TM Pilot

Manual





LinMot® Pilot



October 2026

Doc.: 0185-1188-E_1V4_MA_TM-Pilot

© 2026 NTI AG
This work is protected by copyright.
Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, microfilm, storing in an information retrieval system, not even for didactical use, or translating, in whole or in part, without the prior written consent of NTI AG.

LinMot® is a registered trademark of NTI AG.

Note
The information in this documentation reflects the stage of development at the time of press and is therefore without obligation. NTI AG reserves itself the right to make changes at any time and without notice to reflect further technical advance or product improvement.

NTI AG
LinMot
Bodenaeckerstrasse 2
CH-8957 Spreitenbach

Tel.: +41 56 419 91 91
Fax.: +41 56 419 91 92
Email: office@LinMot.com
Homepage: www.LinMot.com

Table of Contents

1 Overview 7
2 Quick Start Guide 8
3 WebVisu 9
3.1 Title Bar 10
3.2 Operation 11
3.3 Events 13
3.4 Control 13
3.4.1 Axis Control 14
3.4.2 Quick Control 15
3.5 Program 15
3.6 Setup 17
3.6.1 System Config 18
3.7 Info 19
3.8 Navigation Menu Bar 20
4 Program Commands 20
4.1 Pilot Commands 20
4.1.1 Go To 20
4.1.2 Label 20
4.1.3 If ... Go To 21
4.1.4 If Motions Done 22
4.1.5 Stop Motions 22
4.1.6 User Message 22
4.1.7 Comment 23
4.1.8 Pause on Request 23
4.1.9 Complete on Request 24
4.2 Timing & Delay 24
4.2.1 Wait Time 24
4.2.2 Wait Logic 25
4.2.3 Wait User 25
4.2.4 Wait Motions 26
4.3 Program Control 27
4.3.1 Call Program 27
4.3.2 Run Program 27
4.3.3 Stop Program 28
4.4 Inputs & Outputs 28
4.4.1 Set Output 28
4.4.2 Set Output by logic 29
4.5 Variables 29
4.5.1 Set Variable 29
4.5.2 Set Variable by logic 29
4.5.3 Select Dataset 30
4.6 Axes Motion 30
4.6.1 Move Linear Axes 30
4.6.2 Move Rotary Axes 30
4.6.3 Set Force/Torque 31
4.7 Robotic Motion 31
4.7.1 Set Coordinate System 31

4.7.2 Set Tool 32
4.7.3 Move Robotic 32
4.7.4 Move Circular 33
4.8 LinMot Grippers 34
4.8.1 Move Gripper 34
4.8.2 Grip by Position 35
4.8.3 Grip by Force 36
4.8.4 Ungrip 37
4.8.5 Gripper Monitoring 37
4.8.6 Set Grip Force 38
4.9 Conveyor Tracking 38
4.9.1 Control Target Detection 38
4.9.2 Set Active Target 39
4.9.3 Wait for Active Target 39
4.9.4 Sync to Active Target 39
4.9.5 Move Synchronized 40
4.9.6 Desync from Conveyor 41
4.9.7 Clear Target from Buffer 43
4.9.8 Clear Buffer 43
4.10 Capping 43
4.10.1 Find Seating Point 43
4.10.2 Find Thread Start 44
4.10.3 Cap to Angle 45
4.10.4 Cap to Torque 46
4.10.5 Tare 47
4.10.6 Check Positions 47
4.10.7 Check Standstill 48
4.10.8 Check Force/Torque 49
4.11 Common Inputs 50
5 Applications 53
5.1 LinMot Grippers 53
5.1.1 Control 53
5.1.2 Setup 54
5.1.3 Widgets 54
5.2 Conveyor Tracking 54
5.2.1 Control 54
5.2.2 Connecting and Configuring a Conveyor 55
5.3 Capping 59
5.3.1 Control 59
5.3.2 Setup 60
6 System 61
6.1 System States 61
6.2 Program Execution 62
6.2.1 Program Interruption 62
6.2.2 Stopping Behaviour of Axes 64
6.2.3 Program Manager 64
6.2.4 Program Pre-Compiler 65
6.3 Data References 65
6.3.1 User Variables 66

6.3.2 Dataset Variables 67
6.3.3 Program Variables 67
6.3.4 Axis Parameters 67
6.3.5 In- / Outputs 68
6.3.6 Process Outputs 68
6.3.7 Monitoring / Parameter Channels 68
6.3.8 Motion Defaults 68
6.3.9 PLC Interface 69
6.3.10 High Speed Variables 69
6.4 Kinematics Wizard 69
6.4.1 Selection 69
6.4.2 Add-Ons 70
6.4.3 Assignment 70
6.4.4 Directions & Limits 71
6.4.5 Conveyor 72
6.4.6 Channels 72
6.4.7 In- & Outputs 72
6.4.8 Overview 72
6.5 User Management 73
6.5.1 User Levels 73
6.5.2 Initial Users 73
6.5.3 Login and Switch User 73
6.5.4 Logout and Lock 73
6.5.5 Change Password 74
6.5.6 Add User 74
6.5.7 Delete User 74
6.5.8 Change Userlevel 74
6.6 IO Mapping 75
6.7 Event Handling 76
6.7.1 Event Types 76
6.7.2 Event Sources 77
6.7.3 Log Files 78
7 Fieldbus Connection 78
7.1 Functionality and Limitations 78
7.2 Setup 79
7.2.1 Configuration in LinMot-Pilot 79
7.3 Data Structure 80
7.3.1 Inputs 80
7.3.2 Outputs 81
7.4 Examples 83
7.4.1 Profinet with Siemens PLC 83
7.4.2 EtherNet/IP with Rockwell Automation PLC 84
8 Ethernet over EtherCAT 85
9 Troubleshooting 86
9.1 Web Visu Issues 86
10 Licenses 86
11 Appendix 87
11.1 Events 87
11.1.1 List of System Events 87

11.1.2 List of Program Events 88
11.1.3 List of generalized Application Events 89
11.1.4 List of Axes Motion Events 90
11.1.5 List of Robotics Events 90
11.1.6 List of Conveyor Events 90
11.1.7 List of Gripping Events 90
11.1.8 List of Capping Events 91
11.1.9 List of Cognex Events 91
11.1.10 List of LinMot Vision Events 91
11.2 Installation without TM-Manager 91

1 Overview

This software application runs on a LinMot Technology Module and can effortlessly configure and control multi-axis kinematics.

The Web-Visualization is used to visualize and edit the program sequence. It can be accessed over the network with the help of a web-browser.

System key parameters:

• Cycle time	2 ms
• Maximum number of axis	8
• Interface Communication (SPS <-> Technology Module)	Profinet, EtherNet/IP



Note: A Technology Module with Softmotion and CNC/Robotics is required to use the functionality of this software.
(ordering number: 0150-6553)



Note: This Software is not a substitute for safety devices or other protective monitoring devices. Use according safety equipment.

2 Quick Start Guide

This section provides a detailed procedure for commissioning the Technology Module TM01 and LinMot-Pilot. Follow these instructions carefully to ensure proper installation and configuration.

System Overview:
The TM01 acts as the central control unit and connects to up to 8 LinMot drives via EtherCAT. A PC is connected to ETH2 for configuration and access to . Optional PLC integration is possible via Profinet or EtherNet/IP (ETH0).

Prerequisites:

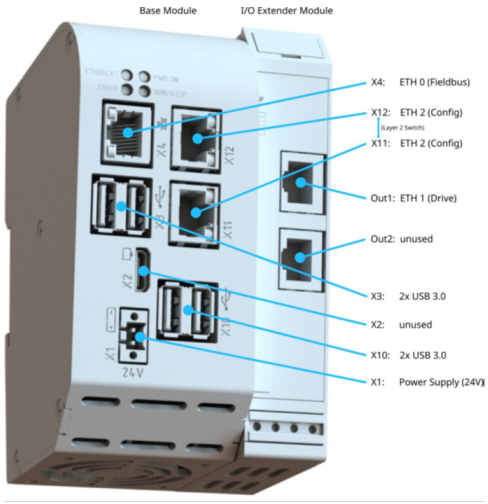
Before starting, verify that the following components and tools are available:

Software:

- LinMot-Pilot Operating System (OS)
- Web browser (Chrome or Edge recommended)

Hardware:

- Regulated 24 VDC power supply (5 A, slow-blow fuse)
- Technology Module TM01
- The license dongle
- LinMot Drives (e.g., C1250-MI or E1450-MI)
- PC connected to ETH2 (X11/X12) of the TM01 directly or by the same network



Hardware Installation:

- Mount TM01 on a 35 mm DIN rail in vertical position. Maintain minimum clearance: 20 mm horizontally, 50 mm vertically.
- Plug the license dongle into a USB port on the TM01.
- Connect 24 VDC to connector X1 and protect the supply with a 5 A slow-blow fuse.
- Connect EtherCAT drives (C1250-MI or E1450-MI) to ETH1 (Out1) and daisy-chain additional drives (X18 → X17).
- Connect PC to ETH2 (X11/X12) directly or by a network for LinMot-Pilot access.
  Optional: Connect Profinet or EtherNet/IP controller to ETH0 (X4).

Software Installation:

The easiest way to install software on the Technology Module and configure it is by using the TM-Manager tool. A full description is available in the Installation Guide for the Technology Module in chapter 6.
To install the application software without the TM-Manager, check Installation without TM-Manager.

Kinematic Setup:

- Select the kinematic layout matching your application.
- Assign drive numbers according to wiring order.
- Define homing procedure and initial positions.
- Save configuration and reboot.

Attention: After reboot, axes can automatically power on or execute homing, based on the configuration in *Setup → System Configuration → General*.

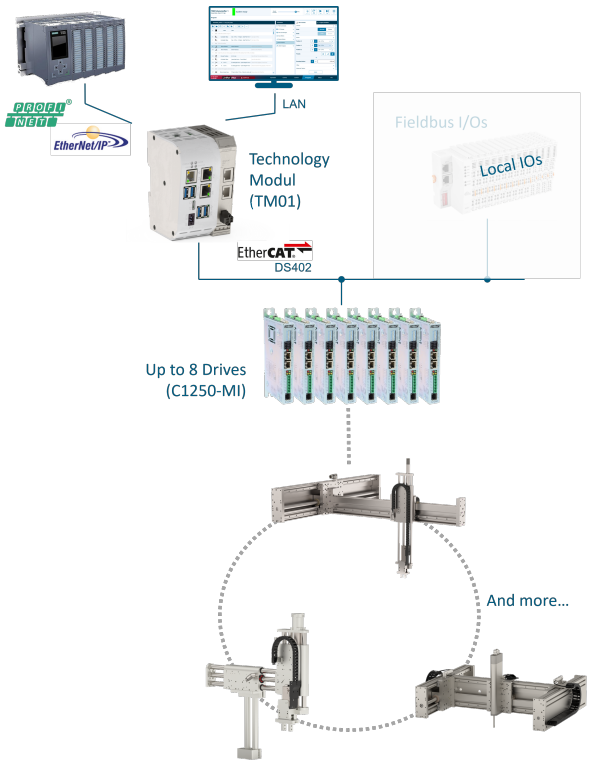
Operating LinMot-Pilot:

- Operation Page: Monitor system status and interact with widgets.
- Events Page: Shows active and most recent events.
- Control Page: Jog axes manually, execute macros.
- Program Page: Create sequences using predefined commands (e.g., Move Axis, Grip, Wait).
- Setup Page: Configure system parameters, user management, application settings.

- Info Page: Information about the installed hardware and software components.

Safety Notes:

- Perform only documented changes.
- Always create backups before firmware or OS updates.
- Never expose Ethernet ports directly to the Internet; use a firewall.
- After kinematic changes, drives will be reset to factory defaults—backup drive configuration first.



3 WebVisu

The Web Visualization (Webvisu) allows you to conveniently monitor and control your system directly in a web browser, without the need for additional software. It can be accessed under the link "http://<ip-address>". (The default IPv4 address is printed on the device label.)

The Webvisu screen is divided into three main areas:

- **Title Bar (top of the screen)**
  The Title Bar consists of two lines. It displays the current status of the system (the system state with a short explanatory message, as well as active messages) and provides the main controls for system operation: the mode toggle (Automatic/Manual), the speed-related controls (Speed Override / Jogging Speed) and the main control buttons (for example Resolve, Start and Pause). Details are described in the chapter Title Bar.
- **Main Workspace (center of the screen)**
  The Main Workspace displays the content of the active page and enables direct interaction with the system. It can be switched between the six main pages: **Operation**, **Events**, **Control**, **Program**, **Setup** and **Info**. On the pages Control, Events, Setup and Info, a sidebar on the left-hand side of the workspace lists the subpages of the page. The breadcrumb in the Title Bar shows the current position in the structure at all times.

- **Navigation Menu Bar (bottom of the screen)**
  The Navigation Menu Bar allows you to switch between the six main pages quickly and easily. In addition, it displays the current date and time, the company logo, the name of the active user (with a user menu for quick actions such as Lock or Logout) and the language selection. Details are described in the chapter Navigation Menu Bar.

The following sections describe these areas in detail and explain how to use the individual pages.

3.1 Title Bar

The Title Bar is the central control element of your system interface, located at the top of the Webvisu and visible on all pages. It provides both status information and direct control options, making it an essential component for system interaction. The Title Bar consists of two lines: the control line at the top and the message line at the bottom. Each line is divided into sections, each serving a specific purpose.

System Name:

On the far left, the name of the current system is displayed. Directly below the name, the currently selected dataset (for example "Set 1") is shown. The selected dataset defines the set of values that the system is currently working with. It can be changed on the DataSet Variables page.

System State:

Next to the system name is the system state indicator. A colored dot and the state name (for example "Uninitialized", "Ready" or "Running") display the current state of the system. Directly below the state name, a short message explains the current situation or the next step to take, for example how to proceed when the system has to be brought to the ready position. If the system enters an error state, the indicator displays the error state. For a detailed description of all system states and the event handling, refer to the chapter System.

Mode Toggle:

The third section contains the mode toggle button, which allows you to switch between Automatic Mode and Manual Mode. In Automatic Mode, the system executes the programs autonomously. In Manual Mode, the system can be operated manually, for example by jogging the axes on the Control page. The selected mode determines the controls that are shown in the next section.

Mode-Specific Controls:

This section dynamically adapts to the selected mode and the active page:

- **Automatic Mode:** A Speed Override control is displayed. It consists of a value field (in %) and a slider and lets you adjust the operating speed of the system by applying a factor, without interrupting automatic operation.
- **Manual Mode on the Control page:** An Increment toggle and a Jogging Speed control (value field and slider) are displayed. The Jogging Speed defines the speed that is used for manual axis movements. The Increment toggle switches between continuous jogging (toggle off, the axis moves as long as the jogging command is active) and incremental jogging (toggle on, the axis moves in defined increments).
- **Manual Mode on the other pages:** The Speed Override control is displayed, as in Automatic Mode.

Main Controls:

On the far right of the control line is the main control section, which contains the primary buttons for system operation. In Manual Mode, additional buttons are displayed to the left of the main buttons. The state of each button (active or grayed out) always depends on the current system state: only active buttons can be used. The buttons are described in detail below:

Button	Name	Description
	Abort	Immediately stops the current process. This button is only displayed in Manual Mode.

	Step	Executes the next step in the program. This button is only displayed in Manual Mode.
	Reset	If a reset program has been set in the System Configuration, a Reset button is also shown in Manual Mode. It executes the reset program to return the system to its initial position.
	Resolve	Resolves the current state of the system. When the state message asks you to resolve, for example after an error or during the initialization, click this button to continue, for example to check the ready position.
	Start	Starts the system. The button is only active when the system is in the Ready state.
	Pause	Pauses the running program. The button is active while the system is in a running state.
	Request Pause	Requests the system to pause at a safe point. The running program is paused as soon as a "Pause on Request" row is executed.
	Request Completion	Requests the system to complete the current cycle. The running program is stopped as soon as a "Complete on Request" row is executed.

Message Line:

The bottom line of the Title Bar consists of two parts:

- **Breadcrumb (left):** On the pages with subpages (Control, Events, Setup and Info), a breadcrumb shows the current position in the page structure, for example "Control → Axis Control → Axial". The path segments before the current page are clickable, so you can jump back to a higher level of the structure. On the main pages, the name of the active page is displayed.
- **Message field (right):** Displays the active messages of the system, for example the text of an active event. If no message is active, the field shows "No messages".

3.2 Operation

The Operation page is the main dashboard for monitoring and controlling the system. It consists of one or more widget pages. Each widget allows you to monitor values or control the system directly, so you can build a dashboard that fits exactly to the way you operate the machine.
If no widget has been added yet, a hint is displayed on the page explaining how to enter Edit Mode in order to add widgets.

Widget Pages (Tabs):

If multiple widget pages are available, they are listed as tabs at the top of the Operation page. Click a tab to switch to the corresponding widget page. In Edit Mode, you can manage the tabs:

- Add a new page: click the plus (+) symbol next to the existing tabs.
- Rename a page: click on the current tab name.
- Delete a page: click the cross next to the tab name.

Edit Mode:

To configure the Operation page, click the Edit button (pencil icon) in the top-right corner of the workspace. Users belonging to the Service or Administrator user groups can enter Edit Mode. When Edit Mode is activated, a grid becomes visible beneath the tabs, providing a clear structure for placing widgets.
To leave Edit Mode, click the Edit button again.

Adding Widgets:

1. In Edit Mode, click the plus (+) symbol on an empty cell of the grid.
2. The widget selection dialog opens, displaying all available widget types.
3. Select the widget you want to add and confirm your selection. The widget is placed on the grid.

Important: Widgets can only be added if there is enough space available on the grid.

Removing Widgets:

In Edit Mode, click the X in the top-right corner of the widget you want to delete.

Configuring Widgets:

Every widget is configured individually. In Edit Mode, click the widget (or its settings button) to open its settings dialog. The dialog shows the parameters of the widget (for example the source variable, the label, the format or the unit) and a live preview of the widget at its original size. Confirm the settings with the button in the footer of the dialog.

Available Widgets:

The following widget types are available on the Operation page. In the following a short description of the available Widgets is given:

Widget	Description
Button	A tile with a primary button. One tap (or click) triggers the action that is assigned to the button.
Dataset Selection	A dropdown with which you select the active dataset from the available datasets.
Display Value	Displays a value from the system. Format and unit can be configured. The display changes its appearance when the value reaches a warning or alarm threshold.
Gauge	An analog display (arc with needle and numeric value) for continuously changing values, for example force, speed or position.
Horizontal Bar	A horizontal bar with a min/max scale that visualizes a value within a range, including warning and alarm areas.
Horizontal Slider	A horizontal slider with which you adjust a value directly, for example a speed factor.
Lamp	A status lamp that shows a state with a fixed color (off, green, amber or red). The state-to-color mapping is configured in the widget settings.
Power Home All	Three stacked buttons: **Power** (enables/disables the axes), **Home** (homes a single axis) and **Home All** (homes all active axes).
Program Selection	A list of the programs on the system. Selecting an entry sets the program on which the system operates.
Quick Run Program	A program list with a start action: select a program and start it directly with one additional tap.
Segmented Button	A button divided into two to four segments. Clicking a segment selects that option, for example switching between position and force control.
Toggle	An on/off switch for a Boolean variable.
Vertical Bar	A vertical bar with a min/max scale that visualizes a value within a range, including warning and alarm areas.
Vertical Slider	A vertical slider with which you adjust a value directly.

Write Value	An input field with which you write a numeric value into the system.

Widgets that are specific to certain applications are described in the chapter Applications.

3.3 Events

The Events page provides an overview of all messages of the system. The available submenus are listed in the sidebar on the left-hand side of the page:
- **Active:** Displays all currently active events, for example active errors and warnings.
- **History:** Displays the past events (the event archive).

Click an entry in the sidebar to open the corresponding table. Both submenus display the events in a table with the following columns:

Column	Content
Type	Category of the event (Error, Warning or Info). Depending on the category, the row is highlighted with the corresponding color.
Arrived	Timestamp when the event occurred.
Gone	Timestamp when the event was resolved. (Only on the History page.)
Source	Origin of the event (for example an axis, a drive or the system).
Number	Event code (for example AX1-3).
Text	Description of the event.
Help	Instructions on how to resolve the issue. (Only on the Active page.)

Filtering:

On the right-hand side of the page title bar, three filter buttons are displayed. Each button shows the number of events in that category, for example "0 errors" or "2 infos". Clicking a button toggles the filter on or off. Multiple filters can be active at the same time: for example, activating Errors and Warnings shows only these two categories.

When you view the History submenu, an additional button is displayed in the title bar: **Delete Archive**. Clicking this button permanently deletes the entire archive of past events.

3.4 Control

The Control page provides the controls for the individual components of your system. The sidebar on the left-hand side lists the available control functions, grouped into three main sections: **Axis Control**, **System Control** and **Application Control**. Click a section to expand it, then click an entry to open the corresponding subpage.

Axis Control:

This section provides the manual control and monitoring of the axes in the system:
- **Axial** — Control and monitoring of the single (axial) axes of the system.
- **Robotic** — Control and monitoring of the axes arranged in robotic groups.

Details are described in the chapter Axis Control.

System Control:

This section provides the controls for the general system functions:
- **Quick Control** — A configurable control interface based on widgets (like the Operation page) for frequently used system functions. Details: see chapter Quick Control.
- **Inputs & Outputs** — Monitoring of the input and output signals of the system.
- **User Variables** — Display and write access to the user variables of the system.
- **DataSet Variables** — Display and modification of the variables of the selected dataset.

- **Program Manager** — Management of the programs on the system, for example loading, saving or deleting program files.

Details are described in the corresponding chapters.

Application Control:

This section provides the controls of the applications that are installed on the system. One entry is displayed per installed application, for example:

- LinMot Grippers
- Conveyor Tracking
- Capping
- Data Recording
- Cognex Interface

Only applications that are installed and licensed are listed. Details for each application are described in the chapter Applications.

If the content of a subpage is wider than the visible area, a horizontal scrollbar at the bottom of the page allows you to display the remaining content.

3.4.1 Axis Control

The Axis Control section allows you to manage and monitor the individual axes of the system in detail.
Important: This page is only operable in Manual Mode. Automatic operation is not possible from this interface.
If your user level does not allow manual operation of the system, the control elements are locked: a lock icon is displayed over the controls and next to the Power toggle, and the controls cannot be operated.
Each axis has its own dedicated control panel, which provides real-time information and the interactive controls for precise operation. Every axis panel contains the following elements:

Axis Header:

- **Axis Name:** Displayed at the top of the panel, for example "x (Robotic Group x)".
- **Power Status Icon:** In the top-right corner of the panel, an icon indicates whether the axis is currently powered on or off.
- **Axis Graphic:** A visual representation of the axis for easy identification (linear axes are shown as a shaft with a slider, rotary axes as a disc). The graphic reflects the actual position of the axis in real time.

Power Control:

The Power toggle enables or disables the axis. Only enabled axes can be moved or set to force.

Current Position:

Displays the real-time position of the axis.

Status Pills:

Below the position, a set of colored status pills shows the current status of the axis. Each pill shows a state text and, when the state is active, the corresponding color:

Pill	Meaning
Com. On / Com. Off	Communication with the axis is active / is not active.
Enabled / Disabled	The axis is enabled for movement / is disabled (see Power toggle).
Homed / Homing	The axis is homed / is currently being homed.
Standstill / Moving	The axis is currently not moving / is currently moving.
Gear In	The drive of the axis is in the gear-in state (ready for operation).
Force Active	Force control is currently active for the axis.
Axis Error	The axis is in an error state. The error is listed on the Events page.

Control Mode:

The control mode switch allows you to choose how the axis is controlled:
- **Position Mode:** The axis is controlled by position.
- **Force Mode:** The axis is controlled by applying a defined force.

Manual Control:

When Position Mode is active, the panel displays:
- **Jog Buttons (- / +):** Move the axis incrementally in small steps, at the speed defined by the Jogging Speed control in the Title Bar.
- **Target Position:** Enter a desired position and press the Execute button (arrow icon) to move the axis to that position.
- **Demand Position:** Displays the commanded position that is currently being applied.

When Force Mode is active, the panel displays instead:
- **Current Force:** Shows the actual force applied by the axis.
- **Target Force:** Enter a desired force value and press the Execute button to apply it.
- **Demand Force:** Displays the commanded force that is currently being applied.

Additional Information:

- **Actual Current:** Shows the current drawn by the motor.
- **Motor Temperature:** Displays the motor temperature in °C.

3.4.2 Quick Control

The Quick Control page provides a configurable control dashboard for the frequently used functions of the system. Like the Operation page, it consists of one or more widget pages: each page is filled with widgets that can be placed freely.
The procedure for creating widget pages and for adding, removing and configuring widgets is identical to the procedure described in the chapter Operation (Edit Mode is entered via the Edit button in the top-right corner of the workspace).

Available Widgets:

In contrast to the Operation page, the Quick Control page is intended for **system control**: in addition to the general widgets, all application-specific widgets are available here. The general widgets are described in the chapter Operation. The application-specific widgets are described in the chapter Applications.

3.5 Program

The Program page allows you to create and edit LinMot-Pilot programs. A program consists of a sequence of commands that are executed one after the other at runtime. This section explains the layout and the functionality of the page.

Page Layout Overview:

The page is divided into two panels:
- **Left panel:** displays the currently opened programs. On the top, one tab is displayed per opened program; below the tabs, the tool bar with the file and editing functions is shown; and the program table displays the commands of the active program.
- **Right panel:** the command editor. On the left-hand side, a toolbox with all available commands is displayed, grouped by category. On the right-hand side, the parameters of the selected command are shown.

Program Tabs:

Each opened program is represented by a tab at the top of the left panel. To switch to a different program, click on the corresponding tab. A tab marked with an asterisk (*) contains unsaved changes. To create a new program, click the plus (+) symbol: a new program is created and displayed as an additional tab.

Program Table:

The program table displays the commands of the active program, one command per row. Each row shows the command icon, the command name and a short description of the parameter values of the command. The indicator at the bottom of the table shows the number of the current row and the maximum number of rows that are supported by a program.

Adding a Command:

To insert a new command into the program:
1. Select the row in the program table (left panel) below which you want to insert the command.
2. In the Commands toolbox (right panel), open the category of the command and select the command you want to add. The parameters of the command are displayed in the command editor.
3. Set the parameters of the command as needed.
4. Click the **Insert** button at the bottom of the command editor to add the command.

Editing an Existing Command:

To modify an existing command:
1. Select the row containing the command you want to edit. The parameters of the command appear in the command editor.
2. If you want to change the command type, select a different command in the Commands toolbox.
3. Modify the parameters as needed.
4. Click the **Insert** button to confirm the changes.

Command Editor:

In the right panel, the parameters of a command are displayed in the command editor. Depending on the command, the parameters are grouped into sections:
- If a command contains several sections, the sections are displayed as numbered tabs at the top of the parameter area (for example **1 Definition · 2 Motion Parameters · 3 Program Behaviour**). Click a tab to switch to the corresponding section. The arrows at the bottom-right of the editor allow you to switch between the sections as well.
- If a command contains only few parameters, they are displayed directly, without sections.

The **?** button at the top-right of the command editor opens the help information for the selected command.

Program Variables:

Next to the command editor, a tab labeled **{x} Program Variables** is displayed. Selecting this tab shows the program variables of the active program instead of the command parameters.

Data Reference Inputs:

Several parameters are entered via a data reference input, which consists of a mode switch and an input field:
- **Numeric mode (π):** You enter a constant value directly into the input field. The button at the right end of the field reads the current value of the corresponding actual value (for example the actual position of an axis) and inserts it into the field.
- **Variable mode ({x}):** Clicking the input field opens the data reference selection, a cascading dialog in which you select the variable you want to use. The name of the selected variable is then displayed in the input field.

Program Editing functions:

The tool bar in the left panel provides the following file and editing functions:

Button	Name	Description
	Open File	opens the file explorer to load a program file
	Save File	save the current program file

	Save All Files	save all currently opened program files
	Save File as	save the current program file under a new name
	Close File	close the current program file
	Close all Files but Selected	close all currently opened files but the selected tab
	Cut	cut the currently selected row(s)
	Copy	copy the currently selected row(s)
	Paste	insert row(s) from the clipboard below the currently selected row
	Undo	undo the last change
	Redo	redo the last change
	Delete Row	delete the currently selected row(s)
	Add Row	add an empty row above the currently selected row
	Comment Out	comment out or uncomment the currently selected row(s)
	Update Descriptions	update the parameter descriptions of the current program file

Tips:

- Always ensure the correct row is selected before inserting or editing commands.

3.6 Setup

The Setup page allows you to configure the system. The settings are divided into two areas, which are listed in the sidebar on the left-hand side of the page. Click an entry to open the corresponding subpage.

Pilot Settings:

Settings for the LinMot-Pilot and its functions. This area contains the following subpages:

- **Control Settings > Axes Motion:** The default speed and acceleration definitions for the linear and rotary axes.
- **Control Settings > Robotic Motion:** The default speed and acceleration definitions for robotic motions.

- **Applications > ...:** The application-specific settings. One subpage is displayed per installed application, for example **LinMot Grippers**, **Conveyor Tracking**, **Capping**, **Data Recording** or **Cognex Interface**. Only applications that are installed and licensed are listed. Details are described in the chapter **Applications**.

System Config:

The general system settings. This area contains the following subpages:
- **General:** Display (time zone), startup behavior, resetting sequence and program control settings. Details: see chapter System Config.
- **User Management:** Management of the users and their user levels. Details: see chapter User Management.
- **File:** Management of the program files and system backups.
- **Kinematics:** The kinematic configuration of the system. Details: see the chapter on kinematics.
- **Network:** The network settings of the system, for example the IP address.
- **PLC Interface:** The connection to external PLCs (for example Profinet or EtherNet/IP). Details: see chapter Fieldbus Connection.

On each settings page, the corresponding parameters are displayed as input fields. Where changes can be saved on a page, two buttons are shown at the bottom-right of the page:
- **Save Changes:** Applies the modified settings.
- **Reset Changes:** Discards the modifications and restores the saved values.

3.6.1 System Config

General Settings:
The General page is divided into sections, each providing specific configuration options:

- **System**
  Allows you to set a custom *System Name*, which is displayed on the top-left corner.

- **Startup Behaviour**
  Defines what happens when the system starts. You can configure whether the axes should automatically be powered on and homed during startup, ensuring the machine is ready for operation immediately after initialization.

- **Resolving Behaviour**
  Defines the behaviour of axes when resolving errors. The behaviour is not shown, when clicking resolve without being in error state. The following options are available:
  - **Power all axes on resolve:**
    All axes are powered on when resolving.
  - **Re-Power after resolving STO:**
    Axes are powered if they were previously disabled by an STO, when resolving the STO.
  - **Never power axes on resolve:**
    No axis is powered on when resolving.

- **Resetting Sequence**
  Lets you select the program that will be executed when the user clicks the Reset button. Additionally, you can specify whether this reset program should automatically run once during startup, providing flexibility for initializing the system according to your operational requirements.

- **Program Control**
  Allows you to choose the default program for the system. You can also define whether this default program should automatically load at startup and whether it should start executing immediately after loading.

File:
- Program:
  This section provides a file explorer for managing program files. The available functions include:
  - Create Folders: Organize your programs by creating new directories.
  - Delete Files: Remove unwanted program files from the system.

- Upload Files: Upload a compressed archive in .tar.gz format containing the desired programs.
- Download Files: Download selected files as a .tar.gz archive for backup or transfer.
The file list displays the name and last modified date of each program file, allowing easy identification and management.

- System:
This section offers tools for system-level configuration and recovery:
- Backup: Creates a backup of the entire system, including all settings, variables, and programs.
- Restore & Reboot: Upload a previously created backup and automatically reboot the system to apply the restored configuration.
- Config Reset: Resets the LinMot-Pilot configuration to default settings. Unsaved data will be lost.
- Factory Reset: Restores the Technology Module to factory settings, which completely removes the LinMot-Pilot software.

Kinematics:
See Kinematic Wizard for a detailed description.

User Management:
See User Management for a detailed description.

PLC Interface:
See PLC Interface Setup for a detailed description.

3.7 Info

The Info page provides detailed information about the system configuration and the software environment. The available submenus are listed in the sidebar on the left-hand side of the page. Click an entry to open the corresponding subpage.

Hardware:

The Hardware submenu shows the currently configured axes, including their article numbers. This provides an overview of the physical setup of the system.
Important: Changes to the hardware configuration are not performed on this page. To modify the hardware setup, use the Kinematic Wizard.

Software:

The Software submenu lists the software installed in the system, divided into two sections:
- **Operating System:** The core software (LinMot-Pilot) with its current version number, the relevant order numbers and license keys.
- **Applications:** All installed applications (for example LinMot Grippers, Conveyor Tracking or Capping), each with its order number and license key.

For each entry, a licensing status is displayed on the right-hand side:
- activated: The software or application is fully activated and licensed.
- Expires in ...: A trial license is active and will expire within the displayed period.
- Expired: The trial period has ended.
- Not activated: Instead of a status text, a button labeled "start trial" is displayed.

A question-mark icon next to each entry displays short information about the corresponding software or application.
[Image]

Important:
- Clicking the "start trial" button starts the trial period for that application.
- If a trial license has expired, it can only be reactivated after the system has been restarted.

For the activation of licenses, contact your sales partner.

3.8 Navigation Menu Bar

The Navigation Menu Bar is located at the bottom of the screen on all pages. It provides the switching between the six main pages as well as system-wide information such as the date and time, the active user and the display language. From left to right, it is divided into the following sections:

- **Date and Time**
  On the far left, the current date and time are displayed. Next to the date, the company logo is shown.

- **User**
  In the center of the bar, a pill displays the name and the user level of the active user. Clicking the pill opens the user menu, which provides quick actions: User Settings, Lock, Logout and Switch User. A detailed description of these functions can be found in the chapter User Management.

- **Language Selection**
  Next to the user pill, the language selection is displayed. The dropdown enables you to change the display language of the user interface. The selection is applied immediately.

- **Navigation Between Main Pages**
  On the right-hand side, one item is displayed for each of the six main pages: **Operation**, **Events**, **Control**, **Program, Setup and Info**. Each item consists of an icon and a label. Clicking an item switches to the corresponding page. The active page is highlighted, so you always see which page is currently displayed.

4 Program Commands

4.1 Pilot Commands

4.1.1 Go To

Redirects the flow to a specified label. It helps guide the sequence by jumping to defined reference points.



Inputs:

Input	Description	Allowed Values
GoTo Label	Select a label to jump to, when command is executed	any of the existing Labels

Functionality:

Continues the execution of the program from the selected label on the next cycle. If the label does not exist, a program error with ID 20 is returned (see Program Events).

4.1.2 Label

Marks a specific point in the sequence. It serves as a reference target for jumps used in 'Go To' or 'If ... Go To ...' commands or acts as option to jump out of commands in error situations like timeouts.



Inputs:

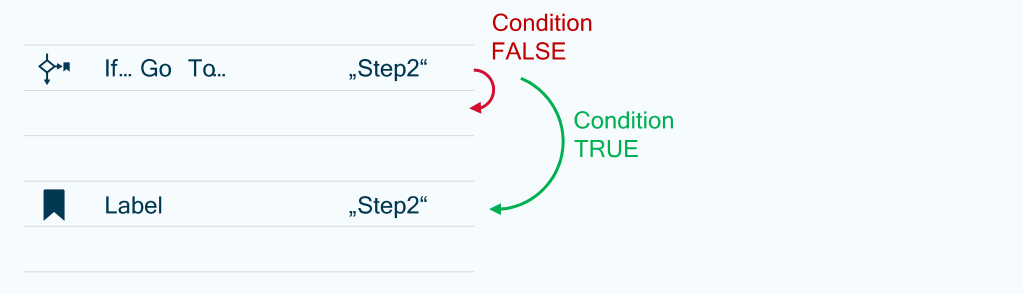
Input	Description	Allowed Values
Label Name	Give a clear name. Used in all Go To dropdowns	text with up to 32 characters

Functionality:

Sets a label as reference for other commands that can use a *Go To* functionality. The *Label Name* is used as a case-sensitive reference. Renaming a label will required according changes in all other commands, referencing this label. Multiple labels with identical names are not allowed and will be indicated by the Program Pre-Compiler. To select a label, a drop-down list is populated with all available labels in the current program, from which a valid label can be selected.

4.1.3 If ... Go To ...

Evaluates a condition and redirects the flow to a defined label if the condition is met. Enables branching and conditional sequence logic.



Inputs:

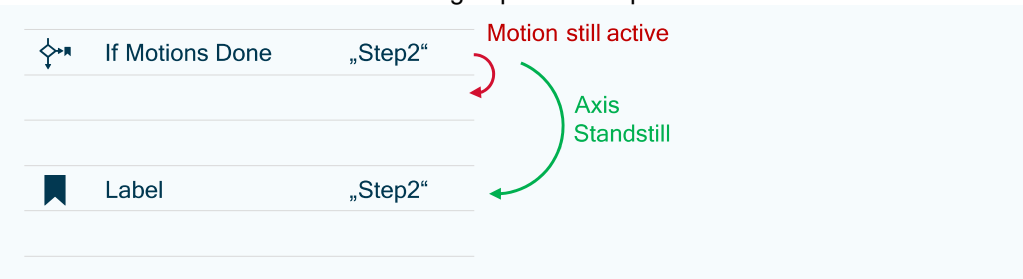
Input	Description	Allowed Values
Logic expression	explained in Common Inputs	
Go To Label	Select a label to jump to, when condition is true	any of the existing Labels

Functionality:

Evaluates the *Expression* at runtime. If the result of the expression is TRUE (numerical value not 0.0) a Go To action to the specified label is performed. If the result is FALSE (numerical value equals 0.0), the program will execute the next row in the upcoming cycle.

4.1.4 If Motions Done

Evaluates if motion of one or several axes or axes groups are completed.



Inputs:

Input	Description	Allowed Values
Check all Motions	When enabled, the condition requires all axes to be stopped	*On* / *Off*
Motion to check	Select motion to be checked for motion done	any of the connected axes or processes
Go To Label	Select a label to jump to, when condition is true	any of the existing Labels

Functionality:

Checks at runtime if *all* configured motions are done. For a motion to be considered done, the setpoint generation has to be finished. It is possible that an axis is still in motion at this point, as it might still be following the final setpoint value for a few cycles. If all configured motions are done, a Go To action to the specified label is performed. If one or more of the configured motions are not done (setpoint generation is still ongoing), the program will execute the next row in the upcoming cycle.

4.1.5 Stop Motions

Stops all or single currently running motion commands. Ensures that movement is halted before transitioning to the next sequence step.

Inputs:

Input	Description	Allowed Values
Stop all Motions	When enabled, all axes interrupt their motion immediately	*On* / *Off*
Motion to stop	Select axis to interrupt motion immediately	any of the connected axes or processes

Functionality:

The configured motions are stopped based on the individual stopping settings of the axis or process.

4.1.6 User Message

Creates a custom message in the Event List. Depending on the selected Event Type, it is handled as information, warning or error and displayed accordingly in the status bar.

Inputs:

Input	Description	Allowed Values

Displayed User Message	Define the message shown in the status bar	text with up to 255 characters
Event Type	Select the severity level of this message	*Information, Warning, Pause on Error* or *Stop on Error*

Functionality:

Will cause an event with Event Source *User*. Events of type *Information* or *Warning* are logged and the program execution continues regularly. For the error types, the program is halted and the error is displayed. When resolving the error, the program will either move to paused state or abort. See Event Handling for more details.

4.1.7 Comment

Adds descriptive text to the sequence without affecting execution. Helps document logic and improve readability.

Inputs:

Input	Description	Allowed Values
Comment	Add a note or title as a separate program line	text with up to 128 characters

Functionality:

As opposed to the comments that can be added to all other commands, the text of this command will be displayed over the whole row. The row will be ignored (i.e. skipped) during program execution.

4.1.8 Pause on Request

Pauses sequence execution when the *Request Pause* button is active or redirects the flow to the specified label. This allows controlled interruption for inspection or manual actions instead of immediate hold.



Inputs:

Input	Description	Allowed Values
Enable Pausing Sequence	Controls whether the program continues with a 'Go To' or pauses immediately	*On* / *Off*
Go To Label	Select a label to jump to, when pause is requested	any of the existing Labels

Functionality:

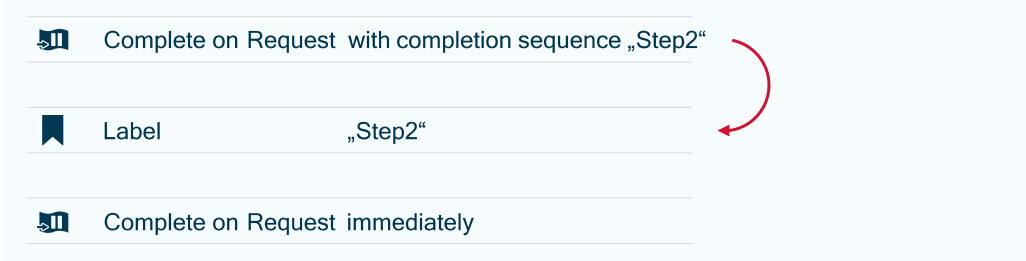
If a pause request is active (i.e., the *Request Pause* button is pressed) when this row is executed, the program will move to the paused state. Note that this does not have the same effect as pressing the *Pause* button, but it behaves the same as pressing the *Step* button, as with *Pause on Request*, no active halting of motions is performed. Axes might therefore still be in motion when the *Paused* state is reached.
The option *Enable Pausing Sequence* allows to perform a Go To instead of pausing immediately if a pause request is active. A second *Pause on Request* with *Enable Pausing Sequence* turned *Off* marks the end of the

pausing sequence and is strictly required, to move the program to the *Paused* state. After pausing the program due to the *Pause on Request*, the *Request Pause* button will be automatically released.

Note: A *Pause on Request* with the option *Enable Pausing Sequence* has to be followed with a second *Pause on Request* with the option turned *Off* at the end of the pausing sequence.

4.1.9 Complete on Request

Stops the sequence execution when the *Request Completion* button is active or redirects the flow to the specified label. This allows controlled ending of the sequence instead of immediate abort.



Inputs:

Input	Description	Allowed Values
Enable Completing Sequence	Controls whether the program continues with a 'Go To' or stops immediately	*On* / *Off*
Go To Label	Select a label to jump to, when completion is requested	any of the existing Labels

Functionality:

If a completion request is active (i.e., the *Req. Completion* button is pressed) when this row is executed, the program will move to the completing state and once all motions are finished, it will move to the *Complete* state. Note that this does not have the same effect as pressing the *Abort* button, but it behaves the same as the program reaching the last set row, as with *Complete on Request*, no active halting of motions is performed. Axes might therefore still be in motion when the *Completing* state is entered.
The option *Enable Completing Sequence* allows to perform a Go To instead of completing immediately if a completion request is active. A second *Complete on Request* with *Enable Completing Sequence* turned *Off* marks the end of the completing sequence and is strictly required, to move the program to the *Completing* state. After completing the program due to the *Complete on Request*, the *Req. Completion* button will be automatically released.

Note: A *Complete on Request* with the option *Enable Completing Sequence* has to be followed with a second *Complete on Request* with the option turned *Off* at the end of the completing sequence.

4.2 Timing & Delay

4.2.1 Wait Time

Waits for a defined duration before continuing the sequence. Enables time-based delays in the workflow.

Inputs:

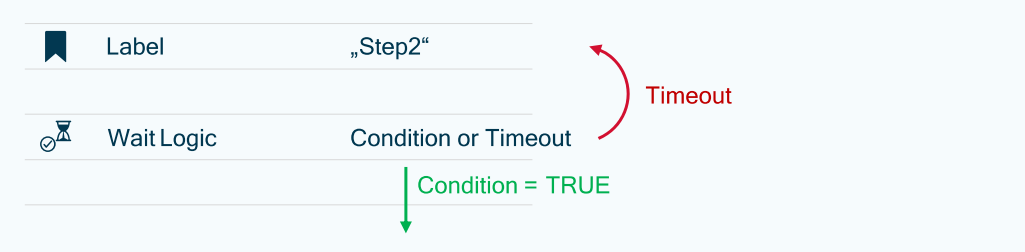
Input	Description	Allowed Values
Wait Time	Define the time to wait before continuing	• positive values only • any Data Reference

Functionality:

The execution of the following row is delayed by the specified *Wait Time*. This command will not influence any ongoing motions or processes but only the execution of the following program commands.

4.2.2 Wait Logic

Waits until a logic condition becomes true or a timeout occurs. If the condition is true, the sequence continues normally. If the timeout expires, the flow is redirected to a user-defined label.



Inputs:

Input	Description	Allowed Values
Logic expression	explained in Common Inputs	
Timeout	explained in Common Inputs	

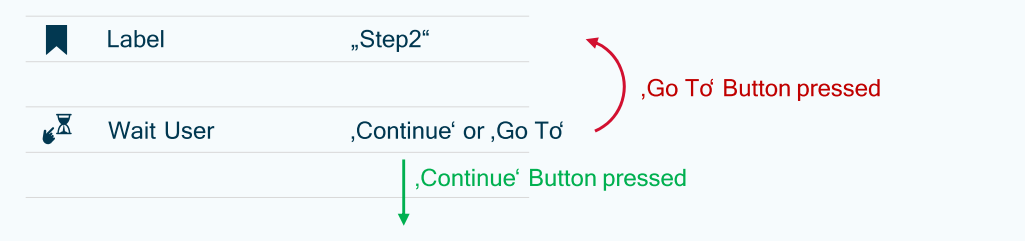
Functionality:

The *Expression* is computed every cycle. As soon as the returned value is logically true (value is not 0.0), the next row will be executed in the upcoming cycle. If a timeout is set and the *Expression* did not return a TRUE value, a Go To to the specified label is performed.

Note: It is possible that the *Expression* will never return TRUE and the program will not be able to continue. Use a *Timeout* to prevent this from happening.

4.2.3 Wait User

Waits for a user interaction before continuing the sequence. A configurable 'Continue' button allows the sequence to proceed normally, while a 'Go To' button van be defined to redirect the flow to a specified label.



Inputs:

Input	Description	Allowed Values
Displayed User Message	Define the message shown in the popup	text with up to 255 characters
Text on 'Continue' Button	Displayed on the button that continues the program	text with up to 32 characters
Enable 'Continue' Button Focus	When enabled, the 'Continue' button appears in blue	*On* / *Off*

Button Continue	Displayed on the button that jumps to a specified label	text with up to 32 characters
Enable 'Go To' Button Focus	When enabled, the 'Go To' button appears in blue	*On* / *Off*
Go To Label	Select a label to jump to when 'Go To' Button is clicked	any of the existing Labels

Functionality:

A pop-up dialog will open in the foreground and can be seen independently of the currently shown menu page. No further rows will be started until the user clicks one of the buttons to either perform a Go To to the specified label or continue the next row. Fig. 18 shows an example for the pop-up dialog with the *Button Go To* on the left ("Yes") and the *Button Continue* on the right ("No").

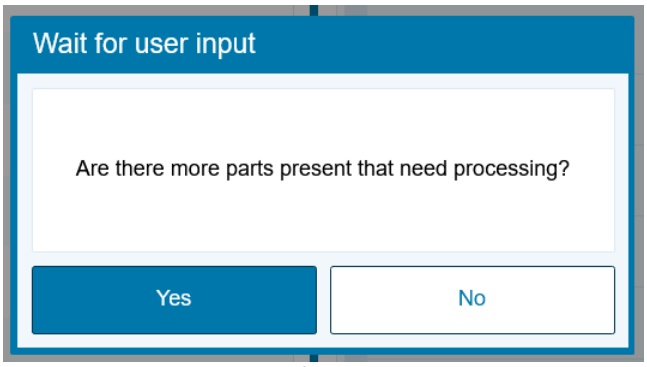
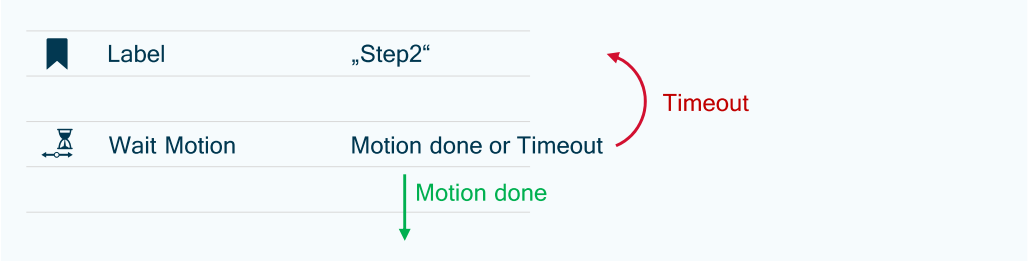


Fig. 18: An example for the Wait User dialog.

4.2.4 Wait Motions

Waits until all or selected axis motions have completed, depending on the chosen configuration. If the motions finish normally, the sequence continues. If the defined timeout expires before completion, the flow is redirected to a user-defined label.



Inputs:

Input	Description	Allowed Values
Wait for all Motions	When enabled, the condition waits until all axes are stopped	*On* / *Off*
Motion to wait for	Select motion to wait for motion done	any of the connected axes or processes
Timeout	explained in Common Inputs	

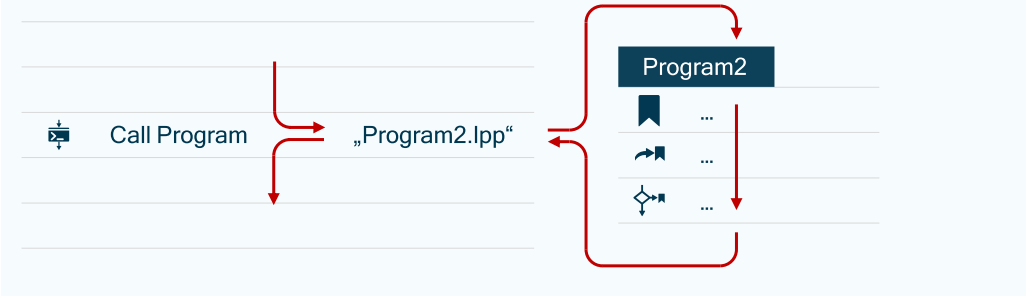
Functionality:

The configured motions are checked every cycle. If *all* configured motions are done, the next row will be executed in the upcoming cycle. If a timeout is set and the motions did not finish in time, a Go To to the specified label is performed. Check If Motions Done for an explanation of when a motion is considered finished.

4.3 Program Control

4.3.1 Call Program

Calls another program and returns to the current sequence once the called program has finished. Useful for modularizing logic or reusing program blocks.



Inputs:

Input	Description	Allowed Values
Program	Select the program to be called	any program saved on the Technology Module

Functionality:

Clicking on the input field for *Program* opens a file-explorer dialog in which the sub-program can be selected. Clicking the -button will switch to or open the configured sub-program in a new tab.
During execution, the sub-program will be switched to or opened if it was not open yet and execution will start from the first row of the sub-program in the same program group from where this command was called. If the execution of the sub-program is finished (last set row was executed), the program execution will return to this *Call Program* command and continue with the next row in the upcoming cycle.

4.3.2 Run Program

Starts another program and executes it in parallel to the calling sequence. The calling program continues without waiting for the other program to finish. Several options allow configuring how the started program behaves, including ho it reacts to pauses in the system and whether it influences or inherits the system status. This enables handling events or parallel motion sequences in separate task.



Inputs:

Input	Description	Allowed Values
Program	Select the program to run as parallel program	any program saved on the Technology Module
Ignore Pause	Program ignores pausing and keeps on running	*On* / *Off*

Ignore Abort	Program ignores aborting and keeps on running	*On* / *Off*
Exclude from System Status	Status of the program will not influence the system status	*On* / *Off*
Restart if Running	If the program is running already, restart if from first row	*On* / *Off*

Functionality:

Clicking on the input field for *Program* opens a file-explorer dialog in which the program can be selected. Clicking the -button will switch to or open the configured program in a new tab.
During execution, the configured program will be run as a parallel process from row 1. If the configured program is running already as a main program of a parallel process, no new parallel process will be started but the properties of the existing parallel process will be updated as configured by the inputs in the command. The program will simply be kept running, or, if *Restart if Running* is selected, the program will be restarted from row 1.

See Program Manager on how to manually control parallel programs.

4.3.3 Stop Program

Stops the execution of a running program immediately. Can be used to abort background programs or terminate parallel sequences.

Inputs:

Input	Description	Allowed Values
Program	Select the program to be stopped immediately	any program saved on the Technology Module

Functionality:

Clicking on the input field for *Program* opens a file-explorer dialog in which the program can be selected. Clicking the -button will switch to or open the configured program in a new tab.
If there is an active program group, where the configured program is the main program, it will be aborted immediately when calling this command. If the configured program is not active, a warning will appear and the program execution continues as usual.

See Program Manager on how to manually control parallel programs.

4.4 Inputs & Outputs

4.4.1 Set Output

Sets a digital or analog output to a defined value. The output an be switched on, off or assigned a specific value. Optional pulse mode allows the output to automatically reset after a defined duration without pausing the sequence.

Inputs:

Input	Description	Allowed Values
Target Output	Select output to be set	Data Reference of a configured output
Value	Define state to set to target output	*True / False*
Pulse Output	When enabled, output turns off after the set duration without pausing execution	*On / Off*
Duration	Time after output is switched off	positive values only

Functionality:

Sets the value of the Data Reference defined as *Target Output* to the configured value (True = 1, False = 0). If *Pulse Output* is enabled, the *Target Output* will be set to *True* initially and switched to *False* after the time configured in *Duration*. Program execution will continue after the *Target Output* was set to *True* initially.

4.4.2 Set Output by logic

Sets an output based on a logical condition. The output value is determined by evaluating a user-defined expression, enabling dynamic output control depending on sensor values, variables or other conditions.

Inputs:

Input	Description	Allowed Values
Target Output	Select output to be set	Data Reference of a configured output
Logic expression	explained in Common Inputs	

Functionality:

Evaluates the *Expression* at runtime and writes the result to the Data Reference defined as *Target Output*.

4.5 Variables

4.5.1 Set Variable

Assigns a variable to a specific value. Useful for updating counters, flags, parameters, or any user-defined data during sequence execution

Inputs:

Input	Description	Allowed Values
Target Variable	Select variable to be set	any Data Reference that allows writing its value
Value	Enter value to set to target variable	• any numerical value • any Data Reference

Functionality:

Writes the configured value or the value of the configured Data Reference at runtime to the Data Reference defined as *Target Variable*.

4.5.2 Set Variable by logic

Assigns a variable using a logic function built from values, other variables, sensor readings, and logical operations. The resulting expression is evaluated at runtime, allowing the variable to update dynamically based on current conditions.

Inputs:

Input	Description	Allowed Values
Target Variable	Select variable to be set	any Data Reference that allows writing its value
Logic expression	explained in Common Inputs	

Functionality:

Evaluates the *Expression* at runtime and writes the result to the Data Reference defined as *Target Variable*.

4.5.3 Select Dataset

Selects a dataset to be used by subsequent commands. This allows switching between predefined data tables or parameter sets during program execution.

Inputs:

Input	Description	Allowed Values
Select Dataset to Load	Einter fix or dynamic by variable	*Selection* / *Variable*

Functionality:

Switches the selected dataset of the Variable Manager. If the dataset is selected by a variable, the value of the Data Reference configured in *Selection Variable* is evaluated at runtime, rounded to the nearest integer and the selected dataset is switched to the dataset representing this value. All negative values and values rounded to 0 of the Data Reference will set *None* as selected dataset and values above 255 will produce an error.

4.6 Axes Motion

4.6.1 Move Linear Axes

Moves one or more linear axes to the specified target positions. Supports absolute and relative positioning with configurable speed and motion parameters.

Inputs:

Input	Description	Allowed Values
Axis to move	Select axis from list	any of the connected, linear axes that is not configured yet
Positioning Mode	Choose position or distance-based movement	• *Absolute:* Move axis to an absolute position • *Relative:* Move axis by a certain distance
Position / Distance	Set the target position	• numerical value within the axis limits configured in Kinematic Configuration • any Data Reference
Offset	Set the offset distance to the target	• any numerical value • any Data Reference
Positioning speed	explained in Common Inputs (*Kinematics*)	
Continue to next line when motion...	explained in Common Inputs	

Functionality:

Add up to eight axes with individual *Mode, Position / Distance* and *Offset*. The movements of the axes will all start at the same time but usually finish at different times, due to possibly different strokes of the motions. The command is considered finished once all configured axes have finished their motion.

4.6.2 Move Rotary Axes

Moves one or more rotary axes to defined angular positions. Allows absolute or relative rotation with adjustable speed and motion settings

Command is functionally identical to Move Linear Axes, with the difference that only rotary axes can be configured as opposed to linear axes and on additional input is available.

Inputs:

Input	Description	Allowed Values
Moving Direction	Select rotation direction to the target position	*shortest*, *negative*, *positive*

4.6.3 Set Force/Torque

Sets the applied force or torque to a defined value. Useful for compliant movements, force-controlled tasks, or adjusting how the system responds during contact or pressure-sensitive operations.

Inputs:

Input	Description	Allowed Values
Axis to operate	Select axis to set force / torque	any of the connected axes
Target Force / Torque	Set the force / torque applied by the axis	• any numerical value in N / Nm • any Data Reference
Enable Speed Limit	When enabled, speed is limited to max velocity	*On* / *Off*
Max Velocity	Max speed during force / torque application	• any numerical value in m/s / °/s • any Data Reference

Functionality:

Will switch the selected axis into force control mode and set the configured *Force / Torque*. If *Enable Speed Limit* is *On*, the speed limit of the selected axis will be set to the configured value. Therefore, the UPID 1511h (*Speed Limit*) will be set on the drive, limiting the velocity at which the axis can move when controlled.

4.7 Robotic Motion

4.7.1 Set Coordinate System

Defines the coordinate system to be used by subsequent robotic movements when *Last Set Coordinate System* is selected in those commands.

Inputs:

Input	Description	Allowed Values
Coordinate System	Defines the coordinate system used by subsequent robotic movements when *Last Set Coordinate System* is selected there	• *Machine Coordinate System:* Coordinate System defined by the axes • *Conveyor Coordinate System:* Coordinate System defined by the Conveyor • *Custom Coordinate Systems:* Coordinate Systems defined in the setup

Functionality:

Sets the coordinate system, which is then used by Move Robotic, Move Circular and Sync to Active Target whenever *Last Set Coordinate System* is selected. A dynamic target coordinate system cannot be set here; it is established by Sync to Active Target.

4.7.2 Set Tool

Defines the tool to be used by subsequent robotic movements when *Last Set Tool* is selected in those commands.

Inputs:

Input	Description	Allowed Values
Tool	Select the tool for robotic movements	• *Custom Tools:* Tools defined in the setup

Functionality:

Sets the tool, which is then used by Move Robotic, Move Circular and Sync to Active Target whenever *Last Set Tool* is selected.

4.7.3 Move Robotic

Moves the configured kinematic group in cartesian space using the currently selected coordinate system. Supports absolute and relative positioning with configurable speed and motion parameters.

Inputs:

Input	Description	Allowed Values
Positioning Mode	Choose position or distance-based movement	• *to Position:* Move kinematic system to an absolute position • *by Distance:* Move kinematic system by a certain distance
Path planning	Choose linear for straight line, otherwise dynamic	*Linear* or *Dynamic*
Coordinate System	Used to interpret the position values	• *Last Set Coordinate System:* Coordinate System set by Set Coordinate System • *Machine Coordinate System:* Coordinate System defined by the axes • *Conveyor Coordinate System:* Coordinate System defined by the Conveyor • *Custom Coordinate Systems:* Coordinate Systems defined in the setup
Tool	Used to interpret the position values	• *No Tool:* no tool offset is applied • *Last Set Tool:* tool set with Set Tool • *Custom Tools:* Tools defined in the setup
Position / Distance X	Relative to the coordinate system (*Position*) / Along the selected coord. system (*Distance*)	• numerical value within the axis limits configured in Kinematic Wizard • any Data Reference
Position / Distance Y	Relative to the coordinate system (*Position*) / Along the selected coord. system (*Distance*)	• numerical value within the axis limits configured in Kinematic Wizard • any Data Reference
Position / Distance Z	Relative to the coordinate system (*Position*) / Along the selected coord. system (*Distance*)	• numerical value within the axis limits configured in Kinematic Wizard • any Data Reference
Position / Distance RZ	Relative to the coordinate system (*Position*) / Along the selected coord. system (*Distance*)	• numerical value within the axis limits configured in Kinematic Wizard • any Data Reference

Rounding Radius	defines the distance from the starting point at which the current motion begins. Using this value creates a smooth transition (blending) between two consecutive movements, eliminating sharp corners and ensuring a continuous path.	• numerical value within the axis limits configured in Kinematic Wizard • any Data Reference
Positioning Speed	explained in Common Inputs (Kinematics)	
Continue to next line when motion...	explained in Common Inputs	

Functionality:

Moves the kinematic system defined in the Kinematic Wizard to a specified position or a specified distance in a coordinate system. The command is considered finished once all configured axes from the kinematic system have finished their motion.
The *Path planning* selects the trajectory: for *Linear* all configured axes are commanded with the same velocity profile, so the tool moves on a straight line; for *Dynamic* the axes move point-to-point (PTP) with their own profiles, so the tool path is not a straight line, and the Positioning Speed is given as a percentage of the dynamic limits configured in *Axes Motion* of the setup. The rotation is commanded via Position/Distance RZ.



Note: Due to the path planning of robotic motions, the time between starting this command and start of the motion (movement of the axes) takes approx. 20 ms.

4.7.4 Move Circular

Moves the configured kinematic group in a circular path through an auxiliary point in cartesian space using the currently selected coordinate system. Supports absolute and relative positioning with configurable speed and motion parameters.

Inputs:

Input	Description	Allowed Values
Positioning Mode	Choose position or distance-based movement	• *to Position:* Move kinematic system to an absolute position • *by Distance:* Move kinematic system by a certain distance
Coordinate System	Used to interpret the position values	• *Last Set Coordinate System:* Coordinate System set by Set Coordinate System • *Machine Coordinate System:* Coordinate System defined by the axes • *Conveyor Coordinate System:* Coordinate System defined by the Conveyor • *Custom Coordinate Systems:* Coordinate Systems defined in the setup
Tool	Used to interpret the position values	• *Custom Tools:* Tools defined in the setup
Position / Distance X	Relative to the coordinate system (*Position*) / Along the selected coord. system (*Distance*)	• numerical value within the axis limits configured in Kinematic Wizard • any Data Reference
Position / Distance Y	Relative to the coordinate system (*Position*) / Along the selected coord. system (*Distance*)	• numerical value within the axis limits configured in Kinematic Wizard • any Data Reference

Position / Distance Z	Relative to the coordinate system (*Position*) / Along the selected coord. system (*Distance*)	• numerical value within the axis limits configured in Kinematic Wizard • any Data Reference
Position / Distance RZ	Relative to the coordinate system (*Position*) / Along the selected coord. system (*Distance*)	• numerical value within the axis limits configured in Kinematic Wizard • any Data Reference
Auxillary Position / Distance X	Relative to the coordinate system (*Position*) / Along the selected coord. system (*Distance*)	• numerical value within the axis limits configured in Kinematic Wizard • any Data Reference
Auxillary Position / Distance Y	Relative to the coordinate system (*Position*) / Along the selected coord. system (*Distance*)	• numerical value within the axis limits configured in Kinematic Wizard • any Data Reference
Auxillary Position / Distance Z	Relative to the coordinate system (*Position*) / Along the selected coord. system (*Distance*)	• numerical value within the axis limits configured in Kinematic Wizard • any Data Reference
Auxillary Position / Distance RZ	Relative to the coordinate system (*Position*) / Along the selected coord. system (*Distance*)	• numerical value within the axis limits configured in Kinematic Wizard • any Data Reference
Rounding Radius	defines the distance from the starting point at which the current motion begins. Using this value creates a smooth transition (blending) between two consecutive movements, eliminating sharp corners and ensuring a continuous path.	• numerical value within the axis limits configured in Kinematic Wizard • any Data Reference
Positioning Speed	explained in Common Inputs (Kinematics)	
Continue to next line when motion...	explained in Common Inputs	

Functionality:

Moves the kinematic system defined in the Kinematic Wizard along a circular arc in a coordinate system. The arc is defined by the end position (Position / Distance X–RZ) and the border point (Auxiliary Position / Distance X–RZ): the tool moves clockwise along the arc, passing the border point and ending at the end position. In *by Distance* mode both the end position and the border point are given as offsets from the start position of the motion. The command is considered finished once all configured axes from the kinematic system have finished their motion.

Note: The arc of a single command must be smaller than 360°; a full circle has to be programmed as two consecutive commands.

Note: Due to the path planning of robotic motions, the time between starting this command and start of the motion (movement of the axes) takes approx. 20 ms.

4.8 LinMot Grippers

4.8.1 Move Gripper

Move gripper jaw(s) to a specified position or by a specified distance.

Inputs:

Input	Description	Allowed Values
Gripper	Select gripper from list	any of the available gripper instances

Positioning Mode	Choose position or distance-based movement	• *to Position:* Move gripper jaw(s) to an absolute position • *by Distance:* Move gripper jaw(s) by a certain distance
Jaw Behaviour	Choose how the jaw positions are interpreted	*Mirrored, Single Jaw 1, Single Jaw 2* or *Parallel*
Position / Distance	Set the target position or movement distance (based on selected *Positioning Mode*)	• numerical value within the axis limits configured in Kinematic Configuration • any Data Reference
Positioning Speed	explained in Common Inputs (*Kinematics*)	
Continue to next line when motion...	explained in Common Inputs	

Functionality:

Move the gripper jaw(s) based on the configuration of *Move Mode* and *Jaw Mode*. The interactions of the two modes are summarized in the table below. For single axis grippers, only *Mirrored* is available for *Jaw Mode*. The jaw mode *Mirrored* always references the sum of both gripper jaws, in double axis grippers.

Jaw Mode \ *Move Mode*	**Absolute**	**Relative**
Mirrored	Move both jaws to half the defined, absolute position.	Move both jaws half the defined, relative distance.
Single Jaw 1	Move jaw 1 to defined, absolute position.	Move jaw 1 the defined, relative distance.
Single Jaw 2	Move jaw 2 to defined, absolute position.	Move jaw 2 the defined, relative distance.
Parallel	Move jaw 1 to defined, absolute position. Jaw 2 follows motion while keeping constant jaw width.	Move jaw 1 the defined, relative distance. Jaw 2 follows motion while keeping constant jaw width.

The *Move Gripper* command will always reset the *Grip Active* flag (see Control) and deactivate the position or force monitoring.

4.8.2 Grip by Position

Grip an object based on a defined gripping position. Optionally, a pre-position with a varying set of speed parameters can be defined and/or the force can be checked during the gripping process.

Inputs:

Input	**Description**	**Allowed Values**
Gripper	Select gripper from list	any of the available gripper instances
Grip Position	Define where the part is gripped	• numerical value within the axis limits configured in Kinematic Configuration • any Data Reference
Positioning Speed	explained in Common Inputs (Kinematics)	
Pre-Position	explained in Common Inputs	
Enable Force Check	When enabled, the applied force at the grip position is verified	*On* / *Off*
Expected Force Feedback	Force present at grip position	• positive, numerical value up to 500% • any Data Reference
Expected Force Direction	In which direction will the force apply to the jaws	*Closing* / *Opening* (see Setup)

Allowed Tolerance	Acceptable difference of the result	• numerical value above 0% and up to 500% • any Data Reference
Go To Label if out of Tolerance	Select a label to jump to when force check fails	any of the existing Labels
Continue to next line when motion...	explained in Common Inputs	

Functionality:

Move the gripper jaw(s) to a defined, absolute position and set the *Grip Active* flag (see Control). If *Enable Force Check* is *On*, the *Grip Active* (see Control) will only be set if the actual force is within the defined target force window. The force will already be check during the gripping motion. If the force is too high (actual force is outside *Expected Force Feedback* + *Allowed Tolerance*) at any point during the gripping motion, the *Force too High* flag (see Control) will be set, the motion stopped and the *Target Force* will be applied. If the force is too low (actual force is outside *Expected Force Feedback* - *Allowed Tolerance*) at the *Target Position*, the *Force too Low* flag (see Control) will be set. In both cases, the *Grip Active* (see Control) will not be set.

4.8.3 Grip by Force

Grip an object with a defined gripping force. Contact will be enabled automatically, independent on the current position. Optionally, a pre-position with a varying set of speed parameters can be defined and/or the position can be checked during the gripping process.

Inputs:

Input	Description	Allowed Values
Gripper	Select gripper from list	any of the available gripper instances
Force Direction	Direction of the applied force	*Closing* / *Opening* (see Setup)
Grip Force	Force to apply on the part	• positive, numerical value up to 500% • any Data Reference
Positioning Speed	explained in Common Inputs (Kinematics)	
Pre-Position	explained in Common Inputs	
Enable Position Check	When enabled, the final position is verified after part is gripped	*On* / *Off*
Expected Grip Position	Target Position when part is gripped	• numerical value within the axis limits configured in Kinematic Configuration • any Data Reference
Allowed Tolerance	Acceptable difference of the result	• numerical value above 0 • any Data Reference
Go To Label if out of Tolerance	Select a label to jump to when position check fails	any of the existing Labels
Continue to next line when motion...	explained in Common Inputs	

Functionality:

Move the gripper jaw(s) with the direction depending on the *Force Direction*. As soon as the force on the gripper reaches the specified *Force*, the motion is stopped, the gripper is changed to force controlled mode with the specified *Force* as target value and the *Grip Active* flag (see Control) is set. For a double jawed gripper, both jaws have to detect contact individually and the second jaw to do so, is set to force controlled, while the first one is set to position control and kept at constant position. If *Enable Position Check* is *On*, the *Grip Active* (see Control) will only be set if the actual position is within the defined target position window. The position will

already be check during the gripping motion. If the position is too high (actual position is outside *Expected Grip Position* + *Allowed Tolerance*) at any point during the gripping motion, the *Position above Target Window* flag (see Control) will be set, the motion stopped and no force will be applied. If the position is too low (actual position is outside *Expected Grip Position* - *Allowed Tolerance*) at any point during the gripping motion, the *Position below Target Window* flag (see Control) will be set. In both cases, *Grip Active* (see Control) will not be set.



Note: The automatic enabling of contact requires the force to exceed the specified gripping force. With higher velocities of the gripping motion, this exceedence will increase as well. To limit these high, short-time forces, move to a pre-position quickly and executed the *Grip by Force* command with slower kinematics.

4.8.4 Ungrip

Release an object by moving to a defined position. A pre-position with a varying set of speed parameters can be defined optionally.

Inputs:

Input	Description	Allowed Values
Gripper	Select gripper from list	any of the available gripper instances
Ungrip Position	Define the parking position	• numerical value within the axis limits configured in Kinematic Configuration • any Data Reference
Positioning Speed	explained in Common Inputs (Kinematics)	
Pre-Position	explained in Common Inputs	
Continue to next line when motion...	explained in Common Inputs	

Functionality:

Move the gripper jaw(s) to a defined, absolute position and reset the *Grip Active* flag (see Control).
The *Ungrip* command can only be executed if *Grip Active* (see Control) is set. The *Ungrip* command will always reset the *Grip Active* flag (see Control) and deactivate the position or force monitoring.

4.8.5 Gripper Monitoring

Enable or disable monitoring of the gripper position and/or force.

Inputs:

Input	Description	Allowed Values
Gripper	Select gripper from list	any of the available gripper instances
Enable Position Monitoring	When enabled, the actual gripper position is monitored within the allowed range	*On* / *Off*
Target Position to monitor	Expected position	• numerical value within the axis limits configured in Kinematic Configuration • any Data Reference
Allowed Tolerance	Acceptable difference to target position	• numerical value above 0 • any Data Reference
Enable Force Monitoring	When enabled, the actual gripper force is monitored within the allowed range	*On* / *Off*

Target Force to monitor	Expected force	• positive, numerical value up to 500% • any Data Reference
Expected Force Direction	In which direction will the force apply to the jaws	*Closing* / *Opening* (see Setup)
Allowed Tolerance	Acceptable difference to target force	• numerical value above 0% and up to 500% • any Data Reference

Functionality:

If enabled, the position and/or force will be checked continuously. If position and/or force lay outside the permissible range (see Grip by Position or Grip by Force) the according indicator flags (*Force too low/high* or *Position above/below Target Window* in Control) are set and *Grip Active* is reset.
The position or force monitoring can only be enabled if *Grip Active* (see Control) is set. While position or force monitoring is active, the *Monitoring Active* flag (see Control) is set.

4.8.6 Set Grip Force

Set a specified force to gripper jaw(s) while a grip is already active.

Inputs:

Input	Description	Allowed Values
Gripper	Select gripper from list	any of the available gripper instances
Grip Force	Force to apply on the jaws	• positive, numerical value up to 500% • any Data Reference
Force Direction	Direction of the applied force	*Closing* / *Opening* (see Setup)
Jaw Behaviour	Choose how the force is applied on the jaws	*Mirrored, Single Jaw 1* or *Single Jaw 2*

Functionality:

The gripping force can only be set or changed if *Grip Active* (see Control) is set. The *Jaw Mode* specifies which jaw(s) are set to force control mode and will experience the specified force. *Single Jaw 1* and *Single Jaw 2* set the only the according jaw, while *Opposing* sets both jaws to force control mode. For single axis grippers, only *Opposing* is available for *Jaw Mode*. The *Parallel* mode is not available for this command. Note that for double jawed grippers, the force controlled jaw can switch when using the *Set Grip Force* command.

4.9 Conveyor Tracking

4.9.1 Control Target Detection

Starts or stops the target detection of the selected conveyor.

Inputs:

Input	Description	Allowed Values
Conveyor Belt	Select conveyor belt from list	• any conveyor belt
Select Operation	Choose between Start and Stop	• Start • Stop

Functionality:

Starts or stops the target detection configured in the conveyor belt settings for the selected conveyor: new targets are added to the buffer only while the detection is running. The current state is shown in the control tab of the conveyor.

4.9.2 Set Active Target

Defines the active target on the selected conveyor belt.

Inputs:

Input	Description	Allowed Values
Conveyor Belt	Select conveyor belt from list	• any conveyor belt
Target	Select a target from list	• oldest • oldest + n • newest • newest -n

Functionality:

Defines the active target on the selected conveyor belt. The active target represents a dynamically tracked position within the conveyor's coordinate system.
This target is then used by the Sync to Active Target command to guide the kinematic system while synchronizing with the conveyor movement. The active target can only be changed while the selected conveyor has no active sync process and is not tracking a conveyor; until then the command waits on this line (the tracking is released by Desync from Conveyor).

4.9.3 Wait for Active Target

Waits until the active target of the selected conveyor reaches the earliest sync position.

Inputs:

Input	Description	Allowed Values
Conveyor Belt	Select conveyor belt from list	• any conveyor belt

Functionality:

Waits until the active target of the selected conveyor reaches the earliest sync position. If the active target has already passed this position, the command completes immediately.



Note: This line has no timeout and no upper bound: the line releases while the active target is still waiting, even if it has already passed the latest sync position. It is only removed once the latest desync position is reached.

4.9.4 Sync to Active Target

Moves the kinematic system to a defined position within the Active Target coordinate system of the selected conveyor. After the target position in this dynamic coordinate system is reached, the kinematic system remains synchronized with the moving coordinate system and continues to follow it until a 'Desync from Conveyor' command is executed and completed.

Inputs:

Input	Description	Allowed Values
Mode	defines how the goal position is defined	• *Absolute:* Move kinematic system to an absolute position • *Relative:* Move kinematic system by a certain distance

Conveyor Belt	Defines which conveyor belt is used to determine the Active Target coordinate system	• *any conveyor belt*
Tool	Used to interpret the position values	• *No Tool:* no tool offset is applied • *Last Set Tool:* tool set with Set Tool • *Custom Tools:* Tools defined in the setup
Sync at current position	sets the goal position to the current position of the robot at the time of execution	• *On / Off*
Position / Distance X	goal position (*Absolute Mode*) or stroke to be moved (*Relative Mode*) in X-Direction	• numerical value within the axis limits configured in Kinematic Wizard • any Data Reference
Position / Distance Y	goal position (*Absolute Mode*) or stroke to be moved (*Relative Mode*) in Y-Direction	• numerical value within the axis limits configured in Kinematic Wizard • any Data Reference
Position / Distance Z	goal position (*Absolute Mode*) or stroke to be moved (*Relative Mode*) in Z-Direction	• numerical value within the axis limits configured in Kinematic Wizard • any Data Reference
Position / Distance RZ	goal position (*Absolute Mode*) or stroke to be moved (*Relative Mode*) in RZ-Direction	• numerical value within the axis limits configured in Kinematic Wizard • any Data Reference
Rounding Radius	defines the distance from the starting point at which the current motion begins. Using this value creates a smooth transition (blending) between two consecutive movements, eliminating sharp corners and ensuring a continuous path.	• numerical value within the axis limits configured in Kinematic Wizard • any Data Reference
Kinematics	explained in Common Inputs	
Continue to next line when motion...	explained in Common Inputs	

Functionality:

Moves the kinematic system defined in the Kinematic Wizard to a specified position or a specified distance in the Active Target coordinate system of the selected conveyor. The command is considered finished once the dynamic coordinate system is set and the goal position is reached; from then on the kinematic system remains synchronized with the moving coordinate system and continues to follow the active target until a Desync from Conveyor command is executed and completed.
The command can be executed only under the following conditions; otherwise a program error is raised and the line stays:

- the kinematic system is not already synchronized to a conveyor belt
- the target buffer of the selected conveyor contains at least one target



Note: Due to the path planning of robotic motions, the time between starting this command and start of the motion (movement of the axes) takes approx. 20 ms.

4.9.5 Move Synchronized

Moves the kinematic system to a defined position or by a defined displacement within the dynamic Active Target coordinate system of the selected conveyor. After the target position in this dynamic coordinate system is reached, the kinematic system remains synchronized with the moving coordinate system and continues to follow it until a 'Desync from Conveyor' command is executed and completed.

Inputs:

Input	Description	Allowed Values
Mode	defines how the goal position is defined	• *Absolute:* Move kinematic system to an absolute position • *Relative:* Move kinematic system by a certain distance
Conveyor Belt	Defines which conveyor belt is used to determine the Active Target coordinate system	• *any conveyor belt*
Tool	Used to interpret the position values	• *Custom Tools:* Tools defined in the setup
Position / Distance X	goal position (*Absolute Mode*) or stroke to be moved (*Relative Mode*) in X-Direction	• numerical value within the axis limits configured in Kinematic Wizard • any Data Reference
Position / Distance Y	goal position (*Absolute Mode*) or stroke to be moved (*Relative Mode*) in Y-Direction	• numerical value within the axis limits configured in Kinematic Wizard • any Data Reference
Position / Distance Z	goal position (*Absolute Mode*) or stroke to be moved (*Relative Mode*) in Z-Direction	• numerical value within the axis limits configured in Kinematic Wizard • any Data Reference
Position / Distance RZ	goal position (*Absolute Mode*) or stroke to be moved (*Relative Mode*) in RZ-Direction	• numerical value within the axis limits configured in Kinematic Wizard • any Data Reference
Rounding Radius	defines the distance from the starting point at which the current motion begins. Using this value creates a smooth transition (blending) between two consecutive movements, eliminating sharp corners and ensuring a continuous path.	• numerical value within the axis limits configured in Kinematic Wizard • any Data Reference
Kinematics	explained in Common Inputs	
Continue to next line when motion...	explained in Common Inputs	

Functionality:

Moves the kinematic system defined in the Kinematic Wizard to a specified position or by a specified distance in the dynamic Active Target coordinate system of the selected conveyor.
Important: It is only possible to move synchronized with a conveyor belt if the robot has been successfully synced by a 'Sync to Active Target' command before.



Note: Due to the path planning of robotic motions, the time between starting this command and start of the motion (movement of the axes) takes approx. 20 ms.

4.9.6 Desync from Conveyor

Stops the tracking of a conveyor belt by moving the configured kinematic group to a specified position in cartesian space. Supports absolute and relative positioning with configurable speed and motion parameters.

Inputs:

Input	Description	Allowed Values

Positioning Mode	Choose position or distance-based movement	• *to Position:* Move kinematic system to an absolute position • *by Distance:* Move kinematic system by a certain distance
Path planning	Choose linear for straight line, otherwise dynamic	*Linear* or *Dynamic*
Coordinate System	Used to interpret the position values	• *Last Set Coordinate System:* Coordinate System set by Set Coordinate System • *Machine Coordinate System:* Coordinate System defined by the axes • *Conveyor Coordinate System:* Coordinate System defined by the Conveyor • *Custom Coordinate Systems:* Coordinate Systems defined in the setup
Tool	Used to interpret the position values	• *Custom Tools:* Tools defined in the setup
Desync at current position	When On, the position fields of this line are not shown and the motion ends at the current position of the robot in the selected coordinate system: the robot stops following the belt at its current position	• *On / Off*
Position / Distance X	Relative to the coordinate system (*Position*) / Along the selected coord. system (*Distance*)	• numerical value within the axis limits configured in Kinematic Wizard • any Data Reference
Position / Distance Y	Relative to the coordinate system (*Position*) / Along the selected coord. system (*Distance*)	• numerical value within the axis limits configured in Kinematic Wizard • any Data Reference
Position / Distance Z	Relative to the coordinate system (*Position*) / Along the selected coord. system (*Distance*)	• numerical value within the axis limits configured in Kinematic Wizard • any Data Reference
Position / Distance RZ	Relative to the coordinate system (*Position*) / Along the selected coord. system (*Distance*)	• numerical value within the axis limits configured in Kinematic Wizard • any Data Reference
Rounding Radius	defines the distance from the starting point at which the current motion begins. Using this value creates a smooth transition (blending) between two consecutive movements, eliminating sharp corners and ensuring a continuous path.	• numerical value within the axis limits configured in Kinematic Wizard • any Data Reference
Positioning Speed	explained in Common Inputs (Kinematics)	
Continue to next line when motion...	explained in Common Inputs	

Functionality:

Desyncs from the conveyor by moving the kinematic system defined in the Kinematic Wizard to a specified position or a specified distance in a coordinate system. The command is considered finished once all configured axes from the kinematic system have finished their motion; the kinematic system is then released from the dynamic conveyor coordinate system (tracking is inactive). The targets remain in the buffer of the conveyor.

Note: Due to the path planning of robotic motions, the time between starting this command and start of the motion (movement of the axes) takes approx. 20 ms.

4.9.7 Clear Target from Buffer

Clears the selected Target from the selected conveyor belt.

Inputs:

Input	Description	Allowed Values
Conveyor Belt	Select conveyor belt from list	• any conveyor belt
Target	specifies which target is removed from the buffer	• active • oldest • oldest + n • newest • newest -n

Functionality:

Clears the selected Target from the selected conveyor belt. The command can only be executed while the selected conveyor has no active sync process and is not tracking a conveyor; until then the command waits on this line.

4.9.8 Clear Buffer

Clears all Targets from the selected conveyor belt.

Inputs:

Input	Description	Allowed Values
Conveyor Belt	Select conveyor belt from list	• any conveyor belt

Functionality:

Clears all Targets from the selected conveyor belt. The command can only be executed while the selected conveyor has no active sync process and is not tracking a conveyor; until then the command waits on this line.

4.10 Capping

4.10.1 Find Seating Point

Moves the linear axis of the capper whilst monitoring the linear force. Once the force reaches a defined threshold, the motion is stopped and a defined force is applied to the linear axis. Used to detect and contact the seating point of a thread.

Inputs:

Input	Description	Allowed Values
Select Capper	Choose the capper to operate	any of the available capper instances
Endposition of Search	Set lowest limit for search	• numerical value • any Data Reference
Search Speed	explained in Common Inputs (*Kinematics*)	
Target Force	Target force at seating point	• numerical value • any Data Reference

Allowed Tolerance	Related to target force	• positive, numerical value unequal zero • any Data Reference
Target Standstill Speed	Speed to accept as standstill	• positive, numerical value • any Data Reference
Evaluation time	Duration both conditions must be fulfilled	• positive, numerical value
Timeout Duration	Define the time until the timeout occurs	• positive, numerical value
Behaviour if out of Tolerance or Timeout	Command stops with error or performs Go To	• *Direct Error* • any of the existing Labels

Functionality:

The linear axis of the selected capper is moved towards *Endposition of Search* with the speed parameters defined in *Search Speed*. During this motion, the axis force is limited to the value defined in *Target Force*, to prevent high force peaks when contacting a part. While moving, the linear axis is being check for the target force and standstill. The condition for target force is met, if the actual force of the linear axis falls within the range of *Target Force +/- Allowed Tolerance*. The condition for standstill is met, if the absolute value of the actual velocity of the linear axis is smaller than *Target Standstill Speed*. If both conditions are met continuously for the *Evaluation time*, the command is considered successful, the motion is stopped, the linear axis is set to force control and the *Target Force* is applied to the linear axis. If the conditions can not be met within the *Timeout Duration*, the command is aborted and the *Command aborted* and *Command timed out* flags (see Control) are set. The *Force-Check Failed* and *Linear Standstill-Check failed* are set, if the according condition is not met at the time of the timeout. If *Endposition of Search* is reached and the motion is finished, the command is aborted and the *Command aborted* and *Position-Check failed* flags (see Control) are set.
If the command is aborted and *Direct Error* is selected for *Behaviour if out of Tolerance or Timeout*, an error is shown an program execution will be paused. If a label is selected, the program execution will jump to the label without displaying an error.

4.10.2 Find Thread Start

Moves the rotary axis of the capper whilst monitoring the linear position. Once a jump in linear position is detected, the rotary motion is stopped. Used to detect the thread start, once contact with the thread was enabled.

Inputs:

Input	Description	Allowed Values
Select Capper	Choose the capper to operate	any of the available capper instances
Search Direction	Select direction opposite to thread direction	*Clockwise* / *Counter-clockwise* (see Setup)
Search Speed	explained in Common Inputs (*Kinematics*)	
Force to Apply	Linear force during rotation	• positive, numerical value • any Data Reference
Force Direction	Direction of the applied force	*Towards cap* / *Away from cap* (see Setup)
Stop on Falling Distance	Position deviation to trigger stop	• positive, numerical value • any Data Reference
Max Rotation Angle	Expected thread start position	• positive, numerical value • any Data Reference
Timeout Duration	Time to wait for command to complete	• positive, numerical value
Behaviour if out of Tolerance or Timeout	Command stops with error or performs Go To	• *Direct Error* • any of the existing Labels

Functionality:

The rotary axis of the selected capper is moved in the direction defined by *Search Direction* while a constant force is applied to the linear axis in the direction defined by *Force Direction*. The position of the linear axis is monitored and the minimum / maximum position is stored. Once a position deviation to the minimum / maximum position is larger than the defined distance in *Stop on Falling Distance* in the direction opposite *Force Direction*, the command is considered successful and the motion is stopped.
If no position step larger than the defined *Stop on Falling Distance* is found within the *Timeout Duration*, the command is aborted and the *Command aborted* and *Command timed out* flags (see Control) are set. *Max Rotation Angle* defines the maximum distance, that the rotary axis will move. If this distance is reached, the command is aborted and the *Command aborted* and *Position-Check failed* flags (see Control) are set.
If the command is aborted and *Direct Error* is selected for *Behaviour if out of Tolerance or Timeout*, an error is shown an program execution will be paused. If a label is selected, the program execution will jump to the label without displaying an error.
Capping Angle (see Control) will represent the total covered angle after the command has finished.



Note: Ensure the linear axis has already enabled contact with the thread when calling this command. Otherwise, chances are that the contacting motion of the linear axis is falsely detected as the thread start.

4.10.3 Cap to Angle

Moves the rotary axis of the capper to a defined target angle. The linear axis can be synchronized by applied force or position. Used to cap to a predefined angle, independent of the required torque.

Inputs:

Input	Description	Allowed Values
Select Capper	Choose the capper to operate	any of the available capper instances
Rotation Direction	Select direction along thread direction	• *Clockwise* / *Counter-clockwise* (see Setup)
Operation Mode of Linear Axis	explained in Common Inputs	
Target Angle	Final angle of the cap	• positive, numerical value unequal zero • any Data Reference
Rotation Speed	explained in Common Inputs (*Kinematics*)	
Allowed Tolerance	Target angle to current angle	• positive, numerical value unequal zero • any Data Reference
Evaluation time	Duration both conditions must be fulfilled	• positive, numerical value
Timeout Duration	Define the time until the timeout occurs	• positive, numerical value
Behaviour if out of Tolerance or Timeout	Command stops with error or performs Go To	• *Direct Error* • any of the existing Labels

Functionality:

The rotary axis rotates along the *Rotation Direction* for a total distance of the *Target Angle* with the defined *Rotation Speed*. The actual position of the rotary axis is monitored and compared to the *Target Angle*. If the actual position is within *Target Angle +/- Allowed Tolerance* for the duration of *Evaluation time*, the command is considered successful, the motion is stopped if not already finished and the linear axis is set to position control.
If the *Target Angle* can not be reached within the *Timeout Duration*, the command is aborted and the *Command aborted* and *Command timed out* flags (see Control) are set.

If the command is aborted and *Direct Error* is selected for *Behaviour if out of Tolerance or Timeout*, an error is shown an program execution will be paused. If a label is selected, the program execution will jump to the label without displaying an error.
Capping Angle (see Control) will represent the total covered angle after the command has finished.
The behaviour of the linear axis during the capping is defined by the settings in *Operation Mode of Linear Axis*.

4.10.4 Cap to Torque

Applies a defined torque to the rotary axis of the capper with pre-positioning and initial torque point detection as options. The linear axis can be synchronized by applied force or position. Used to cap to a predefined torque, independent of the actual capping angle.

Inputs:

Input	Description	Allowed Values
Select Capper	Choose the capper to operate	any of the available capper instances
Rotation Direction	Select direction along thread direction	• *Clockwise* / *Counter-clockwise* (see Setup)
Operation Mode of Linear Axis	explained in Common Inputs	
Enable Fast Pre-Positioning	When enabled, rotate fast before initial torque point	*On* / *Off*
Enable Initial Torque Point Detection	When enabled, search first measurable rise of torque before tightening	*On* / *Off*
Speed Limit	When applying Target Torque	• any positive, numerical value • any Data Reference
Force to Apply	Force applied during tightening	• any positive, numerical value • any Data Reference
Force Direction	Direction of the applied force	*Towards cap* / *Away from cap* (see Setup)
Pre-Position	Switchpoint to slow down	• any positive, numerical value • any Data Reference
Pre-Positioning Rotation Speed	explained in Common Inputs (*Kinematics*)	
Initial Torque Point	Switchpoint before tightening	• any positive, numerical value • any Data Reference
Initial Torque Seach Speed	explained in Common Inputs (*Kinematics*)	
Target Torque	Final torque after tightening	• numerical value • any Data Reference
Allowed Tolerance	Related to target force	• positive, numerical value unequal zero • any Data Reference
Target Standstill Speed	Speed to accept as standstill	• positive, numerical value • any Data Reference
Evaluation time	Duration both conditions must be fulfilled	• positive, numerical value
Timeout Duration	Define the time until the timeout occurs	• positive, numerical value
Behaviour if out of Tolerance or Timeout	Command stops with error or performs Go To	• *Direct Error* • any of the existing Labels

Functionality:

If *Enable Fast Pre-Positioning* is *On*, the rotary axis rotates along the *Rotation Direction* for a total distance of the *Pre-Position* with the defined *Pre-Positioning Rotation Speed*.
If *Enable Initial Torque Point Detection* is *On*, the rotary axis rotates along the *Rotation Direction* with the defined *Initial Torque Seach Speed* until the actual torque exceeds the *Initial Torque Point*. If *Enable Fast Pre-Positioning* is *On*, the rotary axis is not stopped between pre-positioning and initial torque point search, but the velocity will be adjusted to the defined *Speed Limit*. As soon as this torque is reached, the rotary motion is stopped.
The rotary axis is set to force control and the *Target Torque* is applied. The *Speed Limit* is set by setting the UPID 1511h (*Speed Limit*) on the drive, limiting the velocity at which the rotary axis can move when controlled. During this step, a force has to be applied to the linear axis. If *Operation Mode of Linear Axis* is set to *Sync. with Rotary Axis*, the controller mode of the linear axis is switched to force control. During tightening, the rotary axis is being checked for the target torque and standstill. The condition for target torque is met, if the actual torque of the rotary axis falls within the range of *Target Torque +/- Allowed Tolerance*. The condition for standstill is met, if the absolute value of the actual velocity of the rotary axis is smaller than *Target Standstill Speed*. If both conditions are met continuously for the *Evaluation time*, the command is considered successful and both axes are switched to position control. If the conditions can not be met within the *Timeout Duration*, the command is aborted and the *Command aborted* and *Command timed out* flags (see Control) are set. The *Torque-Check Failed* and *Rotary Standstill-Check failed* are set, if the according condition is not met at the time of the timeout.
If the command is aborted and *Direct Error* is selected for *Behaviour if out of Tolerance or Timeout*, an error is shown an program execution will be paused. If a label is selected, the program execution will jump to the label without displaying an error.
Capping Angle (see Control) will represent the total covered angle after the command has finished.
The behaviour of the linear axis during the capping is defined by the settings in *Operation Mode of Linear Axis*.

4.10.5 Tare

Offsets the force / torque on the linear and / or rotary axis of the capper to result in a force of 0 N / torque of 0 Nm. The capper will use the offset values for all future force / torque measurements. Used to correct for additional load on the capper.

Inputs:

Input	Description	Allowed Values
Select Capper	Choose the capper to operate	any of the available capper instances
Execute Force Tare on Linear Axis	The currently measured force is set to 0 N	*On / Off*
Execute Torque Tare on Rotary Axis	The currently measured torque is set to 0 Nm	*On / Off*

Functionality:

If *Execute Force Tare on Linear Axis* is *On*, the currently measured force of the linear axis is stored and subtracted from all future measurements, resulting in an output force of 0 N. This affects the *Linear Capper Force* value (see Control) as well as all comparisons and checks of other capping commands.
If *Execute Torque Tare on Rotary Axis* is *On*, the behaviour is identical to that of the linear axis.

4.10.6 Check Positions

Checks the position of the rotary and linear axis and the total capping angle against a specified expected value with a certain tolerance and evaluation time. If any of the checks is not successful, an error is returned or a Go To is performed.

Inputs:

Input	Description	Allowed Values
Select Capper	Choose the capper to operate	any of the available capper instances

Enable Rotary Angle Check	When enabled, the rotary angle is compared to target	*On / Off*
Expected absolute rotary Angle	Target value for comparison	• numerical value • any Data Reference
Settings for Value Check	explained in Common Inputs	
Enable Capping Angle Check	When enabled, the total capping angle is compared to target	*On / Off*
Expected total Capping Angle	Target value for comparison	• numerical value • any Data Reference
Settings for Value Check	explained in Common Inputs	
Enable Linear Position Check	When enabled, the linear position is compared to target	*On / Off*
Expected linear Position	Target value for comparison	• numerical value • any Data Reference
Settings for Value Check	explained in Common Inputs	
Enable Timeout	When enabled, the command continues with a 'Go To' after the timeout occurs	*On / Off*
Timeout Duration	Define the time until the timeout occurs	any positive, numerical value
Behaviour if out of Tolerance or Timeout:	Command stops with error or performs Go To	• *Direct Error* • any of the existing Labels

Functionality:

For each enabled check, the actual value of the capper is compared to the expected value. If the difference is lower than the value specified in *Allowed Tolerance (+/-)* for the duration of *Evaluation time*, the check is considered successful. For the command to be successful, all enabled checks have to be successful once, but not necessarily at the same time. Once all enabled checks were successful at least once, the command continues. If a timeout is enabled and the command did not succeed within the *Timeout Duration*, an error is reported or a 'Go To' is performed.
The *Rotary Angle Check* uses the *Rotary Capper Position* (see Control) as actual value for comparison.
The *Capping Angle Check* uses the *Capping Angle* (see Control) as actual value for comparison.
The *Linear Position Check* uses the *Linear Capper Position* (see Control) as actual value for comparison.

4.10.7 Check Standstill

Checks the actual velocity of the rotary and linear axis against zero with a certain tolerance and evaluation time. If any of the checks is not successful, an error is returned or a Go To is performed.

Inputs:

Input	Description	Allowed Values
Select Capper	Choose the capper to operate	any of the available capper instances
Enable Rotary Standstill Check	When enabled, the speed is checked for standstill	*On / Off*
Expected Standstill Speed	Speed to accept as standstill	• positive, numerical value unequal zero • any Data Reference
Evaluation time	Duration condition must be fulfilled	• positive, numerical value

Enable Linear Standstill Check	When enabled, the speed is checked for standstill	*On* / *Off*
Expected Standstill Speed	Speed to accept as standstill	• positive, numerical value unequal zero • any Data Reference
Evaluation time	Duration condition must be fulfilled	• positive, numerical value
Enable Timeout	When enabled, the command continues with a 'Go To' after the timeout occurs	*On* / *Off*
Timeout Duration	Define the time until the timeout occurs	any positive, numerical value
Behaviour if out of Tolerance or Timeout:	Command stops with error or performs Go To	• *Direct Error* • any of the existing Labels

Functionality:

For each enabled check, the actual velocity of the capper is compared to zero. If the difference (i.e. the absolute value of the velocity) is lower than the value specified in *Expected Standstill Speed* for the duration of *Evaluation time*, the check is considered successful. For the command to be successful, all enabled checks have to be successful once, but not necessarily at the same time. Once all enabled checks were successful at least once, the command continues. If a timeout is enabled and the command did not succeed within the *Timeout Duration*, an error is reported or a 'Go To' is performed.
The *Rotary Standstill Check* uses the *Rotary Capper Velocity* (see Control) as actual value for comparison.
The *Linear Standstill Check* uses the *Linear Capper Velocity* (see Control) as actual value for comparison.

4.10.8 Check Force/Torque

Checks the position of the rotary and linear axis and the total capping angle against a specified expected value with a certain tolerance and evaluation time. If any of the checks is not successful, an error is returned or a Go To is performed.

Inputs:

Input	Description	Allowed Values
Select Capper	Choose the capper to operate	any of the available capper instances
Enable Torque Check	When enabled, the applied torque is compared to target	*On* / *Off*
Expected Torque	Target value for comparison	• positive, numerical value • any Data Reference
Torque Direction	Direction of the applied torque	*Clockwise* / *Counter-clockwise* (see Setup)
Settings for Value Check	explained in Common Inputs	
Enable Force Check	When enabled, the linear force is compared to target	*On* / *Off*
Expected Force	Target value for comparison	• positive, numerical value • any Data Reference
Force Direction	Direction of the applied force	*Towards cap* / *Away from cap* (see Setup)
Settings for Value Check	explained in Common Inputs	
Enable Timeout	When enabled, the command continues with a 'Go To' after the timeout occurs	*On* / *Off*
Timeout Duration	Define the time until the timeout occurs	any positive, numerical value

Behaviour if out of Tolerance or Timeout:	Command stops with error or performs Go To	• *Direct Error* • any of the existing Labels

Functionality:

For each enabled check, the actual value of the capper is compared to the expected value. If the difference is lower than the value specified in *Allowed Tolerance (+/-)* for the duration of *Evaluation time*, the check is considered successful. For the command to be successful, all enabled checks have to be successful once, but not necessarily at the same time. Once all enabled checks were successful at least once, the command continues. If a timeout is enabled and the command did not succeed within the *Timeout Duration*, an error is reported or a 'Go To' is performed.
The *Torque Check* uses the *Rotary Capper Torque* (see Control) as actual value for comparison.
The *Force Check* uses the *Linear Capper Force* (see Control) as actual value for comparison.

4.11 Common Inputs

Program Behaviour:

Input	Description	Allowed Values
Continue to next line when motion...	*has finished*: wait for this command to finish before moving to next row *has started*: move to next row as soon as execution of this command has started	*has finished* or *has started*

Logic Expression:

Allows to define an arithmetic expression which is evaluated logically or numerically.

Input	Description	Allowed Values
Number	a numerical constant used in the expression	any numerical value
Variable	a Data Reference whose value is evaluated at runtime to be used in the expression	any Data Reference

The arithmetic expression can be created by combining numbers, Data References and operators and is evaluated either logically (expression is TRUE or FALSE) or numerically (e.g., when assigning a value to a Data Reference as in Calculate Value command). A maximum of 32 expression elements (numbers, Data References or operators) is allowed. Trying to add more will show the warning "Max. Elements reached". Additionally, the number of characters in the expression is limited to 255. Exceeding this limit will show the warning "Max. Characters reached". This limit should usually not be reached as long as a valid expression is entered. If either the maximum number of elements or characters is reached, it is advised to split the expression into several sub-expressions and storing the interim results in a variable using the Calculate Value command.
The entered expression can be calculated by using the *Test* button. It will evaluate the expression using current values for Data References. Please note that the value of the result can change during program execution, as Data Reference values can change too. The result of the expression is shown in the field to the right of the button. If the expression is invalid and cannot be evaluated (e.g., lacking arguments for operators, missing brackets, invalid operators, division by 0), the result will show 0.0 and the warning "Invalid Equation" will be displayed.

The precedence of the operators is as listed from highest to lowest precedence:
- Brackets: *(* and *)*
- Unary operators: *NOT* or - before numbers, brackets or variables
- Multiplication and division: *x* and */*
- Addition and subtraction: + and -
- (In-)equality: =, <>, <, <=, > and >=
- Logical conjunction: *AND*
- Logical disjunction: *OR*

The result of the expression will automatically be converted to a logical or numerical value, depending on the last performed operation. A logical result will still be a floating-point numerical value with 0.0 representing FALSE and 1.0 representing TRUE. Similarly, a numerical value of 0.0 is interpreted as logical FALSE and any other numerical value is interpreted as TRUE. The expression "5 > 2 + 4" will return 0.0, as the last operation based on precedence is "5 > 6" and therefore a logical operation. The expression "(5 > 2) + 4" will return 5.0, as the last operation is "1.0 + 4" and therefore a numerical operation.

Timeout:

Allows to set a timeout. If the command was not performed in the specified duration, a Go To to the specified label is performed.

Input	Description	Allowed Values
Enable Timeout	When enabled, the command continues with a 'Go To' after the timeout occurs	*On* / *Off*
Timeout Duration	Define the time until the timeout occurs	any positive, numerical value
Go To Label	Select a label to jump to on timeout	any of the existing Labels

Kinematics:

Sets the kinematic parameters of a movement.

Input	Description	Allowed Values
Presets	buttons to set and indicate kinematic parameter values	*Slow*, *Moderate*, *Fast* or deselected
Velocity	the velocity of the motion in m/s (linear axis) or °/s (rotary axis)	• any positive, numerical value • any Data Reference
Acceleration	the acceleration of the motion in m/s^2 (linear axis) or $°/s^2$ (rotary axis)	• any positive, numerical value • any Data Reference
Deceleration	the deceleration of the motion in m/s^2 (linear axis) or $°/s^2$ (rotary axis)	• any positive, numerical value • any Data Reference
Jerk	the jerk of the motion in m/s^3 (linear axis) or $°/s^3$ (rotary axis)	• any positive, numerical value • any Data Reference

Clicking any of the three *Presets* buttons will set the Data References to the according Motion Defaults for speed and category. Simultaneously, the buttons also indicate if all four Data References are set to the correct Motion Defaults, as shown in Fig. 19. If any of the kinematic parameters is set to a numerical value or a different Data Reference, none of the buttons will be displayed as pressed.



Fig. 19: The preset buttons indicate that all Data References are currently set to the slow motion defaults of the appropriate category.

The slug indicates *Slow*, the duck *Moderate* and the rabbit *Fast* Motion Defaults. When inserting a command into an empty row, the kinematic parameters will be set to *Slow* by default.

Pre-Position:

If enabled, the gripper jaw(s) will move to the specified pre-position with separate kinematic values before starting the gripping process.

Input	Description	Allowed Values
Enable Pre-Position	enable the motion to the pre-position	*On* / *Off*

Pre-Position Mode	defines how the actual pre-position is calculated	• *Absolute:* Move gripper jaw(s) to the specified pre-position • *Relative:* Move gripper jaw(s) to the specified distance from the gripping position or actual position
Pre-Position	the actual pre-position (*Absolute*) or distance to actual pre-position (*Relative*)	• any positive, numerical value • any Data Reference
Positioning Speed	explained in Common Inputs (Kinematics)	

If *Relative* is selected as *Pre-Position Mode*, the actual pre-position is calculated based on the selected command and actual position.

- **Grip by Position**:
  The actual pre-position is set the specified distance (absolute value taken) away from the *Target Position against* the direction of motion.
  E.g.: if the current jaw position is 4.2 mm, target position is 8.4 mm and the pre-position is set to 1.5 mm. The actual pre-position is calculated as 8.4 - 1.5 = 6.9 mm as the direction of motion is towards positive (from 4.2 mm to 8.4 mm).
  The gripper jaw(s) will never move against the direction of motion, even if the calculated pre-position does not lay between the current and the target position.
  E.g.: if the current jaw position is 4.2 mm, target position is 3.1 mm and the pre-position is set to 1.5 mm. The actual pre-position is calculated as 3.1 + 1.5 = 4.6 mm as the direction of motion is towards negative (from 4.2 mm to 3.1 mm). However, since this would require the jaw(s) to move backwards first, the pre-position is set to the current position of 4.2 mm (effectively no pre-position motion).
- **Grip by Force**:
  The *Relative* mode is only allowed if the *Position Check* is enabled, since the *Target Position* of the position check is used as reference. The calculation of the actual pre-position is handled the same way as for *Grip by Position*.
- **Ungrip**:
  The actual pre-position is set the specified distance (absolute value taken) away from the current position *along* the direction of motion.
  E.g.: if the current jaw position is 4.2 mm, target position is 8.4 mm and the pre-position is set to 1.5 mm. The actual pre-position is calculated as 4.2 + 1.5 = 5.7 mm as the direction of motion is towards positive (from 4.2 mm to 8.4 mm).

Operation Mode of Linear Axis:

Capping commands allow for different modes of synchronization between the linear and rotary axis, which are outlined below.

- **Ignored:** The current control mode of the linear axis is kept.
- **Apply Force:** The linear axis is set to force control mode and a constant force is applied.
  If the position of the linear axis exceeds the *Lower Limit Position*, the command is aborted and the *Position-Check failed* flag (see Control) is set.
- **Sync. with Rotary Axis:** The linear axis is coupled positionally to the rotary axis. Allows to follow a known thread with minimal friction.
  If a *Lift-Off-Distance* is defined, the linear axis will move the configured distance opposite the selected *Linear Direction* before the capping starts.

Input	Description	Allowed Values
Force to Apply (for mode *Apply Force*)	Force applied during tightening	• any positive, numerical value • any Data Reference
Force Direction (for mode *Apply Force*)	Direction of the applied force	*Towards cap* / *Away from cap* (see Setup)
Lower Limit Position (for mode *Apply Force*)	Stop position in case no bottle	• any numerical value • any Data Reference

Distance per turn (for mode *Sync. with Rotary Axis*)	Linear Distance per 360°	• any numerical value • any Data Reference
Linear Direction (for mode *Sync. with Rotary Axis*)	Select direction in which linear axis moves	*Towards cap* / *Away from cap* (see Setup)
Lift-Off-Distance (for mode *Sync. with Rotary Axis*)	To minimize thread contact	• any numerical value • any Data Reference
Lift-Off Speed (for mode *Sync. with Rotary Axis*)	explained in Common Inputs (Kinematics)	

Settings for Value Check:

Capping commands allow for a range of values to be checked against an expected value.

Input	Description	Allowed Values
Allowed Tolerance (+/-)	Related to expected value	• positive, numerical value unequal zero • any Data Reference
Evaluation time	Duration condition must be fulfilled	• positive, numerical value

5 Applications

5.1 LinMot Grippers

The LinMot Grippers Application is used to control grippers with one or two active jaws. If the license is available, the according program commands and the control and setup pages are available. Up to eight gripper instances can be added during the Kinematic Configuration.

5.1.1 Control

The control page shows the current state of the process output of each gripper instance by clicking the according tab in the header row.

Process Outputs:

- **Error:** An error occurred in the application
- **Grip Active:** A grip is currently active and the jaws are in contact
- **Monitoring Active:** Position or force monitoring for gripper is active
- **Position Above Target:** The current position is larger than the specified monitoring window
- **Position Below Target:** The current position is smaller than the specified monitoring window
- **Force too High:** The current force is higher than the specified monitoring window
- **Force too Low:** The current force is lower than the specified monitoring window
- **Gripper Position:** Usually the distance between the gripper jaws
  For two-axis grippers, this value is equal to the sum of positions of both jaws. For one-axis grippers, this value is equal to the position of the first jaw.
- **Position Jaw 1:** The current position of the first jaw
- **Position Jaw 2:** The current position of the second jaw
  Only available for two-axis grippers
- **Force Jaw 1:** The current force of the first jaw
- **Force Jaw 2:** The current force of the second jaw
  Only available for two-axis grippers.

5.1.2 Setup

The setup page allows to adjust general and instance-specific settings of the gripping application.

Default Speed Parameters:

The default speed parameters can be adjusted in the general tab and are used by all gripper instances and available as Data References.

Individual Gripper Settings:

The following settings can all be individually defined for each gripper instance.

- **Gripper Name:** Define a unique name for this gripper
  The gripper name is shown in dropdowns for gripper selections.
- **Stopping Deceleration:** Deceleration used for halting motions
- **Stopping Jerk:** Jerk used for halting motions
- **TimeOut Time:** Timeout for internal processes
- **Force Limit during Pre-Positioning:** Max. force allowed during pre-psoition motion
  If this force is exceeded, the gripping command will abort and an error will be returned
- **Set Force during Pre-Positioning:** Applied force if force limit was reached
  If the force limit is exceeded, this force will be applied
- **Position Transmission Factor:** Motor position is multiplied by this value to get gripping position
  This factor is required for certain grippers and will be automatically set for all LinMot Grippers
- **Direction Definition:** "Closing" is defined as moving towards ... positions
  Allows defining direction of "Opening" and "Closing", based on the direction definitions of the grippers axes

5.1.3 Widgets

The gripper widget can be added in the Quick Control.

Gripper Control:

The gripper control widgets can be used to manually grip and ungrip a gripper instance. Clicking the *Grip* or *Ungrip* button executes the Grip by Force or Ungrip command with predefined speed parameters in the background. The *Target Force* is used for the *Grip* command of the widget.
When editing of the Quick Control is enabled by clicking the button in the top right corner, additional configuration options are available:

- **Gripper Instance:** Select the gripper instance, that is controlled by this widget
- **Ungrip Position:** Set the position, to which the ungrip motion is performed
- **Gripping Velocity:** Set the velocity of the gripper during gripping
- **Gripping Direction:** Select the direction of the gripping motion

5.2 Conveyor Tracking

The Conveyor Tracking Application lets the robotics group pick, place, or process parts on a moving conveyor belt without stopping the belt. The belt position is measured with an encoder, and the position of every detected part (target) is tracked continuously.

The application consists of the setup described in Connecting and Configuring a Conveyor, the runtime control on the Control tab, and the program commands Sync to Active Target, Move Synchronized, Move Robotic, Desync from Conveyor, Set Active Target, Clear Buffer, Clear Target from Buffer, Wait for Active Target, and Control Target Detection. Errors are listed in List of Conveyor Events.

5.2.1 Control

Process Outputs:

- **Conveyor State:** TRUE when a valid conveyor is assigned to this instance (correctly initialized), FALSE otherwise. Shown as a diagnostic aid.

- **Tracking Active:** TRUE while the axis group is synchronized with a conveyor and follows the moving coordinate system (set by Sync to Active Target, released by Desync from Conveyor).

Target Overview:

Below the Process Outputs section, the Target Overview is displayed.
The conveyor belt visualization shows all currently detected targets and their positions on the conveyor. The active target is highlighted in green.
Below this visualization, an overview table lists all detected targets. The active target is also highlighted in green in the table.
To delete all targets, click the Delete button in the title bar. To delete a single target or set a specific target as active, click the corresponding button in the row of the desired target.

5.2.2 Connecting and Configuring a Conveyor

Conveyor tracking allows a robotics group to pick, place, or process parts while they are transported on a moving conveyor belt, without stopping the belt for each operation. To do so, LinMot-Pilot continuously tracks the belt movement with an encoder and calculates the position of every part in real time.

The calculation is based on three coordinate systems that build upon each other:
- **MCS – Machine Coordinate System** is the fixed reference frame of the machine.
- **CCS – Conveyor Coordinate System** is attached to the conveyor belt and is defined relative to the MCS.
- **TCS – Target Coordinate System** describes an individual part (target) on the belt and is defined relative to the CCS.

Setting up a conveyor is done in three stages, which are described step by step in the following sections:
- **Connect the encoder** to any drive in the system.
- **Run the Kinematics Wizard** to assign the encoder to an axis and route its value to LinMot-Pilot through a monitoring channel.
- **Configure the Conveyor Belt Settings** to define the conveyor geometry, the trigger and target behavior, the working zones, and the encoder scaling.

Connecting the Encoder:
The conveyor movement is measured by an incremental encoder, typically driven by a measuring wheel that runs on the belt. The encoder is wired to the *X13* connector of any drive in the system – this does not have to be a drive that moves the robotics group. The selected drive counts the encoder pulses and makes the counter value available to LinMot-Pilot through a monitoring channel, which is configured in the following steps.

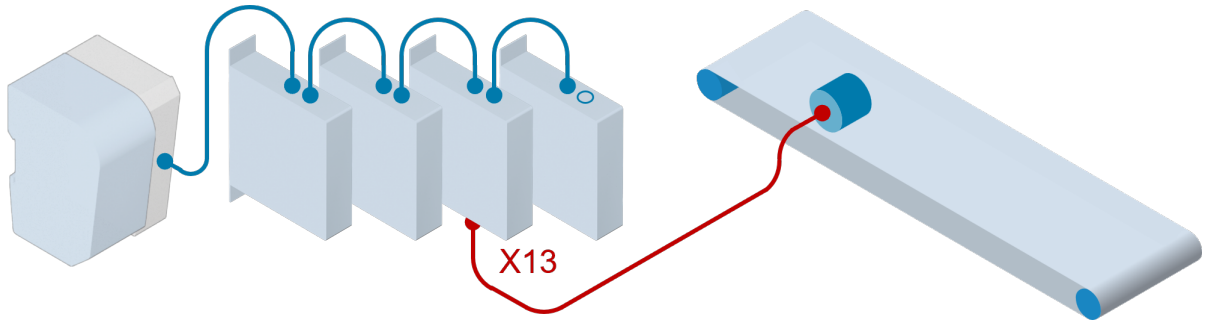


Fig. 20: Encoder wired to a drive via X13 and measuring the conveyor belt.

Note: Note to which drive the encoder is connected. This drive must be selected in the Kinematics Wizard so that its encoder counter is transferred to LinMot-Pilot.

Kinematics Wizard – Conveyor:
The encoder is assigned to a drive in the *Conveyor* step of the Kinematics Wizard. The table lists every axis in the system with its *Sensor Type* and *Monitoring Channel*. For the axis whose drive the encoder is connected to, select the sensor type that matches the encoder. All other axes remain set to *No Sensor*.

The following sensor types are available:

- **Incremental ABZ Encoder** for an encoder with an additional zero (index) pulse Z.
- **Incremental AB Encoder** for an encoder with the A and B tracks only.
- **No Sensor** for axes without an encoder.

The *Monitoring Channel* defines which channel transfers the encoder counter to LinMot-Pilot for the conveyor tracking calculations. Selecting the channel here automatically maps the encoder counter to this channel on the following *Channels* page of the wizard. Note the selected channel number, as it is needed later when the *Encoder Counter Variable* is assigned in the conveyor belt settings. *Count Direction* sets the count direction of the encoder. The movement direction of the conveyor belt has to align with the X-axis of the kinematic configuration.

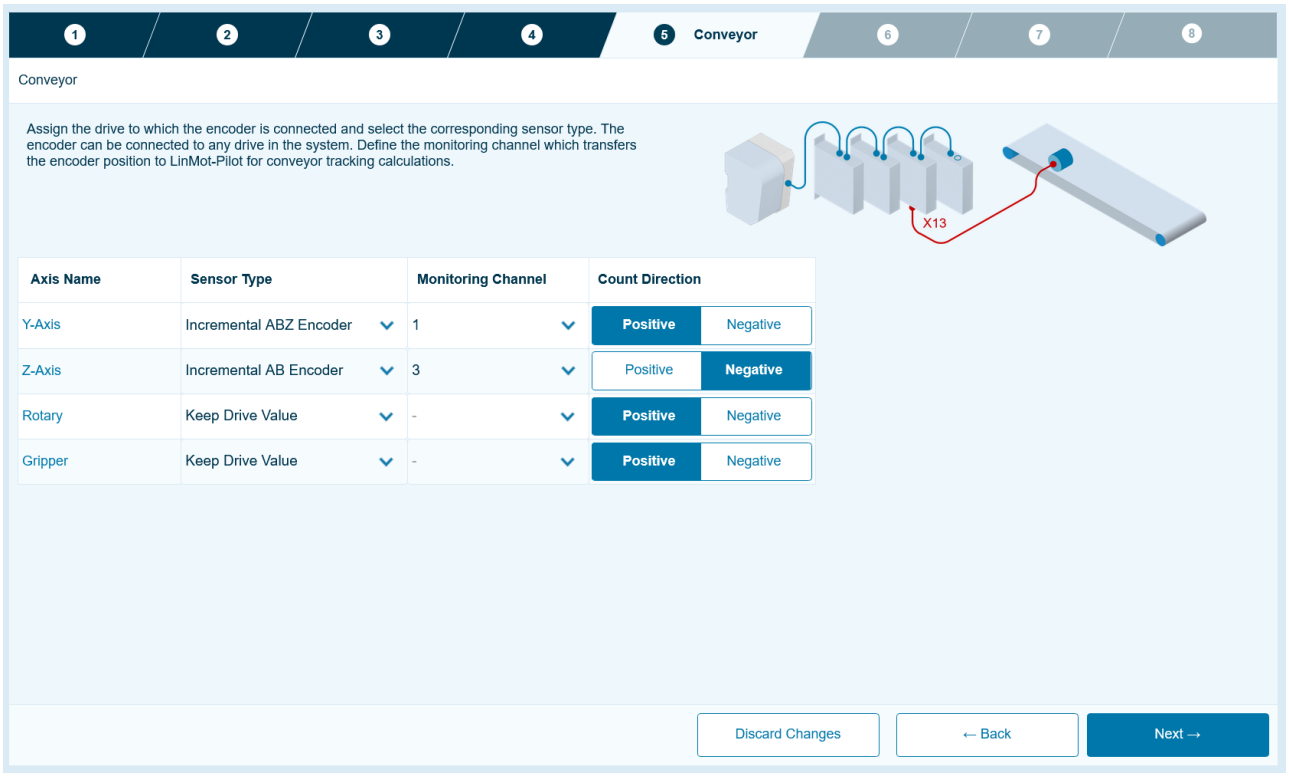


Fig. 21: Assigning the encoder to a drive, sensor type and monitoring channel in the Kinematics Wizard.

Kinematics Wizard – Channels:
Because a monitoring channel was selected in the previous step, the encoder counter parameter (UPID *1B84h*, “Ext. Enc. Counter”) is already entered on that channel for the corresponding axis. No further change is required for basic conveyor tracking – simply verify that the “Ext. Enc. Counter” entry appears on the expected channel. This monitoring channel is the data reference that is later selected as the *Encoder Counter Variable*.

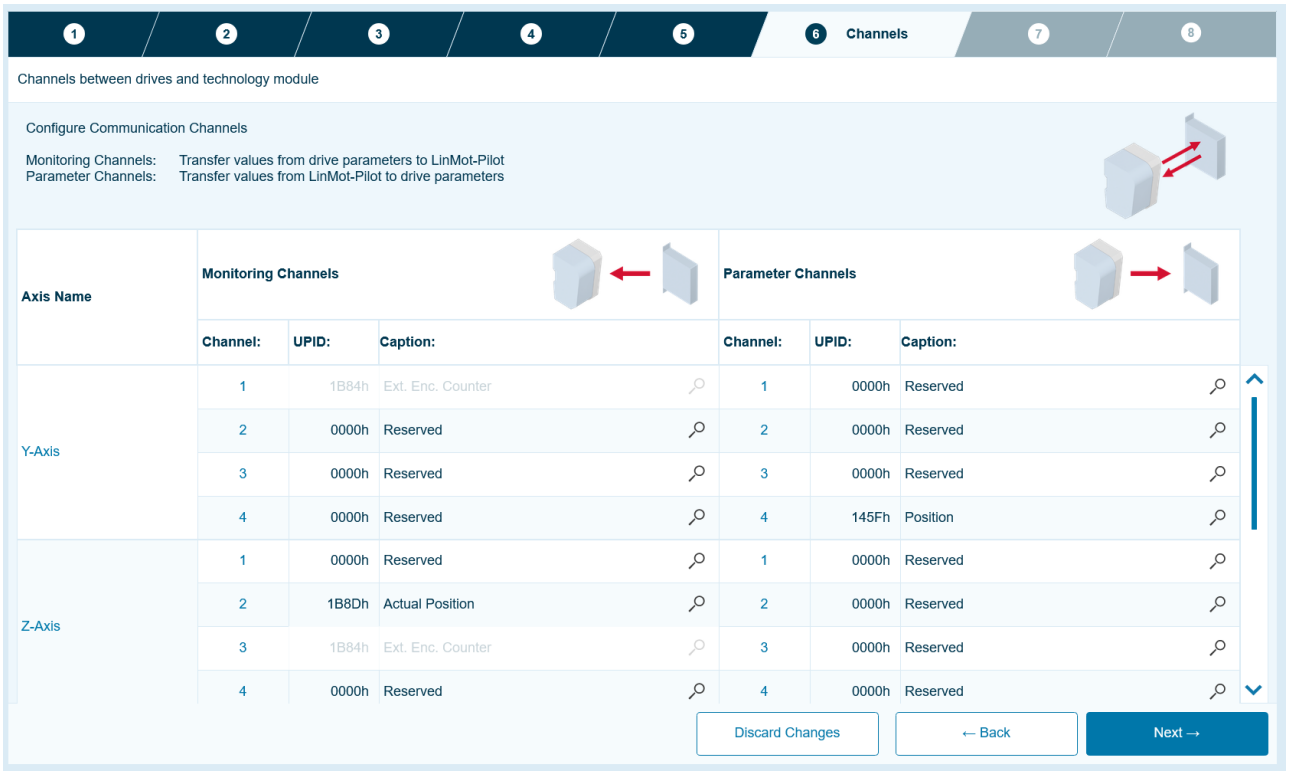


Fig. 22: Communication channels – the encoder counter (Ext. Enc. Counter, UPID 1B84h) is mapped automatically to the selected monitoring channel.

General Conveyor Belt Settings:

After the Kinematics Wizard is completed, the conveyor is configured on the conveyor belt settings page. Several conveyors can be managed on separate tabs; a new one is added with *Add Conveyor*.

First, a name is assigned to the conveyor in the *Conveyor Name* field. This name is used throughout LinMot-Pilot to identify the conveyor in the visualizations.

The position and orientation of the belt are defined by the Conveyor Coordinate System (CCS) relative to the Machine Coordinate System (MCS). It is recommended to start with the orientation and then set the offsets:

- **Conveyor rz** sets the orientation, i.e. by how many degrees the belt moving direction differs from the x-axis of the robotics group. The visualization updates in 90° steps and serves as a visual aid only.
- **Conveyor x**, **Conveyor y** and **Conveyor z** set the origin of the CCS relative to the MCS.

The offsets can be entered manually or taught: manually move the robot to the origin of the conveyor and press the teaching buttons for the x, y and z positions.

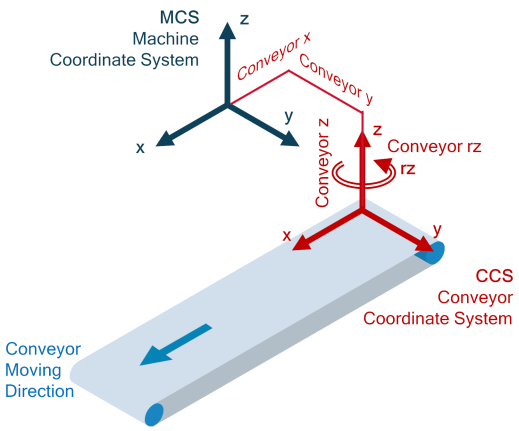


Fig. 23: The Conveyor Coordinate System (CCS) is defined relative to the Machine Coordinate System (MCS).

Trigger & Target Settings:
This section defines how targets are added to the buffer and how they are aligned on the belt. The *Trigger Mode* selects how new targets are generated:

- **By Signal** adds a target when a trigger signal is received, for example from a light barrier.
- **By Data Reference** adds a target when the configured data reference triggers.
- **By Distance** adds a new target automatically each time the belt has moved a defined distance. The trigger is only active for the forward belt direction.

The *Trigger Data Reference* selects the data reference that is used for triggering. When *By Data Reference* is selected, an additional *Trigger Threshold* appears, which defines the threshold value at which the data reference triggers.

Three options control whether target detection continues in special system states:

- **Continue Target Detection During Pause** keeps adding new targets to the buffer while the system is paused.
- **Continue Target Detection During Error** keeps adding new targets to the buffer while the system is in an error state.
- **Continue Target Detection when completed or aborted** keeps adding new targets until detection is stopped by command or manually.

The alignment of a target on the belt is set with **Target x**, **Target y** and **Target z** and **Target rz**, which define the origin and orientation of the Target Coordinate System (TCS) relative to the CCS. As with the conveyor offsets, these values can be entered manually or taught.

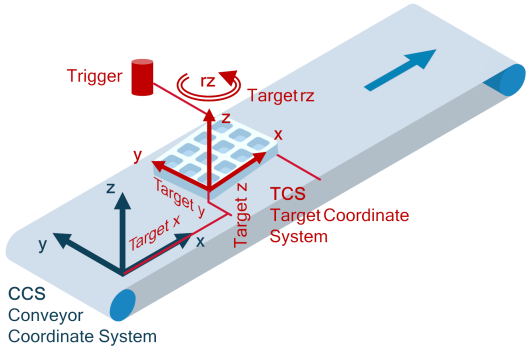


Fig. 24: The Target Coordinate System (TCS) describes a part on the belt, defined relative to the CCS.

Conveyor Positions:
The conveyor positions divide the belt into zones that a part passes through as it travels along the belt: the *Queue*, the *Working Area*, the *Critical Area*, and finally *Out of Reach*. The boundaries of these zones are defined by the following positions, measured along the belt from the conveyor origin:

- **Earliest Sync Position** is the earliest position at which a part can be reached by the robot. It marks the start of the working area.
- **Critical Zone Position** is the start of the critical zone, for example to slow down the conveyor so that a part can still be processed.
- **Latest Sync Position** is the end of the critical area: parts beyond this position are out of reach for the robot.
- **Latest Desync Position** is the latest reachable part position and marks the end of the working area. Parts that pass this position are lost and are removed from the buffer.

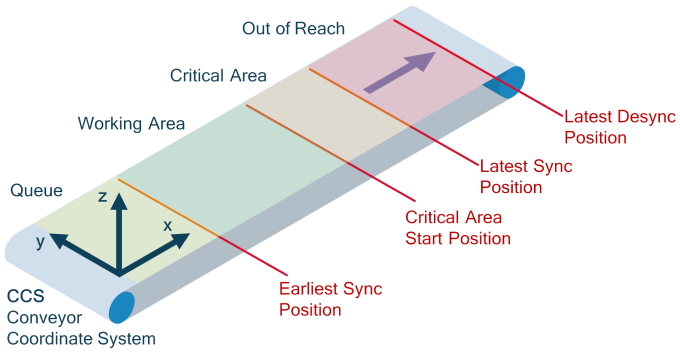


Fig. 25: Conveyor zones and the positions that define their boundaries.

Encoder Settings:
The encoder settings convert the raw encoder counter into a real conveyor position. First, select the *Encoder Counter Variable* – the variable mapped to the encoder counter. The easiest option is the monitoring channel of the drive to which the encoder is connected, as configured in the Kinematics Wizard.
The remaining parameters define the scaling and filtering:

- **Increments per Rotation** is the number of pulses for one full rotation of the measuring wheel (see datasheet).
- **Measuring wheel circumference** is the belt travel distance of one full wheel rotation (see datasheet).
- **Bit Width** is the size of the encoder counter (default: 16).
- **Position Filter** reduces noise in the position signal (default: 10).
- **Velocity Filter** reduces noise in the velocity signal (default: 5).

After saving the changes, the scaling can be verified in the section at the bottom of the page. Move the conveyor by hand or with the drive and compare the calculated values with the real movement:

- **Current Encoder Counts** are the encoder counts since the last reset (4 counts per encoder pulse). Use *Reset* to set them back to zero.
- **Calculated Conveyor Position** is used to verify the calculated position against the real conveyor position.
- **Conveyor Speed** is used to verify the calculated speed against the real conveyor speed.

If the calculated position and speed match the real movement of the belt, the encoder scaling is correct.

Delete Conveyor Belt:
Changes on the conveyor belt settings page take effect only after they are applied with *Save Changes*; *Reset Changes* discards them. To remove the current conveyor, press *Delete Conveyor* at the end of the page.

5.3 Capping

The Capping Application is used to control cappers with one linear and one rotary axis. If the license is available, the according program commands and the control and setup pages are available. Up to eight gripper instances can be added during the Kinematic Configuration.

5.3.1 Control

The control page allows direct control of the capper axes and shows the current state of the process output of each capper instance by clicking the according tab in the header row.

Capper Control:

The linear and rotary axis of the capper can be controlled manually similar to the Axis Control but with limited functionality and displayed information. When jogging the axes using the arrow-buttons, the jogging speed from the slider in the Title Bar is used and the direction of motion is based on the configured direction definitions of the cappers Setup page. Jogging using the upwards pointing arrow will move the linear capper axis away from the cap (and vice versa) and jogging using the left-pointing arrow will move the rotary capper axis in clockwise direction (and vice versa).

Process Outputs:

- **Process State*:** The current state of the application process
- **Linear Capper Position:** Current position of the linear axis of the capper
- **Linear Capper Force:** Current force of the linear axis of the capper
- **Linear Capper Velocity:** Current velocity of the linear axis of the capper
- **Rotary Capper Position:** Current position of the rotary axis of the capper
- **Rotary Capper Torque:** Current torque of the rotary axis of the capper
- **Rotary Capper Velocity:** Current velocity of the rotary axis of the capper
- **Capping Angle:** Total angle that was covered during the capping process
- **Command aborted:** The latest command was aborted
- **Command timed out:** The latest command timed out
- **Position-Check failed:** The position of the linear axis was out of tolerance
- **Angle-Check failed:** The position of the rotary axis was out of tolerance
- **Capping Angle-Check failed:** The total capping angle was out of tolerance
- **Force-Check failed:** The force of the linear axis was out of tolerance
- **Torque-Check failed:** The torque of the rotary axis was out of tolerance
- **Linear Standstill-Check failed:** The linear axis was not in standstill
- **Rotary Standstill-Check failed:** The rotary axis was not in standstill
- **Error:** An error occurred in the application

***Values of Process State:**

-1	Error occurred
0	Command finished / process ready
10	Process Stopping
101	Find Seating Point: Motion active, waiting for force and standstill
102	Find Seating Point: Force and standstill detected, stopping motion and halting axis
201	Find Thread Start: Motion active, waiting for thread detection
202	Find Thread Start: Thread detected, stopping motion and halting axis
301	Cap to Angle
401	Cap to Torque: Pre-positioning active
402	Cap to Torque: Initial Torque Point detection
403	Cap to Torque: Tightening to target torque
404	Cap to Torque: Target torque and standstill reached, changing axes to position control and finishing
501	Check Values

5.3.2 Setup

The setup page allows to adjust general and instance-specific settings of the capping application and add capping instances.

Default Speed Parameters:

The default speed parameters can be adjusted in the general tab and are used by all capper instances and available as Data References. The defaults can be set for linear and rotary axes individually.

Individual Capper Settings:

The following settings can all be individually defined for each capper instance.

- **Capper Name:** Set custom name for capper that is used in visualizations
  The capper name is shown in dropdowns for capper selections.

- **Linear Capper Axis:** Select the linear axis for the capper
  Only linear axes can be selected. If the axis is part of the robotics group, simultaneous robotics / conveyor and capping commands are not possible.
- **Rotary Capper Axis:** Select the rotary axis for the capper
  Only rotary axes can be selected. If the axis is part of the robotics group, simultaneous robotics / conveyor and capping commands are not possible.
- **Stopping Deceleration (Linear):** Deceleration used for halting motions
- **Stopping Jerk (Linear):** Jerk used for halting motions
- **Stopping Deceleration (Rotary):** Deceleration used for halting motions
- **Stopping Deceleration (Jerk):** Jerk used for halting motions
- **Towards Cap Definition:** "Towards Cap" is defined as moving towards ... positions
  Allows defining direction of "Towards Cap" and "Away from Cap", based on the direction definitions of the linear axis and the capper setup
- **Clockwise Definition:** "Clockwise" is defined as moving towards ... positions
  Allows defining direction of "Clockwise" and "Counter-clockwise", based on the direction definitions of the rotary axis and the capper setup
- **Delete Capper:**
  Deletes the capper. Axes can still be used for axes motions, but capper instance is not available any more.

Adding Capper:

Capping instances can be added by clicking *Add Capper* in the title bar and configuring the newly opened tab.

6 System

6.1 System States

LinMot-Pilot contains a main state machine to control the system (startup, error handling, etc.) and the program (execution and control of programs, light gray part in Fig. 26) which is an adaption of the PackML State Diagram (see the OMAC Website for more detail).

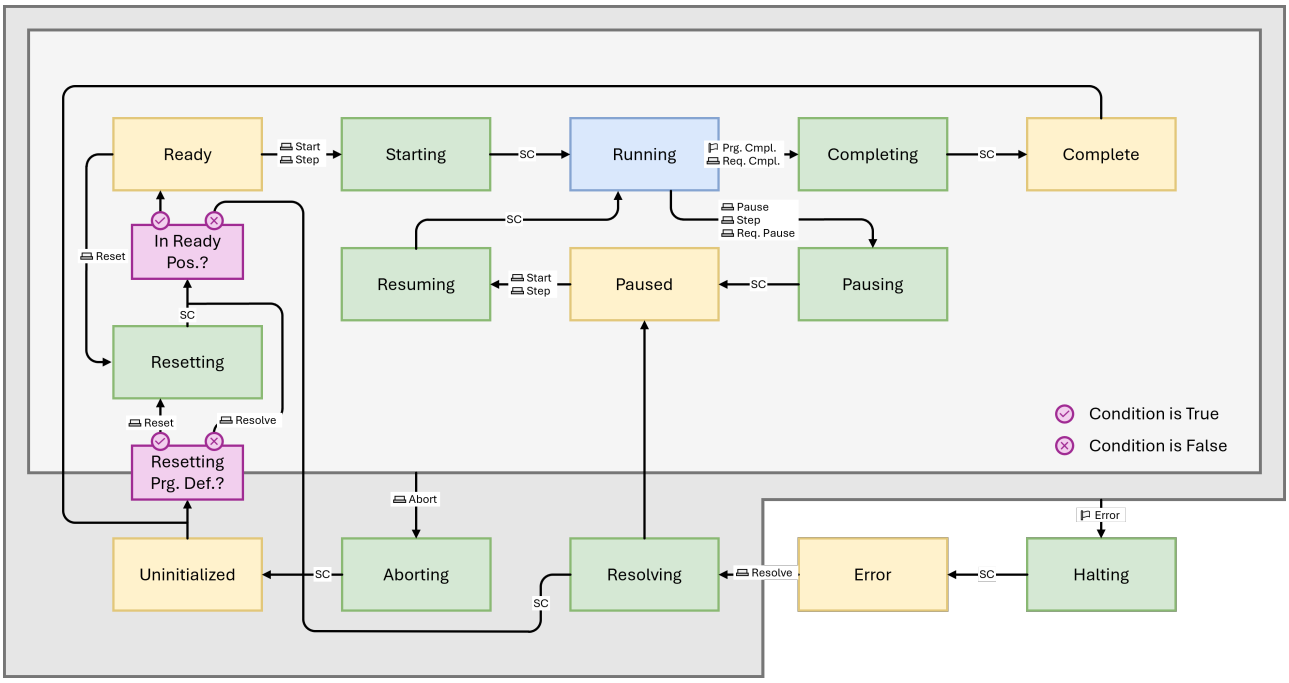


Fig. 26: The state machine of LinMot-Pilot.

Fig. 27 shows the available states and how changes between the states can occur. *SC* indicates that the state changed will be performed automatically without user input. Otherwise, required button presses (button symbol) or conditions (flag symbol) are indicated. Green states are acting states, which represent some processing activity. Orange states are wait states which are used to identify that the system has achieved a defined set of conditions.

The individual states are listed below with the *Numerical Value* representing the value that will be transmitted as Outputs of the Fieldbus interface.

Name	Numerical Value	Description
Startup	0	System is still starting up. Will automatically move to *Uninitialized* state after startup.
Halting	1	Program is halting motions
Error	2	An error has occurred and must be resolved
Resolving	3	Errors are being resolved
Aborting	4	Program was stopped and is halting motions
Uninitialized	5	System is in a potentially undefined state and can be moved to ready position
Complete	6	Program was completed and system is in a potentially undefined state
Resetting	7	System es executing the resetting program
Ready	8	System is in defined position from which a program can be started
Starting	9	Program is being started but not yet running
Pausing	10	Program is being paused and is halting motions
Paused	11	Program is paused and can be resumed
Resuming	12	Program is being resumed but not yet running
Running	13	Program is running
Completing	14	Program has finished execution and is waiting for all processes to finish

6.2 Program Execution

Programs can be executed by button inputs, PLC commands or other program commands. The program execution is linked to the System States in a way so the system state represents the program execution state.
If a program is run in LinMot-Pilot, it is processed row by row and the commands are executed sequentially. If the *Program Behaviour* (see Common Inputs) of a command is set to *has finished*, LinMot-Pilot will move to the next valid row once the end conditions of the command have been reached. The next valid row is the first row below the current row that contains a valid command (except for *Labels* and *Comments*) and is not commented out. If the *Program Behaviour* is set to *has started*, LinMot-Pilot will move to the next valid row immediately after the current command was started. In any case, the processing of the new command will start in the next cycle. As changing of program rows takes at least one cycle, sequential motions will stand still for 2 ms or more between two commands.
Certain commands that can be executed immediately will allow the command in the next valid row to start within the same cycle. Commands such as logical checks (*If ... Go To ...*, *If Motions Done*), program control (*Go To*, *Run Program*) or variable commands (*Set Output (by logic)*, *Set Variable (by logic)*, *Select Dataset*) can therefore be handled in the same cycle as other commands. No more than 10 commands can be executed per program per cycle.

6.2.1 Program Interruption

When a program is interrupted, either manually using the control buttons or due to an error, motions and processes will be paused or stopped and axis modes might change. The detailed behaviour in possible cases is described below.

Definitions:

- **Motion(s):** A single axis or an application (e.g. a gripper or a capper). Motions are selectable in commands such as Wait Motions, Stop Motions or If Motions Done.
- **Program(s):** Programs that were started using the control buttons or the Run Program or Call Program command. Programs start and stop motions and check motions states but programs themselves can be started, paused or stopped independently of motions.

The table below presents an overview of the behaviour in different cases. See below for a more detailed description.

	Pausing	Aborting	Program Pause	Resuming	Restarting
Programs	pause all programs that were not started with *Ignore Pause*	abort all programs that were not started with *Ignore Abort*	pause only affected program(s)	resume all paused programs	resume selected program from selected row, resume all other programs from paused row
Motions	pause all motions	abort all motions	no effect	resume all paused motions	abort all motions
Axes in Force Control	see Stopping Behaviour of Axes				

Pausing:

The motions and programs can be paused manually (by the *Pause* control button or commands from the PLC interface) or automatically due to an error that allows resuming. When pausing, all motions will be stopped and it's targets and current states will be saved. After pausing or resolving the error, the system will be in the *Paused* state (see System States).

Aborting:

The motions and programs can be aborted manually (by the *Abort* control button or commands from the PLC interface) or automatically due to an error that does not allow resuming. When aborting, all motions will be stopped without saving current states. After aborting or resolving the error, the system will be in the *Uninitialized* state (see System States).

Program Pause:

A program pause is caused by enabling stepping (e.g. by clicking the *Step* control button), pausing in the Program Manager or using the Pause on Request command. It will only cause the affected program execution to halt in the active row, while active motions will not be affected by it. The system will be move to the *Paused* state (see System States) even if motions are still running.

Resuming:

If any paused program is resumed at it's active row, all paused programs will be resumed.

Restarting:

Resuming a paused program from a row, different to the active row is not directly permitted. In this case, a restart is performed. The selected program will be restarted from the selected row, while all other paused programs will be resumed from their active row, i.e. from where they were paused at. All paused motions will be stopped and must be restarted again by an according command.

Stepping in Paused State:

If a single program row is stepped while motions are paused a pop up dialog will allow for two options. *Clear all paused motions* will stop all motions and they must be restarted again by an according command. Only the selected program is resumed afterwards. *Keep motions paused* will neither stop nor resume the paused motions and resume only the selected program.

6.2.2 Stopping Behaviour of Axes

The behaviour of axes in force control when the program execution is paused or aborted can be individually configured for each axis to allow for safe operation. E.g., a gripper axis should keep it's force applied when pausing or aborting to keep the part gripped but the linear axis of a capper should set its force to zero, to reduce pressure on the cap.
The axis behaviour on pausing or error, resuming and aborting can be configured in *Setup* → *Pilot Settings* → *Axes Motion* → *Stopping Settings*. The available selections and its effects are described below.

Behaviour on Pausing/Error:

- **Stay in Force Control, keep Force**
  Axis will not change and keep its force before Pause / Error occurred.
- **Stay in Force Control, Force to Zero**
  Axis will set it's force to zero. The applied force before Pause / Error is stored. Axis can be moved by external forces!
- **Switch to Position Control**
  Axis will switch to standstill in position control and keep its position. The applied force before Pause / Error is stored. Based on the previously applied force and velocity, high forces can remain if the set position is in an unreachable state.

Behaviour on Resuming:

- **No change to Paused State**
  Axis will not change, stay in its current control mode and keep the currently applied force.
- **Restore original Force**
  Axis will switch to force control (if switched to position control before) and apply the stored force, which was applied force before Pause / Error.

Behaviour on Aborting:

See options for *Behaviour on Pausing/Error*.

6.2.3 Program Manager

The Program Manager is available in the Control menu. It presents an overview of all currently active programs (i.e., programs that are running, paused or in a transitional state) which corresponds to the color-coded active tabs in the Program visualization.

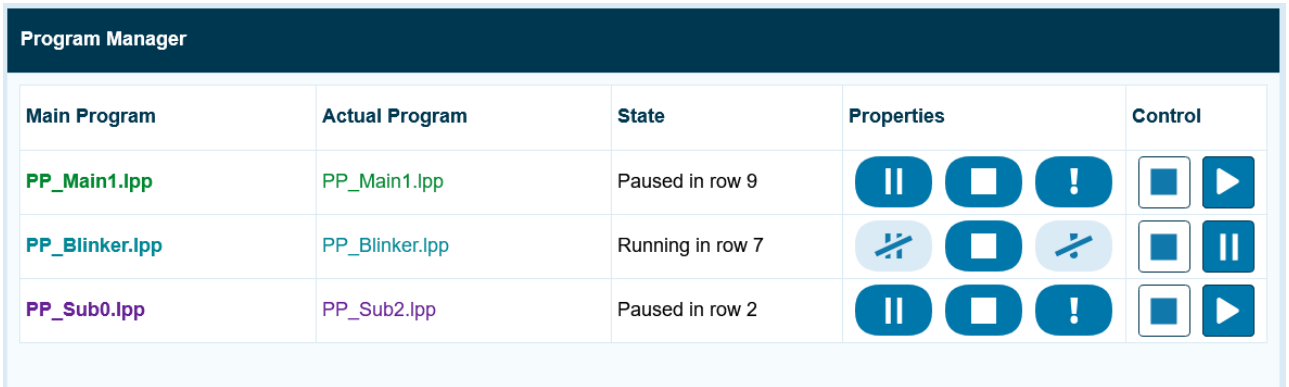


Fig. 28: The program manager lists all active programs with their properties and allows to manually pause, resume or abort individual programs.

Any program can call a sub-program by using the Call Program command. Therefore, the *Actual Program* (i.e., the currently active program) can be different from the *Main Program* (i.e., the program that was initially started). The *Properties* of each program are displayed by icons and represent the input settings of the Run Program command. The control buttons allow to individually abort, pause or resume a program. The overall system state (see System States) can be influenced by manual changes through the program manager.

Note: Aborting, pausing or resuming a program using the program manager will only affect the execution of the individual program but will not immediately stop motion commands or resume buffered motions. For proper pausing, resuming or aborting of motion commands, the control buttons in the Title Bar have to be used.

Using the manual control buttons of the program manager can lead to unexpected behavior. Example: If the system was paused during a motion using the control button in the Title Bar and the program is then resumed through the program manager, the paused motion will not resume and the program execution can become blocked.

Active programs are also indicated by their tab-color in the Program visualization. The color-coding of the tabs is identical to the font color used in the program manager. The tab of the *Actual Program* is always the right-most tab of a group of sub-programs and has a lighter color, than other programs of the same group (see purple program tabs in Fig. 29).

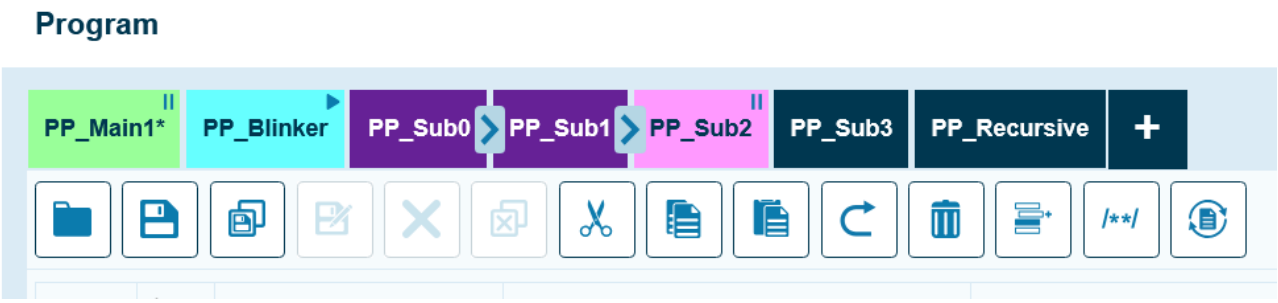


Fig. 29: The program tabs are color-coded to show active programs and their structure.

6.2.4 Program Pre-Compiler

Program rows are continuously checked for invalid configurations by the pre-compiler. If program commands are entered without filling out all required information or if configurations changed and a previously entered command contains now invalid information, program rows are marked red and a warn symbol is shown in the icon column of the program. Hovering over the warn symbol with the mouse cursor shows a text, indicating the reason for the pre-compile error. Additionally, fields in the right panel of the Program tab that contain invalid values are marked yellow.

6.3 Data References

Data References are Placeholders for values that can be used in various places inside LinMot-Pilot. They can be inserted inside logical or numerical expressions for commands (e.g., *Calculate Value* in Pilot Commands) as a variable. When the expression is evaluated, the current value of the Data Reference is determined and used in the calculation. Data References can also be linked to in- or output variables (e.g., for the fieldbus Configuration in LinMot-Pilot or the encoder of the Conveyor Tracking).

Drop-Down Selection:

To select a Data Reference in LinMot-Pilot, the drop-down selection can be used. The drop-down extends when clicking on the input field which allows to select one of the data references. The available elements inside the drop-down selection are dependent of the configured references for the individual types and on the location of the Data Reference selection. E.g., when selecting a *Target Variable* for a *Calculate Value* command, the drop-down will only contain references that are allowed to be written. If more than four elements are available for a tier, a scrollbar is shown that allows selection of all available values. Once a valid reference is clicked in the extended drop-down, it will collapse back to show only the input field.

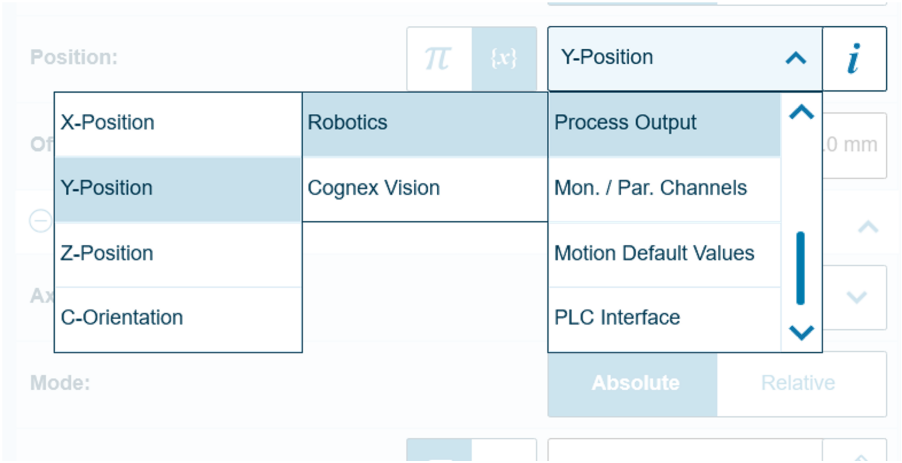


Fig. 30: The Data Reference drop-down.

If no Data Reference is selected, the drop-down will display "Please select a variable". Otherwise, the name of the selected Data Reference is displayed. Note that the name itself can be ambiguous. Clicking on the input field again will extend the selection and display the Data Reference type and categories, allowing to check precisely which reference is selected. Next to input field, a button allows to display the current value of the selected Data Reference when clicking and holding down on it.



The following sections explain more characteristics for die individual types of Data Reference.

Available Categories	Sub-categories of a Data Reference type are used to structure the selection. If a type has no sub-categories, the drop-down visualization will only show two tiers. Otherwise, three tiers are shown.
Available References	The number and meaning of the available references are highly individual for each Data Reference type.
Writing allowed	Not all Data References allow their value to be changed. If writing of a reference is not permitted, the Data Reference might not be shown in the selection drop-down or the corresponding action that tries to write the value might return an error.
Renaming allowed	Some Data Reference allow to be renamed. The name is only a visual aid to identify the Data Reference and swapping the names of two references will not swap the existing allocations. Drop-down visualizations and selections will immediately update to the name change. If the name of a Data Reference was written to a program row description, you will need to update the command in question manually to update the name.

6.3.1 User Variables

User Variables are accessible by each program and process inside LinMot-Pilot and are therefore the main tool to exchange data between several programs i.e., the main and its sub-programs. They can also be used to implement logic checks to control the structure and process of a program.

Available Categories	None
Available References	256 configurable references
Writing allowed	Yes, for all user variables. It is always the *Actual Value* that is written or read.
Renaming allowed	Yes, possible in *Setup* → *Applications* → *Variable Manager*. Variables without a name are not shown in the drop-down selection.

6.3.2 Dataset Variables

Dataset Variables can allow to configure and parametrize a program for multiple parts, products or processes.

Available Categories	None
Available References	256 configurable references[1]
Writing allowed	Yes, for all dataset variables.
Renaming allowed	Yes, possible in *Setup* → *Applications* → *Variable Managerl*. Variables without a name are not shown in the drop-down selection.

[1]Reference Values:

The value of the dataset variable is always dependent on the selected dataset. One of the 255 available datasets can be selected in the visualization page (*Setup* → *Applications* → *Variable Manager*), by command or by fieldbus. If a value is defined for the affected variable and in the selected dataset, this value is read by the Data Reference. If no dataset is selected (dataset ID 0) or no value is defined for the affected variable in the selected dataset, the default value is read. Writing of a dataset variable will always only change the *Actual Value* and never the *Default Value* or a value stored in one of the datasets.

6.3.3 Program Variables

Program Variables are inherently linked to a program and are saved within the program file. It is not possible for a program, to access the Program Variables of a different program.

Available Categories	None
Available References	32 configurable references
Writing allowed	Yes, for all program variables. It is always the *Actual Value* that is written or read.
Renaming allowed	Yes, possible in *Program* → *Program Variables*. Variables without a name are not shown in the drop-down selection.

6.3.4 Axis Parameters

Axis Parameters represent actual values of certain parameters of the connected axes.

Available Categories	Each axis is a separate category. The displayed axis name is based on the Kinematic Configuration.
Available References	For each axis: *Position*, *Velocity*, *Force*, *Standstill*[1]
Writing allowed	No
Renaming allowed	No

[1]Reference Values:

- *Position*: actual position of the axis in its units (mm or °)
- *Velocity*: actual velocity of the axis in its units (mm/s or °/s)
- *Force*: actual force of the axis in its units (N, Nm)
- *Standstill*: indicates if axis is still in motion (FALSE, = 0.0) or motion was finished (TRUE, = 1.0)

6.3.5 In- / Outputs

Analog and digital in- and outputs of connected drives and additional IO-modules connected to the EtherCAT bus are accessible as Data References of this type.

Available Categories	Each device (drives & additional modules) is a separate category. The displayed name is based on the Kinematic Configuration (drives) or the set name on the *Control* page (add. modules).
Available References	All in- / outputs that were configured for the device[1]
Writing allowed	Only for outputs
Renaming allowed	Yes, possible in *Control* → *Application Control* → *I/O Package*.

[1]Configuring of In- & Outputs:

- *Drives*: Enable and rename outputs in the Kinematic Setup (In- & Outputs), rename in- or outputs in system control (*Control* → *System Control* → *Inputs & Outputs*)
- *Additional IO-Devices*: Rename in- / outputs in the Application Control

6.3.6 Process Outputs

Process Outputs represent actual values of processes inside LinMot-Pilot.

Available Categories	Each process is a separate category. Only processes where a valid license is active will be listed.
Available References	Individual for each process (check the *Control* chapter of the Applications)
Writing allowed	No
Renaming allowed	No

6.3.7 Monitoring / Parameter Channels

Monitoring and parameter channels can be configured on the drives during the Kinematic Configuration. They allow reading and writing specific values from and to the connected drives.

Available Categories	Each drive is a separate category. The displayed drive name is based on the Kinematic Configuration.
Available References	Four configurable Monitoring and Parameter Channels each. Only configured channels are available in the reference selection.
Writing allowed	Only to Parameter Channels
Renaming allowed	No

6.3.8 Motion Defaults

Motion Defaults are mainly used to set default speeds of motion commands inside LinMot-Pilot. Since they are handled as Data References internally, they can be used in other places too.

Available Categories	*Linear Axes*, *Rotary Axes*, *Robotics*, and *Gripping*[1]
Available References	For each category: Values for *Velocity, Acceleration, Deceleration*, and *Jerk* for at three speeds (*Slow, Moderate, Fast*)[2].
Writing allowed	No

Available Categories	*Linear Axes*, *Rotary Axes*, *Robotics*, and *Gripping*[1]
Renaming allowed	No

[1]Use of Categories:

When using the *Presets* in the edit visualization of motion commands, the according category is used by default (e.g., *Gripping* for all commands in LinMot Grippers). The references can be changed to other categories manually but it is advised to adhere to the default categories.

[2]Reference Values:

The value of the Motion Defaults are given in the according physical units.
- *Linear Axes, Robotics,* and *Gripping* in m/s, m/s^2 and m/s^3
- *Rotary Axes* in °/s, °/s^2 and °/s^3

6.3.9 PLC Interface

The In- and Output Bits of the Fieldbus Connection are accessible as Data Reference.

Available Categories	*Input Bits*, *Output Bits*
Available References	64 bits each (see Configuration in LinMot-Pilot)
Writing allowed	Only for Output Bits
Renaming allowed	Yes, possible in *Setup* → *System Config* → *PLC Interface*. Bits without a name are not shown in the drop-down selection.

6.3.10 High Speed Variables

Two high speed variables can be configured on the drives. Currently, these variables must be configured manually as they are not integrated in the Kinematic Configuration yet. They allow reading specific values from the connected drives.

Available Categories	Each drive is a separate category. The displayed drive name is based on the Kinematic Configuration.
Available References	Two configurable High Speed Variables each. Only configured variables are available in the reference selection.
Writing allowed	No
Renaming allowed	No

6.4 Kinematics Wizard

The kinematics wizard provides the setup of the basic kinematic structure and additional axes of the system that is to be controlled by LinMot-Pilot. On first start-up or if a change in the connected axes is detected, the user is prompted to complete the installation wizard. Some configurable parameters of the kinematic configuration require the drives to be reinitialized or restarted. Therefore, changing the setup of the kinematic configuration can require a complete reboot of the system. The kinematic configuration can be changed by a user with administrator level and if no program is being executed by clicking the *Change Settings* button on any tab of the kinematic wizard in *Setup* → *System Config* → *Kinematics*.

6.4.1 Selection

The selection tab allows to select a kinematic configuration that represents the physical arrangement of the connected axes. The chosen configuration does not need to include all connected axes, as additional axes can

be configured in a later stage. If no kinematic configuration is suitable or desired, the option *Custom System* can be selected.

Note: Only axes that are configured as part of the kinematics (not as additional axis / Add-On) can be used in Robotic Motion and Conveyor Tracking commands.

The choice of kinematic configuration is therefore not only influenced by the physical setup, but also by the intended process. As axes that are part of the kinematics can not be moved in an independent process while a Robotic Motion or Conveyor Tracking command is active.

Example:

If a semi-gantry with a capper is used in an application with a conveyor belt, it is advised to only configure the X- and Y-axis of the semi-gantry as part of the kinematics. By doing so, the semi-gantry can synchronize to the conveyor belt with the x- and y-axis, while a capping process on the conveyor belt is performed by the z- and rotary axis. If any of the latter axes was configured as part of the kinematics, the capping process could not be started during tracking of the conveyor.

The right side of the page shows the selected kinematic configuration and indicates the number of detected drives and how many have been selected for the kinematic configuration below.



Fig. 31: The selection page shows four detected drives, two of which are used in the kinematic configuration and two more which can be declared as add-ons.

6.4.2 Add-Ons

If more drives are connected to the Technology Module than are required for the chosen kinematic configuration, the additional axes are declared as add-ons. Add-ons can involve a single or multiple axes and must be configured in a way that all connected drives are assigned either inside the kinematic configuration or as an add-on.

6.4.3 Assignment

The process axis of the previously defined kinematic configuration and add-ons have to be assigned to the correct drive in the EtherCAT configuration. The *Drive Nr.* represents the location of the drive in the EtherCAT chain, with drive 1 being the drive that is connected directly to ETH1 (Out1) of the Technology Module and the other drives daisy-chained with increasing numbers. The following parameters are shown or have to be configured for each drive:

Detected Axis

Current process axis of the connected drive. It is read from the stored name of the drive.

Configured Axis

Last configured process axis. Stored from the last time, the kinematic wizard was used or from a backup that was loaded.

Default Parameters

If checked, resets the drive to factory settings, before configuring the drive based on the kinematic configuration and axis setup. This can be forced, if a change in the connected axis was detected.

Note: Defaulting will reset all UPIDs on the drive to default values! Drive configurations that were done manually (e.g., PID-control parameters, sensor configurations, etc.) will be lost.

Newly Assigned Axis

Select the kinematic or additional axis that is connected to the drive with the according *Drive Nr.*.

Robotics Group

Indicates if an axis is part of the selected kinematic configuration. If no rotary axis is part of the kinematics yet, one additional rotary axis can be enabled as part of the robotics group.

Axis Name

Custom name that can be assigned to the axis. The *Axis Name* is used in axis selection drop-down menus, Data References, program command descriptions and as visualization label.

6.4.4 Directions & Limits

The homing procedure and position definitions for each axis are configured. The order of the axes is once again defined by the order of the drives on the EtherCAT bus. The following parameters are shown or have to be configured for each drive:

Axis Name

The axis named defined on the Assignment page.

Change Direction

The button can be toggled to inverse the positive direction of the axis. Move the axis by hand while observing the parameter *Current Position* and ensure that the positive direction is along the arrows in the image for the kinematic configuration or, in case of add-ons, according to your preferences.

Current Position

Show the actual position of the connected axis. This value is a reference only and the zero position and scale may be inaccurate until homing is complete.

Homing Order

The axes are homed in increasing numerical order. Multiple axes can perform the homing simultaneously, if they are all set to the same homing order.

Homing Procedure

Select one of the available homing procedures:

- **Actual Position**
  The current axis position at the time of homing is set to the configured *Home Position*.
- **Mechanical Stop Negative Search**
  Axis moves towards negative direction, until a mechanical stop is detected. Axis position is set to *Home Position* at mechanical stop.
- **Mechanical Stop Positive Search**
  Axis moves towards positive direction, until a mechanical stop is detected. Axis position is set to *Home Position* at mechanical stop.
- **No Drive Homing**
  The current position of the axis is not changed. If a *Home Position* is set, its value is added as an offset to the detected axis position. The offset is not persistent and will not add up after a restart of the system.
  E.g.: The *Home Position* of a rotary axis is set to 90°. At startup the axis is at 45°, after homing the axis position will be 135° without moving the axis. After rebooting and before homing, the axis position will be 45° once again.

Home Position

Reference position used during the homing procedure.

Initial Position

Axis moves to this position during the homing procedure. Has no effect if homing procedure *No Drive Homing* is selected.

Min. Position

Sets the lower axis position limit. Used for axis visualizations (e.g. in Axis Control) and to trigger automatic stopping of axes at the software limits.

Max. Position

Sets the upper axis position limit. Used for axis visualizations (e.g. in Axis Control) and to trigger automatic stopping of axes at the software limits.

Detected Stroke

Display the detected maximum stroke, which can be used to set the *Min.* and *Max. Position*. If the axis is powered of, move it manually both ways towards its limits and *Detected Stroke* will display the difference between the reached minimum and maximum positions. This value is a reference only and the zero position and scale may be inaccurate until homing is complete.



Note: The parameters *Current Position* and *Detected Stroke* hold invalid values if the drive was not defaulted before starting the kinematics wizard. To detect to maximum stroke or define the direction, default the drives prior to changing the kinematic configuration or finish the kinematic wizard and update the values on using the wizard a second time.

6.4.5 Conveyor

If a conveyor belt is used, its encoder can be configured here. The encoder can be connected to the sensor input (X13) of any drive in the system.

Check Connecting and Configuring a Conveyor for more details.

6.4.6 Channels

Up to four monitoring and parameter channels each can be used, to transfer additional information between drive and LinMot-Pilot. Monitoring and parameter channels are configured to the defined UPID, transmitting values to/from the drive by writing/reading the according monitoring channel as a data reference (see Monitoring / Parameter Channels).

The UPIDs can be entered directly in the *UPID* column or by searching for a UPID by name, using the *Caption* column.

6.4.7 In- & Outputs

The drives digital outputs can be controlled by LinMot-Pilot or used for other functionality, e.g. representing part of the status word. If a digital output is enabled on this page, by toggling the *Enable Output* button on, the output is configured to *Interface Output*. If enabled, the output can be controlled by the according data reference (see In- / Outputs) or the system control page.

6.4.8 Overview

The left side of the overview page shows the connected drives, the chosen kinematic configuration and the additional add-ons. The right side allows for powering on/off, jogging and homing of individual axes. The *Homing* button will execute the configured *Homing Procedure* from the Directions & Limits page for only the according

drive. The *Home All* button allows to home all axes according to the specified *Homing Procedures* and *Homing Order*.

6.5 User Management

User Management controls how users are created, assigned permissions, and maintained in the system. Access is based on four hierarchical user levels, which determine what actions a user can perform.

6.5.1 User Levels

- **Watch**
  View-only access. Users can monitor dashboards and system status but cannot perform any actions.

- **Operator**
  Can operate the system: start, pause, and stop programs; manually move axes; control existing widgets.
  Restrictions: Cannot access LinMot-Pilot programs or place new widgets.

- **Service**
  Can perform advanced technical tasks: configure settings, troubleshoot, manage widgets.
  Restrictions: Cannot use the Kinematic Wizard or manage users.

- **Administrator**
  Full system control: add, edit, and delete users; change user levels; configure global settings.
  Restrictions: Limited only by system state (e.g., certain actions blocked while running).

6.5.2 Initial Users

When the system is first started, two users are available by default:

- **admin**
  User Level: Administrator

- **viewer**
  User Level: Watch

These accounts ensure that the system can be accessed and operated immediately.

6.5.3 Login and Switch User

When the browser starts, the Login Page is displayed by default. To prevent unauthorized access, you must log in using a valid username and password combination.

Once logged in, you can switch to a different user account if needed. This can be done via the User Management dropdown in the Navigation Menu Bar, using the Switch User option.
Clicking Switch User opens the Switch User Dialog. In this dialog, enter a valid username and password for the new user. If you decide not to proceed, you can cancel the operation, and the current user will remain logged in.

6.5.4 Logout and Lock

Within the User Management dropdown, you will find two additional functions:

Logout:
The Logout function signs out the current user and redirects to the Login Page. This action restricts access to the LinMot-Pilot until a valid login is performed again.

Lock:

The Lock function works similarly to Logout, but instead of returning to the Login Page, it automatically logs in the predefined user viewer, who has the Watch user level. This mode allows limited access for monitoring purposes only.
In addition to manual locking, the system includes an Auto-Lock feature. This feature performs the same action as the Lock function after a period of inactivity:

- **Administrator:** Auto-lock after 1 minute
- **Service and Operator:** Auto-lock after 5 minutes

When auto-lock is triggered, the current user is logged out, and the viewer account is logged in automatically.

6.5.5 Change Password

On the User Settings page, each user can change their own password. To do this:

1. Click the Change Password button.
2. The Change Password Dialog will open. In the dialog, enter:
   - Current Password
   - New Password
   - Confirm New Password (re-enter the new password)
3. Click Change to confirm the update.

If a user forgets their password, it cannot be recovered. In this case, an Administrator must delete the user account and create a new one.

6.5.6 Add User

Administrators can view all existing users on the User Settings page. To add a new user:

1. Navigate to the User Settings page.
2. Click the Add User button located at the end of the user table. The Add User Dialog will open.
3. In the dialog, enter:
   - Username for the new user (Note: Once created, the username cannot be changed.)
   - Password
   - Confirm Password (re-enter the password)
   - User Level (select from available roles)
4. Click Add to create the new user.

6.5.7 Delete User

On the User Settings page, an Administrator can delete existing users. To delete a user:

1. Locate the user in the User Table.
2. In the corresponding row, click the Delete icon.
3. A confirmation popup will appear to verify the action.
4. Confirm the deletion to proceed.

Important: This action cannot be undone. Once a user is deleted, all associated credentials and settings are permanently removed.

6.5.8 Change Userlevel

On the User Settings page, an Administrator can modify the User Level of an existing user, thereby changing their permissions. To do this:

1. Navigate to the User Settings page.
2. In the User Table, locate the user whose level you want to change.
3. Click the Edit icon next to the current user level. The Change User Level Dialog will appear.
4. Select the new User Level from the available options.

5. Click Change to confirm the update.

6.6 IO Mapping

IO Mapping allows to map the status of LinMot-Pilot and the program execution to output signals and to control the buttons of the Title Bar by input signals.
The setup of the IO Mapping can be configured in the *Control Settings* of the Setup tab.

System Status Mapping

Up to 32 individual mappings of a system status to an output signal can be defined. Clicking on "+ add additional mapping" at the bottom of the table creates a new row to configure an additional mapping by selecting the *Mapped Status*, the *Output Signal* the status is written to and configuring the *Blink Signal* option. Clicking the *Edit* (✎) icon allows to change the *Mapped Status* and to delete the mapping by clicking the *Delete* (🗑) icon. The available statuses are listed below. Any output of an IO device can be selected as *Output Signal*. Note that the same output *can* be selected for multiple mappings, but this option is not supported. In that case, the lowest mapping in the table will overwrite the upper mappings to the same output.
If the *Blink Signal* option is selected, the output will be continuously switched on and off with the pulse duration defined in *Blinking Pulse Duration*, if the status condition is met. The mapping is active and will take effect as soon as the configuration is saved by clicking the *Save Changes* button.

Available Status	Condition
Ready	True, if the system is in the *Ready* System State
Running	True, if the system is in any of those System States: *Starting, Running, Resuming, Completing*
Paused	True, if the system is in any of those System States: *Paused, Pausing*
Complete	True, if the system is in the *Complete* System State
Uninitialized	True, if the system is in any of those System States: *Uninitialized, Aborting, Resetting*
Error	True, if the system is in any of those System States: *Error, Resolving, Halting*
Warning Active	True, if one or more warnings (event of Event Type *Warning*) are active
Pause Requested	True, if the *Request Pause* button is currently pressed
Completion Req.	True, if the *Req. Completion* button is currently pressed

Note: Multiple status mappings to the same output are not supported.

System Controls Mapping

Up to 32 individual mappings of a system control element to an input signal can be defined. Clicking on "+ add additional mapping" at the bottom of the table creates a new row to configure an additional mapping by selecting the *Mapped Control Button* and the *Input Signal* that is used to control the selected button. Clicking the *Edit* (✎) icon allows to change the *Mapped Control Button* and to delete the mapping by clicking the *Delete* (🗑) icon. The available control buttons are listed below. Any input of an IO device can be selected as *Input Signal*. For most buttons, multiple mappings are allowed with the exception of the *Auto / Manual* switch, where only the first mapping in the list will have an effect and following mappings will be ignored.
The *Enable IO Control* button allows to block system control from the mapped inputs signals completely. This allows to configure and test the physical control elements first, before enabling IO control. If IO control is enabled and a mapping to *Auto / Manual* button exists, the button is disabled in the visualization, as the input signal will overwrite any changes made in the visualization.

Available Control Button	Effect
Auto / Manual	sets *Manual* on logic True, *Auto* on logic False of input signal
Reset	activates button on rising edge of input signal

Resolve	activates button on rising edge of input signal
Start	activates button on rising edge of input signal
Pause	activates button on rising edge of input signal
Abort	activates button on rising edge of input signal
Step	activates button on rising edge of input signal
Request Pause	toggles button on rising edge of input signal
Req. Completion	toggles button on rising edge of input signal

Note: Multiple control mappings to the *Auto / Manual* switch are not supported.

6.7 Event Handling

Certain events lead LinMot-Pilot to produce a message that can be shown in the *Event Field* of the Title Bar, the table of the Events visualization page and are logged to two separate files on the Technology Module in the folder "/home/plc/applications/PlcLogic/log". The *Active* page of the Events visualization shows all events, whose cause has not been resolved yet, or that have not been acknowledged yet. As soon as an event is resolved and/or acknowledged (depending on the event), it will be moved to the *History* page that shows the last 200 events. Resolving is often possible by clicking the *Resolve* button. Otherwise, a suggestion to resolve the cause is given in the *Help* column of the *Active* page. The *Resolve* button will also acknowledge all active events.

6.7.1 Event Types

Three different types of events are available in LinMot-Pilot. They distinguish themselves regarding their effect on the System States and the way they are shown in the visualization.

Information:

Events of type information are only active for a single PLC cycle and are thus not visible in the *Active* page of the Events visualization. Informations are mainly used to log important process information to the log files.

Warning:

Events of type warning are displayed with yellow background in the events list and the *Event Field*. Warnings can require resolving, acknowledging or both.
If warnings require resolving, they are shown as long as the the cause for the warning is still active.

- Example: If an axis is disabled in *Auto* mode, the system warning "SYS-53" is displayed. As soon as the axis is enabled again or the system is switched to *Manual* mode, the warning will disappear.

If warnings require acknowledging, they are shown until they are acknowledged using the *Resolve* button.

- Example: If an axis with an error is disabled, and the error can not be cleared on *Resolve*, a system warning "SYS-33" is displayed. As soon as *Resolve* is pressed again, the warning will disappear.

Warnings that are raised during the execution of a program will have no effect on the System States.

Error:

Events of type error are displayed with red background in the events list and the *Event Field*. Errors always require acknowledging and can optionally also require resolving.
If an error occurs during the execution of a program, the System States will change to halt the program and move to the error state. Acknowledging (and resolving) the error will change the system to the paused or uninitialized state, depending on if the specific error allows resuming the program.

6.7.2 Event Sources

Events in LinMot-Pilot cab originate from different processes, application and sub-systems, that are called *Sources*. The event source is always indicated as a prefix in the event ID.

Drive:

Drive events show errors that occurred on the drives.

Prefix in Event-ID	DRV<n>, with <n> representing the number of the affected drive
Number in Event-ID	Represents error ID of according drive in hexadecimal notation. Check LinMot-Talk documentation for more details.
Example: "DRV2-B"	Drive 2 has an error with ID Bh (11 in decimal), which represents the error "Err: Pos Lag Always Too Big".

Axis:

Axis events show errors that set the according axis to error state.

Prefix in Event-ID	AX<n>, with <n> representing the number of the affected axis
Number in Event-ID	Represents the error ID according to the Codesys documentation.
Example: "AX1-11"	Axis 1 has an error with ID 11, which represents the error "Hardware end switch is active".

System:

Most system events occur during the startup sequence of the system.

Prefix in Event-ID	SYS
Number in Event-ID	Represents the Event ID according to the List of System Events in the appendix.
Example: "SYS-53"	One or more axes are not enabled.

User:

An event from the program command User Message occurred.

Prefix in Event-ID	USR
Number in Event-ID	1: Information, 2: Warning, 3: Error with Pause, 4: Error with Stop
Example: "USR-2"	User message of type *Warning* was perfomred.

Program:

An event occurred during execution of a program.

Prefix in Event-ID	PRG
Number in Event-ID	Represents the Event ID according to the List of Program Events in the appendix.
Example: "PRG-4"	Program finished.

Axes Motion:

An error occurred in a process of the axes motion application.

Prefix in Event-ID	AXM
Number in Event-ID	Represents the Event ID according to the List of Axes Motion Events in the appendix.
Example: "AXM-9001"	Axes were paused during relative motion

Robotics:

An error occurred in a process of the robotics application.

Prefix in Event-ID	ROB
Number in Event-ID	Represents the Event ID according to the List of Robotics Events in the appendix.
Example: "ROB-8002"	Application input invalid.

Gripping:

An error occurred in a process of the gripping application.

Prefix in Event-ID	GRP
Number in Event-ID	Represents the Event ID according to the List of Gripping Events in the appendix.
Example: "GRP-9001"	Motion is not possible for 1 DOF grippers.

Conveyor:

An error occurred in a process of the conveyor tracking application.

Prefix in Event-ID	CVY
Number in Event-ID	Represents the Event ID according to the List of Robotics Events in the appendix.
Example: "CVY-9001"	Relative movement between coordinate systems not possible.

Cognex Vision:

An error occurred in a process of the cognex vision application.

Prefix in Event-ID	CGX
Number in Event-ID	Represents the Event ID according to the List of Cognex Events in the appendix.
Example: "CGX-9011"	No device selected

LinMot Vision:

An error occurred in a process of the LinMot vision application.

Prefix in Event-ID	LMV
Number in Event-ID	Represents the Event ID according to the List of LinMot Vision Events in the appendix.
Example: "LMV-9003"	TCP communication timeout

6.7.3 Log Files

All events are logged in two separate files per calendar week. The files contain the same information but formatted differently. They are stored on the Technology Module at */home/plc/applications/PlcLogic/log* as "*Log_Pilot_{YYYY}_CW{WW}.txt*" and "*PilotEvents_{YYYY}_CW{WW}.csv*". All files older than two years are deleted automatically.

7 Fieldbus Connection

A system that is controlled by LinMot-Pilot can be integrated into an existing infrastructure by using the fieldbus connection to control LinMot-Pilot and thereby the operation of the Technology Module. The respective Interface Application license is required to use the PLC interface. See Licenses and Info for more information.

7.1 Functionality and Limitations

Control of certain functionalities of LinMot-Pilot is possible if a fieldbus connection is active and the PLC Interface is enabled. However, the fieldbus connection does not allow for complete control and especially setup of LinMot-Pilot thus the latter always has to be configured appropriately before being able to make use of the fieldbus connection. The following table provides a rough breakdown over which functions are available using the fieldbus connection.

Possible using Fieldbus Connection	Impossible using Fieldbus Connection
• System Control: Powering all axes on / off, Homing all axes, resetting system • Setting the global speed override value • Switching to a selected dataset • Loading one of eight predefined programs • Program Control: Starting, pausing, resuming and aborting the loaded program. Setting pause and completion requests. • Resolving errors • Reading and writing of configured data references and PLC-bits	• Manual axis or application control • Changing the Kinematics or any other system or application settings • Setup and control of the general visualization (e.g. adjusting and interacting w/ Widgets) • Editing or saving programs • Stepping the program or starting a program from a row different than from the first row. • Renaming or forcing of data references and PLC-bits

A more detailed description of the functionality is given in Configuration in LinMot-Pilot.

7.2 Setup

Profinet and EtherNet/IP are supported as fieldbuses. The X4 port of the Technology Module is used for the fieldbus connection. See Examples for fieldbus-specific setup instructions.

7.2.1 Configuration in LinMot-Pilot

The configuration of the PLC-Interface is possible on the according setup page of the visualization. Navigate to *Setup* → *System Config* → *PLC Interface* where configuration, manual overrides and monitoring of the fieldbus connection is handled.

General Setup - Communication:

If one of the supported fieldbuses was connected as described in Connections, LinMot-Pilot will automatically detect and display it. The green pill labeled with the connected fieldbus will be displayed and the gray pill labeled *Offline* will disappear.
Even with an active connection, LinMot-Pilot will not write any output values, the mapping is inactive and the inputs won't have any effect until the PLC-Control is enabled by toggling the *Enable PLC Control:* button and the system is set to *Auto* in the Title Bar. If control is enabled, a confirmation dialog will appear upon saving the changes. Be aware that the PLC will take over control of the system once the dialog is confirmed and the controls from the fieldbus will be interpreted immediately after. This can potentially lead to undesired behavior, if the PLC is not in a suitable state. If the *Enable PLC Control* button is disabled or the system is set to *Manual* mode, the gray *No Control* pill is shown. Else the pill will turn green if a fieldbus is connected, control is enabled and the system is set to *Auto*.
The *Unforce all Inputs* button allows to unforce all input values immediately.
Up to eight programs can be configured as *PLC Program 1:* to *PLC Program 8:*. Clicking on the input field will open a file dialog in which a program file can be selected. The program will not be loaded yet, but the according command from the fieldbus will cause the LinMot-Pilot to open or switch to the configured program. Clicking the trash icon will clear the configured program. If opening a PLC Program is requested but no program is configured, the loading will fail and an error will occur.



Note: Be aware that the PLC will take over control of the system once the PLC Control is enabled and the controls from the fieldbus will be interpreted immediately after. This can potentially lead to undesired behavior, if the PLC is not in a suitable state.

Inputs & Outputs:

The naming convention for in- and outputs is based on the perspective of the Technology Module. *Inputs* are defined as data being sent from the fieldbus to the Technology Module, whereas *Outputs* are data sent from the Technology Module to the fieldbus. Inputs will always be received by the Technology Module but they will have no effect unless the PLC Control is enabled. Therefore, the value of Input Variables and Bits will not be written to the configured Data References and Input Parameters will not take effect on the system. Outputs will always be read and transmitted, regardless of whether PLC Control is active.

In- & Output Variables:

A set of 56 variables (32-bit) is transmitted both as in- and outputs on the fieldbus, which are either available as Data References within LinMot-Pilot or can be mapped to other Data References by selection in the drop down menu. Variables can be individually renamed on the Setup page. The custom name of the bits is displayed in the Data References drop-down and in descriptions. The values are handled as floating point data types within LinMot-Pilot and are multiplied by 100000.0 when converted to the fieldbus and sent as a DWORD of 32-bit. This is explained in more detail in Data Structure.
Only Data References that can be written are allowed to be mapped to PLC Input Variables. For Output Variables, any Data Reference is valid. The column labeled *Value on Bus* shows the currently transmitted value, even if PLC Control is not enabled. The value of any of the Input Variables can be set manually by using the toggle button in the *Force* column. If the button is toggled, the variable will be set to the value configured in the *Actual Value* column. If an Input Variable is mapped and forced, the mapped Data Reference will be set to the specified force value. Note that any changes to the settings (i.e. switching of the Data Reference) take effect only after saving the changes, using the *Save Changes* button at the bottom of the page.

In- & Output Bits:

In addition to the variables, 128 bits for both in- and outputs are transmitted via fieldbus. Bits have the same configuration options as the Variables and can be mapped, renamed and forced (Input Bits only). Note that any changes to the settings (e.g. renaming bits, changing the force values) take effect only after saving the changes, using the *Save Changes* button at the bottom of the page.

In- & Output Parameters:

The final section of the setup page shows the control and monitoring signals. These values are updated each cycle and show the actual values on the fieldbus when the PLC control is enabled. If it is disabled, the *Output Parameters* show the values that would be written to the fieldbus if the PLC control was enabled.

7.3 Data Structure

This section explains the data structure of the in- and output data that is transmitted cyclically via the fieldbus. Control and monitoring signals and variable and bit values are structured into packets of multiples of 4 bytes (32-bit DWORDs) which have to be setup accordingly on the PLC.

7.3.1 Inputs

The inputs (i.e. data transmitted from the PLC to the Technology Module) contain a total of 144 bytes, structured as follows:

Fieldbus Inputs (PLC → TM):

Description	Data Type	Size
Speed Override	UDInt	4 bytes
Input Bits	128 x BOOL	16 bytes
Input Variables	56 x DInt	224 bytes
Control Word	16 bits + 2 x USInt	4 bytes
	Total:	248 bytes

As the *Speed Override* and the *Input Variables* are handled as floating point values in LinMot-Pilot, the values are converted to UDInts by multiplication by 100000. A *Speed Override* of 80% for example, will thus require an input of value 8000000.
The *Control Word* is a structure of its own, which is described in more detail below.

Control Word:

The 16 bits of the Control Word are checked for rising and falling edge and will cause effects according to the information in the table below.

Description	Data Type	Size (Bits)	Effect on Rising Edge	Effect on Falling Edge
Power System	BIT	1	Power all axes on	Power all axes off
Home System	BIT	1	Home all axes	-
Reset	BIT	1	Set *Reset* Button TRUE	Set *Reset* Button FALSE
Resolve	BIT	1	Set *Resolve* Button TRUE	-
Run Program	BIT	1	Start Shown Program	-
Pause Program	BIT	1	Pause Running Program (if currently running)	-
Resume Program	BIT	1	Resume Paused Program (if currently paused)	-
Abort	BIT	1	Abort Running Program	
Request Pause	BIT	1	Set *Request Pause* Button TRUE	Set *Request Pause* Button FALSE
Request Completion	BIT	1	Set *Request Completion* Button TRUE	Set *Request Completion* Button FALSE
Load Program	BIT	1	Load the selected Program	-
Switch Dataset	BIT	1	Switch to selected Dataset	-
b12	*BIT*	*1*	-	-
b13	*BIT*	*1*	-	-
b14	*BIT*	*1*	-	-
b15	*BIT*	*1*	-	-
Selected Program	USInt	8	-	-
Selected Dataset	USInt	8	-	-

7.3.2 Outputs

The outputs (i.e. data transmitted from the Technology Module to the PLC) contain a total of 152 bytes, structured as follows:

Fieldbus Outputs (TM → PLC):

Description	Data Type	Size
Speed Override	UDInt	4 bytes
Output States	2 x UInt	4 bytes
Error ID	1 x UInt + 2 x USInt	4 bytes
Output Bits	128 x BOOL	16 bytes
Output Variables	56 x DInt	224 bytes
Status Word	16 bits + 2 x USInt	4 bytes
	Total:	256 bytes

As the *Speed Override* and the *Output Variables* are handled as floating point values in LinMot-Pilot, the values are converted to UDInts by multiplication by 100000. A *Speed Override* of 80% for example, will thus result in an output of value 8000000.

The *Output States*, *Error ID* and *Status Word* are structures themselves, which are described in more detail below.

Output States:

Description	Data Type	Size (Bits)	Value / Mapping
System State[1]	UInt	8	Mapped to Main State of LinMot-Pilot
Safety State	*UInt*	*8*	*not implemented yet*

[1]See the System States for more detail. The value of *System State* is equal to the numerical value of the current state.

Error ID:

Description	Data Type	Size (Bits)	Value / Mapping
Error ID	UInt	16	ID of the most recent, active Event
Error Source	USInt	8	Source of the most recent, active Event (see Event Handling)
reserved	*USInt*	*8*	*reserved*

Status Word:

Description	Data Type	Size (Bits)	TRUE condition	FALSE condition
System Powered	BIT	1	All axes are powered on (enabled)	One or more axes are powered off (disabled)
System Homed	BIT	1	All axes are homed	One or more axes are not homed
Error	BIT	1	One or more errors are active	No error is active
Pause Requested	BIT	1	*Request Pause* is active (pressed)	*Request Pause* is inactive (not pressed)
Completion Requested	BIT	1	*Request Completion* is active (pressed)	*Request Completion* is inactive (not pressed)
b05	*BIT*	*1*	-	-
b06	*BIT*	*1*	-	-
b07	*BIT*	*1*	-	-
b08	*BIT*	*1*	-	-
b09	*BIT*	*1*	-	-
b10	*BIT*	*1*	-	-
b11	*BIT*	*1*	-	-
b12	*BIT*	*1*	-	-
b13	*BIT*	*1*	-	-
b14	*BIT*	*1*	-	-
b15	*BIT*	*1*	-	-
Selected Program	USInt	8	-	-
Selected Dataset	USInt	8	-	-

7.4 Examples

Described below are the examples provided to use the fieldbus connection of LinMot-Pilot. Note that these are not complete documentations of how to use the example but only outline the most significant parts.

7.4.1 Profinet with Siemens PLC

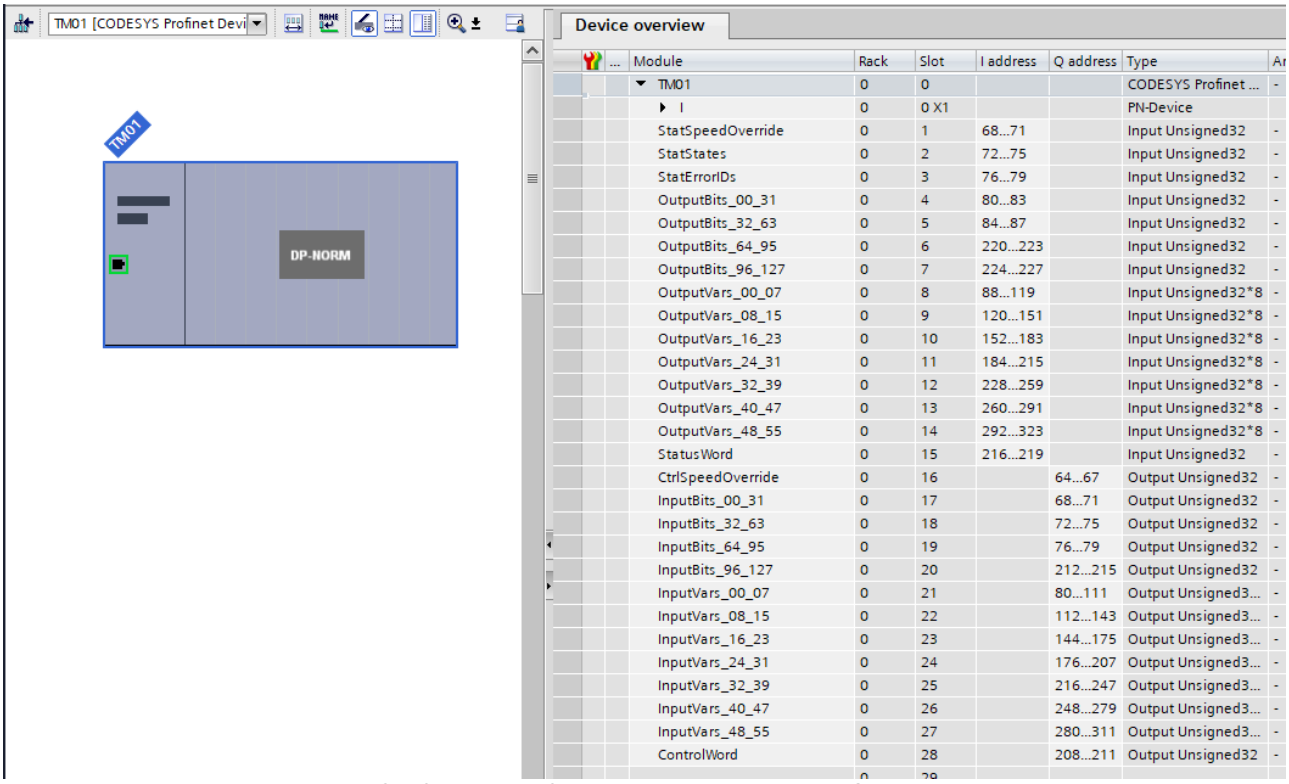
This example was created using a SIMATIC S7-1200 and Siemens TIA V19.

Setup:

Connect the Profinet network to the X4 port of the Technology Module.

Configuration:

The data modules for the Technology Module are configured according to the described Data Structure. The order and size of the modules is critical for correct transmission.



Module	Rack	Slot	I address	Q address	Type
TM01	0	0			CODESYS Profinet ...
I	0	0 X1			PN-Device
StatSpeedOverride	0	1	68...71		Input Unsigned32
StatStates	0	2	72...75		Input Unsigned32
StatErrorIDs	0	3	76...79		Input Unsigned32
OutputBits_00_31	0	4	80...83		Input Unsigned32
OutputBits_32_63	0	5	84...87		Input Unsigned32
OutputBits_64_95	0	6	220...223		Input Unsigned32
OutputBits_96_127	0	7	224...227		Input Unsigned32
OutputVars_00_07	0	8	88...119		Input Unsigned32*8
OutputVars_08_15	0	9	120...151		Input Unsigned32*8
OutputVars_16_23	0	10	152...183		Input Unsigned32*8
OutputVars_24_31	0	11	184...215		Input Unsigned32*8
OutputVars_32_39	0	12	228...259		Input Unsigned32*8
OutputVars_40_47	0	13	260...291		Input Unsigned32*8
OutputVars_48_55	0	14	292...323		Input Unsigned32*8
StatusWord	0	15	216...219		Input Unsigned32
CtrlSpeedOverride	0	16		64...67	Output Unsigned32
InputBits_00_31	0	17		68...71	Output Unsigned32
InputBits_32_63	0	18		72...75	Output Unsigned32
InputBits_64_95	0	19		76...79	Output Unsigned32
InputBits_96_127	0	20		212...215	Output Unsigned32
InputVars_00_07	0	21		80...111	Output Unsigned3...
InputVars_08_15	0	22		112...143	Output Unsigned3...
InputVars_16_23	0	23		144...175	Output Unsigned3...
InputVars_24_31	0	24		176...207	Output Unsigned3...
InputVars_32_39	0	25		216...247	Output Unsigned3...
InputVars_40_47	0	26		248...279	Output Unsigned3...
InputVars_48_55	0	27		280...311	Output Unsigned3...
ControlWord	0	28		208...211	Output Unsigned32

Fig. 32: Configuration of the Technology Module in Siemens TIA

A tag table assigns the addresses of the in- and output data to tags which can be used for further assignments. Each tag covers 32 bits of data and can be split up further, depending on the type of data it represents.

Structures and Data Types:

Structures for in- and output data are create to simplify access to the individual values such as specific bits of the status or control word.
One in- and one output variable of their corresponding data types are defined in a global variable list (GVL).

Converting In- and Outputs:

Functions for reading and writing from and to the Profinet bus are provided. These functions access fieldbus data based on the global tag table and link it to the in- and output variables defined in the GVL. The variables in the GVL can then be used to in other functions or visualizations to control and monitor the Technology Module running LinMot-Pilot.

7.4.2 EtherNet/IP with Rockwell Automation PLC

This example was created using Logix Designer v36, and applies to CompactLogix and ControlLogix controllers from Logix Designer v34 and later.

Setup:

Connect the EtherNet/IP network to the X4 port of the Technology Module. Navigate to *Setup* → *System Config* → *PLC Interface* and configure the IP address and subnet mask of the EtherNet/IP Adapter.

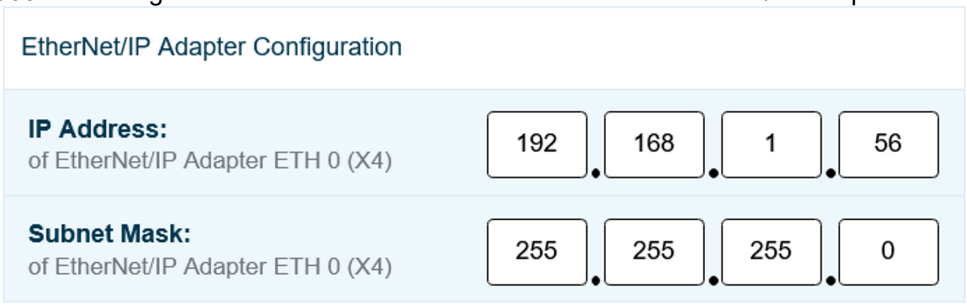


Fig. 33: Configuration of the EtherNet/IP Adapter in LinMot-Pilot

Configuration:

Use the Device Description File Installation Tool, installed with Studio 5000 Logix Designer, to register the EDS file for LinMot-Pilot. The EDS file is available on the LinMot website as a software download to the Technology Module.
Add the LinMot-Pilot (Technology Module) as a device to the I/O Configuration. This will add the I/O tags to the Controller Tags.
Earlier versions than v34 of Logix Designer can use the EDS file, but the data type will be SINT arrays instead of tag structures with names.
The tag structure is as defined in Data Structure.

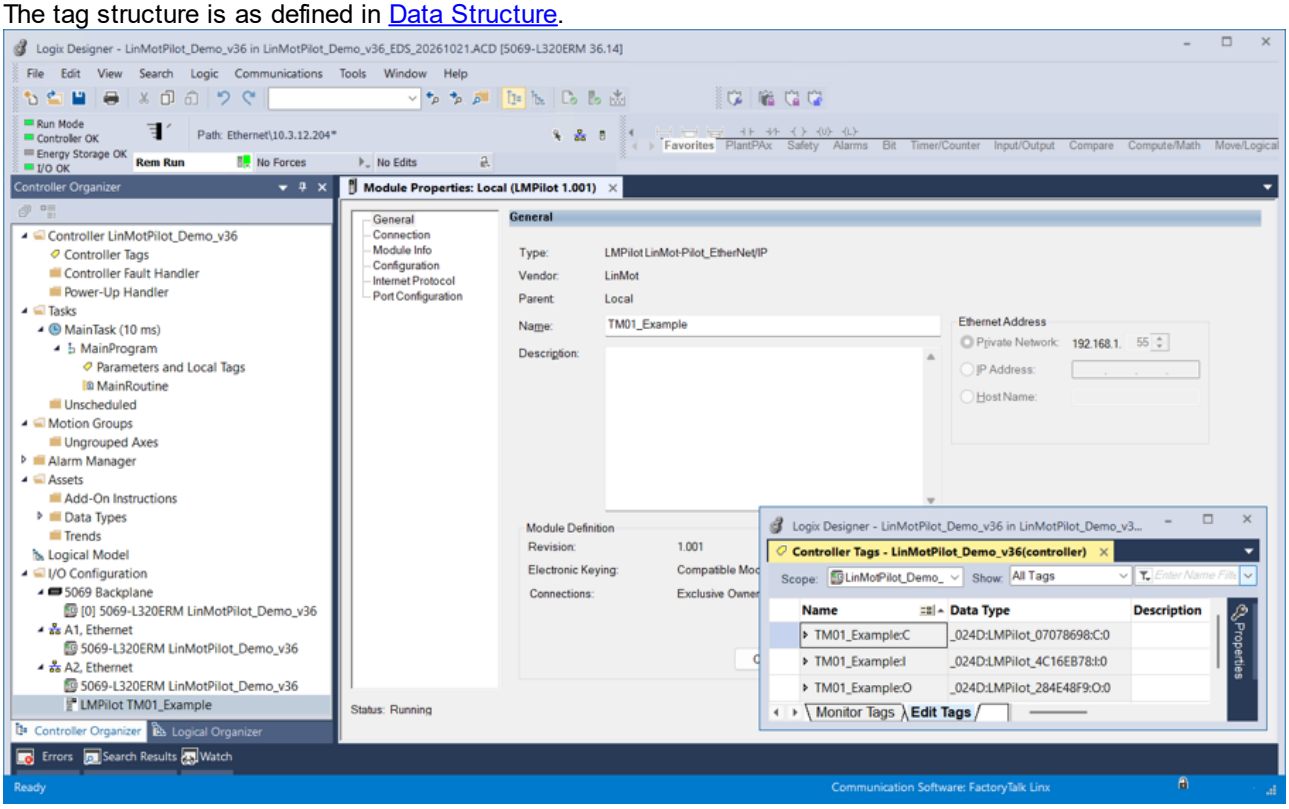


Fig. 34: Configuration of the Technology Module in Logix Designer

8 Ethernet over EtherCAT

As an alternative to the USB-to-RS232 converter, Ethernet over EtherCAT (EoE) can be used to configure LinMot Drives with LinMot-Talk.

The EtherCAT-Drives have to be connected to the EtherCAT Extender Module ETH1 (Out1) on the TM01 as stated in the Quick Start Guide. The TM01 has to be powered and the Process Monitoring Application should be running.

To access the drives from a computer (directly connected to X11 or X12), a custom IPv4-route has to be configured.
Open a command prompt with administrator rights and execute the following command:

```
route add 192.168.42.0 MASK 255.255.255.0 169.254.255.XX
```

(Replace "169.254.255.XX" with the default IPv4-address printed on the label)

Now you can log in to the drives via the Ethernet interface on LinMot-Talk.
Each drive is assigned a unique IPv4-address within the range of 192.168.42.XX:

Drive Number	IPv4 Address
Drive 1	192.168.42.11
Drive 2	192.168.42.12
Drive 3	192.168.42.13
...	...

The custom IPv4-route can be deleted with the following command (or a reboot of the computer):

```
route delete 192.168.42.0 MASK 255.255.255.0 169.254.255.XX
```

9 Troubleshooting

9.1 Web Visu Issues

Issue 1: Text Display Errors in the Web Visualization

Symptom: Text in the WebVisu appears “cut off” or incomplete.
Cause: In some browsers, enabled hardware acceleration can cause rendering issues.
Solution Steps:
1. Open your browser Settings.
2. Navigate to System or Advanced Settings.
3. Disable the option Use hardware acceleration.
4. Restart the browser to apply the changes.

Note: Disabling hardware acceleration may slightly reduce performance in graphics-heavy applications, but text rendering will be correct.

Issue 2: Incorrectly Scaled Window in the Web Visualization
Symptom: The WebVisu window appears too small, too large, or blurry in the browser.
Cause: Incorrect combination of browser zoom and monitor scaling.
Solution Steps:
1. Ensure that the combined scaling of your browser and monitor equals 100%:
2. Browser Zoom×Monitor Zoom=100%\text{Browser Zoom} \times \text{Monitor Zoom} = 100\%
   Browser Zoom×Monitor Zoom=100%
   Example: Monitor scaling 125% → Browser zoom 80%.
3. Open the browser management menu with Ctrl + Shift + Delete.
4. Clear Cookies and Site Settings.
5. Reload the WebVisu page.

→ The page should now display correctly scaled and sharp.

Note: After clearing cookies, you may need to re-enter saved login credentials.

10 Licenses

The following licenses are available for LinMot-Pilot. License-specific program commands and pages in the visualization are only available if a valid license is active (purchased or test license). A description about how to activate additional licenses can be found in the Installation Guide for the Technology Module in chapter 6.3.

Product Code (License)	Content
0187-1000 (OS-LMP01)	LinMot-Pilot Operating System
0187-1001 (TL-AXM01)	Axes Motion for LinMot-Pilot (Included in OS-LMP01, not sold separately)
0187-1002 (TL-ROB01)	Robotic Motion for LinMot-Pilot (Included in OS-LMP01, not sold separately)
0187-1003 (TL-GRP01)	LinMot Grippers for LinMot-Pilot (Included in OS-LMP01, not sold separately)
0187-1004 (TL-CNV01)	Conveyor Tracking for LinMot-Pilot
0187-1005 (TL-VIS_COG01)	Cognex Interface for LinMot-Pilot
0187-1006 (TL-VIS_LMV01)	LinMot Vision for LinMot-Pilot
0187-1007 (TL-CAP01)	Capping for LinMot-Pilot
0187-1008 (TL-REC01)	Data Recording for LinMot-Pilot
0187-1009 (TL-MON01)	Process Monitoring for LinMot-Pilot (2 channels per license, requires TL-REC01)
0187-1010 (TL-PRN01)	Profinet Interface for LinMot-Pilot
0187-1011 (TL-EIP01)	Ethernet/IP Interface for LinMot-Pilot
0187-1012 (TL-OPC01)	OPC UA Interface for LinMot-Pilot

11 Appendix

11.1 Events

11.1.1 List of System Events

ID	Event Type	Description
1	Error	Hardware check failed
2	Error	Reading current modified UPID list failed
3	Error	Defaulting drive parameters failed
4	Error	Reading PNP values failed
5	Error	Stopping Motion Control and Application SW failed
6	Error	Restoring modified UPID list failed
7	Error	Setting drive names failed
8	Error	Setting home type failed
9	Error	Setting initial position failed
10	Error	Setting moving direction failed
11	Error	Reading and saving modified UPID list failed
12	Error	Restarting drives failed
13	Error	Starting Motion Control and Application SW failed
14	Error	Setting integrator limit failed
15	Error	Setting axis group failed
16	Error	Setting motor type failed
17	Error	Setting position limits failed
18	Error	Reading hardware information failed
19	Error	Writing UPID failed
20	Error	Reading UPID failed
21	Error	Setting monitoring and parameter channels failed
22	Error	Setting interface output failed
23	Error	Setting robotic speed limits failed
24	Error	AutoRun program could not be started
25	Error	Loading resetting program failed
26	Error	Saving resetting program failed
27	Error	Halting of axes timed out
28	Error	Resolving timed out
29	Error	Aborting timed out
30	Warning	Reading actual information failed
31	Warning	Reading actual information timed out
32	Error	Reading parameter channel defaults failed
33	Warning	Axis with error was disabled during resolve

34	Error	Homing of an axis failed
35	Error	Drive Detection failed, invalid configuration
36	Error	An axis in force control had an error and had to be reset to switch to position control
37	Error	Waiting for the Motion Control and Application Software startup failed
38	Error	Setting of conveyor sensor type failed
39	Error	Setting the dynamic limits failed
40	Error	Setting the encoder count direction failed
41	Error	Setting the force/torque control type failed
42	Error	Setting the parameter force mask failed
43	Error	Setting the parameter force value failed
50	Error	One or more axes are not homed
51	Error	One or more axes are not enabled
52	Warning	One or more axes are not homed
53	Warning	One or more axes are not enabled
61-68	Warning	STO of drive is pressed
100	Information	EtherCAT master started
101	Information	Startup sequence finished
102	Information	Axis group was reset

11.1.2 List of Program Events

ID	Event Type	Description
1	Information	Program was started
2	Information	Program was paused
3	Information	Program was stopped
4	Information	Program finished
5	Error	Command is unknown / does not exist
6	Error	No license for command available
7	Error	Axes are not configured for command
8	Error	Pointer is uninitialized
9	Error	Axes are not enabled for command
10	Warning	Program was paused during relative motion
11	Warning	Program was paused during trajectory
12	Warning	Program was paused during velocity control
13	Warning	Timeout occurred
14	Error	Library Paused
15	Warning	Program was not running and could not be stopped
16	Error	A timeout occurred and execution was halted
20	Error	Label could not be found
21	Error	Reading of Data Reference failed

22	Error	Writing of Data Reference failed
23	Error	Computation of expression failed
24	Error	Selected Dataset is invalid
25	Error	Writing of UPID failed
26	Error	Reading of UPID failed
27	Error	Grip is not active
28	Error	Selected gripper does not exist, command could not be executed
29	Error	Too many output pulses running
30	Error	Selected capper does not exist, command could not be executed
31	Error	Capping command was aborted
32	Error	Capping command timed out
33	Error	Position-Check failed in capping command
34	Error	Angle-Check failed in capping command
35	Error	Force-Check failed in capping command
36	Error	Torque-Check failed in capping command
37	Error	Linear Standstill failed in capping command
38	Error	Rotary Standstill failed in capping command
40	Error	Selected conveyor does not exist, command could not be executed
41	Error	Axis Group is not synchronized with selected conveyor
42	Error	Axis Group is already synchronized with a conveyor
43	Error	No target to track on conveyor
44	Error	Target is not valid
45	Error	Selected conveyor is not valid
50	Error	Position above monitoring window in gripping command
51	Error	Position below monitoring window in gripping command
52	Error	Force above monitoring window in gripping command
53	Error	Force below monitoring window in gripping command
100	Error	Reading of program failed
101	Error	Writing of program failed
102	Error	Program buffer is full
103	Error	Max. number of running program groups reached
104	Error	

11.1.3 List of generalized Application Events

ID	Event Type	Description
8001h	Error	Application input ignored
8002h	Error	Application input invalid
8003h	Error	Application timed out

8004h	Error	Resuming in given state is invalid
8101h	Error	Axis is not powered
8102h	Error	Axis is not in standstill

11.1.4 List of Axes Motion Events

Events with IDs below 8000h are defined according to the Codesys documentation.
Events with IDs between 8000h and 9000h are defined according to the List of generalized Application Events.

ID	Event Type	Description
9001h	Warning	Axes were paused during relative motion
9002h	Error	Motion buffer is full

11.1.5 List of Robotics Events

Events with IDs below 8000h are defined according to the Codesys documentation.
Events with IDs between 8000h and 9000h are defined according to the List of generalized Application Events.

ID	Event Type	Description
9001h	Error	Relative movement between coordinate systems not possible

11.1.6 List of Conveyor Events

Events with IDs below 8000h are defined according to the Codesys documentation.
Events with IDs between 8000h and 9000h are defined according to the List of generalized Application Events.

ID	Event Type	Description
9001h	Error	The chosen target does not exist
9002h	Error	Selected conveyor is not valid
9003h	Error	Target buffer is full
9004h	Error	Deleting active target while tracking is not possible
9005h	Error	Changing active target while tracking is not possible
9006h	Error	Trigger distance is not valid
9007h	Error	The active target, which is currently being tracked, passed the latest desync position and was removed from the buffer

11.1.7 List of Gripping Events

Events with IDs below 8000h are defined according to the Codesys documentation.
Events with IDs between 8000h and 9000h are defined according to the List of generalized Application Events.

ID	Event Type	Description
9000h	Error	No Grip is active
9001h	Error	Motion is not possible for 1 DOF grippers
9002h	Error	Force control not possible in parallel mode
9003h	Error	Reference for pre-position invalid
9100h	Error	Force limit reached during motion to pre-position
9101h	Error	Changing the operation limit failed. Axis is in wrong state.

11.1.8 List of Capping Events

Events with IDs below 8000h are defined according to the Codesys documentation.
Events with IDs between 8000h and 9000h are defined according to the List of generalized Application Events.

ID	Event Type	Description
9000h	Error	Seating Point was not found
9100h	Error	Changing the operation mode failed. Axis in wrong state.

11.1.9 List of Cognex Events

Events with IDs below 8000h are defined according to the Codesys documentation.
Events with IDs between 8000h and 9000h are defined according to the List of generalized Application Events.

ID	Event Type	Description
9000h	Error	Unknown command
9001h	Error	Invalid tag or number
9002h	Error	Execution Failed
9003h	Error	Login failed
9006h	Error	User has no access
9010h	Error	Invalid mode
9011h	Error	No device selected
9012h	Error	Command allows only one device
9013h	Error	Index out of bounds
900Fh	Error	Telnet connection failed

11.1.10 List of LinMot Vision Events

Events with IDs below 8000h are defined according to the Codesys documentation.
Events with IDs between 8000h and 9000h are defined according to the List of generalized Application Events.

ID	Event Type	Description
9001h	Error	Object detection failed
9002h	Error	Reading of parameters failed
9003h	Error	TCP communication timeout
9004h	Error	Unknown object type

11.2 Installation without TM-Manager

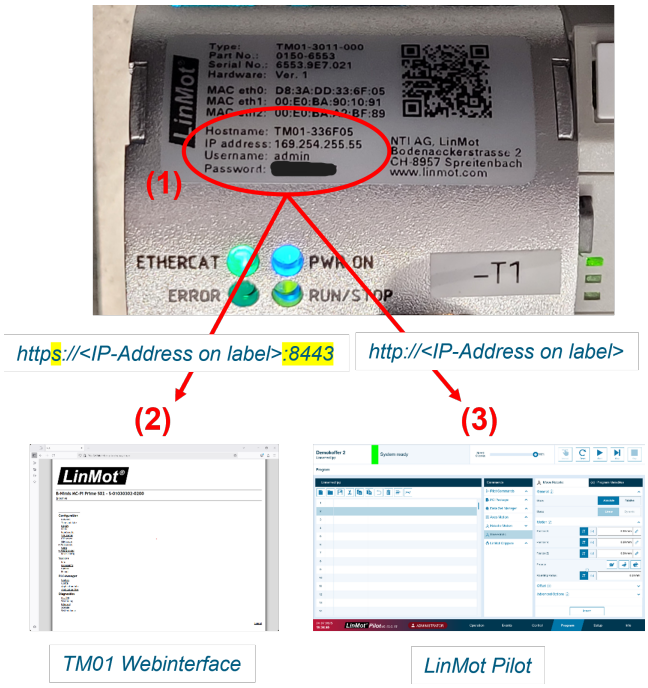
First Access and Login:

- Connect PC to ETH2 (X11/X12) directly
- Open the TM Webinterface using a web browser (`https://<IP-Address on label>:8443`). The default IPv4 address is printed on the label.
- Login using the username and password printed on the label. Tip: Change the default password if physical access is not restricted.

Basic Configuration:

- Optional: Configure ETH2 as static or DHCP under *Configuration → Network → ETH2*. Do not change ETH2:1 (static IP printed on label).

- Set date and time under *Configuration → Time and Date*.
- Download the LinMot-Pilot OS from the official LinMot download portal. `https://download.linmot.com/TM`
- Verify the installed firmware version against the version indicated in the LinMot-Pilot OS (*.tgz* file) name (contains "FWx.x.x"). Both must match. For comparison, navigate in the Webinterface to *System → Info → Firmware Version* and check.
- If the versions do not match, update the firmware. For detailed instructions, refer to the TM01 Installation Guide.



Install LinMot-Pilot OS:

- Navigate to *PLC-Manager → Control* and click Stop All Applications.
- Go to *PLC-Manager → Application files* and click Upload folder to PLC.
- Upload the LinMot-Pilot OS (.tgz file), then click Send file and Start.
- After the upload, navigate to *System → Reboot* and click Reboot Module.
- Finally, open the LinMot-Pilot HMI in your browser via `http://<IP-Address>`.

Contact & Support

SWITZERLAND

NTI AG
Bodenaeckerstrasse 2
CH-8957 Spreitenbach

Sales and Administration:	+41 56 419 91 91 office@linmot.com
Tech. Support:	+41 56 544 71 00 support@linmot.com https://www.linmot.com/support
Tech. Support (Skype):	support.linmot
Fax:	+41 56 419 91 92
Web:	https://www.linmot.com

USA

LinMot USA Inc.
N1922 State Road 120, Unit 1
Lake Geneva, WI 53147
USA

Phone:	262-743-2555
E-Mail:	usasales@linmot.com
Web:	https://www.linmot-usa.com/

Please visit https://www.linmot.com/contact to find the distribution close to you.

Smart solutions are...

