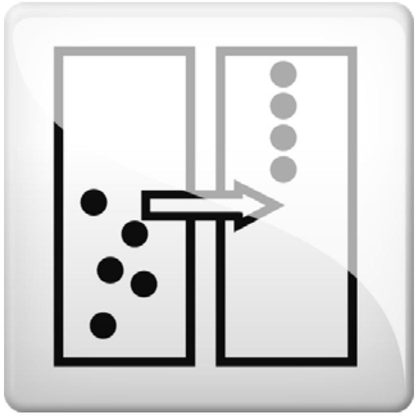
FAST Application Software



Technology module

Track Pick & Place

Reference Manual EN

13531748

Lenze

Contents

1 About this documentation

This documentation ...

- contains detailed information on the functionalities of the "Track 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
2.2	05/2017	TD17	• Content structure has been changed. • General revisions

2.1	11/2016	TD17	• General revision
			• Parameter L_TT1P_scPar_TrackPickAndPlaceBase (📖 29) supplemented.
			• Supplements in the chapter:
			  • Calculation of the profile points with lrBlendingRadius = 0 (📖 42)
			  • Calculation of the profile points with lrBlendingRadius > 0 (📖 48)

2.0	04/2016	TD17	• General editorial revision • Modularisation of the contents for the »PLC Designer« online help
1.0	02/2016	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	(🕮 7)	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	**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

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 "Track Pick & Place"

[3-1] Typical mechanics of the technology module

"Pick & Place" mechanics/kinematics are used in various industry sectors. Classic application areas can be found, for instance, in the packaging technology.

A typical task of the "Track Pick & Place" technology module is the control of gripper to pick up a product or workpiece from a conveying belt and put it onto a different position or conveying belt.

In addition to simple point-to-point positioning processes, the technology module can also synchronise to positions on moving conveying belts.

The workpieces are detected and transferred (path planning) in Cartesian space and are thus kinematics-independent.

The technology module provides a ready-made management system for two conveying belts via adjustable modes.

Maximally 16 conveying belts are supported by the technology module. That way, up to 30 workpieces can be followed on each conveying belt via a corresponding guidance system.

The workspaces on the single conveying belts are defined via parameter setting.

▸ Overview of the functions (📖 12)

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:

- Manual jog (jogging) (📖 38)
- Homing (📖 40)
- Specifying a travel profile (📖 41)
- Calculation of the profile points with lrBlendingRadius = 0 (📖 42)
- Calculation of the profile points with lrBlendingRadius > 0 (📖 48)
- Limitations (maximum values) for the axes on the travel profile (📖 54)
- Starting the travel profile (xExecutePickAndPlace) (📖 56)
- Adding more travel profiles (📖 57)
- Stop/holding function (xPathStop, xStopALL, xPathHalt) (📖 58)
- Interrupting the travel profile/path (xPathInterrupt) (📖 59)
- Gripper control (📖 60)
- Selecting the gripper control mode (📖 61)
- Speed override (📖 63)
- Parameter setting of the conveying belts (📖 64)
- Detection of the workpieces (📖 68)
- Calculation of the path (contour) to a workpiece (📖 69)

»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) (📖 38)**:**

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 "Track 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.

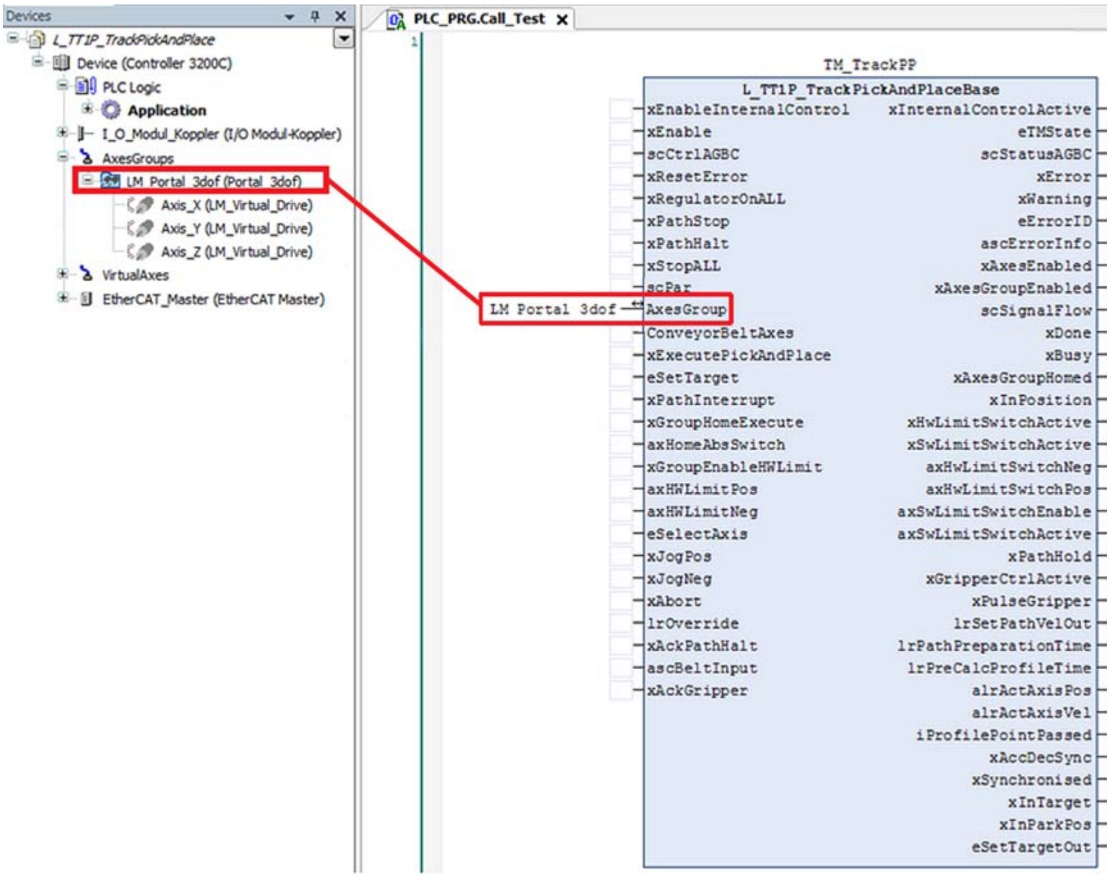
The axes group must be inserted in the device tree. Via the designation, the axes group is connected to the *AxesGroup* input of the technology module.

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.

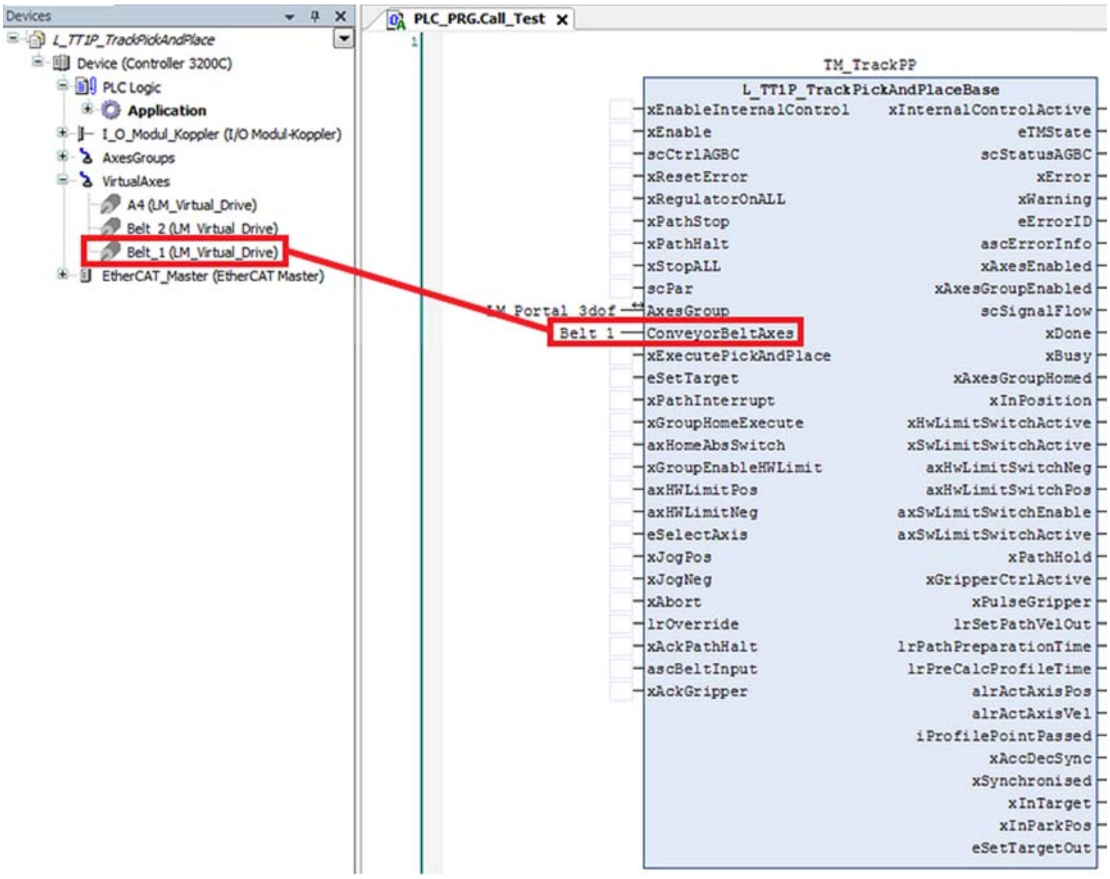
All functional parameters and the parameters for the setpoint generation are only set in one central position via the L_TT1P_scPar_TrackPickAndPlaceBase (📖 29) parameter structure at the *scPar* input of the technology module. The kinematic parameters and the limitations of the individual axes must be set in the axes group.



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

3.4 Interconnection of the technology module with a conveying belt

The axis reference of the conveying belt is directly interconnected to the *ConveyorBeltAxes* input of the technology module. The technology module has a read access to the axis reference of the conveying belt.



[3-3] Example: Interconnecting the technology module with a conveying belt

3.5 Interconnection of the technology module with several conveying belts

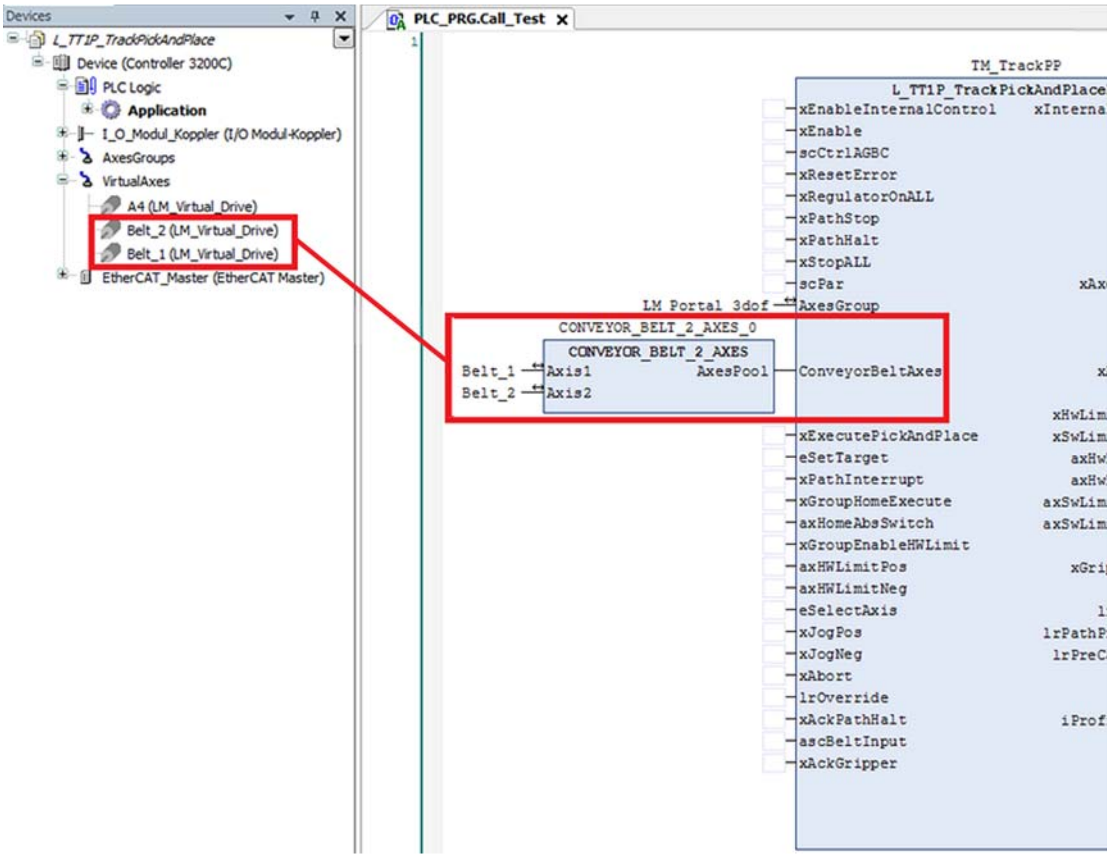
The axis references of the conveying belts are interconnected with the *ConveyorBeltAxes* input of the technology module via the **CONVEYOR_BELT_n_AXES** function block.

The **CONVEYOR_BELT_n_AXES** function block provides the opportunity to transfer several axis references to the technology module. "n" stands for the number of axes (2 to 16).

The technology module has a read access to the axis references of the conveying belts.

Example

Interconnection of two conveying belts via the **CONVEYOR_BELT_2_AXES** function block:



[3-4] Example: Interconnecting the technology module with two conveying belts

3.6 L_TT1P_TrackPickAndPlaceBase function block

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

The base version offers the full functionality of the technology module.

L_TT1P_TrackPickAndPlaceBase

Input type	Input	Output	Output type
BOOL	xEnableInternalControl	xInternalControlActive	BOOL
BOOL	xEnable	eTMState	L_TT1P_States
scCtrl_AGBC	scCtrlAGBC	scStatusAGBC	scStatus_AGBC
BOOL	xResetError	xError	BOOL
BOOL	xRegulatorOnALL	xWarning	BOOL
BOOL	xPathStop	eErrorID	L_IE1P_Error
BOOL	xPathHalt	ascErrorInfo	ARRAY [1..18] OF L_TT1P_scErrorInfo
BOOL	xStopALL	xAxesEnabled	BOOL
L_TT1P_scPar_TrackPickAndPlaceBase	scPar	xAxesGroupEnabled	BOOL
AXES_GROUP_KIN	AxesGroup	scSignalFlow	L_TT1P_scSF_TrackPickAndPlaceBase
INTERFACE	ConveyorBeltAxes	xDone	BOOL
BOOL	xGroupHomeExecute	xBusy	BOOL
ARRAY [1..12] OF BOOL	axHomeAbsSwitch	xAxesGroupHomed	BOOL
BOOL	xGroupEnableHWLimit	xInPosition	BOOL
ARRAY [1..18] OF BOOL	axHwLimitPos	xHwLimitSwitchActive	BOOL
ARRAY [1..18] OF BOOL	axHwLimitNeg	xSwLimitSwitchActive	BOOL
IDENT_IN_GROUP_REF	eSelectAxis	axHwLimitSwitchNeg	ARRAY [1..18] OF BOOL
BOOL	xJogPos	axHwLimitSwitchPos	ARRAY [1..18] OF BOOL
BOOL	xJogNeg	axSwLimitSwitchEnable	ARRAY [1..18] OF BOOL
BOOL	xExecutePickAndPlace	axSwLimitSwitchActive	ARRAY [1..18] OF BOOL
ENUM	eSetTarget	xPathHold	BOOL
BOOL	xPathInterrupt	xGripperCtrlActive	BOOL
BOOL	xAbort	xPulseGripper	BOOL
LREAL	lrOverride	lrSetPathVelOut	LREAL
ARRAY [1..16] OF L_TT1P_TrackConveyorBelt	ascBeltInput	lrPathPreparationTime	LREAL
BOOL	xAckGripper	lrPreCalcProfileTime	LREAL
		alrActAxisPos	ARRAY [1..18] OF LREAL
		alrActAxisVel	ARRAY [1..18] OF LREAL
		iProfilePointPassed	INT
		xProfileInPreparation	BOOL
		xAccDecSync	BOOL
		xSynchronised	BOOL
		xInTarget	BOOL
		xInParkPos	BOOL
		eSetTargetOut	ENUM

3.6.1 Inputs and outputs

Designator Data type	Description
AxesGroup AXES_GROUP_KIN	Reference to the axes group ▸ Interconnection of the technology module with the axes group (📖 14)

3.6.2 Inputs

Designator Data type	Description	
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 lrPathStopDec. • 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) (📖 58)
	FALSE	The state changes to "READY". Further instructions are expected.
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.

Designator Data type	Description	
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) (📖 58)
scPar L_TT1P_scPar_TrackPickAndPlaceBase	The parameter structure contains the parameters of the technology module.	
ConveyorBeltAxes INTERFACE	Interface for: • Reference axis **AXIS_REF** (read access) for the operation with a conveying belt. • Reference to the **CONVEYOR_BELT_2_AXES** function block (read access) for the operation with two conveying belts. • Reference to the **CONVEYOR_BELT_3_AXES** function block (read access) for the operation with three conveying belts. This input must be activated cyclically with the call of the technology module. **Example of a TM call in the Motion task:** `TM_TrackPickAndPlace(AxesGroup := AxesGroup, ConveyorBeltAxes := LM_Drive_Belt);` ▶ Interconnection of the technology module with a conveying belt (📖 15) ▶ Interconnection of the technology module with several conveying belts (📖 16)	
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_TrackPickAndPlaceBase (📖 29) 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	
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	
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). When **Mode eTrackingMode = 0** the target from the eSetTarget input is transferred to the technology module. Several targets can be loaded before the FALSE↗TRUE edge. Preconditions for loading the profile: • The axes are enabled. • The resources of the technology module for path planning are enabled for the next profile (output xProfileInPreparation = FALSE).
eSetTarget ENUM	Relevant in the **Mode eTrackingMode = 0**: The target is accepted with the FALSE↗TRUE edge at the xExecutePickAndPlace input.	
	0	Travel to the static position (lrXTargetPos, lrYTargetPos, lrZTargetPos, lrCTargetPos, lrAux1TargetPos)
	1	CV1: Synchronisation to the workpiece on the conveying belt 1
	2	CV2: Synchronisation to the workpiece on the conveying belt 2
	3	CV3: Synchronisation to the workpiece on the conveying belt 3
	4	CV4: Synchronisation to the workpiece on the conveying belt 4
	5	CV5: Synchronisation to the workpiece on the conveying belt 5

Designator Data type	Description	
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.
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.	
ascBeltInput ARRAY[1..16] OF L_TT1P_TrackConveyorBelt	Connection to the conveying belt (CV1 ... CV16) • ascBeltInput[1] = conveying belt 1 (CV1) ... • ascBeltInput[16] = conveying belt 16 (CV16) ▸ Input structure ascBeltInput[i] (📖 23)	
xAckGripper BOOL	The input is edge-controlled and evaluates the rising edge.	
	FALSE↗TRUE	The gripper is acknowledged and the synchronous travel of the gripper above the conveying belt is aborted. In the process, the lifting motion is executed immediately, independent of the time remaining for the synchronous travel (parameter ascConveyorBeltPar[i].lrSyncTimeOnBelt).

3.6.3 Input structure ascBeltInput[i]

L_TT1P_TrackConveyorBelt

The *ascBeltInput[1..16]* input structure serves for Detection of the workpieces (📖 64).

Up to 30 workpieces can be detected, tracked and processed in parallel on each belt conveyor.

Designator / **Data type**	**Description**	
xWorkpieceOnBeltReceive BOOL	The input is edge-controlled and evaluates the rising edge.	
	FALSE↗TRUE	Tracking the workpiece on the respective conveying belt is started.
xValidPos BOOL	The input is edge-controlled and evaluates the rising edge.	
	FALSE↗TRUE	Delayed setting of the product coordinate system (PCS): The coordinates of the detected workpiece which were valid at the time of the FALSE↗TRUE edge at the xWorkpieceOnBeltReceive input are accepted and included in the calculation of the workpiece tracking.
ascInitialObjectPosition ARRAY [1..6] OF LREAL	Position and alignment of the dynamic product coordinate system (PCS) or the workpiece on the corresponding conveying belt at the time of the FALSE↗TRUE edge at the xWorkpieceOnBeltReceive input • The position must be given in the coordinate system of the conveying belt. • The position is accepted with a FALSE↗TRUE edge at the xValidPos input in the technology module.	
lrAux1TargetPos LREAL	Target position for the "Pick & Place" profile on the conveying belt in Aux1 direction • Unit: units	

3.6.4 Outputs

Designator Data type	Description	
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 (📖 37)	
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_TrackPickAndPlaceBase	Structure of the signal flow	
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.
xHwLimitSwitchActive BOOL	TRUE	At least one axis has reached or approached a hardware limit switch. • The axHWLimitPos or axHwLimitNeg input is also set to TRUE. • The drive is braked to a standstill with the deceleration set in the alrStopDec parameter. • The state changes to "ERROR" with the error message '20500' (HWLimitPos).

Designator Data type	Description	
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 alrStopDec 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’
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’

Designator Data type	Description	
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’
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	

Designator Data type	Description	
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'
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.	
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 added.
xAccDecSync BOOL	TRUE	The synchronisation function is active. The axis is synchronised or desynchronised (clutch opens or closes).
xSynchronised BOOL	TRUE	The axis is synchronised to the conveying belt.
xInTarget BOOL	TRUE	The target position (static position or position on the conveying belt) has been reached.
xInParkPos BOOL	TRUE	The parking position has been reached.

<table>
<tr><th>Designator
Data type</th><th colspan="2">Description</th></tr>
<tr><td rowspan="7">eSetTargetOut
ENUM</td><td colspan="2">The target position to be approached</td></tr>
<tr><td>0</td><td>The target position is the static position
(lrXTargetPos, lrYTargetPos, lrZTargetPos, lrCTargetPos, lrAux1TargetPos)</td></tr>
<tr><td>1</td><td>CV1: The target position is the workpiece on the conveying belt 1</td></tr>
<tr><td>2</td><td>CV2: The target position is the workpiece on the conveying belt 2</td></tr>
<tr><td>3</td><td>CV3: The target position is the workpiece on the conveying belt 3</td></tr>
<tr><td>4</td><td>CV4: The target position is the workpiece on the conveying belt 4</td></tr>
<tr><td>5</td><td>CV5: The target position is the workpiece on the conveying belt 5</td></tr>
</table>

3.6.5 Parameters

L_TT1P_scPar_TrackPickAndPlaceBase

The **L_TT1P_scPar_PickAndPlaceBase** structure contains the parameters of the technology module.

Designator Data type	Description
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
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
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
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**
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.
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'
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

Designator Data type	Description
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
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

Designator Data type	Description
lrAux1MaxJerk LREAL	Maximum jerk for the Aux1 axis during the path interpolation • Unit: units/s^3 • Initial value: 10000
lrGripperClosingTime LREAL	Decelerated gripper control in the closing phase • Positive values: The gripper is triggered before the path profile is completed. • Unit: s
	Mode eTrackingMode = 0: • The target is approached in the eSetTarget input with a FALSE↗TRUE edge at the xExecutePickAndPlace input. • For decelerating the gripper control, the time in the lrGripperClosingTime parameter is accepted. • The lrGripperOpenTime parameter is not used.
	Mode eTrackingMode > 0: The technology module uses the lrGripperClosingTime parameter for closing the gripper and the lrGripperOpenTime parameter for opening the gripper.
lrGripperOpenTime LREAL	Decelerated gripper control in the opening phase • Positive values: The gripper is triggered before the path profile is completed. • Unit: s
	Mode eTrackingMode = 0: • The target is approached in the eSetTarget input with a FALSE↗TRUE edge at the xExecutePickAndPlace input. • For decelerating the gripper control, the time in the lrGripperClosingTime parameter is accepted. • The lrGripperOpenTime parameter is not used.
	Mode eTrackingMode > 0: The technology module uses the lrGripperClosingTime parameter for closing the gripper and the lrGripperOpenTime parameter for opening the gripper.
lrXTargetPos LREAL	Target position in X direction for the "Pick & Place" profile • Unit: units
lrYTargetPos LREAL	Target position in Y direction for the "Pick & Place" profile • Unit: units
lrZTargetPos LREAL	Target position in Z direction for the "Pick & Place" profile • Unit: units
lrATargetPos LREAL	Target position in A direction for the "Pick & Place" profile • Unit: units • Initial value: 180
lrBTargetPos LREAL	Target position in B direction for the "Pick & Place" profile • Unit: units
lrCTargetPos LREAL	Target position in C direction for the "Pick & Place" profile • Unit: units
lrAux1TargetPos LREAL	Target position in Aux1 direction for the "Pick & Place" profile • Unit: units
lrZStartDist LREAL	Difference in height at the start in Z direction • Unit: units

<table>
<tr><th>Designator
Data type</th><th colspan="2">Description</th></tr>
<tr><td rowspan="2">ePathMode
L_TT1P_PickAndPlacePathMode</td><td>0</td><td>Linear:
From the starting point to the target point, linear segments are used that are rounded among each other.</td></tr>
<tr><td>1</td><td>Elliptical:
From the starting point to the target point, an elliptical path profile with the Z height in parameter lrZDistElliptical is used.
An ellipse can only be executed if ...
• the starting point and the target point are <u>not</u> programmed in the axis coordinate system (ACS);
• the conveying belt coordinates are <u>not</u> rotated in A and B axes.
In the machine coordinate system (MCS), the static target position is only approached via an ellipse if the offset of the MCS compared to the world coordinate system (WCS) is not rotated in A and B direction.</td></tr>
<tr><td>lrZDistElliptical
LREAL</td><td colspan="2">Height of the ellipse from the starting point to the target point (only relevant for ePathMode = 1).</td></tr>
<tr><td>lrZTargetDist
LREAL</td><td colspan="2">Difference in height at the end/target in Z direction
• Unit: units</td></tr>
<tr><td>lrBlendingRadius
LREAL</td><td colspan="2">Blending radius for the "Pick & Place" profile
The value specifies the distance from the target point in which the rounding motion is to start.
• Unit: units</td></tr>
<tr><td rowspan="3">xPosInWindow
BOOL</td><td colspan="2">Activation of the tolerance window for following error monitoring
• Initial value: FALSE</td></tr>
<tr><td>TRUE</td><td>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.</td></tr>
<tr><td>FALSE</td><td>The tolerance window is not activated.</td></tr>
<tr><td>lrPosInWindow
LREAL</td><td colspan="2">Size of the tolerance window for following error monitoring
• Unit: units
• Initial value: 0.5</td></tr>
<tr><td>lrTimePosInWindow
LREAL</td><td colspan="2">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</td></tr>
</table>

Designator Data type	Description	
eTrackingMode L_TT1P_TrackPickAndPlaceMode	Mode for approaching the static positions/targets and for a synchronised placement of the workpiece ▸ Gripper control (📖 60)	
	0	For the **mode 0**, the robot handling has to be implemented outside the technology module. • The target is transferred to the technology module via the *eSetTarget* input and approached with a rising edge (FALSE↗TRUE) at the *xExecutePickAndPlace* input. • For decelerating the gripper control, the time in the lrGripperClosingTime parameter is accepted. • The lrGripperOpenTime parameter is not used. ▸ eTrackingMode = 0 (📖 61)
	In the **modes 1 ... 4**, the technology module controls the pick-up and placement of the workpieces by the gripper. In this process, the ... • lrGripperClosingTime parameter is used for closing the gripper; • and the lrGripperOpenTime parameter is used for opening the gripper.	
	1	Static --> CV1: The workpieces are picked up at the static position (parameters lrXTargetPos, lrYTargetPos, lrZTargetPos, lrCTargetPos, lrAux1TargetPos). The workpieces are placed synchronously on the conveying belt 1. ▸ eTrackingMode = 1 (Static => CV1) (📖 61)
	2	CV1 --> Static: The workpieces are picked up from the conveying belt 1. The workpieces are placed on the static position (parameters lrXTargetPos, lrYTargetPos, lrZTargetPos, lrCTargetPos, lrAux1TargetPos). ▸ eTrackingMode = 2 (CV1 => Static) (📖 62)
	3	CV1 --> CV2: The workpieces are picked up from the conveying belt 1. The workpieces are placed synchronously on the conveying belt 2. ▸ eTrackingMode = 3 (CV1 => CV2) (📖 62)
	4	CV2 --> CV1: The workpieces are picked up from the conveying belt 2. The workpieces are placed synchronously on the conveying belt 1. ▸ eTrackingMode = 4 (CV2 => CV1) (📖 62)
ascConveyorBeltPar ARRAY[1..16] OF L_TT1P_scPar_ConveyorBelt	The Array ascConveyorBeltPar[i] (📖 36) serves to parameterise up to 16 conveying belts (CV1 ... CV16): • ascConveyorBeltPar[1]: Conveying belt 1 (CV1) ... • ascConveyorBeltPar[16]: Conveying belt 16 (CV16) ▸ Parameter setting of the conveying belts (📖 64)	

3.6.6 Array ascConveyorBeltPar[i]

L_TT1P_scPar_ConveyorBelt

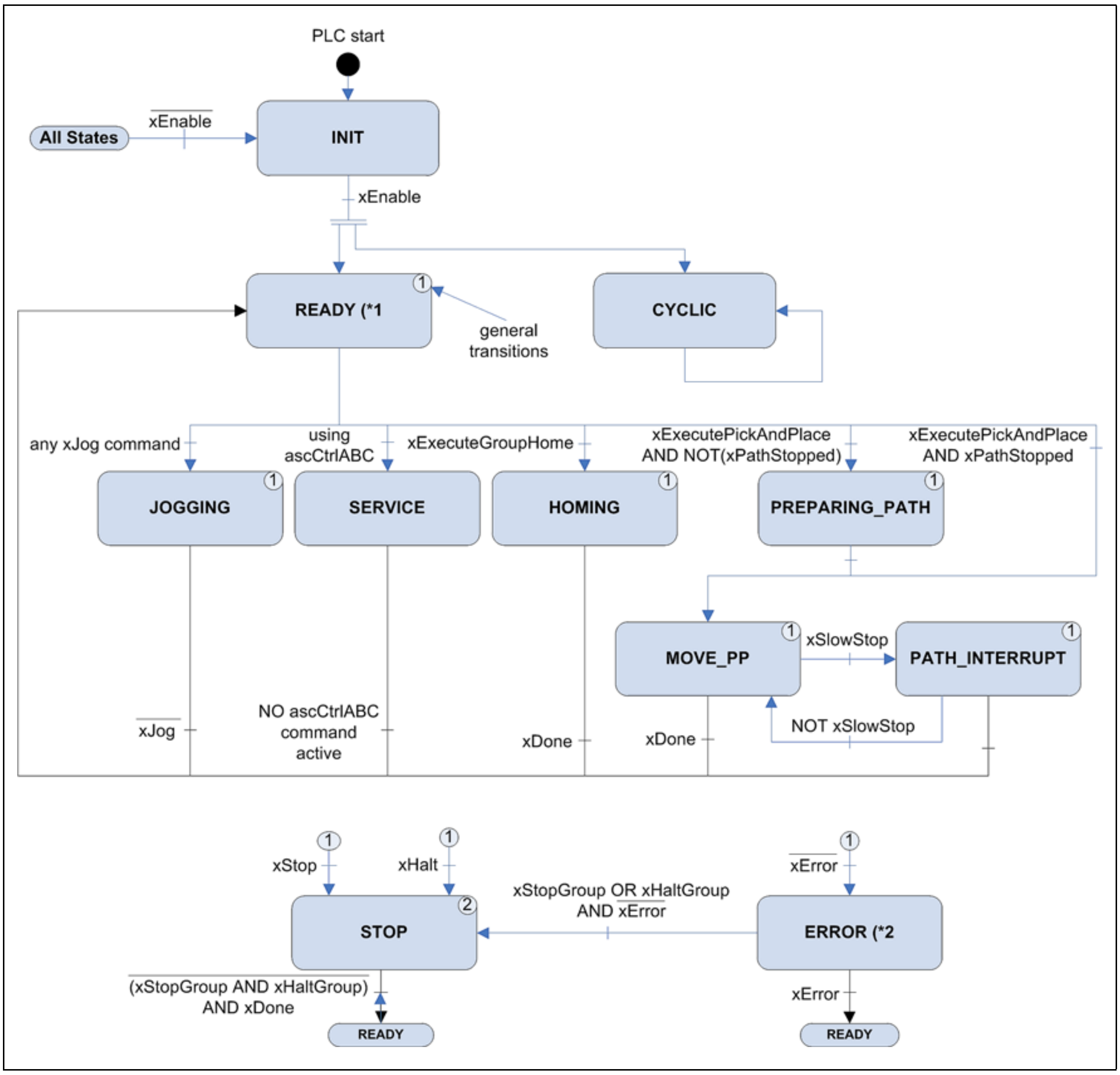
The *ascConveyorBeltPar[1..16]* array serves for ...

- Parameter setting of the conveying belts (📖 64);
- Calculation of the path (contour) to a workpiece (📖 69).

Up to 16 conveying belts (CV1 ... CV16) can be parameterised with the array.

Designator / **Data type**	**Description**
alrParkPos ARRAY [1..6] OF LREAL	Coordinates of the parking position in the machine coordinate system (MCS) for the conveying belt alrParkPos[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
lrZDistBelt LREAL	The height in Z direction in the product coordinate system (PCS) above the synchronous position of the workpiece on the conveying belt
alrConveyorBeltOrigin ARRAY [1..6] OF LREAL	Position and alignment of the conveying belt coordinate system from the point of view of the machine coordinate system (MCS) alrConveyorBeltOrigin[axis]: • [Axis] = 1 : X position in [units] • [Axis] = 2 : Y position in [units] • [Axis] = 3 : Z position in [units] • [Axis] = 4 : A alignment in [deg] • [Axis] = 5 : B alignment in [deg] • [Axis] = 6 : C alignment in [deg]
lrSyncTimeOnBelt LREAL	The duration of the synchronous travel of the "Tool Center Point" (TCP) above the conveying belt • Unit: s • Value '0': The TCP remains synchronously on the conveying belt for 1 clock cycle. **Note:** A FALSE↗TRUE edge at the *xAckGripper* input aborts the synchronous travel. In this process, the lifting motion is executed immediately, independent of the time remaining for the synchronous travel.
lrLowerSyncLimit LREAL	Start of the operating range above the conveying belt
lrUpperSyncLimit LREAL	End of the operating range above the conveying belt

3.7 State machine



[3-5] 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.

3.8 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 (📖 37) 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_TrackPickAndPlaceBase (📖 29) 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.

3.9 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 (📖 37) 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_TrackPickAndPlaceBase (📖 29) 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;
```

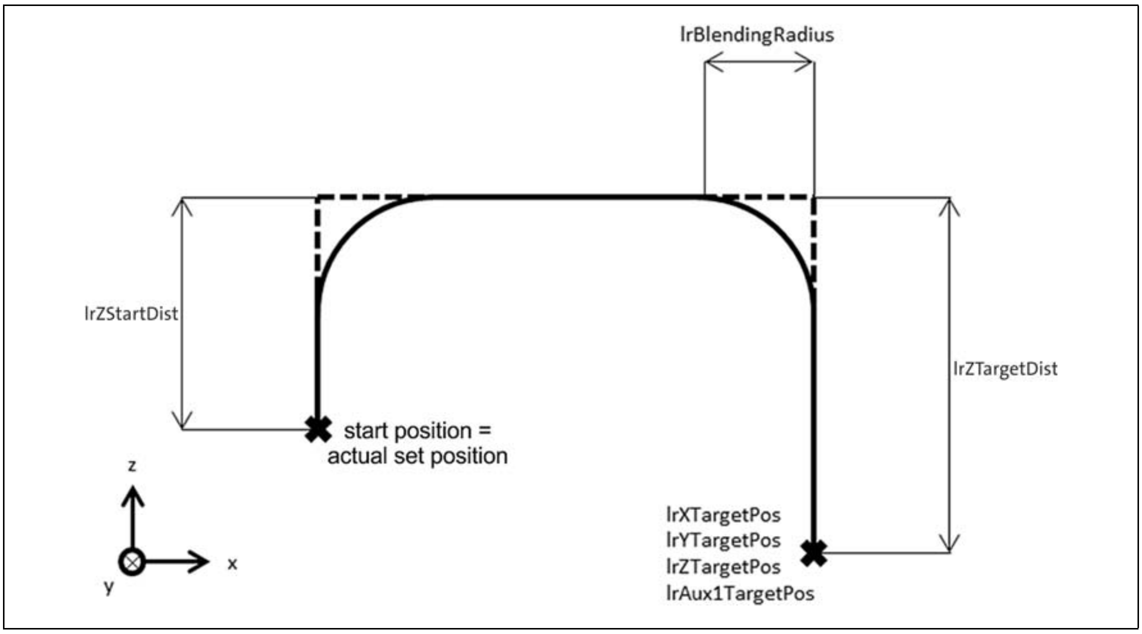
»PLC Designer« Online help

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

3.10 Specifying a travel profile

3.10.1 Linear profile (ePathMode = 0)

A simple "Pick & Place"- profile is defined based on a few parameters.



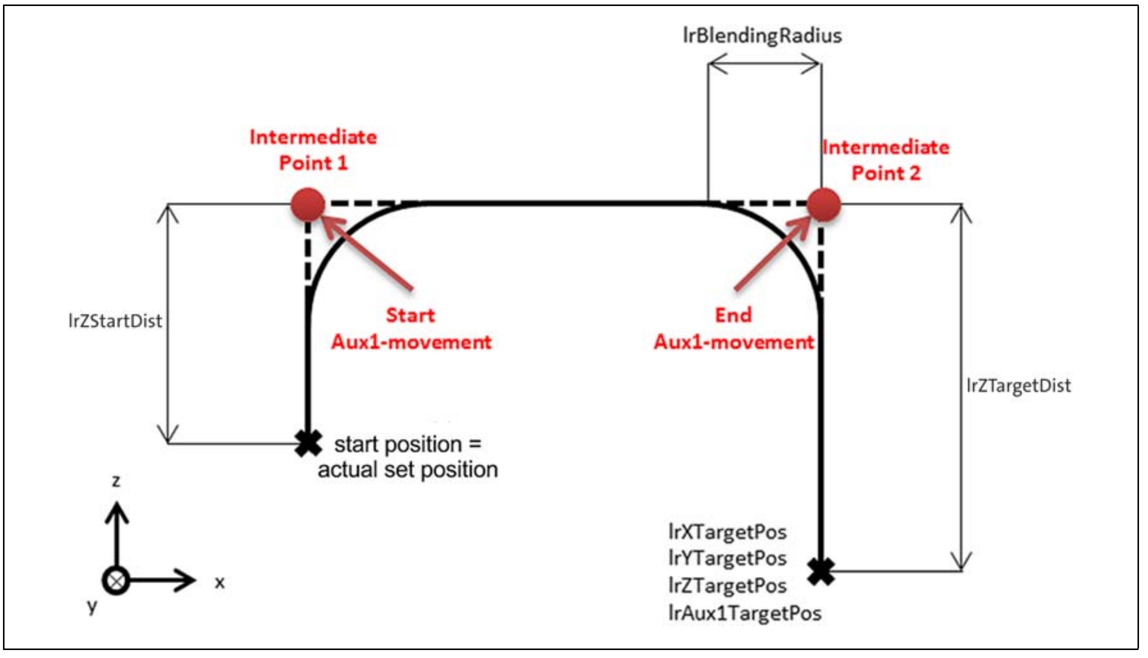
[3-6] Parameters for a simple "Pick & Place" profile

3.11 Calculation of the profile points with lrBlendingRadius = 0

3.11.1 Linear profile (ePathMode = 0)

Note!

The interpolation of the orientation axes A, B, C and the Aux1 auxiliary axis must be completed before the Z position on a conveying belt will be reached.



[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 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 axis 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 L_TT1P_scPar_TrackPickAndPlaceBase (📖 29) parameter structure.

```
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]
```

3.11.2 Ellipse profile (ePathMode = 1)

the ellipse is calculated if ...

- the starting point and the target point are not programmed in the axis coordinate system (ACS);
- the conveying belt coordinates are not rotated in A and B axes.

The contour of an ellipse is always calculated by 180° vertically towards the X-Y plane of the world coordinate system (WCS) of the Intermediate Points 1 and 2.

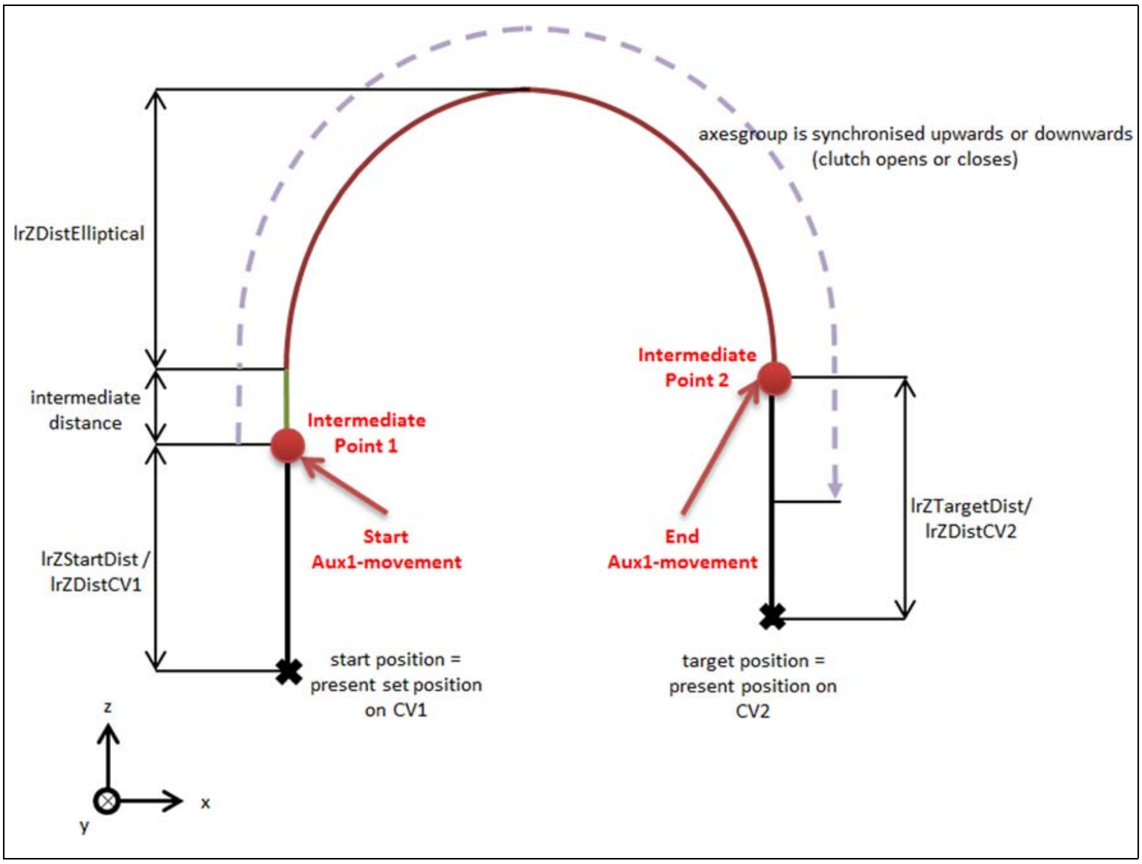
If the Intermediate Points 1 and 2 are on different Z heights, a linear intermediate distance is automatically inserted into the contour to prevent the ellipse profile from being rotated.

Note!

The interpolation of orientation axes A, B, C, and the Aux1 auxiliary axis must be completed before the height *lrZDistBelt* (in Array ascConveyorBeltPar[i] (📖 36)) is reached via the synchronous position of the workpiece on the belt conveyor.

Starting and target position

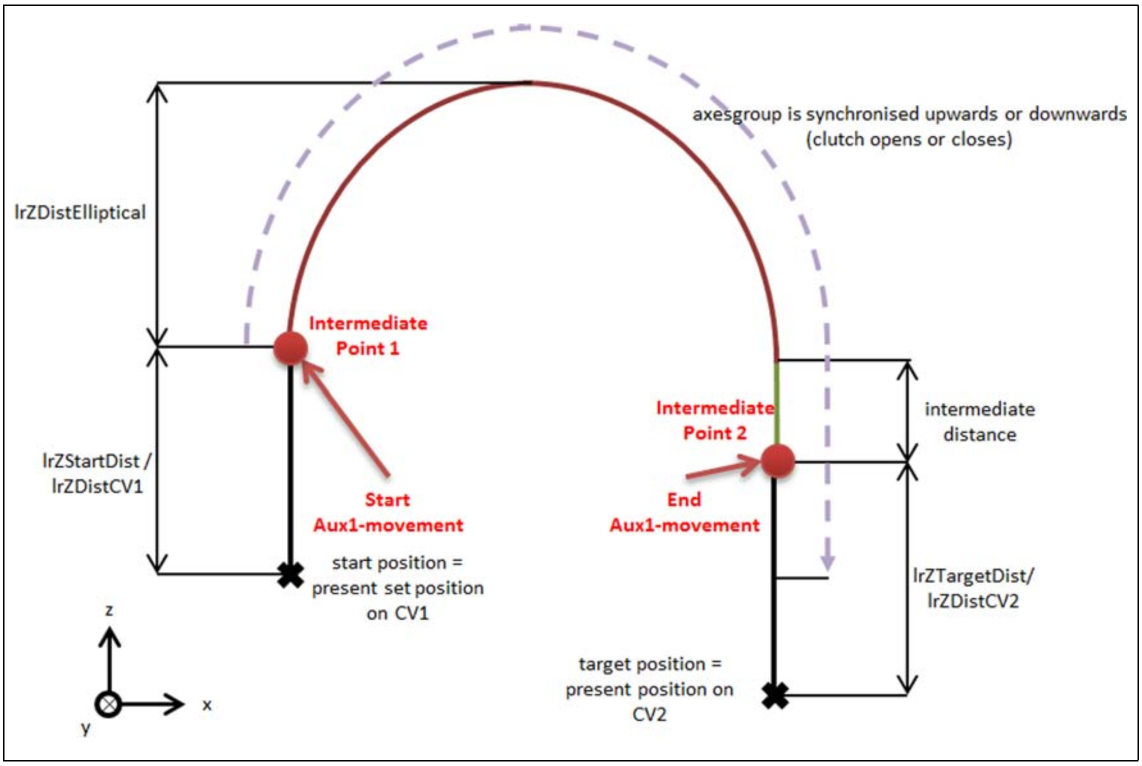
- The starting height *lrZStartDist* (*lrZDistBelt* on the conveying belt 1) and target height *lrZTargetDist* (*lrZDistBelt* on the conveying belt 2) are greater than zero:
  The Intermediate Points 1 and 2 between the starting point and the target point are approached. A short stop is made at the intermediate points (path speed = 0).
- The starting height *lrZStartDist* (*lrZDistBelt* on the conveying belt 1) is zero:
  The starting position is put onto the Intermediate Point 1.
- The target height *lrZTargetDist* (*lrZDistBelt* on the conveying belt 2) is zero:
  The target position is put onto the Intermediate Point 2.



[3-8] Example 1: Intermediate Point 1 is lower than Point 2

In the example 1, the Intermediate Point 1 is lower than the Intermediate Point 2 in Z direction of the world coordinate system (WCS). Thus, a linear intermediate distance is inserted after the Intermediate Point 1 and the starting position of the ellipse to compensate the height difference.

The dashed line shows the path with activated clutch.



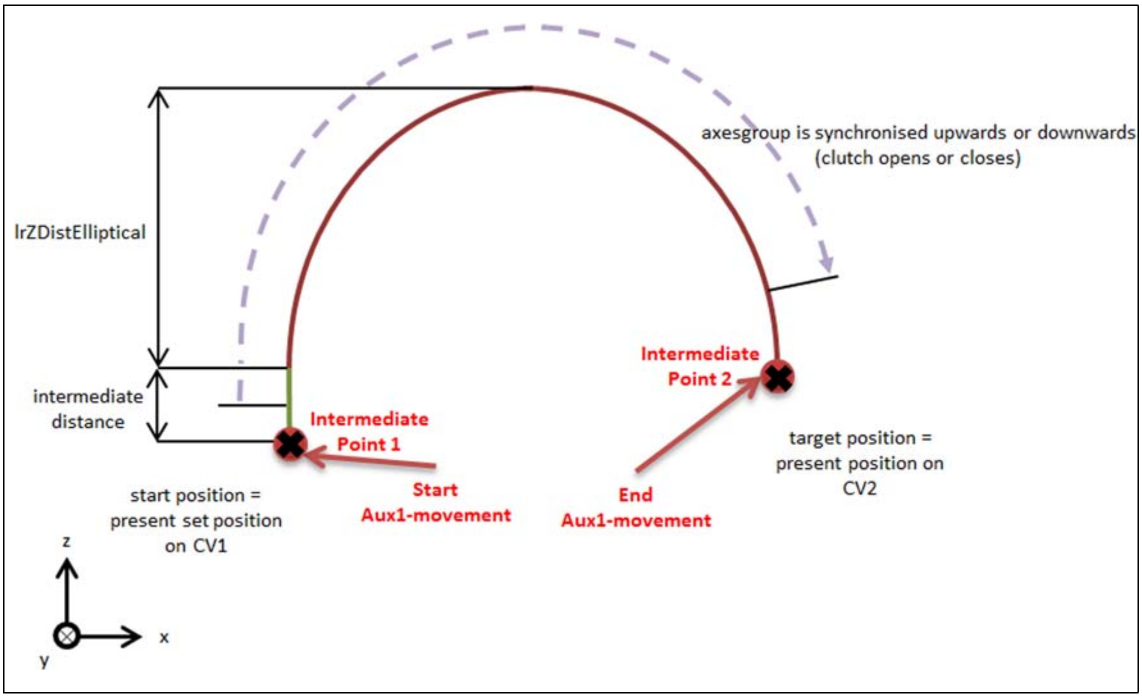
[3-9] Example 2: Intermediate Point 2 is lower than Point 1

In the example 2, the Intermediate Point 2 is lower than the Intermediate Point 1 in Z direction of the world coordinate system (WCS). Thus, a linear intermediate segment is inserted after the Intermediate Point 2 and the end position of the ellipse to compensate the height difference.

The dashed line shows the path with activated clutch.

Along the path, the clutch can ...

- clutch in from the static position to a conveying belt;
- reclutch to the next conveying belt;
- declutch from the conveying belt to the static target position.



[3-10] Example 3: Starting and target height are zero

Example 3 shows an ellipse motion in which the starting and target height are equal to zero. In this case, the starting position is put onto the Intermediate Point 1 and the target position onto the Intermediate Point 2.

As the starting position here is lower than the target position, a linear intermediate distance is automatically inserted between the Intermediate Point 1 and the starting position of the ellipse.

The dashed line shows the path with activated clutch.

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

After a short travel in Z direction, the clutch is activated and deactivated shortly before reaching the Intermediate Point 2.

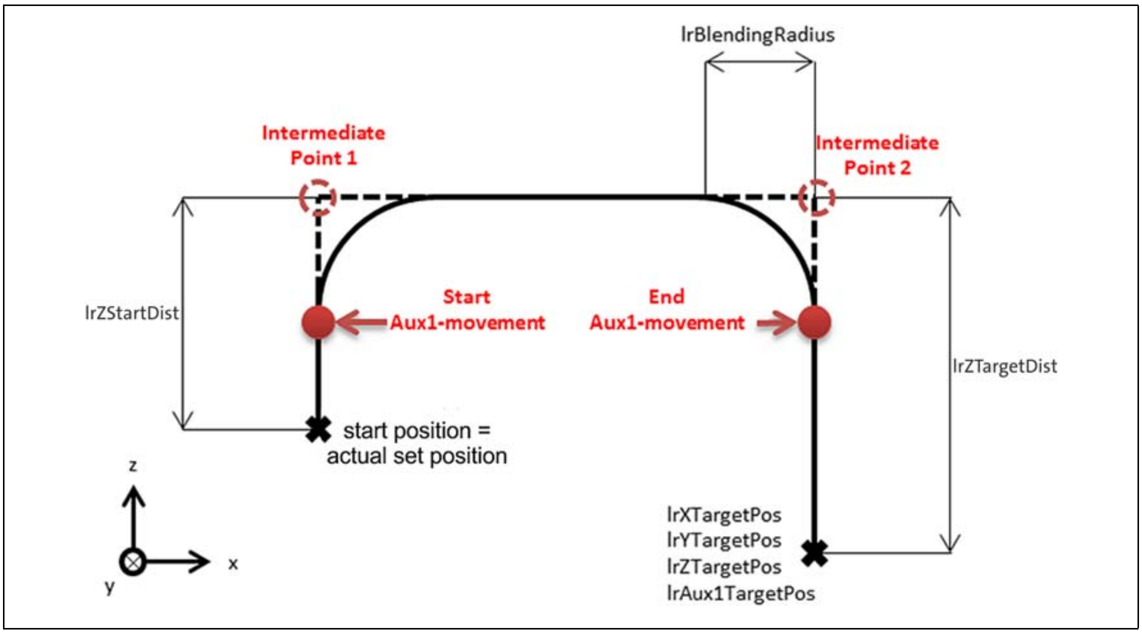
Parameters to be set

The parameters to be set are located in the L_TT1P_scPar_TrackPickAndPlaceBase (📖 29) parameter structure.

```
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]
ePathMode :  L_TT1P_PickAndPlacePathMode;
lrZDistElliptical : LREAL;
```

3.12 Calculation of the profile points with lrBlendingRadius > 0

3.12.1 Linear profile (ePathMode = 0)



[3-11] 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 before 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.

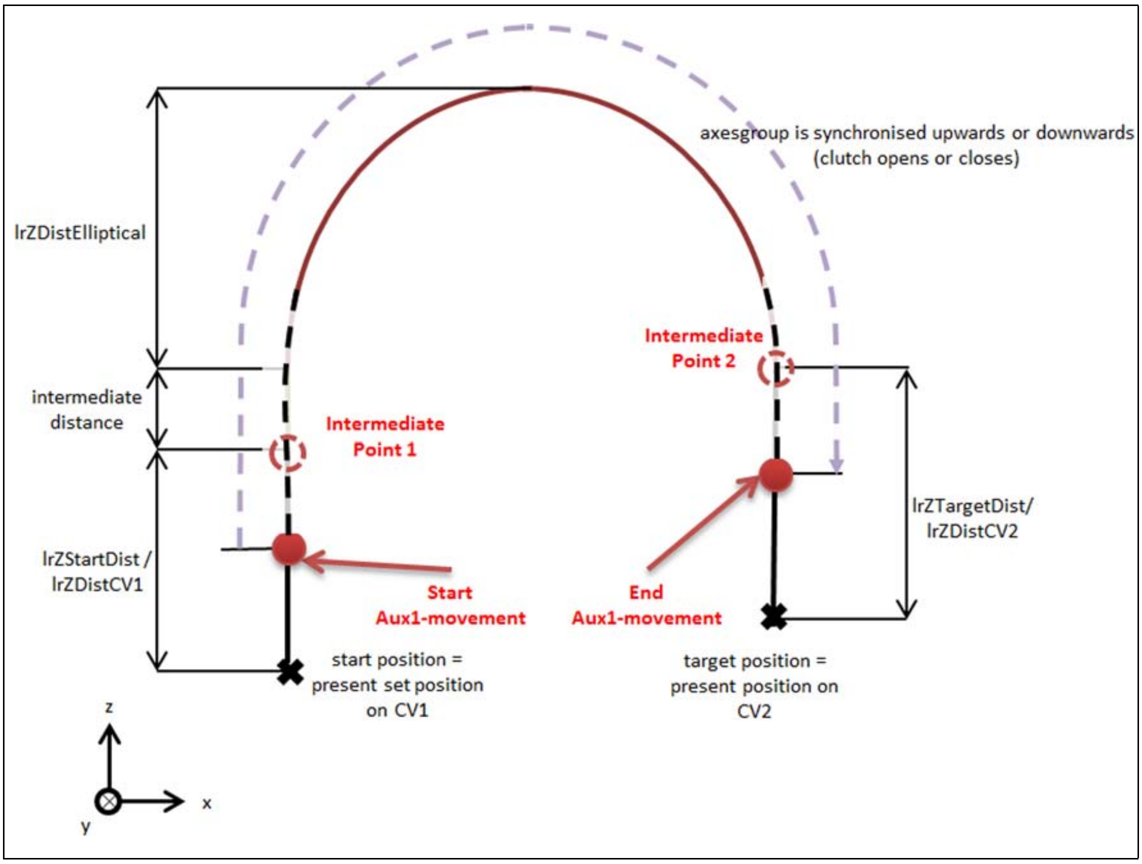
The movement of orientation axes A, B, C, and the auxiliary axis Aux1 starts in the moment in which the rounding movement before the Intermediate Point 1 is starting (depending on the blending radius). The movement of the axes ends when the rounding movement before Intermediate Point 2 has been completed.

Parameters to be set

The parameters to be set are located in the L_TT1P_scPar_TrackPickAndPlaceBase (29) parameter structure.

```
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]
```

3.12.2 Ellipse profile (ePathMode = 1)



[3-12] Example: Rounding of the travel profile at the Intermediate Points

In case of this travel profile, the Intermediate Points 1 and 2 between the starting and target position are not approached but the profile is rounded at these points. The profile is executed without a stop.

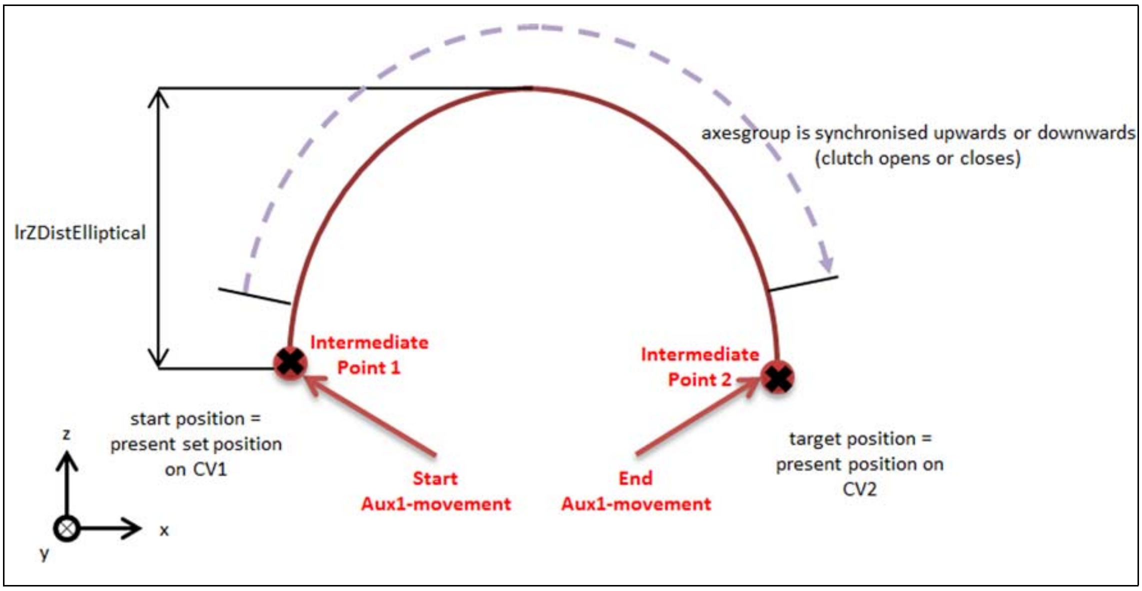
The *lrZStartDist* parameter is used to define the height of the "Tool Center Point" (Intermediate Point 1). The profile travel process is lead over Intermediate Point 1, depending on the blending radius in the *lrBlendingRadius* parameter.

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 is lead over Intermediate Point 2, depending on the *lrBlendingRadius* parameter.

There is a maximum limit for setting the *lrBlendingRadius* parameter. 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 at Intermediate Point 1 starts (depending on the blending radius). The movement of the axes ends when the rounding movement at Intermediate Point 2 has been completed.

The coupling process also starts when the rounding motion at Intermediate Point 1 starts and is completed with the rounding end at Intermediate Point 2.



[3-13] Example: Starting and target height are zero

Figure [3-13] shows an ellipse motion in which the starting and target height are zero and both Intermediate Points are on the same Z height.

In this case, no rounding motions are inserted even if a rounding radius greater than zero is given.

The dashed line shows the path with activated clutch.

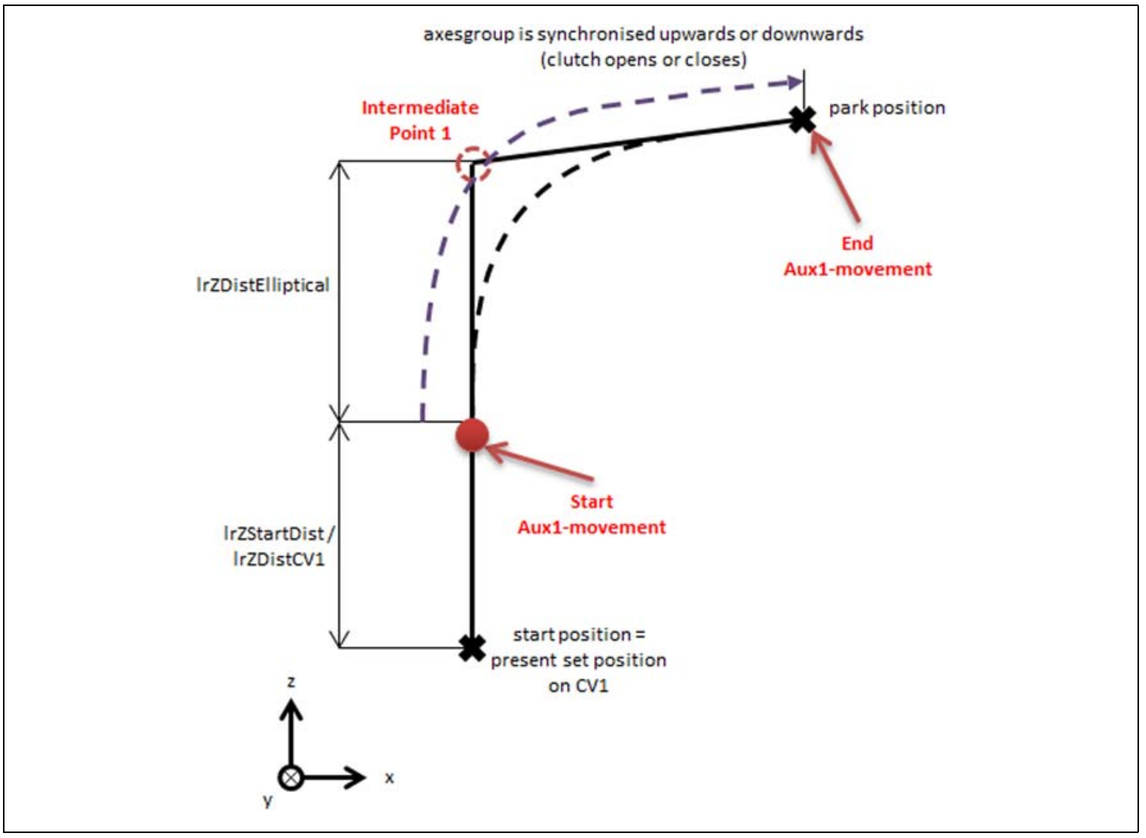
The movement of the orientation axes A, B, C and the auxiliary axis Aux1 begins at a starting position after the Intermediate Point 1 and ends at a target position before the Intermediate Point 2.

After a short travel in Z direction, the clutch is activated and deactivated shortly before reaching the Intermediate Point 2.

Special case: Travel to a parking position

A travel from a conveying belt to a parking position or vice versa always takes place via two linear segments, irrespective of the setting in *ePathMode*.

The two linear segments are connected via rounding.



[3-14] Example: Travel to the parking position of a conveying belt

In this travel profile, an Intermediate Point 1 is inserted between the starting and the parking position. The point is located above the starting position in Z height which consists of the sum of the starting height *lrZStartDist* and the ellipse height *lrZDistElliptical*.

The Intermediate Point 1 is not approached but rounded. The rounding radius consists of the sum of the ellipse height *lrZDistElliptical* and the rounding radius.

The movement of the orientation axes A, B, C and the auxiliary axis Aux1 starts when the rounding movement at the Intermediate Point 1 starts. The movement of the axes ends when the parking position has been reached.

The asynchronisation of a conveying belt also starts when the rounding movement starts at Intermediate Point 1 and is completed when the parking position has been reached.

Parameters to be set

The parameters to be set are located in the L_TT1P_scPar_TrackPickAndPlaceBase (📖 29) parameter structure.

```
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]
ePathMode :  L_TT1P_PickAndPlacePathMode;
lrZDistElliptical :  LREAL;
```

3.13 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:

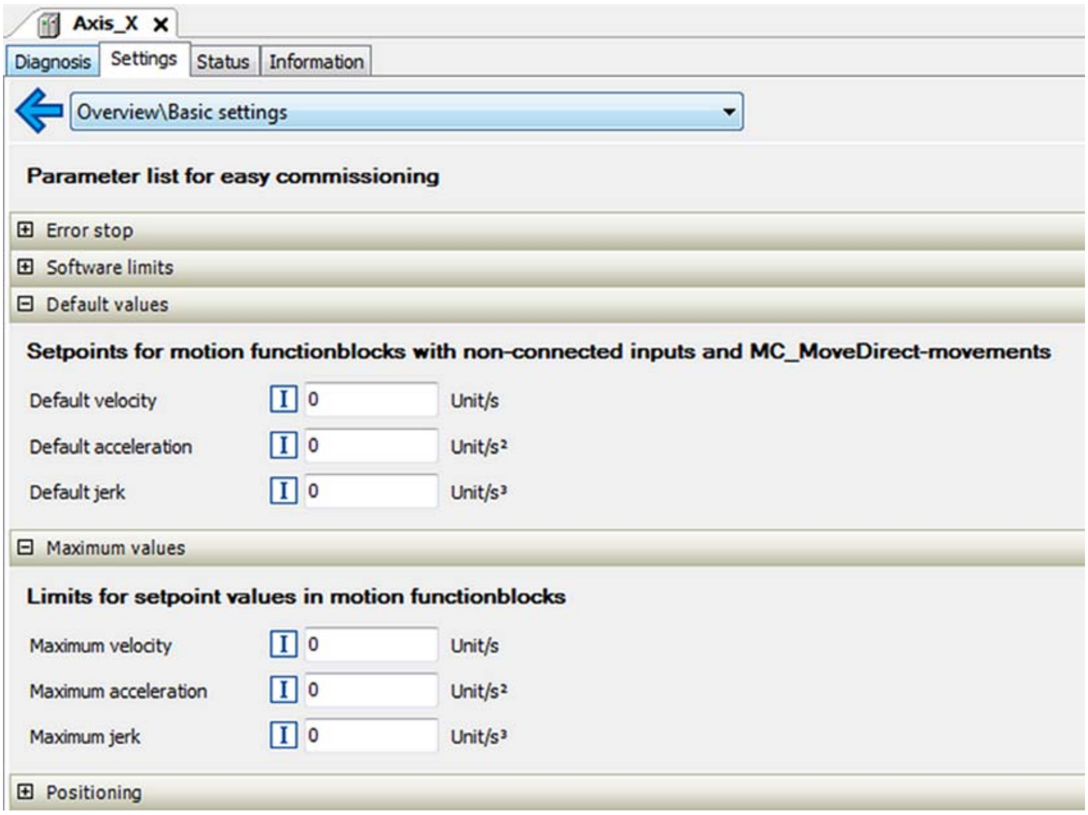
- In the course of the path calculation, the speed, acceleration and jerk of the path are automatically reduced, so that the maximum values of axes X, Y, Z, A, B, C and Aux1 are not exceeded.
- For the interpolation, the maximum values from the axis are taken into consideration. If the maximum values 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_TrackPickAndPlaceBase (📖 29) 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;
```

3.14 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) (📖 58)
- 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) (📖 59)
- *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.

3.15 Adding more travel profiles

Precondition

- All axes are enabled (*xRegulatorOnALL* input = TRUE).
- 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 status is provided via the *xProfileInPreparation* output:

- TRUE: The technology module calculates and prepares the profile (path) in the background.
- FALSE: The preparation of the profile is completed.

A renewed FALSE↗TRUE edge at the *xExecutePickAndPlace* input serves to add another profile. If no other profile can be added, the *xProfileInPreparation* output is set to TRUE.

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

3.16 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) (📖 38)

Stop!

The holding position must be saved after the stop. After Manual jog (jogging) (📖 38), 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_TrackPickAndPlaceBase (📖 29) 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]
```

3.17 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) (🕮 38)

STOP

Stop!

The holding position must be saved after the stop. After Manual jog (jogging) (🕮 38), 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.

3.18 Gripper control

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

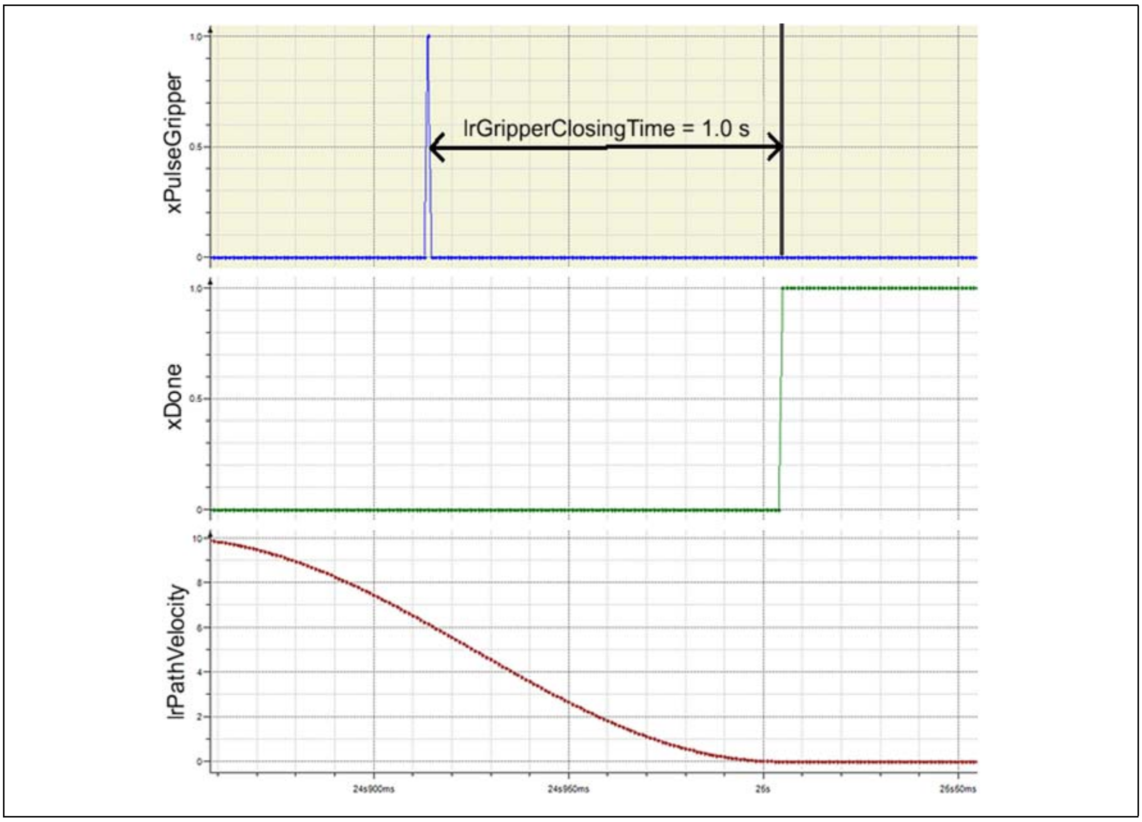
The *xPulseGripper* output is triggered depending on the *lrGripperClosingTime* and/or *lrGripperOpenTime* parameter. Here, positive values cause the gripper to be opened by the set value in seconds before the profile end is reached.

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.

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

Figure [3-15] shows the impact of the *lrGripperClosingTime* parameter from the L_TT1P_scPar_TrackPickAndPlaceBase (29) parameter structure on the *xPulseGripper* output.



[3-15] Impact of the lrGripperClosingTime parameter on the xPulseGripper output

3.19 Selecting the gripper control mode

For simply transporting the workpieces from a static position to a running conveying belt or from a running conveying belt to a second conveying belt, the technology module provides various modes.

Depending on the application, a mode can be selected via the *eTrackingMode* parameter:

- eTrackingMode = 0 (📖 61)
- eTrackingMode = 1 (Static => CV1) (📖 61)
- eTrackingMode = 2 (CV1 => Static) (📖 62)
- eTrackingMode = 3 (CV1 => CV2) (📖 62)
- eTrackingMode = 4 (CV2 => CV1) (📖 62)

3.19.1 eTrackingMode = 0

For this mode, the robot handling must be implemented outside the technology module. The target is transferred to the technology module via the *eSetTarget* input and approached with a FALSE↗TRUE edge at the *xExecutePickAndPlace* input.

In the technology module, the travel profile/path is calculated and executed. In this process, targets can be assigned to the corresponding conveying belts. The synchronisation to the target and the approach of the target are automatically executed by the technology module.

A renewed FALSE↗TRUE edge at the *xExecutePickAndPlace* input and the target information via the *eSetTarget* input serves to load further targets.

- Adding more travel profiles (📖 57)

For decelerating the gripper control, the time in the *lrGripperClosingTime* parameter (in seconds) is accepted.

The *lrGripperOpenTime* parameter is not used.

3.19.2 eTrackingMode = 1 (Static => CV1)

In this mode, the technology module controls the pick-up and placement of the workpieces by the gripper. The targets are automatically defined in an infinite loop.

The workpieces are picked up at the static position (parameters *lrXTargetPos*, *lrYTargetPos*, *lrZTargetPos*, *lrCTargetPos*, *lrAux1TargetPos*). For this purpose, the time for closing the gripper must be set in the *lrGripperClosingTime* parameter (in seconds).

The workpieces are synchronously placed onto the conveying belt 1. For this purpose, the time for opening the gripper must be set in the *lrGripperOpenTime* parameter (in seconds).

The mode can be aborted via the inputs *xAbort*, *xPathHalt*, *xPathStop* or with a FALSE↗TRUE edge at the *xExecutePickAndPlace* input and a new mode selection.

3.19.3 eTrackingMode = 2 (CV1 => Static)

In this mode, the technology module controls the pick-up and placement of the workpieces by the gripper. The targets are automatically defined in an infinite loop.

The workpieces are picked up from the conveying belt 1. For this purpose, the time for closing the gripper must be set in the *lrGripperClosingTime* parameter (in seconds).

The workpieces are placed onto the static position (parameters *lrXTargetPos*, *lrYTargetPos*, *lrZTargetPos*, *lrCTargetPos*, *lrAux1TargetPos*). For this purpose, the time for opening the gripper must be set in the *lrGripperOpenTime* parameter (in seconds).

The mode can be aborted via the inputs *xAbort*, *xPathHalt*, *xPathStop* or with a FALSE↗TRUE edge at the *xExecutePickAndPlace* input and a new mode selection.

3.19.4 eTrackingMode = 3 (CV1 => CV2)

In this mode, the technology module controls the pick-up and placement of the workpieces by the gripper. The targets are automatically defined in an infinite loop.

The workpieces are picked up from the conveying belt 1. For this purpose, the time for closing the gripper must be set in the *lrGripperClosingTime* parameter (in seconds).

The workpieces are synchronously placed onto the conveying belt 2. For this purpose, the time for opening the gripper must be set in the *lrGripperOpenTime* parameter (in seconds).

The mode can be aborted via the inputs *xAbort*, *xPathHalt*, *xPathStop* or with a FALSE↗TRUE edge at the *xExecutePickAndPlace* input and a new mode selection.

3.19.5 eTrackingMode = 4 (CV2 => CV1)

In this mode, the technology module controls the pick-up and placement of the workpieces by the gripper. The targets are automatically defined in an infinite loop.

The workpieces are picked up from the conveying belt 2. For this purpose, the time for closing the gripper must be set in the *lrGripperClosingTime* parameter (in seconds).

The workpieces are synchronously placed onto the conveying belt 1. For this purpose, the time for opening the gripper must be set in the *lrGripperOpenTime* parameter (in seconds).

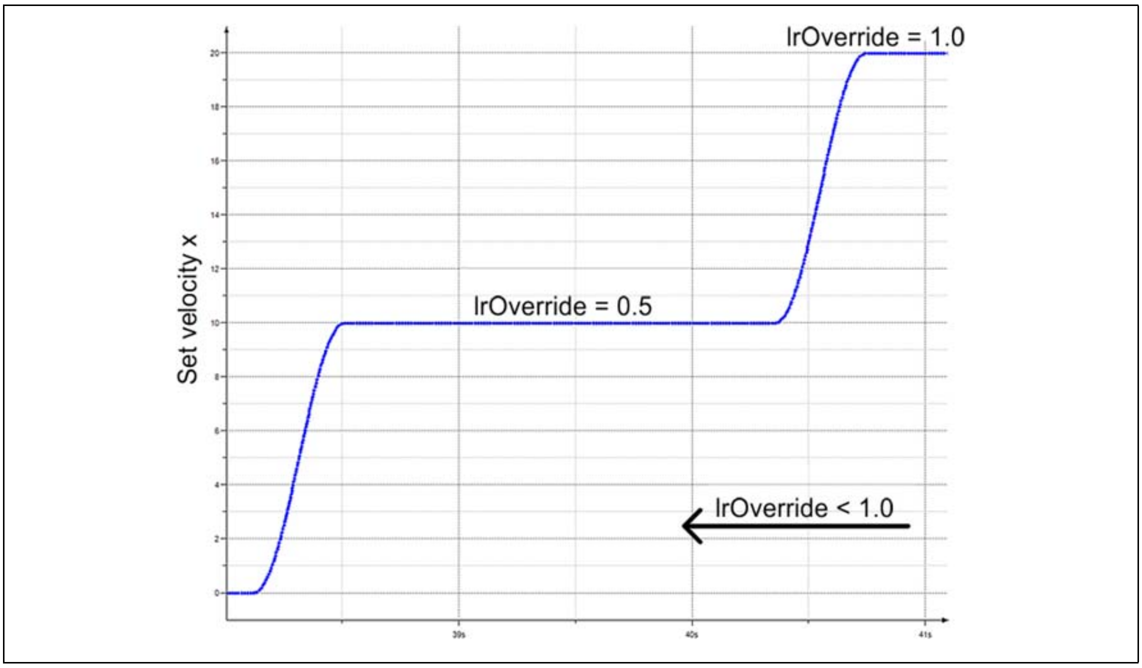
The continuous mode can be completed by setting the *xAbort* input to TRUE. Here, the current path is travelled until the end.

Setting the inputs *xPathHalt* or *xPathStop* to TRUE aborts the mode immediately.

3.20 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 (📖 60).

The impact of *lrOverride* on the speed is shown in figure [3-16]. The value '0.5' halves the speed, the acceleration, and the jerk.



[3-16] Impact of lrOverride on the speed

3.21 Parameter setting of the conveying belts

The conveying belts are parameterised via the Array ascConveyorBeltPar[i] (📖 36).

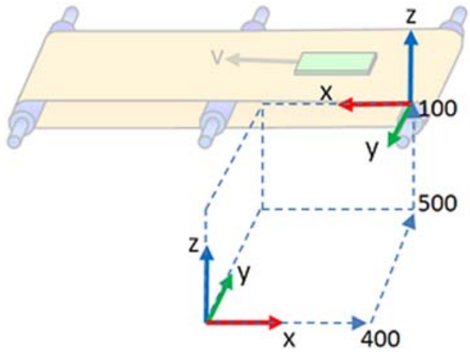
Up to 16 conveying belts can be parameterised with the array.

How to parameterise the conveying belts:

1. Defining the conveying belt coordinate system.

   The *alrConveyorBeltOrigin[1..6]* parameter field serves to define the location and position of the conveying belt. The position and orientation of the conveying belt is defined in the machine coordinate system (MCS). The X axis of the conveying belt must point into the positive direction of motion of the conveying belt.

   Example



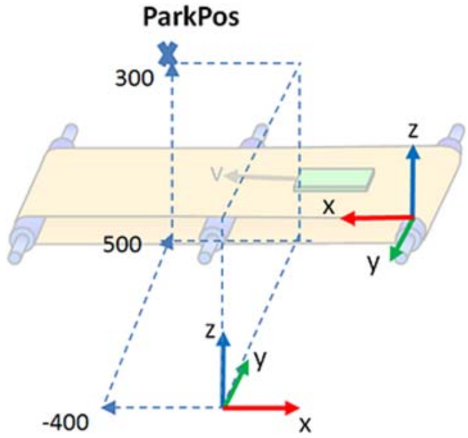
   The origin of coordinates of the conveying belt is in X = 400, Y = 500 and Z = 100. The X axis is positioned in the positive direction of motion of the conveying belt. Thus, the orientation of the origin of coordinates is rotated by C = 180° around the Z axis.

   Parameter field:
   - *alrConveyorBeltOrigin[1]* := 400; // position in X direction [units]
   - *alrConveyorBeltOrigin[2]* := 500; // position in Y direction [units]
   - *alrConveyorBeltOrigin[3]* := 100; // position in Z direction [units]
   - *alrConveyorBeltOrigin[4]* := 0; // position in A direction, rotation around X axis [deg]
   - *alrConveyorBeltOrigin[5]* := 0; // position in B direction, rotation around Y axis [deg]
   - *alrConveyorBeltOrigin[6]* := 180; // position in C direction, rotation around Z axis [deg]

2. Defining the parking position.

   The *alrParkPos[1..6]* parameter field serves to define the location and position of the parking position for the conveying belt. The position and orientation of the parking position is defined in the machine coordinate system (MCS). The parking position is approached when the target on the conveying belt cannot be reached (with regard to the defined operating range) or has not yet been transferred to the technology module.

   Example



   The origin of coordinates of the parking position is located at X = -400, Y = 500 and Z = 300.

   Parameter field:

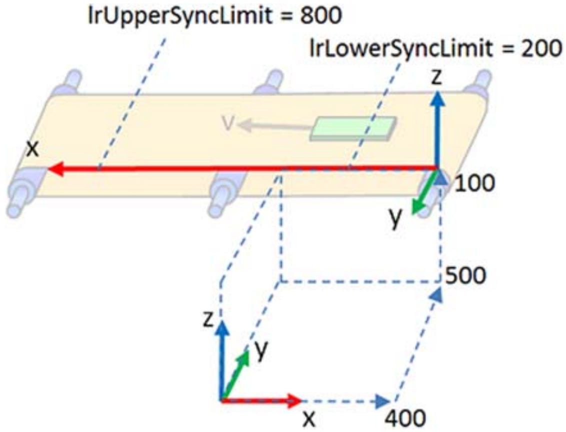
   - *alrParkPos[1]* := -400; // parking position in X direction in [unit]
   - *alrParkPos[2]* := 500; // parking position in Y direction in [unit]
   - *alrParkPos[3]* := 300; // parking position in Z direction in [unit]
   - *alrParkPos[4]* := 0; // parking position in A direction in [unit]
   - *alrParkPos[5]* := 0; // parking position in B direction in [unit]
   - *alrParkPos[6]* := 0; // parking position in C direction in [unit]

3. Defining the operating range on the conveying belt.

   The parameter ...
   - *lrLowerSyncLimit* serves to define the lower limit on the X axis of the conveying belt;
   - *lrUpperSyncLimit* serves to define the upper limit on the X axis of the conveying belt.

   The technology module only synchronises to a workpiece on the conveying belt if the workpiece is located between the lower and upper limit. Otherwise, it is waited in the parking position until a workpiece can be picked up in the defined operating range.

   Example

   The lower limit of the operating range is at position X = 200, the upper limit at X = 800.

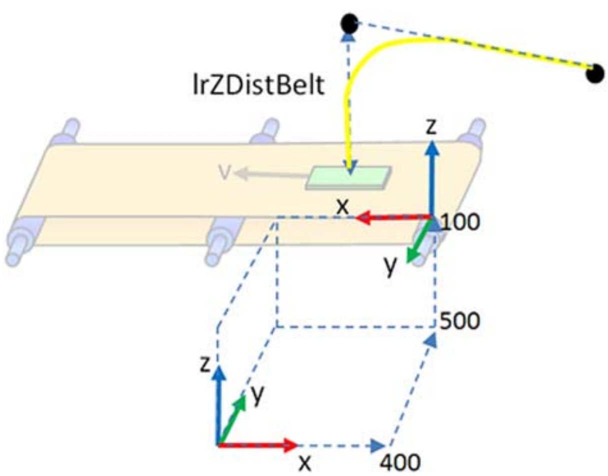
   Parameters for the operating range:
   - *lrLowerSyncLimit* := 200; //lower limit in X direction [units]
   - *lrUperSyncLimit* := 800; // upper limit in X direction [units]

4. Defining the height of the lifting movement above the conveying belt.

   For picking up/putting down the workpiece, the "Tool Center Point" (TCP) of the gripper arriving or leaving from above must be synchronous to the conveying belt. Moreover, prevent the workpiece from dragging on the conveying belt. See Calculation of the path (contour) to a workpiece

   The *lrZDistBelt* parameter serves to define the height in Z direction in the product coordinate system (PCS) above the workpiece on the conveying belt.

   Example

5. Defining the duration of the synchronous travel of the gripper above the conveying belt.

   The duration of the synchronous travel depends on the gripper control. Depending on the gripper type, picking up a workpiece might take a certain amount of time. Thus, it is required that the "Tool Center Point" travels synchronously to the conveying belt above the workpiece for a while before the lifting movement is carried out.

   The *lrSyncTimeOnBelt* parameter serves to define the duration of the synchronous travel above the conveying belt (in seconds) before the lifting motion is carried out. When *lrSyncTimeOnBelt* = 0, the TCP remains above the conveying belt for one clock cycle before the lifting movement is carried out.

Note!

A FALSE↗TRUE edge at the *xAckGripper* input aborts the synchronous travel. In this process, the lifting motion is executed immediately, independent of the time remaining for the synchronous travel.

3.22 Detection of the workpieces

The workpieces on the conveying belts are detected via the Input structure ascBeltInput[i] (📖 23).

With a rising edge (FALSE↗TRUE) at the *xWorkpieceOnBeltReceive* input, a workpiece is detected on the conveying belt by the technology module and tracked via the guidance of the conveying belt. At this time, the position coordinates of the workpiece might still be unknown.

The coordinates of the workpiece valid at the time of the FALSE↗TRUE edge at the *xWorkpieceOnBeltReceive* input are taken over with a FALSE↗TRUE-edge at the *xValidPos* input from the *ascInitialObjectPosition[1..6]* input. The coordinates of the workpiece in the conveying belt coordinate system are defined at the *ascInitialObjectPosition[1..6]* input.

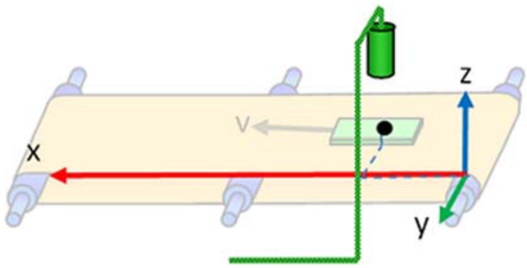
Only when the workpiece has been recognised with *xWorkpieceOnBeltReceive* and the workpiece coordinates have been recognised with *ascInitialObjectPosition[1..6]*, a dynamic product coordinate system (PCS) is generated in the technology module. From this time on, the technology module can "see" a workpiece. When the workpiece is positioned within the operating range of the conveying belt, the workpiece can be accomodated. (See Parameter setting of the conveying belts (📖 64).)

If the order for picking up the workpiece comes too late and if the workpiece has already left the operating range, the information on the workpiece in the technology module will be deleted. The product coordinate system (PCS) for this workpiece will be enabled (deleted). Afterwards, the following workpiece is fetched if it has been detected by the technology module and the conditions for pick-up are fulfilled.

The technology module can detect, track and process up to 30 workpieces in parallel on each conveying belt. If the maximum number (30 workpieces) is reached, a warning with the error number '17918' will be provided at the *eErrorID* output. In this case, no further workpiece can be detected by the technology module.

Example

Detecting a workpiece on a conveying belt and determining the coordinates in the conveying belt coordinate system:



The workpiece is detected at X = 210, Y = -50 and Z = 10 in the conveying belt coordinate system.

Parameter field:

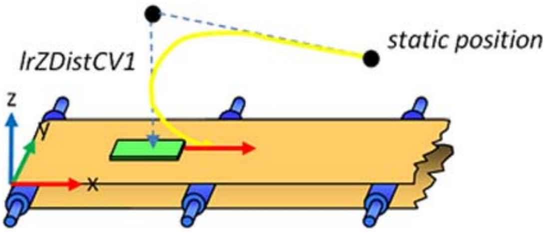
- *ascInitialObjectPosition[1]* := 210; // position in X direction [units]
- *ascInitialObjectPosition[2]* := -50; // position in Y direction [units]
- *ascInitialObjectPosition[3]* := 10; // position in Z direction [units]
- *ascInitialObjectPosition[4]* := 0;
- *ascInitialObjectPosition[5]* := 0;
- *ascInitialObjectPosition[6]* := 0;

3.23 Calculation of the path (contour) to a workpiece

For calculating the path to a workpiece, the parameters from the Array ascConveyorBeltPar[i] (📖 36) are used.

3.23.1 Static position => conveying belt[i]

Linear profile (ePathMode = 0)



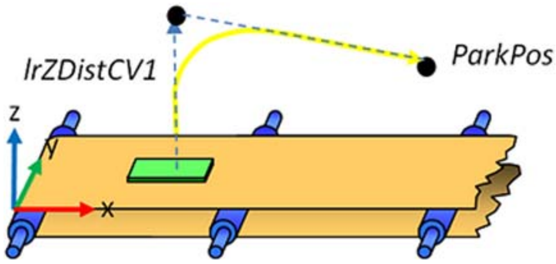
Based on a static position, a valid target (workpiece) is detected on the conveying belt [i]. For this purpose, the path (contour) to a workpiece on the conveying belt [i] is calculated.

If there is no workpiece on the conveying belt, the related parking position (*alrParkPos* parameter) is approached. The parking position will also be approached if the workpiece is outside the defined operating range in the *lrLowerSyncLimit* and *lrUpperSyncLimit* parameters.

As soon as a workpiece has been detected on the conveying belt, the synchronous travel of the gripper above the conveying belt and the path towards the workpiece are calculated. In the contour, an intermediate point is entered via the target position on the relative Z height (*lrZDistBelt* parameter). The intermediate point is approached with a rounding. Thus, a stop at the intermediate point can be avoided. The workpiece is picked up or put down. The synchronisation to the workpiece is completed before lowering movement is completed.

3.23.2 Conveying belt[i] => static position/parking position

Linear profile (ePathMode = 0)



The path (contour) for the gripper is calculated via an intermediate point in the relative height above the conveying belt [i] in Z direction (*lrZDistBelt* direction). The intermediate point is approached with a rounding. If there is no valid target (workpiece) on the conveying belt, the related parking position (*alrParkPos parameter*is approached.

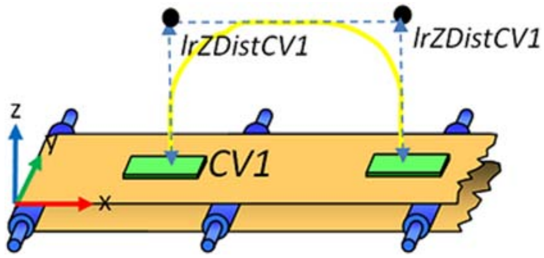
Irrespective of the target position, the lifting movement is always carried out from the synchronous travel. The time for the synchronous travel of the gripper above the target (workpiece) on the conveying belt [i] is defined by the *lrSyncTimeOnBelt* parameter.

A FALSE↗TRUE edge at the *xAckGripper* input aborts the synchronous travel. In this process, the lifting motion is executed immediately, independent of the time remaining for the synchronous travel.

In the lifting movement, the "Tool Center Point" still remains synchronously to the conveying belt. The down-synchronisation takes place on the remaining residual path to the target point.

3.23.3 Conveying belt[i] => conveying belt[i]

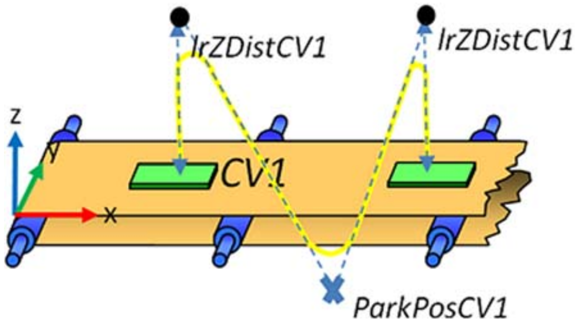
Linear profile (ePathMode = 0)



The technology module can synchronise several times to one and the same conveying belt [i].

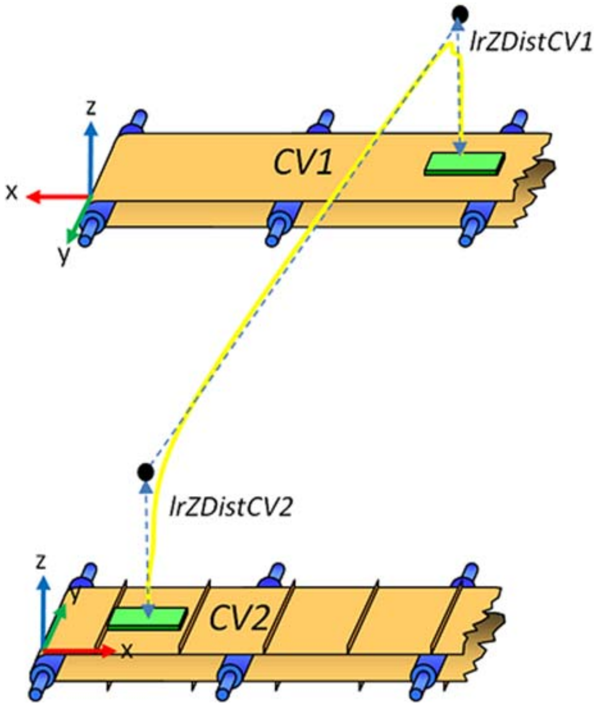
If several valid targets (workpieces) are detected on the conveying belt, it is resynchronised from one target to the next. The gripper travels in the relative height above the conveying belt in Z direction (*lrZDistBelt* parameter) from one workpiece to the next. In this process, two intermediate points are calculated in the contour and executed with the blending radius in the *lrBlendingRadius* parameter.

If the second workpiece on the same conveying belt is not yet available, the parking position (*alrParkPos* parameter) is inserted as intermediate point and approached.



3.23.4 Conveying belt[1] => conveying belt[2]

Linear profile (ePathMode = 0)



During a synchronous travel with a workpiece on the first conveying belt (CV1), a target can be detected on the second conveying belt (CV2).

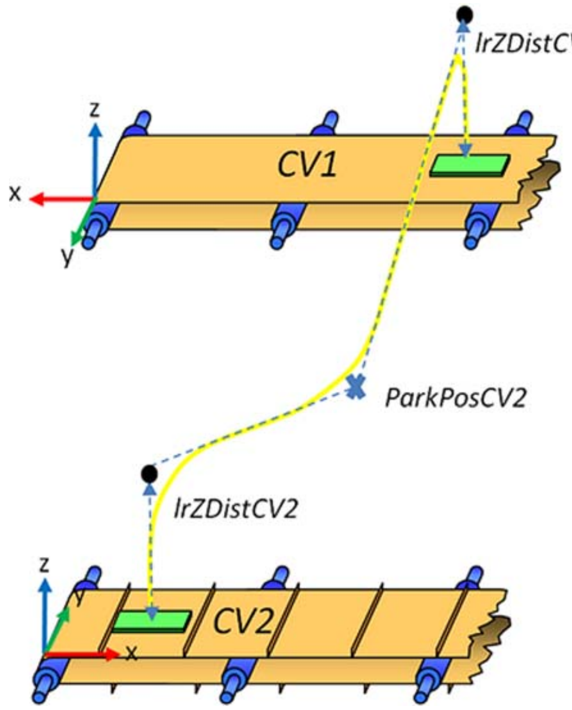
If the target is located on the second conveying belt within the defined operating range (parameter *ascConveyorBeltPar[2].lrLowerSyncLimit* and *ascConveyorBeltPar[2].lrUpperSyncLimit*), it is approached via two intermediate points. The first intermediate point is positioned in the relative height above the first conveying belt in Z direction (*ascConveyorBeltPar[1].lrZDistBelt* parameter).

The synchronous travel consists of the lifting movement and the movement towards the intermediate point above the first conveying belt. Before the intermediate point is reached, a rounding movement is carried out with the blending radius in the *lrBlendingRadius* parameter.

The second intermediate point is positioned in the relative height (*ascConveyorBeltPar[2].lrZDistBelt* parameter) above the target on the second conveying belt onto the contour. At the second intermediate point, a rounding movement is carried out as well with the blending radius in the *lrBlendingRadius* parameter.

During the travel from the first intermediate point to the second, the "Tool Center Point" is resynchronised from the first conveying belt (CV1) to the second conveying belt (CV2). In the lowering movement from the second intermediate point to the target on the second conveying belt (CV2), the TCP is already synchronised.

If the target is located on the second conveying belt (CV2) in front of the limitation in the *ascConveyorBeltPar[2].lrLowerSyncLimit* parameter, the *ascConveyorBeltPar[2].alrParkPos* parking position is inserted as intermediate point. In the parking position it is waited until the target on the second conveying belt can be reached.



3.24 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

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

FEEDBACK

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