FAST Application Software

Technology module

Pick & Place _____

Reference Manual EN

13531730

Lenze

Contents

1 About this documentation

This documentation ...

- contains detailed information on the functionalities of the "Pick & Place" 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 Controller (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« (parameter setting, configuration, diagnostics) • »VisiWinNET® Smart« (visualisation) • »Backup & Restore« (data backup, recovery, update)

More technical documentation for 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	
☐	**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
Parameter setting / 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 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.

1.1 Document history

Version			Description
4.2	05/2017	TD17	• Content structure has been changed. • General revisions
4.1	04/2016	TD17	General revisions
4.0	10/2015	TD17	• Corrections and additions • New: xInPosition output • New: Parameters xPosInWindow, lrPosInWindow, lrTimePosInWindow • Content structure has been changed.
3.0	05/2015	TD17	• General revisions • New: parameters for orientation axes A, B, C • New: Information on the "High" technology module version
2.0	01/2015	TD17	• General editorial revision • Modularisation of the contents for the »PLC Designer« online help
1.0	04/2014	TD00	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
Spelling of numbers		
Decimal separator	Point	The decimal point is always used. For example: 1234.56
Text		
Program name	» «	»PLC Designer« ...
Variable names	*italics*	By setting *bEnable* 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;`
		`...`

Icons		
Page reference	(📖 6)	Reference to further information: 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.

1.3 Definition of the notes used

The following signal words and symbols are used in this documentation to indicate dangers and important information:

Safety instructions

Layout of the safety instructions:

Pictograph and signal word!

(characterise the type and severity of danger)

Note

(describes the danger and gives information about how to prevent dangerous situations)

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 personal injury 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 personal injury if the corresponding measures are not taken.
	**Stop!**	**Danger of property damage** Reference to a possible danger that may result in property damage if the corresponding measures are not taken.

Application notes

Pictograph	Signal word	Meaning
	**Note!**	Important note to ensure trouble-free operation
	**Tip!**	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.

Stop!

Damage or destruction of machine parts

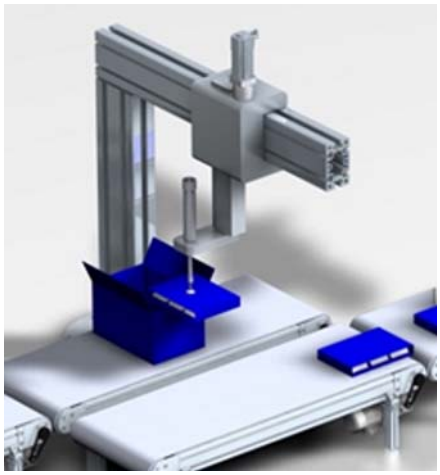
Damage or destruction of machine parts can be caused by ...

- 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 Functional description of "Pick & Place"



[3-1] Typical mechanics of the technology module

"Pick & Place" mechanics/kinematics are utilised in various industrial sectors. Typical applications can for instance be found in the field of packaging technology. Here, workpieces are picked up with a gripper and are placed to other positions.

In addition to simple point-to-point positioning processes, it is also possible to execute whole paths with several interpolation points. This, for example, serves to bypass obstacles.

The Technology module "Pick & Place" enables easy programming of "Pick & Place" profiles with up to 4 axes.

- In the "Base" version, simple travel profiles can be executed. By means of only a few parameters, a target point can be approached from the current actual position of the kinematics. In addition to the rounding parameter, only the starting height and final height have to be specified. The movement of an auxiliary axis (Aux1) is always carried out from the start of the first rounding process to the end of the second rounding process. There is only one parameter for the speed and acceleration of the path over the whole travel profile in each case.
- The "State" version provides an extended function range of the "Base" version:
  The travel profile can be written to with 10 interpolation points. The speed, acceleration and rounding can be set individually for each interpolation point. In addition, the movement of the (Aux1) auxiliary axis can be distributed over several interpolation points in the space. Furthermore it is possible to set breakpoints, for instance to wait for enable.
- The "High" version provides an extended functional range of the "State" version:
  Several profiles consisting of a maximum of 10 interpolation points can be connected and executed in succession without a stop. A path can be implemented from an infinite number of interpolation points and with a variable target.

▸ Overview of the functions (📖 11)

3.1 Overview of the functions

In addition to the basic functions for operating the **L_MC4P_AxesGroupBasicControl** function block and the **holding function**, the technology module offers the following functionalities which are assigned to the "Base", "State", and "High" version:

Functionality	Versions		
	Base	State	High
Manual jog (jogging) (📖 36)	●	●	●
Homing (📖 38)	●	●	●
Specifying a travel profile (📖 39)	●	●	●
Calculation of the profile points with lrBlendingRadius = 0 (📖 40)	●	●	●
Calculation of the profile points with lrBlendingRadius > 0 (📖 42)	●	●	●
Limitations (maximum values) for the axes on the travel profile (📖 44)	●	●	●
Starting the travel profile (xExecutePickAndPlace) (📖 46)	●	●	●
Stop/holding function (xPathStop, xStopALL, xPathHalt) (📖 47)	●	●	●
Interrupting the travel profile/path (xPathInterrupt) (📖 48)	●	●	●
Gripper control (📖 49)	●	●	●
Speed override (📖 50)	●	●	●
Defining a travel profile with up to 10 interpolation points (📖 51)		●	●
Movement of the orientation axes and the auxiliary axis across several interpolation points (📖 53)		●	●
Defining optional breakpoints on the travel profile (📖 54)		●	●
Adding more travel profiles (📖 55)			●

»PLC Designer« Online help

Here you'll find some detailed information with regard to the **L_MC4P_AxesGroupBasicControl** function block and the **holding function**.

3.2 Important notes on how to operate the technology module

Setting of the operating mode

The operating mode for the real axes A1 ... A6 has to be set to "cyclically synchronous position" (csp) because the axes are lead via the master position value.

Controlled start of the axes

Motion commands that are set in the inhibited axis state (*xAxesEnabled* = FALSE) after enable (*xRegulatorOnALL* = TRUE) must be activated again by a FALSE↗TRUE edge.

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

Example Manual jog (jogging) (📖 36)**:**

1. In the inhibited axis state (*xAxesEnabled* = FALSE), *xJogPos* is set to TRUE.
   - *xRegulatorOnALL* = FALSE (axis is inhibited.)
     ==> "READY" state (*xAxesEnabled* = FALSE)
   - Select the axis for the manual jog function via the *eSelectAxis* input.
   - *xJogPos* = TRUE (manual jog is to be executed.)
2. Enable axis.
   - *xRegulatorOnALL* = TRUE
     ==> "READY" state (*xAxesEnabled* = TRUE)
3. Execute manual jog.
   - *xJogPos* = FALSE↗TRUE
     ==> "JOGPOS" state

3.3 Interconnection of the technology module with the axes group

The "Pick & Place" technology module has no direct axis connections. The axes are transferred as group via the *AxesGroup* input of the technology module. An axes group is a combination of axes which can additionally contain kinematic transformations.

Communication between the technology module and the axes group is established via a direct connection.

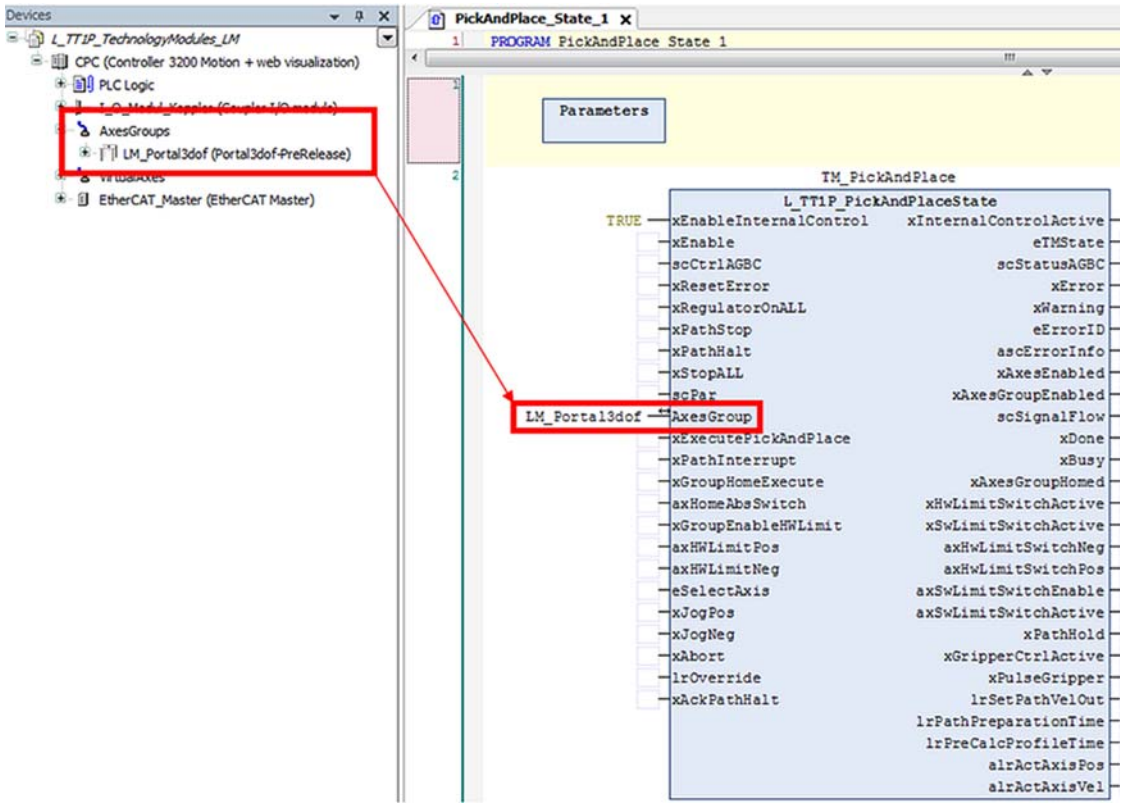
The axes group must be inserted in the device tree. Via the designation, the axes group is connected to the *AxesGroup* input of the technology module.

The setpoint generation for the axes and control of the functions (such as manual jog) are executed in the technology module. The technology module reaches each axis via the axes group. It is the axes group's task to cyclically calculate the kinematic reference between the real and virtual axes.

Example of the Delta3 transformation

An LM_Delta3dof axes group manages the real axes A1 ... A6 and virtual axes X, Y and Z. In a movement of the real axes A1 ... A6, the virtual axes X, Y and Z are controlled along with the direct kinematics. When a movement of the virtual axes X, Y and Z is carried out, the real axes A1 ... A6 are automatically controlled along with the inverse kinematics. Thus, the reference between the real axes and the virtual axes (coordinates of the "Tool Center Point", tool zero point) is always given.

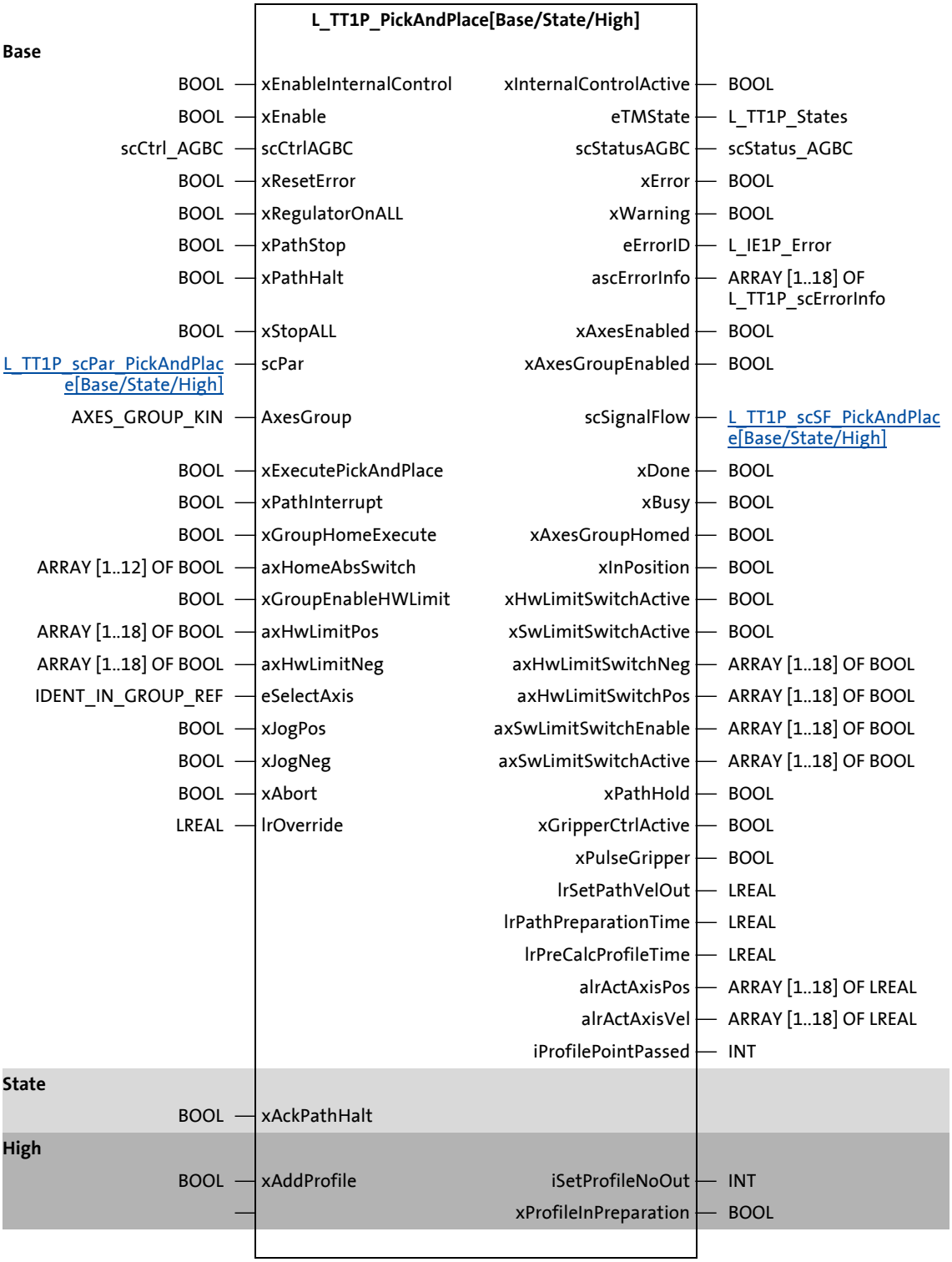
All functional parameters and the parameters for the setpoint generation are only set in one central position via the L_TT1P_scPar_PickAndPlace[Base/State/High] (📖 26) parameter structure at the *scPar* input of the technology module. The kinematic parameters and the limitations of the individual axes must be set in the axes group.



[3-2] Example: Interconnection of the technology module and the transformation Delta3 in the »PLC Designer«

3.4 L_TT1P_PickAndPlace[Base/State/High] function block

The figure shows the relation of the inputs and outputs to the "Base", "State" and "High" versions. The additional inputs and outputs of the "State" and "High" versions are shaded.



3.4.1 Inputs and outputs

Designator Data type	Description	Available in version		
		Base	State	High
AxesGroup AXES_GROUP_KIN	Reference to the axes group ▸ Interconnection of the technology module with the axes group (📖 13)	●	●	●

3.4.2 Inputs

Designator Data type	Description		Available in version		
			Base	State	High
xEnableInternalControl BOOL	TRUE	In the visualisation, the internal control of the axis can be selected via the "Internal Control" axis.	●	●	●
xEnable BOOL	Execution of the function block		●	●	●
	TRUE	The function block is executed.			
	FALSE	The function block is not executed.			
scCtrlAGBC scCtrl_AGBC	Input structure for the **L_MC4P_AxesGroupBasicControl** function block • scCtrlAGBC can be used in the "Ready" state. • If there is a request, the state changes to "Service". • The state change from "Service" back to "Ready" takes place if there are no more requests.		●	●	●
xResetError BOOL	TRUE	Reset error in all axes or software error.	●	●	●
xRegulatorOnALL BOOL	TRUE	Activate controller enable for all axes (via the **MC_Power** function block).	●	●	●
xPathStop BOOL	TRUE	Cancel the active movement and brake the axes to a standstill in a fashion that is accurate to the path (following the travel profile with the deceleration defined via IrPathStopDec. • The state changes to "Stop". • If the stop is executed from the technology module's movement, the axes are stopped on the path. • The technology module remains in the "STOP" state as long as xPathStop, xStopALL or xPathHalt is set to TRUE. • The "READY" state is set automatically when the axes are at standstill and xPathHalt, xPathStop and xStopALL are set to FALSE. • The xStopALL input is also active with "Internal Control". ▸ Stop/holding function (xPathStop, xStopALL, xPathHalt) (📖 47)	●	●	●
	FALSE	The state changes to "READY". Further instructions are expected.			

Designator Data type	Description		Available in version		
			Base	State	High
xPathHalt BOOL	TRUE	Cancel the active movement and brake the axes to a standstill in a fashion that is accurate to the path (following the travel profile with the deceleration defined via lrPathHaltDec. • The state changes to "Stop". • If the halt is executed from the technology module's movement, the axes are stopped on the path. • The technology module remains in the "STOP" state as long as xPathStop, xStopALL or xPathHalt is set to TRUE. • The "READY" state is set automatically when the axes are at standstill and xPathHalt, xPathStop and xStopALL are set to FALSE. • The xStopALL input is also active with "Internal Control".	●	●	●
	FALSE	The state changes to "READY". Further instructions are expected.			
xStopALL BOOL	TRUE	Bring all axes to a standstill individually, irrespectively of the path. **Note**: Executing this function during synchronised movements of the axes group may cause errors. • The individual axes are brought to a standstill independently of each other with the deceleration of the alrStopDec parameter. • The state changes to "Stop". • If the stop is executed from the technology module's movement, the reference to the path is cancelled. • The technology module remains in the "STOP" state as long as xPathStop, xStopALL or xPathHalt is set to TRUE. • The "READY" state is set automatically when the axes are at standstill and xPathHalt, xPathStop and xStopALL are set to FALSE. • The input is also active with "Internal Control". ▸ Stop/holding function (xPathStop, xStopALL, xPathHalt) (📖 47)	●	●	●
scPar L_TT1P_scPar_PickAndPlace[Base/State/High]	The parameter structure contains the parameters of the technology module. The data type depends on the version used (Base/State/High).		●	●	●
xExecutePickAndPlace BOOL	The input is edge-controlled and evaluates the rising edge.		●	●	●
	FALSE↗ TRUE	Path preparation is started (change to the "PREPARING_PATH" state). Then the path is executed or the path interpolation that has been stopped is continued (change to the "MOVE_PP" state).			

Designator Data type	Description		Available in version		
			Base	State	High
xPathInterrupt BOOL	By means of this function, the movement of the path can be stopped. • This function can only be activated in the "MOVE_PP" state. • The deceleration and acceleration ramps of the path are used.		●	●	●
	TRUE	• Brake all axes to standstill in a fashion that is accurate to the path (following the travel profile with the deceleration defined via lrPathStopDec. • The technology module changes to the "PATH INTERRUPT" state.			
	FALSE	• Execution of the path is continued in its breakpoint. • The technology module changes to the "MOVE_PP" state.			
xGroupHomeExecute BOOL	The input is edge-controlled and evaluates the rising edge.		●	●	●
	FALSE↗ TRUE	Start of the reference run (homing) for the real axes A1 ... A6 and Aux1 • Homing depends on the connected axes group. • The axes are only referenced in the order that is given via the aeHomingOrder parameter. • The parameters for homing are included in the L_TT1P_scPar_PickAndPlace[Base/State/High] (📖 26) parameter structure. • Initial value: the axis positions are set to the position in the alrHomePos parameter.			
axHomeAbsSwitch ARRAY [1..12] OF BOOL	Connection for reference switch For homing modes with a reference switch, connect this input to the digital signal which maps the state of the reference switch. axHomeAbsSwitch[axis] : connection of reference switch • [Axis] = 1 : axis 'A1' • [Axis] = 2 : axis 'A2' • [Axis] = 3 : axis 'A3' • [Axis] = 4 : axis 'A4' • [Axis] = 5 : axis 'A5' • [Axis] = 6 : axis 'A6' • [Axis] = 7 : axis 'Aux1'		●	●	●
xGroupEnableHWLimit BOOL	TRUE	Activation for evaluating the travel range limit switch (Hardware limit positions)	●	●	●

Designator Data type	Description		Available in version		
			Base	State	High
axHwLimitPos ARRAY [1..18] OF BOOL	Positive hardware limit switch Connect this input to the corresponding digital input that is connected to the limit switch. axHwLimitPos[axis] : connection of pos. hardware limit switch • [Axis] = 1 : X axis • [Axis] = 2 : Y axis • [Axis] = 3 : Z axis • [Axis] = 4 : A axis • [Axis] = 5 : B axis • [Axis] = 6 : C axis • [Axis] = 7 : axis 'A1' • [Axis] = 8 : axis 'A2' • [Axis] = 9 : axis 'A3' • [Axis] = 10 : axis 'A4' • [Axis] = 11 : axis 'A5' • [Axis] = 12 : axis 'A6' • [Axis] = 13 : axis 'Aux1'		●	●	●
	TRUE	The positive hardware limit switch has been reached or approached. • The axHwLimitSwitchPos output is also set to TRUE. • The axis is brought to a standstill with the deceleration in the alrStopDec parameter. • The state changes to "ERROR" with the error message '20500' (HWLimitPos).			
axHwLimitNeg ARRAY [1..18] OF BOOL	Negative hardware limit switch Connect this input to the corresponding digital input that is connected to the limit switch. axHwLimitNeg[axis] : connection of neg. hardware limit switch • [Axis] = 1 : X axis • [Axis] = 2 : Y axis • [Axis] = 3 : Z axis • [Axis] = 4 : A axis • [Axis] = 5 : B axis • [Axis] = 6 : C axis • [Axis] = 7 : axis 'A1' • [Axis] = 8 : axis 'A2' • [Axis] = 9 : axis 'A3' • [Axis] = 10 : axis 'A4' • [Axis] = 11 : axis 'A5' • [Axis] = 12 : axis 'A6' • [Axis] = 13 : axis 'Aux1'		●	●	●
	TRUE	The negative hardware limit switch has been reached or approached. • The axHwLimitSwitchNeg output is also set to TRUE. • The axis is brought to a standstill with the deceleration in the alrStopDec parameter. • The state changes to "ERROR" with the error message '20501' (HWLimitNeg).			

Designator Data type	Description		Available in version		
			Base	State	High
eSelectAxis IDENT_IN_GROUP_REF	Selection of the axis for the manual jog function		●	●	●
	0	No axis			
	1	X axis			
	2	Y axis			
	3	Z axis			
	4	A axis			
	5	B axis			
	6	C axis			
	7	Axis 'A1'			
	8	Axis 'A2'			
	9	Axis 'A3'			
	10	Axis 'A4'			
	11	Axis 'A5'			
	12	Axis 'A6'			
	13	Axis 'Aux1'			
xJogPos BOOL	TRUE	Traverse the axis selected at the eSelectAxis input in positive direction (manual jog). If xJogNeg is also TRUE, the traversing direction selected first remains set.	●	●	●
xJogNeg BOOL	TRUE	Traverse the axis selected at the eSelectAxis input in negative direction (manual jog). If xJogPos is also TRUE, the traversing direction selected first remains set.	●	●	●
xExecutePickAndPlace BOOL	The input is edge-controlled and evaluates the rising edge.		●	●	●
	FALSE↗ TRUE	Path preparation is started (change to the "PREPARING_PATH" state). Then the path is executed or the path interpolation that has been stopped is continued (change to the "MOVE_PP" state).			
xAbort BOOL	The input is edge-controlled and evaluates the rising edge.		●	●	●
	FALSE↗ TRUE	The travel on the path is aborted. • xAbort can be executed after a stop, a halt, or an interruption (xPathInterrupt) of the path. • For this, the axes must be at a standstill.			
lrOverride BOOL	Path overflow • Initial value: 1.0 • The value '0.5' halves the speed, the acceleration and the jerk. **Note:** For values not equalling '1.0', the internal gripper control does not work. Only use this input for commissioning purposes.		●	●	●
xAckPathHalt BOOL	TRUE	A breakpoint is acknowledged on the path. Breakpoints can be acknowledged anytime, irrespective of whether the path has reached a breakpoint or not.		●	●

Designator Data type	Description		Available in version		
			Base	State	High
xAddProfile BOOL	The input is edge-controlled and evaluates the rising edge.				●
	FALSE↗ TRUE	The profile from the scPar.ascProfilePar parameter is added to the profile in the "Pick & Place" technology module. Preconditions for loading the profile: • The technology module is in the "MOVE_PP" state. • The resources of the technology module for path planning are enabled for the next profile (output xProfileInPreparation = FALSE).			

3.4.3 Outputs

Designator Data type	Description		Available in version		
			Base	State	High
xInternalControlActive BOOL	TRUE	The internal control of the axis is activated via the visualisation. (xEnableInternalControl input = TRUE)	●	●	●
eTMState L_TT1P_States	Current state of the technology module ▸ State machine (📖 33)		●	●	●
scStatusAGBC scStatus_AGBC	Status data structure of the **L_MC4P_AxesGroupBasicControl** function block		●	●	●
xError BOOL	TRUE	There is an error in the technology module.	●	●	●
xWarning BOOL	TRUE	There is a warning in the technology module.	●	●	●
eErrorID L_IE1P_Error	ID of the error or warning message if xError = TRUE or xWarning = TRUE. **"FAST technology modules" reference manual:** Here you can find information on error or warning messages.		●	●	●
ascErrorInfo ARRAY [1..18] OF L_TT1P_scErrorInfo	Error information structure for a more detailed analysis of the error cause at the axes ascErrorInfo[axis]: • [Axis] = 1 : X axis • [Axis] = 2 : Y axis • [Axis] = 3 : Z axis • [Axis] = 4 : A axis • [Axis] = 5 : B axis • [Axis] = 6 : C axis • [Axis] = 7 : axis 'A1' • [Axis] = 8 : axis 'A2' • [Axis] = 9 : axis 'A3' • [Axis] = 10 : axis 'A4' • [Axis] = 11 : axis 'A5' • [Axis] = 12 : axis 'A6' • [Axis] = 13 : axis 'Aux1'		●	●	●
xAxesEnabled BOOL	TRUE	All axes are enabled/switched on.	●	●	●
xAxesGroupEnabled BOOL	TRUE	Axes group enabled/switched on.	●	●	●
scSignalFlow L_TT1P_scSF_PickAndPlace[Base/State/High]	Structure of the signal flow The data type depends on the version used (Base/State/High). ▸ Signal flow diagram (📖 34)		●	●	●
xDone BOOL	TRUE	The request/action has been completed successfully.	●	●	●
xBusy BOOL	TRUE	The request/action is currently being executed.	●	●	●
xAxesGroupHomed BOOL	TRUE	All axes have been referenced (reference known).	●	●	●
xInPosition BOOL	TRUE	The defined position of the "Tool Center Point" has been reached. When the xPosInWindow parameter is TRUE, the "Tool Center Point" (TCP) is monitored within the tolerance window for following error monitoring.	●	●	●

Designator Data type	Description		Available in version		
			Base	State	High
xHwLimitSwitchActive BOOL	TRUE	At least one axis has reached or approached a hardware limit switch. • The xHWLimitPos input is also set to TRUE. • The drive is braked to a standstill with the deceleration set in the aIrStopDec parameter. • The state changes to "ERROR" with the error message ’20500’ (HWLimitPos).	●	●	●
xSwLimitSwitchActive BOOL	TRUE	At least one axis has reached or exceeded a software limit position. • The drive is brought to a standstill with the deceleration set in the lrStopDec parameter. • The state changes to "ERROR" with error message '20306' (SWLimitPos) or '20307' (SWLimitNeg).	●	●	●
axHwLimitSwitchNeg ARRAY [1..18] OF BOOL	TRUE	The negative hardware limit switch has been reached or approached. • The axHwLimitNeg input has to be connected to the digital input that is connected to the limit switch. • The axHwLimitNeg input is also set to TRUE. • The axis is brought to a standstill with the deceleration in the aIrStopDec parameter. • The state changes to "ERROR" with the error message ’20501’ (HWLimitNeg). axHwLimitSwitchNeg[axis]: • [Axis] = 1 : X axis • [Axis] = 2 : Y axis • [Axis] = 3 : Z axis • [Axis] = 4 : A axis • [Axis] = 5 : B axis • [Axis] = 6 : C axis • [Axis] = 7 : axis ’A1’ • [Axis] = 8 : axis ’A2’ • [Axis] = 9 : axis ’A3’ • [Axis] = 10 : axis ’A4’ • [Axis] = 11 : axis ’A5’ • [Axis] = 12 : axis ’A6’ • [Axis] = 13 : axis ’Aux1’	●	●	●

Designator Data type	Description		Available in version		
			Base	State	High
axHwLimitSwitchPos ARRAY [1..18] OF BOOL	TRUE	The positive hardware limit switch has been reached or approached. • The axHwLimitPos input has to be connected to the digital input that is connected to the limit switch. • The axHwLimitPos input is also set to TRUE. • The axis is brought to a standstill with the deceleration in the alrStopDec parameter. • The state changes to "ERROR" with the error message '20500' (HWLimitPos). axHwLimitSwitchPos[axis]: • [Axis] = 1 : X axis • [Axis] = 2 : Y axis • [Axis] = 3 : Z axis • [Axis] = 4 : A axis • [Axis] = 5 : B axis • [Axis] = 6 : C axis • [Axis] = 7 : axis 'A1' • [Axis] = 8 : axis 'A2' • [Axis] = 9 : axis 'A3' • [Axis] = 10 : axis 'A4' • [Axis] = 11 : axis 'A5' • [Axis] = 12 : axis 'A6' • [Axis] = 13 : axis 'Aux1'	●	●	●
axSwLimitEnabled ARRAY [1..18] OF BOOL	TRUE	Activate the monitoring function of the software limit positions for the axis. axSwLimitEnabled[axis]: • [Axis] = 1 : X axis • [Axis] = 2 : Y axis • [Axis] = 3 : Z axis • [Axis] = 4 : A axis • [Axis] = 5 : B axis • [Axis] = 6 : C axis • [Axis] = 7 : axis 'A1' • [Axis] = 8 : axis 'A2' • [Axis] = 9 : axis 'A3' • [Axis] = 10 : axis 'A4' • [Axis] = 11 : axis 'A5' • [Axis] = 12 : axis 'A6' • [Axis] = 13 : axis 'Aux1'	●	●	●
axSwLimitSwitchActive ARRAY [1..18] OF BOOL	TRUE	The axis has reached or exceeded the software limit position. axSwLimitSwitchActive[axis]: • [Axis] = 1 : X axis • [Axis] = 2 : Y axis • [Axis] = 3 : Z axis • [Axis] = 4 : A axis • [Axis] = 5 : B axis • [Axis] = 6 : C axis • [Axis] = 7 : axis 'A1' • [Axis] = 8 : axis 'A2' • [Axis] = 9 : axis 'A3' • [Axis] = 10 : axis 'A4' • [Axis] = 11 : axis 'A5' • [Axis] = 12 : axis 'A6' • [Axis] = 13 : axis 'Aux1'	●	●	●

Designator Data type	Description		Available in version		
			Base	State	High
xPathHold BOOL	TRUE	The path interpolation has been stopped ... • by "Stop" and can be continued with the input xExecutePickAndPlace = TRUE; • with the input xPathInterrupt = TRUE and can be continued with xPathInterrupt = FALSE; • at a defined path point and can be continued by acknowledgement with the input xAckPathHalt = TRUE.	●	●	●
xGripperCtrlActive BOOL	TRUE	Activate gripper control. If the path is interrupted by "Stop", the gripper control is deactivated (xGripperCtrlActive = FALSE).	●	●	●
xPulseGripper BOOL	Control output for automatic gripper control • The output is only active for one cycle. • If the path is interrupted by a stop, gripper control is deactivated (xGripperCtrlActive = FALSE).		●	●	●
lrSetPathVelOut LREAL	Display of the current setpoint path speed • Unit: units/s		●	●	●
lrPathPreparationTime LREAL	Time required for the calculation of the path preparation • Unit: s		●	●	●
lrPreCalcProfileTime LREAL	Time calculated for the travel process of the profile to be executed • Unit: s		●	●	●
alrActPos ARRAY [1..18] OF LREAL	Current position of the axes • Unit: units		●	●	●
	1	X axis			
	2	Y axis			
	3	Z axis			
	4	A axis			
	5	B axis			
	6	C axis			
	7	Axis 'A1'			
	8	Axis 'A2'			
	9	Axis 'A3'			
	10	Axis 'A4'			
	11	Axis 'A5'			
	12	Axis 'A6'			
	13	Axis 'Aux1'			

Designator Data type	Description		Available in version		
			Base	State	High
alrActVel ARRAY [1..18] OF LREAL	Current speed of the axes • Unit: units/s		●	●	●
	1	X axis			
	2	Y axis			
	3	Z axis			
	4	A axis			
	5	B axis			
	6	C axis			
	7	Axis 'A1'			
	8	Axis 'A2'			
	9	Axis 'A3'			
	10	Axis 'A4'			
	11	Axis 'A5'			
	12	Axis 'A6'			
	13	Axis 'Aux1'			
iProfilePointPassed INT	Output of the number of the point that has been reached in the current profile.		●	●	●
iSetProfileNoOut INT	Output of the number of the current profile.				●
xProfileInPreparation BOOL	TRUE	The technology module is preparing the profile.			●
	FALSE	The resources for the preparation of a profile have been enabled. A new profile can be appended via the xAddProfile input.			

3.4.4 Parameters

L_TT1P_scPar_PickAndPlace[Base/State/High]

The **L_TT1P_scPar_PickAndPlace[Base/State/High]** structure contains the parameters of the technology module.

Designator Data type	Description	Available in version		
		Base	State	High
lrPathStopDec LREAL	Deceleration of the path for the input xPathStop = TRUE In order to prevent overtravelling of the programmed target position, this parameter is only taken into consideration for Cartesian movements if the values specified are higher than those of the path that is currently interpolated. • Unit: units/s^2 • Initial value: 10000	●	●	●
lrPathStopJerk LREAL	Jerk of the path for the input xPathStop = TRUE In order to prevent overtravelling of the programmed target position, this parameter is only taken into consideration for Cartesian movements if the values specified are higher than those of the path that is currently interpolated. • Unit: units/s^3 • Initial value: 100000	●	●	●
lrPathHaltDec LREAL	Deceleration of the path for the input xPathHalt = TRUE In order to prevent overtravelling of the programmed target position, this parameter is only taken into consideration for Cartesian movements if the values specified are higher than those of the path that is currently interpolated. • Unit: units/s^2 • Initial value: 100	●	●	●
lrPathJerk LREAL	Jerk of the path for the input xPathHalt = TRUE and the path interpolation • Unit: units/s^3 • Initial value: 100000	●	●	●
alrStopDec ARRAY [1..18] OF LREAL	Deceleration of the individual axes for the input xStopALL = TRUE, or when the hardware limit switches, software limit positions, and following error monitoring are triggered • Unit: units/s^2 • Initial value: 10000 alrStopDec[axis]: • [Axis] = 1 : X axis • [Axis] = 2 : Y axis • [Axis] = 3 : Z axis • [Axis] = 4 : A axis • [Axis] = 5 : B axis • [Axis] = 6 : C axis • [Axis] = 7 : axis 'A1' • [Axis] = 8 : axis 'A2' • [Axis] = 9 : axis 'A3' • [Axis] = 10 : axis 'A4' • [Axis] = 11 : axis 'A5' • [Axis] = 12 : axis 'A6' • [Axis] = 13 : axis 'Aux1'	●	●	●

Designator Data type	Description	Available in version		
		Base	State	High
alrStopJerk ARRAY [1..18] OF LREAL	Jerk of the individual axes for the input xStopALL = TRUE, or when the hardware limit switches, software limit positions, and following error monitoring are triggered • Unit: units/s^3 • Initial value: 100000 alrStopJerk[axis]: • [Axis] = 1 : X axis • [Axis] = 2 : Y axis • [Axis] = 3 : Z axis • [Axis] = 4 : A axis • [Axis] = 5 : B axis • [Axis] = 6 : C axis • [Axis] = 7 : axis 'A1' • [Axis] = 8 : axis 'A2' • [Axis] = 9 : axis 'A3' • [Axis] = 10 : axis 'A4' • [Axis] = 11 : axis 'A5' • [Axis] = 12 : axis 'A6' • [Axis] = 13 : axis 'Aux1'	●	●	●
lrCartesianJogJerk LREAL	Jerk for manual jog of the Cartesian axes X, Y and Z • Unit: units/s^3 • Initial value: 10000	●	●	●
lrCartesianJogVel LREAL	Speed for manual jog of the Cartesian axes X, Y and Z • Unit: units/s • Initial value: 10	●	●	●
lrCartesianJogAcc LREAL	Acceleration for manual jog of the Cartesian axes X, Y and Z • Unit: units/s^2 • Initial value: 100	●	●	●
lrCartesianJogDec LREAL	Deceleration for manual jog of the Cartesian axes X, Y and Z • Unit: units/s^2 • Initial value: 100	●	●	●
lrOrientationJogJerk LREAL	Jerk for manual jog of orientation axes A, B and C • Unit: units/s^3 • Initial value: 10000	●	●	●
lrOrientationJogVel LREAL	Speed for manual jog of orientation axes A, B and C • Unit: units/s • Initial value: 10	●	●	●
lrOrientationJogAcc LREAL	Acceleration for manual jog of orientation axes A, B and C • Unit: units/s^2 • Initial value: 100	●	●	●
lrOrientationJogDec LREAL	Deceleration for manual jog of orientation axes A, B and C • Unit: units/s^2 • Initial value: 100	●	●	●
lrRealAxisJogJerk LREAL	Jerk for manual jog of the real axes A1 ... A6 and Aux1 • Unit: units/s^3 • Initial value: 10000	●	●	●
lrRealAxisJogVel LREAL	Velocity for manual jog of the real axes A1 ... A6 and Aux1 • Unit: units/s • Initial value: 10	●	●	●
lrRealAxisJogAcc LREAL	Acceleration for manual jog of the real axes A1 ... A6 and Aux1 • Unit: units/s^2 • Initial value: 100	●	●	●
lrRealAxisJogDec LREAL	Deceleration for manual jog of the real axes A1 ... A6 and Aux1 • Unit: units/s^2 • Initial value: 100	●	●	●

Designator Data type	Description		Available in version		
			Base	State	High
alrHomePosition ARRAY [1..12] OF LREAL	Home position for the desired axis The reference run (homing) is started with the xGroupHomeExecute input. The axes are only referenced in the order that is given via the aeHomingOrder parameter. • Unit: units • Initial value: 0 alrHomePosition[axis]: • [Axis] = 1 : axis 'A1' • [Axis] = 2 : axis 'A2' • [Axis] = 3 : axis 'A3' • [Axis] = 4 : axis 'A4' • [Axis] = 5 : axis 'A5' • [Axis] = 6 : axis 'A6' • [Axis] = 7 : axis 'Aux1'		●	●	●

aeHomingOrder
ARRAY [1..12] OF L_MC4P_HomingOrder
	Definition of the sequence in which the axes are to be referenced:
	• NoHoming (standard setting)
	• First, Second, Third, Fourth, Fifth, Sixth, Seventh, Eighth, Ninth, Tenth, Eleventh, Twelfth
	aeHomingOrder[axis]:
	• [Axis] = 1 : axis 'A1'
	• [Axis] = 2 : axis 'A2'
	• [Axis] = 3 : axis 'A3'
	• [Axis] = 4 : axis 'A4'
	• [Axis] = 5 : axis 'A5'
	• [Axis] = 6 : axis 'A6'
	• [Axis] = 7 : axis 'Aux1'
	Example
	`aeHomingOrder[1] := First;`
	`aeHomingOrder[2] := First;`
	`aeHomingOrder[3] := Second,,`
	`aeHomingOrder[7] := NoHoming;`
	Axes A1 and A2 are referenced isochronously, then axis A3. Axis Aux1 is not referenced.
			●	●	●

xUseHomeExtParameter BOOL	Selection of the homing parameters to be used • Initial value: FALSE		●	●	●
	FALSE	The homing parameters defined in the axis data are used.			
	TRUE	The **ascHomeExtParameter** homing parameters from the application are used.			
ascHomeExtTP ARRAY [1..12] OF 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.		●	●	●
ascHomeExtParameter ARRAY [1..12] OF L_MC1P_HomeParameter	Homing parameters from the application for the desired axis • Only relevant if xUseHomeExtParameter = TRUE. ascHomeExtParameter[axis]: • [Axis] = 1 : axis 'A1' • [Axis] = 2 : axis 'A2' • [Axis] = 3 : axis 'A3' • [Axis] = 4 : axis 'A4' • [Axis] = 5 : axis 'A5' • [Axis] = 6 : axis 'A6' • [Axis] = 7 : axis 'Aux1'		●	●	●

Designator Data type	Description	Available in version		
		Base	State	High
lrXMaxVel LREAL	Maximum velocity of the Cartesian X axis during the path interpolation • Unit: units/s • Initial value: 10	●	●	●
lrYMaxVel LREAL	Maximum velocity of the Cartesian Y axis during the path interpolation • Unit: units/s • Initial value: 10	●	●	●
lrZMaxVel LREAL	Maximum velocity of the Cartesian Z axis during the path interpolation • Unit: units/s • Initial value: 10	●	●	●
lrAMaxVel LREAL	Maximum velocity for the A axis during the path interpolation • Unit: units/s • Initial value: 10	●	●	●
lrBMaxVel LREAL	Maximum velocity for the B axis during the path interpolation • Unit: units/s • Initial value: 10	●	●	●
lrCMaxVel LREAL	Maximum velocity for the C axis during the path interpolation • Unit: units/s • Initial value: 10	●	●	●
lrAux1MaxVel LREAL	Maximum velocity for the Aux1 axis during the path interpolation • Unit: units/s • Initial value: 10	●	●	●
lrXMaxAccDec LREAL	Maximum acceleration/deceleration for the Cartesian X axis during the path interpolation • Unit: units/s^2 • Initial value: 100	●	●	●
lrYMaxAccDec LREAL	Maximum acceleration/deceleration for the Cartesian Y axis during the path interpolation • Unit: units/s^2 • Initial value: 100	●	●	●
lrZMaxAccDec LREAL	Maximum acceleration/deceleration for the Cartesian Z axis during the path interpolation • Unit: units/s^2 • Initial value: 100	●	●	●
lrAMaxAccDec LREAL	Maximum acceleration/deceleration for the A axis during the path interpolation • Unit: units/s^2 • Initial value: 100	●	●	●
lrBMaxAccDec LREAL	Maximum acceleration/deceleration for the B axis during the path interpolation • Unit: units/s^2 • Initial value: 100	●	●	●
lrCMaxAccDec LREAL	Maximum acceleration/deceleration for the C axis during the path interpolation • Unit: units/s^2 • Initial value: 100	●	●	●
lrAux1MaxAccDec LREAL	Maximum acceleration/deceleration for the Aux1 axis during the path interpolation • Unit: units/s^2 • Initial value: 100	●	●	●

Designator Data type	Description		Available in version		
			Base	State	High
lrXMaxJerk LREAL	Maximum jerk for the Cartesian X axis during the path interpolation • Unit: units/s^3 • Initial value: 10000		●	●	●
lrYMaxJerk LREAL	Maximum jerk for the Cartesian Y axis during the path interpolation • Unit: units/s^3 • Initial value: 10000		●	●	●
lrZMaxJerk LREAL	Maximum jerk for the Cartesian Z axis during the path interpolation • Unit: units/s^3 • Initial value: 10000		●	●	●
lrAMaxJerk LREAL	Maximum jerk for the A axis during the path interpolation • Unit: units/s^3 • Initial value: 10000		●	●	●
lrBMaxJerk LREAL	Maximum jerk for the B axis during the path interpolation • Unit: units/s^3 • Initial value: 10000		●	●	●
lrCMaxJerk LREAL	Maximum jerk for the C axis during the path interpolation • Unit: units/s^3 • Initial value: 10000		●	●	●
lrAux1MaxJerk LREAL	Maximum jerk for the Aux1 axis during the path interpolation • Unit: units/s^3 • Initial value: 10000		●	●	●
lrGripperClosingTime LREAL	Deceleration of the gripper control • Positive values: the gripper is opened before the path profile is completed. • Negative values: the gripper is opened after the path profile has been completed. • Unit: s		●	●	●
xPosInWindow BOOL	Activation of the tolerance window for following error monitoring • Initial value: FALSE		●	●	●
	TRUE	The tolerance window is activated. The xInPosition output is set to TRUE if the "Tool Center Point" (tool zero point) is within the tolerance window.			
	FALSE	The tolerance window is not activated.			
lrPosInWindow LREAL	Size of the tolerance window for following error monitoring • Unit: units • Initial value: 0.5		●	●	●
lrTimePosInWindow LREAL	Period during which the "Tool Center Point" (tool zero point) is within the tolerance window (duration of the following error) • Unit: ms • Initial value: 50		●	●	●
eTargetCoordSystem L_MC4P_CoordSystem	Selection of the coordinate system to be used in the target point: • MCS: Machine Coordinate System (initial value) • PCS_1...16: Product Coordinate System (1...16) • ACS: Axes Coordinate System - **is not supported!**		●		
lrXTargetPos LREAL	Target position in X direction for the "Pick & Place" profile in the Base version • Unit: units		●		

Designator Data type	Description		Available in version		
			Base	State	High
lrYTargetPos LREAL	Target position in Y direction for the "Pick & Place" profile in the Base version • Unit: units		●		
lrZTargetPos LREAL	Target position in Z direction for the "Pick & Place" profile in the Base version • Unit: units		●		
lrATargetPos LREAL	Target position in A direction for the "Pick & Place" profile in the Base version • Unit: units • Initial value: 180		●		
lrBTargetPos LREAL	Target position in B direction for the "Pick & Place" profile in the Base version • Unit: units		●		
lrCTargetPos LREAL	Target position in C direction for the "Pick & Place" profile in the Base version • Unit: units		●		
lrAux1TargetPos LREAL	Target position in Aux1 direction for the "Pick & Place" profile in the Base version • Unit: units		●		
lrZStartDist LREAL	Difference in height at the start in Z direction • Unit: units		●		
lrZTargetDist LREAL	Difference in height at the end/target in Z direction • Unit: units		●		
lrBlendingRadius LREAL	Blending radius for the "Pick & Place" profile in the Base version The value specifies the distance from the target point in which the rounding motion is to start. • Unit: units		●		
lrPathVel LREAL	Limitation of the path speed • Unit: units/s • Initial value: 10		●		
lrPathAcc LREAL	Limitation of the path acceleration • Unit: units/s^2 • Initial value: 100		●		
lrPathDec LREAL	Limitation of the path deceleration • Unit: units/s^2 • Initial value: 100		●		
xSingleStep BOOL	TRUE	Activate single step function: • During the path interpolation, the interpolator stops in every point. • The breakpoint must be acknowledged during the interpolation with input xAckPathHalt = TRUE.		●	●
ascProfilePar ARRAY[1..10] OF L_TT1P_scProfilePar	In the ascProfilePar structure, a maximum of 10 interpolation points can be defined for a travel profile in the Cartesian coordinate system.			●	●

3.4.5 Parameters for interpolation points (travel profile parameters)

L_TT1P_scProfilePar

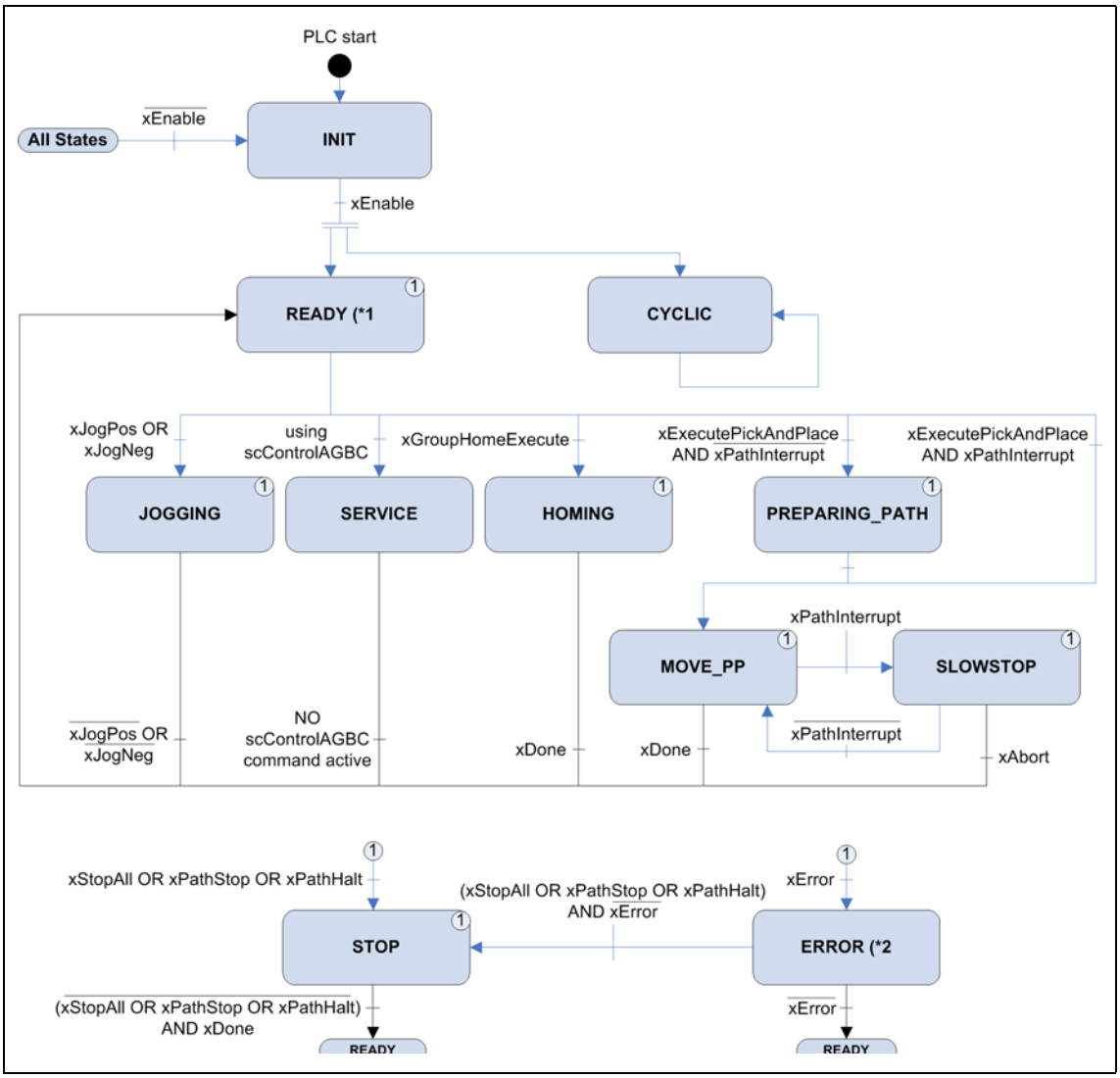
In the *ascProfilePar* parameter in the L_TT1P_scPar_PickAndPlace[Base/State/High] (📖 26) parameter structure, a data array with 10 interpolation points can be specified for a travel profile.

The *ascProfilePar* parameter is only provided in the "State" and "High" technology module versions.

Each interpolation point features the following parameters:

Designator Data type	Description		Available in version		
			Base	State	High
eCoordSystem L_MC4P_CoordSystem	Selection of the coordinate system to be used: • MCS: Machine Coordinate System (initial value) • PCS_1...16: Product Coordinate System (1...16) • ACS: Axes Coordinate System - **is not supported!**			●	●
lrXPos LREAL	Position in X direction • Unit: units			●	●
lrYPos LREAL	Position in Y direction • Unit: units			●	●
lrZPos LREAL	Position in Z direction • Unit: units			●	●
lrAPos LREAL	Position in A direction • Unit: units • Initial value: 180			●	●
lrBPos LREAL	Position in B direction • Unit: units			●	●
lrCPos LREAL	Position in C direction • Unit: units			●	●
lrAux1Pos LREAL	Position in Aux1 direction • Unit: units			●	●
lrPathVel LREAL	Limitation of the path speed • Unit: units/s • Initial value: 0			●	●
lrPathAcc LREAL	Limitation of the path acceleration • Unit: units/s^2 • Initial value: 0			●	●
lrPathDec LREAL	Limitation of the path deceleration • Unit: units/s^2 • Initial value: 0			●	●
lrBlendingRadius LREAL	Blending radius The value specifies the distance from the target point in which the rounding motion is to start. • Unit: units • Initial value: 0			●	●
xHalt BOOL	TRUE	The path is stopped. The breakpoint must be acknowledged during the interpolation with input xAckPathHalt = TRUE.		●	●

3.5 State machine



[3-3] State machine of the technology module

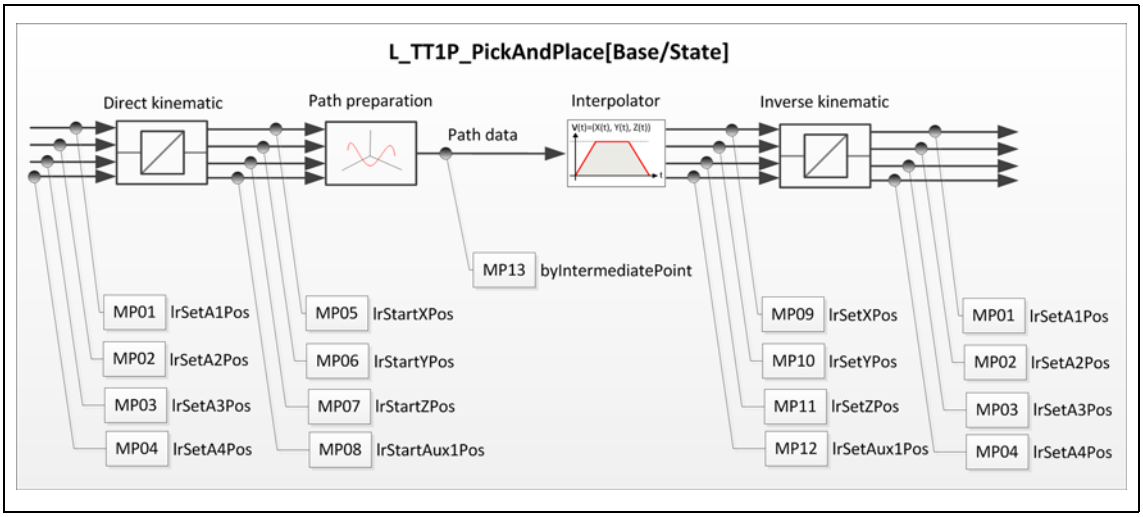
(*1 In the "Ready" state, xRegulatorOnALL has to be set to TRUE.

(*2 In the "ERROR" state, xResetError has to be set to TRUE in order to acknowledge and reset the errors.

3.6 Signal flow diagram

The illustrations show the main signal flow of the functions implemented.

The signal flow of the additional functions such as "manual jog" is not displayed here.



[3-4] Signal flow diagram: Pick & Place

3.6.1 Structure of the signal flow

L_TT1P_scSF_PickAndPlace[Base/State/High]

The contents of the **L_TT1P_scSF_PickAndPlace[Base/State/High]** structure are read-only and offer a practical diagnostics option within the signal flow (Signal flow diagram (📖 34)).

Designator Data type	Description	Available in version		
		Base	State	High
MP00_lwTCheckSum LWORD	Checksum	●	●	●
MP01_lrSetA1Pos LREAL	Position setpoints for axes A1 ... A4 Before a movement in the virtual space (X, Y, Z) is executed, the setpoints of the real axes (A1 ... A4) are converted via the direct kinematics and are used as starting value.	●	●	●
MP02_lrSetA2Pos LREAL		●	●	●
MP03_lrSetA3Pos LREAL		●	●	●
MP04_lrSetA4Pos LREAL		●	●	●
MP05_lrStartXPos LREAL	Starting value for axes X, Y, Z	●	●	●
MP06_lrStartYPos LREAL		●	●	●
MP07_lrStartZPos LREAL		●	●	●
MP08_lrStartAux1Pos LREAL	Starting value for the Aux1 axis	●	●	●
MP09_lrSetXPos LREAL	Position setpoints for axes X, Y, Z	●	●	●
MP10_lrSetYPos LREAL		●	●	●
MP11_lrSetZPos LREAL		●	●	●
MP12_lrSetAux1Pos LREAL	Position setpoint for the Aux1 axis	●	●	●
MP13_byIntermediatePoint BYTE	Number of the intermediate point that is approached in the current path.	●	●	●

3.7 Manual jog (jogging)

Precondition

- The technology module is in the "Ready" state.
- All axes are enabled (*xRegulatorOnALL* input = TRUE).

Execution

The axis to be travelled is selected via the *eSelectAxis* input:

Selection of eSelectAxis	
Value	**Axis to be travelled**
0	No axis
1	X axis
2	Y axis
3	Z axis
4	A axis
5	B axis
6	C axis
7	Axis 'A1'
8	Axis 'A2'
9	Axis 'A3'
10	Axis 'A4'
11	Axis 'A5'
12	Axis 'A6'
13	Axis 'Aux1'

Outside the "READY" state, a change of axis via the *eSelectAxis* input has no impact.

If the *xJogPos* input is TRUE, the axis is traversed in positive direction, and if the *xJogNeg* is TRUE, it is traversed in negative direction. The axis is executed for as long as the input remains TRUE. It is only possible to traverse one axis at a time.

The current travel command cannot be replaced by another jog command. Only if both inputs have been reset, the State machine (📖 33) changes to the "Ready" state again.

Parameters to be set

The parameters to be set for the manual jog are located in the L_TT1P_scPar_PickAndPlace[Base/State/High] (📖 26) parameter structure.

```
lrCartesianJogVel : LREAL := 10;     // for axes X,Y,Z
lrCartesianJogAcc : LREAL := 100;    // for axes X,Y,Z
lrCartesianJogDec : LREAL := 100;    // for axes X,Y,Z
lrCartesianJogJerk : LREAL := 10000; // for axes X,Y,Z
lrOrientationJogVel : LREAL := 10;     // for axes A,B,C
lrOrientationJogAcc : LREAL := 100;    // for axes A,B,C
lrOrientationJogDec : LREAL := 100;    // for axes A,B,C
lrOrientationJogJerk : LREAL := 10000; // for axes A,B,C
lrRealAxisJogVel : LREAL := 10;        // for axes A1...A6,Aux1
lrRealAxisJogAcc : LREAL := 100;       // for axes A1...A6,Aux1
lrRealAxisJogDec : LREAL := 100;       // for axes A1...A6,Aux1
lrRealAxisJogJerk : LREAL := 10000;    // for axes A1...A6,Aux1
```

The parameter values can be changed during operation. They are accepted when the *xJogPos* or *xJogNeg* input is set to TRUE again.

3.8 Homing

Precondition

- The technology module is in the "Ready" state.
- The slave axis is enabled (*xRegulatorOn* = TRUE).

Execution

Homing is started with a rising edge (FALSE↗TRUE) at the *xGroupHomeExecute* input. The axis will be travelling until the home position is reached. After successful homing, the State machine (📖 33) changes back again to the "Ready" state.

Homing is not interrupted if the *xGroupHomeExecute* input is set to FALSE too early.

Depending on the connect axes group, only the real axes A1 ... A6 and Aux1 are referenced.

The axes are only referenced in the order that is given via the *aeHomingOrder* parameter.

Parameters to be set

The parameters for homing are located in the L_TT1P_scPar_PickAndPlace[Base/State/High] (📖 26) parameter structure.

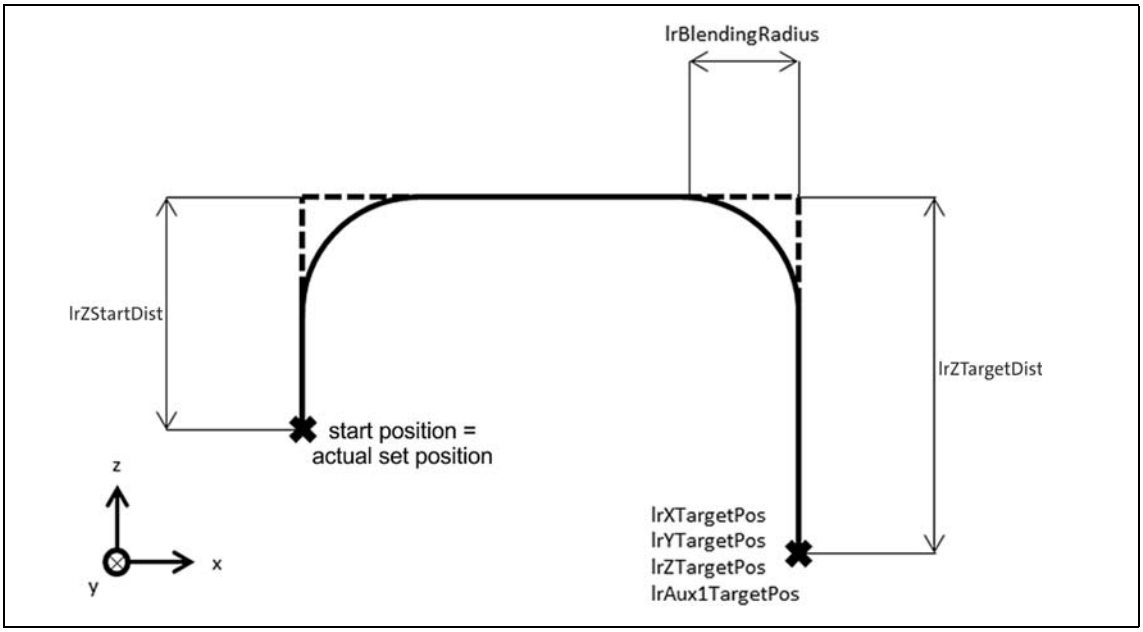
```
aeHomingOrder : ARRAY [1..12] OF L_MC4P_HomingOrder := NoHoming;
xUseHomeExtParameter : BOOL := FALSE;
alrHomePosition : ARRAY OF LREAL := 0;
ascHomeExtParameter : ARRAY OF L_MC1P_HomeParameter;
```

»PLC Designer« Online help

Further information about homing can be found in the description of the **L_MC4P_AxesGroupBasicControl** function block.

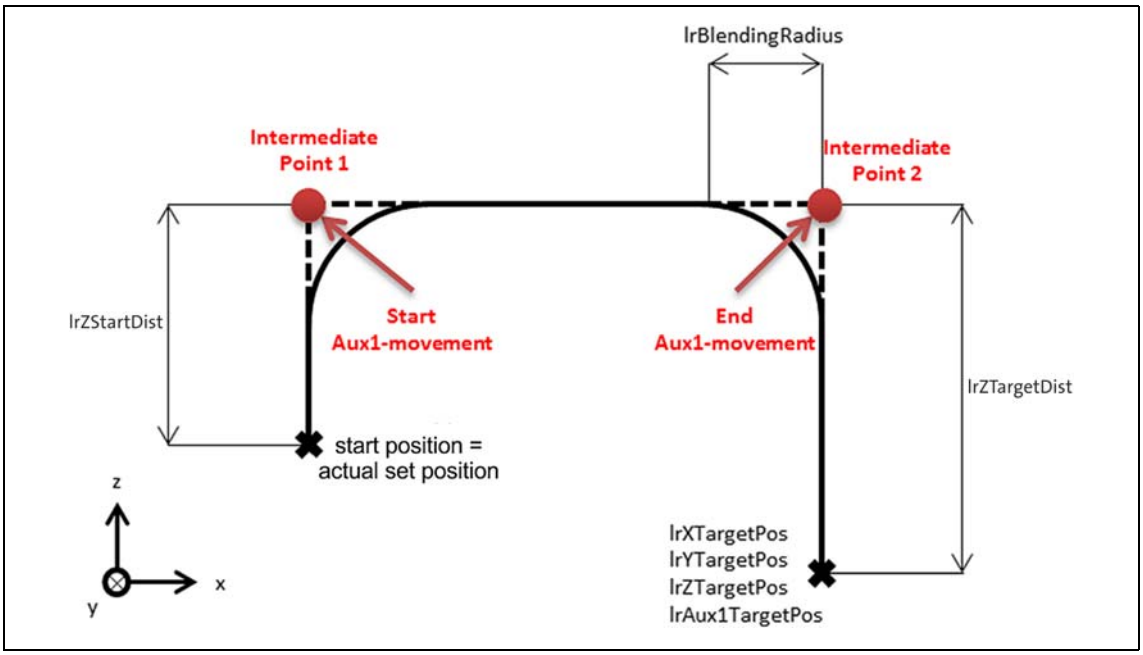
3.9 Specifying a travel profile

In the Base version, a simple "Pick & Place" profile can be defined by means of only a few parameters.



[3-5] Parameters for a simple "Pick & Place" profile

3.10 Calculation of the profile points with lrBlendingRadius = 0



[3-6] Calculation of the profile points with lrBlendingRadius = 0

In the case of this travel profile, the two intermediate points (intermediate points 1 and 2) between the starting and target position are approached and a short stop in the point is carried out (path speed = 0).

The actual position (X, Y, Z, A, B, C, Aux1) of the "Tool Center Point" in a certain space is the starting position of the profile.

The *lrDeltaZStart* parameter is used to define the height of the "Tool Center Point" (Intermediate Point 1) from which the target position in X/Y/Z/A/B/C direction is approached. This point is defined in the same coordinate system of the actual position (X, Y, Z, A, B, C, Aux1) of the "Tool Center Point".

For each interpolation point, a coordinate system can be defined via the *eCoordSystem* parameter. The "Machine Coordinate System" (MCS) and all "Product Coordinate Systems" (PCS 1...16) are supported. The "Axes Coordinate System" (ACS) is not supported.

The coordinates of "Intermediate Point 2" result from the sum of the target coordinates X, Y, Z, A, B, C, Aux1 using the *lrZTargetDist* parameter in the coordinate system. When this point has been reached, the target position is approached in X/Y/Z/A/B/C direction.

The movement of orientation axes A, B, C and auxiliary axis Aux1 starts in "Intermediate Point 1" and ends in "Intermediate Point 2".

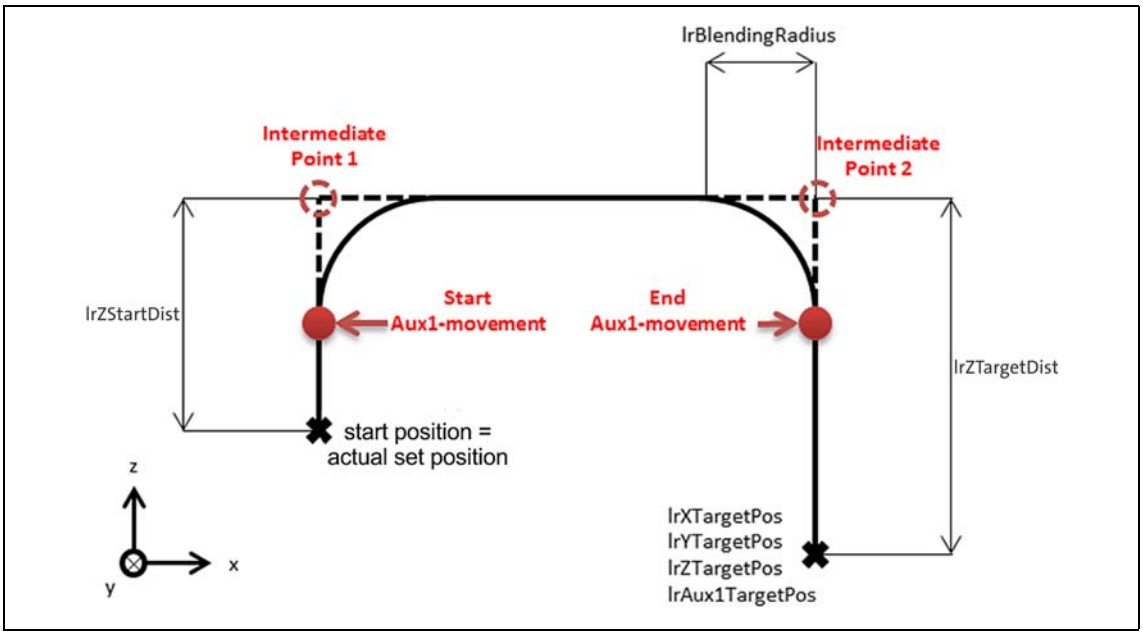
Parameters to be set

The parameters to be set are located in the parameter structures ...

- L_TT1P_scPar_PickAndPlace[Base/State/High] (📖 26);
- L_TT1P_scProfilePar (📖 32).

```
eTargetCoordSystem : L_MC4P_CoordSystem;
lrXTargetPos : LREAL := 0;
lrYTargetPos : LREAL := 0;
lrZTargetPos : LREAL := 0;
lrATargetPos : LREAL := 180;
lrBTargetPos : LREAL := 0;
lrCTargetPos : LREAL := 0;
lrAux1TargetPos : LREAL := 0;
lrZStartDist : LREAL := 0;
lrZTargetDist : LREAL := 0;
lrBlendingRadius : LREAL := 0;
lrPathVel : LREAL := 10;
lrPathAcc : LREAL := 100;
lrPathDec : LREAL := 100;
xPosInWindow : BOOL := FALSE;
lrPosInWindow : LREAL := 0.5; [units]
lrTimePosInWindow : LREAL := 50; [ms]
```

3.11 Calculation of the profile points with lrBlendingRadius > 0



[3-7] Calculation of the profile points with lrBlendingRadius > 0

In the case of this travel profile, the two intermediate points (intermediate points 1 and 2) between the starting and target position are not approached but the profile is rounded around these points. The profile is executed without a stop.

The actual position (X, Y, Z, A, B, C, Aux1) of the "Tool Center Point" in a certain space is the starting position of the profile.

The *lrZStartDist* parameter is used to define the height of the "Tool Center Point" (Intermediate Point 1). The profile travel process is lead around "Intermediate Point 1", depending on the blending radius in the *lrBlendingRadius* parameter.

The coordinates of "Intermediate Point 2" result from the sum of the target coordinates X, Y, Z, A, B, C, Aux1 using the *lrZTargetDist*. parameter. Along the lines of the description above, the profile travel process here also takes place around "Intermediate Point 2", depending on the *lrBlendingRadius* parameter.

There is a maximum limit for setting the blending radius. The technology module internally limits the blending radius to a maximum of half of the path between the adjacent "Intermediate Points".

The movement of orientation axes A, B, C, and the auxiliary axis Aux1 starts in the moment in which the rounding movement around "Intermediate Point 1" is starting (depending on the blending radius). The movement of the axes ends when the rounding movement around "Intermediate Point 2" has been completed.

Parameters to be set

The parameters to be set are located in the parameter structures ...

- L_TT1P_scPar_PickAndPlace[Base/State/High] (📖 26);
- L_TT1P_scProfilePar (📖 32).

```
lrXTargetPos : LREAL := 0;
lrYTargetPos : LREAL := 0;
lrZTargetPos : LREAL := 0;
lrATargetPos : LREAL := 180;
lrBTargetPos : LREAL := 0;
lrCTargetPos : LREAL := 0;
lrAux1TargetPos : LREAL := 0;
lrZStartDist : LREAL := 0;
lrZTargetDist : LREAL := 0;
lrBlendingRadius : LREAL := 0;
lrPathVel : LREAL := 10;
lrPathAcc : LREAL := 100;
lrPathDec : LREAL := 100;
xPosInWindow : BOOL := FALSE;
lrPosInWindow : LREAL := 0.5; [units]
lrTimePosInWindow : LREAL := 50; [ms]
```

3.12 Limitations (maximum values) for the axes on the travel profile

For the path calculation or interpolation, maximum values for the speed, acceleration and jerk can be defined for all axes.

Depending on these values, the resulting travel profile for the path is adapted:

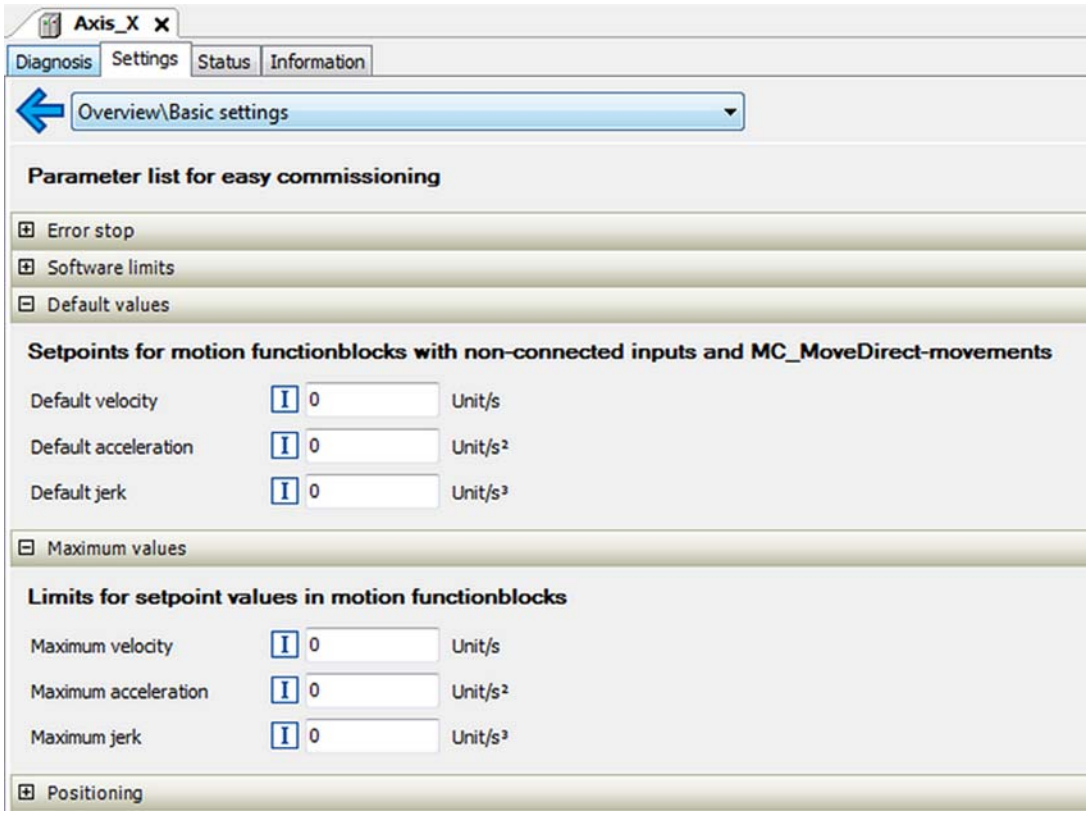
- In the course of the path calculation, the speed, acceleration and jerk of the path are automatically reduced, so that the maximum values of axes X, Y, Z, A, B, C and Aux1 are not exceeded.
- For the interpolation, the maximum values from the axis are taken into consideration. If the maximum values are exceeded in this process, interpolation is carried out with the maximum speed, maximum acceleration and the maximum jerk that have been set.

The limitations are set at two points:

- Via the respective reference axis **AXIS_REF**
- Via parameter setting of the maximum values for velocity, acceleration and jerk for the axes X, Y, Z, A, B, C and Aux1.

The setting ’0’ deactivates the limitation.

In the »PLC Designer« the maximum values can be set under the "Settings" of the reference axis:



Parameters to be set

The parameters to be set are located in the L_TT1P_scPar_PickAndPlace[Base/State/High] (📖 26) parameter structure.

```
lrXMaxVel : LREAL := 10;
lrYMaxVel : LREAL := 10;
lrZMaxVel : LREAL := 10;
lrAMaxVel : LREAL := 10;
lrBMaxVel : LREAL := 10;
lrCMaxVel : LREAL := 10;
lrAux1MaxVel : LREAL := 10;
lrXMaxAccDec : LREAL := 1000;
lrYMaxAccDec : LREAL := 1000;
lrZMaxAccDec : LREAL := 1000;
lrAMaxAccDec : LREAL := 1000;
lrBMaxAccDec : LREAL := 1000;
lrCMaxAccDec : LREAL := 1000;
lrAux1MaxAccDec : LREAL := 1000;
lrXMaxJerk : LREAL := 10000;
lrYMaxJerk : LREAL := 10000;
lrZMaxJerk : LREAL := 10000;
lrAMaxJerk : LREAL := 10000;
lrBMaxJerk : LREAL := 10000;
lrCMaxJerk : LREAL := 10000;
lrAux1Jerk : LREAL := 10000;
```

3.13 Starting the travel profile (xExecutePickAndPlace)

With a rising edge (FALSE↗TRUE) at the input *xExecutePickAndPlace*, the travel profile is started. For this purpose, the profile for the path is calculated and prepared in the background.

The *lrPreCalcProfileTime* output shows the time calculated for the travel from the start to the target point of the current profile to be travelled.

Possibilities of influencing the profile

- Input *xPathStop* = TRUE
  With input *xPathStop* = TRUE, all axes are brought to a standstill so that they are accurate to the path (following the travel profile).
  When the movement has been started again, execution of the path is continued from its breakpoint.
  ▸ Stop/holding function (xPathStop, xStopALL, xPathHalt) (📖 47)
- Input *xStopALL* = TRUE
  With input *xStopALL* = TRUE, all axes are brought to a standstill individually – irrespectively of the path .
  When "Stop" has been executed, the technology module changes to the "ERROR" state (display at the *eTMState* output).
- *xPathInterrupt* input = TRUE
  ▸ Interrupting the travel profile/path (xPathInterrupt) (📖 48)
- *xAbort* input = TRUE
  With *xAbort* = TRUE, the travel profile can be cancelled.
  The *xAbort* input can only be used if the axes have been stopped via *xPathStop*, *xPathHalt* or *xPathInterrupt* before.

3.14 Stop/holding function (xPathStop, xStopALL, xPathHalt)

xPathStop and xPathHalt inputs

With the input *xPathStop* = TRUE or *xPathHalt* = TRUE, all axes are brought to a standstill with the deceleration defined via the *lrPathStopDec* or *lrPathHaltDec* parameter, so that they are accurate to the path (following the travel profile). This state is shown with the output *xPathHold* = TRUE.

The technology module changes to the "STOP" state (display at the *eTMState* output).

With the stop function, the jerk limitation is preselected via the *lrPathStopJerk* parameter, and with the holding function it is defined via *lrPathJerk*.

If the *xPathStop* and *xPathHalt* inputs are reset to FALSE, the axes are brought to a standstill. Then the technology module is set to the "Ready" state.

From the "Ready" state, the travel process can be continued again with a new FALSE↗TRUE edge at the *xExecutePickAndPlace* input.

If the travel profile is not to be continued, it has to be aborted with the input *xAbort* = TRUE in the "Ready" state of the "STOP" function. Aborting the travel profile is only possible when the axes are at a standstill.

If the technology module is in the "STOP" state and the travel profile has not been aborted yet, the axes can be traversed using the *xJogPos* or *xJogNeg* jog input. This may for instance be required for running the kinematics to a service position.

▸ Manual jog (jogging) (📖 36)

Stop!

The holding position must be saved after the stop. After Manual jog (jogging) (📖 36), all axes must be traversed to the holding positions. This is the application programmer's task; the technology module does not carry out this process automatically!

If this does not happen, a movement to the next target point takes place, which may cause unintended movements in the workspace.

xStopALL input

With input *xStopALL* = TRUE, all axes are brought to a standstill individually – irrespectively of the path – with the deceleration defined by the *alrStopDec* parameter.

The jerk limitation is defined via the *alrStopJerk* parameter.

The technology module changes to the "STOP" state (display at the *eTMState* output).

When the movement has been started again, the path is executed again from the start.

Parameters to be set

The parameters to be set are located in the L_TT1P_scPar_PickAndPlace[Base/State/High] (📖 26) parameter structure.

```
lrPathStopDec : LREAL := 10000;          // [units/s^2]
lrPathStopJerk : LREAL := 100000;         // [units/s^3]
lrPathHaltDec : LREAL := 10000           // [units/s^2]
lrPathJerk : LREAL := 100000;            // [units/s^3]
alrStopDec : ARRAY OF LREAL := 10000;    // [units/s^2]
alrStopJerk : ARRAY OF LREAL := 100000;  // [units/s^3]
```

3.15 Interrupting the travel profile/path (xPathInterrupt)

This function can only be activated in the "MOVE_PP" state.

With the input *xPathInterrupt* = TRUE, all axes are brought to a standstill with the deceleration defined via the *lrPathStopDec* parameter, so that they are accurate to the path (following the travel profile). This state is shown by output *xPathHold* = TRUE.

The technology module changes to the "PATH_INTERRUPT" state (display at the *eTMState* output).

By using *xPathInterrupt*, the automatic gripper control for the current path is interrupted.

- The *xGripperCtrlActive* gripper output is set to FALSE.
- The *xPulseGripper* signal is only set at the end of the profile.

If *xPathInterrupt* is set to FALSE, execution of the path is continued from its breakpoint.

If the travel profile is not to be continued, it has to be aborted by input *xAbort* = TRUE. Aborting the travel profile is only possible when the axes are at a standstill.

If the technology module is in the "STOP" state and the travel profile has not been aborted yet, the axes can be traversed using the *xJogPos* or *xJogNeg* jog input. This may for instance be required for running the kinematics to a service position.

▸ Manual jog (jogging) (📖 36)

Stop!

The holding position must be saved after the stop. After Manual jog (jogging) (📖 36), all axes must be traversed to the holding positions. This is the application programmer's task; the technology module does not carry out this process automatically!

If this does not happen, a movement to the next target point takes place, which may cause unintended movements in the workspace.

3.16 Gripper control

The gripper control is activated if a non-zero value is set in the *lrGripperClosingTime* parameter. This is also shown by output *xGripperCtrlActive* = TRUE.

The x*xPulseGripper* output is controlled depending on the *lrGripperClosingTime* parameter:

- Positive values have the effect that the gripper is opened by the value set in seconds before the profile end is reached.
- Negative values have the effect that the gripper is opened by the value set in seconds after the profile end has been reached.

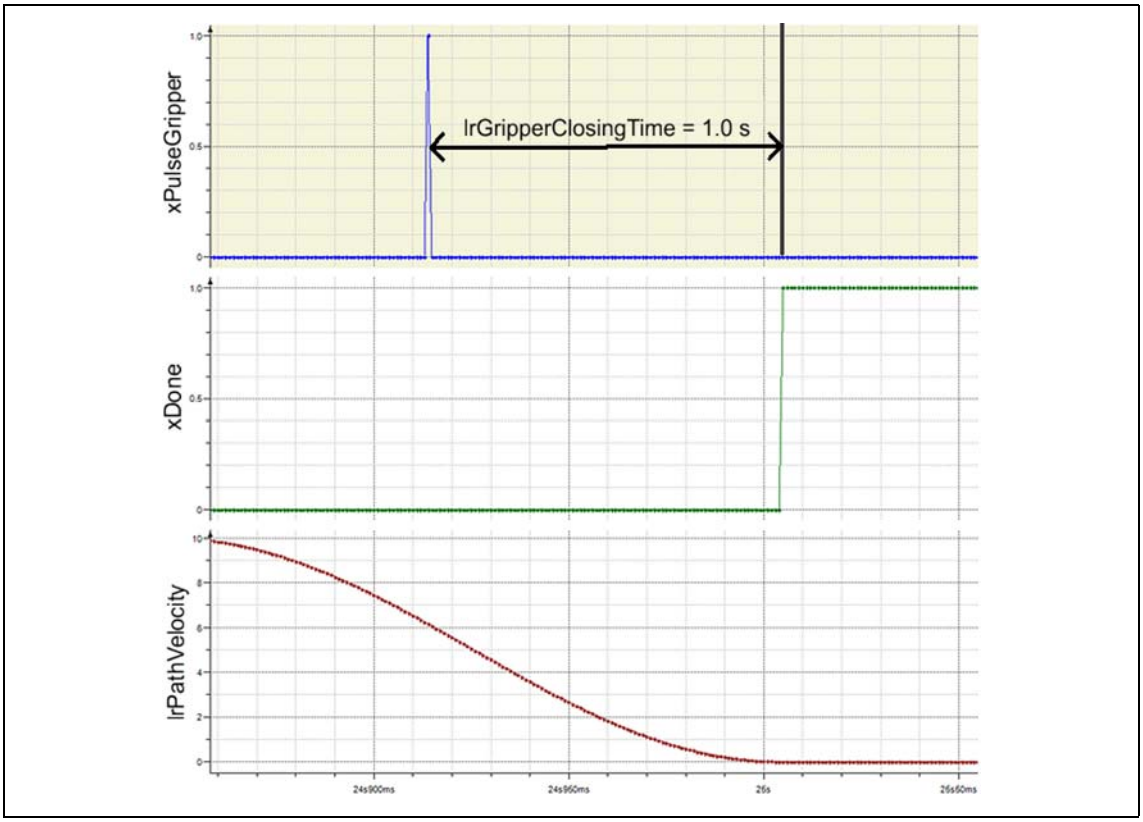
The *xPulseGripper* output is only active for one cycle.

The gripper control only works for as long as the travel process is not interrupted by a stop. Otherwise gripper control is deactivated for the path.

If a value not equalling '1.0' is set in the *lrOverride* input, gripper control is also deactivated. The Speed override (📖 50) influences the travelling duration of the path; therefore it is not possible to implement a correct gripper control.

Deactivation of the gripper control is shown by output *xGripperCtrlActive* = FALSE.

Figure [3-8] shows the impact of the *lrGripperClosingTime* parameter from the L_TT1P_scPar_PickAndPlace[Base/State/High] (📖 26) parameter structure on the *xPulseGripper* output.

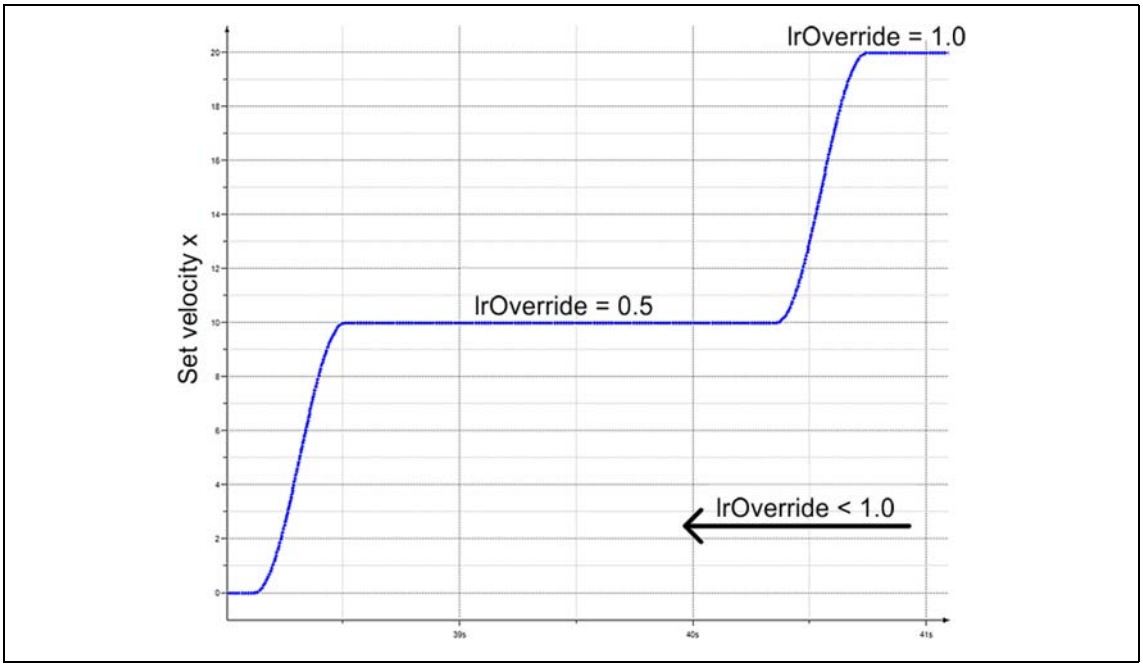


[3-8] Impact of the lrGripperClosingTime parameter on the xPulseGripper output

3.17 Speed override

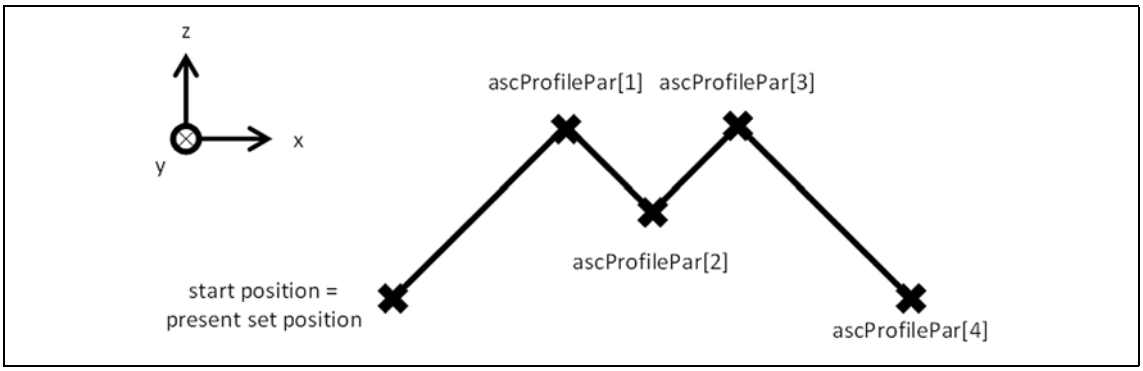
Via the *lrOverride* input, the interpolation of the path can be influenced. The speed, acceleration and jerk are multiplied by the override factor set. The initial value of the input is '1.0'. A value not equalling '1.0' or a change in value during the travel process deactivates the Gripper control (📖 49).

The impact of *lrOverride* on the speed is shown in figure [3-9]. The value '0.5' halves the speed, the acceleration, and the jerk.



[3-9] Impact of lrOverride on the speed

3.18 Defining a travel profile with up to 10 interpolation points



[3-10] Example: Path planning in the State version

Within the travel profile, maximally 10 interpolation points can be freely defined and parameterised via the *ascProfilePar* parameter in the L_TT1P_scPar_PickAndPlace[Base/State/High] (📖 26) parameter structure.

The parameters to be set for the interpolation points are located in the L_TT1P_scProfilePar (📖 32) parameter structure.

For each interpolation point, a coordinate system can be defined via the *eCoordSystem* parameter. The "Machine Coordinate System" (MCS) and all "Product Coordinate Systems" (PCS 1...16) are supported. The "Axes Coordinate System" (ACS) is not supported.

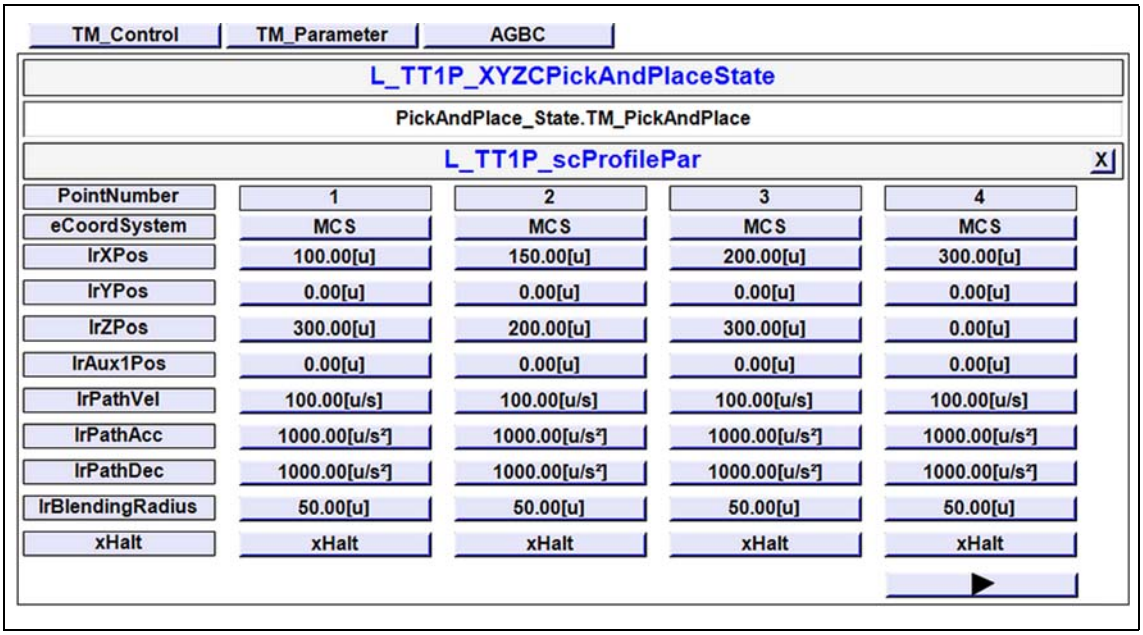
For a travel profile with less than 10 interpolation points, the speed selections for the points that are not available must be set to the value '0' in the *lrPathVel* parameter in each case.

If the *lrPathVel* parameter is 0 for one interpolation point and *lrPathVel* > 0 for the following interpolation point, the parameter set with *lrPathVel* = 0 is not taken into consideration. Therefore parameter sets are easily masked out.

For parameter sets with valid *lrPathVel* values (> 0) and the parameters *lrPathAcc* = 0 or *lrPathDec* = 0, always the last valid acceleration value (> 0) in the profile is used.

If, for instance, always the same acceleration is to be used in a travel profile, *lrPathAcc* > 0 and *lrPathDec* > 0 must only be set for the first interpolation point in the profile.

If *lrPathAcc* = 0 and *lrPathDec* = 0 are set for the first interpolation point, the standard acceleration *lrPathAcc* = *lrPathDec* = 10 units/s^2 is used, which may cause the movement to take a very long time.



PointNumber	1	2	3	4
eCoordSystem	MCS	MCS	MCS	MCS
lrXPos	100.00[u]	150.00[u]	200.00[u]	300.00[u]
lrYPos	0.00[u]	0.00[u]	0.00[u]	0.00[u]
lrZPos	300.00[u]	200.00[u]	300.00[u]	0.00[u]
lrAux1Pos	0.00[u]	0.00[u]	0.00[u]	0.00[u]
lrPathVel	100.00[u/s]	100.00[u/s]	100.00[u/s]	100.00[u/s]
lrPathAcc	1000.00[u/s²]	1000.00[u/s²]	1000.00[u/s²]	1000.00[u/s²]
lrPathDec	1000.00[u/s²]	1000.00[u/s²]	1000.00[u/s²]	1000.00[u/s²]
lrBlendingRadius	50.00[u]	50.00[u]	50.00[u]	50.00[u]
xHalt	xHalt	xHalt	xHalt	xHalt

[3-11] Example: Parameters of interpolation points 1 ... 4 of a travel profile

3.19 Movement of the orientation axes and the auxiliary axis across several interpolation points

The orientation axes A, B, C and the Aux1 auxiliary axis can be moved across several interpolation points.

Example of the auxiliary axis (Aux1)

If the auxiliary axis is to be rounded starting from interpolation point 1 to interpolation point 5, the value ’999’ must be set as position value for "" in interpolation points 2, 3, and 4. Now the technology module knows that it always has to search for the next valid target (i.e. the next valid target point for "Aux1"). For interpolation point 5, a valid value must be entered in this example.

Irrespective of the control of main axes X, Y and Z, the auxiliary axis always stops at a a point if ...

- a target point (entries for Aux1 not equalling ’999’) is defined,
- a breakpoint (*xHalt* = TRUE) has been programmed, or
- the single step function (*xSingleStep* = TRUE) is activated.

Restriction for orientation axes A, B, C

For the orientation axes, the value ’999’ must not be used if the selection of the *eCoordSystem* coordinate system (in parameter structure L_TT1P_scProfilePar (📖 32)) changes in the points in which the orientation axes are rounded.

Differences in rounding between the orientation axes and the auxiliary axis

The differences in rounding with the value ’999’ between the orientation axes A, B and C and the Aux1 auxiliary axis are:

	Orientation axes A, B, C	Aux1 auxiliary axis
Coupling to the path in X/Y/Z direction	Path-based: The speed, acceleration and deceleration for the A/B/C axes depend on the path. This means that the speed, acceleration and deceleration can change between the rounding points.	Time-based: The auxiliary axis is executed optimally in the rounding points without acceleration and deceleration phases.
Selection of the coordinate system	Each target position must be programmed when the coordinate system is changed. The rounding coding with the value ’999’ is impermissible if different coordinate systems are selected !	Programming of the auxiliary axis does not depend on the selection of the coordinate system. The rounding coding with the value ’999’ may be used if different coordinate systems are selected.
Definition of the target positions in the interpolation points with a blending radius > 0	The orientation axes are rounded in the interpolation points.	The auxiliary axis is stopped in the interpolation points. The path is also stopped in X/Y/Z direction.
Programming of a breakpoint (xHalt = TRUE) and Pre-acknowledgement (📖 54) during the path interpolation		

3.20 Defining optional breakpoints on the travel profile

Basically, a breakpoint can be set at every interpolation point (*xHalt* = TRUE). Each of these breakpoints programmed has to be acknowledged.

There are two possibilities of acknowledgement:

- Acknowledgement as a real breakpoint
- Pre-acknowledgement

Acknowledgement as a real breakpoint

The axes reach the point programmed and come to a stop. An acknowledgement signal at the *xAckPathHalt* input is waited for. When the acknowledgement process has been carried out successfully, the next point is approached.

Pre-acknowledgement

Breakpoints on the path can be designed as optional waiting points, meaning that when the travel profile is started, it is not clear yet whether the target can be approached. If the target position up to a specific point on the path is not free yet (free storage area or conveying belt), a stop must be made at the breakpoint programmed. If, however, the storage area gets free while the profile is executed, the breakpoint can be pre-acknowledged and the breakpoint is overtravelled. The axes do not reach a standstill. If a smoothening process is programmed, the path is smoothed and executed without jerk.

Note!

The pre-acknowledgement signal (input *xAckPathHalt*) must be set at an early stage.

If acknowledgement is executed too late it may happen that the pre-acknowledgement is used for the next but one breakpoint. The breakpoint approached then has to be acknowledged nevertheless.

Examples: Breakpoint is set in interpolation point 5.

The pre-acknowledgement signal must be received before interpolation point 4. If the technology module is already in the path segment between point 4 and point 5, smoothening cannot take place anymore in the case of a pre-acknowledgement.

It is also possible to send several pre-acknowledgement signals in succession. In this case, as many breakpoints are acknowledged as there were FALSE↗TRUE edges at the *xAckPathHalt* input.

Single step function

The travel profile can also be executed in single step mode, inserting a breakpoint at each interpolation point. This has to be set before starting the travel profile using the parameter *xSingleStep* = TRUE of the L_TT1P_scPar_PickAndPlace[Base/State/High] (📖 26) parameter structure.

Acknowledgement of the breakpoint after reaching the interpolation point is effected by means of input *xAckPathHalt* = TRUE. Pre-acknowledgement can also be carried out here.

3.21 Adding more travel profiles

Precondition

- All axes are enabled (*xRegulatorOnALL* input = TRUE).
- A travel profile with maximally 10 interpolation points has been defined. Each point may contain an optional hold.
  ▸ Defining a travel profile with up to 10 interpolation points (📖 51)
- Changing the profiles that have been accepted in the technology module is not possible.

Execution

With a rising edge (FALSE↗TRUE) at the *xExecutePickAndPlace* input, the profile is started.

The output *xProfileInPreparation* = TRUE reflects the status that the technology module is calculating and preparing the profile (the path) in the background.

With the output *xProfileInPreparation* = FALSE, the preparation of the profile is completed and the next profile can be appended. For this purpose, the profile points can be newly defined in the *ascProfilePar* parameter.

With a rising edge (FALSE↗TRUE) at the *xAddProfile* input, the profile is appended from the *ascProfilePar* parameter. The output *xProfileInPreparation* = TRUE is set.

The profiles can be influenced via the inputs *xPathStop*, *xPathHalt*, *xPathInterrupt* or *lrOverride*, irrespective of the number of profiles loaded.

3.22 CPU utilisation (example Controller 3231 C)

The following table shows the CPU utilisation in microseconds using the example of the 3231 C controller (ATOM™ processor, 1.6 GHz).

Versions	Interconnection of the technology module	CPU utilisation	
		Average	Maximum peak

Base	xEnable := TRUE;
	xRegulatorOn := TRUE;
	xExecutePickAndPlace := TRUE;
		200 µs	511 µs

State	xEnable := TRUE;
	xRegulatorOn := TRUE;
	xExecutePickAndPlace := TRUE;
		230 µs	565 µs

High	xEnable := TRUE;
	xRegulatorOn := TRUE;
	xExecutePickAndPlace := TRUE;
		230 µs	565 µs

Note!

The path preparation is processed over several task cycles. The duration of the path preparation process depends on the task utilisation of the CPU. The time between the activation of the path travel process and the execution of the path travel process can be delayed by up to 25 cycles.

Index

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.

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

Lenze Automation GmbH
Postfach 10 13 52, 31763 Hameln
Hans-Lenze-Straße 1, 31855 Aerzen
GERMANY
HR Hannover B 205381
+49 5154 82-0
+49 5154 82-2800
lenze@lenze.com
www.lenze.com

Service
Lenze Service GmbH
Breslauer Straße 3, 32699 Extertal
GERMANY
008000 24 46877 (24 h helpline)
+49 5154 82-1112
service@lenze.com