

# Technology module



Basic Motion \_\_\_\_\_

Reference manual

EN

# Contents

---

|          |                                                         |           |
|----------|---------------------------------------------------------|-----------|
| <b>1</b> | <b>About this document</b>                              | <b>3</b>  |
| 1.1      | Document history                                        | 5         |
| 1.2      | Conventions used                                        | 5         |
| 1.3      | Layout of the safety instructions                       | 6         |
| <b>2</b> | <b>Safety instructions</b>                              | <b>7</b>  |
| <b>3</b> | <b>Overview of the technology module functions</b>      | <b>9</b>  |
| 3.1      | Important notes on how to operate the technology module | 10        |
| <b>4</b> | <b>L_TB2P_BasicMotionBase function block</b>            | <b>11</b> |
| 4.1      | Inputs                                                  | 12        |
| 4.2      | Outputs                                                 | 13        |
| 4.3      | L_TB2P_scCtrlBasicMotion                                | 14        |
| 4.4      | L_TB2P_scStatusBasicMotion                              | 16        |
| 4.5      | Parameter                                               | 18        |
| <b>5</b> | <b>State machine</b>                                    | <b>19</b> |
| <b>6</b> | <b>Description of the basic functions</b>               | <b>20</b> |
| 6.1      | Important notes on how to operate the technology module | 20        |
| 6.2      | Manual jog (jogging)                                    | 21        |
| 6.3      | Homing                                                  | 23        |
| 6.4      | QSP application                                         | 26        |
| 6.5      | Halt                                                    | 27        |
| 6.6      | Absolute positioning                                    | 28        |
| 6.7      | Relative positioning                                    | 30        |
| 6.8      | Continuous travel with a defined velocity               | 33        |
| 6.9      | Aborting and completing a positioning process           | 36        |
| 6.10     | Override for speed                                      | 38        |
|          | <b>Index</b>                                            | <b>40</b> |
|          | Your opinion is important to us                         | 41        |

## 1 About this document

This documentation ...

- contains detailed information on the functionalities of the "Basic Motion" 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)<br>Lenze controllers (product information, technical data)                                                                                                                                                                 |
| System manuals                        | Visualisation (system overview/sample topologies)                                                                                                                                                                                                                                           |
| Communication manuals<br>Online helps | Bus systems <ul style="list-style-type: none"><li>• Controller-based Automation EtherCAT®</li><li>• Controller-based Automation CANopen®</li><li>• Controller-based Automation PROFIBUS®</li><li>• Controller-based Automation PROFINET®</li></ul>                                          |
| Reference manuals<br>Online helps     | Lenze Controllers: <ul style="list-style-type: none"><li>• Controller 3200 C</li><li>• Controller c300</li><li>• Controller p300</li><li>• Controller p500</li></ul>                                                                                                                        |
| Software manuals<br>Online helps      | Lenze Engineering Tools: <ul style="list-style-type: none"><li>• »PLC Designer« (programming)</li><li>• »Engineer« (parameter setting, configuration, diagnostics)</li><li>• »VisiWinNET® Smart« (visualisation)</li><li>• »Backup &amp; Restore« (data backup, recovery, update)</li></ul> |

## More technical documentation with regard to Lenze components

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

| Planning / configuration / technical data                                         |                                                                                                                                                                                                                                                               |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <input type="checkbox"/>                                                          | <b>Product catalogues</b> <ul style="list-style-type: none"> <li>• Controller-based Automation</li> <li>• Controllers</li> <li>• Inverter Drives/Servo Drives</li> </ul>                                                                                      |
| Mounting and wiring                                                               |                                                                                                                                                                                                                                                               |
|  | <b>Mounting instructions</b> <ul style="list-style-type: none"> <li>• Controllers</li> <li>• Communication cards (MC-xxx)</li> <li>• I/O system 1000 (EPM-Sxxx)</li> <li>• Inverter Drives/Servo Drives</li> <li>• Communication modules</li> </ul>           |
| <input type="checkbox"/>                                                          | <b>Hardware manuals</b> <ul style="list-style-type: none"> <li>• Inverter Drives/Servo Drives</li> </ul>                                                                                                                                                      |
| Parameterisation / configuration / commissioning                                  |                                                                                                                                                                                                                                                               |
| <input type="checkbox"/>                                                          | <b>Online help/reference manuals</b> <ul style="list-style-type: none"> <li>• Controllers</li> <li>• Inverter Drives/Servo Drives</li> <li>• I/O system 1000 (EPM-Sxxx)</li> </ul>                                                                            |
| <input type="checkbox"/>                                                          | <b>Online help/communication manuals</b> <ul style="list-style-type: none"> <li>• Bus systems</li> <li>• Communication modules</li> </ul>                                                                                                                     |
| Sample applications and templates                                                 |                                                                                                                                                                                                                                                               |
| <input type="checkbox"/>                                                          | <b>Online help / software and reference manuals</b> <ul style="list-style-type: none"> <li>• i700 application sample</li> <li>• Application Samples 8400/9400</li> <li>• FAST Application Template Lenze/PackML</li> <li>• FAST technology modules</li> </ul> |

### Symbols:

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



### Tip!

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

[www.lenze.com](http://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                     |
|---------|---------|------|---------------------------------|
| 4.0     | 04/2021 | TD06 | Error corrections & supplements |
| 3.0     | 04/2020 | TD06 | Error corrections & supplements |
| 2.0     | 12/2019 | TD06 | Error corrections & supplements |
| 1.0     | 04/2019 | TD06 | First edition                   |

### 1.2 Conventions used

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

| Type of information | Highlighting                                                                           | Examples/notes                                                |
|---------------------|----------------------------------------------------------------------------------------|---------------------------------------------------------------|
| Numeric notation    |                                                                                        |                                                               |
| Decimal separator   | Point                                                                                  | The decimal point is always used.<br>For example: 1234.56     |
| Text                |                                                                                        |                                                               |
| Program name        | » «                                                                                    | »PLC Designer« ...                                            |
| Variable names      | <i>italics</i>                                                                         | By setting <i>bEnable</i> to TRUE...                          |
| Function blocks     | <b>Bold</b>                                                                            | The <b>L_MC1P_AxisBasicControl</b> function block ...         |
| Function libraries  |                                                                                        | The <b>L_TT1P_TechnologyModules</b> function library ...      |
| Source code         | Font<br>"Courier new"                                                                  | ...<br>dwNumerator := 1;<br>dwDenominator := 1;<br>...        |
| Symbols             |                                                                                        |                                                               |
| Page reference      |  5) | Reference to further information:<br>Page number in PDF file. |

#### Variable names

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 Layout of the safety instructions

### **DANGER!**

Indicates an extremely hazardous situation. Failure to comply with this instruction will result in severe irreparable injury and even death.

### **WARNING!**

Indicates an extremely hazardous situation. Failure to comply with this instruction may result in severe irreparable injury and even death.

### **CAUTION!**

Indicates a hazardous situation. Failure to comply with this instruction may result in slight to medium injury.

### **NOTICE**

Indicates a material hazard. Failure to comply with this instruction may result in material damage.

### Application notes

| Pictograph                                                                          | Signal word | Meaning                       |
|-------------------------------------------------------------------------------------|-------------|-------------------------------|
|  | <b>Tip!</b> | Useful tip for easy handling  |
|  |             | Reference to another document |

## 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 Overview of the technology module functions

| Functionality                                                        | Description                                                           |
|----------------------------------------------------------------------|-----------------------------------------------------------------------|
| <a href="#">Manual jog (jogging)</a> (📖 21)                          | Traverse axis in manual operation                                     |
| <a href="#">Homing</a> (📖 23)                                        | Reference axis                                                        |
| <a href="#">OSP application</a> (📖 26)                               | Stop movement via application quick stop deceleration ramp (AXIS_REF) |
| <a href="#">Halt</a> (📖 27)                                          | Stop movement via a halt deceleration ramp                            |
| <a href="#">Absolute positioning</a> (📖 28)                          | Move axis to a unique absolute position                               |
| <a href="#">Relative positioning</a> (📖 30)                          | Move axis to a target position relative to the current position       |
| <a href="#">Continuous travel with a defined velocity</a> (📖 33)     | Execute continuous travel                                             |
| <a href="#">Aborting and completing a positioning process</a> (📖 36) | End positioning                                                       |
| <a href="#">Override for speed</a> (📖 38)                            | Define overrides for speed                                            |

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

### 3.1 Important notes on how to operate the technology module

#### Start of the axes

The start-up of the axes after the controller is disabled depends on the type of movement commissioned.

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 gets automatically active when the controller is enabled as long as no higher-priority function is active. This movement type includes
  - [Handfahren \(Jogging\)](#) (📖 39)
  - [QSP Application](#) (📖 45)
- After the controller has been enabled, the movement type with edge-activated function must be activated with a FALSE↗TRUE edge. This movement type includes
  - [Referenzfahrt \(Homing\)](#) (📖 43)



#### Example of a movement type with level-activated function: Manual jog (jogging)

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

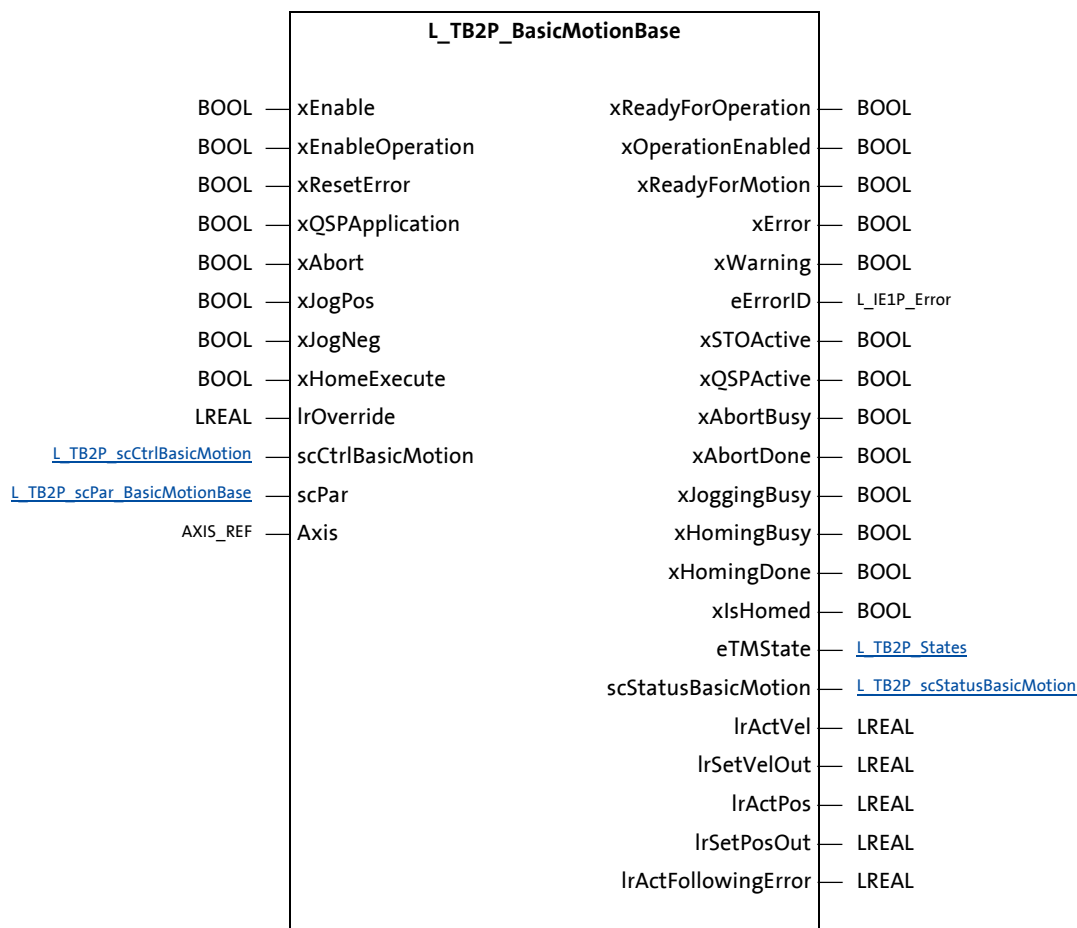


#### Example of a movement type with edge-activated function: Homing

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

## 4 L\_TB2P\_BasicMotionBase function block

The illustration shows that the inputs and outputs are part of the function block.



## 4.1

## 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 the "ready for operation" state. Controller enable is issued.                                                                                                                                                                                                                                                                                                 |
|                                                               |           | FALSE The axis is not ready for operation.                                                                                                                                                                                                                                                                                                                                                 |
| xResetError                                                   | BOOL      | The input is edge-controlled and evaluates the rising edge.                                                                                                                                                                                                                                                                                                                                |
|                                                               |           | FALSE Reset axis error or software error.                                                                                                                                                                                                                                                                                                                                                  |
|                                                               |           | TRUE The axis error or software error is reset.                                                                                                                                                                                                                                                                                                                                            |
| xQSPApplication                                               | BOOL      | The input is level-controlled and evaluates the HIGH level.                                                                                                                                                                                                                                                                                                                                |
|                                                               |           | TRUE The active movement is aborted and the axis is brought to a standstill via the <a href="#">QSP application</a> (□ 26) stop ramp. <ul style="list-style-type: none"> <li>The state changes to "Stop".</li> <li>The technology module remains in the "STOP" status as long as <i>xQSPApplication</i> remains TRUE.</li> </ul>                                                           |
| xAbort                                                        | BOOL      | TRUE The active movement is cancelled and the axis is brought to a standstill profile deceleration ( <i>scPar.IrDec</i> ). <ul style="list-style-type: none"> <li>The state changes to "Stop".</li> </ul> The deceleration ramp can be aborted by commissioning of a new movement (e.g. <i>xMoveVelExecute</i> )<br><a href="#">▶ Aborting and completing a positioning process</a> (□ 36) |
|                                                               |           | FALSE The active movement continues.                                                                                                                                                                                                                                                                                                                                                       |
| xJogPos                                                       | BOOL      | The input is level-controlled and evaluates the HIGH level.<br><a href="#">▶ Manual jog (jogging)</a> (□ 21)                                                                                                                                                                                                                                                                               |
|                                                               |           | TRUE Traverse axis in positive direction (manual jog).<br>If <i>xJogNeg</i> is also TRUE, the <i>eErrorID</i> output displays the "InvalidInputJogPosJogNeg" error message and the axis stops.                                                                                                                                                                                             |
| xJogNeg                                                       | BOOL      | The input is level-controlled and evaluates the HIGH level.<br><a href="#">▶ Manual jog (jogging)</a> (□ 21)                                                                                                                                                                                                                                                                               |
|                                                               |           | TRUE Traverse axis in negative direction (manual jog).<br>If <i>xJogPos</i> is also TRUE, the <i>eErrorID</i> output displays the "InvalidInputJogPosJogNeg" error message and the axis stops.                                                                                                                                                                                             |
| xHomeExecute                                                  | BOOL      | The input is edge-controlled and evaluates the rising edge.<br><a href="#">▶ Homing</a> (□ 23)                                                                                                                                                                                                                                                                                             |
|                                                               |           | FALSE Start homing.                                                                                                                                                                                                                                                                                                                                                                        |
|                                                               |           | TRUE The function is aborted via the <i>xHalt</i> input.                                                                                                                                                                                                                                                                                                                                   |
| IrOverride                                                    | LREAL     | With the <i>IrOverride</i> factor, the defined setpoint speed is multiplied (e.g. an override value of 0.5 generates half the setpoint speed) <ul style="list-style-type: none"> <li>Value range: 0 ... 1.0</li> <li>Initial value: 1.0</li> </ul> <b>Note:</b> Acceptance of the value is carried out irrespective of the status in <i>xContinuousUpdate</i> .                            |
|                                                               |           |                                                                                                                                                                                                                                                                                                                                                                                            |
| scCtrlBasicMotion<br><a href="#">L_TB2P_scCtrlBasicMotion</a> |           | Structure input for controlling basic motion functions                                                                                                                                                                                                                                                                                                                                     |
| scPar<br><a href="#">L_TB2P_scPar_BasicMotionBase</a>         |           | The parameter structure contains the parameters of the technology module.                                                                                                                                                                                                                                                                                                                  |
| Axis<br>AXIS_REF                                              |           | Reference to the axis                                                                                                                                                                                                                                                                                                                                                                      |

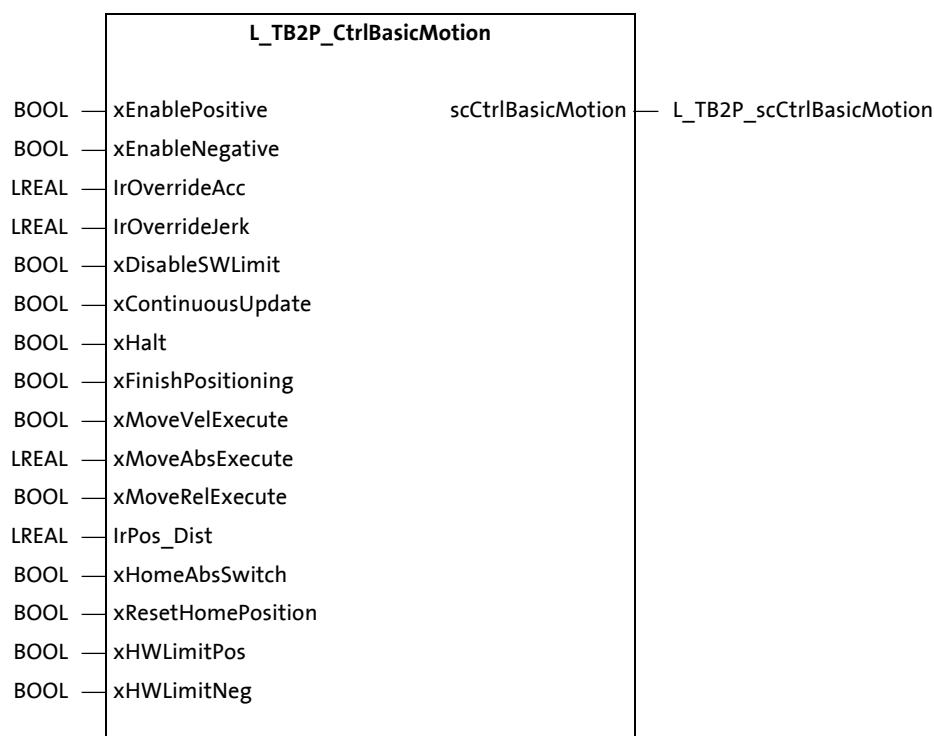
## 4.2

## Outputs

| Designator          | Data type                                  | Description                                                                                                                                                                                                      |
|---------------------|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| xReadyForOperation  | BOOL                                       | The axis is faultless.<br>The DC bus voltage is available.<br>"Safe Torque Off" (STO) is not active.                                                                                                             |
| xOperationEnabled   | BOOL                                       | The axis is enabled.<br>The brake is released.                                                                                                                                                                   |
| xReadyForMotion     | BOOL                                       | The axis is enabled and faultless.<br>The brake is released.                                                                                                                                                     |
| xError              | BOOL                                       | TRUE<br>There is an error in the technology module.                                                                                                                                                              |
| xWarning            | BOOL                                       | TRUE<br>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.<br><b>"FAST &amp; technology modules" reference manual:</b><br>Here you can find information on error or warning messages. |
| xSTOActive          | BOOL                                       | TRUE<br>"Safe Torque Off" (STO) was enabled and is active.<br>The axis is torque-free and the inverter is inhibited.                                                                                             |
| xQSPActive          | BOOL                                       | TRUE<br>The <i>xQSPApplication</i> input is set to TRUE.<br>The technology module is in the "STOP" state, the axis is brought to a standstill.<br>The output is reset as soon as the axis is at a standstill.    |
| xAbortBusy          | BOOL                                       | TRUE<br>The function activated is aborted by the <i>xAbsort</i> input.                                                                                                                                           |
| xAbortDone          | BOOL                                       | TRUE<br>The activated function is aborted by the <i>xAbsort</i> input.                                                                                                                                           |
| xJoggingBusy        | BOOL                                       | TRUE<br>The manual jog function is active.<br>The technology module is in the "JOGGING" state.                                                                                                                   |
| xHomingBusy         | BOOL                                       | TRUE<br>Homing is executed.<br>The technology module is in the "HOMING" state.                                                                                                                                   |
| xHomingDone         | BOOL                                       | TRUE<br>Homing has been completed successfully.                                                                                                                                                                  |
| xIsHomed            | BOOL                                       | TRUE<br>The axis has been referenced ("Reference known").<br><i>xIsHomed</i> may only be evaluated when the <i>xAxisAvailable</i> status is reported.                                                            |
| eTMState            | <a href="#">L_TB2P_States</a>              | Current state of the technology module                                                                                                                                                                           |
| scStatusBasicMotion | <a href="#">L_TB2P_scStatusBasicMotion</a> | Structure of the status data of the function block                                                                                                                                                               |
| IrActVel            | LREAL                                      | Current velocity<br>• Unit: units/s                                                                                                                                                                              |
| IrSetVelOut         | LREAL                                      | Setpoint speed<br>• Unit: units/s                                                                                                                                                                                |
| IrActPos            | LREAL                                      | Actual position<br>• Unit: units                                                                                                                                                                                 |
| IrSetPosOut         | LREAL                                      | Setpoint position<br>• Unit: units                                                                                                                                                                               |
| IrActFollowingError | LREAL                                      | Current following error<br>• Unit: units                                                                                                                                                                         |

## 4.3

## L\_TB2P\_scCtrlBasicMotion



| Designator         | Data type | Description                                                                                               |                                                                                                                                                                                                                                                                             |
|--------------------|-----------|-----------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| xEnablePositive    | BOOL      | TRUE                                                                                                      | Enable traversing to the positive direction.<br>• Initial value: TRUE                                                                                                                                                                                                       |
| xEnableNegative    | BOOL      | TRUE                                                                                                      | Enable traversing to the negative direction.<br>• Initial value: TRUE                                                                                                                                                                                                       |
| IrOverrideAcc      | LREAL     | Factor for the global acceleration override<br>• Value range: 0.1 ... 1.0<br>• Initial value: 1.0 (100 %) |                                                                                                                                                                                                                                                                             |
| IrOverrideJerk     | LREAL     | Factor for the global jerk override<br>• Value range: 0.1 ... 1.0<br>• Initial value: 1.0 (100 %)         |                                                                                                                                                                                                                                                                             |
| xDisableSWLimit    | BOOL      | TRUE                                                                                                      | Deactivate software limit positions.<br>This input overrides the setting configured in the axis (subindex 50) and deactivates the software limit position monitoring.                                                                                                       |
|                    |           | FALSE                                                                                                     | The setting of the axis is effective.                                                                                                                                                                                                                                       |
| xContinuousUpdate  | BOOL      | TRUE                                                                                                      | Profile parameters are checked cyclically for change and are accepted.<br>• Initial value: TRUE                                                                                                                                                                             |
| xHalt              | BOOL      | TRUE                                                                                                      | The active movement is aborted and the axis is brought to a standstill via the halt ramp.<br>• The state changes to "Stop".<br>The deceleration ramp can be aborted by commissioning of a new movement (e.g. <i>xMoveVelExecute</i> )<br>► <a href="#">Halt</a> (📖 27)      |
| xFinishPositioning | BOOL      | FALSE<br>TRUE                                                                                             | An interrupted positioning is continued.<br>• The positioning interrupted before via <i>xAbort</i> is completed.<br>• Distances of a relative positioning already covered are taken into account.<br>► <a href="#">Aborting and completing a positioning process</a> (📖 36) |

| Designator         | Data type | Description   |                                                                                                                                                                                                                                                                                                                                                            |
|--------------------|-----------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| xMoveVelExecute    | BOOL      | FALSE<br>TRUE | Start continuous travel at a defined speed (MoveVelocity).<br>▶ <a href="#">Continuous travel with a defined velocity</a> (□ 33)                                                                                                                                                                                                                           |
| xMoveAbsExecute    | BOOL      | FALSE<br>TRUE | Start absolute positioning (MoveAbsolute).<br>▶ <a href="#">Absolute positioning</a> (□ 28)                                                                                                                                                                                                                                                                |
| xMoveRelExecute    | BOOL      | FALSE<br>TRUE | Start relative positioning (MoveRelative).<br>▶ <a href="#">Relative positioning</a> (□ 30)                                                                                                                                                                                                                                                                |
| lrPos_Dist         | LREAL     |               | <b>Absolute positioning</b><br>Absolute target position relating to the zero position of the axis.<br>• Unit: units<br>• Initial value: 0                                                                                                                                                                                                                  |
|                    |           |               | <b>Relative positioning</b><br>Distance to be travelled relating to the set position at the starting time of the command.<br>• Unit: units<br>• Initial value: 0                                                                                                                                                                                           |
| xHomeAbsSwitch     | BOOL      | TRUE          | Connection for reference switch<br>For homing modes with a reference switch, connect this input to the digital signal which maps the state of the reference switch.                                                                                                                                                                                        |
| xResetHomePosition | BOOL      | TRUE          | Reset "Home position known" status information ( <i>xisHomed</i> output).                                                                                                                                                                                                                                                                                  |
| xHWLimitPos        | BOOL      |               | Positive hardware limit switch<br>Connect this input to the corresponding digital input that is connected to the limit switch.                                                                                                                                                                                                                             |
|                    |           | TRUE          | The positive hardware limit switch has been reached or approached.<br>• The <i>xHWLimitSwitchPos</i> structure output is also set to TRUE.<br>• The axis is brought to a standstill with the deceleration of the <a href="#">QSP application</a> (□ 26).<br>• The state changes to "ERROR" with the "HWLimitPos" error message ( <i>eErrorID</i> = 20500). |
| xHWLimitNeg        | BOOL      |               | Negative hardware limit switch<br>Connect this input to the corresponding digital input that is connected to the limit switch.                                                                                                                                                                                                                             |
|                    |           | TRUE          | The negative hardware limit switch has been reached or approached.<br>• The <i>xHWLimitSwitchNeg</i> structure output is also set to TRUE.<br>• The axis is brought to a standstill with the deceleration of the <a href="#">QSP application</a> (□ 26).<br>• The state changes to "ERROR" with the "HWLimitNeg" error message ( <i>eErrorID</i> = 20501). |

## 4 L\_TB2P\_BasicMotionBase function block

### 4.4 L\_TB2P\_scStatusBasicMotion

#### 4.4 L\_TB2P\_scStatusBasicMotion



| Designator                         | Data type | Description                                                                            |
|------------------------------------|-----------|----------------------------------------------------------------------------------------|
| xFuncBusy                          | BOOL      | TRUE Processing of the activated function running.<br>The DC bus voltage is available. |
| xFunDone                           | BOOL      | TRUE Execution of the activated function completed successfully.                       |
| xFunCommandAborted                 | BOOL      | TRUE The function activated has been aborted by another function block.                |
| xHaltBusy                          | BOOL      | TRUE The holding function is executed                                                  |
| xHaltDone                          | BOOL      | TRUE The holding function has been completed successfully.                             |
| eControlMode<br>L_MC1P_ControlMode |           | Current control mode of the axis                                                       |
| xHwLimitSwitchNeg                  | BOOL      | TRUE Negative hardware limit switch exceeded.                                          |
| xHwLimitSwitchPos                  | BOOL      | TRUE Positive hardware limit switch exceeded.                                          |
| xSwLimitEnabled                    | BOOL      | TRUE Monitoring of the software limit positions is activated.                          |

| Designator             | Data type         | Description                       |                                                                                                                                                                                                                                        |
|------------------------|-------------------|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| xSwLimitSwitchNeg      | BOOL              | TRUE                              | Negative software limit position exceeded.                                                                                                                                                                                             |
| xSwLimitSwitchPos      | BOOL              | TRUE                              | Positive software limit position exceeded.                                                                                                                                                                                             |
| eAxisType              | L_MC1P_DriveType  | Connected axis type               |                                                                                                                                                                                                                                        |
| eAxisState             | L_MC1P_AXIS_STATE | Current PLCOpen state of the axis |                                                                                                                                                                                                                                        |
| xAxisWarning           | BOOL              | TRUE                              | There is a warning in the axis data or in a connected function block.                                                                                                                                                                  |
| xAxisError             | BOOL              | TRUE                              | The output is TRUE in case of: <ul style="list-style-type: none"> <li>• Error in the axis data</li> <li>• Error of a connected function block</li> <li>• Error messages of the function block at the <i>eErrorID</i> output</li> </ul> |
| xDriveWarning          | BOOL              | TRUE                              | There is a warning in the inverter.                                                                                                                                                                                                    |
| xDriveError            | BOOL              | TRUE                              | There is an error in the inverter.                                                                                                                                                                                                     |
| dwDriveErrorID         | DWORD             | TRUE                              | If <i>xDriveError</i> = TRUE or <i>xDriveWarning</i> = TRUE:<br>Error number of the error which has occurred in the inverter.                                                                                                          |
| xConstantVelocity      | BOOL              | TRUE                              | The defined speed has been reached for the first time.                                                                                                                                                                                 |
| xAccelerating          | BOOL              | TRUE                              | The axis is accelerated.<br>(Increase in the speed value)                                                                                                                                                                              |
| xDecelerating          | BOOL              | TRUE                              | The axis is decelerated or braked.<br>(Decrease in the speed value)                                                                                                                                                                    |
| xDirectionPositive     | BOOL              | TRUE                              | The axis travels in positive direction.                                                                                                                                                                                                |
| xDirectionNegative     | BOOL              | TRUE                              | Axis travels in negative direction.                                                                                                                                                                                                    |
| xInPosition            | BOOL              | TRUE                              | The target position has been reached.                                                                                                                                                                                                  |
| xAxisAvailable         | BOOL              | TRUE                              | The axis can be used in the application, i.e. the initialization has been completed correctly.<br>Valid actual values and the communication with the lower-level drive are available.                                                  |
| xStandstillSetVelocity | BOOL              | TRUE                              | The setpoint speed of the axis is zero.                                                                                                                                                                                                |
| xStandstillActVelocity | BOOL              | TRUE                              | The actual speed is within the defined window ( <i>Window standstill</i> , subindex 132)                                                                                                                                               |

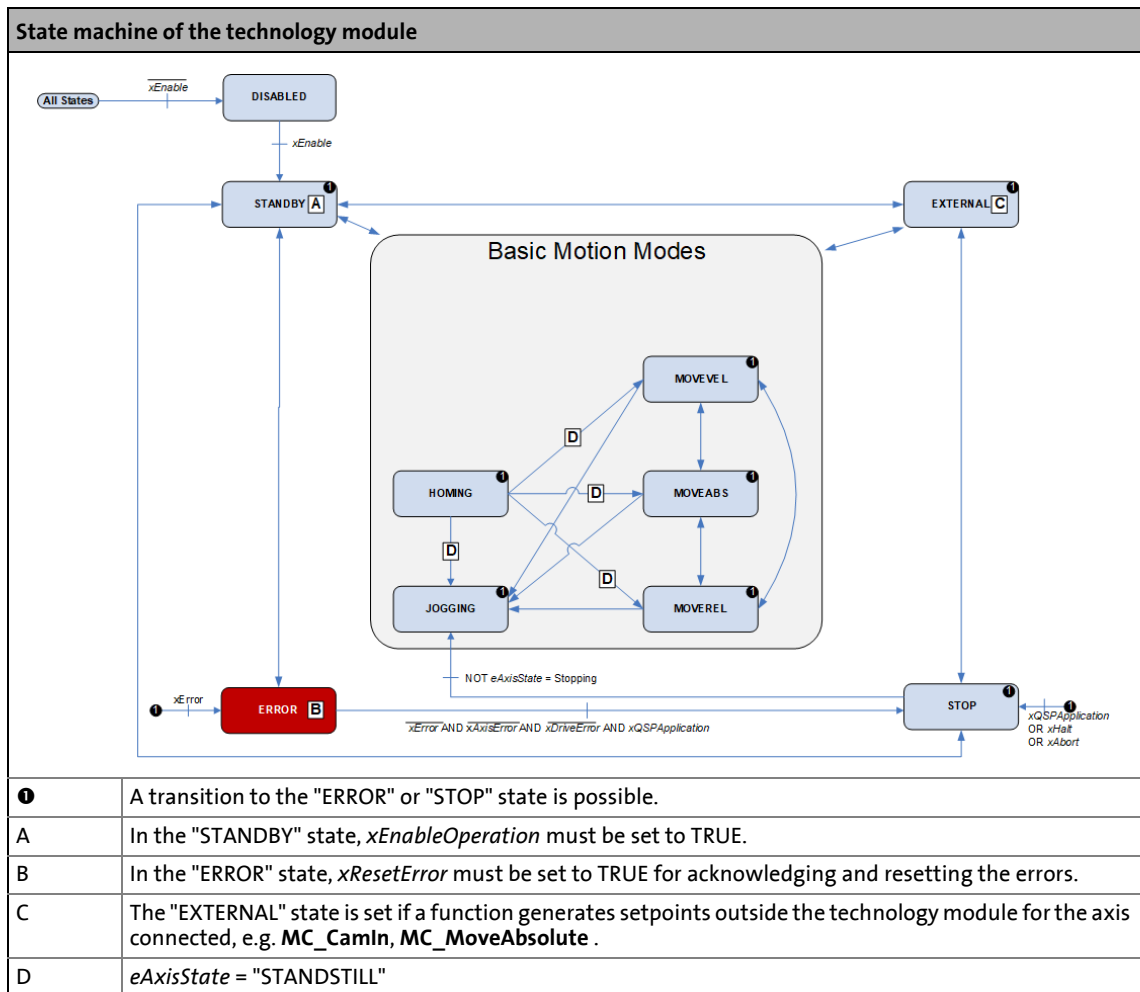
## 4.5

## Parameter

## L\_TB2P\_scPar\_BasicMotionBase

| Designator  | Data type      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IrVel       | LREAL          | Speed for movements (MoveAbs, MoveRel, MoveVel) that are commissioned via the <b>scCtrlBasicMotion</b> structure.<br><ul style="list-style-type: none"> <li>Unit: units/s</li> <li>Initial value: 3600</li> </ul>                                                                                                                                                                                                                                                                                           |
| IrAcc       | LREAL          | Acceleration for movements (MoveAbs, MoveRel, MoveVel) that are commissioned via the <b>scCtrlBasicMotion</b> structure.<br>Specification of the maximum velocity variation which is to be used for acceleration.<br><ul style="list-style-type: none"> <li>Unit: units/s<sup>2</sup></li> <li>Initial value: 36000</li> </ul>                                                                                                                                                                              |
| IrDec       | LREAL          | Deceleration for movements (MoveAbs, MoveRel, MoveVel) that are commissioned via the <b>scCtrlBasicMotion</b> structure.<br>Specification of the maximum speed variation which is to be used for deceleration to standstill.<br><ul style="list-style-type: none"> <li>Unit: units/s<sup>2</sup></li> <li>Initial value: 100</li> </ul>                                                                                                                                                                     |
| IrJerk      | LREAL          | Jerk for movements (MoveAbs, MoveRel, MoveVel) that are commissioned via the <b>scCtrlBasicMotion</b> structure.<br><ul style="list-style-type: none"> <li>Unit: units/s<sup>3</sup></li> <li>Initial value: 0</li> </ul>                                                                                                                                                                                                                                                                                   |
| eDirection  | MC_DIRECTION   | Traversing direction for movements (MoveAbs, MoveVel) that are instructed via the <b>scCtrlBasicMotion</b> structure:<br><ul style="list-style-type: none"> <li><i>mcCurrentDirection</i>: Maintain current direction (only in case of modulo axes)</li> <li><i>mcPositiveDirection</i>: Positive direction</li> <li><i>mcNegativeDirection</i>: Negative direction</li> <li><i>mcShortestWay</i>: Shortest way (only in case of modulo axes)</li> <li>Initial value: <i>mcPositiveDirection</i></li> </ul> |
| scHomeExtTP | MC_TRIGGER_REF | Transfer of an external touch probe event:<br><ul style="list-style-type: none"> <li>Only relevant for "external encoder" touch probe configuration.</li> <li>For describing the MC_TRIGGER_REF structure, see the <b>MC_TouchProbe</b> function block.</li> </ul>                                                                                                                                                                                                                                          |

## 5 State machine



## 6 Description of the basic functions

### Overview of the basic functions

- ▶ [Manual jog \(jogging\)](#) (📖 21)
- ▶ [Homing](#) (📖 23)
- ▶ [QSP application](#) (📖 26)
- ▶ [Halt](#) (📖 27)
- ▶ [Absolute positioning](#) (📖 28)
- ▶ [Relative positioning](#) (📖 30)
- ▶ [Continuous travel with a defined velocity](#) (📖 33)
- ▶ [Aborting and completing a positioning process](#) (📖 36)
- ▶ [Override for speed](#) (📖 38)

### 6.1 Important notes on how to operate the technology module

#### Controlled start of the axes

Motion commands that are set in the inhibited axis state ( $xAxisEnabled = FALSE$ ) after enable ( $xRegulatorOn = TRUE$ ) must be activated again by a  $FALSE \rightarrow TRUE$  edge.

In this way it is prevented that the drive starts in an uncontrolled manner after controller enable.



**Example** [Manual jog \(jogging\)](#) (📖 21):

1. In the inhibited axis state ( $xOperationEnabled = FALSE$ ),  $xJogPos$  is set to TRUE.
  - $xEnableOperation = FALSE$  (axis is inhibited.)  
==> "READY" state ( $xOperationEnabled = FALSE$ )
  - $xJogPos = TRUE$  (manual jog is to be executed.)
2. Enable axis.
  - $xEnableOperation = TRUE$   
==> "READY" state ( $xOperationEnabled = TRUE$ )
3. Execute manual jog.
  - $xJogPos = FALSE \rightarrow TRUE$   
==> "JOGPOS" state

## 6.2 Manual jog (jogging)

### Prerequisite

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

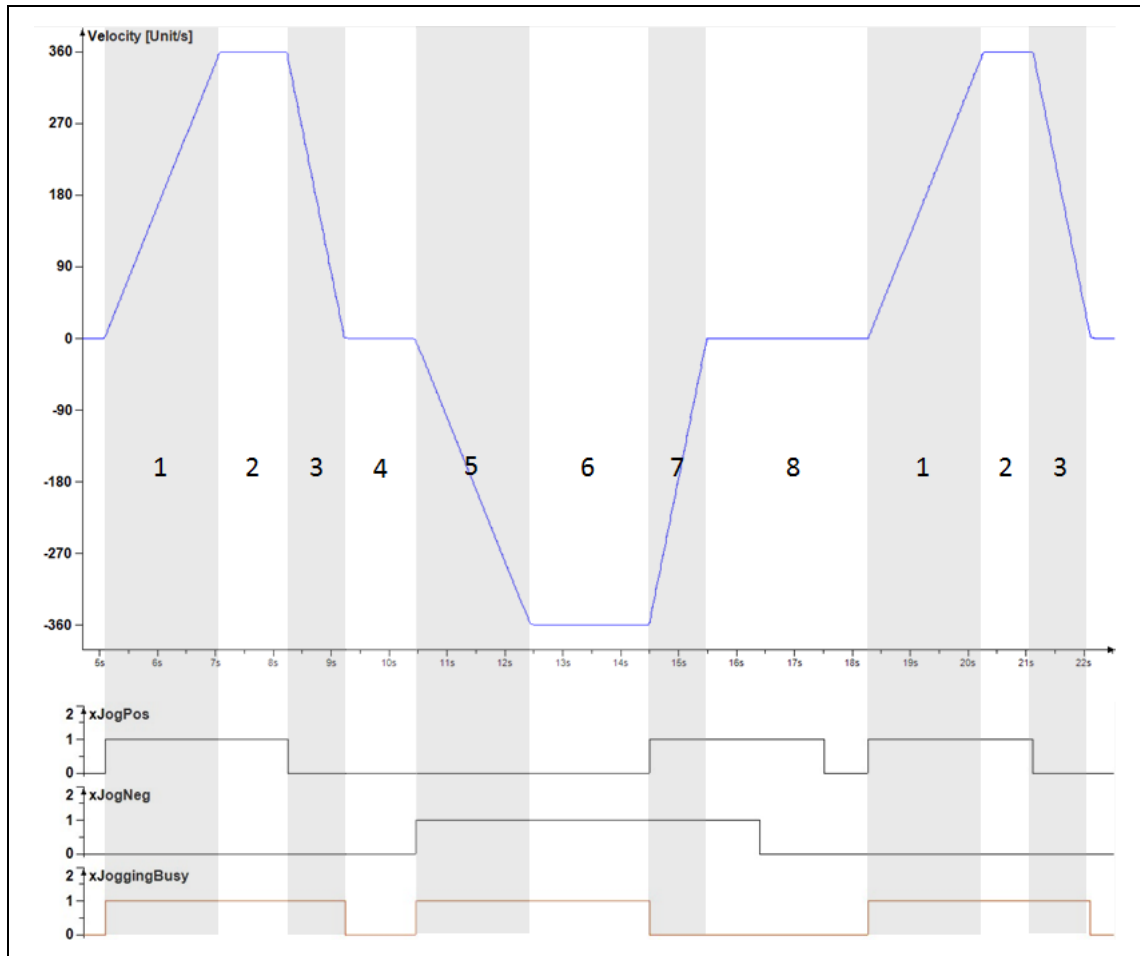
### Execution

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 *xJogNegPos* = 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.

## 6.3 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.

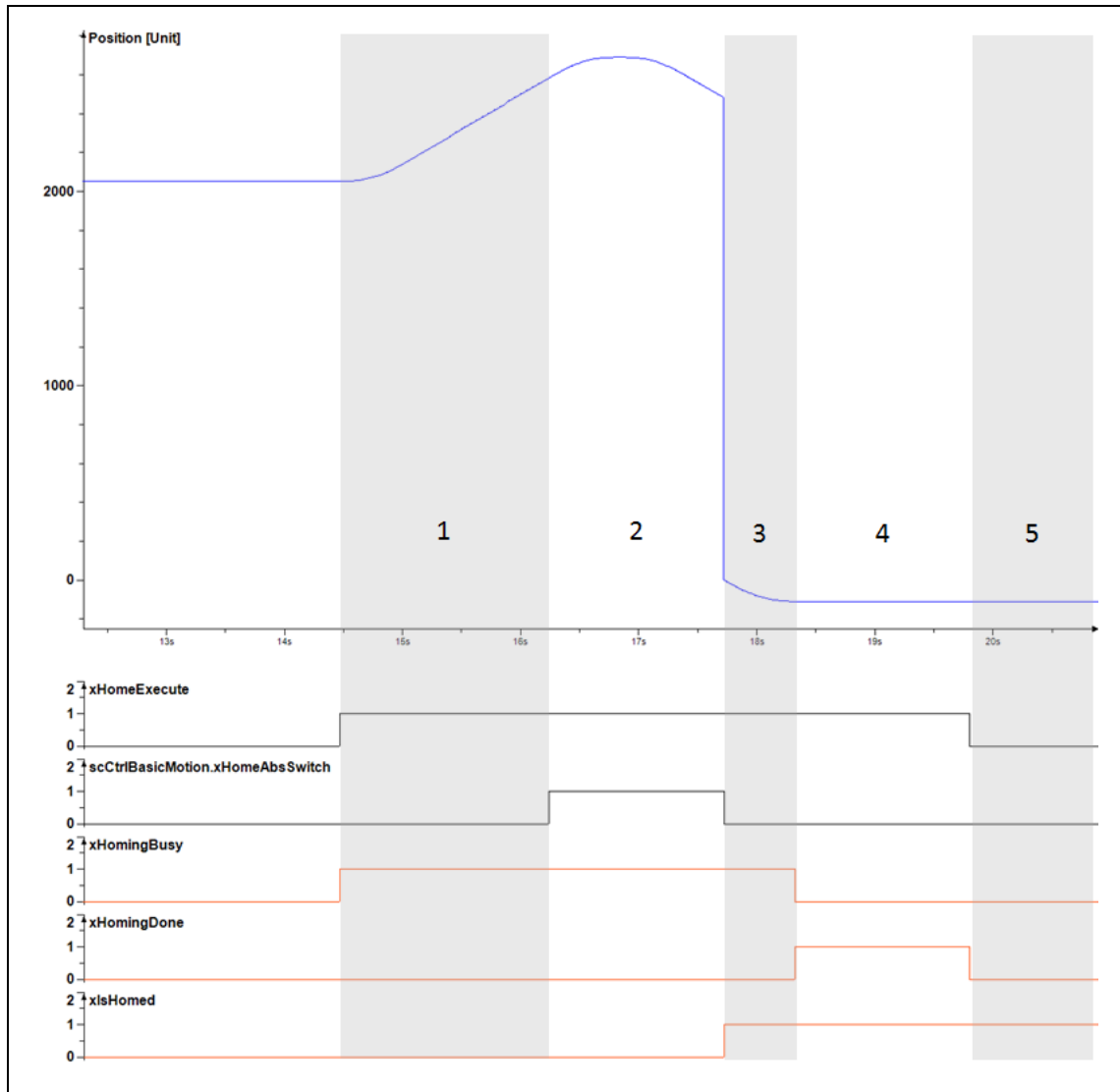
If *xHomeExecute* = FALSE, the state is changed to "STANDBY" and *xHomingDone* is set = FALSE.

### 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



[6-1] 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.

6.4 QSP application

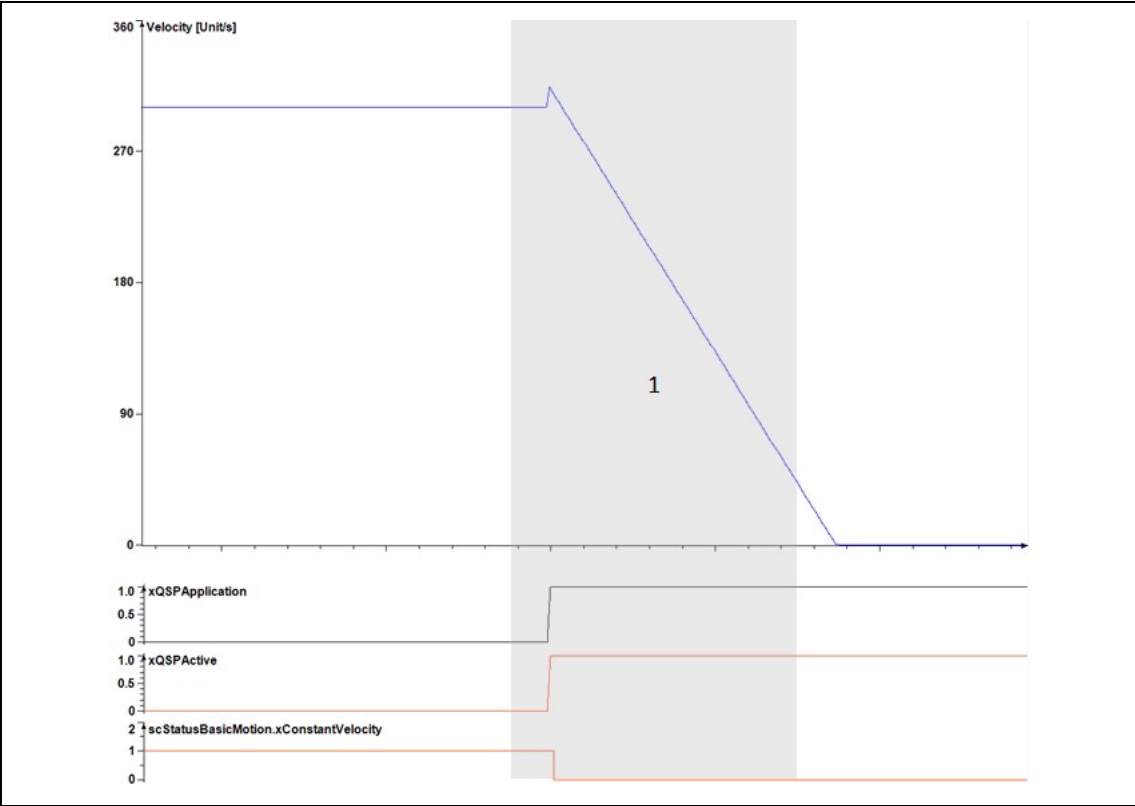
Prerequisite

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

Execution

With a rising edge (FALSE→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→TRUE       | FALSE→TRUE | TRUE→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.

## 6.5 Halt

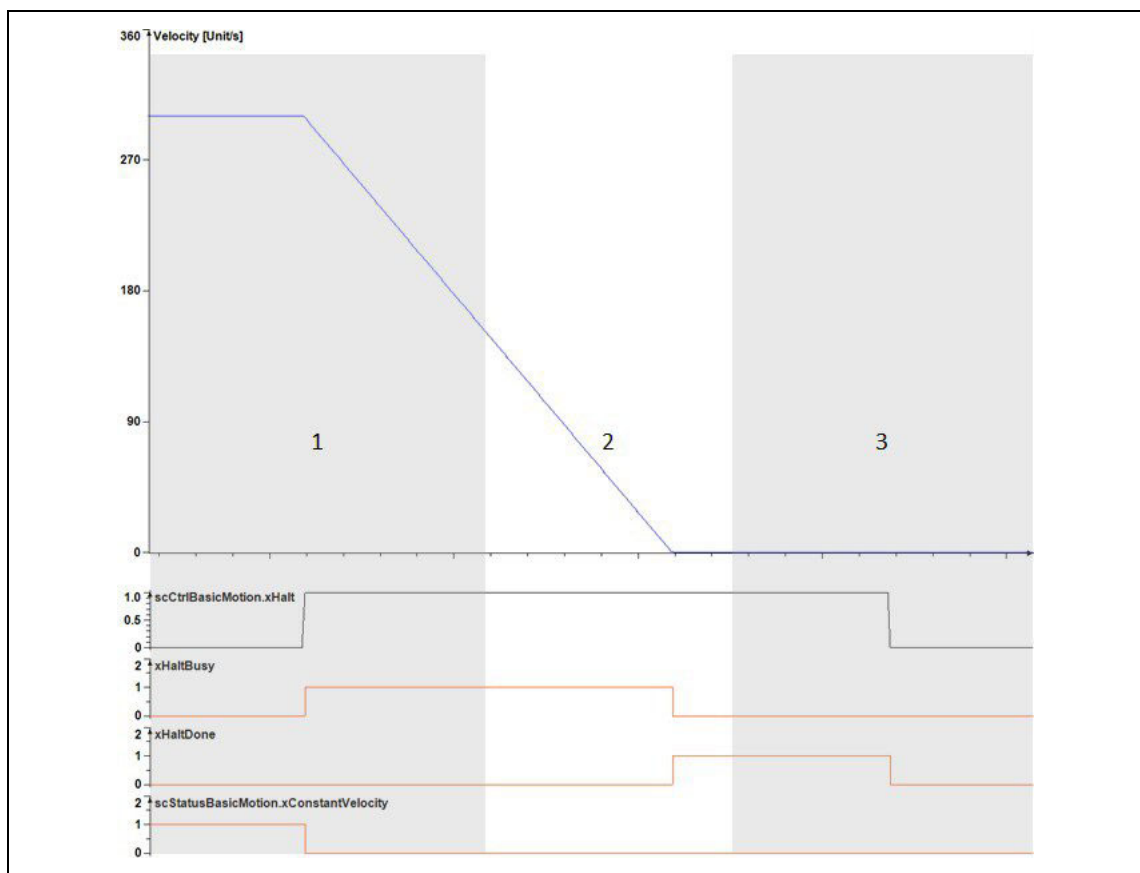
### Prerequisite

- 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

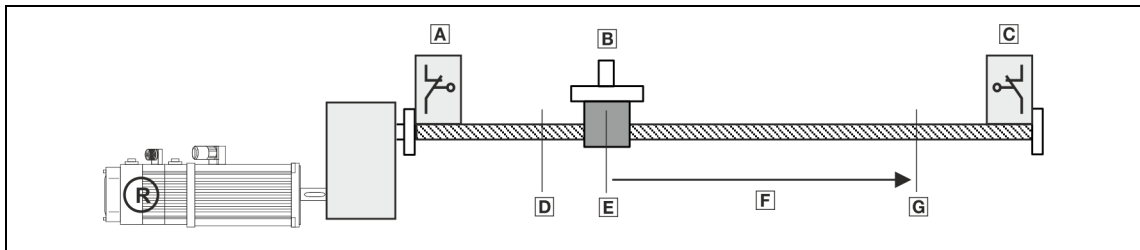


| No | xHalt                    | xHaltBusy                | xHaltDone                | xConstant Velocity       | eTMState                   | Info                                                                                                         |
|----|--------------------------|--------------------------|--------------------------|----------------------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

### Parameters to be set

The parameter for the halt can be found in the *AXIS\_REF* in subindex 186 and 187.

## 6.6 Absolute positioning



[6-2] 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>IrPos_Dist</i> input) |

With absolute positioning, the target position in the *IrPos\_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 *IrSetPos\_Dist* input is started on the basis of the parameters for the motion profile (see below). The technology module changes to the "MOVEABS" state.

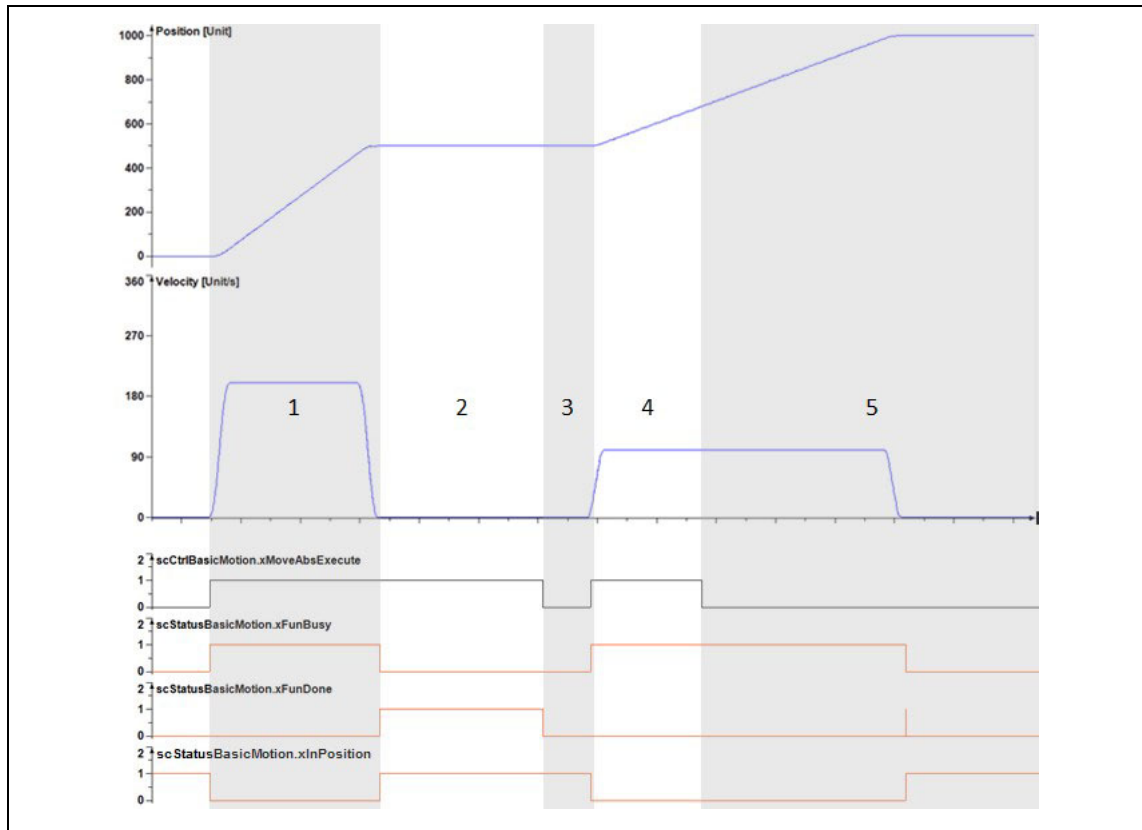
In order to reach the target position, the axis is brought to a standstill via the *IrDec* parameter. The axis reaches a standstill exactly in the target position.

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

During the travel process, the absolute target position in the *IrPos\_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



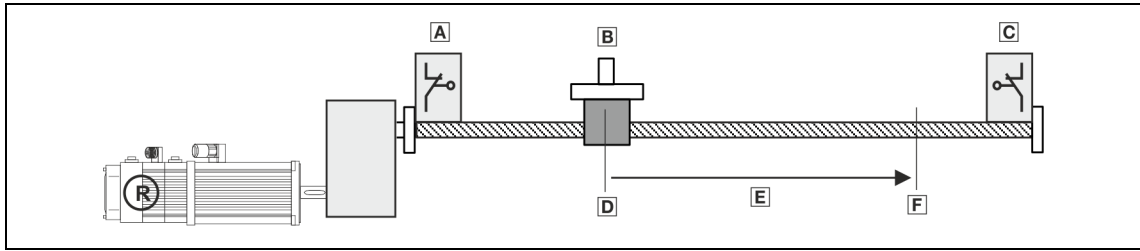
| No | xMoveAbsExecute | xFuncBusy    | xFuncDone            | xInPosition  | eTMState          | Info                                                                                                                                                                                       |
|----|-----------------|--------------|----------------------|--------------|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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/s<sup>2</sup>]
- *IrDcc*, initial value: 36000 [units/s<sup>2</sup>]
- *IrJerk*, initial value: 0 [units/s<sup>3</sup>]

Further parameters for absolute positioning can be found in the *AXIS\_REF* in subindex 130 ... 137.

## 6.7 Relative positioning



[6-3] 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.

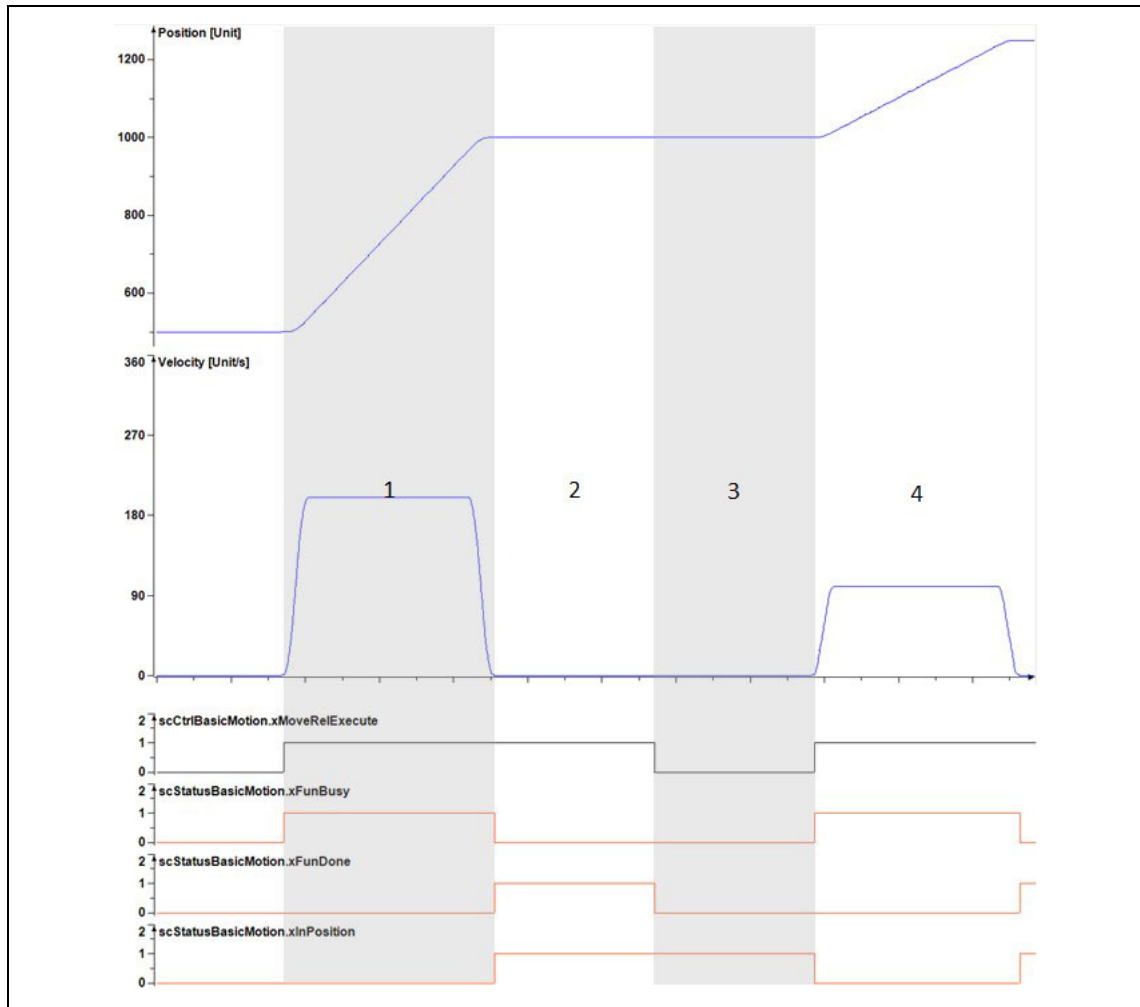
In order to reach the target position, the axis is brought to a standstill via the *lrDec* parameter. The axis reaches a standstill exactly in the target position.

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 | <code>xMoveRelExecute</code> | <code>xFuncBusy</code>     | <code>xFunDone</code>     | <code>xInPosition</code>  | <code>eTMState</code>             | Info                                                                                                                                                                                            |
|----|------------------------------|----------------------------|---------------------------|---------------------------|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | FALSE ↗<br>TRUE              | FALSE ↗<br>TRUE            | FALSE                     | FALSE                     | STANDBY →<br>MOVEREL              | The axis starts the positioning run to the relative target position 500 units.                                                                                                                  |
| 2  | TRUE                         | TRUE ↘<br>FALSE            | FALSE ↗<br>TRUE           | FALSE ↗<br>TRUE           | MOVEREL →<br>STANDBY              | As soon as the relative target position has been reached, <code>xInPosition</code> and <code>xFunDone</code> are set to TRUE.                                                                   |
| 3  | FALSE ↗<br>TRUE              | FALSE                      | TRUE ↘<br>FALSE           | TRUE                      | STANDBY                           | When <code>xMoveRelExecute</code> is set = FALSE, <code>xFunDone</code> changes from TRUE ↘ FALSE.                                                                                              |
| 4  | FALSE ↗<br>TRUE              | FALSE ↗<br>TRUE ↘<br>FALSE | TRUE ↘<br>FALSE ↗<br>TRUE | TRUE ↘<br>FALSE ↗<br>TRUE | STANDBY →<br>MOVEREL →<br>STANDBY | The axis starts the positioning run to the relative target position 250 units. After reaching the relative target position, <code>xInPosition</code> and <code>xFunDone</code> are set to TRUE. |

**Parameters to be set**

- *IrVel*, initial value: 3600 [units/s]
- *IrAcc*, initial value: 36000 [units/s<sup>2</sup>]
- *IrDcc*, initial value: 36000 [units/s<sup>2</sup>]
- *IrJerk*, initial value: 0 [units/s<sup>3</sup>]

Further parameters for relative positioning can be found in the AXIS\_REF in subindex 130 ... 137.

## 6.8 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:

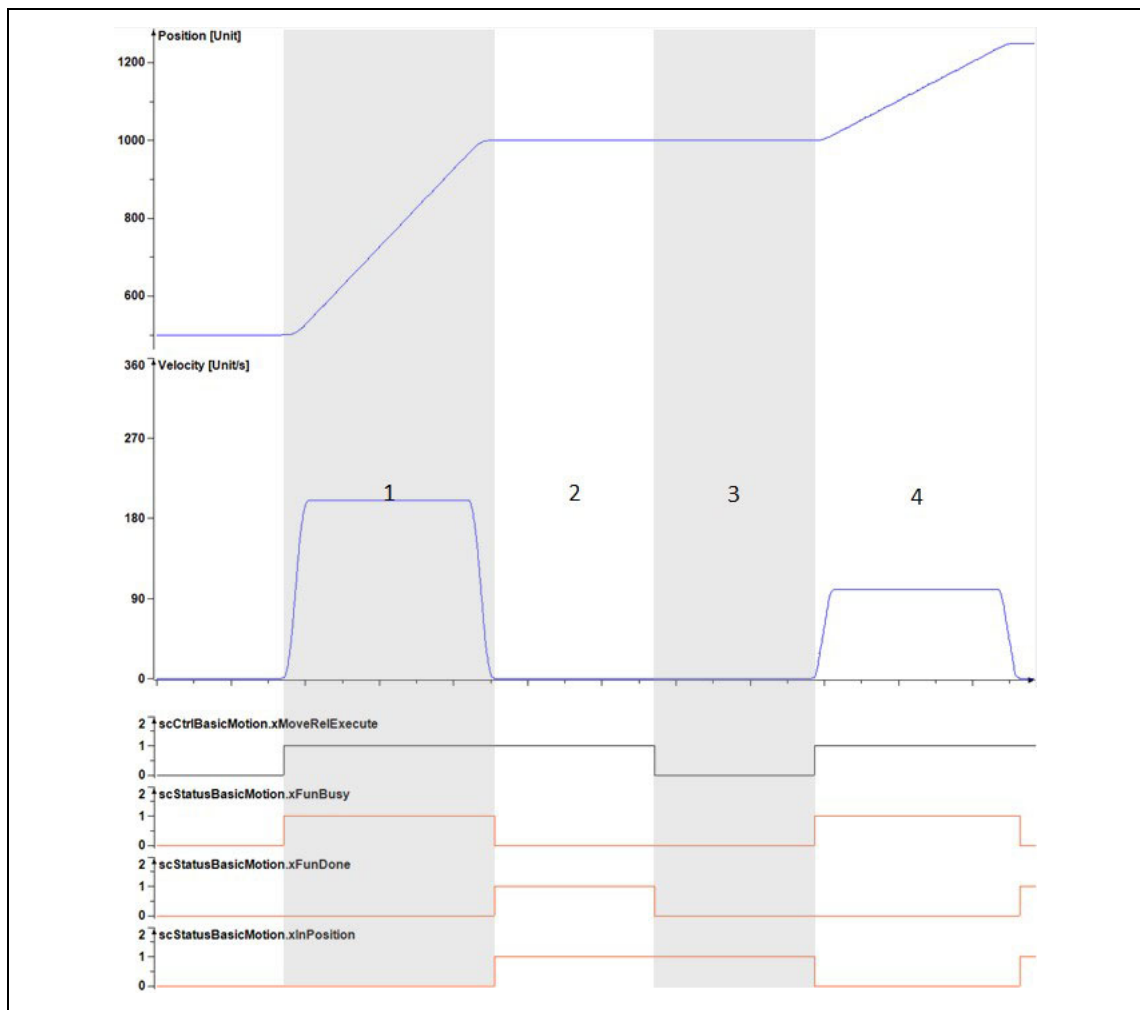
- Input *xDisableSWLimit* = FALSE
- Output *xSwLimitEnabled* = TRUE
- Output *xSwLimitSwitchPos/Neg* = TRUE
- Output *xHwLimitSwitchPos/Neg* = TRUE

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 | xMoveVelExecute            | xAbort          | xFuncBusy       | xFuncDone                 | xConstantVelocity | eTMState             | Info                                                                                                                                                                |
|----|----------------------------|-----------------|-----------------|---------------------------|-------------------|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | FALSE ↗<br>TRUE ↘<br>FALSE | FALSE           | FALSE ↗<br>TRUE | FALSE                     | FALSE             | STANDBY →<br>MOVEVEL | The axis accelerates to the target speed of 200 units/s.                                                                                                            |
| 2  | FALSE                      | FALSE           | TRUE            | FALSE ↗<br>TRUE           | FALSE ↗<br>TRUE   | MOVEVEL              | After reaching the target speed, xFuncDone changes from FALSE ↗ TRUE and xConstantVelocity changes from FALSE ↗ TRUE.                                               |
| 3  | FALSE                      | FALSE ↗<br>TRUE | TRUE ↘<br>FALSE | TRUE ↘<br>FALSE ↗<br>TRUE | TRUE ↘<br>FALSE   | MOVEVEL →<br>STANDBY | The speed run is terminated by xAbort = TRUE and brought to a standstill with the profile deceleration. When standstill has been reached, xFuncBusy is set = FALSE. |

**Parameters to be set**

- *IrVel*, initial value: 3600 [units/s]
- *IrAcc*, initial value: 36000 [units/s<sup>2</sup>]
- *IrDcc*, initial value: 36000 [units/s<sup>2</sup>]
- *IrJerk*, initial value: 0 [units/s<sup>3</sup>]

Further parameters for continuous travel can be found in the AXIS\_REF in subindex 50 ... 53 and subindex 104 ... 105.

## 6.9 Aborting and completing a positioning process

### Prerequisite

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

### Execution

An active positioning process is aborted with a rising edge ( $FALSE \rightarrow TRUE$ ) at the  $xAbort$  input.

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

- Exception: TM SpeedControl and TM TablePositioning**

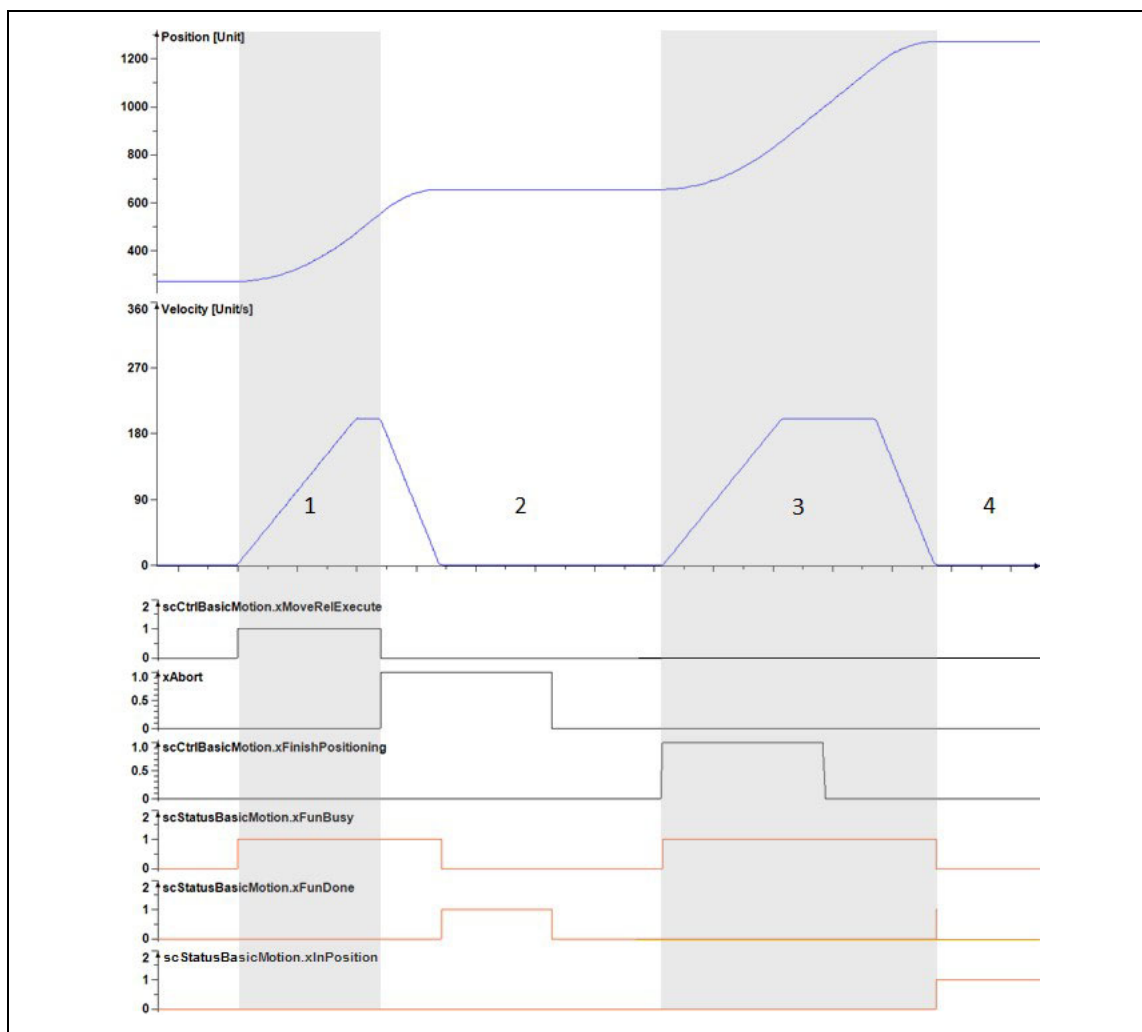
Both technology modules provide several profile data sets, one of which can be active.

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 ( $FALSE \rightarrow TRUE$ ) at the  $xFinishPositioning$  input.

### Travel diagram



## 6.9 Aborting and completing a positioning process

---

| No | xMoveRel<br>Execute | xAbort                   | xFinish<br>Positioning   | xInPosition             | eTMState                        | Info                                                                                                                                                                                       |
|----|---------------------|--------------------------|--------------------------|-------------------------|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | FALSE ↗<br>TRUE     | FALSE                    | FALSE                    | FALSE                   | STANDBY→<br>MOVEREL             | The axis starts the positioning process with the relative target position 1000 units.                                                                                                      |
| 2  | TRUE↘<br>FALSE      | TRUE↘<br>FALSE           | FALSE                    | FALSE                   | MOVEREL→<br>STOP→<br>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 ↗<br>TRUE          | FALSE↗<br>TRUE↘<br>FALSE | FALSE                   | STANDBY→<br>MOVEREL             | With <i>xFinishPositioning</i> = TRUE, the positioning process is continued.                                                                                                               |
| 4  | FALSE               | FALSE                    | FALSE                    | FALSE ↗<br>TRUE         | MOVEREL→<br>STANDBY             | As soon as the relative target position has been reached, <i>xInPosition</i> and <i>xFunDone</i> are set to TRUE.                                                                          |
|    | FALSE ↗<br>TRUE     | FALSE↗<br>TRUE↘<br>FALSE | TRUE↘<br>FALSE↗<br>TRUE  | TRUE↘<br>FALSE↗<br>TRUE | STANDBY→<br>MOVEREL→<br>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. |

## 6.10 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:

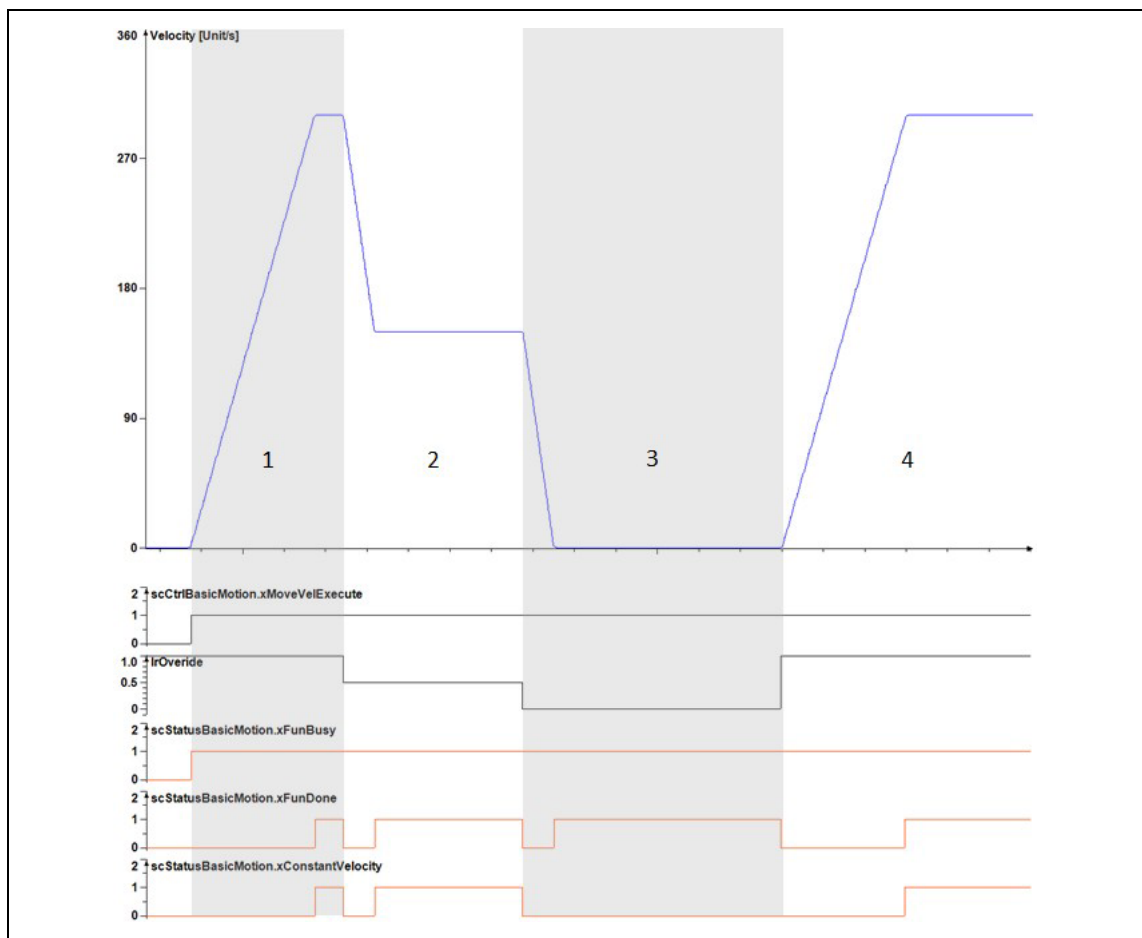
- *IrOverride*, initial value: 1.0
- *scCtrlBasicMotion.IrOverrideAcc*, initial value: 1.0 (100 %)
- *scCtrlBasicMotion.IrOverrideJerk*, initial value: 1.0 (100 %)

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.



## 6.10 Override for speed

---

| No | xMoveVel<br>Execute | IrOverride | xFuncBusy       | xFunDone                | xConstant<br>Velocity   | eTMState            | Info                                                                                                             |
|----|---------------------|------------|-----------------|-------------------------|-------------------------|---------------------|------------------------------------------------------------------------------------------------------------------|
| 1  | FALSE ↗<br>TRUE     | 1          | FALSE ↗<br>TRUE | FALSE ↗<br>TRUE         | FALSE ↗<br>TRUE         | STANDBY→<br>MOVEVEL | The axis accelerates to the target speed of 300 units/s.                                                         |
| 2  | TRUE                | 0.5        | TRUE            | TRUE↘<br>FALSE↗<br>TRUE | TRUE↘<br>FALSE↗<br>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↘<br>FALSE↗<br>TRUE | TRUE↘<br>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↘<br>FALSE↗<br>TRUE | FALSE ↗<br>TRUE         | MOVEVEL             | The override value is set to 1 and the axis accelerates to 300 unit/s with the profile deceleration              |

# Index

---

## A

Aborting a positioning process [36](#)  
Absolute positioning [28](#)  
Application notes [6](#)

## B

Basic functions of the technology modules [20](#)

## C

Competing a positioning process [36](#)  
Continuous travel with a defined velocity [33](#)  
Controlled start of the axes [20](#)  
Conventions used [5](#)

## D

Document history [5](#)

## E

E-mail to Lenze [41](#)

## F

Feedback to Lenze [41](#)

## H

Halt, [27](#)  
Homing [23](#)

## I

inputs [12](#)

## L

L\_TB2P\_BasicMotionBase [11](#)  
L\_TB2P\_BasicMotionBase function block [11](#)  
L\_TB2P\_scCtrlBasicMotion [14](#)  
L\_TB2P\_scPar\_BasicMotionBase [18](#)  
L\_TB2P\_scStatusBasicMotion [16](#)  
Layout of the safety instructions [6](#)

## M

Manual jog (jogging) [21](#)

## N

Notes on how to operate the technology module [10](#), [20](#)

## O

Outputs [13](#)  
Override for speed [38](#)

## P

Parameter structure L\_TB2P\_scPar\_BasicMotionBase [18](#)

## Q

QSP application [26](#)

## R

Relative positioning [30](#)

## S

Safety instructions [7](#)  
Speed override [38](#)  
Start of the axes [10](#), [20](#)  
State machine [19](#)  
States [19](#)

## T

Target group [4](#)

## V

Variable names [5](#)



## **Your opinion is important to us**

These instructions were created to the best of our knowledge and belief to give you the best possible support for handling our product.

If you have suggestions for improvement, please e-mail us to:

[feedback-docu@lenze.com](mailto:feedback-docu@lenze.com)

Thank you for your support.

*Your Lenze documentation team*

