-transferable right to have the application examples used by technically trained personnel. Any change to the application examples is your responsibility. Sharing the application examples with third parties or copying the application examples or excerpts thereof is permitted only in combination with your own products. The application examples are not required to undergo the customary tests and quality inspections of a chargeable product; they may have functional and performance defects as well as errors. It is your responsibility to use them in such a manner that any malfunctions that may occur do not result in property damage or injury to persons.

Disclaimer of liability

Siemens shall not assume any liability, for any legal reason whatsoever, including, without limitation, liability for the usability, availability, completeness and freedom from defects of the application examples as well as for related information, configuration and performance data and any damage caused thereby. This shall not apply in cases of mandatory liability, for example under the German Product Liability Act, or in cases of intent, gross negligence, or culpable loss of life, bodily injury or damage to health, non-compliance with a guarantee, fraudulent non-disclosure of a defect, or culpable breach of material contractual obligations. Claims for damages arising from a breach of material contractual obligations shall however be limited to the foreseeable damage typical of the type of agreement, unless liability arises from intent or gross negligence or is based on loss of life, bodily injury or damage to health. The foregoing provisions do not imply any change in the burden of proof to your detriment. You shall indemnify Siemens against existing or future claims of third parties in this connection except where Siemens is mandatorily liable.

By using the application examples you acknowledge that Siemens cannot be held liable for any damage beyond the liability provisions described.

Other information

Siemens reserves the right to make changes to the application examples at any time without notice. In case of discrepancies between the suggestions in the application examples and other Siemens publications such as catalogs, the content of the other documentation shall have precedence.

The Siemens terms of use (<https://support.industry.siemens.com>) shall also apply.

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 constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place. For additional information on industrial security measures that may be implemented, please visit <https://www.siemens.com/industrialsecurity>.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed at: <http://www.siemens.com/industrialsecurity>.

Table of Contents

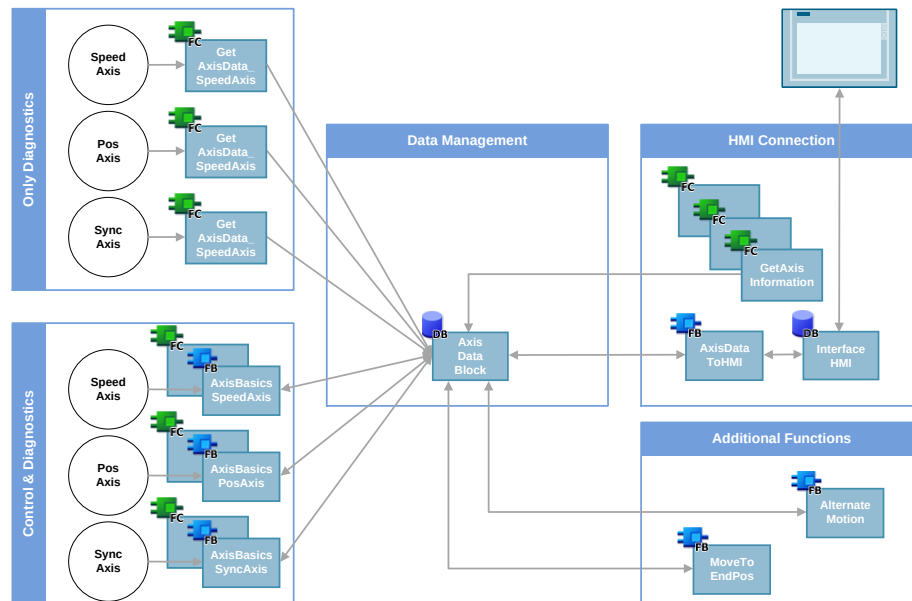
Legal information	2
1 Introduction	4
1.1 Overview.....	4
1.1.1 Axis diagnostics.....	4
1.1.2 Axis diagnostics and control.....	5
1.1.3 Central data management.....	5
1.1.4 Additional functions	7
1.1.5 HMI user interface	7
1.2 How the application example works	7
1.2.1 Monitoring and control of the axes	7
1.2.2 Using the additional functions	8
1.2.3 Central data management.....	8
1.2.4 HMI user interface	8
1.3 Components used	9
2 Engineering	10
2.1 Integration into the user project.....	10
2.1.1 Program blocks	10
2.1.2 HMI user interface	10
2.1.3 Using some of the included program blocks	10
2.1.4 Using single pages or objects of the HMI user interface.....	11
2.2 Description of the blocks and data types	12
2.2.1 "LAxisBasics_AxisDataBlock" DB	12
2.2.2 "LAxisBasics_GetAxisData" FC	17
2.2.3 "LAxisBasics_AxisBasics" FB	18
2.2.4 "LAxisBasics_AxisDataToHMI" FB / "LAxisBasics_InterfaceHMI" DB / "LAxisBasics_GetAxisInformation" FC	21
2.2.5 "LAxisBasics_MoveToEndPos_Axis" FB	23
2.2.6 "LAxisBasics_AlternateMotion_Axis" FB.....	26
2.3 Description of the HMI user interface.....	28
2.3.1 Structure of the HMI project	28
2.3.2 Structure of the HMI operating screens	29
2.3.3 State screen	30
2.3.4 Operating screens of the functions	31
2.3.5 Diagnostic screens.....	34
2.3.6 HMI settings screen.....	37
2.3.7 Slide-in screen for operating page selection	38
3 Appendix	39
3.1 Service and Support.....	39
3.2 Links and literature	40
3.3 Change documentation	41

1 Introduction

1.1 Overview

The "LAxisBasics" block library provides function blocks for integration into your own user programs; they allow you to easily diagnose and control the "Axis" motion control technology object.

Figure 1-1 "LAxisBasics" block library overview



1.1.1 Axis diagnostics

The "LAxisBasics" block library contains functions (FCs) that allow you to cyclically read out the following data of an axis:

- Setpoint and actual position of the axis.
- Setpoint and actual values of the dynamic response parameters of the axis such as velocity and acceleration.
- Status of the axis by reading out the following status information:
 - Status word
 - Warning word
 - Error word
 - Additional error detail information
- Status of the encoder of the axis
- Unit settings of the parameter values of the axis

Note

The functions (FCs) listed here are an alternative to the AxisBasics function blocks listed in the next chapter if you only want to diagnose but not control the appropriate axes.

1.1.2 Axis diagnostics and control

The "LAxisBasics" block library contains function blocks (FBs) that allow you to control an axis and cyclically read out the axis data.

The function blocks of the "LAxisBasics" block library support the following basic functions of the "Axis" motion control technology objects:

- MC_Power:
Enable axis / supply axis with the required power.
- MC_Reset:
Acknowledge warnings and errors on the axis. In addition, this function allows you to restart the technology object.
- MC_Home (not for axis type speed axis):
Home the technology object with the modes available for the appropriate axis, for example:
 - 0: Direct homing (absolute)
 - 1: Direct homing (relative)
 - 2: Passive homing
 - 3: Active homing
 - 5: Active homing (<TO>.Homing.HomePosition)
 - 6: Absolute encoder adjustment (relative)
 - 7: Absolute encoder adjustment (absolute)
 - 8: Passive homing
 - 9: Abort passive homing
 - 10: Passive homing (<TO>.Homing.HomePosition)
- MC_MoveJog:
Manually move the technology object in jog mode with separate inputs for the positive and negative direction.

Note

The function blocks (FBs) for controlling the axis already include the functions (FCs) for diagnosing an axis listed in the previous chapter.

1.1.3 Central data management

The "LAxisBasics" block library contains a global data block (DB) where the data for diagnosing and controlling the axes required for the user program can be saved.

The global data block contains an array where each array element represents an axis. Using the PLC data type of the global data block, the number of array elements can be customized to the number of axes in the user program.

Figure 1-2 Global data block of the "LAxisBasics_AxisDataBlock" axis data

Axis	Array[0..1] of "typeAxisBasicsData"		☒	☒	☒	☒	Array of all axes in
Axis[0]	"typeAxisBasicsData"		☒	☒	☒	☒	Array of all axes in
Axis Information	"typeAxisInformation"		☒	☒	☒	☒	Axis information (tr
TypeOfAxis	Int	0	☒	☒	☒	☒	0: No axis / 1: Spee
AxisName	String	"	☒	☒	☒	☒	Write in the name i
AxisDescription	String	"	☒	☒	☒	☒	Write in an addition
Axis Data	"typeAxisData"		☒	☒	☒	☒	Actual axis data
Axis Positions	"typeAxisPositions"		☒	☒	☒	☒	Axis positions
Axis Status	"typeAxisStatus"		☒	☒	☒	☒	Axis status
Axis Control	"typeAxisControl"		☒	☒	☒	☒	Axis movement via l
AxisControlCommand	"typeAxisControlCommand"		☒	☒	☒	☒	Command interfac
AxisControlStatus	"typeAxisControlStatus"		☒	☒	☒	☒	Status interface fro
MoveToEndPos	"typeMoveToEndPos"		☒	☒	☒	☒	Control of move
MoveToEndPosCommand	"typeMoveToEndPosCommand"		☒	☒	☒	☒	Command interfac
MoveToEndPosStatus	"typeMoveToEndPosStatus"		☒	☒	☒	☒	Status interface to
AlternateMotion	"typeAlternateMotion"		☒	☒	☒	☒	Control of alternat
AlternateMotionCommand	"typeAlternateMotionCommand"		☒	☒	☒	☒	Command interfac
AlternateMotionStatus	"typeAlternateMotionStatus"		☒	☒	☒	☒	Status interface to
Axis[1]	"typeAxisBasicsData"		☒	☒	☒	☒	Array of all axes in

Each array element of the global data block contains the following data of the assigned axis:

- Axis information
This data is entered in the data block using the "GetAxisInformation" function (FC):
 - Axis type, for example Speed Axis, Positioning Axis and Synchronous Axis.
 - Axis name as a string variable for use on the HMI user interface.
 - Additional axis description as a string variable for optional use on the HMI user interface.
- Diagnostic data
This is where the diagnostic data listed in Chapter [1.1.1](#) is saved.
- Axis control data
The basic functions of the axis listed in Chapter [1.1.2](#) can be controlled via this data area.
- Data for the additional function "Move to end position" (MoveToEndPos)
- Data for the additional function "Move between two selectable positions" (AlternateMotion)

Note

For high-performance processing of the axis data in the cyclic user program, it is recommended to read out the data of an Axis technology object only once at the start of OB1. Then the program is further processed with the saved data.

Central data management of the "LAxisBasics" block library supports you in implementing this programming recommendation.

However, for running the "LAxisBasics" functions, it is not mandatory to use central data management in a self-created user program.

Note

Through an extension of the PLC data type that represents the array element of the global data type, additional axis-related data for the user program can also be stored in the global data block.

1.1.4 Additional functions

The "LAxisBasics" block library additionally contains function blocks for the following additional functions, only available for axis type positioning axis and axis type synchronous axis:

- Move to end position ("MoveToEndPos")
Function block for positioning an axis with the additional option to move to a preceding stop and exert a pressure on this stop with selectable force.
- Move between two selectable positions ("AlternateMotion")
Function block for moving an axis between two selectable positions for test and optimization purposes. The motion can be permanent or based on a selectable number.

1.1.5 HMI user interface

An HMI user interface for a TP900 Comfort HMI panel is included for all the functionalities available in the "LAxisBasics" block library. This user interface allows you to diagnose and control the axes stored in the global data block.

To ensure that the user interface can also be used as a basis for your own machine operation, the implementation of the supplied HMI user interface considers the following:

- Modular HMI user interface design:
Each status display and each function of the "LAxisBasics" block library has already been implemented in the HMI user interface as a single screen page. This allows you to transfer user interface screen pages, individually or completely, to your own user projects.
- The HMI program can be easily connected to the user program:
A single HMI tag is used to connect the control and display elements of the HMI user interface to the user program. This tag is connected to the "HMI Interface" global data block in the user program.
- Data multiplexing in the controller
Each control and display element of the HMI user interface is limited to one axis from the global data block (DB) of central data management. The data for the HMI user interface is selected using the "AxisDataToHMI" function block additionally supplied in the "LAxisBasics" block library.
- Multilingual HMI user interface:
The screen pages of the HMI user interface are provided in English (US) and German and new languages can be added quickly and easily. The user interface already includes the required change language feature.

1.2 How the application example works

1.2.1 Monitoring and control of the axes

The desired functions (FCs) "LAxisBasics_GetAxisData_..." or function blocks (FBs) "LAxisBasics_AxisBasics_..." must be called in the cyclic user program for each axis in the project that you want to diagnose or control.

The "LAxisBasics" block library includes an independent function (FC) or function block (FB) for each of the following axis types:

- Speed Axes (SpeedAxis TO)
- Positioning Axes (PositioningAxis TO)
- Synchronous Axes (SynchronousAxis TO)

Note

Use the function (FC) if you only want to read out the data of an axis. If you want to additionally control the axis, use the appropriate function block (FB) that already contains the 'read out data' function.

1.2.2 Using the additional functions

In the cyclic user program, the appropriate function block (FB) must be called for each axis for which you want to use the additional functions supplied in the "LAxisBasics" block library.

1.2.3 Central data management

The global data block (DB) from the "LAxisBasics" block library is a suggestion for central data management of all the axis data of a project in the user program.

However, the data provided by the block library's functions (FCs) and function blocks (FBs) can also be used separately. The block library comes with specific PLC data types for all the data; in the user program, they can be used to define tags.

Note

Using PLC data types for the specific axis data allows easy and clear data handling in the user program.

Note

If you want to use the HMI user interface supplied in the "LAxisBasics" block library, it is recommended to use central data management in the user program.

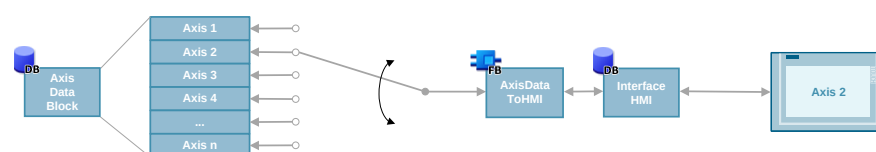
1.2.4 HMI user interface

Through multiplexing, the HMI user interface supplied in the "LAxisBasics" block library allows you to easily diagnose and control all the axes stored in central data management.

In the user program, multiplexing is performed using the supplied "AxisDataToHMI" function block that provides the following axis selection options:

- Advance axis selection for multiplexing using two buttons on the HMI user interface (forward/backward).
- Direct selection of an axis by entering and enabling the appropriate array index from central data management.
This function allows you to implement direct buttons for selected axes on the HMI user interface.

Figure 1-3 Multiplexing for the HMI user interface



1.3 Components used

This application example was created with the following hardware and software components:

Table 1-1 Hardware components

Component	No.	Article no.	Note
SIMATIC CPU 1511-1 PN	1	6ES7 511-1AK01-0AB0	Firmware V2.6 or higher
SIMATIC TP900 Comfort	1	6AV2 124-0JC01-0AX0	Firmware V15.1.0.0 or higher

Table 1-2 Software components

Component	No.	Article no.	Note
TIA Portal V15.1			Engineering framework
STEP 7 Professional V15.1	1	6ES7 822-1AA05-0YC5	SIMATIC STEP 7 Prof. V15.1 Powerpack Basic; Floating License; SW and documentation on DVD; License key on a USB stick.

Table 1-3 Technology objects

Component	Version
Technology object: speed, positioning or synchronous axis	V4.0 or higher
Motion control instructions	V4.0 or higher

This application example consists of the following components:

Table 1-4 Components of the application example

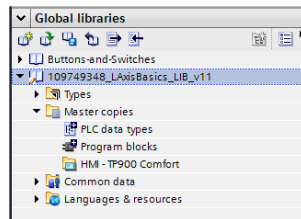
Component	File name	Note
Documentation	109749348_AxisBasics_DOC_v13_en.pdf	This document.
Block library	109749348_AxisBasics_LIB_v13.zip	The block library contains all the blocks that are needed to integrate the functionality into a self-created user program.
Sample program	109749348_AxisBasics_PROJ_v13.zip	In the sample program, the block library's blocks are integrated into a user program with three axes and connected to the HMI user interface.

2 Engineering

2.1 Integration into the user project

The "LAxisBasics" block library includes all the master copies you need to integrate the functionalities described in this application example into a self-created user program.

Figure 2-1 "LAxisBasics" block library



2.1.1 Program blocks

To integrate the program blocks, use drag and drop to move the PLC data types master copy from the library to the PLC data types of your project and the Program blocks master copy from the library to the program blocks of your project.

2.1.2 HMI user interface

To integrate the HMI user interface, use drag and drop to move the HMI – TP900 Comfort master copy from the library directly to your project. This creates a TP900 Comfort HMI panel in your project.

2.1.3 Using some of the included program blocks

Integration of the program blocks always integrates all blocks and PLC data types into the user program.

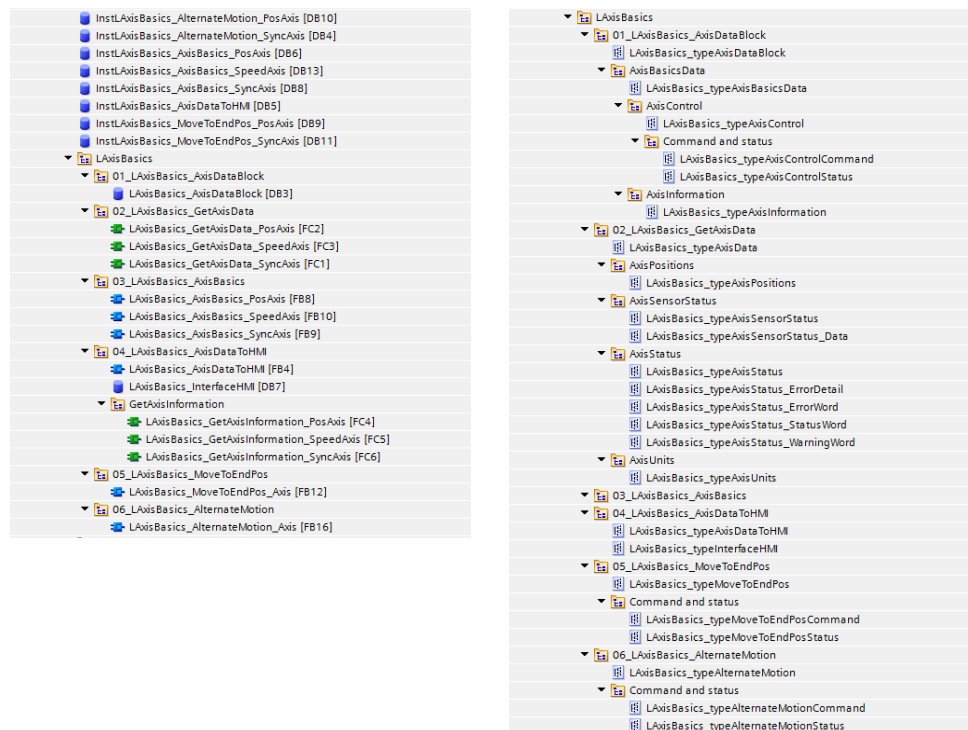
The program blocks and PLC data types are structured into individual subfolders. If you do not want to use all program blocks in the user program, you can delete unneeded folders from the LAxisBasics root directory.

The following functions are linked to the folders from the program blocks and PLC data types directory:

- **LAxisBasics_AxisDataBlock**
This folder contains the global data block (DB) / the associated PLC data types for central data management.
The objects contained in this folder can be optionally used in a self-created user program.
- **LAxisBasics_GetAxisData**
This folder contains the functions (FCs) for determining the diagnostic information of the axes.
This folder must not be removed.
- **LAxisBasics_AxisBasics**
This folder contains the function blocks (FBs) for diagnostics and control of the axes.
This folder can be removed if you do not want to use the blocks from the block library to control the axes.

- **LAxisBasics_AxisDataToHMI**
This folder contains all the function blocks (FBs), functions (FCs) and the interface data block (DB) for connecting the blocks to the supplied HMI user interface.
If you do not want to use the supplied HMI user interface, this folder can be removed.
- **LAxisBasics_MoveToEndPos**
This folder contains the function block (FB) for the additional function "Move to end position".
If you do not want to use this function, the folder can be removed.
- **LAxisBasics_AlternateMotion**
This folder contains the function block (FB) for the additional function "Alternating motion between two positions".
If you do not want to use this function, the folder can be removed.

Figure 2-2 Program blocks and PLC data types

**Note**

The available PLC data types can, of course, also be used in self-created user programs.

2.1.4 Using single pages or objects of the HMI user interface

If you only want to use single pages or elements of the included HMI user interface pages in a self-created HMI user interface, you can simply copy them from the supplied HMI device.

For the data connection of the pages or elements, you can either use the connection to the interface data block (DB) of the supplied HMI user interface or connect single objects to the appropriate data sources.

Figure 2-3 Data connection of the HMI to the interface data block (DB)

InterfaceHMI									
Name	Data type	Connection	PLC name	PLC tag	Address	Access mode	Acquisition cycle	Logged	Source comment
InterfaceAxisDataToHMI	LAxisBasics_typeAxisDataToHMI	HMI_Connection_1	PLC_1	LAxisBasics_InterfaceHMIAxisDataToHMI		<symbolic access>	100 ms		Axis data transferred to ...
<Add new>									

2.2 Description of the blocks and data types

2.2.1 "LAxisBasics_AxisDataBlock" DB

The global data block provides the basis for the "LAxisBasics" block library's central data management.

Table 2-1: Parameters of the "LAxisBasics_AxisDataBlock" DB

Parameter	Data type	Start value	Description
axis	Array[0..n] of "LAxisBasics_typeAxisBasicsData"	-	Memory area for each axis of the user program. The contained array must always start with index 0. The end index n can be customized to the existing number of axes.

The individual memory areas for each axis, i.e., each array element included in the data block are composed as described below.

[LAxisBasics_typeAxisBasicsData]

Table 2-2: Structure of the memory area for each axis

Parameter	Data type	Start value	Description
axisInformation	"LAxisBasics_typeAxisInformation"	-	Additional axis information for display on the HMI user interface.
axisData	"LAxisBasics_typeAxisData"	-	Current data and diagnostic information of the axis.
axisControl	"LAxisBasics_typeAxisControl"	-	Data for controlling the axis using the basic functions.
moveToEndPos	"LAxisBasics_typeMoveToEndPos"	-	Data for controlling the additional function "Move to end position".
alternateMotion	"LAxisBasics_typeAlternateMotion"	-	Data for controlling the additional function "Alternating motion between two positions".

[LAxisBasics_typeAxisInformation]

Table 2-3: Structure of the axis information

Parameter	Data type	Start value	Description
typeOfAxis	INT	0	Axis type: 0: No axis 1: Speed Axis 2: Positioning Axis

Parameter	Data type	Start value	Description
			3: Synchronous Axis
axisName	STRING	""	String of the axis name for display on the HMI user interface.
axisDescription	STRING	""	Additional information about the axis for optional display on the HMI user interface.

[LAxisBasics_typeAxisData]

Table 2-4: Structure of the axis data and diagnostic information

Parameter	Data type	Start value	Description
axisPositions	"LAxisBasics_typeAxisPositions"	-	Current positions and dynamic response data of the axis.
axisStatus	"LAxisBasics_typeAxisStatus"	-	Current status and diagnostic information of the axis.
axisSensorStatus	"LAxisBasics_typeAxisSensorStatus"	-	Current status of the connected axis encoders.
axisUnits	"LAxisBasics_typeAxisUnits"	-	Current settings of parameter units.

[LAxisBasics_typeAxisPosition]

Table 2-5: Structure of the axis positions

Parameter	Data type	Start value	Description
position	LREAL	0.0	Setpoint position
velocity	LREAL	0.0	Setpoint velocity
acceleration	LREAL	0.0	Setpoint acceleration
actualPosition	LREAL	0.0	Actual position
actualVelocity	LREAL	0.0	Actual velocity
actualAcceleration	LREAL	0.0	Actual acceleration
actualSpeed	LREAL	0.0	Current motor speed
operativeSensor	UDINT	0	Active sensor

[LAxisBasics_typeAxisStatus]

Table 2-6 Structure of the axis status

Parameter	Data type	Start value	Description
statusWord	"LAxisBasics_typeAxisStatus_StatusWord"	-	Status word (DWORD) of the axis represented bit by bit with plain text name.
warningWord	"LAxisBasics_typeAxisStatus_WarningWord"	-	Warning word (DWORD) of the axis represented bit by bit with plain text name.
errorWord	"LAxisBasics_typeAxisStatus_ErrorWord"	-	Error word (DWORD) of the axis represented bit by bit with plain text name.

Parameter	Data type	Start value	Description
errorDetail	"LAxisBasics_typeAxisStatus_ErrorDetail"	-	Additional information about the error cause on the axis: - Error number - Error response

Note

For the exact meaning of the individual bits of the status words, refer to the SIMATIC S7-1500 Motion Control documentation.

[LAxisBasics_typeAxisSensorStatus]

Table 2-7: Structure of the encoder status data

Parameter	Data type	Start value	Description
sensor	Array[1..4] of "LAxisBasics_typeAxisSensorStatus_Data"	-	Current status of the connected axis encoders.

[LAxisBasics_typeAxisSensorStatus_Data]

Table 2-8: Structure of the encoder status data

Parameter	Data type	Start value	Description
state	DINT	0	Status of actual encoder value 0: Not valid 1: Waiting for valid status 2: Valid
communication OK	BOOL	False	Cyclic bus communication between controller and encoder.
error	BOOL	False	Encoder error.
absEncoder Offset	LREAL	0.0	Home point offset to the value of an absolute value encoder.
control	BOOL	False	Encoder is active.
position	LREAL	0.0	Encoder position
velocity	LREAL	0.0	Encoder velocity
adaptionState	DINT	0	Status of automatic data transfer of encoder parameters: 0: Data not transferred 1: Data transfer in progress 2: Data transfer complete 3: Data transfer not possible 4: Error during data transfer

[LAxisBasics_typeAxisUnits]

Table 2-9: Structure of axis unit information

Parameter	Data type	Start value	Description
lengthUnit	UDINT	0	Unit for position
velocityUnit	UDINT	0	Unit for velocity
timeUnit	UDINT	0	Unit for time
torqueUnit	UDINT	0	Unit for torque
forceUnit	UDINT	0	Unit for force

Table 2-10 Mapping of the unit values to the parameter values

Parameter value	length Unit	velocityUnit			time Unit	torque Unit	force Unit
		Velocity	Accel.	Jerk			
1004	rad						
1005	°						
1010	m						
1011	km						
1013	mm						
1014	µm						
1015	nm						
1018	ft						
1019	in						
1021	mi						
1054					s		
1056					ms		
1061		m/s	m/s ²	mm/s ³			
1062		mm/s	mm/s ²	mm/s ³			
1063		m/h	m/s ²	mm/s ³			
1064		km/h	m/s ²	mm/s ³			
1066		in/s	in/s ²	in/s ³			
1067		ft/s	ft/s ²	ft/s ³			
1069		in/min	in/s ²	in/s ³			
1070		ft/min	ft/s ²	ft/s ³			
1075		mi/h	ft/s ²	ft/s ³			
1082		1/s	1/s ²	1/s ³			
1083		1/min	1/s ²	1/s ³			
1086		rad/s	rad/s ²	rad/s ³			
1093							ozf
1094							lbf
1120							N
1122							kN
1126						Nm	
1128						kNm	
1521		°/s	°/s ²	°/s ³			

Parameter value	length Unit	velocityUnit			time Unit	torque Unit	force Unit
		Velocity	Accel.	Jerk			
1522		°/min	°/s ²	°/s ³			
1523		rad/min	rad/s ²	rad/s ³			
1524		mm/min	mm/s ²	mm/s ³			
1525		m/min	m/s ²	mm/s ³			
1526		mm/h	mm/s ²	mm/s ³			
1527		km/min	m/s ²	mm/s ³			
1528		1/h	1/s ²	1/s ³			
1529						lbf in	
1530						lbf ft	
1531						ozf in	
1532						ozf ft	
1533						pdl in	
1534						pdl ft	
1535							pdl

NOTE The units for acceleration and jerk, shown at the HMI user interface, are derived from the velocity units of the technology object (TO).

[LAxisBasics_typeAxisControl]

Table 2-11: Structure of the control data of the axis ("typeAxisControl" data type)

Parameter	Data type	Start value	Description
axisControl Command	"LAxisBasics_typeAxisControl Command"	-	Input data for the "AxisBasics" function block for axis control.
axisControl Status	"LAxisBasics_typeAxisControl Status"	-	Output data of the "AxisBasics" function block as a status during axis control.

Note For the structure of the individual data areas, refer to the interface description for the "AxisBasics" function block.

moveToEndPos / alternateMotion

Table 2-12: Structure of the control data of the axis

Parameter	Data type	Start value	Description
moveToEndPos Command	"LAxisBasics_typeMoveToEndPos Command"	-	Input data for the "MoveToEndPos" function block.
moveToEndPos Status	"LAxisBasics_typeMoveToEndPos Status"	-	Output data of the "MoveToEndPos" function block.

Table 2-13: Structure of the control data of the axis

Parameter	Data type	Start value	Description
alternateMotion Command	"LAxisBasics_typeAlternateMotion Command"	-	Input data for the "AlternatMotion" function block.
alternateMotion Status	"LAxisBasics_typeAlternateMotion Status"	-	Output data of the "AlternateMotion" function block.

Note

For the structure of the individual data areas, refer to the interface description for the "MoveToEndPos" / "AlternateMotion" function blocks.

2.2.2 "LAxisBasics_GetAxisData" FC

The function collects the status and diagnostic data of an axis.

Note

An independent function is available for each axis type, SpeedAxis, PosAxis and SyncAxis. Use the appropriate function for the axis type you are using.

Data that is not provided by the axis type used is assigned start values by the block.

Figure 2-4 "LAxisBasics_GetAxisData" FC

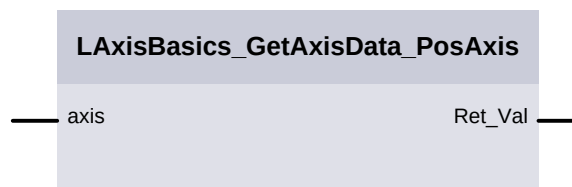


Table 2-14: Structure of the block interface

Parameter	Data type	Start value	Description
Input parameter			
axis	TO_SpeedAxis TO_PositioningAxis TO_SynchronousAxis	-	Connection of the technology object
Output parameter			
LAxisBasics_GetAxisData (Ret_Val)	"LAxisBasics_typeAxisData"	-	Return of the current status and diagnostic information of the axis.

Note

For the exact status and diagnostic information, see Chapter [2.2.1](#), section "[LAxisBasics_typeAxisData]".

2.2.3 "LAxisBasics_AxisBasics" FB

This function block allows you to control the following basic functionalities of an axis:

- Enable/disable axis (MC_Power)
- Acknowledge axis error/reinitialize axis (MC_Reset)
- Home axis (MC_Home)
This functionality is not available at the function block of axis type SpeedAxis.
- Manually move axis (MC_MoveJog)

In addition, the block collects the status and diagnostic data described in Chapter [2.2.2](#).

Note

An independent function block is available for each axis type, SpeedAxis, PosAxis and SyncAxis. Use the appropriate function block for the axis type you are using.

Figure 2-5 "LAxisBasics_AxisBasics" FB



Table 2-15: Structure of the block interface

Parameter	Data type	Start value	Description
Input parameters			
axis	TO_SpeedAxis TO_PositioningAxis TO_SynchronousAxis	-	Connection of the technology object

Parameter	Data type	Start value	Description
powerEnable	BOOL	False	Enable of the axis: - False: Axis disabled - True: Axis enabled
powerStartMode	DINT	1	Enable mode of the axis: 0: Controlled mode 1: Position-controlled This parameter is not available at the function block of axis type SpeedAxis.
powerStopMode	INT	0	Stop mode of the axis: 0: Stop with emergency stop 1: Immediate stop 2: Stop with max. dynamic response
resetExecute	BOOL	False	A rising edge resets/restarts the axis.
resetRestart	BOOL	False	Selection of reset or restart: 0: Reset technology alarms. 1: Reinitialize axis and reset alarms.
homeExecute	BOOL	False	A rising edge homes the axis. This parameter is not available at the function block of axis type SpeedAxis.
homePosition	LREAL	0.0	Home position Home position processing depends on the selected mode. This parameter is not available at the function block of axis type SpeedAxis.
homeMode	INT	0	Mode selection for homing: 0: Direct homing (absolute) 1: Direct homing (relative) 2: Passive homing 3: Active homing 4: Reserved 5: Active homing (<TO>.Homing.HomePosition) 6: Absolute encoder adjustment (relative) 7: Absolute encoder adjustment (absolute) 8: Passive homing 9: Abort passive homing 10: Passive homing (<TO>.Homing.HomePosition) This parameter is not available at the function block of axis type SpeedAxis.
jogForward	BOOL	False	Move the axis in the positive direction while the parameter is true.
jogBackward	BOOL	False	Move the axis in the negative direction while the parameter is true.
jogVelocity	LREAL	100.0	Velocity/speed for the motion.
jogAcceleration	LREAL	-1.0	Acceleration

Parameter	Data type	Start value	Description
jogDeceleration	LREAL	-1.0	Deceleration
jogJerk	LREAL	-1.0	Jerk
jogPosition Controlled	BOOL	True	Selection of the control type for the motion: 0: Open-loop operation 1: Closed-loop operation This parameter is not available at the function block of axis type SpeedAxis.
Output parameters			
powerStatus	BOOL		The axis is enabled and can be moved.
powerBusy	BOOL		The "MC_Power" function is active.
resetDone	BOOL		Axis successfully reset/restarted.
resetBusy	BOOL		The "MC_Reset" function is active.
homeDone	BOOL		Axis successfully homed. This parameter is not available at the function block of axis type SpeedAxis.
homeBusy	BOOL		The "MC_Home" function is active. This parameter is not available at the function block of axis type SpeedAxis.
jogInVelocity	BOOL		The axis is moved at the specified velocity.
jogBusy	BOOL		The "MC_MoveJog" function is active.
command Aborted	BOOL		One of the motion control instructions active in the function block is overridden by another motion control instruction. The relevant instruction is specified in greater detail via the "StatusID" output.
error	BOOL		An error occurred while executing a motion control instruction included in the function block. The relevant instruction is specified in greater detail via the "StatusID" output.
errorID	WORD		Error code of the motion control instruction.
statusID	INT		Specifies the motion control instruction where an error event occurred: 0: No error 1: MC_Power 2: MC_Reset 3: MC_Home 4: MC_MoveJog
axisData	"LAxisBasics_typeAxisData"	-	Return of the current status and diagnostic information of the axis.

Note

For the exact status and diagnostic information, see Chapter [2.2.1](#), section "[LAxisBasics_typeAxisData]".

2.2.4 "LAxisBasics_AxisDataToHMI" FB / "LAxisBasics_InterfaceHMI" DB / "LAxisBasics_GetAxisInformation" FC

Data exchange between central data management and the HMI user interface takes place via the function and function block by selecting the desired axis.

The "LAxisBasics_InterfaceHMI" global data block provides the selected data for the HMI user interface in the user program. This data block is then used to connect the HMI user interface.

For information about how the individual blocks work, see Chapter [1.2.4](#).

"LAxisBasics_AxisDataToHMI" FB

This function block is used to retrieve the axis data of the axis currently selected on the HMI user interface from central data management and provide it for display.

Figure 2-6 "LAxisBasics_AxisDataToHMI" FB

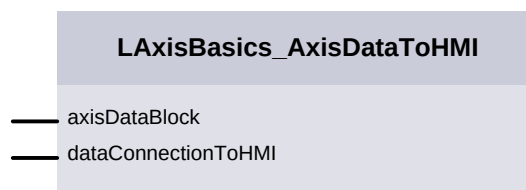


Table 2-16: Structure of the block interface

Parameter	Data type	Start value	Description
In/out parameters			
axisDataBlock	Array[*] of "LAxisBasics_typeAxisBasicsData"	-	Transfers the entire array from central data management. Due to the array[*] operator, the actual length of the array, i.e., the number of axes stored in it, is not determined in this function block until runtime. For the structure of the parameter, see Chapter 2.2.1 .
dataConnectionToHMI	"LAxisBasics_typeAxisDataToHMI"	-	Returns the data of the selected axis to the "LAxisBasics_InterfaceHMI" DB for transfer to the HMI user interface. The desired axis is selected via the index data within this parameter. The following table shows the structure of this parameter.

"LAxisBasics_InterfaceHMI" DB

This data block establishes the data connection between the axis data and the HMI user interface.

The following table shows the structure of the "LAxisBasics_InterfaceHMI" data block for transferring the data to/from the HMI user interface.

Table 2-17: Structure of the "LAxisBasics_InterfaceHMI" data block

Parameter	Data type	Start value	Description
axisBasicsData	"LAxisBasics_ typeAxisBasicsData"	-	Current positions and dynamic response data of the axis. For the structure of the parameter, see Chapter 2.2.1 .
actualAxisIndex	DINT	-	Array index of the currently selected axis. The data of this axis is currently copied to the "LAxisBasics_InterfaceHMI" DB and displayed on the HMI user interface.
selectPrevious Axis	BOOL	-	A rising edge decreases the current array index by one, resulting in the previous axis being selected.
selectNextAxis	BOOL	-	A rising edge increases the current array index by one, resulting in the next axis being selected.
directIndex Selection	DINT	-	Direct selection of an array index for axis selection. The selected index must be within the allowed range for the array indices.
executeDirect IndexSelection	BOOL	-	A rising edge selects the axis with the directly selected index. The command is only executed if the directly specified index is within the allowed range for the array indices.

Note

The direct index allows you to implement direct buttons for single axes on the HMI user interface: by setting the "directIndexSelection" parameter to the appropriate index value when pressing the button and setting the "executeDirectIndexSelection" parameter to true while the button is pressed.

"LAxisBasics_GetAxisInformation" FC

This function allows you to enter additional information one time in central data management when the cyclic user program starts. It is not designed for changes to this data during user program runtime.

Figure 2-7 "LAxisBasics_GetAxisInformation" FC

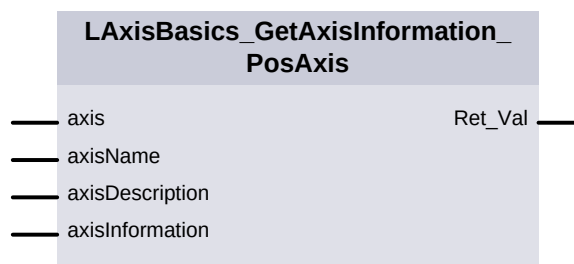


Table 2-18: Structure of the block interface

Parameter	Data type	Start value	Description
Input parameters			
axis	TO_SpeedAxis TO_PositioningAxis TO_SynchronousAxis	-	Connection of the technology object
axisName	STRING	""	Axis name to be displayed on the HMI user interface. If no axis name is specified here, the technology object identifier from the user program is entered as the name.
axisDescription	STRING	""	Additional axis description for use on the HMI user interface. This information is currently not displayed on the supplied HMI user interface.
Output parameters			
LAxisBasics_GetAxisInformation (Ret_Val)	BOOL	-	If the appropriate information was written to central data management, the value true is returned here.
In/out parameters			
axisInformation	"LAxisBasics_typeAxisInformation"	-	Connection of the memory area of the appropriate axis in central data management.

Note

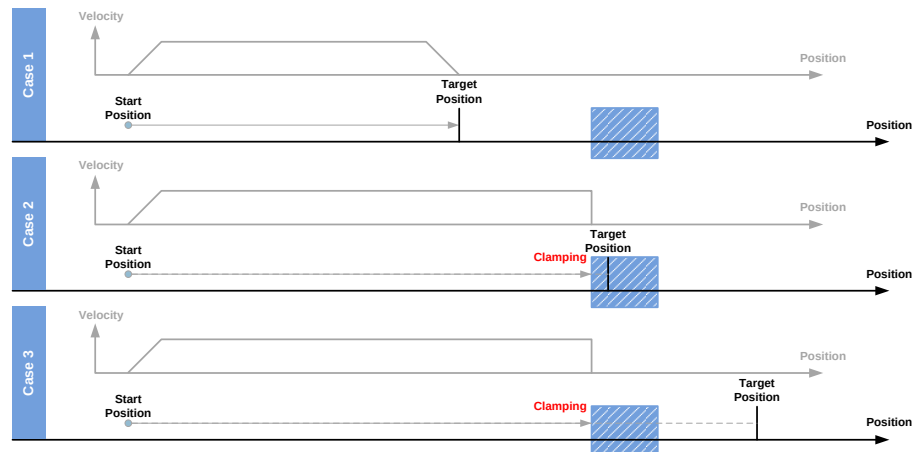
This function is only executed once when starting the user program. However, the block can also be called permanently in the cyclic user program.

2.2.5 "LAxisBasics_MoveToEndPos_Axis" FB

This function block allows you to perform a positioning job on an axis to which the following applies:

- If the axis can reach the specified target position, it moves to this position, which completes the positioning job.
- If the specified target position is in or behind an obstacle, the axis approaches the obstacle and exerts a pressure on it with the specified torque value until the axis motion enable is removed on the block.

Figure 2-8 Configuration examples for the "LAxisBasics_MoveToEndPos_Axis" FB



The torque value the axis uses to exert a pressure on the obstacle is specified in the unit configured on the technology object.

Figure 2-9 "LAxisBasics_MoveToEndPos_Axis" FB

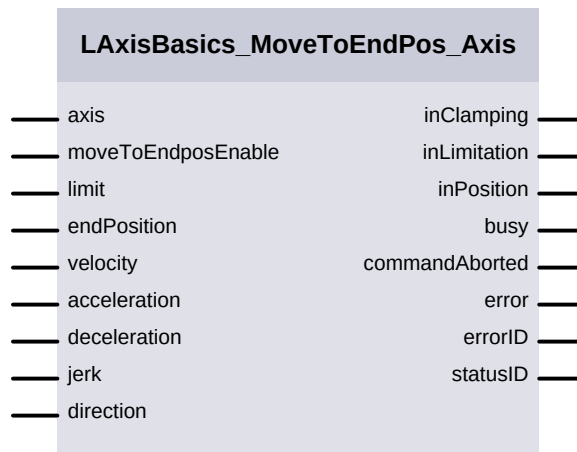


Table 2-19: Structure of the block interface

Parameter	Data type	Start value	Description
Input parameters			
axis	TO_PositioningAxis	-	Connection of the technology object
moveToEndPos Enable	BOOL	False	Axis motion enable - False: Stop motion or cancel clamping. - True: Enable/start motion

Parameter	Data type	Start value	Description
limit	LREAL	0.0	Force/torque limit value (in the configured unit) If the drive and telegram do not support a force/torque limit, the specified value is irrelevant. ≥0: Use the specified value. <0: Use the value configured on the technology object.
endPosition	LREAL	0.0	Absolute target position for positioning.
velocity	LREAL	-1.0	Velocity
acceleration	LREAL	-1.0	Acceleration
deceleration	LREAL	-1.0	Deceleration
jerk	LREAL	-1.0	Jerk
direction	INT	1	Direction of movement of the axis Evaluated only when "modulo" active. 1: Positive direction 2: Negative direction 3: Shortest path
Output parameters			
inClamping	BOOL	False	The drive is kept at the fixed stop (clamping), the axis position is within the positioning tolerance.
inLimitation	BOOL	False	The drive operates at the force/torque limit.
inPosition	BOOL	False	Target position reached
busy	BOOL	False	The job is being processed
command Aborted	BOOL	False	The job was aborted by another job while processing.
error	BOOL	False	An error occurred while processing the job.
errorID	WORD	16#0	Block error code.
statusID	INT	0	Specifies the motion control instruction where an error event occurred: 0: No error 1: MC_TorqueLimiting 2: MC_MoveAbsolute 3: MC_Halt

Note Changes to the input parameters of the function block will not be applied until the axis motion is re-enabled via the "moveToEndPosEnable" block input.

Note The axis motion/clamping by the axis is aborted/stopped by setting the block's "moveToEndPosEnable" input to "False".

2.2.6 "LAxisBasics_AlternateMotion_Axis" FB

This function block allows you to execute an alternating motion on an axis for drive setting or other test purposes.

The conditions for this motion are as follows:

- The axis is positioned back and forth between a start position and a reversing position.
- The motion always starts at the start position.
- The motion can be executed with a defined number of movements or permanently.
- The end position of the axis can be defined by the number of movements or, when the motion is permanent, by the selection input on the block.

Figure 2-10 Axis motions of the "LAxisBasics_AlternateMotion_Axis" FB

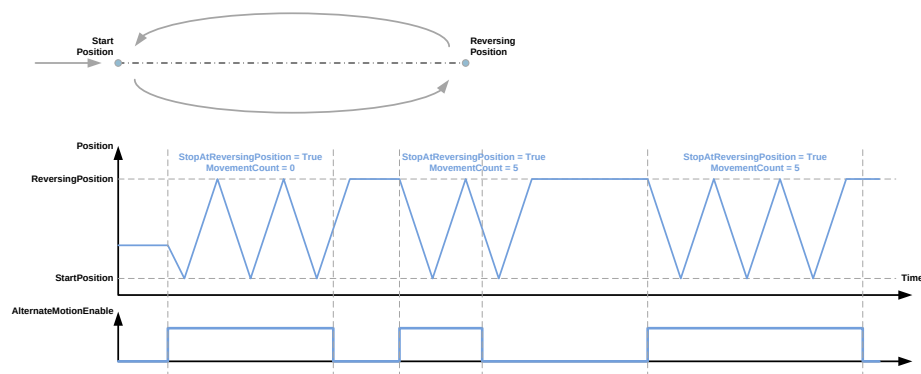


Figure 2-11 "LAxisBasics_AlternateMotion_Axis" FB



Table 2-20: Structure of the block interface

Parameter	Data type	Start value	Description
Input parameters			
axis	TO_PositioningAxis	-	Connection of the technology object
alternateMotion Enable	BOOL	False	Axis motion enable - False: Stop motion. - True: Start motion.
startPosition	LREAL	0.0	Start position of the alternating motion. When the motion starts, the axis first moves to the start position.
reversing Position	LREAL	0.0	Reversing position of the alternating motion.
stopAtReversing Position	BOOL	False	Defines the position where the motion stops when the enable is removed: - False: Motion stops at start position. - True: Motion stops at reversing position. This setting is only considered if no specified number of movements is selected.
movementCount	INT	0	Selection of a specified number of movements: =0: The alternating motion is executed while the axis motion enable is active. >0: The specified number of movements is executed. The motion's end position is defined by the number of movements: - Odd number: Motion ends at start position. - Even number: Motion ends at reversing position.
velocity	LREAL	-1.0	Velocity
acceleration	LREAL	-1.0	Acceleration
deceleration	LREAL	-1.0	Deceleration
jerk	LREAL	-1.0	Jerk
Output parameters			
done	BOOL	False	Function successfully executed.
busy	BOOL	False	The function is active.
command Aborted	BOOL	False	One of the motion control instructions active in the function block is overridden by another motion control instruction. The relevant instruction is specified in greater detail via the "StatusID" output.
error	BOOL	False	An error occurred while executing a motion control instruction included in the function block. The relevant instruction is specified in greater detail via the "StatusID" output.

Parameter	Data type	Start value	Description
errorID	WORD	16#0	Error code of the motion control instruction.
statusID	INT	0	Specifies the motion control instruction where an error event occurred: 0: No error 1: MC_MoveAbsolute (StartPos.) 2: MC_MoveAbsolute (Rev.Pos.)

Note

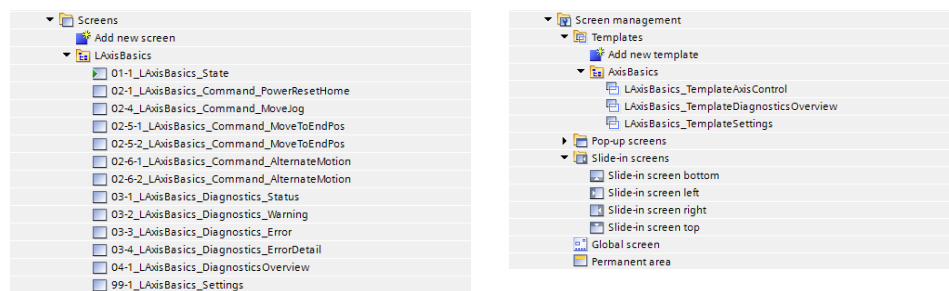
Changes to the input parameters of the function block will not be applied until the axis motion is re-enabled via the "alternateMotionEnable" block input.

2.3 Description of the HMI user interface

2.3.1 Structure of the HMI project

The HMI user interface for a TP900 Comfort HMI panel supplied in the "LAxisBasics" block library has the following structure:

Figure 2-12 HMI screens and HMI screen management



For each function included in the block library, the HMI project contains separate screen pages that can also be taken from the project individually.

Templates

For the individual pages, the HMI project's screen management provides three templates as a basis for the screen pages. They are assigned to the individual HMI user interface pages as follows:

- "LAxisBasics_TemplateAxisControl"
This template is the basis for all pages that display axis-related data.
- "LAxisBasics_TemplateDiagnosticsOverview"
This is the template for the cross-CPU "Diagnostics Overview" page.
- "LAxisBasics_TemplateSettings"
This is the template for the HMI-related settings page for the panel settings.

NOTE

To set up your own operating pages on HMI, you could use or copy the screen template "LAxisBasics_TemplateSettings".

Slide-in screens

From screen management, the "Slide-in screen right" is used for the HMI user interface. On the user interface, this slide-in screen allows you to toggle between the operating pages for the functions and the panel settings pages.

Figure 2-13 Using the right-hand slide-in screen from screen management

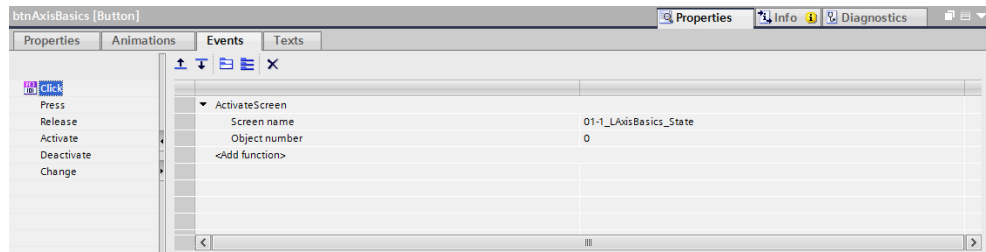


Note

If you want to use the supplied HMI user interface for your next machine operation steps, this slide-in screen allows you to switch to the other operating pages of the machine.

The call of the other operating pages can be easily entered and linked in the button areas that have already been prepared.

Figure 2-14 Slide-in screen event for the call of the HMI operating pages

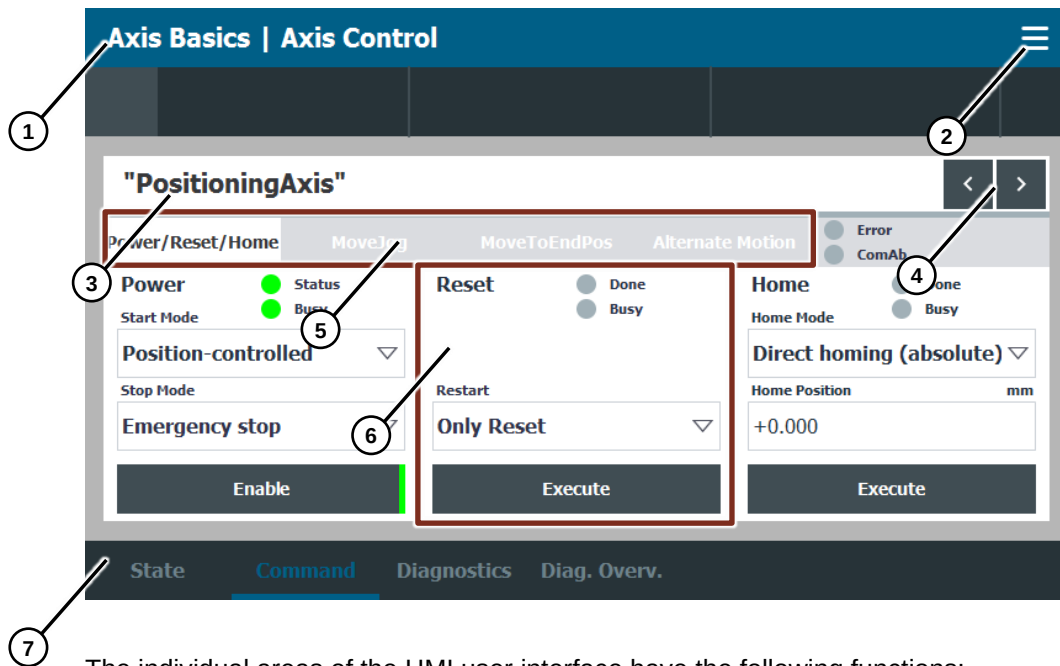


2.3.2 Structure of the HMI operating screens

The operating screens of the HMI user interface supplied in the "LAxisBasic" block library always have the structure shown in the following figure.

Some areas of the user interface are assigned to special functions/operator actions. Other areas can be used for user-defined HMI operating screens.

Figure 2-15 Structure of the HMI project's operating screens



The individual areas of the HMI user interface have the following functions:

Table 2-21 Functions of the HMI user interface areas

No.	Function / description
1.	Header of the functional area of the HMI user interface's currently selected operating pages.
2.	Opens the slide-in screen for selecting the function areas of the HMI user interface.
3.	Displays the axis name stored in central data management (DB).
4.	Selection buttons of the axis to be displayed/controlled.
5.	Subselection of the individual operating pages of the different axis functions.
6.	Operating area of the different axis functions.
7.	Allows you to select the individual function areas for control/diagnostics of the selected axis.

Note

All operating screens in the block library are provided in English and German. On the panel settings page, you can toggle between the two languages.

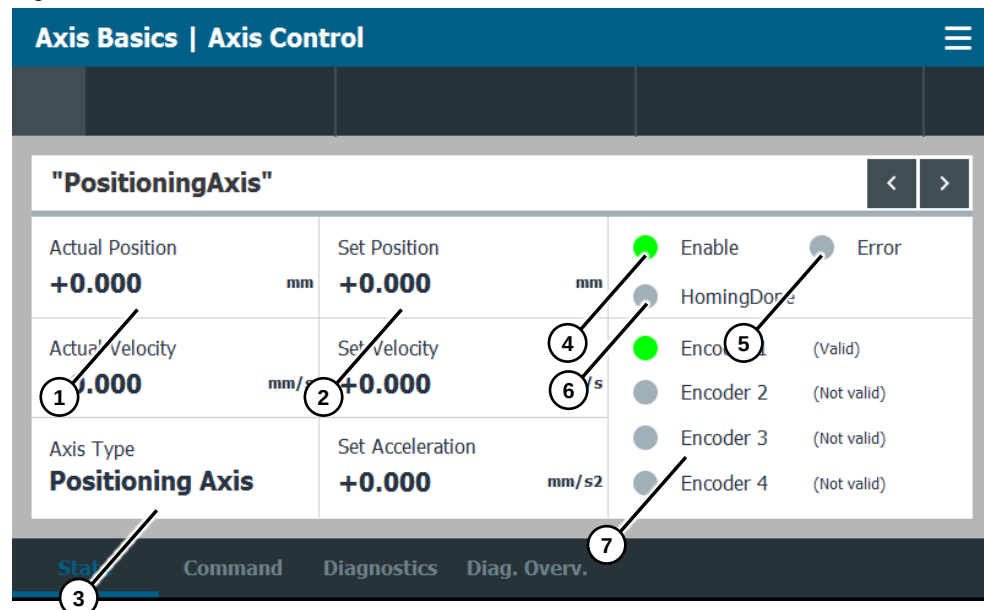
2.3.3 State screen

The State screen is an overview screen that displays the cyclically determined axis data, specifically the current axis positions and the dynamic response values of the axis.

The first column of the display area (1) displays the actual values of the axis. The second column (2) shows the setpoint values of the axis. The first column additionally displays the type of the currently selected axis (3) in plain text.

The third column displays additional information about the axis out of status word such as its current enable status (4), any error (5), homing state of axis (6) and the encoder that is currently active on the axis (7) with the actual state of the encoder.

Figure 2-16 HMI – State

**NOTE**

Not all displayed functions on this HMI screen are available with axis type SpeedAxis. Not available functions are displayed in light gray on the screen.

2.3.4 Operating screens of the functions

The operating screens of the individual axis control functions can include operating areas for one or three functions.

Parameter selection (1) allows you to parameterize the control function in plain text. Selecting the appropriate button (2) enables/executes the function.

The top area displays the current state of the respective function (3).

The subselection line of each control function operating page summarizes occurring errors or warnings for all on HMI screen shown control functions (4). In the event of an error, it also displays the ErrorID to specify the error cause and the relevant control function in plain text.

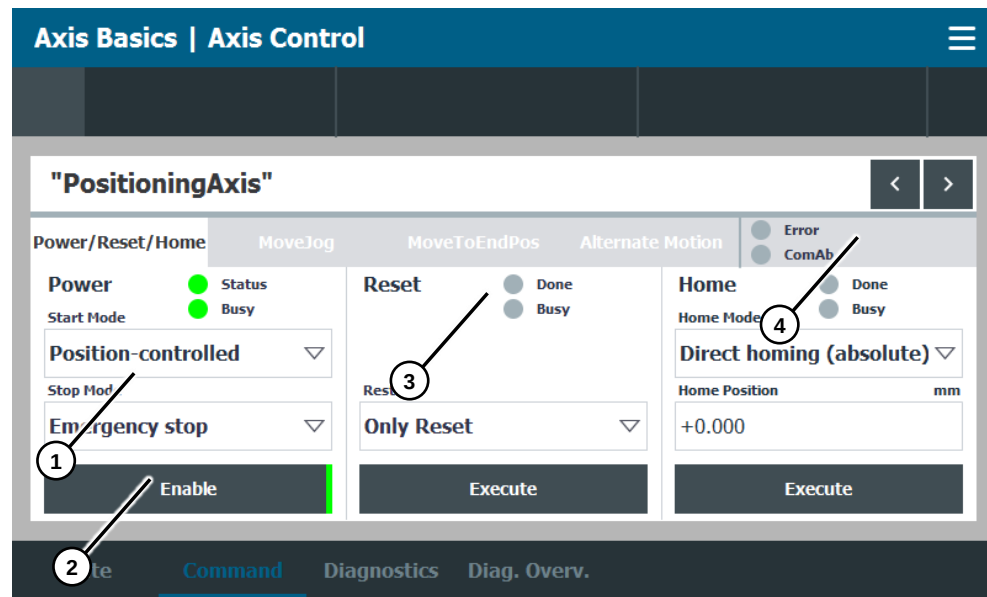
NOTE

Not all displayed functions on this HMI screen are available with axis type SpeedAxis. Not available functions are displayed in light gray on the screen.

Power, Reset, Home – enabling, resetting and homing the axis

This operating page allows you to enable/disable the axis, acknowledge error messages and home the axis.

Figure 2-17 HMI – Command – Power, Reset, Home

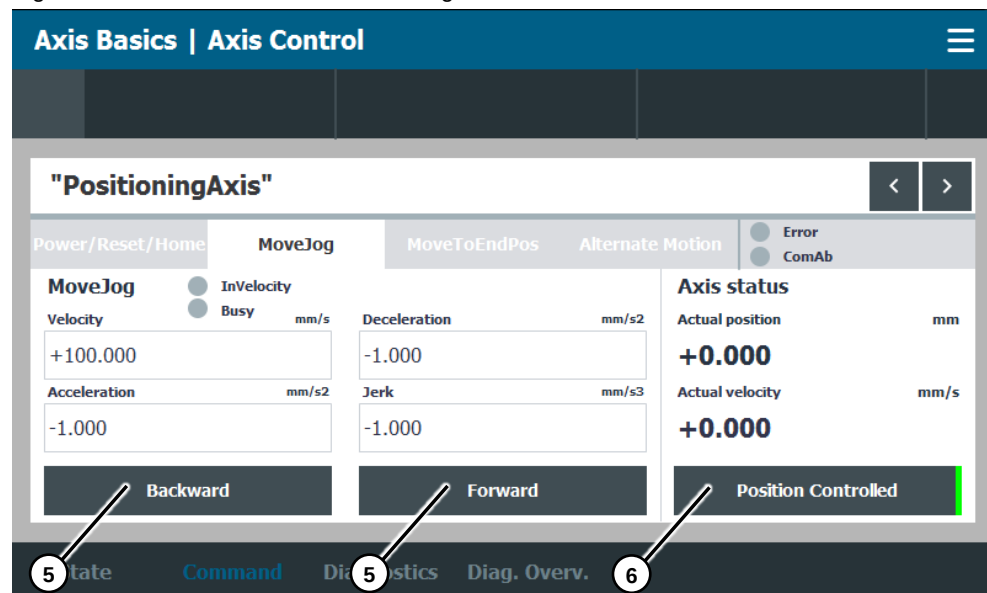


MoveJog – moving the axis in jog mode

This operating page allows you to move the axis manually in jog mode. Pressing the appropriate button (5) moves the axis in the selected direction while the button is pressed.

With button (6) you could switch off and on position control operation in jog mode.

Figure 2-18 HMI – Command – MoveJog

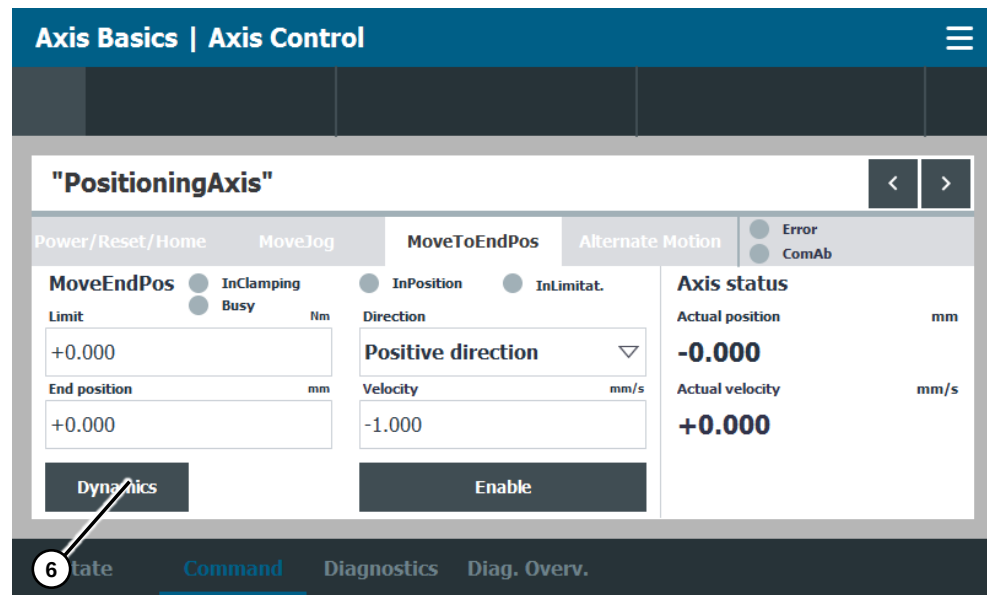


MoveToEndPos – moving to end position

These two operating pages allow you to move the axis to an end position. The axis exerts a defined pressure on the end position, provided that the end position is reached before the specified target position.

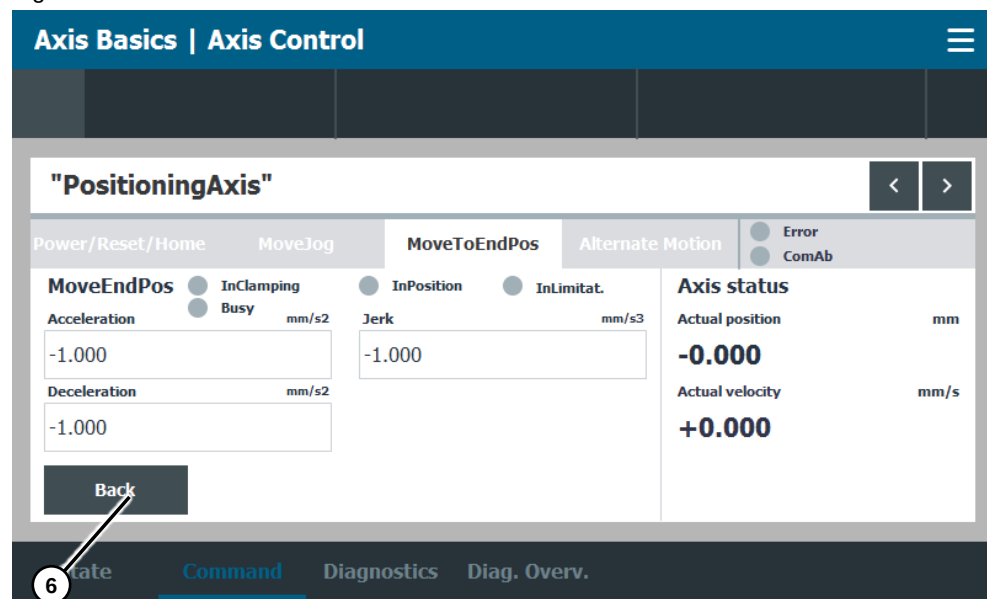
A button (6) allows you to toggle between the two pages.

Figure 2-19 HMI – Command – MoveToEndPos 1



After switching to the second operating page, you can specify the dynamic response values for 'move to target position'/'move to end position'.

Figure 2-20 HMI – Command – MoveToEndPos 2



AlternateMotion – alternate positioning between two positions

These two operating pages allow you to alternately position the axis between two selectable positions.

A button (6) allows you to toggle between the two pages.

Figure 2-21 HMI – Command – AlternateMotion 1

Axis Basics | Axis Control

"PositioningAxis"

Power/Reset/Home MoveJog MoveToEndPos **Alternate Motion** Error ComAb

Alt. Motion Done Busy

Start position mm
+0.000

Reversing position mm
+0.000

Stop mode
Stop at start position

Movement count
0

Axis status

Actual position mm
-0.000

Actual velocity mm/s
+0.000

Dynamics Enable

6 State **Command** Diagnostics Diag. Overv.

After switching to the second operating page, you can specify the dynamic response values for the axis motion.

Figure 2-22 HMI – Command – AlternateMotion 2

Axis Basics | Axis Control

"PositioningAxis"

Power/Reset/Home MoveJog MoveToEndPos **Alternate Motion** Error ComAb

Alt. Motion Done Busy Error

Velocity mm/s
-1.000

Acceleration mm/s²
-1.000

Deceleration mm/s²
-1.000

Jerk mm/s³
-1.000

Axis status

Actual position mm
-0.000

Actual velocity mm/s
+0.000

Back

6 State **Command** Diagnostics Diag. Overv.

2.3.5 Diagnostic screens

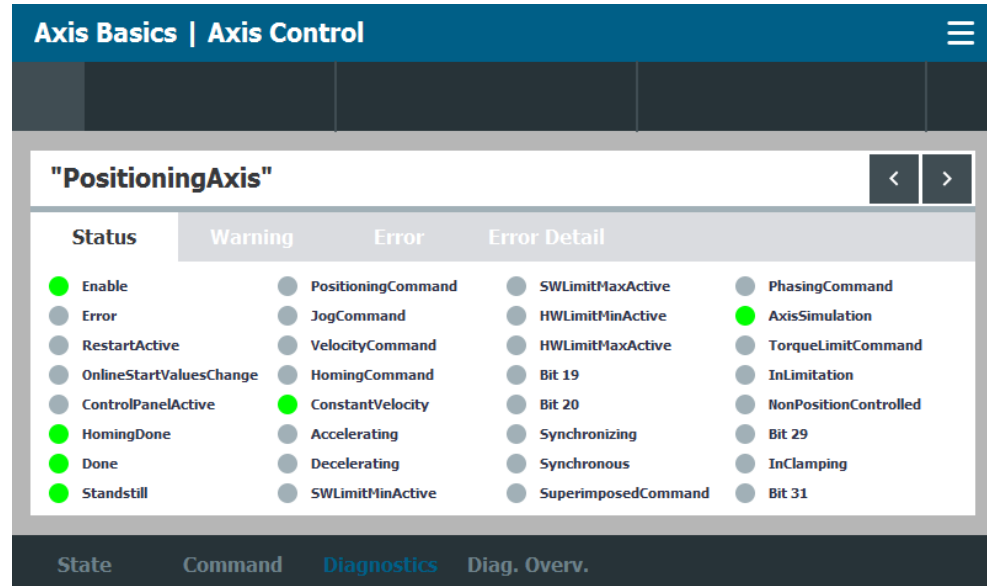
The screens for the diagnostic information of the selected axis display – bit by bit – the status of the individual StatusWord, WarningWord and ErrorWord bits.

The bits are in plain text, which allows you to retrieve the desired information quickly and easily from the display.

StatusWord

This screen displays, bit by bit, the status of the StatusWord from the technology object.

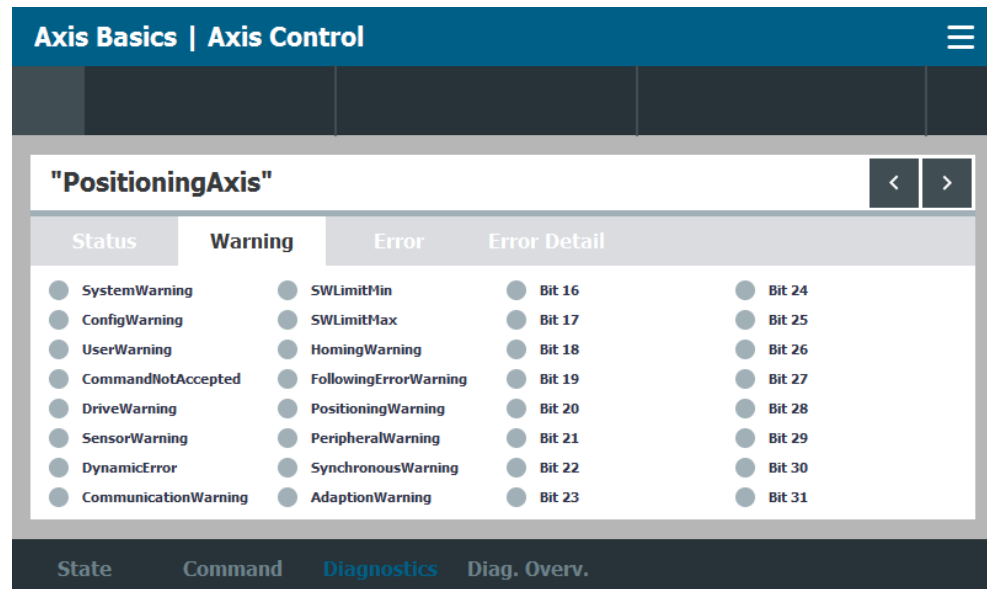
Figure 2-23 HMI – Diagnostics – Status



WarningWord

This screen displays, bit by bit, the status of the WarningWord from the technology object.

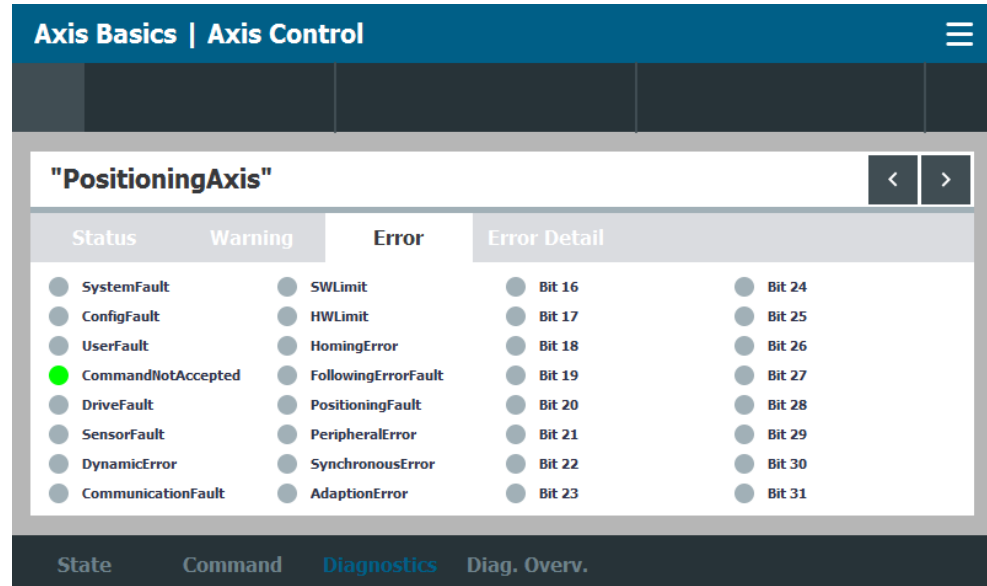
Figure 2-24 HMI – Diagnostics – Warning



ErrorWord and ErrorDetail

This screen displays, bit by bit, the status of the ErrorWord from the technology object.

Figure 2-25 HMI – Diagnostics – Error

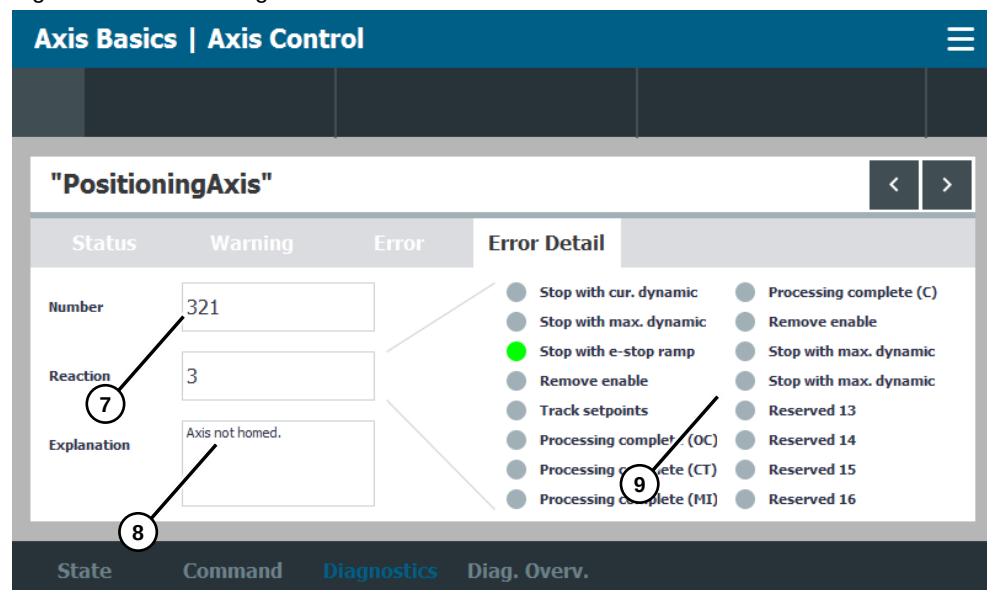


The ErrorDetail screen displays additional information about the error cause from the technology object.

It displays the additional technology alarm of the technology object as an error number (7) and at the same time in plain text (8).

Likewise, it shows the technology object's error response as a number and, in addition, its detailed responses (9).

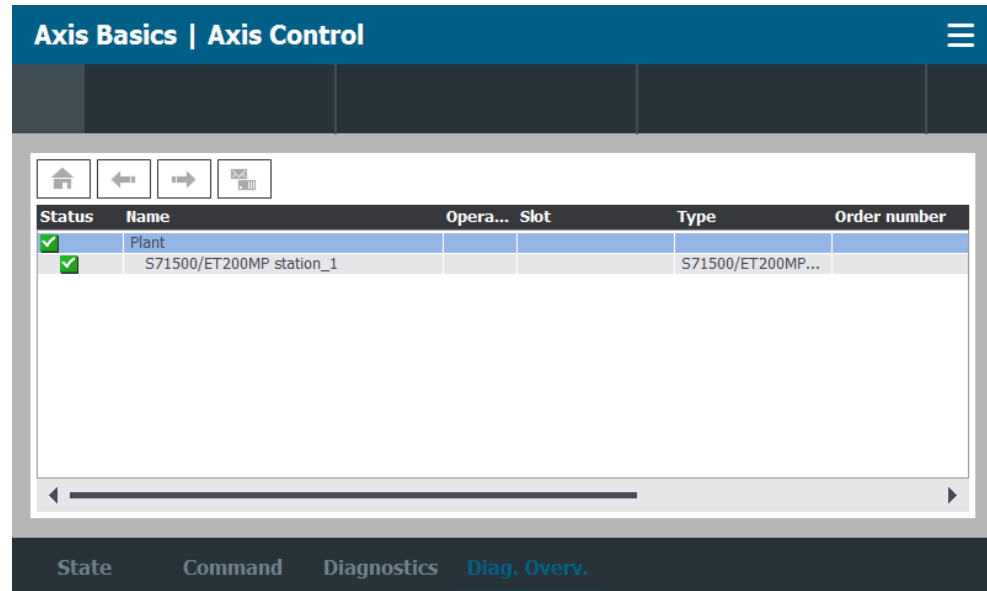
Figure 2-26 HMI – Diagnostics – ErrorDetail



DiagnosticsOverview – CPU-wide system diagnostics

This screen displays the "DiagnosticsOverview" HMI object for CPU-wide system diagnostics. It provides additional diagnostic information from the CPU's entire hardware configuration.

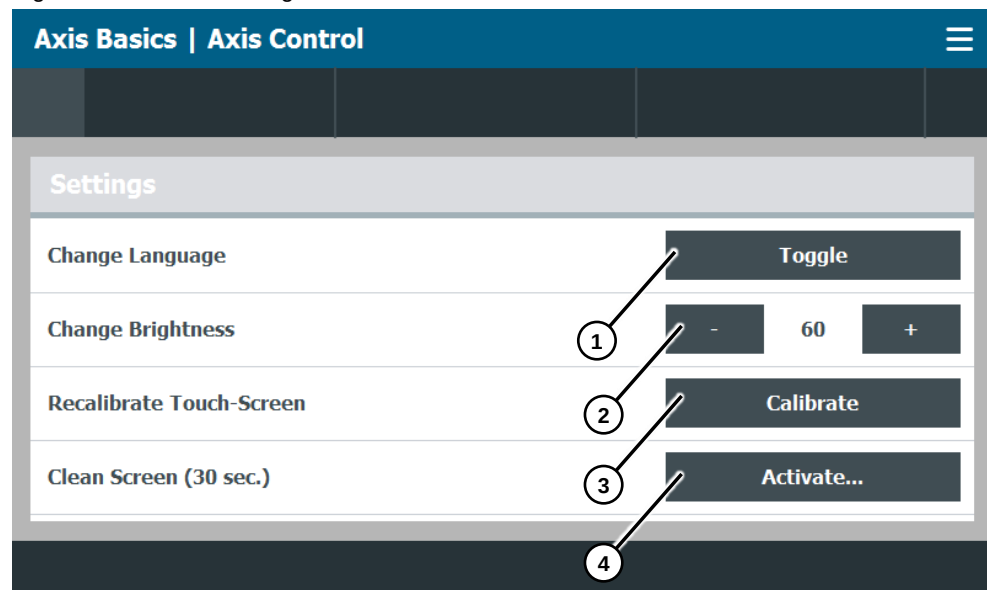
Figure 2-27 HMI – Diagnostics – DiagnosticsOverview



2.3.6 HMI settings screen

On the panel settings screen, you can toggle between English and German (1), change the panel's brightness (2), recalibrate the touchscreen (3) and activate the Clean screen (4).

Figure 2-28 HMI – Settings

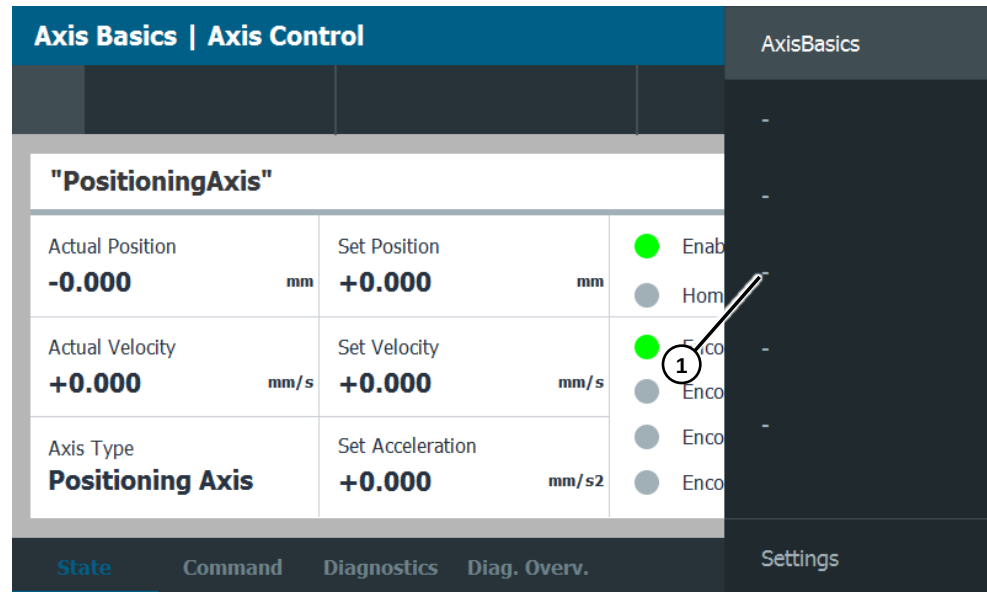


2.3.7 Slide-in screen for operating page selection

The button in the HMI operating screen's header (see [Figure 2-15](#)) allows you to activate the right-hand slide-in screen where you can select the HMI user interface pages.

If you want to use the HMI user interface for your next machine operation steps, these operating pages can also be opened using the prepared slide-in screen placeholder (1).

Figure 2-29 HMI – SlideInScreen – Right



3 Appendix

3.1 Service and Support

Industry Online Support

Do you have any questions or do you need support?

With Industry Online Support, our complete service and support know-how and services are available to you 24/7.

Industry Online Support is the place to go to for information about our products, solutions and services.

Product Information, Manuals, Downloads, FAQs and Application Examples – all the information can be accessed with just a few clicks:

<https://support.industry.siemens.com>

Technical Support

Siemens Industry's Technical Support offers you fast and competent support for any technical queries you may have, including numerous tailor-made offerings ranging from basic support to custom support contracts.

You can use the web form below to send queries to Technical Support:

www.siemens.com/industry/supportrequest

Service offer

Our service offer includes the following services:

- Product Training
- Plant Data Services
- Spare Part Services
- Repair Services
- Field & Maintenance Services
- Retrofit & Modernization Services
- Service Programs & Agreements

For detailed information about our service offer, please refer to the Service Catalog:

<https://support.industry.siemens.com/cs/sc>

Industry Online Support app

The "Siemens Industry Online Support" app provides you with optimum support while on the go. The app is available for iOS and Android:

<https://support.industry.siemens.com/cs/ww/en/sc/2067>

3.2 Links and literature

Table 3-1

No.	Topic
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Link to the entry page of the application example https://support.industry.siemens.com/cs/ww/en/view/109749348
\3\	SIMATIC Function Manual "SIMATIC S7-1500 S7-1500T Motion Control V4.0 in the TIA Portal V15" Edition: 12/2017 Order number: A5E37577655-AB https://support.industry.siemens.com/cs/ww/en/view/109749263
\4\	Topic page "SIMATIC Technology" https://support.industry.siemens.com/cs/ww/en/view/109751049
\5\	Application Example "SIMATIC/SIMOTION Intelligent Belt - Multi Belt Control" https://support.industry.siemens.com/cs/ww/en/view/48812744
\6\	Application Example "Migration Guide: S7-31xT to S7-1500(T)" https://support.industry.siemens.com/cs/ww/en/view/109743136
\7\	Application Example "SIMATIC S7-1200/S7-1500 and SIMOTION: Acyclic Data Exchange" https://support.industry.siemens.com/cs/ww/en/view/109479553
\8\	Application Example "SIMATIC/SIMOTION Library LCamHdl – Creation of cam disks at runtime" https://support.industry.siemens.com/cs/ww/en/view/105644659
\9\	Application Example "Technology Objects of the SIMATIC S7-1500(T)" https://support.industry.siemens.com/cs/ww/en/view/109743134
\10\	Application Example "SIMATIC S7-1500T Flying Saw" https://support.industry.siemens.com/cs/ww/en/view/109744840
\11\	Application Example "S7-1500T: Synchronizing in Standstill "CamInStandStill"" https://support.industry.siemens.com/cs/ww/en/view/109745764
\12\	Application Example "Migration Guide: SIMATIC S7-300/S7-400 to SIMATIC S7-1500" https://support.industry.siemens.com/cs/ww/en/view/109478811
\13\	Application Example "Using the MC-PreServo and MC-PostServo organization blocks" https://support.industry.siemens.com/cs/ww/en/view/109741575
\14\	Application Example "Time-based IO with the SIMATIC S7-1500 and TM Timer" https://support.industry.siemens.com/cs/ww/en/view/109738186
\15\	Application Example "Measuring input with Time-based IO" https://support.industry.siemens.com/cs/ww/en/view/109480157

3.3 Change documentation

Table 3-2

Version	Date	Modifications
V1.1	04/2018	First version
V1.1.1	06/2019	Added information about TO version (Table 1-3)
V1.2	12/2019	Added check if encoder and actor of axis is ready before enabling or referencing
V1.3	05/2020	Fixed error in IsAxisReady function blocks