TM Pilot

Manual

January 2026

Doc.: 0185-1188-E_1V1_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

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.

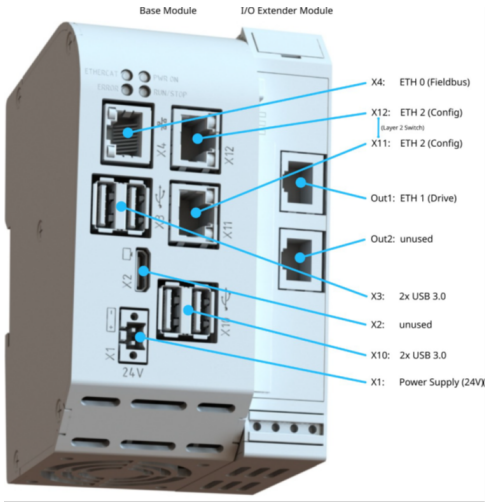
2 Quick Start Guide

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 LinMot Pilot. 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:

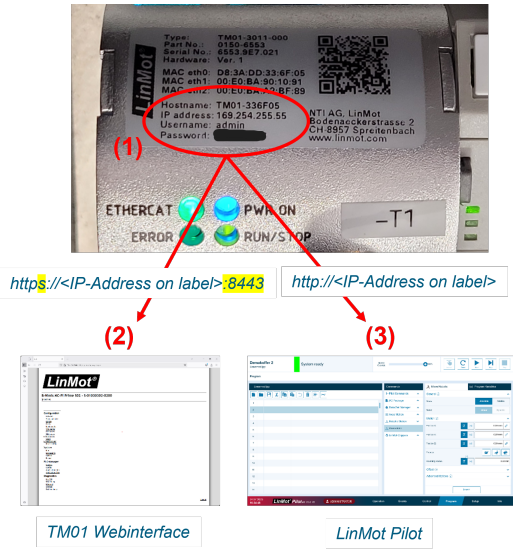
- Regulated 24 VDC power supply (5 A, slow-blow fuse)
- Technology Module TM01
- The license dongle
- LinMot Drives (e.g., C1250-MI or E1450-MI)
- PC connected directly to ETH2 (X11/X12) of the TM01

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 directly to ETH2 (X11/X12) for Webinterface and LinMot Pilot access.
  Optional: Connect Profinet or EtherNet/IP controller to ETH0 (X4).

First Access and Login:

- 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>`.

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.

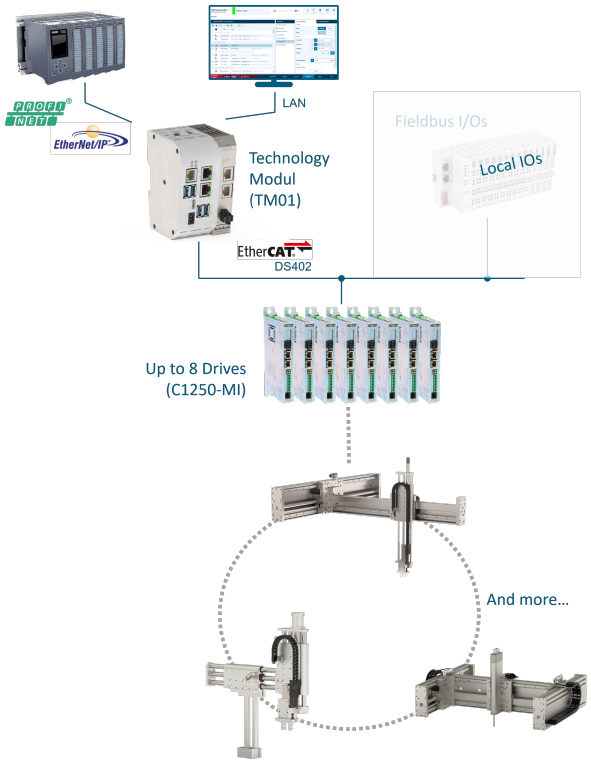
Attention: After reboot, axes will automatically execute homing.

Operating LinMot-Pilot:

- Operation Page: Monitor system status and interact with widgets.
- 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, licenses.
- File Management: Backup and restore configuration and programs.

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. It can be accessed by a web browser under the following link "http://<ip-address>". (The default IPv4-Address is printed on the label).

The Webvisu is divided into three main sections:

- **Title Bar with User Controls**
  Used to control the system and display important information such as current events.
- **Main Workspace**
  Displays the visualization of your system and enables direct interaction with it. The Main Workspace can be switched between six different pages.
- **Navigation Menu Bar**
  Allows to switch between these workspaces quickly and easily.

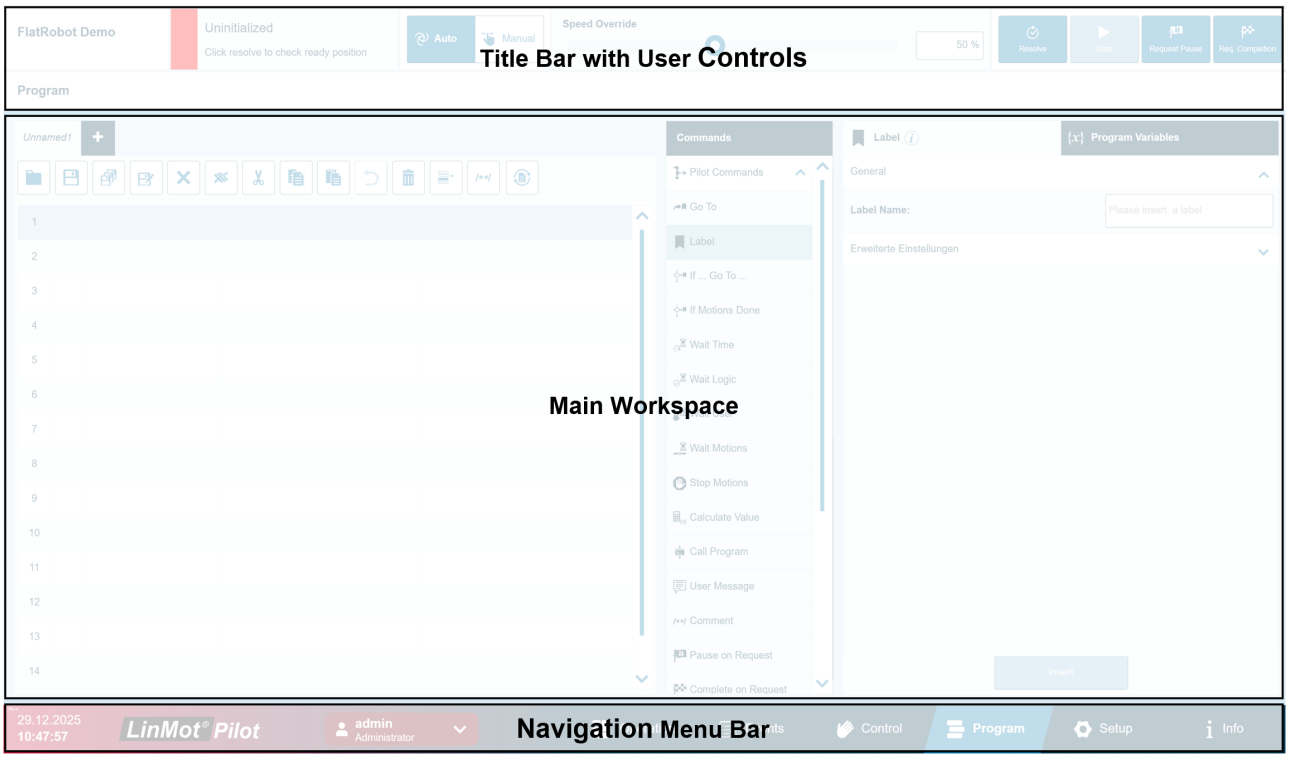


Fig. 1: The Webvisu interface with highlighted sections: Title Bar, Main Workspace and Navigation Menu Bar.

The following sections explain how to use these areas to work efficiently and safely with the Webvisu.

3.1 Title Bar

The Title Bar is the central control element of your system interface, located at the top of the GUI. It provides both status information and direct control options, making it an essential component for system interaction. The Title Bar is divided into five distinct sections, each serving a specific purpose.

System Name:

On the far left, the current system name is displayed. This name can be customized by clicking directly on it, allowing you to assign a unique identifier to your system. This feature is particularly useful when managing multiple systems simultaneously.

[Image]

Event Field:

Next to the system name is the event field. This area displays active events such as errors or warnings. If no events are active, the field shows the current system state. For more details on event handling and system states, refer to the chapter System.
[Image]

Mode Toggle:

The third section contains the mode toggle button, which allows you to switch between Automatic Mode and Manual Mode. The selected mode determines the controls available in the next section.
[Image]

Mode-Specific Controls:

This section dynamically adapts to the selected mode. In Automatic Mode, a Speed Override Slider is displayed. This slider lets you adjust the system operating speed by applying a factor, without interrupting automatic operation.

If the system is in Manual Mode, additional control buttons appear:

Button	Name	Description
Image	Abort	Immediately stops the current process
Image	Step	Executes the next step in the program
Image	Reset	If a reset program is set, a Reset button will also be shown to return the system to its initial position

[Image]

Control Section:

On the far right of the Title Bar is the control section, which contains the primary buttons for system operation. These buttons are described in detail below:

Button	Name	Description
Image	Resolve	
Image	Start	
Image	Pause	
Image	Request Pause	
Image	Request Completion	

[Image]

3.2 Operation

The Operation page is designed for interacting with widgets placed in the visualization. These widgets allow you to monitor and control various aspects of the system directly from the Webvisu interface.
If multiple widget pages are available, you can switch between them using the tabs located at the top of the Operation page. Each tab represents a different widget page, enabling you to organize controls and information efficiently.
By pressing the Edit button located at the top-right corner of the workspace, users belonging to the Service or Administrator user groups can enter the Edit Mode of the Operation page.

Tabs:

Tabs allow you to organize widgets into separate pages for better clarity.
In Edit Mode, you can:
- Add new tabs to create additional widget pages by clicking the plus (+) symbol next to the existing tabs.
- Rename existing tabs for better clarity by clicking on the current name.
- Delete tabs that are no longer needed by clicking on the cross next to the name.

Widgets:

When Edit Mode is activated, the grid becomes visible, providing a clear structure for placing widgets.

Adding Widgets:
1. Click the plus (+) symbol on the grid.
2. The selection dialog opens, allowing you to choose the desired widget.
3. Confirm your selection to place the widget on the grid.

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

Removing Widgets:
- Click the X in the top-right corner of the widget you want to delete.

Configuring Widgets:
- Assign system variables or parameters to the widget.
- Adjust display options (e.g., labels, units).

In the following a short description of the available Widgets is given:

Cycle Time History:

Last 100 Cycles:

Gauge:

Toggle Variable:

3.3 Events

The Events page provides an overview of system messages and notifications. It is divided into two submenus:
- **Active:** Displays all currently active events.
- **History:** Displays past events.

You can switch between the two submenus by clicking the corresponding menu item on the left-hand side of the page. Both submenus display events in a table format, but with different columns:
- **Type:** Category of the event (Error, Warning, Info)
- **Arrived:** Timestamp when the event occurred
- **Gone:** Timestamp when the event was resolved (only History)
- **Source:** Origin of the event
- **Number:** Event ID or code
- **Text:** Description of the event
- **Help:** Instructions on how to resolve the issue (only Active)

The Events page provides filtering options to help you focus on specific event types. There are three filter buttons located in the title bar of the workspace:
- **Error:** Displays only error messages.
- **Warnings:** Displays warning messages.
- **Infos:** Displays informational messages.

You can activate multiple filters at the same time. For example, selecting both Error and Warnings will show only those two categories.

When viewing the History submenu, an additional button appears in the title bar that allows you to permanently delete the entire archive of past events.

3.4 Control

The Control Page provides three main submenus for managing and monitoring system operations:

Axis Control:

The Axis Control section allows direct control of individual axes. Users can:
- Manually operate axes by sending commands directly.
- Monitor real-time status values of each axis.

This section is essential for precise adjustments and troubleshooting axis-related movements.

Quick Control:

The Quick Control section enables users to create a customized control interface similar to the Operation Page. Key features include:
- Widget Placement: Users can add application-specific widgets for quick access.
- Application-Dependent Widgets: These widgets can control components such as a gripper or other specialized tools.

This section is designed for fast and intuitive control of frequently used functions.

Application Control:

The Application Control section provides access to additional parameters related to specific applications. Users can:
- View application-specific values for monitoring purposes.
- Write or modify values to adjust application behavior.

This section is ideal for fine-tuning and configuring advanced application settings.
--> link to Chapter Applications

3.4.1 Axis Control

The Axis Control section allows users to manage and monitor individual axes in detail.
Important: This page is only operable in Manual Mode. Automatic operation is not possible from this interface.
Each axis in the system has its own dedicated control panel, providing real-time information and interactive controls for precise operation. Every axis panel contains the following elements:

Axis Header:

- Axis Name: Displayed at the top (e.g., Single Axis 1).
- Axis Graphic: A visual representation of the axis for easy identification.

[Image]

Power Control:

- Power Button: Enables or disables the selected axis.

[Image]

Current Position:

- Displays the real-time position of the axis.

[Image]

Status Indicators:

- Communication On: Indicates that communication with the axis is active.
- Enabled: Shows that the axis is enabled for movement.
- Standstill: Indicates that the axis is currently not moving.
-

[Image]

Control Mode:

Allows switching between two modes:
- Position Mode: Control the axis by position.
- Force Mode: Control the axis by applying a specific force.

[Image]

Manual Control:

When Position Control is active, the panel displays:
- Jog Buttons (- / +): Move the axis incrementally in small steps.
- Target Position: Enter a desired position and press the Execute button to move the axis to that position.
- Demand Position: Displays the commanded position currently being applied.

[Image]

When Force Control is active, the panel displays:
- 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 currently being applied.

[Image]

Additional Information:

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

[Image]

3.4.2 Quick Control

The Quick Control page provides a flexible interface for creating custom control layouts using widgets. These widgets allow direct interaction with axes and other system components, enabling fast and intuitive operation. The process for adding and deleting widgets on the Quick Control page is identical to the procedure on the Operation Page.
Widgets that are specific to certain applications (e.g., controlling a gripper or a specialized tool) are described in the Applications chapter. General Widgets will be described in the following.

Widget 1:

3.5 Program

The Program Page allows you to create and edit LinMot Pilot Programs. This section explains the layout and functionality of the page.

Page Layout Overview:

- Left Panel: Displays the current program in a table view.
- Middle Panel: Contains the list of available commands.
- Right Panel: Displays settings for the selected command.

[Image]

Program Tabs:

Each program is represented by a tab at the top of the screen. To switch to a different program, simply click on the corresponding tab. If you want to create a new program, click on the tab with the “+” symbol. This action will add a new unnamed program, which will then be displayed in the table.

Adding a Command:

To insert a new command into the program:
1. Select the desired row in the table (left panel) where you want to insert the command.

2. In the middle panel, choose the command you want to add.
3. In the right panel, configure the settings for the selected command.
4. Click the Insert button to add the command to the selected row.

Editing an Existing Command:

To modify an existing command:
1. Click on the row containing the command you want to edit. The current settings will automatically appear in the right panel.
2. If you want to change the command type: Select a new command from the middle panel.
3. Update the settings in the right panel as needed.
4. Click the Apply button to confirm the changes.

Program Editing functions:

Button	Name	Description
Image	Open File	opens
Image	Save File	save current file
Image	Save All Files	save all currently opened files
Image	Save File as	save current file as ...
Image	Close File	close current file
Image	Close all Files	close all currently opened files
Image	Cut	cut currently selected row(s)
Image	Copy	copy currently selected row(s)
Image	Paste	insert row(s) from the clipboard to currently selected row(s)
Image	Undo/Redo?	undo/redo the last change
Image	Delete Row	delete the currently selected row(s)
Image	Add Row	add an empty row above the currently selected row
Image	Comment Out	comment out the currently selected row(s)
Image	Update Descriptions	update the descriptions of the current file

Tips:

- Always ensure the correct row is selected before inserting or editing commands.
- Use the Apply button only after making all necessary changes to avoid overwriting settings unintentionally.

3.6 Setup

Geben Sie hier den Text ein.

3.6.1 System Config

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

- Display
  Allows you to set the time zone for the system. This ensures that all displayed times correspond to your local or preferred region.

- 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.

- 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:
todo

Network:
todo

PLC Interface:
todo

3.7 Info

The Info page provides detailed information about the system configuration and software environment. It is divided into two submenus:
- Hardware Displays the current hardware configuration.
- Software Displays software version details and application licenses.

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:

At the top of the page, the current software version number is displayed.
Below the version information, all application licenses are listed. Clicking on the license name expands a description of the license.

For each license, the following status is shown:
- Activated The license is fully activated.

- Expires in ... The license is in trial mode.
- Expired The trial period has ended.
- Not activated Instead of a status text, a button labeled "start trial" is displayed.

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

3.8 Navigation Menu Bar

The main navigation bar is located at the bottom of the window and provides quick access to essential information and functions. It is divided into several sections:

- **Date and Time**
  On the far left, the current date and time are displayed.

- **User Management**
  Next to the company logo is the User Management dropdown. This menu allows you to use Quick Actions like Switch User, Lock and Logout. A detailed description of these functions can be found in the User Management chapter.

- **Language Selection**
  The Language Selection dropdown enables you to change the display language of the user interface.

- **Navigation Between Main Pages**
  To the right, you will find buttons for switching between the main pages of the application.

[Image of the Main Navigation Bar]

4 Program Commands

All commands as a list

4.1 Pilot Commands

Geben Sie hier den Text ein.

4.1.1 Go To

Jumps to a specified label in the sequence.

Inputs:

Input	Description	Allowed Values
GoTo Label	label in program to jump to	any of the existing Labels
Comment	explained in Common Inputs	

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

Defines a target point for *Go To*, *If … Go TO …* and timeout actions.

Inputs:

Input	Description	Allowed Values
Label Name	name of the label used for program control	text with up to 32 characters
Comment	explained in Common Inputs	

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 ...

Perform a Go To action based on a logic expression.

Inputs:

Input	Description	Allowed Values
Expression	explained in Common Inputs	
Go To → change to *Go To Label*	label in program to jump to	any of the existing Labels
Comment	explained in Common Inputs	

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

Perform a Go To action based on selected motions having finished.

Inputs:

Input	Description	Allowed Values
Select All Motions	check all available axes and processes	*On* / *Off*
Motion	individual axis or process to be checked	any of the connected axes, *Robotics* or *Gripping* process
Go To → change to *Go To Label*	label in program to jump to	any of the existing Labels
Comment	explained in Common Inputs	

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 Wait Time

Wait for a set duration before continuing the program.

Inputs:

Input	Description	Allowed Values
Wait Time	duration of the delay in miliseconds	positive values only
Comment	explained in Common Inputs	

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.

4.1.6 Wait Logic

Wait until the a logic expression computes to TRUE, before continuing the program.

Inputs:

Input	Description	Allowed Values
Expression	explained in Common Inputs	
Timeout	explained in Common Inputs	
Comment	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.

4.1.7 Wait User

Display a message in a pop-up dialog and wait for a user input, before continuing the program.

Inputs:

Input	Description	Allowed Values
Instruction	message shown in the pop-up dialog	text with up to 255 characters
Button Go To	text on the button that causes Go To	text with up to 32 characters
Button Continue	text on the button that causes program to continue	text with up to 32 characters
Go To	label in program to jump to	any of the existing Labels

→ change to *Go To Label*		
Comment	explained in Common Inputs	

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. 2 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").

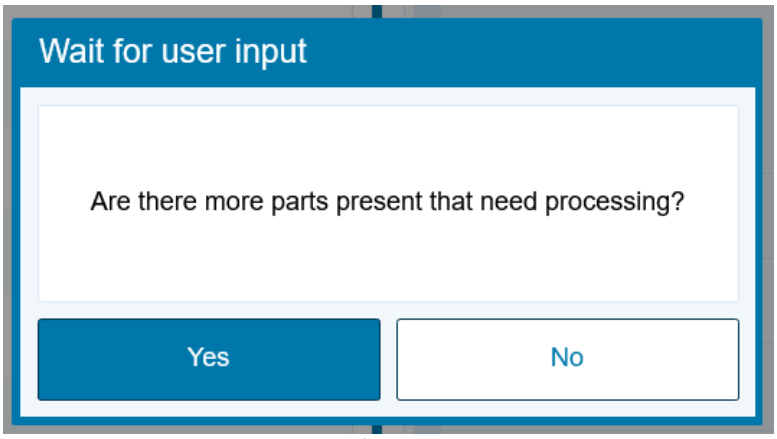


Fig. 2: An example for the Wait User dialog.

4.1.8 Wait Motions

Wait until selected motions have finished, before continuing the program.

Inputs:

Input	Description	Allowed Values
Select All Motions	wait for all available axes and processes	*On* / *Off*
Motion	individual axis or process to be checked	any of the connected axes, *Robotics* or *Gripping* process
Timeout	explained in Common Inputs	
Comment	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.

4.1.9 Stop Motions

Stop the selected motions immediately.

Inputs:

Input	Description	Allowed Values
Select All Motions	stop all available axes and processes	*On* / *Off*

Motion	individual axis or process to be stopped	any of the connected axes, *Robotics* or *Gripping* process
Comment	explained in Common Inputs	

Functionality:

The configured motions are stopped based on the individual stopping settings of the axis or process. Add Link to Setup / Stopping Settings as example.

4.1.10 Calculate Value

Compute the numerical value of an expression during runtime and write it to a Data Reference.

Inputs:

Input	Description	Allowed Values
Target Variable	Data Reference to be written to	any Data Reference that allows writing its value
Expression	explained in Common Inputs	
Comment	explained in Common Inputs	

Functionality:

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

4.1.11 Call Program

Open or switch to a program, execute it from row 1 and return after execution.

Inputs:

Input	Description	Allowed Values
Program	the sub-program to be called	any program saved on the
Comment	explained in Common Inputs	

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. 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.

4.1.12 Run Program

Not implemented yet.

4.1.13 User Message

should it be renamed to "User Event"?

Creates a user-defined event that is logged and displayed according to the Event Type.

Inputs:

Input	Description	Allowed Values
Message	message to be used as event-text	text with up to 255 characters
Type	type of event and consequence	*Information, Warning, Pause on Error* or *Stop on Error*
Comment	explained in Common Inputs	

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.

4.1.14 Comment

Adds a line with a note that will be ignored during execution.

Inputs:

Input	Description	Allowed Values
Comment	note to be displayed over the whole program row	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.

4.1.15 Pause on Request

Pause the program execution if a pause request is active (i.e., the *Request Pause* button is pressed).

Inputs:

Input	Description	Allowed Values
Use Pausing Sequence	perform a Go To instead of handling request immediately	*On* / *Off*
GoTo Label → change to *Go To Label*	label in program to jump to	any of the existing Labels
Comment	explained in Common Inputs	

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 *Use Pausing Sequence* allows to perform a Go To instead of pausing immediately if a pause request is active. A second *Pause on Request* with *Use 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 *Use Pausing Sequence* has to be followed with a second *Pause on Request* with the option turned *Off* at the end of the pausing sequence.

4.1.16 Complete on Request

Complete the program execution if a completion request is active (i.e., the *Req. Completion* button is pressed).

Inputs:

Input	Description	Allowed Values
Use Completing Sequence	perform a Go To instead of handling request immediately	*On* / *Off*
GoTo Label → change to *Go To Label*	label in program to jump to	any of the existing Labels
Comment	explained in Common Inputs	

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 *Use Completing Sequence* allows to perform a Go To instead of completing immediately if a completion request is active. A second *Complete on Request* with *Use 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 *Use 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 IO Package

Geben Sie hier den Text ein.

4.2.1 Set Output

Not implemented yet.

4.2.2 Pulse Output

Not implemented yet.

4.2.3 Calculate Output

Compute the numerical value of an expression during runtime and write it to an output.

Inputs:

Input	Description	Allowed Values
Target Variable	Data Reference of output to be written to	Data Reference of a configured output
Expression	explained in Common Inputs	
Comment	explained in Common Inputs	

Functionality:

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

4.3 Variable Manager

Geben Sie hier den Text ein.

4.3.1 Select Dataset

Switch the selected dataset by a fixed or variable selection.

Inputs:

Input	Description	Allowed Values
Select Dataset by	choice if dataset is set by constant value or Data Reference	*Selection* / *Variable*
Direct Selection	the dataset to be set	any of the 255 datasets
Selection Variable	variable containing the value for the desired dataset	any Data Reference
Comment	explained in Common Inputs	

Functionality:

Switches the selected dataset of the Variable Manager. If *Select Dataset by* is set to *Selection*, the selected dataset will be switched to the dataset configured in *Direct Selection*. If *Select Dataset by* is set to *Variable*, the value of the Data Reference configured in *Selection Variable* is evaluated at runtime, rounded to the nearest integer and the selected dateset 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.

4.4 Axes Motion

Geben Sie hier den Text ein.

4.4.1 Move Linear Axes

Move up to eight different linear axes to their own target positions with one common set of speed parameters.

Inputs:

Input	Description	Allowed Values
Axis	axis that is being moved according to configurations	any of the connected, linear axes that is not configured yet
Mode	defines how the goal position is defined	• *Absolute:* Move axis to an absolute position • *Relative:* Move axis by a certain distance

Position / Distance	goal position (*Absolute Mode*) or stroke to be moved (*Relative Mode*)	• numerical value within the axis limits configured in Kinematic Configuration • any Data Reference
Offset	shifts the goal position by this value closer to the current position	• any numerical value • any Data Reference
Kinematics	explained in Common Inputs	
Sequence Mode	explained in Common Inputs	
Comment	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.

4.4.2 Move Rotary Axes

Move up to eight different rotary axes to their own target positions with one common set of speed parameters.

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

4.4.3 Set Force/Torque

Set the force or torque of an axis to a specific value and optionally a speed limit.

Inputs:

Input	Description	Allowed Values
Axis	axis on which force/torque is set	any of the connected axes
Force / Torque	target force or torque to be set	• any numerical value in N / Nm • any Data Reference
Enable Speed Limit	select if a speed limit should be set to the axis	*On* / *Off*
Velocity	velocity limit that is set to the axis	• any numerical value in m/s / °/s • any Data Reference
Comment	explained in Common Inputs	

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.

Note: The *Set Force/Torque* command currently has no effect on axes that are part of a kinematic group. Only individual axes (named "Single Axis *n*) can use this feature.

4.5 Robotic Motion

Geben Sie hier den Text ein.

4.5.1 Set Coordinate System

Set coordinate system..

Inputs:

Input	Description	Allowed Values
Coordinate System	coordinate system to be set	• *Machine Coordinate System:* Coordinate System defined by the axes • *Conveyor Coordinate System:* Dynamic Coordinate System defined by the Conveyor • *Custom Coordinate Systems:* Coordinate Systems defined in the setup
Comment	explained in Common Inputs	

Functionality:

Set coordinate system, which then can be used by the Move Robotic command.

4.5.2 Move Robotic

Move your kinematic system to a specified position or by a specified distance in a coordinate system.

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
Coordinate System	defines the coordinate system in which the kinematic system moves	• *Last Set Coordinate System:* Coordinate System set by Set Coordinate System • *Machine Coordinate System:* Coordinate System defined by the axes • *Conveyor Coordinate System:* Dynamic Coordinate System defined by the Conveyor • *Custom Coordinate Systems:* Coordinate Systems 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 Configuration • 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 Configuration • 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 Configuration • any Data Reference
Position / Distance R	goal position (*Absolute Mode*) or stroke to be moved (*Relative Mode*) in R-Direction	• numerical value within the axis limits configured in Kinematic Configuration • 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	• numerical value within the axis limits configured in Kinematic Configuration • any Data Reference

	(blending) between two consecutive movements, eliminating sharp corners and ensuring a continuous path.	
Kinematics	explained in Common Inputs	
Sequence Mode	explained in Common Inputs	
Comment	explained in Common Inputs	

Functionality:

Moves the kinematic system defined in the Kinematic Configuration 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.

4.6 LinMot Grippers

Geben Sie hier den Text ein.

4.6.1 Move Gripper

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

Inputs:

Input	Description	Allowed Values
Move Mode	defines how the goal position is defined	• *Absolute:* Move gripper jaw(s) to an absolute position • *Relative:* Move gripper jaw(s) by a certain distance
Jaw Mode	defines which jaws and how they move	*Opposing, Single Jaw 1, Single Jaw 2* or *Parallel*
Position / Distance	goal position (*Absolute Mode*) or stroke to be moved (*Relative Mode*) in mm	• numerical value within the axis limits configured in Kinematic Configuration • any Data Reference
Kinematics	explained in Common Inputs	
Sequence Mode	explained in Common Inputs	
Comment	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 *Opposing* is available for *Jaw Mode*.

Jaw Mode \ *Move Mode*	Absolute	Relative
Opposing	Move both jaws to defined, absolute position.	Move both jaws 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.

4.6.2 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
Target Position	gripping position in mm, to which the jaws will move	• numerical value within the axis limits configured in Kinematic Configuration • any Data Reference
Kinematics	explained in Common Inputs	
Pre-Position	explained in Common Inputs	
Enable Force Check	check if the gripping force is inside a certain range	*On* / *Off*
Target Force	desired gripping force in % of the nominal force	• positive, numerical value up to 500% • any Data Reference
Force Direction	desired direction of the gripping force	*Closing* / *Opening* (see Setup)
Allowed Deviation	allowed force deviation from *Target Force* in % of the nominal force	• numerical value above 0% and up to 500% • any Data Reference
Sequence Mode	explained in Common Inputs	
Comment	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 *Target Force* + *Allowed Deviation*) 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 *Target Force* - *Allowed Deviation*) 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.

4.6.3 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
Force	desired gripping force in % of the nominal force	• positive, numerical value up to 500% • any Data Reference
Force Direction	desired direction of the gripping force	*Closing* / *Opening* (see Setup)
Kinematics	explained in Common Inputs	
Pre-Position	explained in Common Inputs	

Enable Position Check	check if the gripping position is inside a certain range	*On / Off*
Target Position	desired gripping position in mm, at which the jaws will make contact	• numerical value within the axis limits configured in Kinematic Configuration • any Data Reference
Allowed Deviation	allowed position deviation from *Target Position* in mm	• numerical value above 0 • any Data Reference
Sequence Mode	explained in Common Inputs	
Comment	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 *Target Position* + *Allowed Deviation*) 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 *Target Position* - *Allowed Deviation*) 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.

4.6.4 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
Target Position	ungripping position in mm, to which the jaws will move	• numerical value within the axis limits configured in Kinematic Configuration • any Data Reference
Kinematics	explained in Common Inputs	
Pre-Position	explained in Common Inputs	
Sequence Mode	explained in Common Inputs	
Comment	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.

4.6.5 Gripper Monitoring

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

Inputs:

Input	Description	Allowed Values
Enable Position Monitoring	check if the gripping position stays inside a certain range	*On* / *Off*
Target Position	desired position of the jaws in mm	• numerical value within the axis limits configured in Kinematic Configuration • any Data Reference
Allowed Deviation	allowed position deviation from *Target Position* in mm	• numerical value above 0 • any Data Reference
Enable Force Monitoring	check if the gripping force stays inside a certain range	*On* / *Off*
Target Force	desired force in % of the nominal force	• positive, numerical value up to 500% • any Data Reference
Force Direction	desired direction of the force	*Closing* / *Opening* (see Setup)
Allowed Deviation	allowed force deviation from *Target Force* in % of the nominal force	• numerical value above 0% and up to 500% • any Data Reference
Comment	explained in Common Inputs	

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.

4.6.6 Set Grip Force

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

Inputs:

Input	Description	Allowed Values
Force	desired gripping force in % of the nominal force	• positive, numerical value up to 500% • any Data Reference
Force Direction	desired direction of the gripping force	*Closing* / *Opening* (see Setup)
Jaw Mode	defines to which jaw(s) the force is applied	*Opposing, Single Jaw 1* or *Single Jaw 2*
Comment	explained in Common Inputs	

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.

4.7 Conveyor Tracking

Geben Sie hier den Text ein.

4.7.1 Sync to Conveyor

Geben Sie hier den Text ein.

4.7.2 Desync from Conveyor

Geben Sie hier den Text ein.

4.8 Cognex Interface

Geben Sie hier den Text ein.

4.8.1 Login Cognex

Geben Sie hier den Text ein.

4.8.2 Logout Cognex

Geben Sie hier den Text ein.

4.8.3 Load Job Cognex

Geben Sie hier den Text ein.

4.8.4 Trigger Cognex

Geben Sie hier den Text ein.

4.8.5 Set Value Cognex

Geben Sie hier den Text ein.

4.8.6 Get Value Cognex

Geben Sie hier den Text ein.

4.9 Common Inputs

Advanced Options:

Input	Description	Allowed Values
Sequence Mode	Defines when the program execution of the next row in the program will start. *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*
Comment	add a comment to the command that will be displayed in the right-most program column	text with up to 128 characters

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	start timer and perform Go To if timer runs out	*On* / *Off*
Timeout Time	time duration after which Go To is performed in seconds	any positive, numerical value
On Timeout Go To → change to *Go To Label*	label in program to jump to	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. 3. 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.

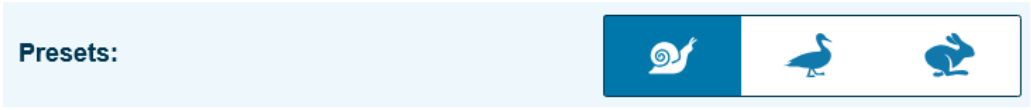


Fig. 3: The preset buttons indicate that all Data References are currently set to the slow motion defaults of the appropriate category.

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
Kinematics	Parameters used only for the motion to the pre-position. Explained in Common Inputs.	

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).

5 Applications

Geben Sie hier den Text ein.

5.1 LinMot Grippers

Geben Sie hier den Text ein.

5.1.1 Control

Geben Sie hier den Text ein.

Process Outputs:

- Error:
- Grip Active:
- Monitoring Active:
- Position Above Target:
- Position Below Target:
- Force too High:
- Force too Low:
- Position Jaw 1:
- Position Jaw 2:
- Force Jaw 1:
- Force Jaw 2:

Widgets:

5.1.2 Setup

Geben Sie hier den Text ein.

Default Speed Parameters:

Stopping Settings:

- Stopping Deceleration:
- Stopping Jerk:

Process Settings:

- TimeOut Time:
- Force Limit during Pre-Positioning:

- Set Force during Pre-Positioning:
- Position Transmission Factor:
- Direction Definition:

Widget Settings:

- Gripping Velocity:
- Gripping Direction:
- Ungrip Position:

5.2 Conveyor Tracking

Geben Sie hier den Text ein.

5.2.1 Control

Geben Sie hier den Text ein.

5.2.2 Setup

Geben Sie hier den Text ein.

5.3 Variable Manager

Geben Sie hier den Text ein.

6 System

Geben Sie hier den Text ein.

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. 4) which is an adaption of the PackML State Diagram (see the OMAC Website for more detail).

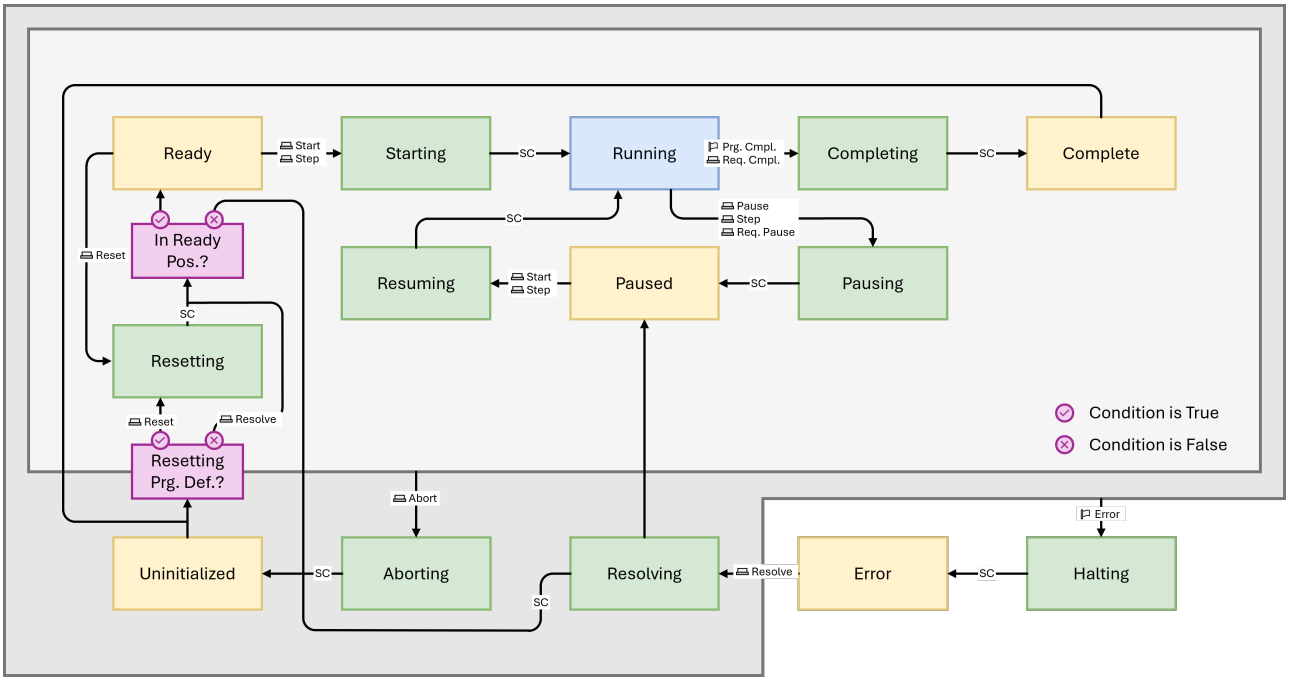


Fig. 4: The state machine of LinMot-Pilot.

Fig. 5 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 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.

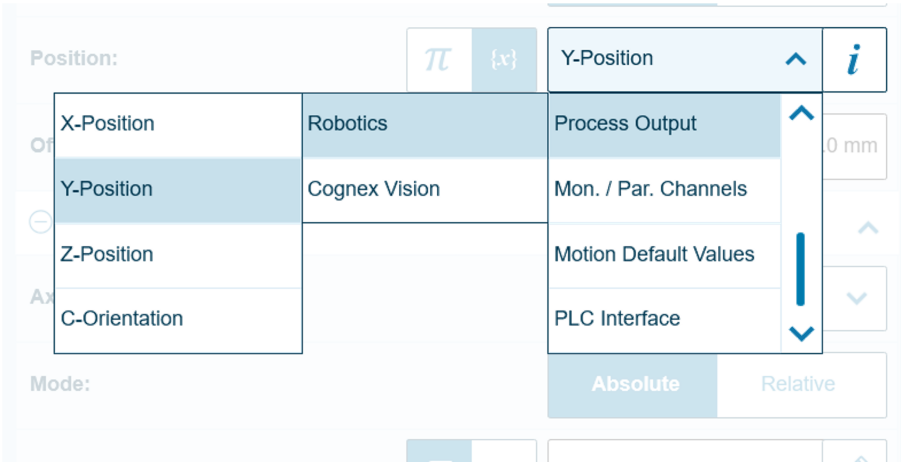


Fig. 6: The Data Reference drop-down.

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

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.
	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.

6.2.1 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.

6.2.2 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.

6.2.3 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.

6.2.4 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)

6.2.5 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 configured in- / outputs for each device
Writing allowed	Only for outputs
Renaming allowed	Yes, possible in *Control* → *Application Control* → *I/O Package*.

6.2.6 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

6.2.7 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

Available Categories	Each drive is a separate category. The displayed drive name is based on the Kinematic Configuration.
Renaming allowed	No

6.2.8 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
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

6.2.9 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 Event Handling

Certain events lead LinMot-Pilot to produce a message that can be shown in the ?? Event Field -> Namen anpassen ?? 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.

6.3.1 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.

6.3.2 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.

Prefix in Event-ID	SYS
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.

Pilot Commands:

Not implemented yet.

Prefix in Event-ID	PCD
Number in Event-ID	
Example: "PRG-B"	

Axes Motion:

Not implemented yet.

Prefix in Event-ID	AXM
Number in Event-ID	
Example: "PRG-B"	

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. Not implemented yet.

Prefix in Event-ID	CGX
Number in Event-ID	
Example: "PRG-B"	

LinMot Vision:

An error occurred in a process of the LinMot vision application. Not implemented yet.

Prefix in Event-ID	LMV
Number in Event-ID	
Example: "PRG-B"	

Lift and Rotate:

An error occurred in a process of the lift and rotate application. Not implemented yet.

Prefix in Event-ID	LNR
Number in Event-ID	
Example: "PRG-B"	

6.3.3 Log Files

All events are logged in two separate files. "Log_Pilot_2025_CW52.txt" "PilotEvents_2025_VW52.csv"

6.4 Kinematic Configuration

Geben Sie hier den Text ein.

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.
[Image]

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.
[Image]

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.

[Image]

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 Program Pre-Compiler

Geben Sie hier den Text ein.

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.

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	• Manual axis or application control • Changing the Kinematics or any other system or application settings

Possible using Fieldbus Connection	Impossible using Fieldbus Connection
• 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	• 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

Currently, only Profinet and EthernetIP are supported as fieldbuses. The setup for the two technologies is described below.

7.2.1 Connections

Profinet:

Connect the Profinet-Network to the X4 port of the Technology Module.

EthernetIP:

Navigate to *Setup* → *System Config* → *Network* and configure the Ethernat Adapter ???accoding to what???.

7.2.2 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 blue lamp at *Active Fieldbus:* indicates an established connection and the label will show the currently connected fieldbus type.
Even with an active connection, LinMot-Pilot will not write any output values and the inputs won't take any effect until the PLC-Control is enabled by toggling the *Enable PLC Control:* button. 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.
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 32 variables (32-bit) is transmitted both as in- and outputs on the fieldbus, which can be mapped to Data References by selection in the drop down menu. 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 *Actual Value on Bus* shows the currently transmitted value, even if PLC Control is not enabled. 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, 64 bits for both in- and outputs are transmitted via fieldbus. The bits cannot be mapped to Data References, as they are available as Data References anyway. On the setup page, each bit can be individually renamed and the output bits can be forced to a defined value. Again, the column labeled *Actual Value on Bus* shows the currently transmitted value, even if PLC Control is not enabled. The custom name of the bits is displayed in the Data References drop-down and in descriptions. 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 cylcle 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	64 x BOOL	8 bytes
Input Variables	32 x UDInt	128 bytes
Control Word	16 bits + 2 x USInt	4 bytes
	Total:	144 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 100000.0. 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	Abort Running Program
Pause Program	BIT	1	Pause Running Program (if currently running)	-
Resume Program	BIT	1	Resume Paused Program (if currently paused)	-
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	-
b11	*BIT*	*1*	-	-
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	64 x BOOL	8 bytes
Output Variables	32 x UDInt	128 bytes
Status Word	16 bits + 2 x USInt	4 bytes
	Total:	152 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 100000.0. 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.

7.4.1 Profinet with Siemens PLC

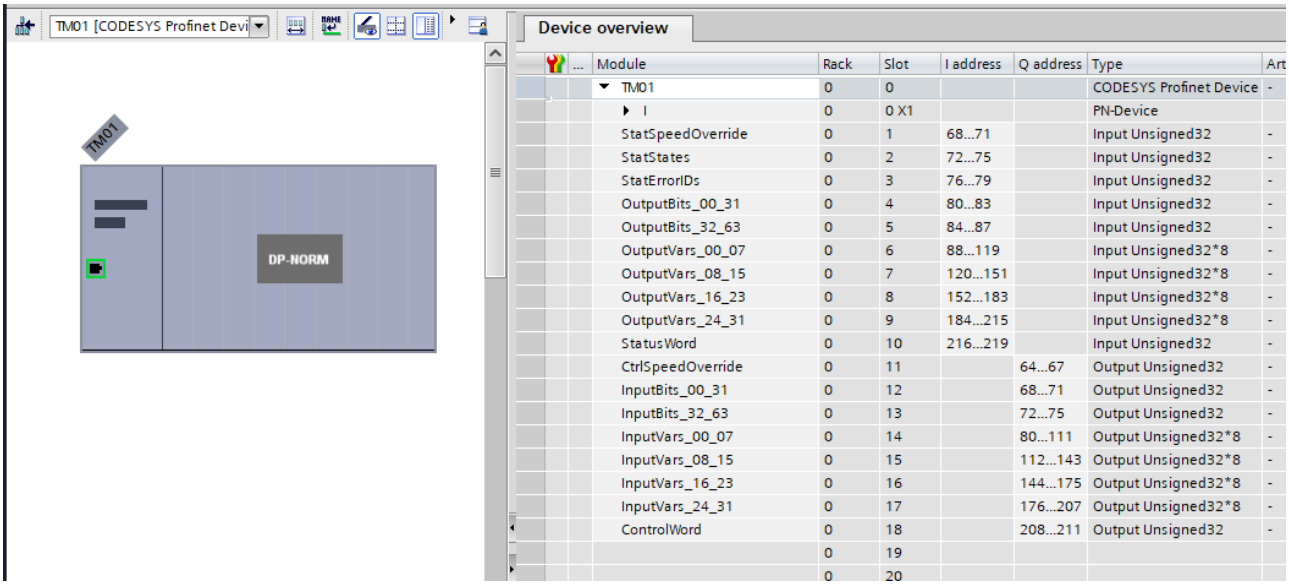
The example project described here can be found ??? LINK ???. Note that this is not a complete documentation of how to use the example but only outlines the most significant parts.

Setup:

This example was created using a SIMATIC S7-1200 and Siemens TIA V19. Connect the Technology Module to the Profinet network of the PLC as described in Connections.

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	Art
TM01	0	0			CODESYS Profinet Device	-
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	-
OutputVars_00_07	0	6	88...119		Input Unsigned32*8	-
OutputVars_08_15	0	7	120...151		Input Unsigned32*8	-
OutputVars_16_23	0	8	152...183		Input Unsigned32*8	-
OutputVars_24_31	0	9	184...215		Input Unsigned32*8	-
StatusWord	0	10	216...219		Input Unsigned32	-
CtrlSpeedOverride	0	11		64...67	Output Unsigned32	-
InputBits_00_31	0	12		68...71	Output Unsigned32	-
InputBits_32_63	0	13		72...75	Output Unsigned32	-
InputVars_00_07	0	14		80...111	Output Unsigned32*8	-
InputVars_08_15	0	15		112...143	Output Unsigned32*8	-
InputVars_16_23	0	16		144...175	Output Unsigned32*8	-
InputVars_24_31	0	17		176...207	Output Unsigned32*8	-
ControlWord	0	18		208...211	Output Unsigned32	-
	0	19				
	0	20				

Fig. 7: 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.

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 X4), 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

If a fatal error occurs, the process monitoring task gets interrupted and one of the following errors will show up on a separate page.
These type of errors should be reported to the technical support at LinMot. Only a device reboot can reset the error.

Error Code	Description
0h	No error

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

Product Code (License)	Content
0187-1000 (OS-LMP01)	LinMot Pilot Operating System (Technologie Modul)
0187-1001 (TL-AXM01)	Axes Motion (LinMot Pilot Technology Library)
0187-1002 (TL-ROB01)	Robotic Motion (LinMot Pilot Technology Library)
0187-1003 (TL-GRP01)	Gripper Function (LinMot Pilot Technology Library)
0187-1004 (TL-CNV01)	Conveyor Tracking Function (LinMot Pilot Technology Library)
0187-1005 (TL-VIS_COG01)	Vision Interface for Cognex SmartCam (LinMot Pilot Technology Library)
0187-1006 (TL-VIS_LMV01)	Vision Interface for LinMot Vision (LinMot Pilot Technology Library)

11 Appendix

Geben Sie hier den Text ein.

11.1 Events

Geben Sie hier den Text ein.

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
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

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
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
100	Error	Reading of program failed
101	Error	Writing of program failed
102	Error	Program buffer is full

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
8101h	Error	Axis is not powered
8102h	Error	Axis is not in standstill

11.1.4 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

11.1.5 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

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...