

Technology module



Virtual Master Base (V2) -----

Reference manual

EN



13588978

Contents

1	About this document	3
1.1	Document history	5
1.2	Conventions used	6
1.3	Definition of the notes used	7
2	Safety instructions	8
3	Overview of the technology module functions	10
3.1	Functional description	10
3.2	TM-functional overview	11
3.3	Important notes on how to operate the technology module	12
4	Application ranges	13
5	L_TF2P_VirtualMasterBase function block	14
5.1	inputs	15
5.2	outputs	16
5.3	Parameter	19
6	State machine	21
7	Signal flow diagram	23
8	Description of the basic functions	24
8.1	Manual jog (jogging)	25
8.2	Homing	27
8.3	QSP application	30
8.4	Halt	31
8.5	Absolute positioning	32
8.6	Relative positioning	34
8.7	Continuous travel with a defined velocity	37
8.8	Aborting and completing a positioning process	39
8.9	Override for speed	42
9	Description of the technology functions	44
9.1	Single cycle operation	44
9.2	Continuous driving operation	46
9.3	Loading of starting position	48
9.4	Position synchronism	50
9.4.1	Ramp-based synchronisation process	50
	Index	54
	Your opinion is important to us	55

1 About this document

This documentation ...

- contains detailed information on the functionalities of the "Virtual Master Base" technology module;
- is part of the "Controller-based Automation" manual collection. It consists of the following sets of documentation:

Documentation type	Subject
Product catalogue	Controller-based Automation (system overview, sample topologies) Lenze controllers (product information, technical data)
System manuals	Visualisation (system overview/sample topologies)
Communication manuals Online helps	Bus systems • Controller-based Automation EtherCAT® • Controller-based Automation CANopen® • Controller-based Automation PROFIBUS® • Controller-based Automation PROFINET®
Reference manuals Online helps	Lenze controllers: • Controller 3200 C • Controller c300 • Controller p300 • Controller p500
Software manuals Online helps	Lenze engineering tools: • »PLC Designer« (programming) • »Engineer« (parameterisation, configuration, diagnostics) • »VisiWinNET® Smart« (visualisation) • »Backup & Restore« (backup, restore, update)

Further sets of technical documentation with regard to Lenze products

Further information on Lenze products which can be used in conjunction with Controller-based Automation can be found in the following sets of documentation:

Design / configuration / technical data	
☐	Product catalogues • Controller-based Automation • Controllers • Inverter Drives/Servo Drives
Mounting and wiring	
	Mounting instructions • Controllers • Communication cards (MC-xxx) • I/O system 1000 (EPM-Sxxx) • Inverter Drives/Servo Drives • Communication modules
☐	Hardware manuals • Inverter Drives/Servo Drives
Parameterisation / configuration / commissioning	
☐	Online help/reference manuals • Controllers • Inverter Drives/Servo Drives • I/O system 1000 (EPM-Sxxx)
☐	Online help/communication manuals • Bus systems • Communication modules
Sample applications and templates	
☐	Online help / software and reference manuals • i700 application sample • Application Samples 8400/9400 • FAST Application Template Lenze/PackML • FAST technology modules

Symbols:

-  Printed documentation
- ☐ PDF file / online help in the Lenze engineering tool



Tip!

Current sets of documentation and software updates with regard to Lenze products can be found in the Download area at:

www.lenze.com

Target group

This documentation is intended for all persons who plan, program and commission a Lenze automation system on the basis of the Lenze FAST Application Software.

1 About this document

1.1 Document history

1.1 Document history

Version			Description
3.0	04/2021	TD06	Error corrections & supplements
2.0	12/2019	TD06	Error corrections & supplements
1.0	09/2019	TD29	First edition

1 About this document

1.2 Conventions used

This documentation uses the following conventions to distinguish between different types of information:

Type of information	Writing	Examples/notes
Numeric notation		
Decimal separator	Point	The decimal point is always used. For example: 1234.56
Text		
Program name	» «	»PLC Designer« ...
Variable identifiers	<i>Italics</i>	By setting <i>bEnable</i> to TRUE...
Function blocks	Bold	The L_MC1P_AxisBasicControl function block ...
Function libraries		The L_TT1P_TechnologyModules function library ...
Source code	Font "Courier new"	... dwNumerator := 1; dwDenominator := 1; ...
Symbols		
Page reference	 6	Reference to further information: Page number in PDF file.

Variable identifiers

The conventions used by Lenze for the variable names of Lenze system blocks, function blocks, and functions are based on the "Hungarian Notation". This notation makes it possible to identify the most important properties (e.g. the data type) of the corresponding variable by means of its name, e.g. xAxisEnabled.

1 About this document

1.3 Definition of the notes used

1.3 Definition of the notes used

This document uses the following signal words and symbols to indicate hazards and important information:

Safety instructions

Layout of the safety instructions:



Pictograph and signal word!

(characterises the type of hazard and its severity)

Note

(describes the hazard and provides tips on how it can be avoided)

Pictograph	Signal word	Meaning
	Danger!	Danger of personal injury through dangerous electrical voltage Reference to an imminent danger that may result in death or serious injuries if the corresponding measures are not taken.
	Danger!	Danger of personal injury through a general source of danger Reference to an imminent danger that may result in death or serious injuries if the corresponding measures are not taken.
	Stop!	Danger of property damage Reference to a potential hazard that may cause damage to material assets if the corresponding measures are not taken.

Application notes

Pictograph	Signal word	Meaning
	Note!	Important note for trouble-free operation
	Tip!	Useful tip for simple handling
		Reference to another documentation

2 Safety instructions

Please observe the safety instructions in this documentation when you want to commission an automation system or a plant with a Lenze Controller.



The device documentation contains safety instructions which must be observed!

Read the documentation supplied with the components of the automation system carefully before you start commissioning the Controller and the connected devices.

DANGER!

High electrical voltage

Injury to persons caused by dangerous electrical voltage

Possible consequences

Death or severe injuries

Protective measures

Switch off the voltage supply before working on the components of the automation system.

After switching off the voltage supply, do not touch live device parts and power terminals immediately because capacitors may be charged.

Observe the corresponding information plates on the device.

DANGER!

Injury to persons

Risk of injury is caused by ...

- unpredictable motor movements (e.g. unintended direction of rotation, too high velocities or jerky movement);
- impermissible operating states during the parameterisation while there is an active online connection to the device.

Possible consequences

Death or severe injuries

Protective measures

- If required, provide systems with installed inverters with additional monitoring and protective devices according to the safety regulations valid in each case (e.g. law on technical equipment, regulations for the prevention of accidents).
 - During commissioning, maintain an adequate safety distance to the motor or the machine parts driven by the motor.
-

NOTICE

Damage or destruction of machine parts

Damage or destruction of machine parts can be caused by ...

- Short circuit or static discharges (ESD);
- unpredictable motor movements (e.g. unintended direction of rotation, too high velocities or jerky movement);
- impermissible operating states during the parameterisation while there is an active online connection to the device.

Protective measures

- Always switch off the voltage supply before working on the components of the automation system.
 - Do not touch electronic components and contacts unless ESD measures were taken beforehand.
 - If required, provide systems with installed inverters with additional monitoring and protective devices according to the safety regulations valid in each case (e.g. law on technical equipment, regulations for the prevention of accidents).
-

3 Overview of the technology module functions

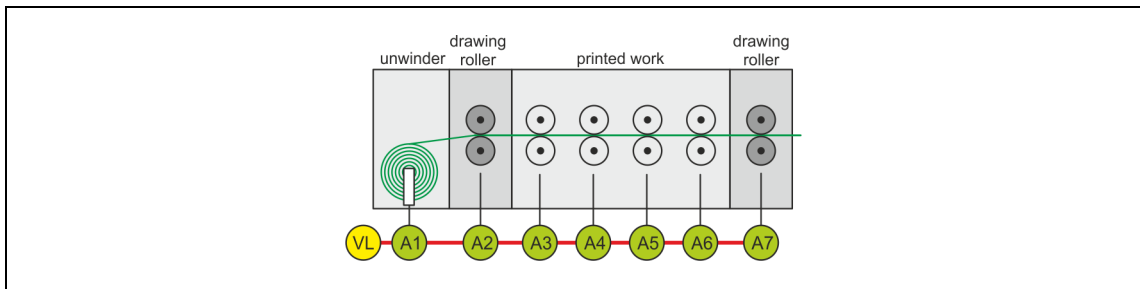
3.1 Functional description

3 Overview of the technology module functions

3.1 Functional description

A master axis (master) is higher-level in relation to other axes (slaves) and specifies setpoints for these. The master axis thus coordinates the movements of the slave axes, e.g. for a synchronism.

A virtual master axis is not a real existing axis; it only serves as reference variable for the slave axes to be coordinated.



[3-1] Application example for a virtual master axis (VMA) that coordinates 7 axes (Ax)

With this standardized application, a virtual master axis can be projected in the machine.

3.2

TM-functional overview

Functionality	Description
Basic Motion Base (V2):	
Manual jog (jogging) (📖 25)	Traverse axis in manual operation
Homing (📖 27)	Reference the axis
OSP application (📖 30)	Stop motion via an application quick stop deceleration ramp (AxisRef)
Halt (📖 31)	Stop motion via a stop deceleration ramp
Relative positioning (📖 34)	Move axis to a target position relative to the current position
Continuous travel with a defined velocity (📖 37)	Execute continuous travel
Aborting and completing a positioning process (📖 39)	End the positioning
Override for speed (📖 42)	Specify overrides for speed
Virtual Master Base (V2):	
Single cycle operation (📖 44)	For clocked applications
Continuous driving operation (📖 46)	For synchronous operations
Loading of starting position (📖 48)	Adjust position of the virtual master with the position of the real master axis
Ramp-based synchronisation process (📖 50)	Synchronization of the virtual master axis from its current position to the real master axis via a profile generator

Furthermore the technology module can be used for status and error messages and for resetting an error.

3.3

Important notes on how to operate the technology module

Start of the axes

The start of the axes after the controller has been inhibited depends on the requested movement type.

A distinction is made between the movement type with level-activated function and the movement type with edge-activated function.

- The movement type with level-activated function is automatically activated when the controller is enabled, as long as no higher priority function is active. This movement type includes
 - [Manual jog \(jogging\)](#) (□ 25)
 - [QSP application](#) (□ 30)
- The movement type with edge-activated function must be activated with a FALSE→TRUE edge after the controller has been enabled. This movement type includes
 - [Homing](#) (□ 27)



Example for a movement type with level-activated function: manual jog (jogging)

1. In the inhibited axis state ($xOperationEnabled = FALSE$), $xJogPos$ is set to TRUE.
 - $xEnableOperation = FALSE$ (axis is inhibited.)
==> **STANDBY** state ($xOperationEnabled = FALSE$)
 - $xJogPos = TRUE$ (manual jog is to be executed.)
2. Enable axis.
 - $xEnableOperation = TRUE$
==> **JOGGING** state ($xOperationEnabled = TRUE$)



Example for a movement type with edge-activated function: homing

1. In the inhibited axis state ($xOperationEnabled = FALSE$), $xHomeExecute$ is set to TRUE.
 - $xEnableOperation = FALSE$ (axis is inhibited.)
==> **STANDBY** state ($xEnableOperation = FALSE$)
 - $xHomeExecute = TRUE$ (referencing is to be executed.)
2. Enable axis.
 - $xEnableOperation = TRUE$
==> **STANDBY** state (output $xOperationEnabled = TRUE$)
3. Execute homing.
 - $xHomeExecute = FALSE \rightarrow TRUE$
==> **HOMING** state

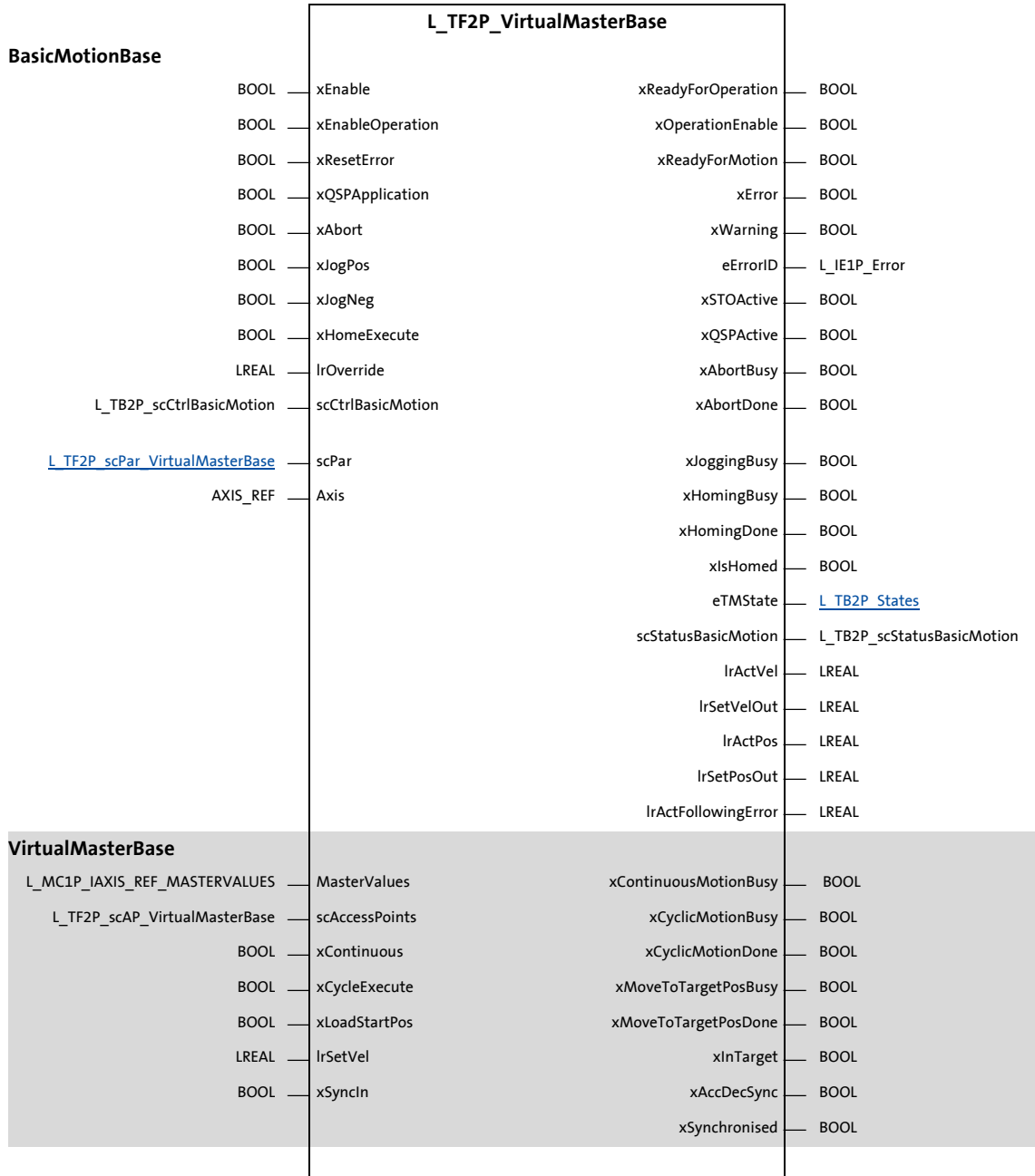
4 Application ranges

Typical applications:

- Synchronous operations
- Clocked systems

5 L_TF2P_VirtualMasterBase function block

The figure shows the affiliation of the inputs and outputs of the function block.



5.1

inputs

Designator	Data type	Description
xEnable	BOOL	Execution of the function block
		TRUE The function block is executed.
		FALSE The function block is not executed.
xEnableOperation	BOOL	TRUE The axis is switched to ready for operation. Controller enable is issued.
xResetError	BOOL	TRUE Reset axis error or software error.
xQSPApplication	BOOL	TRUE Active motion is aborted and axis is brought to a standstill via the OSP application deceleration ramp (□ 30). - The state changes to "STOP". - The technology module remains in the "STOP" state as long as <i>xQSPApplication</i> = TRUE.
xAbort	BOOL	TRUE Active motion is aborted and axis is brought to a standstill via the active profile deceleration. The state changes to "STOP". The deceleration ramp can be aborted by activating a new motion (e.g. <i>xMoveVelExecute</i>)
xJogPos	BOOL	TRUE Traverse axis in positive direction (manual jog). If <i>xJogNeg</i> is also TRUE, the InvalidInputJogPosJogNeg error message is output via the <i>eErrorID</i> output and the axis stops.
xJogNeg	BOOL	TRUE Traverse axis in negative direction (manual jog). If <i>xJogPos</i> is also TRUE, the InvalidInputJogPosJogNeg error message is output via the <i>eErrorID</i> output and the axis stops.
xHomeExecute	BOOL	The input is edge-controlled and evaluates the rising edge.
		FALSE Start homing.
		TRUE The function is aborted via the <i>scCtrlBasicMotion.xHalt</i> structure input.
IrOverride	LREAL	Using the <i>IrOverride</i> factor, the specified setpoint speed is multiplied (e.g. an override value of 0.5 generates half the setpoint speed) - Value range: 0 ... 1.0 - Initial value: 1.0 Note: The acceptance of the value is independent of the status in <i>xContinuousUpdate</i>
scCtrlBasicMotion L_TB2P_scCtrlBasicMotion		Structure of the status data You can find information on status data here: "Basic Motion Base V2" reference manual
scPar L_TF2P_scPar_VirtualMasterBase		The parameter structure contains the parameters of the technology module. ► L_TF2P_scPar_VirtualMasterBase (□ 19)
Axis	AXIS_REF	Reference to the axis
MasterValues L_MC1P_IAXIS_REF_MASTERVERVALUES		Connection for the master values (master axis)
scAccessPoints L_TF2P_scAP_VirtualMasterBase		Structure of the access points The access points can be used to influence the signal flow or technology module functions.

Designator	Data type	Description
xContinuous	BOOL	Start and stop continuous driving operation Continuous driving operation (□ 46) The setpoint speed for continuous driving operation is specified via the <i>IrSetVel</i> input. The function is aborted via synchronization to the master values of the master axis (<i>xSyncln</i> = TRUE). • Initial value: FALSE
		TRUE Execute continuous driving operation. The technology module is in the "CONTMOTION" state.
		TRUE → FALSE Stop continuous driving operation and move to the <i>scPar.IrTargetPos</i> target position.
xCycleExecute	BOOL	Execute single cycle operation Single cycle operation (□ 44) The setpoint speed for the single cycle is specified via the <i>IrSetVel</i> input. The input is edge-controlled and evaluates the rising edge. • Initial value: FALSE
		FALSE → TRUE Execute single cycle operation and move to the <i>scPar.IrTargetPos</i> target position. The technology module is in the "CYCLICMOTION" state and remains in this state until the motion has ended.
xLoadStartPos	BOOL	Loading of starting position Loading of starting position (□ 48) • This function can also be executed when the axis is inhibited or when <i>xQSPApplication</i> / <i>scCtrlBasicMotion.xHalt</i> = TRUE. • This function <u>cannot</u> be executed during position synchronism. • The input is edge-controlled and evaluates the rising edge. • Initial value: FALSE
		FALSE → TRUE Load starting position from <i>scPar.IrStartPos</i> .
IrSetVel	LREAL	Setpoint speed for continuous driving operation and single cycle operation. The direction of rotation is set via the <i>scPar.eDirection</i> parameter. New speed values are accepted at any time without a renewed edge change at one of the <i>xContinuous</i> or <i>xCycleExecute</i> control inputs. • Unit: unit/s • Initial value: 10 unit/s • Valid values: • Single cycle operation: positive values • Continuous driving operation: positive und negative values
xSyncln	BOOL	Synchronize master axes Ramp-based synchronisation process (□ 50) The velocity and position of the virtual master axis are matched to the real master axis. If synchronization is present, the <i>xSynchronized</i> output is set to TRUE. • Initial value: FALSE
		TRUE Start synchronization

5.2

outputs

Designator	Data type	Description
xReadyForOperation	BOOL	The axis is in correct working order. The DC-bus voltage is available. "Safe Torque Off" (STO) is not active.
xOperationEnabled	BOOL	The axis is enabled. The brake is released.
xReadyForMotion	BOOL	The axis is enabled and in correct working order. The brake is released.
xError	BOOL	TRUE There is an error in the technology module.

Designator	Data type	Description
xWarning	BOOL	TRUE There is a warning in the technology module.
eErrorID	L_IE1P_Error	ID of the error or warning message if <i>xError</i> = TRUE or <i>xWarning</i> = TRUE. "FAST technology modules" reference manual: Here you can find information on error or warning messages.
xSTOActive	BOOL	TRUE "Safe Torque Off" (STO) was triggered and is active. The axis is free of torque, the controller is inhibited.
xQSPActive	BOOL	TRUE The <i>xQSPApplication</i> input was set. The technology module is in the "STOP" state; the axis is brought to a standstill. The output is reset as soon as the axis is at standstill.
xAbortBusy	BOOL	TRUE The activated function is aborted via the <i>xAbort</i> input.
xAbortDone	BOOL	TRUE The activated function has been aborted via the <i>xAbort</i> input.
xJoggingBusy	BOOL	TRUE The Manual jog (jogging) (□ 25) function is active. The technology module is in the "JOGGING" state.
xHomingBusy	BOOL	TRUE The Homing (□ 27) is executed. The technology module is in the "HOMING" state.
xHomingDone	BOOL	TRUE Homing has been successfully completed.
xIsHomed	BOOL	TRUE The axis has been referenced ("Reference known"). <i>xIsHomed</i> may only be evaluated when the <i>scStatusBasicMotion.xAxisAvailable</i> status is reported.
eTMState	L_TB2P_States	Current state of the technology module
scStatusBasicMotion	L_TB2P_scStatusBasicMotion	Structure of the status data You can find information on status data here: "Basic Motion Base V2" reference manual
IrActVel	LREAL	Current speed • Unit: unit/s
IrSetVelOut	LREAL	Setpoint speed • Unit: unit/s
IrActPos	LREAL	Current position • Unit: unit
IrSetPosOut	LREAL	Setpoint position • Unit: unit
IrActFollowingError	LREAL	Current following error • Unit: unit
xContinuousMotionBusy	BOOL	TRUE Continuous driving operation is active. The technology module is in the "CONTMOTION" state.
xCyclicMotionBusy	BOOL	TRUE Single-cycle operation is active. The technology module is in the "CYCLICMOTION" state.
xCyclicMotionDone	BOOL	TRUE A single cycle was executed. The technology module continues to remain in the "CYCLICMOTION" state as long as the <i>xCycleExecute</i> input is set to TRUE.
xMoveToTargetPosBusy	BOOL	TRUE The axis moves to the <i>scPar.IrTargetPos</i> target position. The technology module is in the "MOVE_TO_TARGET_POS" state.
xMoveToTargetPosDone	BOOL	TRUE The travel to the target position has been successfully completed. The technology module remains in the "MOVE_TO_TARGET_POS" state.
xInTarget	BOOL	TRUE The axis is in the target position as per <i>scPar.IrTargetPos</i> and is at standstill.

Designator	Data type	Description	
xAccDecSync	BOOL	TRUE	The synchronization function is active. The virtual master axis is matched to the real master axis. The technology module is in the "ACC_DEC_SYNC" state.
xSynchronised	BOOL	TRUE	The velocity and position of the virtual master axis are matched to the real master axis. The technology module is in the "SYNCHRONIZED" state.

5.3

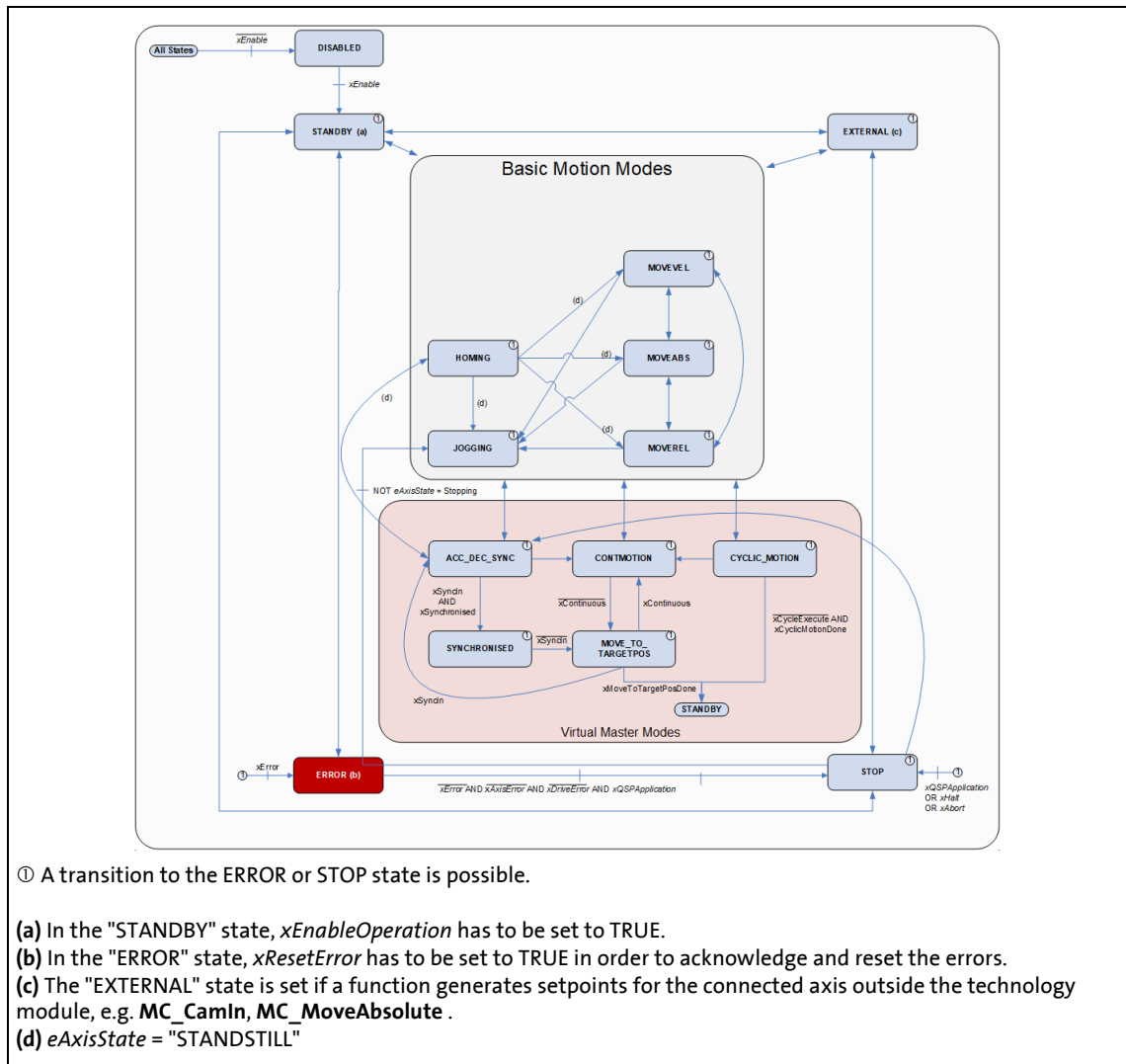
Parameter

L_TF2P_scPar_VirtualMasterBase

Designator	Data type	Description						
IrVel	LREAL	Speed for motions activated via the scCtrlBasicMotion structure. You can find information on status data here: "Basic Motion Base" reference manual - Unit: unit/s - Initial value: 3600						
IrAcc	LREAL	Acceleration for motions activated via the scCtrlBasicMotion structure. Specification of the maximum velocity variation which is to be used for acceleration. - Unit: unit/s² - Initial value: 36000						
IrDec	LREAL	Deceleration for motions activated via the scCtrlBasicMotion structure. Specification of the maximum speed variation which is to be used for deceleration to standstill. - Unit: unit/s² - Initial value: 100						
IrJerk	LREAL	Jerk for motions activated via the scCtrlBasicMotion structure. - Unit: unit/s³ - Initial value: 0						
eDirection	MC_DIRECTION	Traversing direction: <i>mcCurrentDirection</i>: Retain current direction (only in case of modulo axis) <i>mcPositiveDirection</i>: Positive direction <i>mcNegativeDirection</i>: Negative direction <i>mcShortestWay</i>: Shortest distance (only in case of modulo axis) - Initial value: <i>mcPositiveDirection</i>						
scHomeExtTP	MC_TRIGGER_REF	Transfer of an external touch probe event: - Only relevant for "external encoder" touch probe configuration. - For describing the MC_TRIGGER_REF structure, see the MC_TouchProbe function block.						
eAllowedDirection	L_TB2P_AllowedDirection	Allowed direction of rotation Initial value: positive and negative direction (0) <table><tr><td>0</td><td>Positive and negative direction of rotation</td></tr><tr><td>1</td><td>Only positive direction of rotation</td></tr><tr><td>2</td><td>Only positive direction of rotation</td></tr></table>	0	Positive and negative direction of rotation	1	Only positive direction of rotation	2	Only positive direction of rotation
0	Positive and negative direction of rotation							
1	Only positive direction of rotation							
2	Only positive direction of rotation							
IrSyncVel	LREAL	Maximum additional speed during the clutch-in/declutch process - Unit: unit/s - Initial value: 100						
IrSyncAcc	LREAL	Maximum additional acceleration during the clutch-in/declutch process - Unit: unit/s² - Initial value: 1000						
IrSyncDec	LREAL	Maximum additional deceleration during the clutch-in/declutch process - Unit: unit/s² - Initial value: 1000						
IrSyncJerk	LREAL	Maximum additional jerk during the clutch-in/declutch process - Unit: unit/s³ - Initial value: 100 000						

Designator	Data type	Description
IrStartPos	LREAL	Starting position to be loaded This starting position is accepted with a rising edge at the <i>xLoadStartPos</i> input - Unit: unit - Initial value: 0
IrTargetPos	LREAL	Target position depends on the operating mode: "Single-cycle operation" operating mode: Target position for a single cycle (input <i>xCycleExecute</i> = FALSE $\rightarrow$ TRUE). "Continuous driving operation" operating mode: Target position after termination of continuous driving operation (input <i>xContinuous</i> = TRUE $\rightarrow$ FALSE). "Position synchronism" operating mode: Target position to be approached after uncoupling from the real master axis (input <i>xSyncln</i> = TRUE $\rightarrow$ FALSE). Note when using modulo axes - The position outside the cycle length is calculated back into the modulo cycle. Positioning is carried out beyond the modulo cycle length to the target position <i>TargetPos</i>. - A negative position is not permitted. In the event of an error, the technology module switches with the error message <i>PosOutOfCycleLength</i> (20665) if a negative target position (<i>IrTargetPos</i> < 0) is assigned. - Unit: unit - Initial value: 0

6 State machine



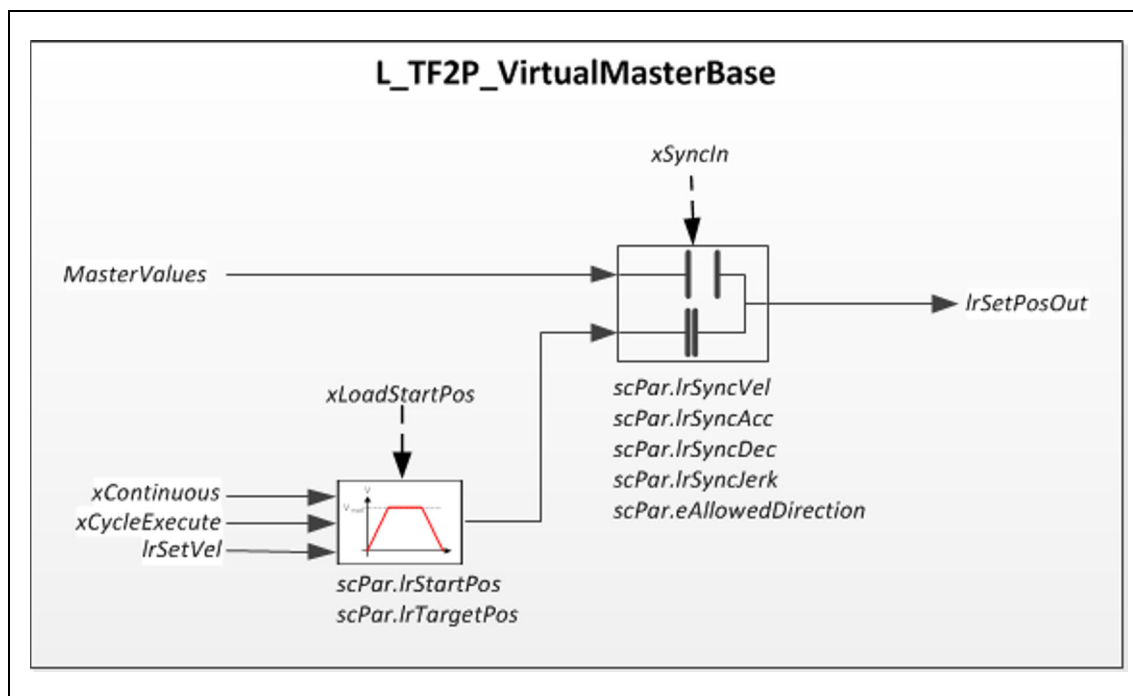
[6-1] State machine of the technology module

States of the eTMState output (L_TB2P_States)

No.	L_TB2P_States	Description
-1	DISABLED	Technology module deactivated
0	STANDBY	Technology module is on standby
1	STOP	Stop/Halt active
2	HOMING	Homing active
3	JOGGING	Manual jog active
4	MOVEVEL	Continuous travel with a defined velocity active
5	MOVEABS	Absolute positioning active
6	MOVEREL	Relative positioning active
7	EXTERNAL	An external function generates setpoints for the connected axis
8	ERROR	Error state
30	ACC_DEC_SYNC	Synchronization function active
31	SYNCHRONIZED	Axes are synchronized
36	MOVE_TO_TARGET_POS	Movement to target position active
60	CONTMOTION	Continuous driving operation
130	CYCLICMOTION	Single cycle operation active
1000	SYSTEMFAULT	System fault

7 Signal flow diagram

The following figure shows the main signal flow of the function:



[7-1] Signal flow of the technology module

8 Description of the basic functions

Overview of the basic functions

- [Manual jog \(jogging\)](#) (📖 25)
- [Homing](#) (📖 27)
- [QSP application](#) (📖 30)
- [Halt](#) (📖 31)
- [Absolute positioning](#) (📖 32)
- [Relative positioning](#) (📖 34)
- [Continuous travel with a defined velocity](#) (📖 37)
- [Aborting and completing a positioning process](#) (📖 39)
- [Override for speed](#) (📖 42)

8 Description of the basic functions

8.1 Manual jog (jogging)

8.1 Manual jog (jogging)

Prerequisite

- The axis is enabled (*xReadyForMotion* = TRUE).

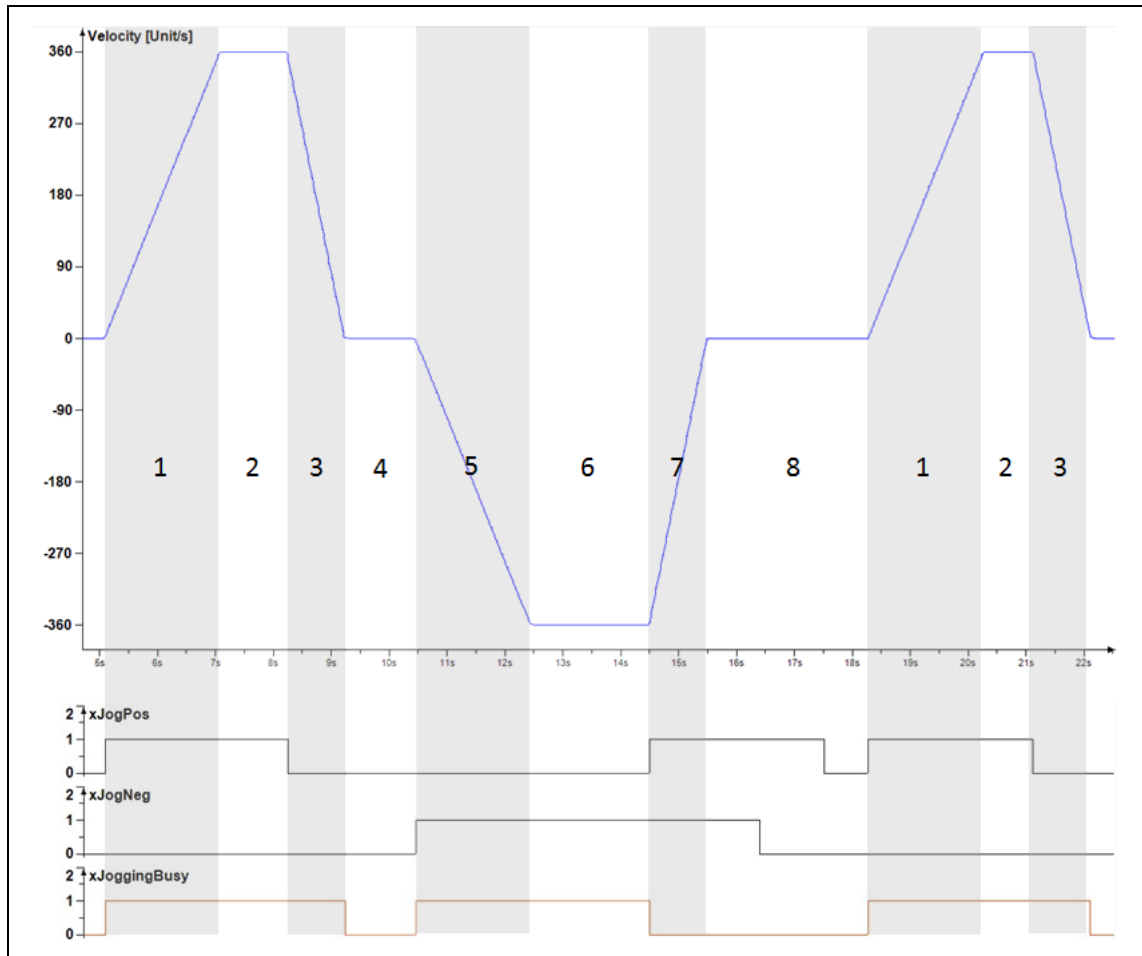
Execution

With the *xJogPos* input = TRUE, the axis is traversed in the positive direction and with the *xJogNeg* input = TRUE, it is traversed in the negative direction. The axis is traversed as long as *xJogPos* or *xJogNeg* remain set to TRUE.

The current travel command cannot be replaced by the other jog command. If *xJogPos* = TRUE and *xJogNeg* = TRUE, the axis brakes into standstill and the *eTMState* status output changes to the "ERROR" state.

Only if both inputs have been reset, the technology module changes back to the "STANDBY" state.

Travel diagram



No	xJogPos	xJogNeg	xJogging Busy	eTMState	Info
1	FALSE → TRUE	FALSE	FALSE → TRUE	STANDBY → JOGGING	The axis accelerates to the positive manual speed 360 units/s with 180 unit/s².
2	TRUE	FALSE	TRUE	JOGGING	The axis travels constantly with 360 units/s.
3	FALSE → TRUE	FALSE	TRUE	JOGGING	The axis decelerates to a standstill with 360 unit/s².
4	FALSE	FALSE	TRUE → FALSE	JOGGING → STANDBY	The axis has reached a standstill.
5	FALSE	FALSE → TRUE	FALSE → TRUE	STANDBY → JOGGING	The axis accelerates to the negative manual speed -360 units/s with 180 unit/s².
6	FALSE	TRUE	TRUE	JOGGING	The axis travels constantly with -360 units/s.
7	TRUE	TRUE	TRUE → FALSE	ERROR	The axis decelerates to a standstill with 360 unit/s².
8	FALSE	FALSE	FALSE	ERROR → STANDBY	The axis is at a standstill and changes to "STANDBY" if both inputs xJogPos and xJogNeg have the value FALSE.

Parameters to be set

The parameters for manual jog can be found in the AXIS_REF in subindex 181 ... 184.

The parameter values can be changed during operation. They are accepted and become effective immediately when setting the inputs xJogPos = TRUE or xJogNeg = TRUE.

The manual jog parameters from the PLC program can also be written via the L_MC1P_SetBasicParameter block.

8.2 Homing

Prerequisites

- The technology module is in "STANDBY" mode.
- The axis is enabled (*xReadyForMotion* = TRUE).

Execution

With a rising edge (FALSE→TRUE) at the *xHomeExecute* input, homing is started (output *xHomingBusy* = TRUE). The axis travels until the home position is reached.

When the reference switch has been reached, *xisHomed* switches to TRUE.

When the homing process has been completed successfully, *xHomingDone* changes to TRUE.

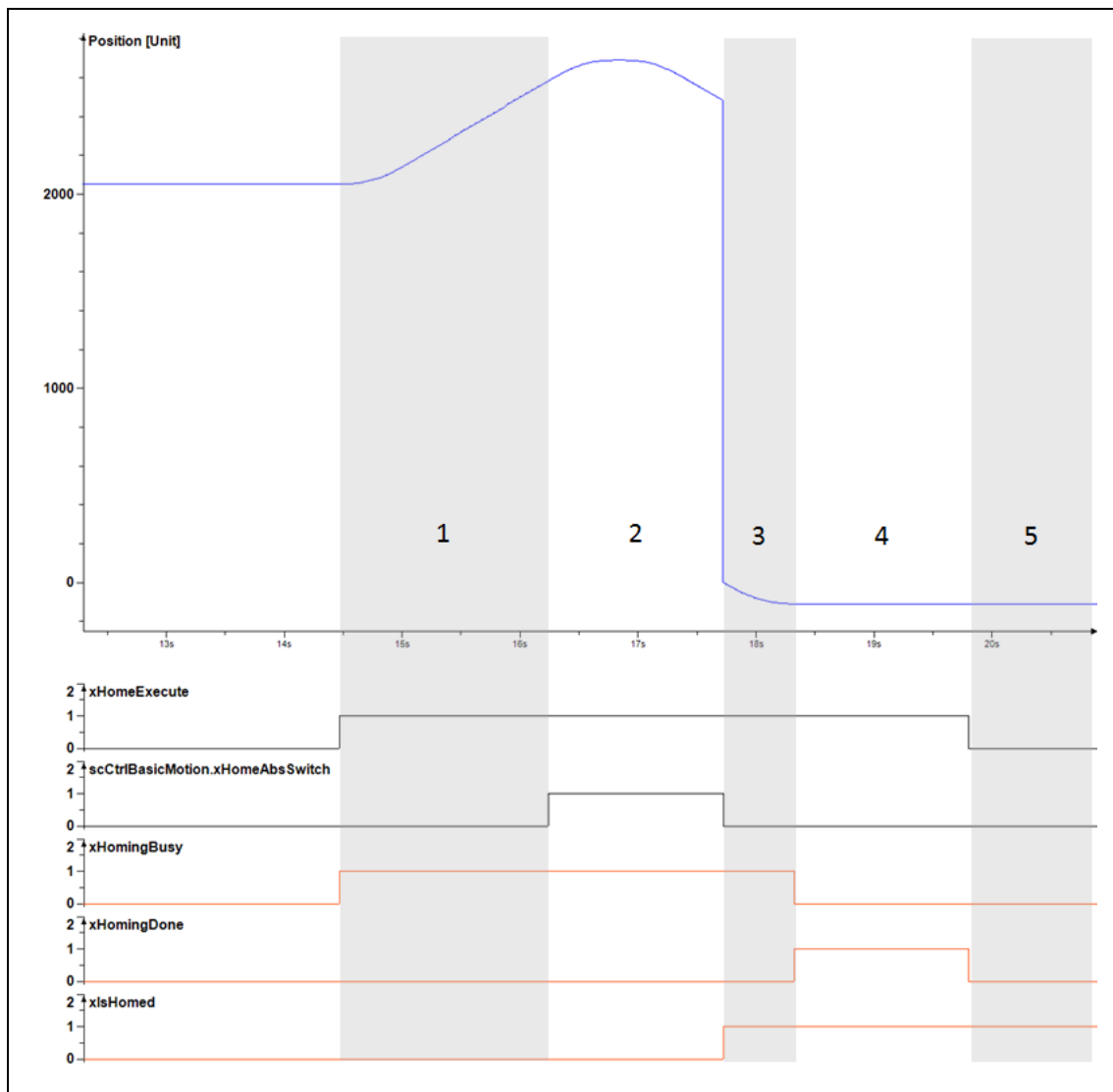
If *xHomeExecute* = FALSE, the state is changed to "STANDBY" and *xHomingDone* is set = FALSE.

Abort of the homing process

The function is aborted via the *xHalt* input.

Homing is not interrupted if *xHomeExecute* is set to = FALSE early.

Travel diagram



[8-1] Homing in the CwRpCwRn module with the home position "0"

No	xHome Execute	xHome AbsSwitch	xHoming Busy	xHoming Done	xIsHomed	eTMState	Info
1	FALSE → TRUE	FALSE	FALSE → TRUE	FALSE	FALSE	STANDBY → HOMING	The axis starts homing in positive direction.
2	TRUE	TRUE → FALSE	TRUE	FALSE	FALSE	HOMING	The axis changes the direction of rotation due to the positive edge of xHomeAbsSwitch.
3	TRUE	FALSE → TRUE	TRUE	FALSE	FALSE → TRUE	HOMING	The negative edge of xHomeAbsSwitch sets the position of the axis to the home position 0 units and the axis is brought to a standstill.
4	TRUE	FALSE	TRUE → FALSE	FALSE → TRUE	TRUE	HOMING	As soon as the axis has reached the standstill, xHomingBusy is set = FALSE and xHomingDone is set = TRUE. xHomingDone remains TRUE for as long as xHomeExecute is set = TRUE, but always at least for one clock cycle.
5	TRUE → FALSE	FALSE	FALSE	TRUE → FALSE	TRUE	HOMING → STANDBY	The axis is ready for operation and can be positioned in an absolute fashion.

Parameters to be set

The parameters for homing can be found in the `AXIS_REF` in subindex 70 ... 83.

The manual jog parameters from the PLC program can also be written via the `L_MC1P_SetBasicParameter` block.

8.3 QSP application

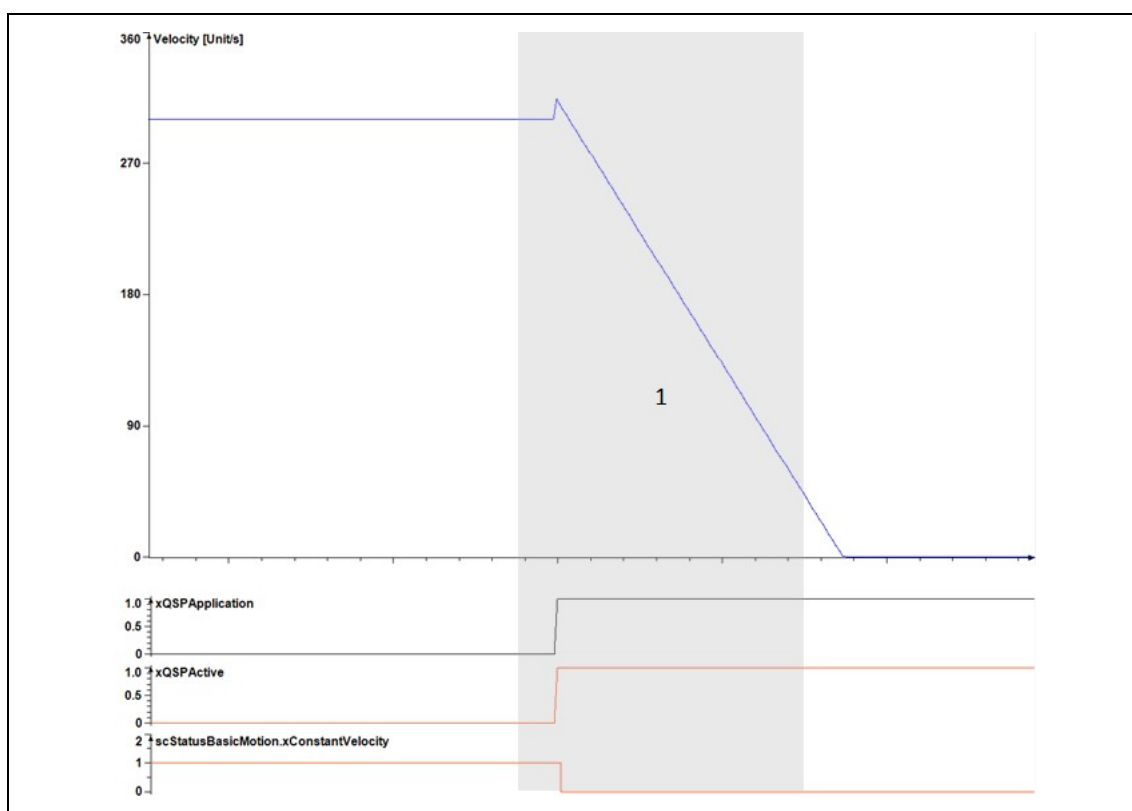
Prerequisite

- The axis is enabled ($xReadyForMotion = TRUE$).

Execution

With a rising edge ($FALSE \rightarrow TRUE$) at the $xQSPApplication$ input, the current movement of the axis is interrupted and the axis is brought to a standstill on the basis of its last current speed value on the QSP application deceleration ramp. The parameter for the QSP application deceleration ramp is set in the $AXIS_REF$.

Travel diagram



No	xQSP Application	xQSPActive	xConstant Velocity	eTMState	Info
1	$FALSE \rightarrow TRUE$	$FALSE \rightarrow TRUE$	$TRUE \rightarrow FALSE$	MOVEVEL → STOP	The speed setpoint of the axis is set to the last current speed and brought to a standstill via the QSP deceleration ramp.

Parameters to be set

The parameters for "QSP Application" can be found in the $AXIS_REF$ in subindex 48 and 49.

The manual jog parameters from the PLC program can also be written via the $L_MC1P_SetBasicParameter$ block.

8.4 Halt

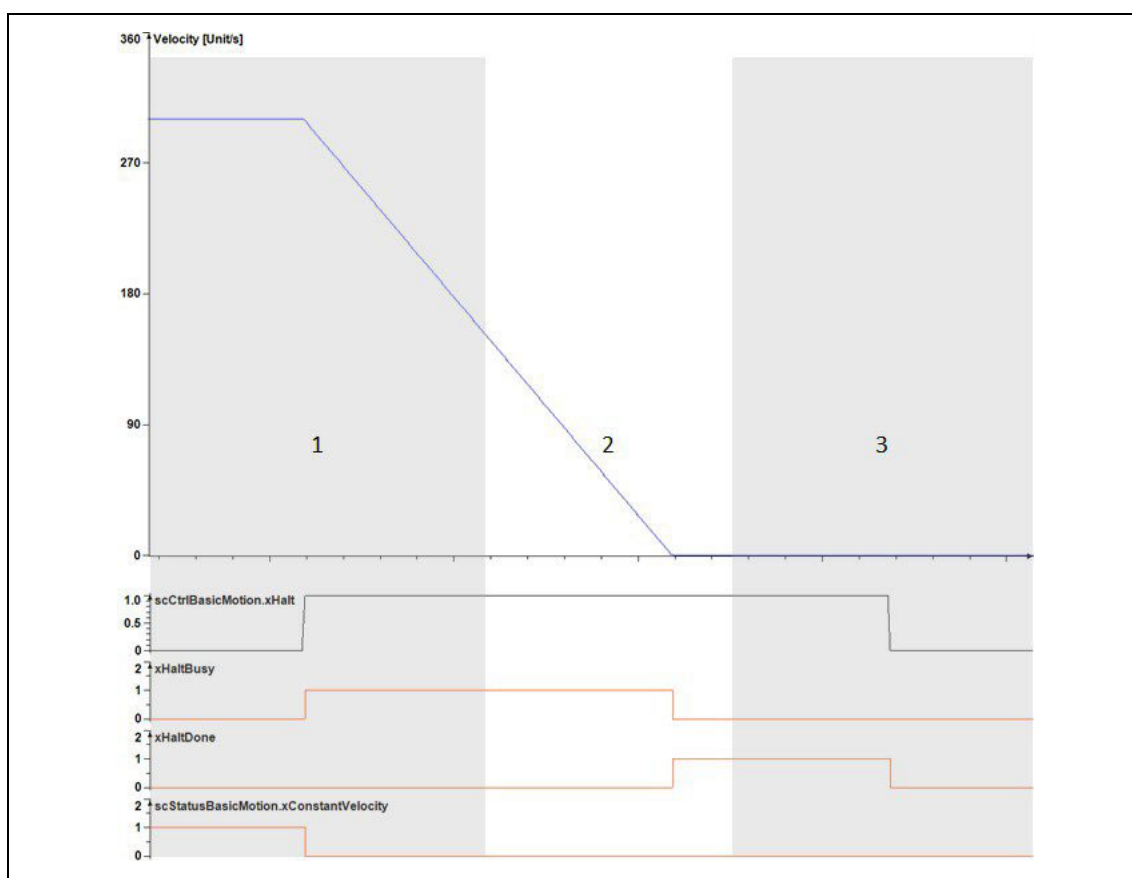
Prerequisite

- The axis is enabled ($xReadyForMotion = TRUE$).

Execution

With a rising edge ($FALSE \rightarrow TRUE$) at the `scCtrlBasicMotion.xHalt` structure input, the current movement of the axis is interrupted and the axis is brought to a standstill with a halt deceleration ramp.

Travel diagram



No	xHalt	xHaltBusy	xHaltDone	xConstant Velocity	eTMState	Info
1	$FALSE \rightarrow TRUE$	$FALSE \rightarrow TRUE$	$FALSE$	$TRUE \rightarrow FALSE$	$MOVEVEL \rightarrow STOP$	The axis decelerates with the halt deceleration ramp: $xHaltBusy = TRUE$ $xConstantVelocity = FALSE$.
2	$TRUE$	$TRUE \rightarrow FALSE$	$FALSE \rightarrow TRUE$	$FALSE$	$STOP$	The axis is brought to a standstill: $xHaltBusy = FALSE$ $xHaltDone = TRUE$
3	$TRUE \rightarrow FALSE$	$FALSE \rightarrow TRUE$	$TRUE \rightarrow FALSE$	$FALSE$	$STOP \rightarrow STANDBY$	$xHalt = FALSE$ $xHaltDone = TRUE \rightarrow FALSE$.

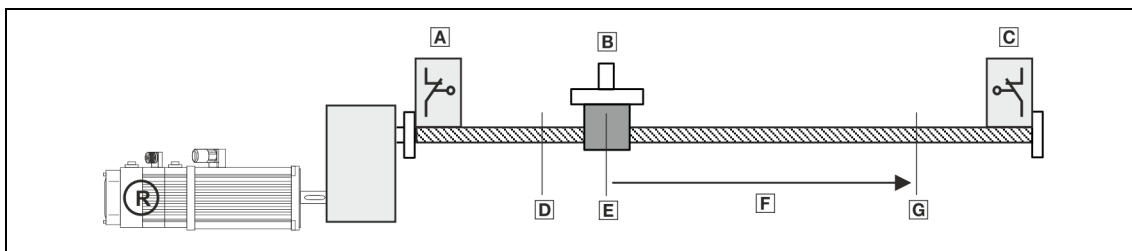
Parameters to be set

The parameter for the halt can be found in the `AXIS_REF` in subindex 186 and 187.

8 Description of the basic functions

8.5 Absolute positioning

8.5 Absolute positioning



[8-2] Example: Absolute positioning of a linear axis

Pos.	Description
A	Negative hardware limit switch
B	Load (e.g. slide)
C	Positive hardware limit switch
D	Zero position of the measuring system
E	Current actual position
F	Distance to be travelled
G	Target position (<i>lrPos_Dist</i> input)

With absolute positioning, the target position in the *lrPos_Dist* input (G) is defined as a unique absolute position, relating to the zero position of the measuring system (D). The distance to be travelled (F) results from the distance between the current actual position (E) and the absolute target position (G).

Prerequisites

- The axis is enabled (*xReadyForMotion* = TRUE).
- The axis is referenced (*xisHomed* = TRUE).

Execution

With a rising edge (FALSE→TRUE) at the *xMoveAbsExecute* input, a travel to the absolute target position in the *lrSetPos_Dist* input is started on the basis of the parameters for the motion profile (see below). The technology module changes to the "MOVEABS" state.

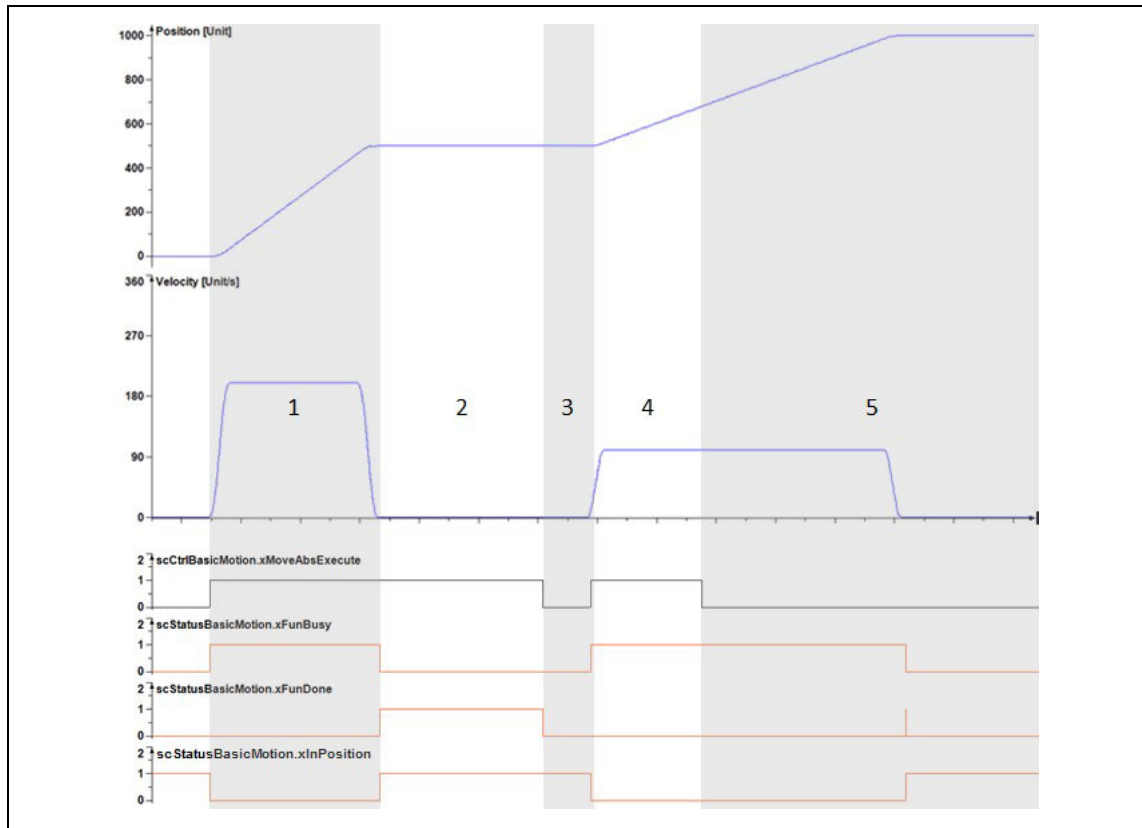
In order to reach the target position, the axis is brought to a standstill via the *lrDec* parameter. The axis reaches a standstill exactly in the target position.

After reaching the target position, the state changes back to STANDBY (*xMoveAbsExecute* = FALSE).

During the travel process, the absolute target position in the *lrPos_Dist* input can be changed, which requires triggering the Execute input for positioning again.

Positioning can be interrupted via *xAbort*, *xHalt* or *xQSPApplication*. Via *xFinishPositioning*, positioning can be continued again and completed.

Travel diagram



No	xMoveAbsExecute	xFuncBusy	xFuncDone	xInPosition	eTMState	Info
1	FALSE → TRUE	FALSE → TRUE	FALSE	TRUE → FALSE	STANDBY → MOVEABS	The axis starts the positioning run to the absolute target position 500 units.
2	TRUE	TRUE → FALSE	FALSE → TRUE	FALSE → TRUE	MOVEABS → STANDBY	When the target position has been reached, <i>xInPosition</i> is set to TRUE.
3	TRUE → FALSE	FALSE	TRUE → FALSE	TRUE	STANDBY	When <i>xMoveAbsExecute</i> is set to FALSE, the <i>xFuncDone</i> status signal changes from TRUE → FALSE.
4	FALSE → TRUE	FALSE → TRUE	FALSE	TRUE → FALSE	STANDBY → MOVEABS	The axis starts the positioning run to the absolute target position 1000 units.
5	TRUE → FALSE	TRUE → FALSE	FALSE → TRUE → FALSE	FALSE → TRUE	MOVEABS → STANDBY	After the target position has been reached, <i>xFuncDone</i> remains TRUE for exactly one cycle, because <i>xMoveAbsExecute</i> = FALSE has been requested during the positioning process.

Parameters to be set

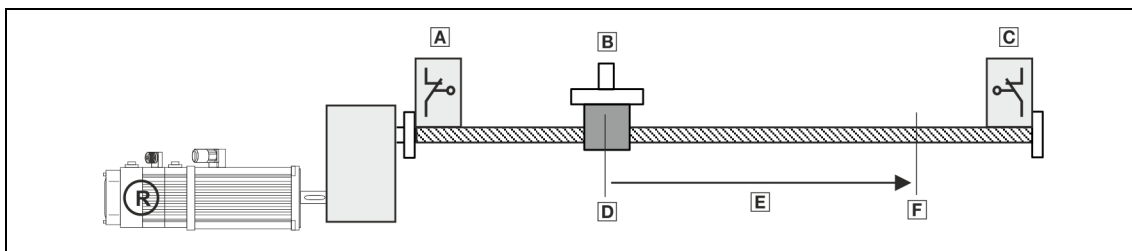
- *IrVel*, initial value: 3600 [units/s]
- *IrAcc*, initial value: 36000 [units/s²]
- *IrDcc*, initial value: 36000 [units/s²]
- *IrJerk*, initial value: 0 [units/s³]

Further parameters for absolute positioning can be found in the *AXIS_REF* in subindex 130 ... 137.

8 Description of the basic functions

8.6 Relative positioning

8.6 Relative positioning



[8-3] Example: Relative positioning of a linear axis

Pos.	Description
A	Negative hardware limit switch
B	Load (e.g. slide)
C	Positive hardware limit switch
D	Current actual position
E	Distance to be travelled (= <i>lrPos_Dist</i> input)
F	Target position (input current actual position + <i>lrPos_Dist</i>)

With relative positioning, the target position defined in the *lrPos_Dist* input corresponds to the distance to be travelled ($F = E$), meaning that the drive exactly travels the distance defined from the current actual position (D).

Prerequisite

- The axis is enabled (*xReadyForMotion* = TRUE).

Execution

With a rising edge (FALSE→TRUE) at the *xMoveRelExecute* input, a travel to the relative target position (*lrPos_Dist* input) is started on the basis of the parameters for the motion profile (see below). The technology module changes to the "MOVEREL" state.

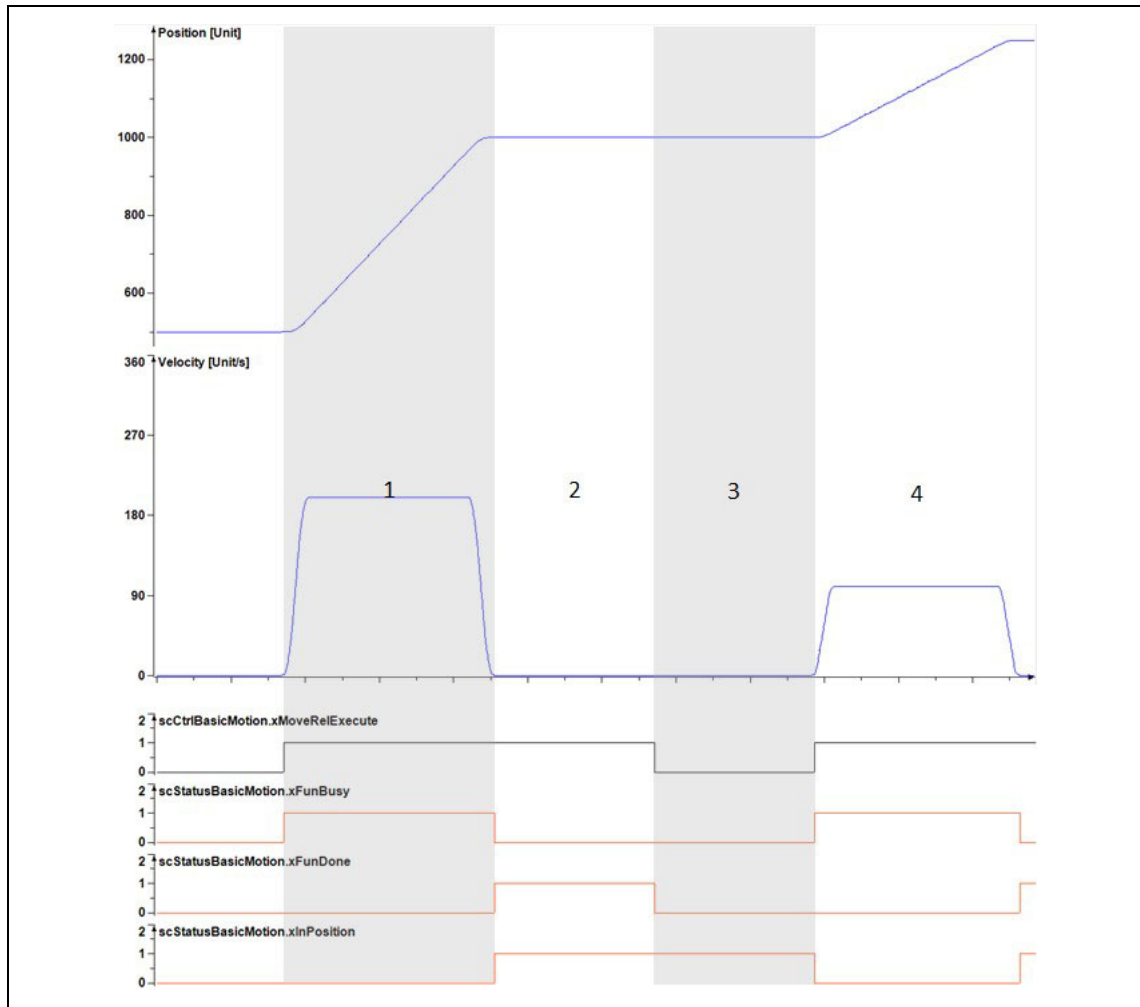
In order to reach the target position, the axis is brought to a standstill via the *lrDec* parameter. The axis reaches a standstill exactly in the target position.

After reaching the target position, the state changes back to STANDBY (*xMoveRelExecute* = FALSE).

During the travel process, the absolute target position in the *lrPos_Dist* input can be changed, which requires triggering the Execute input for positioning again.

Positioning can be interrupted via *xAbort*, *xHalt* or *xQSPApplication*. Via *xFinishPositioning*, positioning can be continued again and completed.

Travel diagram



No	xMoveRelExecute	xFuncBusy	xFuncDone	xInPosition	eTMState	Info
1	FALSE ↗ TRUE	FALSE ↗ TRUE	FALSE	FALSE	STANDBY → MOVEREL	The axis starts the positioning run to the relative target position 500 units.
2	TRUE	TRUE ↘ FALSE	FALSE ↗ TRUE	FALSE ↗ TRUE	MOVEREL → STANDBY	As soon as the relative target position has been reached, <i>xInPosition</i> and <i>xFuncDone</i> are set to TRUE.
3	FALSE ↗ TRUE	FALSE-	TRUE ↘ FALSE	TRUE	STANDBY	When <i>xMoveRelExecute</i> is set = FALSE, <i>xFuncDone</i> changes from TRUE ↘ FALSE.
4	FALSE ↗ TRUE	FALSE ↗ TRUE ↘ FALSE	TRUE ↘ FALSE ↗ TRUE	TRUE ↘ FALSE ↗ TRUE	STANDBY → MOVEREL → STANDBY	The axis starts the positioning run to the relative target position 250 units. After reaching the relative target position, <i>xInPosition</i> and <i>xFuncDone</i> are set to TRUE.

Parameters to be set

- *IrVel*, initial value: 3600 [units/s]
- *IrAcc*, initial value: 36000 [units/s²]
- *IrDcc*, initial value: 36000 [units/s²]
- *IrJerk*, initial value: 0 [units/s³]

Further parameters for relative positioning can be found in the AXIS_REF in subindex 130 ... 137.

8.7 Continuous travel with a defined velocity

Prerequisite

- The axis is enabled (*xReadyForMotion* = TRUE).

Execution

With a rising edge (FALSE → TRUE) at the *xMoveVelExecute* input, the continuous travel is started on the basis of the parameters for the motion profile (see below). The technology module changes to the "MOVEVEL" state.

The axis travels until a software limit position has been reached or a hardware limit switch has been approached:

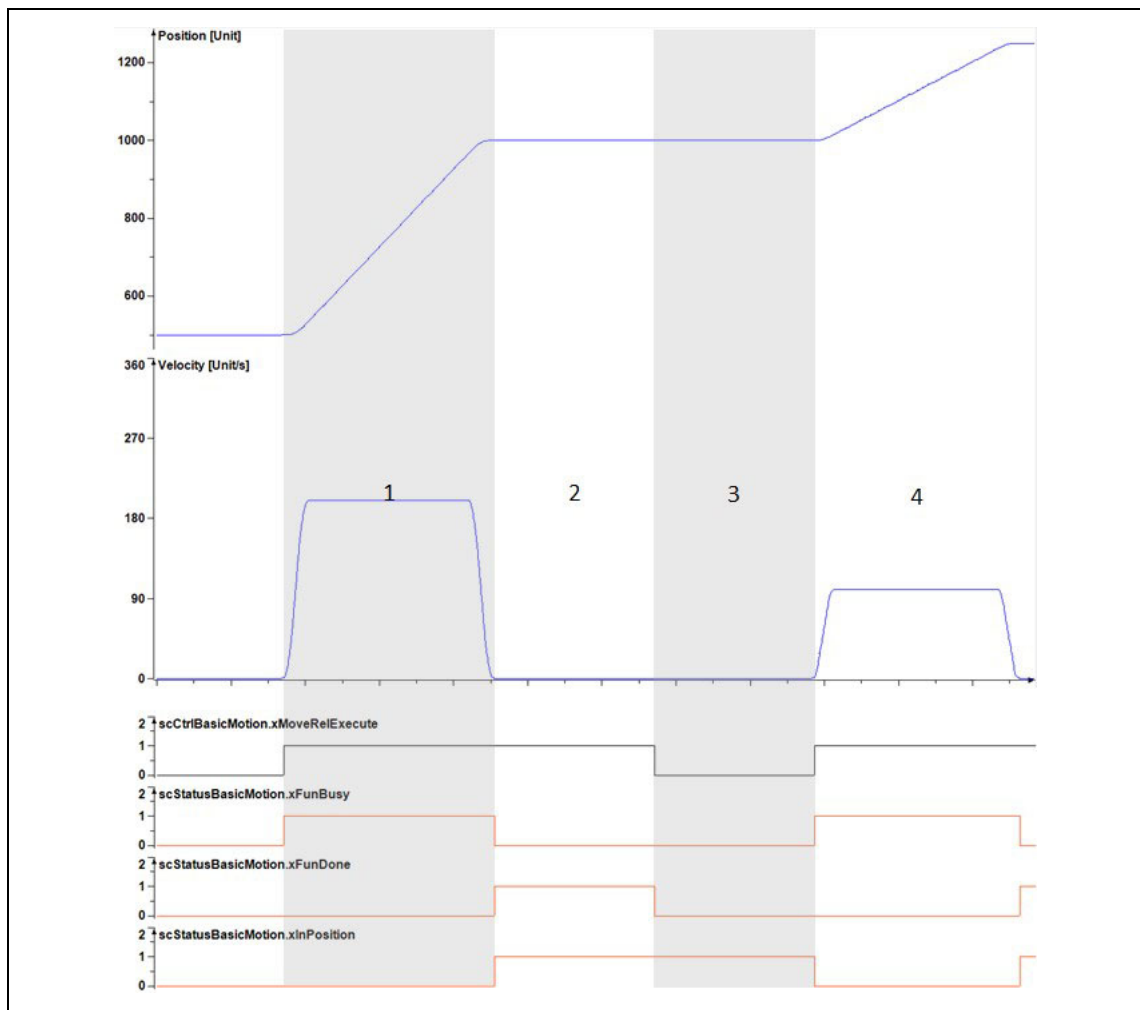
- Input *xDisableSWLimit* = FALSE
- Output *xSwLimitEnabled* = TRUE
- Output *xSwLimitSwitchPos/Neg* = TRUE
- Output *xHwLimitSwitchPos/Neg* = TRUE

When a software limit position has been reached, the axis is brought to a standstill via the *IrDec* parameter.

When a hardware limit switch has been reached, the axis is brought to a standstill via the QSP application deceleration ramp.

Then the state is changed back to "STANDBY".

Travel diagram



No	<code>xMoveVelExecute</code>	<code>xAbort</code>	<code>xFuncBusy</code>	<code>xFuncDone</code>	<code>xConstantVelocity</code>	<code>eTMState</code>	Info
1	FALSE ↗ TRUE ↘ FALSE	FALSE	FALSE ↗ TRUE	FALSE	FALSE	STANDBY → MOVEVEL	The axis accelerates to the target speed of 200 units/s.
2	FALSE	FALSE	TRUE	FALSE ↗ TRUE	FALSE ↗ TRUE	MOVEVEL	After reaching the target speed, <code>xFuncDone</code> changes from FALSE ↗ TRUE and <code>xConstantVelocity</code> changes from FALSE ↗ TRUE.
3	FALSE	FALSE ↗ TRUE	TRUE ↘ FALSE	TRUE ↘ FALSE ↗ TRUE	TRUE ↘ FALSE	MOVEVEL → STANDBY	The speed run is terminated by <code>xAbort = TRUE</code> and brought to a standstill with the profile deceleration. When standstill has been reached, <code>xFuncBusy</code> is set = FALSE.

Parameters to be set

- *IrVel*, initial value: 3600 [units/s]
- *IrAcc*, initial value: 36000 [units/s²]
- *IrDcc*, initial value: 36000 [units/s²]
- *IrJerk*, initial value: 0 [units/s³]

Further parameters for continuous travel can be found in the `AXIS_REF` in subindex 50 ... 53 and subindex 104 ... 105.

8.8**Aborting and completing a positioning process****Prerequisite**

- The axis is enabled (*xReadyForMotion* = TRUE).

Execution

An active positioning process is aborted with a rising edge (FALSE→TRUE) at the *xAbort* input.

The axis travels to a standstill with the deceleration ramp *scPar.IrDec*.

- **Exception: TM SpeedControl and TM TablePositioning**

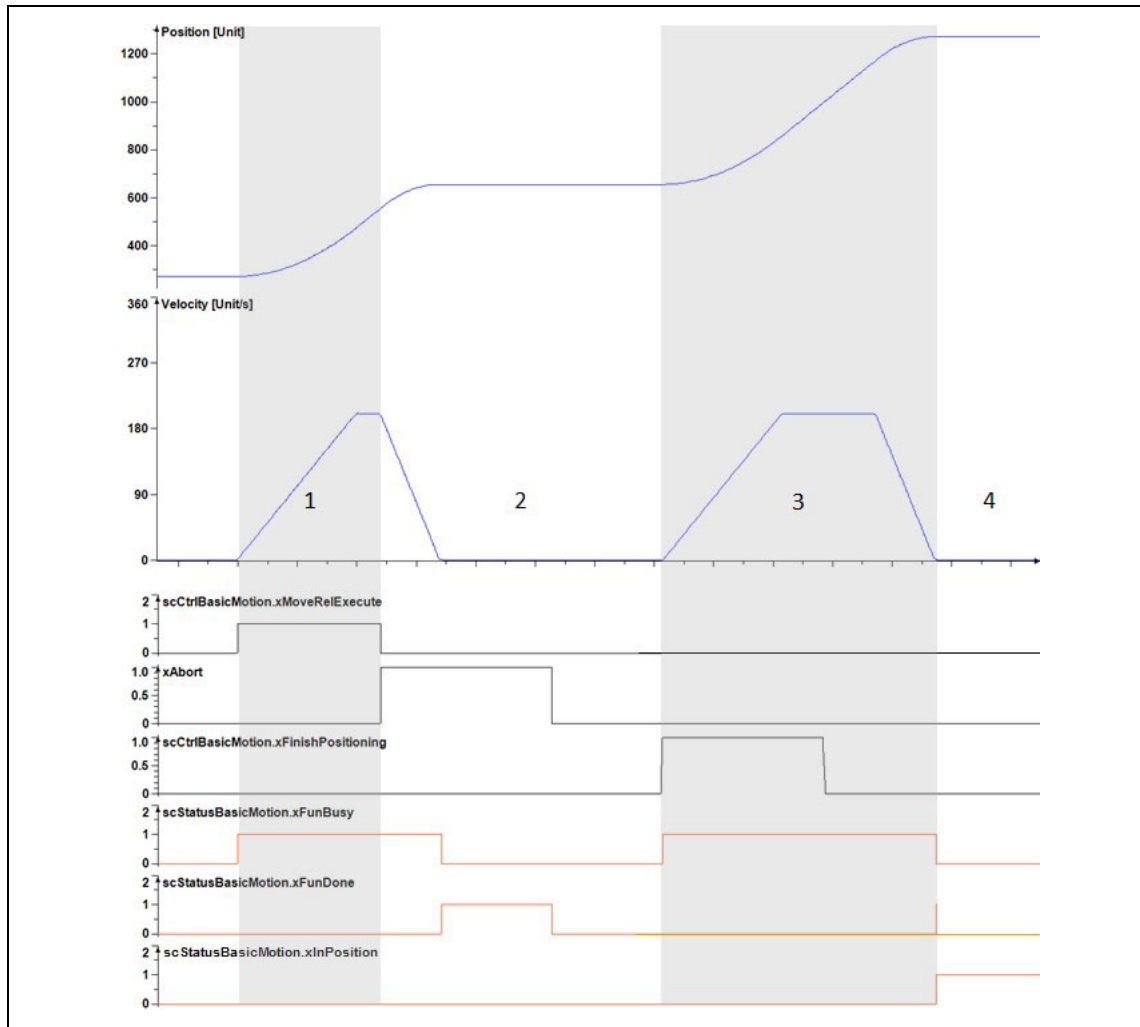
Both technology modules provide several profile data sets, one of which can be active.

If one of these profile data sets is active, the axis moves to standstill with the deceleration ramp of this active profile data set.

If no profile data set is active, the default deceleration ramp *scPar.IrDec* is used.

The positioning process is completed with a rising edge (FALSE→TRUE) at the *xFinishPositioning* input.

Travel diagram



No	xMoveRel Execute	xAbort	xFinish Positioning	xInPosition	eTMState	Info
1	FALSE ↗ TRUE	FALSE	FALSE	FALSE	STANDBY→ MOVEREL	The axis starts the positioning process with the relative target position 1000 units.
2	TRUE↘ FALSE	TRUE↘ FALSE	FALSE	FALSE	MOVEREL→ STOP→ STANDBY	The positioning process is aborted with <i>xAbsort</i> = TRUE. The axis is brought to a standstill with the profile deceleration.
3	FALSE	FALSE ↗ TRUE	FALSE↗ TRUE↘ FALSE	FALSE	STANDBY→ MOVEREL	With <i>xFinishPositioning</i> = TRUE, the positioning process is continued.
4	FALSE	FALSE	FALSE	FALSE ↗ TRUE	MOVEREL→ STANDBY	As soon as the relative target position has been reached, <i>xInPosition</i> and <i>xFunDone</i> are set to TRUE.
	FALSE ↗ TRUE	FALSE↗ TRUE↘ FALSE	TRUE↘ FALSE↗ TRUE	TRUE↘ FALSE↗ TRUE	STANDBY→ MOVEREL→ STANDBY	The axis starts the positioning process with the relative target position 250 units. After reaching the relative target position, <i>sxInPosition</i> and <i>xFunDone</i> are set to TRUE.

8.9

Override for speed

Prerequisite

- The axis is enabled ($xReadyForMotion = \text{TRUE}$).

Execution

In order to be able to set overrides for the active profile data, the following inputs are available at the technology module:

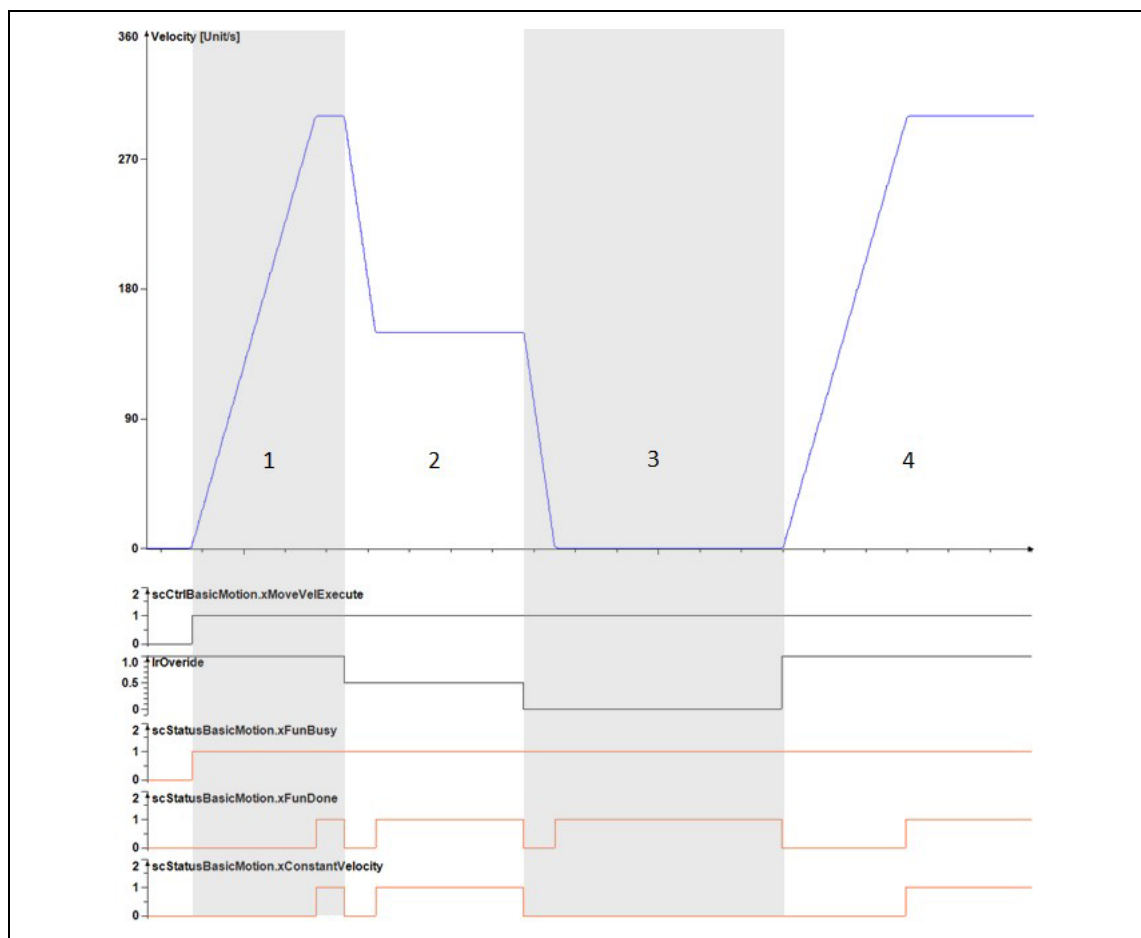
- *IrOverride*, initial value: 1.0
- *scCtrlBasicMotion.IrOverrideAcc*, initial value: 1.0 (100 %)
- *scCtrlBasicMotion.IrOverrideJerk*, initial value: 1.0 (100 %)

For profile-generating functions, separate overrides for speed, acceleration, deceleration and jerk can be set.

The values set act as a factor and are multiplied with the value active in each case.

Example

In this example, the use of the *IrOverride* input is shown.



8 Description of the basic functions

8.9 Override for speed

No	xMoveVel Execute	IrOverride	xFuncBusy	xFunDone	xConstant Velocity	eTMState	Info
1	FALSE ↗ TRUE	1	FALSE ↗ TRUE	FALSE ↗ TRUE	FALSE ↗ TRUE	STANDBY→ MOVEVEL	The axis accelerates to the target speed of 300 units/s.
2	TRUE	0.5	TRUE	TRUE↘ FALSE↗ TRUE	TRUE↘ FALSE↗ TRUE	MOVEVEL	The override value is set to 0.5 and the axis decelerates to 150 unit/s with the profile deceleration
3	TRUE	0	TRUE	TRUE↘ FALSE↗ TRUE	TRUE↘ FALSE	MOVEVEL	The override value is set to 0 and the axis is brought to a standstill to 0 unit/s with the profile deceleration
4	TRUE	1	TRUE	TRUE↘ FALSE↗ TRUE	FALSE ↗ TRUE	MOVEVEL	The override value is set to 1 and the axis accelerates to 300 unit/s with the profile deceleration

9 Description of the technology functions

9.1 Single cycle operation

Preconditions

- The technology module is in an error-free state.
- The drive is enabled (*xReadyForMotion* = TRUE).

Type

The single cycle operation is started when the *xCycleExecute* input = TRUE.

The cycle starts at the current *lrActPos* position of the virtual master axis and ends at the *scPar.lrTargetPos* target position.

The direction of rotation is defined using the *scPar.eDirection* parameter:

- Value "1" = positive direction (initial value)
- Value "2" = negative direction

The traveling speed is specified at the *lrSetVel* input. Only positive speeds are permissible.

If the single cycle operation is interrupted, e.g. via "QSP Application" or a clutch-in to the real master axis, the *xCycleExecute* input has to be reset to TRUE.

Inputs relevant for the function

- *xCycleExecute*
- *lrSetVel*, initial value: 10 [unit/s]

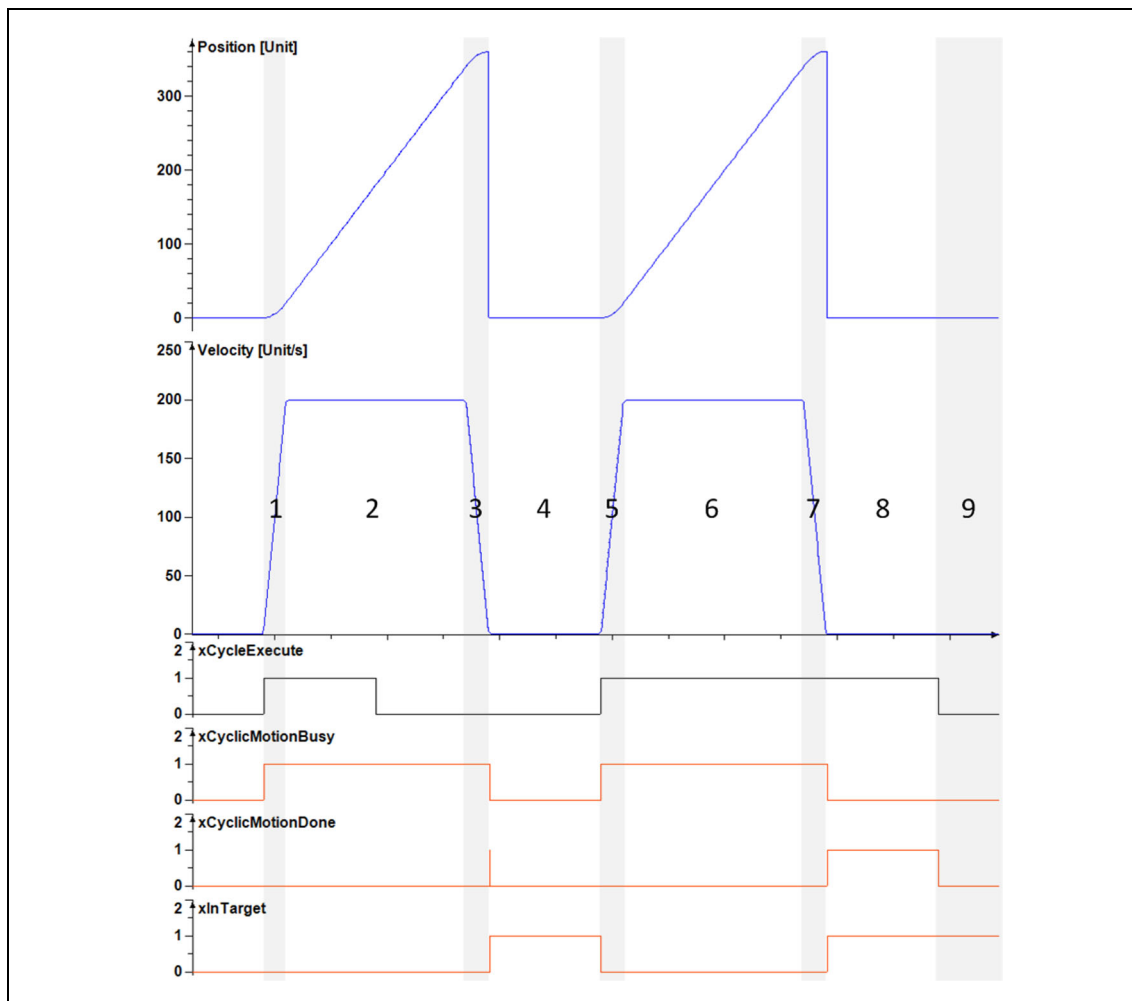
Outputs relevant for the function

- *xCyclicMotionBusy*
- *xCyclicMotionDone*
- *xInTarget*

Parameters relevant for the function

- *scPar.eDirection*, initial value: positive direction
- *scPar.lrAcc*, initial value: 36000 [unit/s²]
- *scPar.lrDec*, initial value: 36000 [unit/s²]
- *scPar.lrJerk*, initial value: 0 [unit/s³]

Example



[9-1] Single cycle operation

No	xCycleExecute	xCyclicMotionBusy	xCyclicMotionDone	xInTarget	eTMState	Info
1	FALSE → TRUE	FALSE → TRUE	FALSE	FALSE	STANDBY → CYCLICMOTION	Starting from the current position <i>IrActPos</i> = 0, the axis accelerates at <i>scPar.IrAcc</i> to the setpoint speed <i>IrSetVel</i> = 200 unit/s.
2	TRUE → FALSE	FALSE	FALSE	FALSE	CYCLICMOTION	The axis moves at a constant speed.
3	FALSE	TRUE	FALSE	FALSE	CYCLICMOTION	The axis decelerates at <i>scPar.IrDec</i> and comes to a standstill in the target position <i>scPar.IrTargetPos</i> = 0.
4	FALSE	TRUE → FALSE	FALSE → TRUE (for 1 cycle) → FALSE	FALSE → TRUE	CYCLICMOTION → STANDBY	Standstill
5	FALSE → TRUE	FALSE → TRUE	FALSE	TRUE → FALSE	CYCLICMOTION	Starting from the current position <i>IrActPos</i> = 0, the axis accelerates at <i>scPar.IrAcc</i> to the setpoint speed <i>IrSetVel</i> = 200 unit/s.
6	TRUE	TRUE	FALSE	FALSE	CYCLICMOTION	The axis moves at a constant speed.
7	TRUE	TRUE	FALSE	FALSE	CYCLICMOTION	The axis decelerates at <i>scPar.IrDec</i> and comes to a standstill in the target position <i>scPar.IrTargetPos</i> = 0.
8	TRUE	TRUE → FALSE	FALSE → TRUE	FALSE → TRUE	CYCLICMOTION	Standstill
9	TRUE → FALSE	FALSE	TRUE → FALSE	TRUE	CYCLICMOTION → STANDBY	Standstill

9.2 Continuous driving operation

Preconditions

- The technology module is in an error-free state.
- The drive is enabled (*xReadyForMotion* = TRUE).

Type

Continuous driving operation is started with the *xContinuous* input = TRUE and remains active until *xContinuous* = FALSE is set or until it is released by a synchronization process (*xSyncln* input = TRUE).

The cycle starts at the *lrSetPosOut* setpoint position of the virtual master axis and ends at the *scPar.lrTargetPos* target position.

The direction of rotation is defined using the *scPar.eDirection* parameter:

- Value "1" = positive direction (initial value)
- Value "2" = negative direction

The traveling speed is specified at the *lrSetVel* input. Positive and negative speeds are permissible in continuous driving operation.

If continuous driving operation is interrupted, e.g. via "QSP Application", the axis restarts after quitting the **STOP** state and switches to the **CONTMOTION** state.

If continuous driving operation has been released by clutch-in on the real master axis and the *xContinuous* input remains set to TRUE, then a transition directly back to continuous driving operation occurs upon declutching.

Inputs relevant for the function

- *xContinuous*
- *lrSetVel*, initial value: 10 [unit/s]

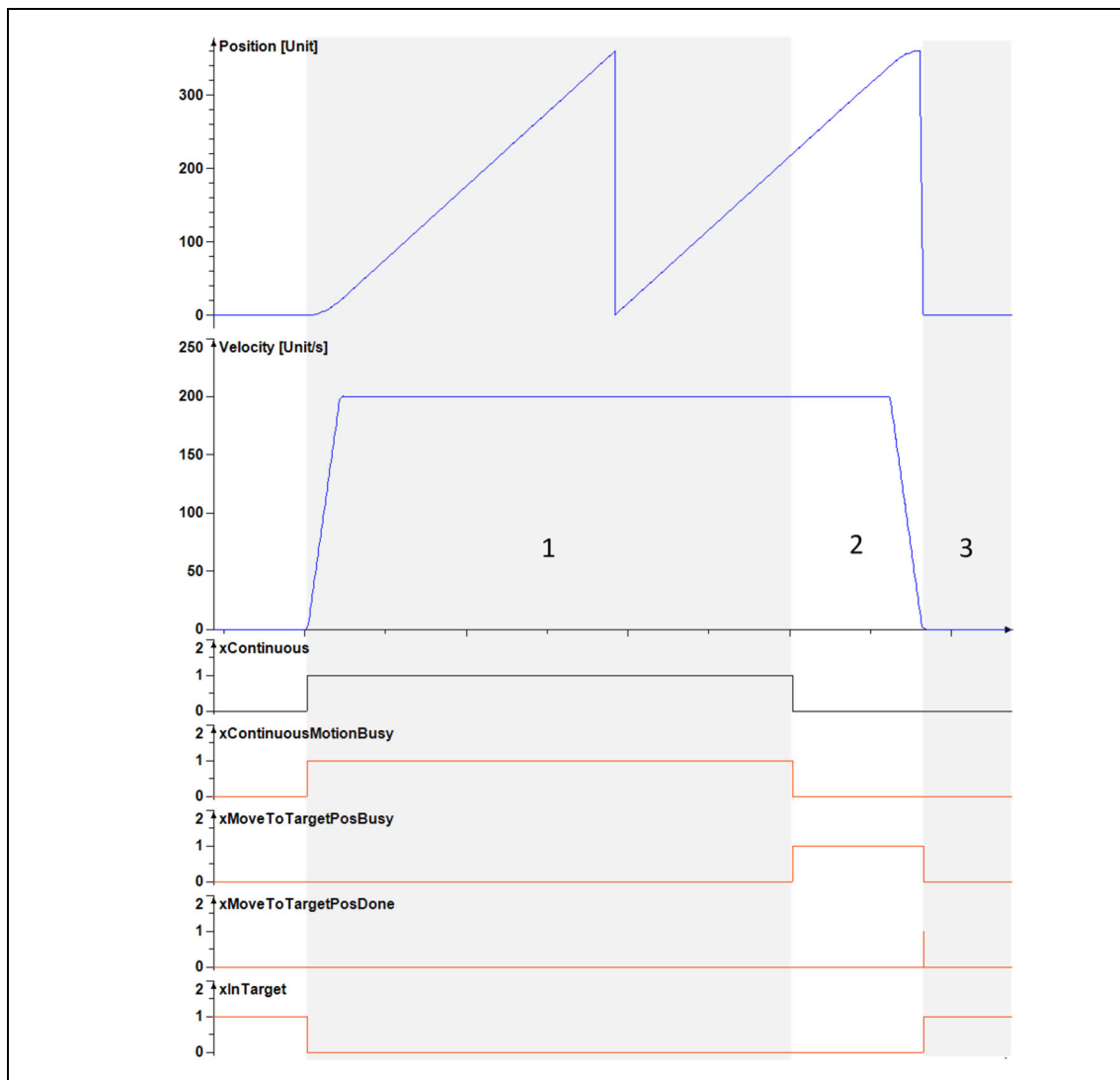
Outputs relevant for the function

- *xContinuousMotionBusy*
- *xMoveToTargetPosBusy*
- *xMoveToTargetPosDone*
- *xInTarget*

Parameters relevant for the function

- *scPar.eDirection*, initial value: positive direction
- *scPar.lrAcc*, initial value: 36000 [unit/s²]
- *scPar.lrDec*, initial value: 36000 [unit/s²]
- *scPar.lrJerk*, initial value: 0 [unit/s³]

Example



[9-2] Continuous driving operation

No	xContinuous	xContinuousMotionBusy	xMoveToTarget-PosBusy	xMoveToTarget-PosDone	xInTarget	eTMState	Info
1	FALSE→TRUE	FALSE→TRUE	FALSE	FALSE	TRUE→FALSE	STANDBY → CYCLICMOTION	Continuous driving operation is started. The axis accelerates at <i>scPar.IrAcc</i> to the setpoint speed <i>IrSetVel</i> = 200 unit/s and then moves at a constant speed.
2	TRUE→FALSE	TRUE→FALSE	FALSE→TRUE	FALSE	FALSE	CYCLICMOTION → MOVE_TO_TARGET_POS	Continuous driving operation is ended. The axis moves to the target position <i>scPar.IrTargetPos</i> = 0 and decelerates at <i>scPar.IrDec</i> to standstill.
3	FALSE	FALSE	TRUE→FALSE	FALSE→TRUE (for 1 cycle) → FALSE	FALSE→TRUE	MOVE_TO_TARGET_POS → STANDBY	Standstill

9.3 Loading of starting position

The "Load starting position" function supports the adjustment of the position of the virtual master with the position of the real master axis.

Example: The real master axis stands at 60°. When the "Load starting position" function is executed, the starting position of the virtual master is also set to 60°.

Preconditions

- The technology module is in an error-free state.
- The drive is enabled (*xReadyForMotion* = TRUE).

Type



Stop!

Damage to machine parts

Machine parts can be damaged by a "strike" at the drive shaft(s).

Possible consequences

Damage or destruction of machine parts

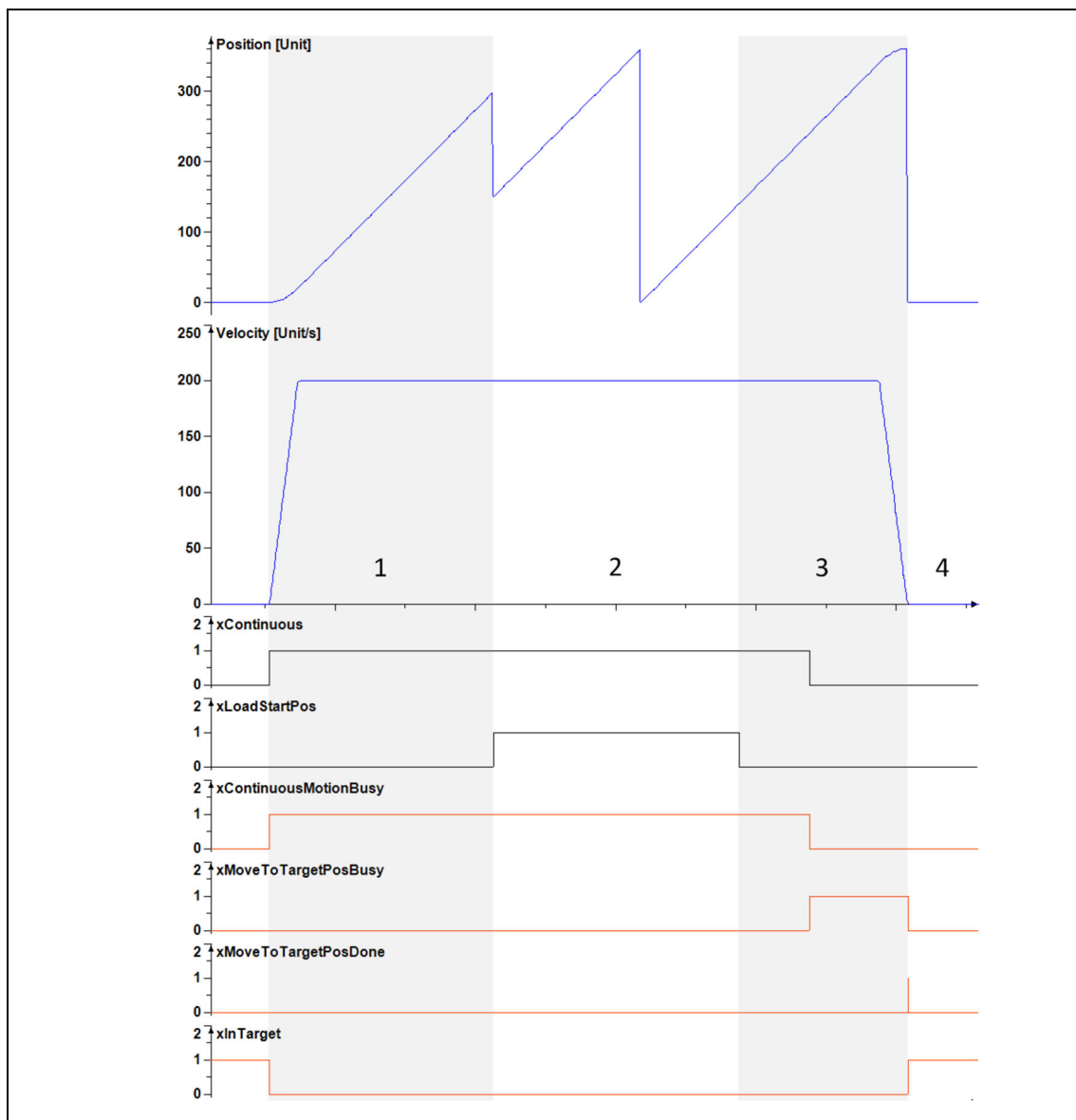
Protective measures

Only activate the "Load starting position" function if ...

- the master axis is at standstill or
- all following axes are decoupled.

When the *xLoadStartPos* input is set to TRUE, the set *scPar.IrStartPos* starting position is directly transferred ("without smoothing") as the *IrSetPosOut* setpoint position.

The "Load starting position" function cannot be executed in the **ERROR**, **ACC_DEC_SYNC** and **SYNCHRONIZED** states.



[9-3] "Load starting position" function

No	xContinuous	xLoadStartPos	xContinuousMotionBusy	xMoveToTargetPosBusy	xMoveToTargetPosDone	xInTarget	eTMState	Info
1	FALSE → TRUE	FALSE	FALSE → TRUE	FALSE	FALSE	TRUE → FALSE	STANDBY → CONTMOTION	Continuous driving operation is started. The axis accelerates at <i>scPar.IrAcc</i> to the setpoint speed <i>IrSetVel</i> = 200 unit/s and then moves at a constant speed.
2	TRUE	FALSE → TRUE	TRUE	FALSE	FALSE	FALSE	CONTMOTION	The starting position is loaded: The setpoint position of the axis <i>IrSetPosOut</i> jumps to the starting position <i>scPar.IrStartPos</i> = 150 unit. CAUTION: This position jump may lead to severe machine damage! Therefore the starting position should only be loaded at standstill!

No	xContinuous	xLoadStart Pos	xContinuous- MotionBusy	xMoveTo Target-PosBusy	xMoveTo Target-PosDone	xInTarget	eTMState	Info
3	TRUE↘FALSE	TRUE↘ FALSE	TRUE↘FALSE	FALSE↗TRUE	FALSE	FALSE	CONTMOTION → MOVE_TO_ TARGET_POS	Continuous driving operation is ended. The axis moves to the target position <i>scPar.IrTargetPos</i> = 0 and decelerates at <i>scPar.IrDec</i> to standstill.
4	FALSE	FALSE	FALSE	TRUE↘FALSE	FALSE↗TRUE (for 1 cycle) ↘FALSE	FALSE↗TRUE	MOVE_TO_TAR GET_POS → STANDBY	Standstill

9.4 Position synchronism

9.4.1 Ramp-based synchronisation process

Preconditions

- The technology module is in an error-free state.
- The drive is enabled (*xReadyForMotion* = TRUE).

Type

If the *xSyncIn* input is set to TRUE, the virtual master axis synchronizes from its current position to the real master axis via a profile generator.

Starting and target conditions for the synchronization:

- Standstill -> motion
- Motion -> standstill
- Motion -> motion
- Standstill -> standstill

The profile parameters *scPar.IrSyncVel*, *scPar.IrSyncAcc*, *scPar.IrSyncDec* and *scPar.IrSyncJerk* act in addition to the speeds and decelerations of the master axis.

The maximum allowed speed to catch up a position difference in addition to the master value is specified via the *scPar.IrSyncVel* parameter.

The allowed direction of rotation is set via the *scPar.eAllowedDirection* parameter.

Only adjustable for:

- Continuous driving operation (*xContinuous* input = TRUE)
- Synchronization with the real master axis (*xSyncMasterAxis* = TRUE)

The ramp-based synchronization process is independent of the movement of the real master axis. The virtual master axis synchronizes with the position if the real master axis is at a standstill.

Synchronisation can only be performed from these states:

- **STANDBY**: the virtual axis is synchronized to the standing or moving master from a standstill.
- **CONTMOTION** (continuous driving operation): the virtual axis is synchronized to the standing or moving master when moving.

Inputs relevant for the function

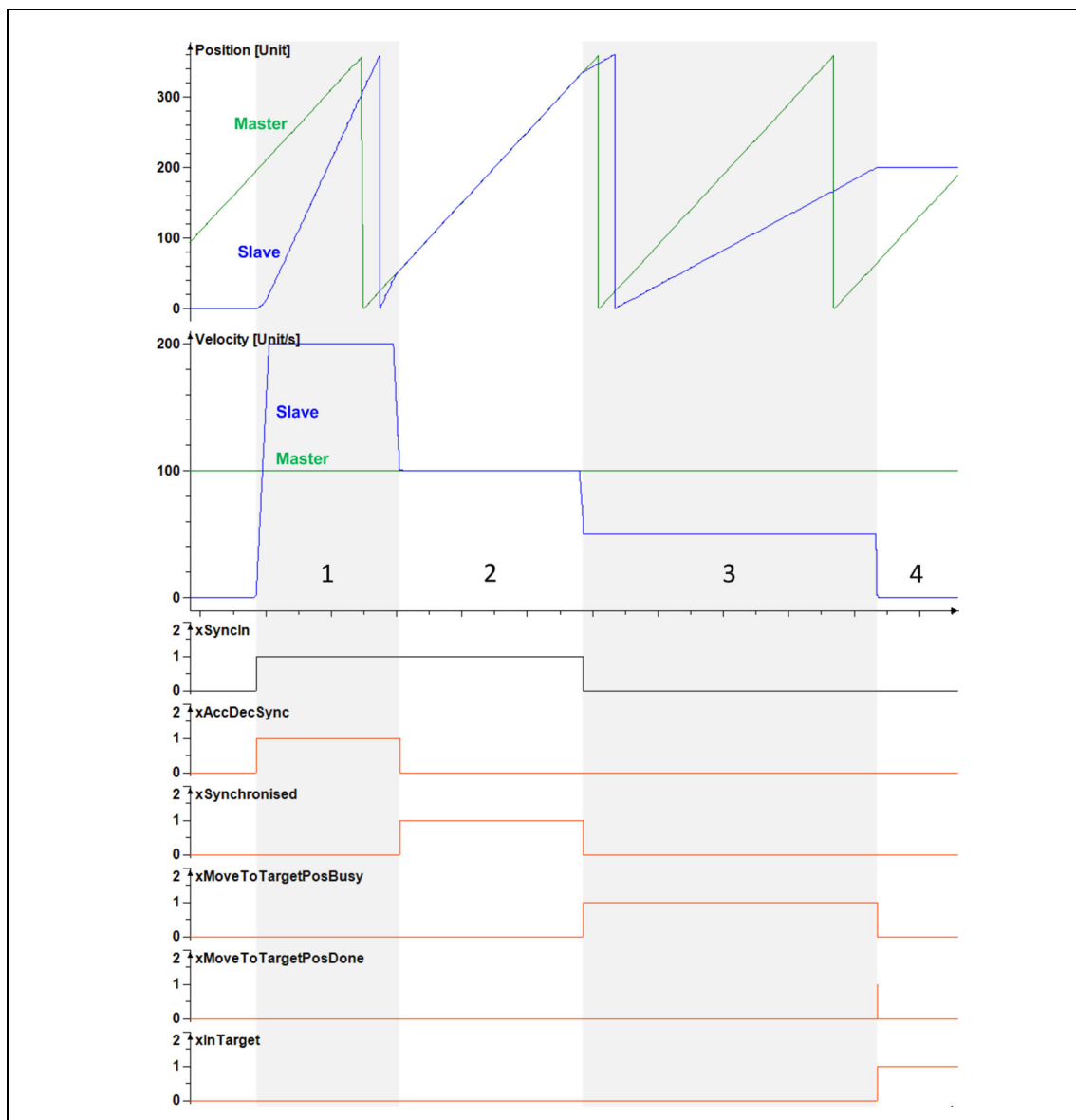
- *xSyncln*
- *lrSetVel*, initial value: 10 [unit/s]

Outputs relevant for the function

- *xAccDecSync*
- *xSynchronised*

Parameters relevant for the function

- *scPar.eAllowedDirection*, initial value: 0 (positive and negative direction)
- *scPar.lrSyncVel*, initial value: 100 [unit/s]
- *scPar.lrSyncAcc*, initial value: 1000 [unit/s²]
- *scPar.lrSyncDec*, initial value: 1000 [unit/s²]
- *scPar.lrSyncJerk*, initial value: 1 000 000 [unit/s³]



[9-4] "Load starting position" function

No	xSyncIn	xAccDecSync	xSynchronised	xMoveToTarget-PosBusy	xMoveToTarget-PosDone	xInTarget	eTMState	Info
1	FALSE → TRUE	FALSE → TRUE	FALSE	FALSE	FALSE	FALSE	STANDBY → ACC_DEC_SYNC	The ramp-based synchronization process is started and the axis accelerates at $scPar.lfSyncAcc$ and moves oversynchronously to the master speed (master speed = 100 unit/s + $scPar.lfSyncVel$ = 100 unit/s).
2	TRUE	TRUE → FALSE	FALSE → TRUE	FALSE	FALSE	FALSE	ACC_DEC_SYNC → SYNCHRONOUS	The synchronization process is completed and the virtual master axis moves with its position synchronized to the real master axis.

No	xSyncIn	xAccDecSync	xSynchronized	xMoveToTarget-PosBusy	xMoveToTarget-PosDone	xInTarget	eTMState	Info
3	TRUE↘ FALSE	FALSE	TRUE↘ FALSE	FALSE↗ TRUE	FALSE	FALSE	SYNCHRONIZED → MOVE_TO_TARGET_POS	The position synchronism is canceled and the virtual master axis moves at the setpoint speed <i>IrSetVel</i> = 50 unit/s to the target position <i>scPar.IrTargetPos</i> = 200 unit.
4	FALSE	FALSE	FALSE	TRUE↘ FALSE	FALSE↗ TRUE (for 1 cycle) ↘ FALSE	FALSE↗ TRUE	MOVE_TO_TARGET_POS → STANDBY	The axis comes to a standstill in the target position.



Note!

When declutching (*xSyncIn* = FALSE), if no setpoint speed is specified at the *IrSetVel* input, the axis decelerates at *scPar.IrDec* to a standstill and remains in the "MOVE_TO_TARGET_POS" state.

Symbole

"Virtual Master Base" TM parameters [19](#)

A

Aborting a positioning process [39](#)

Absolute positioning [32](#)

Application notes [7](#)

C

Competing a positioning process [39](#)

Continuous driving operation [46](#)

Continuous travel with a defined velocity [37](#)

Conventions used [6](#)

D

Document history [5](#)

E

E-mail to Lenze [55](#)

eTMState [22](#)

F

Feedback to Lenze [55](#)

Functional description of "Virtual Master Base (V2)" [10](#)

H

Halt, [31](#)

Homing [27](#)

I

inputs [15](#)

L

L_TB2P_States [22](#)

L_TF2P_scPar_VirtualMasterBase [19](#)

L_TF2P_scPar_VirtualMasterBase parameter structure [19](#)

L_TF2P_VirtualMasterBase [14](#)

L_TF2P_VirtualMasterBase function block [14](#)

Layout of the safety instructions [7](#)

Loading of starting position [48](#)

M

Manual jog (jogging) [25](#)

N

Notes on how to operate the technology module [12](#)

O

outputs [16](#)

Override for speed [42](#)

P

Position synchronisation (time-based) [50](#)

Position synchronism [50](#)

Q

QSP application [30](#)

R

Ramp-based synchronisation process [50](#)

Relative positioning [34](#)

S

Safety instructions [7](#), [8](#)

Signal flow diagram [23](#)

Signal flow of the "Virtual Master Base" technology module [23](#)

Single cycle operation [44](#)

Speed override [42](#)

Start of the axes [12](#)

State machine [21](#)

States [21](#)

States of the eTMState output [22](#)

T

Target group [4](#)

Technology module functions (overview) [11](#)

V

Variable identifiers [6](#)

Virtual Master Base (functional description) [10](#)



Your opinion is important to us

We have compiled these instructions to the best of our knowledge with the objective of supporting you optimally when you handle our product.

Perhaps we have not succeeded in achieving this objective in every respect. If you have suggestions for improvement, please e-mail us to:

feedback-docu@lenze.com

Thank you very much for your support.

Your Lenze documentation team

