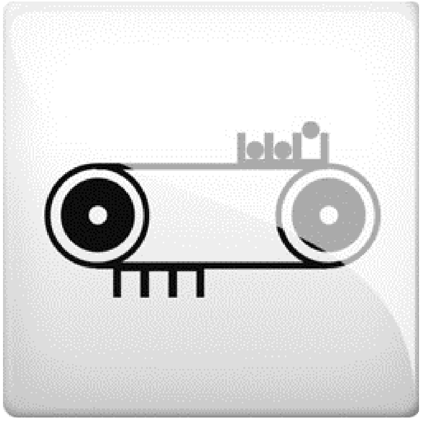
FAST Application Software



# Technology Module

Magic Track

Reference Manual EN

13517861

Lenze

# Contents

# 1 About this documentation

This documentation ...

- contains detailed information on the functionalities of the "Magic Track" technology module within the scope of the FAST Application Software;
- is part of the "Controller-based Automation" manual collection. It consists of the following sets of documentation:

| Documentation type | Subject |
|---|---|
| **System manuals** | Visualisation (system overview/sample topologies) |
| **Communication manuals**<br>**Online helps** | Bus systems<br>• Controller-based Automation EtherCAT®<br>• Controller-based Automation CANopen®<br>• Controller-based Automation PROFIBUS®<br>• Controller-based Automation PROFINET® |
| **Reference manuals**<br>**Online helps** | Lenze Controllers:<br>• Controller 3200 C<br>• Controller c300<br>• Controller p300<br>• Controller p500 |
| **Software manuals**<br>**Online helps** | Lenze Engineering Tools:<br>• »PLC Designer« (programming)<br>• »Engineer« (parameter setting, configuration, diagnostics)<br>• »VisiWinNET® Smart« (visualisation)<br>• »Backup & Restore« (backup, restore, 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:

| Mounting & wiring |
| --- |
| 🕮 **Mounting instructions**<br>• Controllers<br>• Communication cards (MC-xxx)<br>• I/O system 1000 (EPM-Sxxx)<br>• Inverters, Servo Drives<br>• Communication modules |
| ☐ **Hardware manuals**<br>• Inverters, Servo Drives |
| ☐ **Operating instructions**<br>• Controllers<br>• Servo system ECS (ECSxE, ECSxM) |
| **Sample applications/Using application templates** |
| ☐ **Online help/reference manuals**<br>• i700 application sample<br>• Application Samples<br>• Application Template<br>• FAST technology modules |
| **Parameterisation, configuration, commissioning** |
| ☐ **Online help/reference manuals**<br>• Controllers<br>• Inverters, Servo Drives<br>• I/O system 1000 (EPM-Sxxx) |
| ☐ **Online help/communication manuals**<br>• Bus systems<br>• Communication modules |
| ☐ **Operating instructions**<br>• Servo system ECS (ECSxE, ECSxM) |

**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 addresses to all persons who plan, program, and commission a Lenze automation system on the basis of the FAST application software and as part of "Controller-based Automation".

## 1.1 Document history

| Version | | | Beschreibung |
| --- | --- | --- | --- |
| 1.0 | 08/2016 | TD29 | 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.<br>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<br>`"Courier new"` | `...`<br>`dwNumerator := 1;`<br>`dwDenominator := 1;`<br>`...` |
| Icons | | |
| Page reference | (📖 6) | Reference to further information:<br>Page number in PDF file. |

### Variable names

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

## 1.3 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**<br>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**<br>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**<br>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!

**Unpredictable motor movement**

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 an online connection to the device is establishedPersonal injury by dangerous electrical voltage

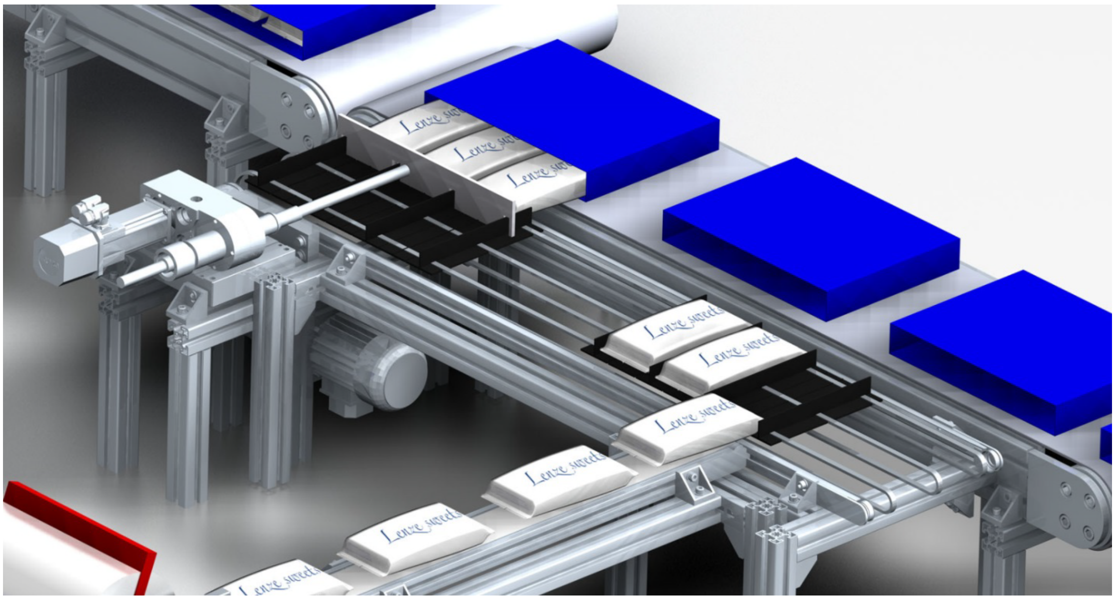
**Possible consequences**

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.

# 3 Functional description



[3-1] Typical mechanics of the technology module

In order to fill the individually packed products into a multiple packaging, they are grouped by means of a magic belt, called MagicTrack. Grouped transport trays (called tracks in the following) are generated on parallel conveying belts or chains (called rail in the following). As soon as these tracks are filled with products in the loading station, they travel jointly to the unloading position. There, the grouped products are moved sideways into the multiple packaging. The tracks then return to the loading station to be loaded again.

A TM Magic Track consists of two or more rails on which the tracks are moved independently of each other but cannot pass each other. Each track can consist of any number of trays and gaps of different sizes.

It is possible to have more than one track on a rail or to increase the track number by the number of rails. The former is achieved by dividing the modulo value. Here, the conditions for symmetrical distribution, identical track structure (trays) and the same number of tracks per rail must be fulfilled.

The "Magic Track" technology module consists of a **L_TT1_MagicTrackControl** control unit and several **L_TT1P_MagicTrackAxis** axis control units. A MagicTrack, for instance, consisting of two rails with one track each per rail is controlled via a TM MagicTrackControl and two TMs MagicTrackAxis.

The task of the TM MagicTrackControl is to manage the operating process and to monitor several axis control units TM MagicTrackAxis.

The TM MagicTrackAxis takes over the guidance of an axis. An axis simultaneously is a rail in the MagicTrack technology.

All parameters and control signals are processed in the TM MagicTrackControl and the connected TMs MagicTrackAxis are transmitted. The movement and profile calculation are prepared and calculated separately for each rail in the respective TM MagicTrackAxis and executed independently of each other.

The scaling (Base, State, High) of the functions of the MagicTrack technology module is mapped on the TM MagicTrackControlBase, State and High. The TM MagicTrackAxis is available without scaling in one version and supports the Base, State and High version of the TM MagicTrackControl.

- In the **Base version**, the MagicTrack technology can at least consist of two or more rails. One track is always controlled on one rail. One track can consist of trays and gaps between the trays of different sizes. In the Base version, the clocked loading and unloading is supported. The collision monitoring function ensures that the tracks to not pass each other. The tracks keep a minimum distance to each other to prevent the tracks from collisions. The minimum distance can be parameterised. The jerk-limited velocity trapezoid is used for calculation the profiles of the axes.
- In the **State version**, the functional range of the Base version is extended. Here, the synchronous loading and unloading via master value axes is enabled in addition to clocked loading/unloading.
- The **High version** additionally offers the opportunity to execute the profiles synchronously to a master value via cam.

**Note**

The preconditions for the operation of the MagicTrack technology are as follows:
- All axes are modulo axes and have the same cycle length
- The cycle length of an axis corresponds to the cycle length of a rail on which a train moves

## 3.1 Overview of the technology module functions

In addition to the basic functions for operating the **L_MC1P_AxisBasicControl** function block, the **Stop function** and the **Holding function**, the technology module offers the following functionalities that are assigned to the "Base", "State" and "High" versions:

| Functionality | Versions | | |
|---|---|---|---|
| | Base | State | High |
| Manual jog (jogging) | ● | ● | |
| Group jogging | ● | ● | |
| Homing | ● | ● | |
| Collision monitoring | ● | ● | |
| Several trays in one track | ● | ● | |
| asymmetrical tray lengths | ● | ● | |
| Gaps between trays | ● | ● | |
| Automatic homing | ● | ● | |
| Clocked loading | | ● | |
| Clocked unloading | | ● | |

**»PLC Designer« online help**

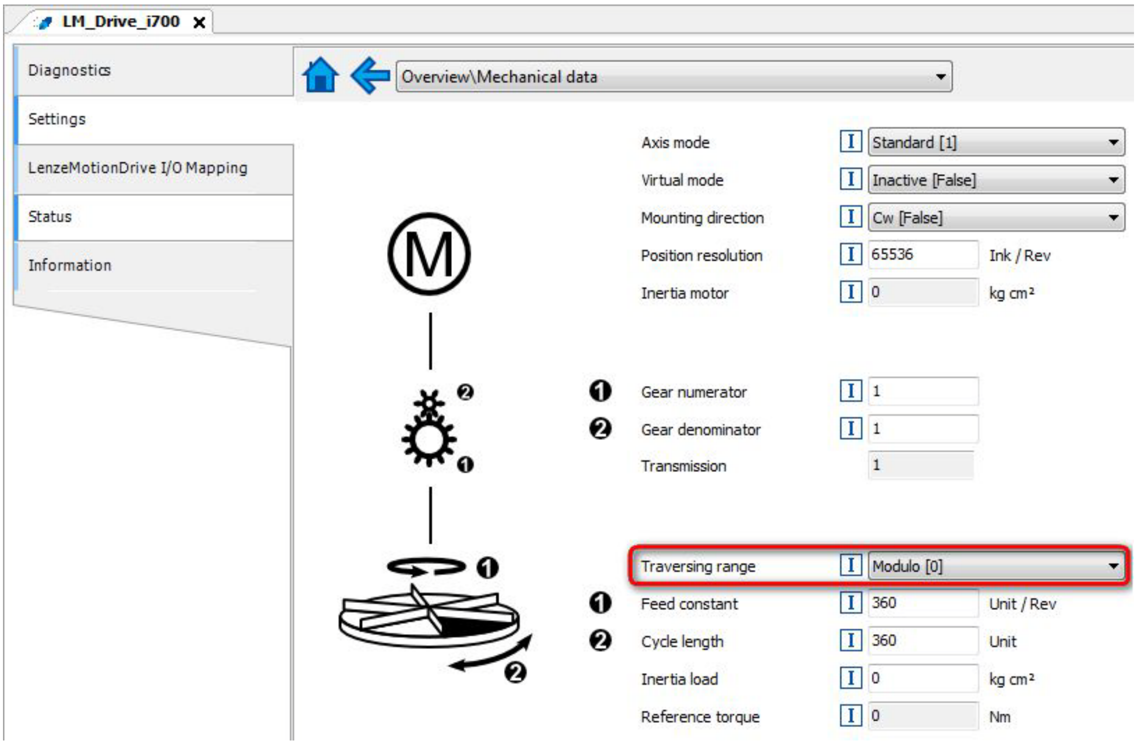
Here you will find detailed information on the **L_MC1P_AxisBasicControl** function block, the **stop function** and the **holding function**.

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

The "Magic Track" technology module only supports rotary axes:

- The master axis has to be a rotary axis and
- the slave axis has to be a rotary axis.

Go to the »PLC Designer« and set the "Modulo" machine measuring system for each axis under the **Settings** tab:




[3-2]

### Setting of the operating mode

The operating mode for the slave axis has to be set to "cyclically synchronous position" (csp) because the axis is led via the master position value.

### Controlled start of the axes

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

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

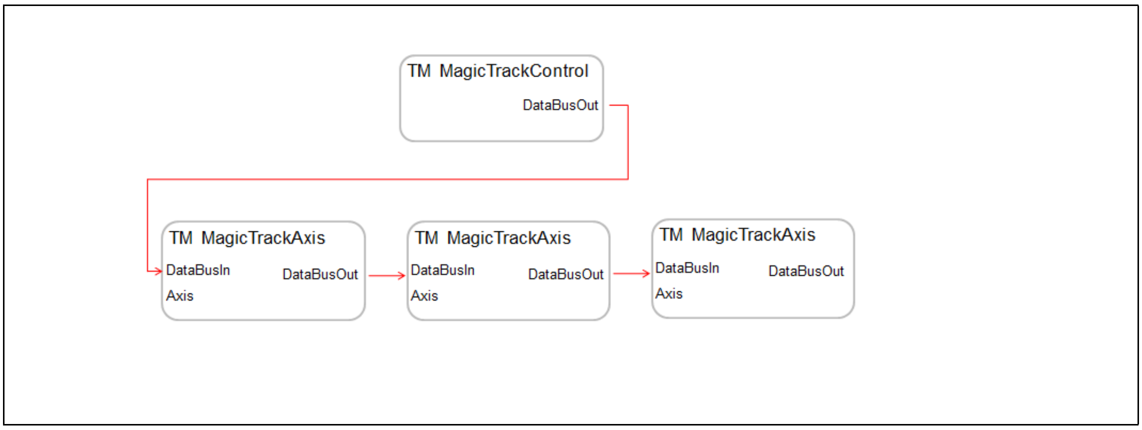
**Example** (📖 27)**:**

1. In the inhibited axis state (*xAxisEnabled* = FALSE), *xJogPos* is set to TRUE.
   - *xRegulatorOn* = FALSE (axis is disabled.
     ==> "READY" state (*xAxisEnabled* = FALSE)
   - *xJogPos* = TRUE (manual jog is to be executed.)
2. Enable axis.
   - *xRegulatorOn* = TRUE
     ==> "READY" state (*xAxisEnabled* = TRUE)
3. Execute manual jog.
   - *xJogPos* = FALSE↗TRUE
     ==> "JOGPOS" state

### Interconnection of the MagicTrack technology module

The MagicTrack technology always comprises a TM MagicTrackControl (Base, State or High) and two or more TMs MagicTrackAxis.

The TM MagicTrackControl takes over the control of the total MagicTrack technology and the management of the parameters. One axis each is connected via the TM MagicTrackAxis. One axis controls one rail. Thus, for instance, threeTMs MagicTrackAxis are required for three rails. All TMs MagicTrackAxis within the technology must be connected to one TM MagicTrackControl via the communication channel of the technology modules.



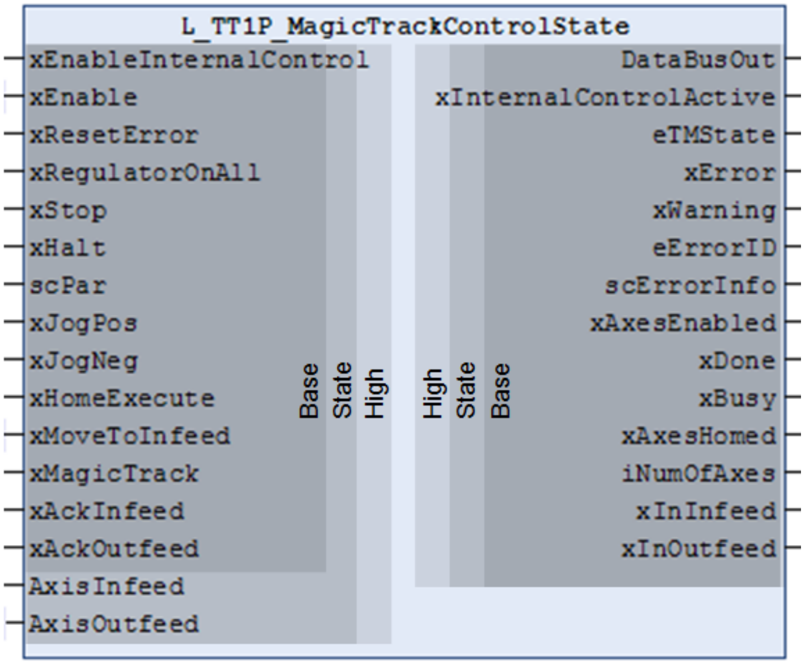

[3-3]

The interconnection of the single TMs MagicTrackAxis should show the mechanical structure. Here, the first TM MagicTrackAxis is connected after the TM MagicTrackControl. The first TM MagicTrack-Axis controls the axis with the first track on the first rail. The first track always travels in front of the second track within the mechanical structure of a MagicTrack. Behind the first TM MagicTrackAxis, the second TM MagicTrackAxis is connected via the communication channel. The second TM MagicTrackAxis controls the second track on the second rail. The second track must be located behind the first track and in front of the third track within the mechanical structure of a MagicTrack.

The call sequence must take place according to the interconnection of the tracks in the motion task.

## 3.3 Function block L_TT1P_MagicTrack[Base/State]

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




[3-4]

### 3.3.1 Inputs

| Designator | Beschreibung | Available in version | | |
|---|---|---|---|---|
| Data type | | Base | State | High |
| xEnableInternalControl<br>BOOL | The internal control of the axis can be selected via the "Internal Control" button (button is visible in the visualisation). | ● | ● | ● |
| xEnable<br>BOOL | Enabling of the module<br>TRUE: The block is executed<br>FALSE: The block is not calculated | ● | ● | ● |
| scControlABC<br>scCtrl_ABC | • Input structure for AxisBasicControl<br>• scControlABC can be used if the state is "Ready", the state machine then changes to the "Service" state<br>• The state change from "Service" back to "Ready" takes place if there are no more requests | ● | ● | ● |
| xResetError<br>BOOL | Reset axis error or software error. | ● | ● | ● |
| xRegulatorOnALL<br>BOOL | Controller enable of all axes that belong to the MagicTrack.<br>This input sets the controller enable for all axes via the communication channel. | ● | ● | ● |
| xStop<br>BOOL | Cancel the active movement for all axes and brake the axis to standstill with the deceleration defined via **scPar.lrStopDec**.<br>The state changes to "STOP". | ● | ● | ● |
| xHalt<br>BOOL | The active movement is cancelled via the rising edge of the xHalt and the axis is braked to standstill with the deceleration defined via **scPar.lrHaltDec**.<br>During the deceleration, the TM changes to the "STOP" state.<br>The TM remains in the STOP state as long as xStop, xHalt is set. | ● | ● | ● |
| scPar<br>STRUCT | Parameter structure for technology module-relevant values<br>Data type depends on the Base/State/High version used | ● | ● | ● |
| scAccessPoints<br>STRUCT | Structure with the access points.<br>Data type depends on the Base/State/High version used | ● | ● | ● |
| xJogPos<br>BOOL | All axes are moved synchronously in a group in positive direction (manual jog).<br>If xJogNeg is also TRUE, the traversing direction selected first remains set. | ● | ● | ● |
| xJogNeg<br>BOOL | All axes are moved synchronously in a group in negative direction (manual jog).<br>If xJogPos is also TRUE, the traversing direction selected first remains set. | ● | ● | ● |

| Designator<br>Data type | Beschreibung | Available in version | | |
|---|---|---|---|---|
| | | Base | State | High |
| xHomeExecute<br>BOOL | All axes are synchronised in a group to the master track. Each rail with a track has its own TP sensor. The master track is moved for a bit more than one round. After a complete round, the detected positions from the TP sensor are offset against the current position of the tracks and directly referenced to the axis.<br>The *xDone* output is set if homing for the axis took place. | ● | ● | ● |
| xMoveToInfeed<br>BOOL | TRUE: Reference run: All tracks travel to the loading stations. The track that is closest to the infeed travels to the loading station to load the trays. The following tracks arrange successively. | ● | ● | ● |
| xMagicTrack<br>BOOL | TRUE: Enable of the automatic operation of Magic-Track.<br>Here, the tracks are loaded in the loading station. Afterwards the tracks travel the unloading station and are unloaded there. Then they return to the loading station again. | ● | ● | ● |
| xAckInfeed<br>BOOL | Relevant for clocked loading of the tracks.<br>With the rising edge of the xAckInfeed input, the loading of a tray is acknowledged. Acknowledgement can be executed anytime, irrespective of whether a track is in the loading station or not.<br>After acknowledgement, it is positioned to the next tray position.<br>After loading all trays, the track automatically travels to the unloading station. | ● | ● | ● |
| xAckOutfeed<br>BOOL | Relevant for clocked unloading of the tracks.<br>With the rising edge of the xAckOutfeed input, the unloading of a tray is acknowledged. Acknowledgement can be executed anytime, irrespective of whether a track is in the unloading station or not.<br>After acknowledgement, it is positioned to the next tray position.<br>After unloading all trays, the track automatically travels to the loading station. | ● | ● | ● |
| AxisInfeed<br>AXIS_REF | Reference to an axis for synchronous loading of the tracks:<br>Master axis in the loading station.<br>When entering the loading station, the track is synchronised to the infeed. It moves synchronously to the infeed. When passing the track end with the infeed position, the synchronisation is interrupted and the track travels to the unloading station. | | ● | ● |
| AxisOutfeed<br>Axis_Ref | Reference to an axis for synchronous unloading of the tracks.<br>Master axis in the unloading station.<br>When entering the unloading station, the track is synchronised to the outfeed. It moves synchronously to the outfeed. When passing the track end with the outfeed position, the synchronous travel is completed and the track travels to the loading station. | | ● | ● |

### 3.3.2 Outputs

| Designator<br>Data type | Beschreibung | Available in version | | |
|---|---|---|---|---|
| | | Base | State | High |
| DataBusOut<br>INTERFACE | Communication link .<br>The communication bus should be connected to the DataBusIn input of the following TM MagicTrackAxis. | ● | ● | ● |
| xInternalControlActive<br>BOOL | The control is activated via the visualisation.<br>(The input xEnableInternalControl is TRUE.) | ● | ● | ● |
| eTMState<br>L_TT1P_States | Current state of the state machine | ● | ● | ● |
| xError<br>BOOL | Status signal: an error in the technology module is pending | ● | ● | ● |
| xWarning<br>BOOL | Status signal: a warning in the technology module is pending | ● | ● | ● |
| eErrorID<br>L_IE1P_Error | Error message if “xError” is active | ● | ● | ● |
| scErrorInfo<br>L_TT1P_scErrorInfo | Error information structure for the more detailed analysis of the error cause | ● | ● | ● |
| scSignalFlow<br>STRUCT | the structure contains IPs, APs, MPs and OPs from the signal flow plan.<br>Data type depends on the Base/State/High version used | ● | ● | ● |
| xAxesEnabled<br>BOOL | All axes in the TM MagicTrack are enabled | ● | ● | ● |
| xAxesHomed<br>BOOL | TRUE: All axes have been referenced ("Reference known") | ● | ● | ● |
| xDone<br>BOOL | TRUE: Request has been completed successfully | ● | ● | ● |
| xBusy<br>BOOL | TRUE: Request is currently active | ● | ● | ● |
| iNumOfAxes<br>INT | Number of axes in the TM MagicTrack | ● | ● | ● |
| xInInfeed<br>BOOL | A tray is in the loading station. | ● | ● | ● |
| xOutInfeed<br>BOOL | A tray is in the unloading station. | ● | ● | ● |

### 3.3.3 Parameters

**L_TT1P_scPar_MagicTrack[Base/State]**

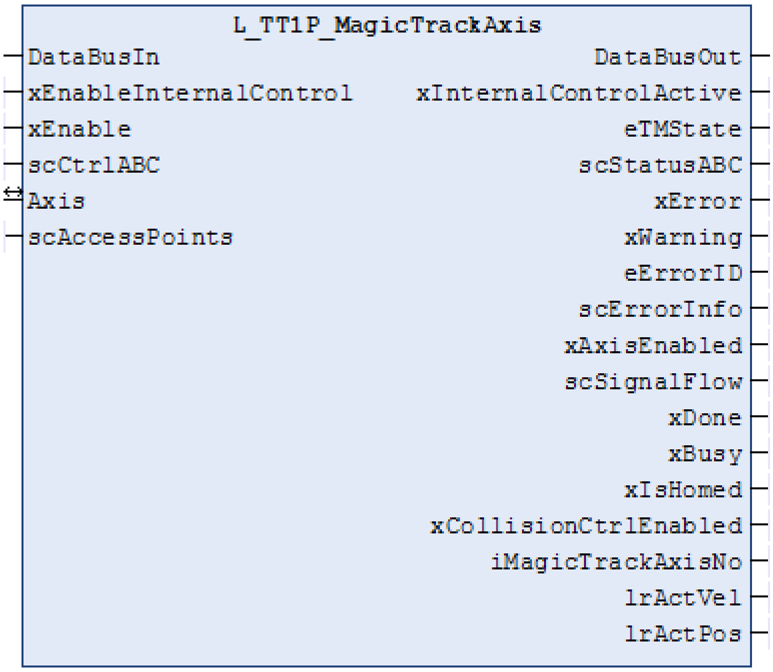
The **L_TT1P_scPar_MagicTrack[Base/State/High]** structure contains the parameters of the technology module.

| Designator / Data type | Beschreibung | Available in version | | |
|---|---|---|---|---|
| | | Base | State | High |
| lrStopDec<br>LREAL | Deceleration for the "Stop" function when hardware/ software limit positions, following error monitoring are triggered<br>• In the unit [r /s]<br>• Initial value := 10000 | ● | ● | ● |
| lrStopJerk<br>LREAL | Jerk for the Stop function and when the HW limit positions, SW limit positions, following error monitoring are triggered<br>• Scaled in the unit [r/s³]<br>• Initial value := 10 000 | ● | ● | ● |
| lrHaltDec<br>LREAL | Deceleration in [unit/s²]<br>• Specification of the velocity variation which is to be used for maximum deceleration to standstill.<br>• Only positive values permissible.<br>• In the unit [r /s]<br>• Initial value := 100 | ● | ● | ● |
| lrJerk<br>LREAL | Limit jerk in the unit [r/s³] in the movement:<br>• Hold<br>• Initial value := 1000 | ● | ● | ● |
| lrJogJerk<br>LREAL | Jerk for manual jog<br>• Unit: units/s3<br>• Initial value: 10000 | ● | ● | ● |
| lrJogVel<br>LREAL | Maximum velocity to be used for manual jog.<br>• Unit: units/s<br>• Initial value: 10 | ● | ● | ● |
| lrJogAcc<br>LREAL | Acceleration for manual jog<br>Specification of the maximum velocity variation which is to be used for acceleration.<br>• Unit: units/s2<br>• Initial value: 100 | ● | ● | ● |
| lrJogDec<br>LREAL | Deceleration for manual jog<br>Specification of the maximum speed variation which is to be used for deceleration to standstill.<br>• Unit: units/s2<br>• Initial value: 100 | ● | ● | ● |
| xCollisionControl<br>BOOL | TRUE: Collision monitoring is activated. | ● | ● | ● |
| lrSafeDist<br>LREAL | Safety distance between tracks which must be observed. | ● | ● | ● |
| eStartMode<br>WORD | 0: Continuing the travel from the current position for each track.<br>1: All tracks start from the feed station | ● | ● | ● |

| Designator<br>Data type | Beschreibung | Available in version | | |
|---|---|---|---|---|
| | | Base | State | High |
| ascContainer<br>ARRAY[1..32] OF L_TT1P_Container; | Parameterising the trays and gaps within one track. | ● | ● | ● |
| eHomeMode<br>L_TT1P_ MagicTrack-HomeMode | Referencing mode<br>Possible settings:<br>0: SetPositionDirect: Direct reference setting. Same function as MC_SetPosition without moving the axis.<br>101: GroupVelMode: Technological homing mode for the TM MagicTrack. In this course, all axes are executed synchronised with velocity. The home positions are set via the TPs of the axes.<br>Initial value: 0 | ● | ● | ● |
| alrHomePosition<br>ARRAY[1..12] OF LREAL | Home position to be set for the desired axis. Homing is started with the xExecuteGroupHome input. | ● | ● | ● |
| ascHomeExtTP<br>ARRAY[1..12] OF LREAL | Transfer of an external touch probe event for the axes<br>• For describing the MC_TRIGGER_REF structure, see the **MC_TouchProbe** function block. | ● | ● | ● |
| lrHomeJerk<br>LREAL | Jerk for homing<br>• Unit: units/s²<br>• Initial value: 10000 | ● | ● | ● |
| lrHomeAcc<br>LREAL | Acceleration for homing<br>Specification of the maximum velocity variation which is to be used for acceleration.<br>• Unit: units/s²<br>• Initial value: 100 | ● | ● | ● |
| lrHomeDec<br>LREAL | Deceleration for homing<br>Specification of the maximum speed variation which is to be used for deceleration to standstill.<br>• Unit: units/s²<br>• Initial value: 100 | ● | ● | ● |
| lrHomeVel<br>LREAL | Maximum velocity to be used for homing.<br>• Unit: units/s<br>• Initial value: 10 | ● | ● | ● |
| eInfeedMode<br>WORD | Mode selection for loading of the tracks<br>0: Clocked infeed (loading)<br>1: Synchronous infeed (loading) | ● | ● | ● |
| eOutfeedMode<br>WORD | Mode selection for unloading the tracks<br>0: Clocked outfeed (unloading)<br>1: Synchronous outfeed (unloading) | ● | ● | ● |
| lrInfeedPos<br>LREAL | Position of the loading station in the unit [units]. | ● | ● | ● |
| lrOutfeedPos<br>LREAL | Position of the unloading station in the unit [units].. | ● | ● | ● |
| lrInfeedSyncLength<br>LREAL | The length of the loading station for the synchronous travel of the tracks | ● | ● | ● |
| lrOutfeedSyncLength<br>LREAL | The length of the unloading station for the synchronous travel of the tracks | ● | ● | ● |

| Designator<br>Data type | Beschreibung | Available in version | | |
|---|---|---|---|---|
| | | Base | State | High |
| lrDelayTimeAckInfeed<br>LREAL | The time from triggering the xAckInfeed input to the start of positioning the track in seconds [s]. | ● | ● | ● |
| lrDelayTimeAckOutfeed<br>LREAL | The time from triggering the xAckOutfeed input to the start of positioning the track in seconds [s]. | ● | ● | ● |
| lrToInfeedJerk<br>LREAL<br>lrToInfeedMaxAcc<br>LREAL<br>lrToInfeedMaxDec<br>LREAL<br>lrToInfeedMaxVel<br>LREAL | Path parameters for travelling from the unloading station to the loading station. | ● | ● | ● |
| lrInInfeedJerk<br>LREAL<br>lrInInfeedMaxAcc<br>LREAL<br>lrInInfeedMaxDec<br>LREAL<br>lrInInfeedMaxVel<br>LREAL | Path parameters for travelling to the loading station. | ● | ● | ● |
| lrToOutfeedJerk<br>LREAL<br>lrToOutfeedMaxAcc<br>LREAL<br>lrToOutfeedMaxDec<br>LREAL<br>lrToOutfeedMaxVel<br>LREAL | Path parameters for travelling from the unloading station to the loading station. | ● | ● | ● |
| lrInOutfeedJerk<br>LREAL<br>lrInOutfeedMaxAcc<br>LREAL<br>lrInOutfeedMaxDec<br>LREAL<br>lrInOutfeedMaxVel<br>LREAL | Path parameters for travelling to the unloading station. | ● | ● | ● |

## 3.4 Function block L_TT1P_MagicTrackAxis




[3-5]

### 3.4.1 Inputs/outputs

| Designator<br>Data type | Beschreibung | Available in version | | |
|---|---|---|---|---|
| | | Base | State | High |
| Axis<br>AXIS_REF | Reference to the axis | ● | ● | ● |

### 3.4.2 Inputs

| Designator<br>Data type | Beschreibung | Available in version | | |
|---|---|---|---|---|
| | | Base | State | High |
| DataBusIn<br>STRUCT | Communication link .<br>The communication bus must be connected to the DataBusIn input of the following TM MagicTrackAxis. | ● | ● | ● |
| xEnableInternalControl<br>BOOL | The internal control of the axis can be selected via the "Internal Control" button (button is visible in the visualisation). | ● | ● | ● |
| xEnable<br>BOOL | Enabling of the module<br>TRUE: The block is executed. The communication bus is activated.<br>FALSE: The block is not calculated | ● | ● | ● |
| scCtrlABC<br>scCtrl_ABC | • Input structure for AxisBasicControl<br>• scControlABC can be used if the state is "Ready", the state machine then changes to the "Service" state<br>• The state change from "Service" back to "Ready" takes place if there are no more requests | ● | ● | ● |

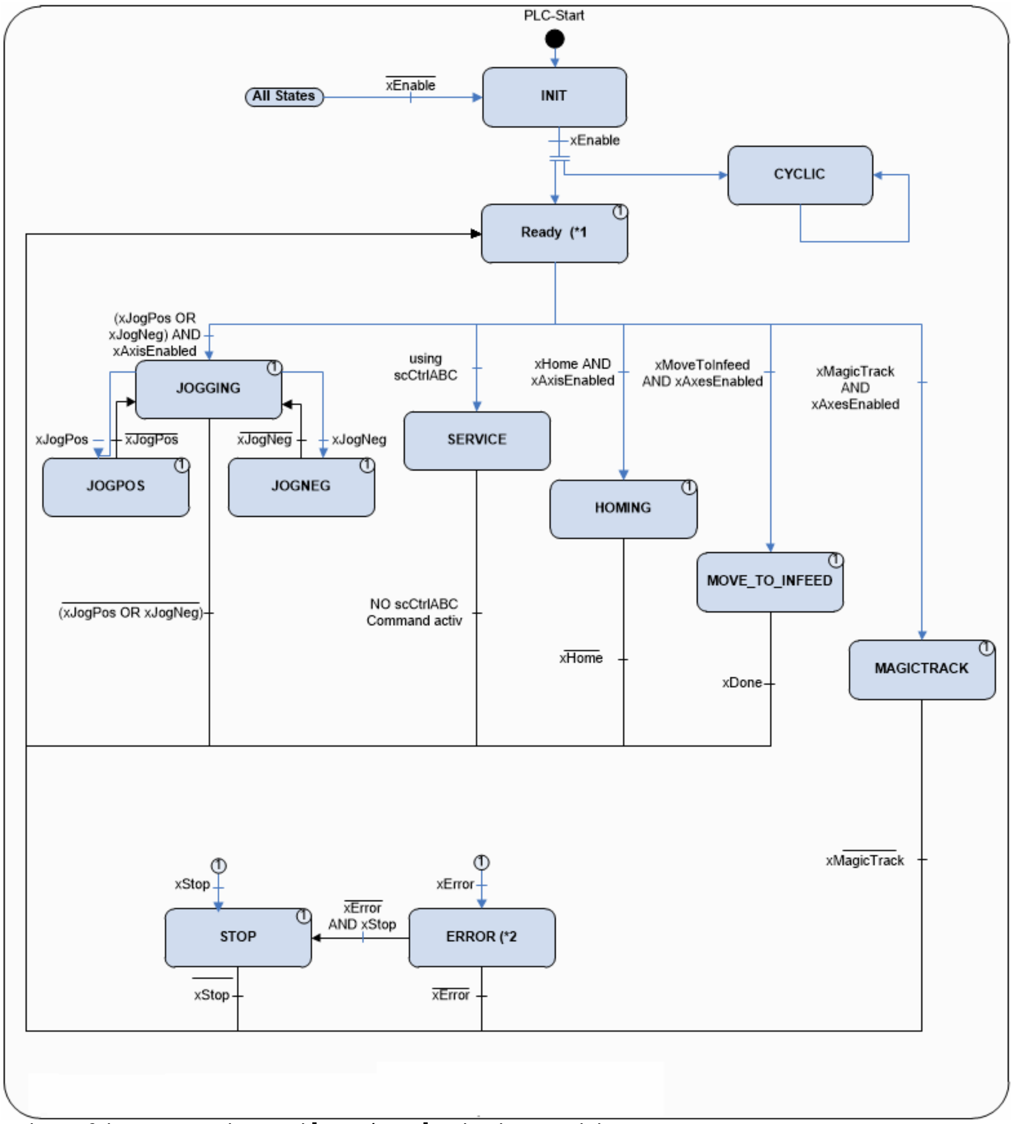
### 3.4.3 Outputs

| Designator<br>Data type | Beschreibung | Available in version | | |
|---|---|---|---|---|
| | | Base | State | High |
| DataBusOut<br>STRUCT | Communication channel. The communication bus must be connected to the slave track at the DataBusIn input of the TM L_TT1P_MagicTrackAxis. | ● | ● | ● |
| xInternalControlActive<br>BOOL | The control is activated via the visualisation.<br>(The input xEnableInternalControl is TRUE.) | ● | ● | ● |
| eTMState<br>L_TT1P_States | Current state of the state machine | ● | ● | ● |
| scStatusABC<br>scStatus_ABC | Structure of the status data of the AxisBasicControl block | ● | ● | ● |
| xError<br>BOOL | Status signal: an error in the technology module is pending | ● | ● | ● |
| xWarning<br>BOOL | Status signal: a warning in the technology module is pending | ● | ● | ● |
| eErrorID<br>L_IE1P_Error | Error message if xError is active or warning signal if xWarning is active | ● | ● | ● |
| scErrorInfo<br>L_TT1P_scErrorInfo | Error information structure for the more detailed analysis of the error cause | ● | ● | ● |
| scSignalFlow<br>STRUCT | the structure contains IPs, APs, MPs and OPs from the signal flow plan.<br>Data type depends on the Base/State/High version used | ● | ● | ● |

| Designator<br>Data type | Beschreibung | Available in version | | |
|---|---|---|---|---|
| | | Base | State | High |
| xAxisEnabled<br>BOOL | Axis is enabled | ● | ● | ● |
| xBusy<br>BOOL | TRUE: Request is currently active | ● | ● | ● |
| xIsHomed<br>BOOL | The axis has been referenced (reference known). | ● | ● | ● |
| iAxisNo<br>INT | The number of the track. | ● | ● | ● |
| lrActVel<br>LREAL | Current velocity<br>• Unit: units/s | ● | ● | ● |
| lrActPos<br>LREAL | Current position<br>• Unit: units | ● | ● | ● |

## 3.5 State machine

### 3.5.1 L_TT1P_MagicTrackControl [Base/State]

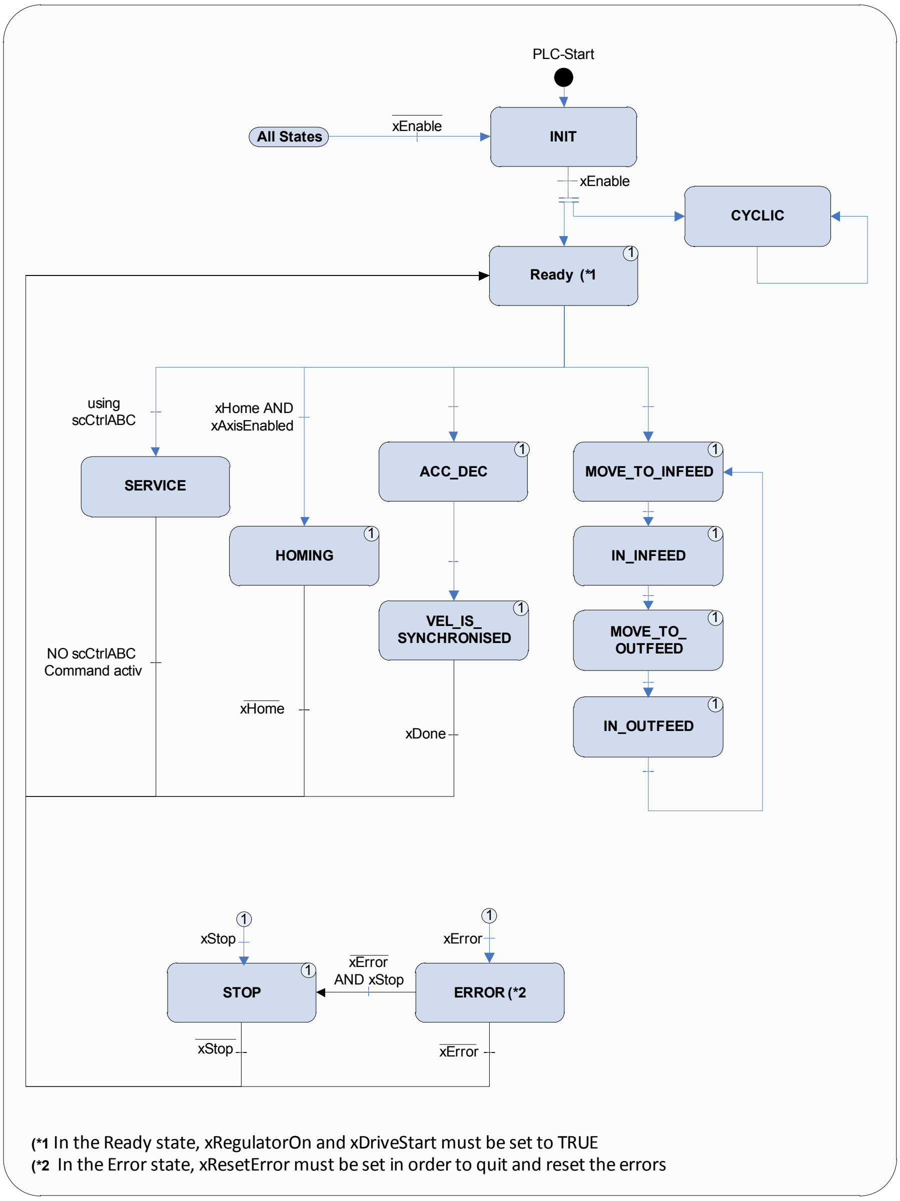



[3-6] State machine of the MagicTrackControl [Base /State] technology module

(*1 In the "Ready" state, xRegulatorOn 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.

### 3.5.2 L_TT1P_MagicTrackAxis




[3-7] State machine of the MagicTrackAxis technology module

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

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

## 3.6 Description of the technology module functions

The individual functions are assigned to the "Base", "State" and "High" versions of the technology module.

### 3.6.1 Parameter setting of a track

The TMs MagicTrackAxis are numbered consecutively. The axis on the first rail is the axis with the number 1. The axis on the second rail is the axis with the number 2. The axis numbers result from the sequence of the interconnected TMs MagicTrackAxis. Each axis moves a track on a rail.

Each track can consist of several trays and gaps. The track in MagicTrack is parameterised via the scPar.ascContainer array with 32 elements of the L_TT1P_MagicTrackContainer structure.

The trays are defined via the parameter setting of the tray lengths scPar.ascContainer[i].lrContainerLegth > 0 and the number of trays of the same length with scPar.ascContainer[i]. iNumOfContainer > 0.

The gaps are defined with the gap length **scPar.ascContainer[i].lrContainerLegth > 0** and with the coding of the number of trays **scPar.ascContainer[i]. iNumOfContainer = 0**. The scPar.ascContainer[i]. dwInfeedMask parameter serves to define the mask for the trays that are to be loaded in the loading station. In the mask, the first bit stands for the first tray. If the corresponding bit is set to TRUE, the track stops at the desired axis in the loading station. The same is parameterised with the scPar.ascContainer[i]. dwOutfeedMask mask for the unloading station. The masks are only evaluated in clocked loading and/or clocked unloading.

In the following example, a track is parameterised in the MagicTrack to illustrate the principle of defining a track with masking the tray.

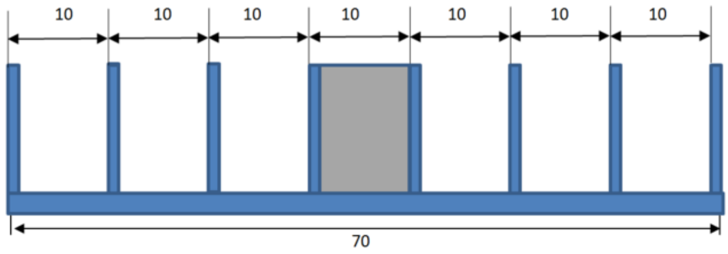
Parameter setting of trays: An array of struct serves to parameterise the trays, gaps and breakpoints of the loading and unloading station.

**L_TT1P_MagicTrackContainer:**

| lrContainerLegth<br>*Length of the tray or the gap* | iNumOfContainer<br>*iNumOfContainer > 0:*<br>Number of the trays with the same length<br><br>iNumOfContainer = 0:<br>this defines a gap | dwInfeedMask<br>Definition of the break-points regarding the tray positions<br><br>1 Stop,<br>0 Release | dwOutfeedMask<br>Definition of the break-points regarding the tray positions<br><br>1 Stop,<br>0 Release |
|---|---|---|---|
| 10 | 3 | 111 | 100 |
| 10 | 0 | 0 | 0 |
| 10 | 3 | 111 | 100 |
| 0 | 0 | 0 | 0 |

In the table, a track has been defined with the following properties.

- A total of 6 trays and one gap are defined.
- The first three trays are defined with 10 length units.
- Between the first three trays and the last three trays, there is a gap of 10 length units.
- The first three trays also have a length of 10 length units.
- The track must stop at each tray in the loading station. In the unloading station, the track must stop at the first and fourth tray.




[3-8] Example of the tray distribution in a track

The track length results from the sum of the lengths of trays and gaps. In this example, the total track length is 70 length units.

### 3.6.2 Jogging

**Precondition**

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

**Execution**

The manual jog is triggered via the inputs xJogPos and xJogNeg via the L_TT1P_MagicTrackControl control module.

All axes are synchronised via the TMs MagicTrackAxis. The "VEL_IS_SYNCHRONISED" state is activated. The TM MagicTrackControl changes to the "Jogging", "JogPos" or "JogNeg" state and the axes are traversed synchronously to each other as long as the input xJogPos or xJogNeg is set.

The synchronous movement of the axes prevent a collision. Thus, no collision monitoring takes place during the inching mode. The travel command cannot be replaced by another jog command. Only if both jogging states are reset, the state machine of the technology modules TM MagicTrack-Control and TM MagicTrackAxis changes to the "Ready" state.

**Parameters to be set**

The parameters for the manual jog are located in the parameter structure L_TT1P_scPar_MagicTrack[Base/State]. (🕮 17)

```
lrJogVel : LREAL := 10;      // Velocity [units/s]
lrJogAcc : LREAL := 100;     // Acceleration [units/s^2]
lrJogDec : LREAL := 100;     // Deceleration [units/s^2]
lrJogJerk : LREAL := 10000;  // Jerk [units/s^3]
```

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

### 3.6.3 Homing

#### Precondition

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

#### Execution

Homing is started via the xHomeExecute input. The state machine changes to the ‚Homing‘ state. When the xHomeExecute input is reset early, homing is not interrupted. After homing is completed, the state machine changes to the "Ready" state. The xDone output is set.

The type of homing is selected via the eHomeMode parameter. In the TM MagicTrack, two homing modes are supported:

- eHomeMode = 0 // SetPositionDirect:
  The "SetPositionDirect“ homing mode is selected via the parameter eHomeMode = SetPosition-Direct. In this mode, the position from scPar.alrHomePosition[i] is directly set in the axis of the ith TM MagicTrackAxis.
- eHomeMode = 101 // GroupVelMode:
  The "Reference all axes synchronously to TP" homing mode is selected via the parameter eHomeMode = 101 „GroupVelMode“.

The precondition for this is that each axis or each rail comes with an installed TP sensor for detecting a mark. The axis that is controlled via the first TM MagicTrackAxis is traversed slightly more than a Modulo cycle length. The other axes are coupled and controlled along with synchronous velocity. After a complete round, the detected positions from the TP sensors of the respective axes are offset against the corresponding scPar.alrHomePosition[i] home positions and written to the axis.

The xDone output is set if homing for the axis took place.

After exiting the homing function for all axes, the state machine of the technology modules TM MagicTrackControl and TM MagicTrackAxis changes to the "Ready" state. The xAxesHomed output is set.

#### Parameters to be set

The parameters for homing are located in the parameter structure L_TT1P_scPar_MagicTrack[Base/State/] . (🕮 17)

```
eHomeMode                       : L_TT1P_MagicTrackHomeMode;
alrHomePosition                 : ARRAY[1..12] OF LREAL;
scHomeExtTP                     : ARRAY[1..12] OF  MC_TRIGGER_REF
```

### 3.6.4 Reference run

#### Precondition

- The technology module is in the "Ready" state.
- All axes within the Magictrack are referenced (xAxesHomed = TRUE).
- All axes of the Magictrack are enabled (*xRegulatorOn* = TRUE).

#### Execution

A reference run is started via the xMoveToInfeed input via the TM MagicTrackControl.

The state machine changes to the "MOVE_TO_INFEED" state. An early reset of the xMoveToInfeed input prevents the travel from being interrupted.

During the reference run, all tracks are moved to the scPar.lrInfeedPos loading station. Each track is traversed from the associated TM MagicTrackAxis independent of other tracks. In the reference run, the collision monitoring is activated. The track that is the closest to the loading station in driving direction (positive direction of rotation of the axis), travels into the loading station. The following tracks are arranged one after another with the safe distance scPar.lrSafeDist.

After exiting the reference run, the state machine changes to the "Ready" state. The xDone output is set.

#### Parameters to be set

The parameters for the reference run are located in the parameter structure L_TT1P_scPar_MagicTrack[Base/State/] . (📖 17)

```
lrSafeDist :LREAL;
lrInfeedPos : LREAL;
lrOutfeedPos : LREAL;
lrToInfeedJerk : LREAL;
lrToInfeedMaxAcc: LREAL;
lrToInfeedMaxDec: LREAL;
lrToInfeedMaxVel: LREAL;
```

### 3.6.5 Collision monitoring

#### Precondition

- The technology module is in the "Ready" state.
- All axes within the Magictrack are referenced (xAxesHomed = TRUE).
- The collision monitoring is activated via the parameter scPar.xCollisionControl = TRUE.

#### Execution

The collision monitoring serves to monitor the tracks and their distances between each other. As soon as the safe distance scPar.lrSafeDist between two tracks is violated, the collision monitoring triggers an error status. Here, all axes are braked to standstill via the scPar.lrStopDec ramp. The error message eErrorID = 17138 (CollisionDetected) is output at the TM.

Usually, the tracks are moved independent of each other. The safe distance scPar.lrSafeDist between the tracks is considered via the TM MagicTrackAxis in the profile calculation so that it will not be exceeded.

Thus, the collision monitoring is a safety mechanism that is to prevent a collision in the event of an error of the profile. The collision monitoring can be deactivated via the parameter scPar.xCollisionControl = FALSE.

#### Parameters to be set

The parameters for the reference run are located in the parameter structure L_TT1P_scPar_MagicTrack[Base/State/] . (📖 17)

```
lrSafeDist :LREAL;
xCollisionControl : BOOL;
```

### 3.6.6 Automatic operation of the MagicTrack technology module

**Precondition**

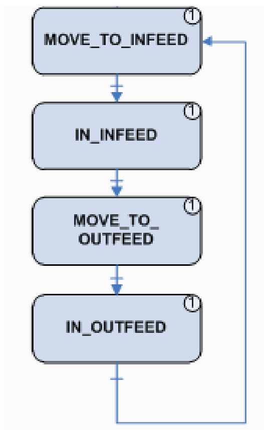
- The technology module is in the "Ready" state.
- All axes within the Magictrack are referenced (xAxesHomed = TRUE).
- All axes of the Magictrack are enabled (*xRegulatorOn* = TRUE).

**Execution**

The automatic operation is started via the xMagicTrack input := TRUE via the TM MagicTrackControl. A reset of the xMagicTrack input = FALSE stops the operation. All axes are braked to standstill via the scPar.lrHaltDec ramp.

In the automatic operation, the tracks are loaded via the loading station and emptied in the unloading station. A travel via the loading station and unloading station is called MagicTrack cycle.

A MagicTrack cycle consists of four phases:




[3-9] A MagicTrack cycle consists of four phases

MOVE_TO_INFEED:

In the first phase, the track is traversed to the loading station. In this phase, the parameters lrToInfeedMaxVel, lrToInfeedMaxAcc, lrToInfeedMaxDec and lrToInfeedJerk are used for profile calculation. If the loading station is already occupied by another track, the track is synchronised with a safe distance scPar.lrSafeDist behind the track running ahead. The train cancels the synchronisation to the track running ahead and travels into the loading station as soon as it gets free.

IN_INFEED:

In the Base version, clocked loading of the trays in a track is supported. Here, the tray that is enabled via the parameter setting of the lrInfeedMask mask for loading, is stopped in the loading station. The xAckInfeed input acknowledges the loading process of the tray. After the rate time scPar.lrDelayTimeAckInfeed has elapsed, the track is positioned to the next loading of a tray.

If the loading of the trays has already been acknowledged in advance via the rising edge of the xAckInfeed input, the track waits the rate time scPar.lrDelayTimeAckInfeed and is then started to travel to the next loading position of the tray.

For the positioning processes in the loading station, the parameters lrInInfeedMaxVel, lrInInfeedMaxAcc, lrInInfeedMaxDec and lrInInfeedJerk are used for profile calculation.

When the loading of all trays is completed, the track travels to the unloading station.

MOVE_TO_OUTFEED:

For travelling from the loading station to the unloading station, the profile parameters lrToOufeedMaxVel, lrToOutfeedMaxAcc, lrToOutfeedMaxDec and lrToOutfeedJerk are used. If the unloading station is already occupied by another track, the track is synchronised with a safe distance scPar.lrSafeDist behind the track running ahead. The track cancels the synchronisation to the track running ahead and travels into the unloading station as soon as it gets free.

IN_OUTFEED:

In the Base version, the trays are unloaded in the unloading station in a clocked way. Here, the tray that is enabled via the parameter setting of the lrOutfeedMask mask for unloading, is positioned in the unloading station. The xAckOutfeed input acknowledges the unloading process of the tray. After the rate time scPar.lrDelayTimeAckOutfeed has elapsed, the tray is positioned to the next unloading of a tray.

If the unloading of the trays has already been acknowledged in advance via the rising edge of the xAckInfeed input, the train waits the rate time scPar.lrDelayTimeAckInfeed. Afterwards it is started to travel to the next unloading of a tray.

For the positioning processed in the loading station, the parameters lrInOutfeedMaxVel, lrInOutfeedMaxAcc, lrInOutfeedMaxDec and lrInOutfeedJerk are used for profile calculation.

If the unloading of all trays is completed, the track travels to the loading station. For travelling from the unloading station to the loading station, the profile parameters lrToInfeedMaxVel, lrToInfeedMaxAcc, lrToInfeedMaxDec and lrToInfeedJerk are used.

### Parameters to be set

The parameters for the reference run are located in the parameter structure L_TT1P_scPar_MagicTrack[Base/State/]. (📖 17)

```
lrSafeDist :LREAL;
xCollisionControl : BOOL;
ascContainer :ARRAY[1..32] OF L_TT1P_Container;
lrInfeedPos:LREAL;
lrOutfeedPos:LREAL;
lrToInfeedJerk :LREAL;
lrToInfeedMaxAcc:LREAL;
lrToInfeedMaxDec:LREAL;
lrToInfeedMaxVel:LREAL;
lrInInfeedJerk:LREAL;
lrInInfeedMaxAcc:LREAL;
lrInInfeedMaxDec:LREAL;
lrInInfeedMaxVel:LREAL;
lrToOutfeedJerk:LREAL;
lrToOutfeedMaxAcc:LREAL;
lrToOutfeedMaxDec:LREAL;
lrToOutfeedMaxVel:LREAL;
lrInOutfeedJerk:LREAL;
lrInOutfeedMaxAcc:LREAL;
lrInOutfeedMaxDec:LREAL;
lrInOutfeedMaxVel:LREAL;
```

## 3.7 Function extensions of the MagicTrackControlState technology module

### 3.7.1 Extension of the automatic operation

**Precondition**

- The technology module is in the "Ready" state.
- All axes within the Magictrack are referenced (xAxesHomed = TRUE).
- The collision monitoring is activated via the parameter scPar.xCollisionControl = TRUE.

**Execution**

The scPar.eInfeedMode parameter serves to define the type of loading a track.

- eInfeedMode = 0 (pulse)
  The parameter setting scPar.eInfeedMode = 0 serves to select the clocked loading in the loading station. Here, the loading of a track takes place as already described in the Base version.
- eInfeedMode = 1 (sync)
  The parameter setting scPar.eInfeedMode = 1 serves to select the synchronous loading of a track in the loading station. When the track enters the loading station, the track is already synchronised to the AxisInfeed axis via the velocity. The length of the loading station is defined via the scPar.lrInfeedSyncLength parameter. The track is released from the synchronous travel and moves on to the unloading station as soon as the track end has left the end of the loading station.

The scPar.eOutfeedMode parameter serves to define the type of unloading a track.

- eOutfeedMode = 0 (pulse)
  The parameter setting scPar.eOutfeedMode = 0 serves to select the clocked unloading of the track in the unloading station. Here, the unloading of the track takes place as already described in the Base version.
- eOutfeedMode = 1 (sync)
  The parameter setting scPar.eOutfeedMode = 1 serves to select the synchronous unloading of a track in the unloading station. When the track enters the unloading station, the track is synchronised to the AxisOutfeed axis via the velocity. The length of the unloading station is defined via the scPar.lrInfeedSyncLength parameter. The length of the station defines the position of the release for the synchronous travel. The track is released from the synchronous travel and moves on to the loading station as soon as the track end has left the end of the unloading station.

**Parameters to be set**

The parameters for the reference run are located in the parameter structure L_TT1P_scPar_MagicTrack[Base/State/]. (📖 17)

```
eInfeedMode; WORD ;                    //0 pulse, 1 sync
eInfeedMode; WORD ;                    //0 pulse, 1 sync
```

## 3.8 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** |
| 1xTM MagicTrackContorlBase<br>2xTM MagicTrackAxis | xEnable := TRUE; xRegulatorOn := TRUE; xMagicTrack := TRUE; | 185 s | 345 s |
| 1xTM MagicTrackContorlState<br>2xTM MagicTrackAxis | xEnable := TRUE; xRegulatorOn := TRUE; xMagicTrack:= TRUE; | 185 s | 359 s |

FEEDBACK

## Your opinion is important to us

We have created these instructions to the best of our knowledge with the objective of providing you with the best possible support when handling our product.

Perhaps we have not succeeded in achieving this objective in every respect. If you notice this, please send your suggestions and points of criticism in a short e-mail to:

feedback-docu@Lenze.de

Thank you for your support.

*Your Lenze documentation team*

Lenze Automation GmbH
Postfach 10 13 52, D-31763 Hameln
Hans-Lenze-Straße 1, D-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, D-32699 Extertal
Germany
008000 24 46877 (24 h helpline)
+49 5154 82-1112
service@lenze.com