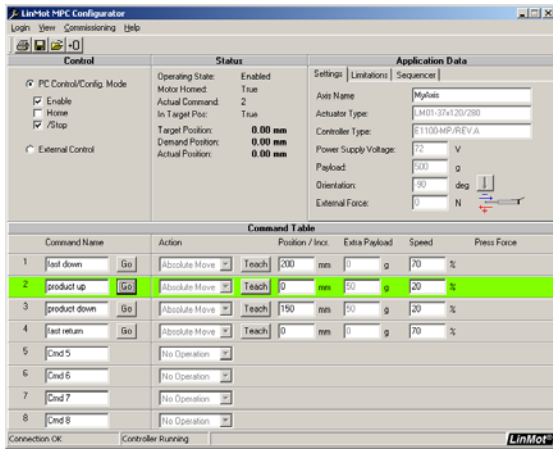

LinMot®



Multi Position Cylinder



MPC Version 3.2 User Manual

© 2007 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

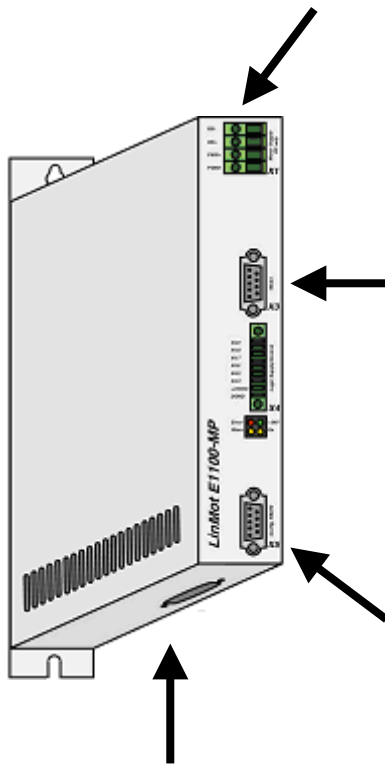
The information in this documentation reflects the stage of development at the time of press and is therefore without obligation. NTI AG. reserves itself the right to make changes at any time and without notice to reflect further technical advance or product improvement. Please refer to the latest edition of our "General business terms"

Document version 1.2/ as, May, 2006

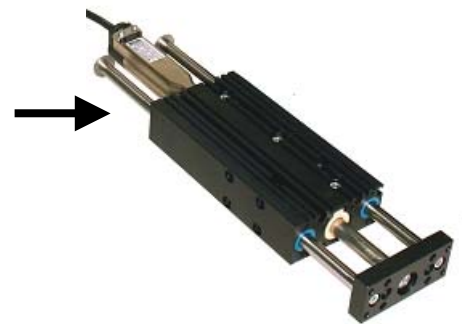
1	System Overview	4
2	A 10-Step Commissioning Example	5
2.1	Step 1: Installation of the MPC Configurator Software on the PC	5
2.2	Step 2: Installation of the MPC Firmware on the Controller	6
2.3	Step 3: Login	7
2.4	Step 4: Actuator Selection.....	8
2.5	Step 5: Application Settings	9
2.6	Step 6: Motion Command Definition.....	10
2.7	Step 7: Enabling the Motor and Homing.....	11
2.8	Step 8: Testing Single Commands and Command Sequences.....	13
2.9	Step 9: Saving the configuration	14
2.10	Step 10: External Control from PLC or IPC.....	15
3	MPC Configurator Settings	17
3.1	Actuator Setup	17
	Settings for Modules and Motors	18
	Advanced Settings for Motors	18
3.2	IO Interface	21
	Operation State Control.....	22
	Command Selection Mode.....	22
	Digital Input Filters.....	25
	Command Executed Feedback.....	26
3.3	Regeneration Resistor	28
3.4	Position Control Loop.....	29
3.5	Application Settings.....	31
4	MPC Commands	33
4.1	Absolute / Relative Move.....	33
4.2	Press.....	35
4.3	Teach Function	36
5	MPC System Control.....	37
5.1	Operating State Diagram.....	37
5.2	Operating State Control Signals.....	38
5.3	Operating State Dependent Outputs	39
5.4	Control Signal Sequence for Enabling and Homing the Motor	40
5.5	Command Call and Feedback with E1100-MP Controller	41
5.6	Command Call and Feedback with E1100-MT Controller	42
5.7	Error Acknowledge	43
5.8	Emergency Stop Behavior	43
A.	MPC Controller Types.....	44
B.	Connector Location on the MPC Controller.....	45

1 System Overview

MPC Controller: E1100 servo controller with a maximum of 8 programmable positions in the 'MP' version and 256 in the 'MT' version.

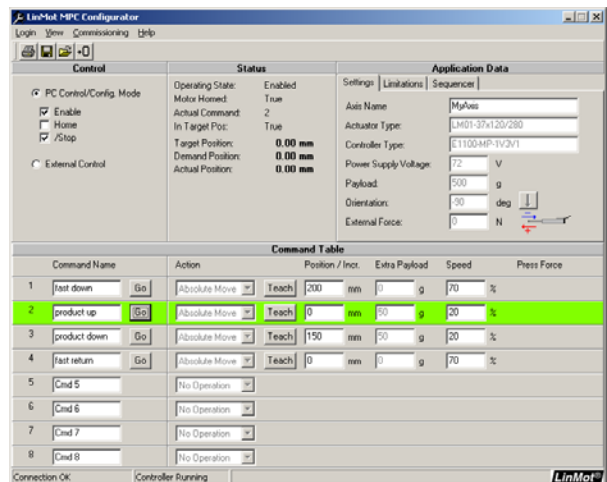


The **MPC Actuator** (typically a linear module) is driven by the MPC Controller. The controller sets the Demand Position and controls the actuator's Actual Position by using the position sensor feedback from the actuator.



For the commissioning of the system the PC software **MPC Configurator** is used. The program communicates over a RS232 connection with the MPC Controller.

The **primary control system** (PLC or IPC) controls the MPC Controller by using **digital IOs**.



2 A 10-Step Commissioning Example

The following example demonstrates the simplicity of commissioning the Multi Position Cylinder system. The example shows the typical procedure for the commissioning of a MPC axis. It provides an overview of the required settings.

Application example: A product of 50g has to be picked and placed by using a vertically mounted linear module LM01-37x120/280. A gripper with weight 500g is attached to the linear module.

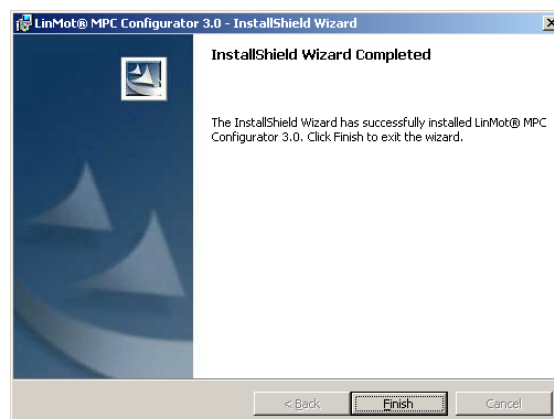
2.1 Step 1: Installation of the MPC Configurator Software on the PC

The MPC Configurator software runs on computers with one of the following Microsoft Windows operating systems:

- Windows NT 4.0
- Windows 2000
- Windows XP

The computer must have a free COM port for the communication with the controller. If the computer doesn't have any COM ports, then the communication can alternatively be established over USB by using an USB to RS232 converter.

The latest version of the MPC Configurator software is available from the **LinMot®** web site www.LinMot.com. Executing the program *MPC3R1setup.exe* installs the MPC Configurator software on your PC.



After successful installation, the MPC Configurator can be started from the MS Windows start menu:

Start -> Programs -> LinMot -> MPC Configurator 3.1

2.2 Step 2: Installation of the MPC Firmware on the Controller

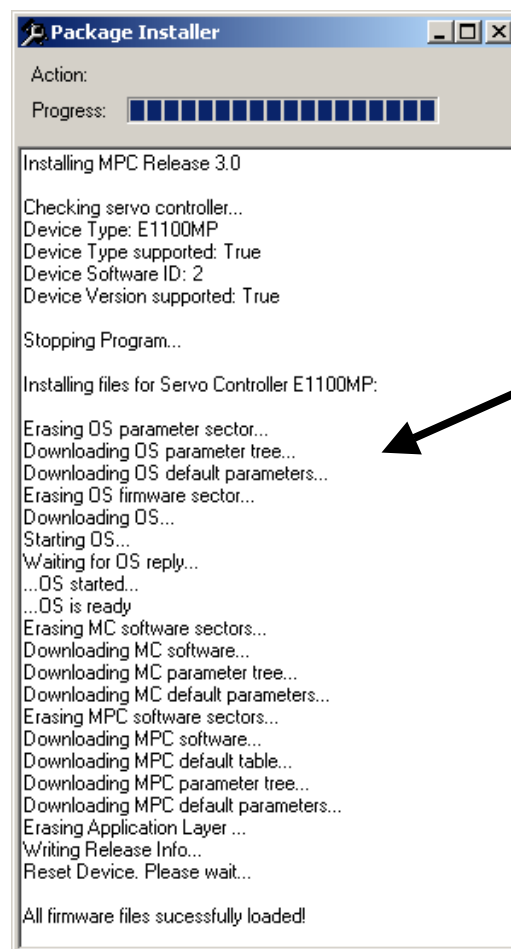
It is recommended to update the controller's firmware before the first commissioning. The current firmware version is part of the MPC Configurator installation and can be downloaded to the controller by using the MPC Configurator software (see Step 1).

The controller has to be powered up (green LED is on) and the PC's COM port must be connected with the configuration interface connector X5 on the controller. For this purpose a 1:1 RS232 cable has to be used (not a null-modem cable!).

After starting the MPC Configurator, the download can be started by clicking the menu item *Commissioning -> Firmware Download*. The script file *MPC3R1.sct* has to be used for fully automated download of all necessary files in the current firmware release.

When the PC's COM port is selected, the *Package Installer* window opens. The MPC Configurator will begin downloading the files.

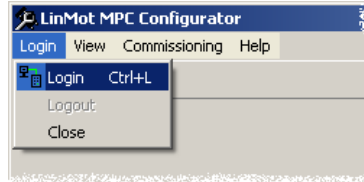
When all firmware parts are successfully installed, the *Package Installer* window can be closed.



The Package Installer window informs you about the firmware installation process.

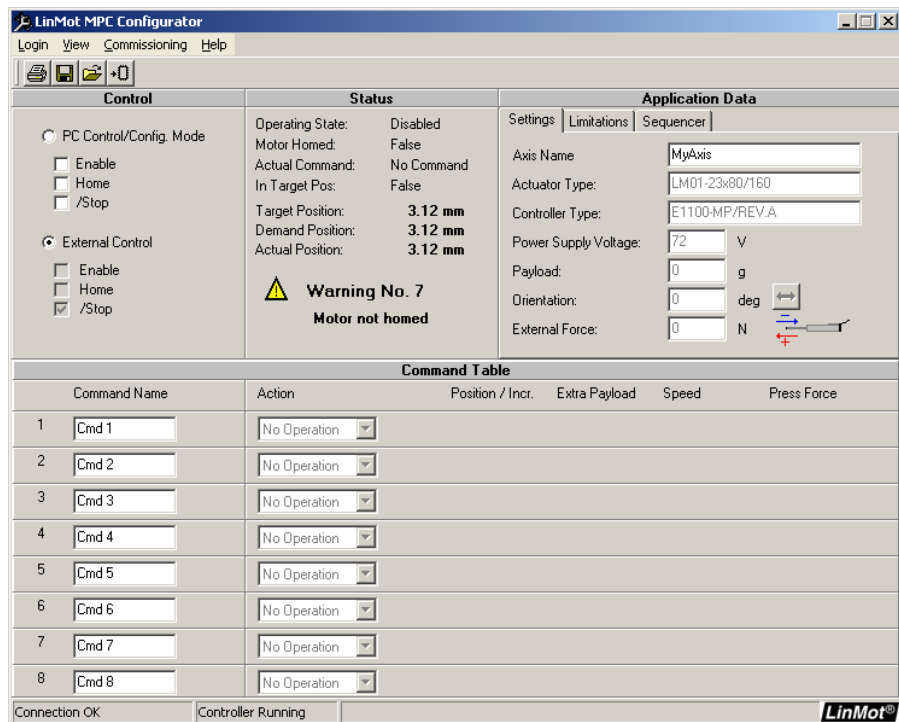
2.3 Step 3: Login

In order to configure the controller click the menu item *Login* -> *Login*:



Select the PC's COM port connected to the controller from the pop-up window. Afterwards the software starts uploading the configuration from the controller. Wait until the process is completed (up to 30s).

After successful login the MPC Configurator user interface is shown.

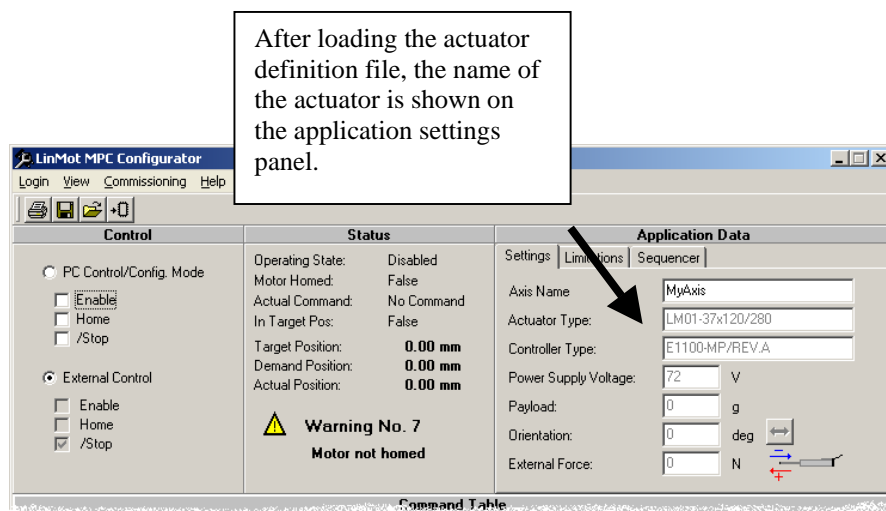


2.4 Step 4: Actuator Selection

Before the axis can be configured, you need to load the the technical data for the particular MPC actuator you are using. These data are stored in so called *Actuator Definition Files* (*.adf).

The menu item *Commissioning -> Actuator Change* opens the dialog window for choosing a definition file.

For our application example we have to select the file *LM01-37x120-280.adf*.



Note: The MPC software concept was particularly developed for using linear modules (from manufacturers defined combination of linear motor and linear guide unit, e.g. LinMot’s *LM01* module family). The *Actuator Definition File* (*.adf) of a linear module contains the technical data of the motor (max. current, force constant, stroke etc.) and the mechanical data of the guide unit (masses, stroke, stop positions). In our application example we are using a linear module. If a linear motor without LinMot’s mechanical guide needs to be commissioned, additional settings for defining the mechanics of the system should be considered (see chapter ‘MPC Configurator Setting: Actuator Setting’).

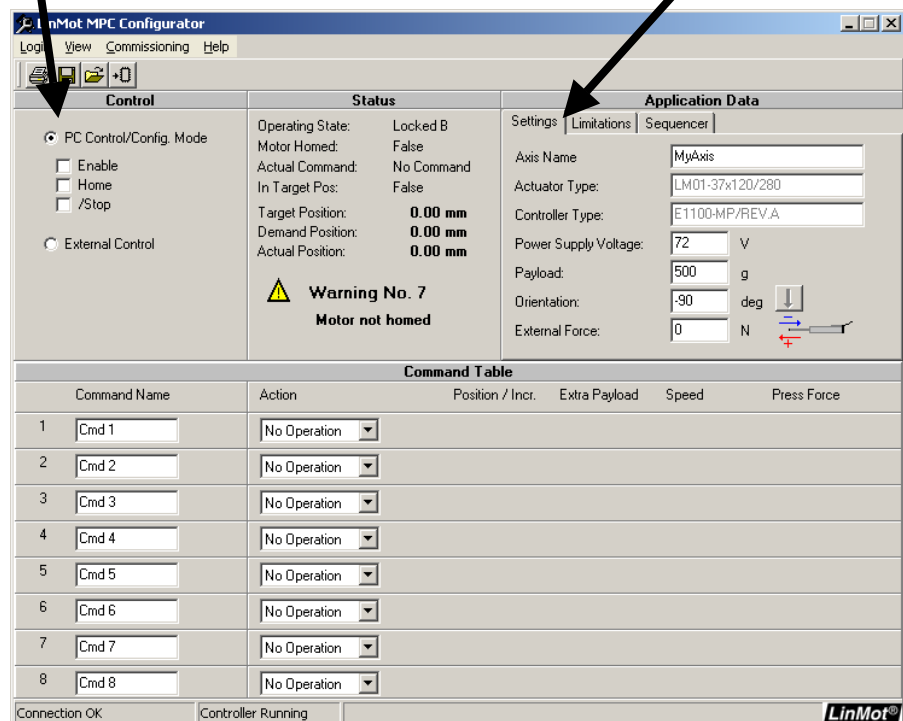
2.5 Step 5: Application Settings

For achieving best control performance, the load must be defined precisely.

Based on the application data the controller calculates the optimal motion profiles and sets the feed forward parameters of its position control loop.

The application data settings can be altered if *PC Control/Config. Mode* is selected.

Power supply voltage, payload, orientation (horizontal, vertical) and external forces have to be defined on the application data settings panel.



For our application example we set the payload mass value to 500g (gripper mass). The masses of the motor and the guidance have been defined for the software when the ADF-file was loaded. Since the product mass is only moved during a segment of the cycle we do not add the value here. The value (50g) will be a segment specific parameter (see next step).

For vertical applications the 'Orientation' angle is defined by the location of the stator's cable egress. In applications where the motor's slider is moving, the value has to be set according to the following rule: If the stator's cable exits from the top this angle is -90° . Conversely if the cable exits from the bottom this angle is $+90^\circ$.

2.6 Step 6: Motion Command Definition

Up to 8 commands can be defined in the MPC Command Table (256 commands with E1100-MT controllers).

For our Example we need to define only four commands:

Command No. 1: Fast move to the pick position.

Command No. 2: Slow move up after picking the product.

Command No. 3: Slow move down to the place position.

Command No. 4: Fast return to the start position.

For ordinary motion tasks we use the command type *Absolute Move*.

Optional text to describe the intention of the command

Command's target position

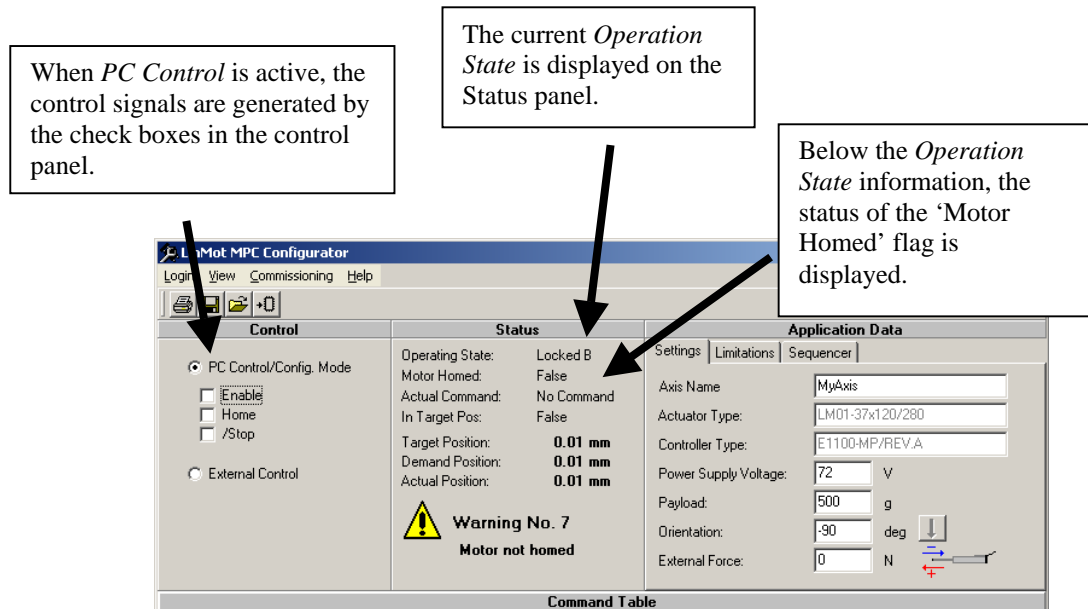
Command Name		Action	Position / Incr.	Extra Payload	Speed	Press Force
1	fast down <input type="button" value="Go"/>	Absolute Move ▾ Teach	200 mm	0 g	70 %	
2	product up <input type="button" value="Go"/>	Absolute Move ▾ Teach	0 mm	50 g	20 %	
3	product down <input type="button" value="Go"/>	Absolute Move ▾ Teach	150 mm	50 g	20 %	
4	fast return <input type="button" value="Go"/>	Absolute Move ▾ Teach	0 mm	0 g	70 %	
5	Cmd 5	No Operation ▾				
6	Cmd 6	No Operation ▾				
7	Cmd 7	No Operation ▾				
8	Cmd 8	No Operation ▾				

If for some motion commands an additional load mass has to be expected (e.g. mass of a product), it is defined in the corresponding command lines.

The speed of the motion is defined as a %-value. 100% is the maximum speed the motor can achieve with the given load data.

2.7 Step 7: Enabling the Motor and Homing

The three control signals *Enable*, *Home* and */Stop* (inverted emergency stop) control the operating state of the MPC Controller. Only when the MPC Controller is in the *Enabled* state and the motor has been homed can motion commands be executed.



The motor will not be powered before the controller is in the *Enabled* state. In the *Enabled* state the motor's position is actively controlled by the digital position control loop. Because the motor has an incremental position measurement system, it is necessary to reinitialize the motor's position after each power off of the controller. Homing is done by moving against a mechanical stop. When the stop has been found, the controller reinitializes the position system relative to the stop position.

The following steps are necessary for bringing the controller to *Enabled* state:

1. *PC Control/Config. Mode* has to be activated (if not already done)
2. Clear the *Enable* signal
3. Set the */Stop* signal (because it is an inverted signal, it has to be set for normal operation)
4. Set the *Enable* signal

After applying this procedure, the controller is in *Enabled* state and the motor is powered on.

The homing procedure can be enabled by setting the *Home* signal. The motor begins to move and the *Operation State* has changed to *Homing*.

After the motor has found the mechanical stop, it moves to a position 5mm away from the stop. This position is now the 0mm position of the MPC reference system.

Successful finish of the homing procedure is shown in the status panel (Motor Homed: True). After clearing the *Home* signal, the controller returns to *Enabled* state and is then ready for the execution of motion commands.

2.8 Step 8: Testing Single Commands and Command Sequences

If the controller is in *Enabled* state and the motor is homed, motion commands can be started by clicking the command's *Go* button (*PC Control* has to be active).

On the *Status* panel the *Target Position* of the current command, the *Demand Position* (interpolation between start and target position) and the *Actual Position* is shown.

Command Name	Action	Position / Incr.	Extra Payload	Speed	Press Force
1 fast down	Absolute Move	200 mm	0 g	70 %	
2 product up	Absolute Move	0 mm	50 g	20 %	
3 product down	Absolute Move	150 mm	50 g	20 %	
4 fast return	Absolute Move	0 mm	0 g	70 %	
5 Cmd 5	No Operation				
6 Cmd 6	No Operation				
7 Cmd 7	No Operation				
8 Cmd 8	No Operation				

The command line's color indicates whether a command is in execution (grey) or already executed (green).

With the *Sequencer* tool it is possible to test command sequences. It can be very helpful during the commissioning process. It allows testing complete cycles without any overlaid controller (very useful during 'burn in').

The numbers of the commands have to be typed comma separated. Dwell times between two motion commands are defined by a "P" followed by the pause time in milliseconds (e.g. time which is necessary for closing a gripper.).

The time which was needed to do the whole sequence (including stand still times) is measured in the controller and displayed on the Sequencer panel.

2.9 Step 9: Saving the configuration

At the end of the commissioning process, the configuration (settings and command table) has to be saved.



It is absolutely necessary that the configuration is saved to the non-volatile memory (flash memory) on the controller. If the configuration is not copied to the flash memory, the settings and commands will be lost with the next power down of the controller.

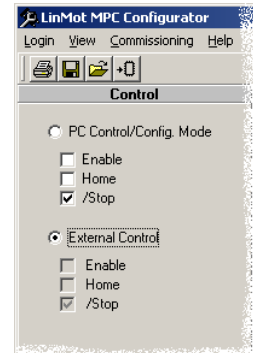
It is possible to save a copy of the configuration to a file on the computer. If the controller has to be replaced or other axes with the same configuration have to be set up, the saved file can be imported.

It is recommended to print out the configuration. These sheets become part of the plant's documentation. A copy of the current configuration should always be in the cabinet, where the MPC Controller is installed.

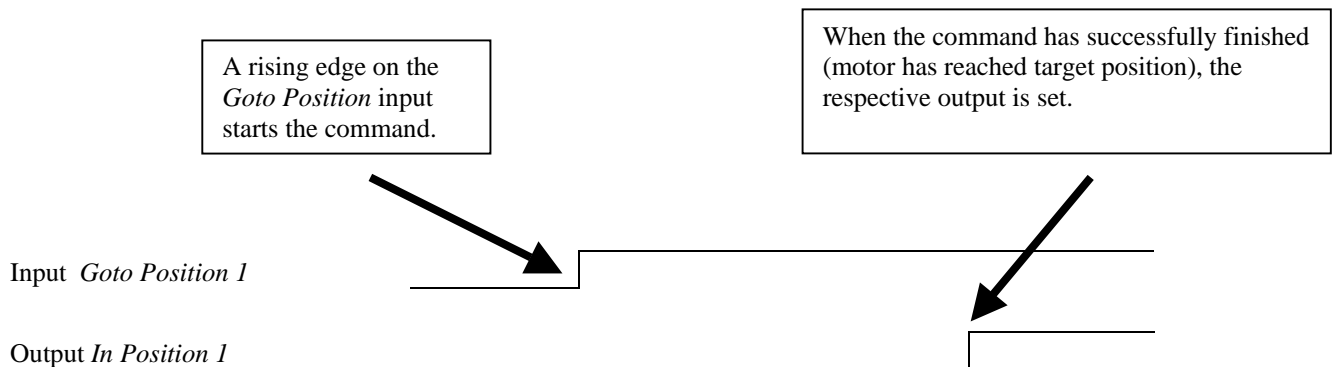
2.10 Step 10: External Control from PLC or IPC

By selecting the item *External Control* the system control is transferred to the IO interface.

The three control signals *Enable*, *Home* and */Stop* are now generated by the corresponding digital inputs. For enabling and referencing the motor, the superior control system (PLC or IPC) has to generate the same signal sequence as we did manually in Step 7.



Also execution of commands is now triggered over digital inputs (*Goto Position 1 .. Goto Position 8*). If only eight (or less) commands are defined, then there is a dedicated digital input for each command. A rising edge on one of those inputs starts the corresponding command. When the command is executed (motor has reached target position), the respective digital output (*In Position 1 .. In Position 8*) is set.



If more than eight motion commands are defined (which is possible only with MT type controllers), then the mode for addressing commands can be changed (see chapter “MPC Configurator Settings: IO Interface”).

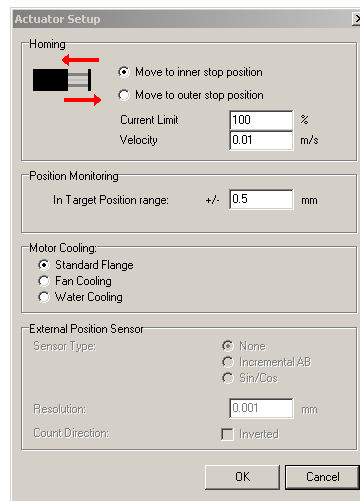
We now are at the end of our short commissioning example. With a fine tuning the system performance can be further optimized. The position and speed values or the position control parameter setting can be changed (see chapter 'MPC Configurator Setting: Position Control'). The command parameters position and speed and the control parameters can be altered even while the system is running!

3 MPC Configurator Settings

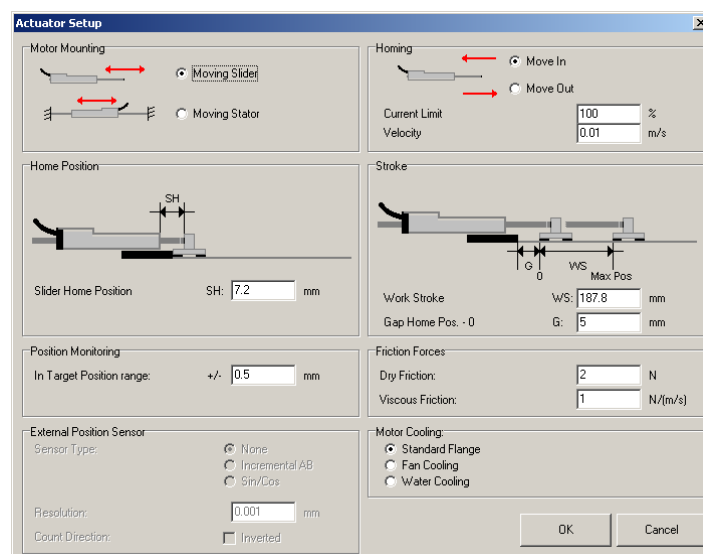
3.1 Actuator Setup

The technical data of the MPC actuators are stored in *Actuator Definition Files* (*.adf). Beside the ADF files for linear modules there are ADF files for the basic linear motors. The ADF files of linear modules contain technical informations about the guide units in addition (weights, strokes and stop positions).

Therefore less parameters have to be set if a predefined linear module is used. More parameters have to be set if a bare motor is chosen as actuator. The menu item *Commissioning* -> *Actuator Setup* opens the actuator setup window. The appearance of the window depends on the selected actuator type.



Actuator Setup window for linear modules



Advanced Actuator Setup window if an adf-file was loaded, which contains only data about the linear motor.

Settings for Modules and Motors

Homing: Before positioning commands can be executed, the motor has to be homed (referenced). During the homing procedure the motor moves against a mechanical stop. The stop position is detected through the rising motor current. You can define the direction and speed of this reference move and the maximal motor current.

In Target Position Range: A command is completed, when the set point generation has finished (*Demand Position = Target Position*) and the *Actual Position* is within the *In Target Position Range* around the command's *Target Position*. When this happens the *In Position* output is set. If the motor's actual position leaves the *In Target Position Range* (through applying external forces), the output is cleared.

Motor Cooling: The controller software estimates the motor's temperature by monitoring the power consumption. If the motor has an extra cooling system, the calculated temperature model has to be set accordingly.

External Positions Sensor: This option is available for controller types E1100-MT and E1100-MT-HC. By using an external position sensor, the positioning accuracy can be increased. The achievable accuracy mainly depends on the feedback and the mechanical systems. Incremental AB sensors (RS422 level) as well as analog sine/cosine sensors (1Vpp) can be used. If an external position sensor is configured, it is strongly recommended to check the settings before the motor is enabled. When the motor is moved manually, the *Actual Position* value should change accordingly. Specifically the count direction has to be set correctly!

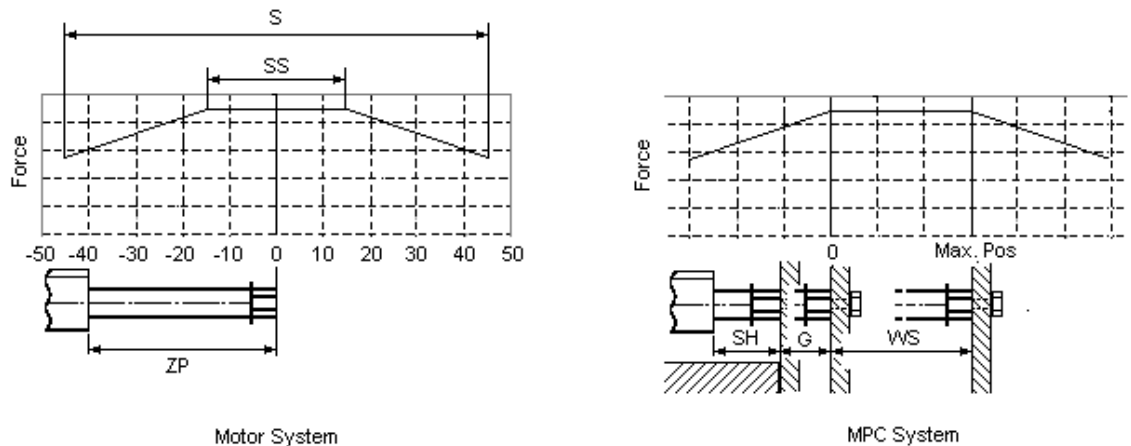
Advanced Settings for Motors

Motor Mounting: *LinMot®* – linear motors can be mounted on two different ways. Either the slider (passive magnetic rod) or the stator (active part of the motor) is moving. The mass of stator and slider are different. For achieving best position control performance, the parameter *Motor Mounting* has to be set correctly.

Slider Home Position (SH): This parameter defines the position of the mechanical reference stop. For setting the parameter correctly, the motor is moved manually to the stop position. Then the distance between front end of stator and front end of slider has to be measured. If the slider's front end is inside the stator at this position, then this value is negative.

Work Stroke (WS): The value of *WS* defines the MPC work stroke, which begins with 0mm and ends at *Max. Pos. = WS*. Only positions inside the work stroke range *WS* can be programmed to be the *Target Position* of motion commands.

Gap Home Pos. - 0 / Gap Max. Pos. – Home Pos. (G): The value of G defines the beginning of the work stroke range WS respectively to the stop position. During the homing procedure the motor moves to the stop position and moves then to the beginning of the work stroke range.



The values of SH , WS and G have to be set having regard to the stroke/force diagram of the motor (values ZP , SS and S from the motor's data sheet). Also the mechanical arrangement of the application has to be considered.

By setting the values of SH , WS and G , the relation between the motor's coordinate system (0mm at position *Stator End* + ZP) and the MPC reference system (0mm = beginning of work stroke range) is defined.

The following rules have to be considered:

- The mechanical stop has to be found within the stroke range S . Therefore the following condition has to be fulfilled when the value of SH is set:

$$ZP - S/2 > SH > ZP + S/2$$

For some motor types the value of $S/2$ is greater than the value of ZP . In those cases the value of SH may be negative. A negative value for SH means, that the slider's front end moves into the stator while referencing.

- The motor can be moved within the range of G (while referencing) and WS (ordinary positioning commands). This is an MPC software limitation. The motor can apply forces over the stroke range S (electro magnetic limitation). Because the ranges G and WS have to be within S , following conditions have to be considered:

Homing Mode *Move In*: $SH + G + WS < ZP + S/2$

Homing Mode *Move Out*: $SH - G - WS > ZP - S/2$

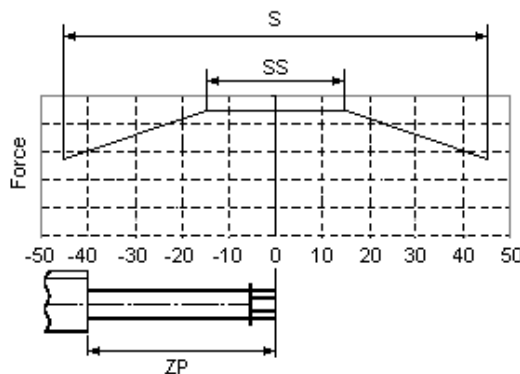
- In the range of SS the motor can apply its maximal force. In general it is recommended to set the range WS equal to SS . This has to be considered already when the axis is designed. By setting the parameter G , the beginning of the work stroke range WS can be set to the beginning of SS .

Example: A linear motor P01-23x80/30x90 has to be referenced by moving the slider's front end in direction of the stator's front end. This example will use the 'snap' ring on the front of the slider as the mechanical homing stop. Home will be defined as the location where this ring abuts the front of the stator. The MPC work stroke range WS should be equal to the range SS .

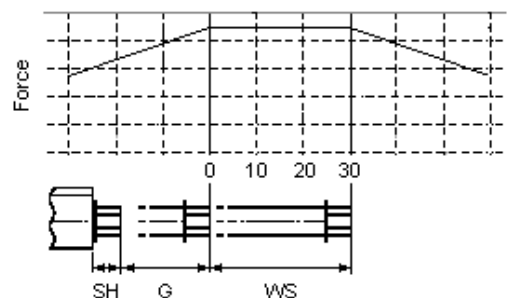
Settings:

- The parameter *Homing Mode* has to be set to *Move In*.
- $SH = 7.2$ mm: According to the design drawing of the slider, this is the distance from the snapper ring inner side to the slider front end.
- $WS = 30$ mm: Size of the SS range defined in the motor's data sheet.
- $G = 17.8$ mm : With regard to the stroke/force diagram of the motor, the value of G has to be calculated by using the following formula:

$$G = ZP - SH - SS/2 = 40\text{mm} - 7.2\text{mm} - 30\text{mm}/2 = 17.8\text{mm}$$



Motor System

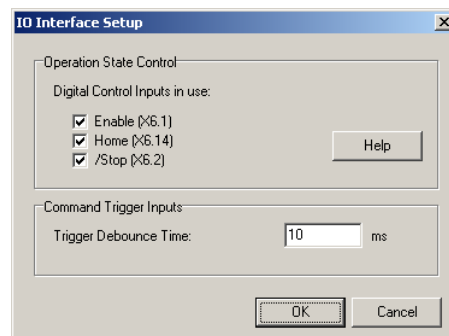


MPC System

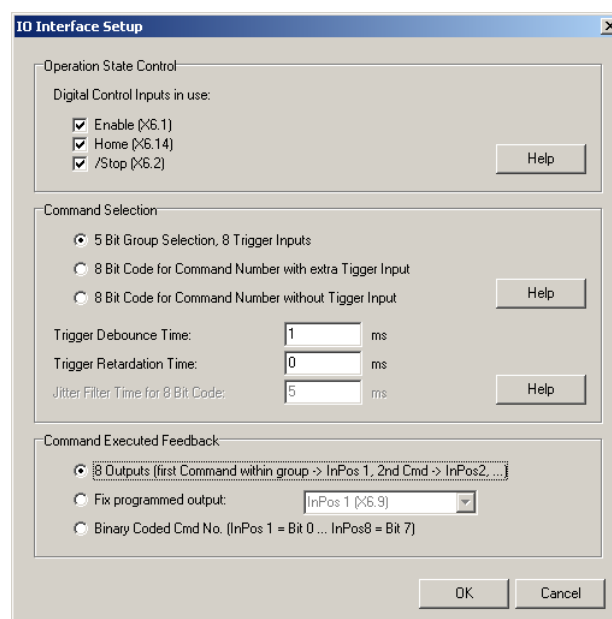
3.2 IO Interface

The two MPC controller types MP and MT implement the same state machine model. For the control of the operating state there are three digital inputs on each controller: *Enable*, *Home* and */Stop*.

But there are differences between the two controller types regarding to how commands can be started and how the controller sets the *command executed* feedback. For each of the eight motion commands of an MP controller, there is a dedicated digital input for the command start (*Goto Position 1 .. Goto Position 8*). There are eight outputs for the corresponding commands (*In Position 1 .. In Position 8*). With an MT type controller it is possible to define up to 256 different motion commands. There are still only 8 inputs and 8 outputs. The MPC Configurator provides different modes for the call and the feedback of the motion commands when an MT type controller is used. The IO Interface Setup window looks different when an MT controller is connected.



IO interface setup for MP controller



Advanced IO interface setup for MT controllers

Operation State Control

There are three digital inputs on the connector X6 for the control of the operating state machine of the controller:

Pin	Name	Function
X6.1	Enable	Enables the motor
X6.14	Home	Enables the homing procedure
X6.2	/Stop	Emergency stop (inverted signal)

The usage of these inputs is recommended, but voluntary. They can be switched off by software. For the inputs which are not in use, the controller software sets the corresponding state transition request flags automatically (auto-enable after powerup, auto-home after enabling, no emergency stop functionality).



Attention: For safety reasons it is strongly recommended to use all the three control inputs. Safety issues when not using these control inputs must be carefully evaluated for the machine.

Command Selection Mode

The command selection mode is alterable only if an MT type controller is used. **If an MP controller is in use, the Command Selection mode is not changeable.** The command selection of the MP controllers corresponds to the MT mode *5 Bit Group Selection, 8 Trigger Inputs* (see below), with the limitation that no groups can be selected. In that case the Group ID is always zero, which means that up to eighth different commands can be used.

If on an MT-Controller only the first 8 commands are used, then its default behavior is exactly the same as if an MP controller is used.

For MT controllers one of the following command selection modes has to be selected:

- 5 Bit Group Selection, 8 Trigger Inputs (default)
- 8 Bit Code for Command Number with extra Trigger Input
- 8 Bit Code for Command Number without Trigger Input

A) 5 Bit Group Selection, 8 Trigger Inputs:

The 256 MPC Commands are arranged in 32 Groups (Group 0..Group 31). Each group contains eight commands.

For the group selection there are five digital inputs on the connector X6 (*Grp0..Grp4*). Before the execution of a MPC command, the superior control system (PLC, IPC) has to set the group address to these inputs (binary coded):

Pin	Name	Function
X6.15	Grp0/Trig	Bit 0 of the Group ID
X6.3	Grp1	Bit 1 of the Group ID
X6.16	Grp2	Bit 2 of the Group ID
X6.4	Grp3	Bit 3 of the Group ID
X6.17	Grp4	Bit 4 of the Group ID

Through a rising edge on one of the following eight trigger inputs the commands are started:

Pin	Name	Function
X6.5	Goto Position 1	Starts 1 st command of the sel. group
X6.18	Goto Position 2	Starts 2 nd command of the sel. group
X6.6	Goto Position 3	Starts 3 rd command of the sel. group
X6.19	Goto Position 4	Starts 4 th command of the sel. group
X6.7	Goto Position 5	Starts 5 th command of the sel. group
X6.20	Goto Position 6	Starts 6 th command of the sel. group
X6.8	Goto Position 7	Starts 7 th command of the sel. group
X6.21	Goto Position 8	Starts 8 th command of the sel. group

B) 8 Bit Code for Command Number with extra Trigger Input:

The command to be executed is selected by its binary coded number, which is set onto eight inputs. A rising edge on an extra trigger input starts the selected command.

The binary coded command number has to be set to the following digital inputs:

Pin	Name	Function
X6.5	Goto Position 1	Bit 0 of the command number
X6.18	Goto Position 2	Bit 1 of the command number
X6.6	Goto Position 3	Bit 2 of the command number
X6.19	Goto Position 4	Bit 3 of the command number
X6.7	Goto Position 5	Bit 4 of the command number
X6.20	Goto Position 6	Bit 5 of the command number
X6.8	Goto Position 7	Bit 6 of the command number
X6.21	Goto Position 8	Bit 7 of the command number

The command is going to be started when a rising edge on the extra trigger input is detected:

Pin	Name	Function
X6.15	Grp0/Trig	Starts the selected command

C) 8 Bit Code for Command Number without Trigger Input:

The command to be executed is selected by its binary coded command number:

Pin	Name	Function
X6.5	Goto Position 1	Bit 0 of the command number
X6.18	Goto Position 2	Bit 1 of the command number
X6.6	Goto Position 3	Bit 2 of the command number
X6.19	Goto Position 4	Bit 3 of the command number
X6.7	Goto Position 5	Bit 4 of the command number
X6.20	Goto Position 6	Bit 5 of the command number
X6.8	Goto Position 7	Bit 6 of the command number
X6.21	Goto Position 8	Bit 7 of the command number

The command starts automatically: When an input pattern (=command number) has remained stable for the jitter filter time, then the corresponding command is going to be started (without any other trigger signals).

Digital Input Filters

Depending on the *Command Selection Mode* other kinds of software filters are applied on the digital input:

A) 5 Bit Group Selection, 8 Trigger Inputs:

The 8 command trigger inputs are each equipped with a digital debounce filter. After a signal edge is detected on an input, the corresponding input becomes locked for the duration of the adjustable *Trigger Debounce Time*. This filter prevents multiple initiations of a command even if the input signal is chattering (important for *Relative Move* commands).

If the selection of the command group and activation of the trigger signal has to be realized in the same PLC cycle, then the trigger input signals may be additionally retarded (parameter *Trigger retardation Time*). This is to assure that group selection is active before a command is executed.

B) 8 Bit Code for Command Number with extra Trigger Input:

On the trigger input (X6.15) there is a digital debounce filter. After a signal edge is detected on this input, the input becomes locked for the duration of the *Trigger Debounce Time*. This filter prevents multiple initiation of a command even if the input signal is chattering (important for *Relative Move* commands).

If the command selection (8 Bit Code) and the triggering have to be realized in the same PLC cycle, then the trigger signal may be additionally retarded. This is to assure that command selection is active before a command is executed.

C) 8 Bit Code for Command Number without Trigger Input:

In this mode commands are selected and started when the command number pattern has remained stable for the adjustable jitter filter time on the 8 command selection inputs. The Jitter-Filter protects against unintentional execution of commands due to jitter effects between the command inputs.

Note: The debounce filters on the trigger inputs don't lead to any delays for start of any command. The command is going to be started just after the first positive edge on the trigger input. But then the input is not sampled by software for the adjustable *Debounce Filter Time*.

In contrast to the debounce filter, the jitter filter leads always to a delayed execution of commands. If the jitter filter is in use (no trigger inputs), then the selected command will not be executed before the input pattern has remained stable for the *Jitter Filter Time*. So after selection of the command we have a minimal delay given by the value of the parameter *Jitter Filter Time*.

Command Executed Feedback

On the connector X6 there are eight outputs, which show, whether a command has successfully executed (*In Position 1 .. In Position 8*).

If an MP controller is used, the mapping is fix. For each command there is a dedicated output. The output is activated when a command has successfully executed. As soon as a new command is started, all outputs on the X6 connector are cleared.

In the setup window for the IO-Interface of an MT Controller the user can choose between different command executed feedback modes:

A) 8 Outputs (default):

In this mode all the outputs on X6 are in use. On the outputs it is shown, which command within the selected group has been executed (similar to the MP controllers where only group 0 is existing).

Pin	Name	Function
X6.9	In Position 1	1 st command of the sel. group executed
X6.22	In Position 2	2 nd command of the sel. group executed
X6.10	In Position 3	3 rd command of the sel. group executed
X6.23	In Position 4	4 th command of the sel. group executed
X6.11	In Position 5	5 th command of the sel. group executed
X6.24	In Position 6	6 th command of the sel. group executed
X6.12	In Position 7	7 th command of the sel. group executed
X6.25	In Position 8	8 th command of the sel. group executed

E.g. when command no. 39 has been executed, then the output *In Position 7* is activated (command no. 39 is the 7th command within group 4).

B) Fix programmed output:

In the fix programmed output mode the same output is used for all commands. Which output has to be used can be chosen in the IO Interface Setup window.

C) Binary coded command number:

The ID of the executed command is set to the eight outputs as binary code:

Pin	Name	Function
X6.9	In Position 1	Bit 0 of the executed command
X6.22	In Position 2	Bit 1 of the executed command
X6.10	In Position 3	Bit 2 of the executed command
X6.23	In Position 4	Bit 3 of the executed command
X6.11	In Position 5	Bit 4 of the executed command
X6.24	In Position 6	Bit 5 of the executed command
X6.12	In Position 7	Bit 6 of the executed command
X6.25	In Position 8	Bit 7 of the executed command

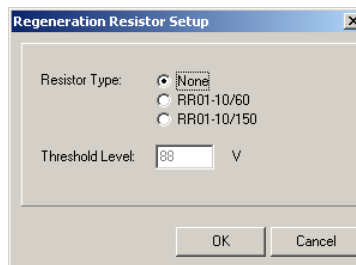
Note: Beside the outputs *In Position 1 .. In Position 8* on the X6 connector, there is another *In Position* output on the front side connector X4 (pin X4.4). This output also becomes active, when a command has been executed, and becomes cleared, when a new command is started (see chapter “Operating state dependent output signals”). The only difference is the feedback of the *Press* command (see chapter “MPC Commands”). In most cases only the *In Position* output X4.4 is connected with the superior controller and used by the sequential program there. The outputs from socket X6 are mainly used for monitoring purposes.

3.3 Regeneration Resistor

When big load masses have to be decelerated by the linear motor, then kinetic energy is converted back to electrical energy (generator effect). The energy is then stored in the DC power bus where the voltage rises. It can happen that this DC level will exceed 90V, which leads to a transition to error state (error message: “Motor supply voltage too high”).

For some application it is therefore necessary to dissipate this energy in a regenerative resistor. The resistor has to be connected to pin X1.3 (RR+) and X1.4 (RR-).

If a regeneration resistor is used, it has to be configured with the MPC Configurator. This is done via the menu item *Commissioning -> Regeneration Resistor*.



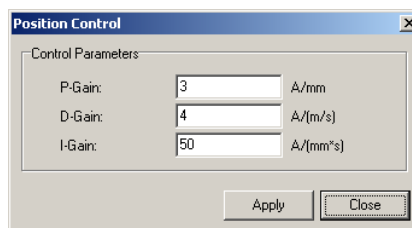
In the setup window the *Resistor Type* and the *Threshold Level* have to be defined. When the DC link voltage rises over the *Threshold Level*, the controller begins to dump power to the regenerative resistor.

3.4 Position Control Loop

The motor's position is controlled through a digital PID controller. The advanced user may optimize the application with these control parameters. So even under difficult circumstances an optimal control performance can be reached. We strongly suggest that care is taken when altering the control parameters. With a bad controller setup the system can begin to oscillate and may harm the plant!

It would exceed the range of this document to explain the principles of PID controllers. Information about the shape and function of PID controllers can be found in the corresponding technical literature.

Via the menu entry *Commissioning* -> *Position Control* the user gets access to the gain parameters of the PID controller.



Some empirical hints for the controller setup:

P-Gain: The greater the *P-Gain* is set, the better the *Actual Position* follows *the Demand Position* value. But with overlarge values, the system tends to oscillations.

D-Gain: The numerical value should be set to a value between the 1.5 fold and the 4 fold of the *P-Gain* value. The greater the value is, the more oscillations are damped, but the more noise from the position sensors is amplified.

I-Gain: With the I-gain the steady state position deviation can be eliminated. For high dynamic application the *I-Gain* often has negative impact to the controller performance. For such applications it is therefore recommended to set the *I-Gain* value to zero. If an I-Gain is needed, then it is advisable to have a value of at least 30 A/(mm*s).

The controller gains can't be optimized independently. They should always be considered as a parameter set.

There isn't any universal optimal controller setup. The application specific optimum is defined by weighting the different affects that can be influenced with the PID controller setup, e.g. stiffness, overshootings, transient time, position lag, steady state position deviation, noise emission, etc..

The values given in the following tables have to be understood as guideline values:

Typical setup for high-dynamic applications:

Controller	soft	middle	Stiff
P-Gain	1	2	4
D-Gain	2	4	10
I-Gain	0	0	0

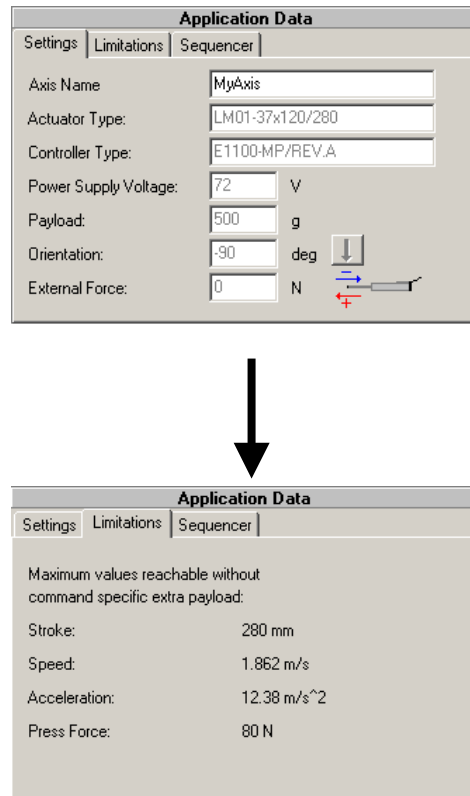
Typical setup for slower applications:

Controller	soft	middle	Stiff
P-Gain	1	2	4
D-Gain	2	4	10
I-Gain	50	50	100

Note: The motor current setpoint value is calculated as the sum of the position control loop output and some feed forward values. The feed forward values are automatically calculated by the controller based on the motor and application data. The position control performance is therefore affected by those values. The control performance can be improved by setting the application data according to reality (actuator type, power supply voltage, orientation and load mass)!

3.5 Application Settings

Based on the application data settings the controller calculates maximal values for stroke, velocity, acceleration and press force. They therefore affect directly the motion profiles and the control behavior of the motor. The more accurate the data settings reflect reality, the better the control performance.



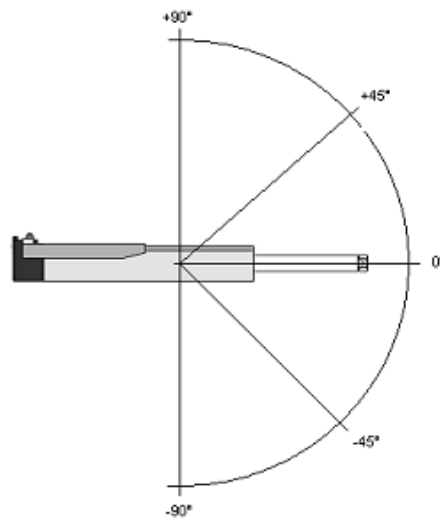
Actuator Type: Name of the currently used actuator. It isn't possible to edit the actuator's name. If another actuator has to be used, then it has to be changed via the menu item *Commissioning -> Actuator Change*. The actuator which fits best to the application requirements can be determined with the sizing tool *LinMot-Designer*.

Controller Type: This information is read out of the controller and can not be changed. With HC (High Current) versions of MP and MT controllers higher forces and therefore shorter positioning times can be reached.

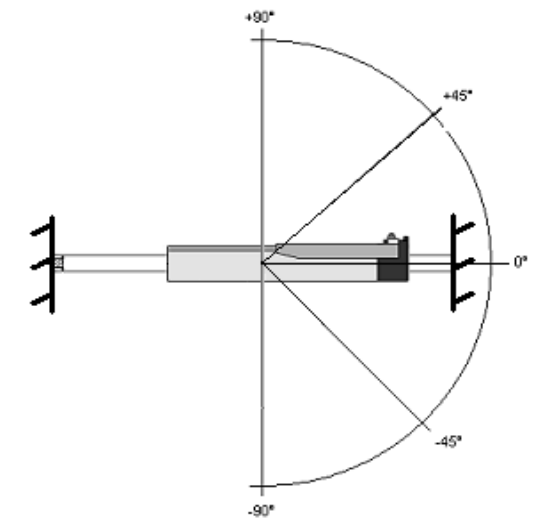
Power Supply Voltage: Supply voltage for the motors. Controllers and motors are designed for 72V power supply. Lowering the supply voltage lowers the system's velocity, acceleration and press capabilities.

Payload: Here the moved load mass, which is attached to the actuator, has to be defined. If the actuator is a predefined linear module (e.g. LM01-...), then the mass of the guidance must **not** be added here (it is already part of the actuator definition file). If the actuator is a bare linear motor (P01-...), then the whole moving mass (including mass of the flange, guidance etc.) has to be entered here, but without the moved motor mass (stator or slider mass). The motor mass value is part of the actuator definition file and therefore automatically added.

Orientation: The mounting angle is defined according to the following drawings:

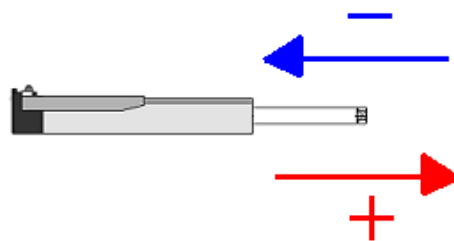


Moving Slider

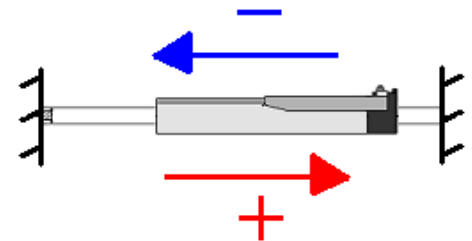


Moving Stator

External Force: It often makes sense to compensate continuous external forces with external passive elements (e.g. gravitation forces with MagSprings, see www.magspring.com). The force, which is applied by such elements has to be entered with correct sign:



Moving Slider



Moving Stator

4 MPC Commands

The motion commands are defined in the command table of the MPC Configurator. The following command types are implemented:

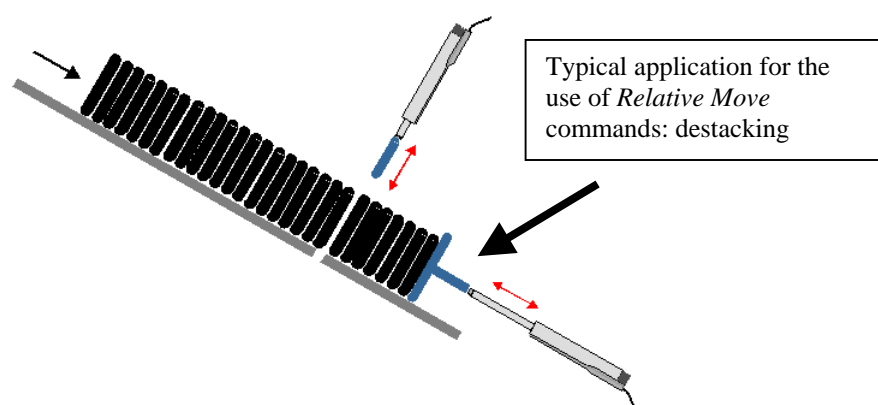
Befehlstyp	Aktion
Absolute Move	Move to an absolute defined target position
Relative Move	Incremental move
Press	Move with limited force

New commands can be entered into the command table when the control mode is set to *PC Control/Config Mode* and the *Enable* flag is cleared.

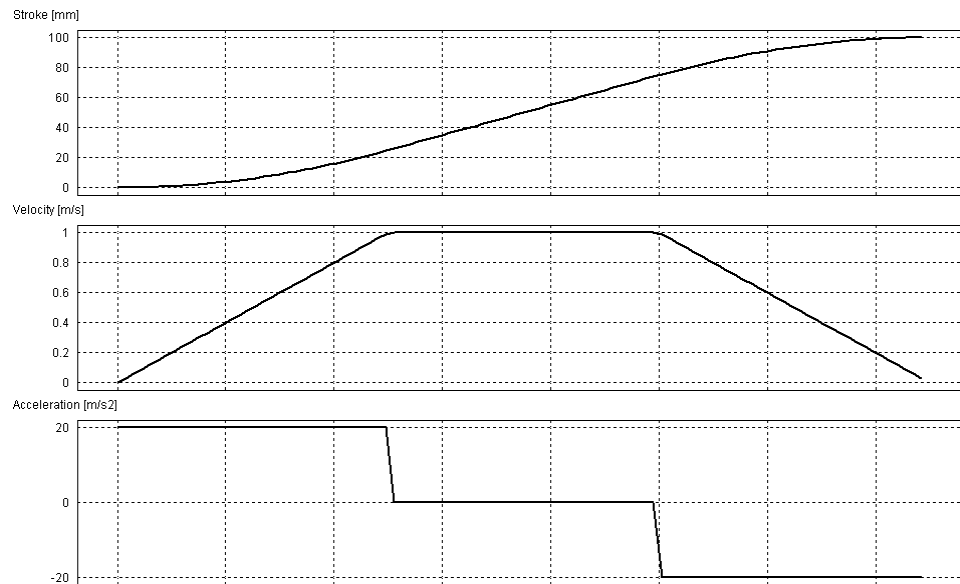
The command's parameters *Position/Increment*, *Speed* and *Press Force* can be altered even in run time (they are considered the next time the command is called).

4.1 Absolute / Relative Move

Positioning tasks are solved with the command types *Absolute Move* and *Relative Move*. The difference between these two commands is that for an *Absolute Move* command the target position is given as a distance from the 0-point, whereas for a *Relative Move* command a value is given as a distance from the location where the command is received. When a *Relative Move* command is called, the new target position is calculated by adding the command's increment to the current target position value.



When a positioning command is called, the *Demand Position* value does not “jump” to the new *Target Position* value. With the given load data and the speed value the controller calculates a demand position profile between the former and the new *Target Position* (Point-to-Point interpolation with limited acceleration, velocity and deceleration).



The motion’s speed has to be defined as a percent value. 100% corresponds to the shortest possible positioning time, which is internally calculated based on motor and application data. If a certain stroke is realized with 20% speed, then 5x more time is required for doing the movement, than if the same stroke would be done with maximal (100%) speed.

When defining the motion speed, the following basic principle should be considered: **Motions should be defined as fast as necessary, but as slow as the application allows.** Faster movements produce a disproportional requirement of power dissipation. This power is dissipated as heat back to motor and controller. Also the position control performance tends to be better for slower motions (no overshoots).

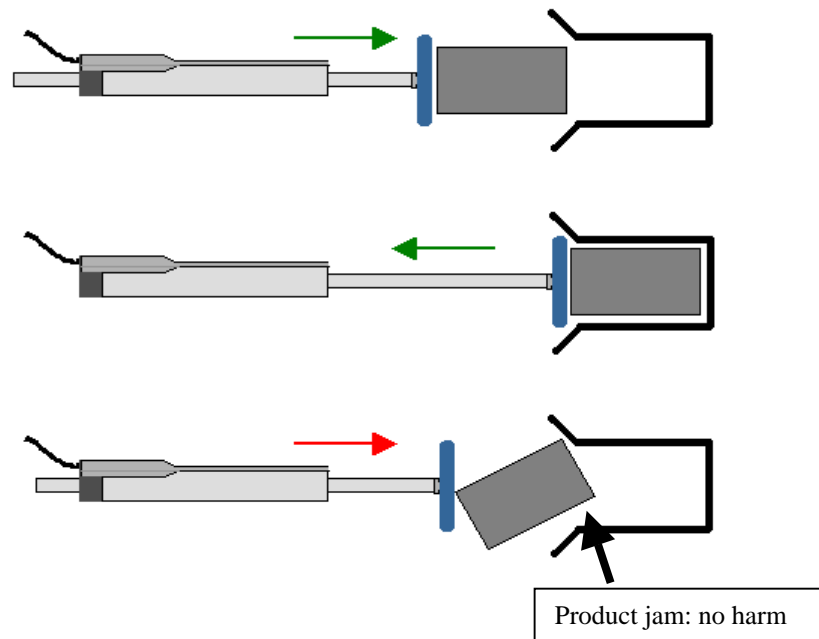
The execution of *Relative Move* commands, which would exceed the MPC work stroke, is denied at run time. By generating a warning the controller signals, that the command will not be executed.

4.2 Press

By defining a *Press* command a target position value has to be entered, similar to *Absolute Move* commands. For *Press* commands an additional **force limit** has to be defined. When a *Press* command is called, the motor moves towards the command's target position, whereby the motor never applies more than the given *Press Force* (motor current limitation).

When the motor has reached the target position, the *In Position* output on the front side socket X4 is set (pin X4.4). When the motor is applying the given *Press Force* the command specific *In Position* output on the socket X6 is active. The output remains active as long as motor is applying *Press Force* and is cleared otherwise.

Typically the *Press* command is used for pushing parts into others (see the following application example).

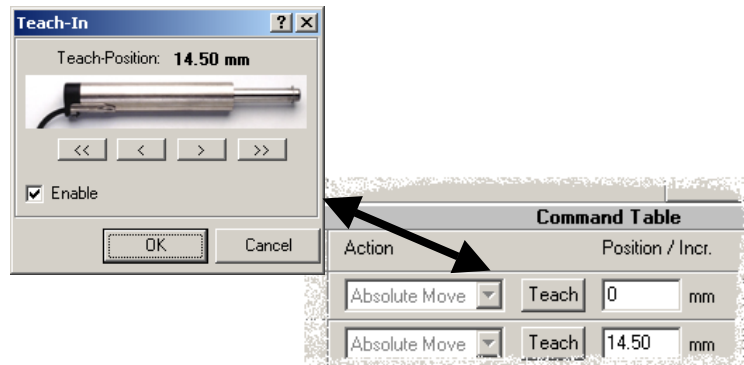


Press application example: A product has to be pushed into its packaging unit. For the push motion the *Press* command is used. If it is possible to push the entire product into the packaging, then at the end of the movement the output X4.4 is set. Because the motor's force is limited with the parameter *Press Force*, it is possible to detect if the push process failed. When the motor has to apply the maximal allowed force (product jam), the corresponding output on the socket X6 is activated. The overlaid process controller can react to this signal (e.g. by ejecting product and/or packaging unit).

4.3 Teach Function

The target position of *Absolute Move* and *Press* commands can be defined by entering the corresponding number on the MPC screen or by using the *Teach* function.

The Teach function is available when the control is set to *PC Control/Config. Mode* and the motor must have been previously homed.

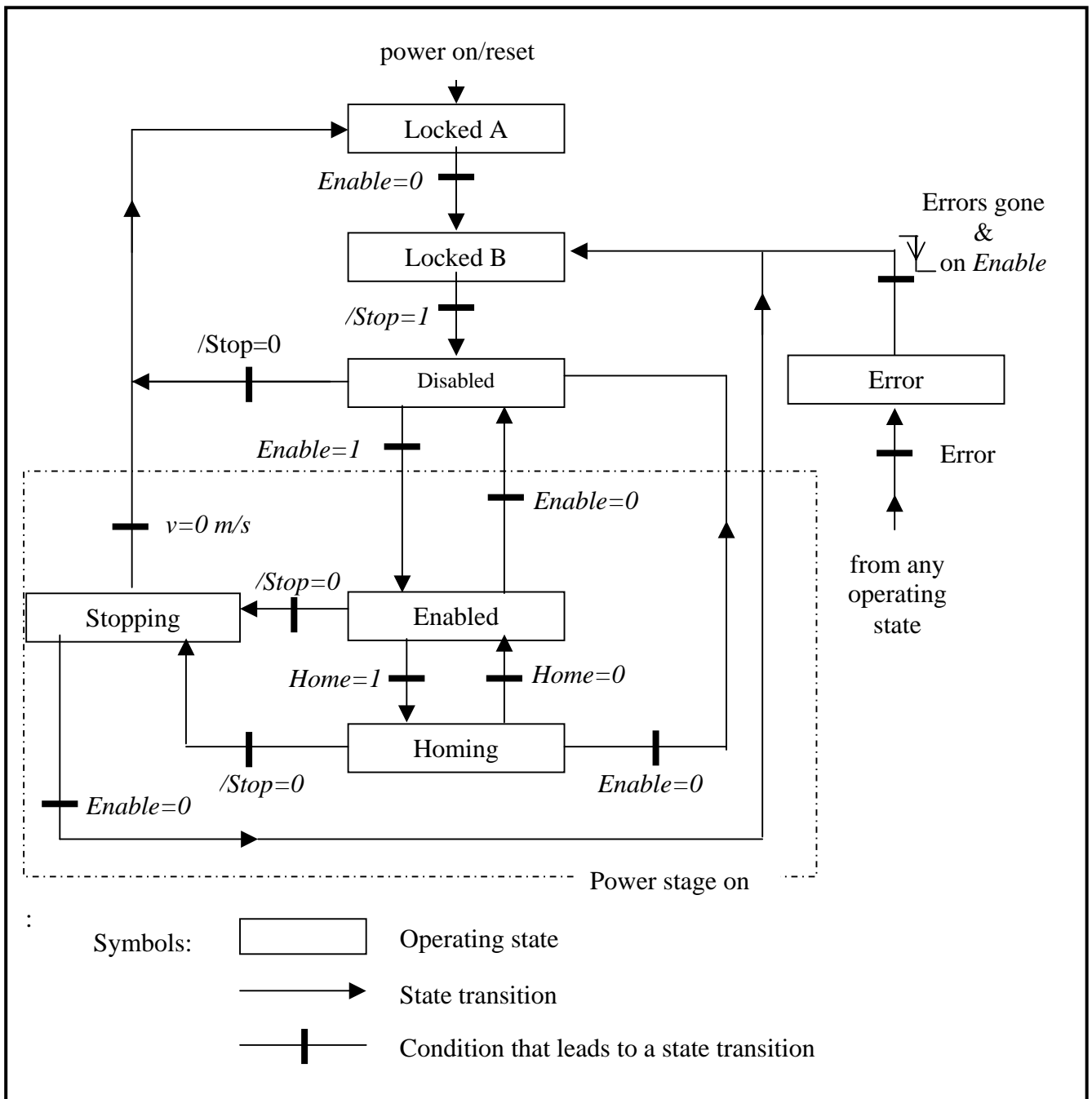


When the *Enable* signal is set, the motor can be moved by clicking the arrow buttons (jog move mode). In *Disabled* state the motor can manually be brought to the desired target position. By pressing the OK button, the current teach position is copied to the command table.

5 MPC System Control

5.1 Operating State Diagram

From power on until commands can be called, the controller runs through different operating states. The state transitions are released by control signals or error conditions (e.g. “Motor supply voltage to low”).



The current operating state is displayed on the status panel of the MPC Configurator:

Status	
Operating State:	Enabled
Motor Homed:	True
Actual Command:	2
In Target Pos:	True
Target Position:	0.00 mm
Demand Position:	0.00 mm
Actual Position:	0.00 mm

5.2 Operating State Control Signals

The three control signals *Enable*, *Home* and */Stop* can be generated by different instances. During the commissioning process (*PC Control/Config Mode* is activated) the control signals are set via the software check boxes on the control panel. If *External Control* is activated, the control signals are given by the corresponding digital inputs on the socket X6. If one or more of those inputs were disabled during commissioning, then the corresponding state transition request flags are set automatically (see chapter ‘MPC Configurator Setup: IO Interface’).

The control signals do have the following functions:

Enable (Input X6.1): The *Enable* signal is used for switching between *Enabled* and *Disabled* state. For safety reasons after power up the signal level has to be set to logic 0 (to leave state *Locked A*) before the motor can be enabled afterwards. The *Enable* signal is also used to acknowledge the Error state (see chapter ‘Error Acknowledge’).

Home (X6.14): This signal enables the homing procedure. The *Home* signal has to be set until the homing procedure has finished (*In Position* output becomes active at the end, see below). Afterwards the signal can be reset. The controller comes back to *Enabled* state and is then ready to execute the motion commands defined in the command table.

/Stop (X6.2): The */Stop* signal is used for the emergency stop functionality. Because it is an inverted signal (‘/’ = ‘not’), the input has to be on logic level 1 for normal operation. After power up the controller remains in the state *Locked B* until the control signal */Stop* is set.

5.3 Operating State Dependent Outputs

On the front side socket X4 there are four outputs. Their logic level depends on the current operating state:

/Brake (output X4.3): This output is designated for direct connection to a mechanical parking brake. The output is set to logic level 1 (brake released), when the motor is active controlled by the MPC Controller. This is the case in the operating states *Enabled*, *Homing* and *Stopping*.

In Position (X4.4): The *In Position* output can only become active in the operating states *Enabled* and *Homing*. It is set, when the homing procedure or any positioning command has been executed successfully (*Actual Position* is within *In Target Position Range*). The output is cleared, when a new command is called or when external forces push the motor out of its *In Target Position Range*.

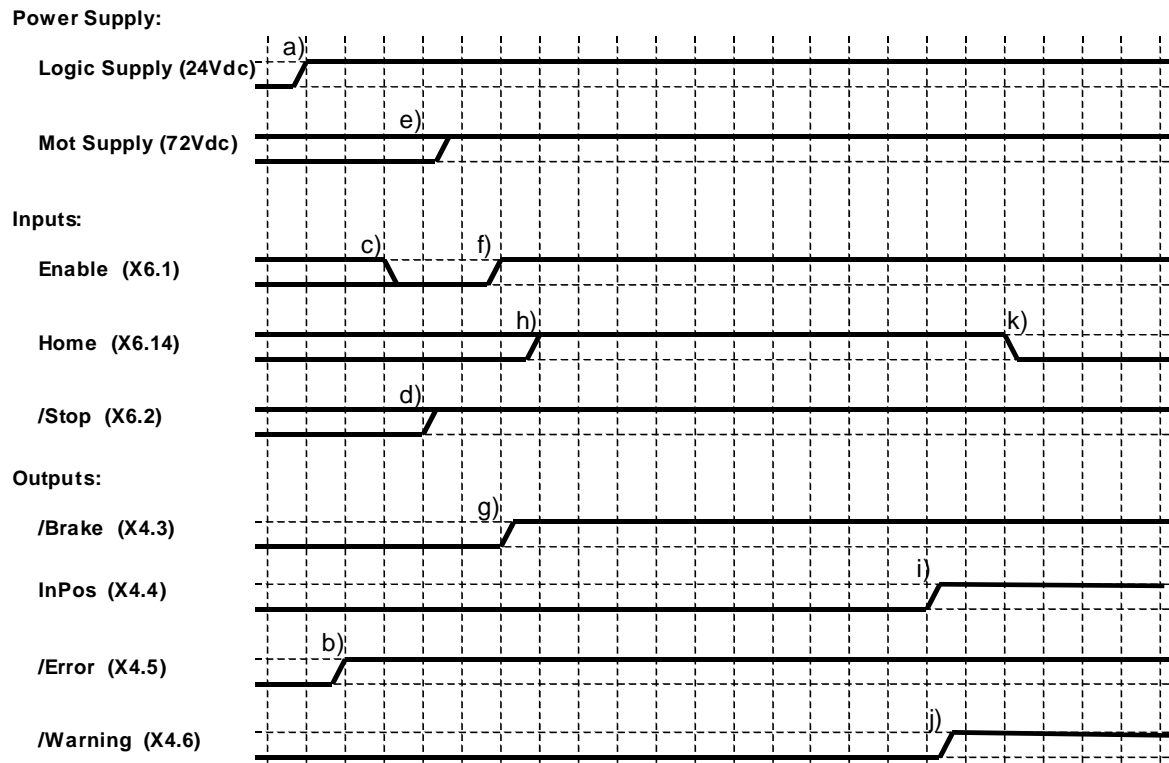
/Error (X4.5): This output shows, whether the controller is in operating state *Error*. Since it is an inverted output (prefix '/'), the output is set to logic level 1, when the controller is **not** in *Error* state.

/Warning (X4.6): The controller software monitors some relevant operation parameters (e.g. motor and controller temperature, DC link voltage, position lag, etc.). If one of these parameters has reached or exceeded a critical value, the overlaid controller is notified via the */Warning* output. Again, it is an inverted signal and therefore the output is **set**, when there is **no** warning pending. The output cleared as soon all operating parameters are in a safe range again. Warnings don't have to be acknowledged from the superior control system (in contrast to errors).

Note: The fact 'motor not homed' is also treated as a warning condition. Therefore the */Warning* output can only become activated after the homing procedure has been successfully completed.

5.4 Control Signal Sequence for Enabling and Homing the Motor

The following figure shows an example sequence for enabling and referencing the motor:



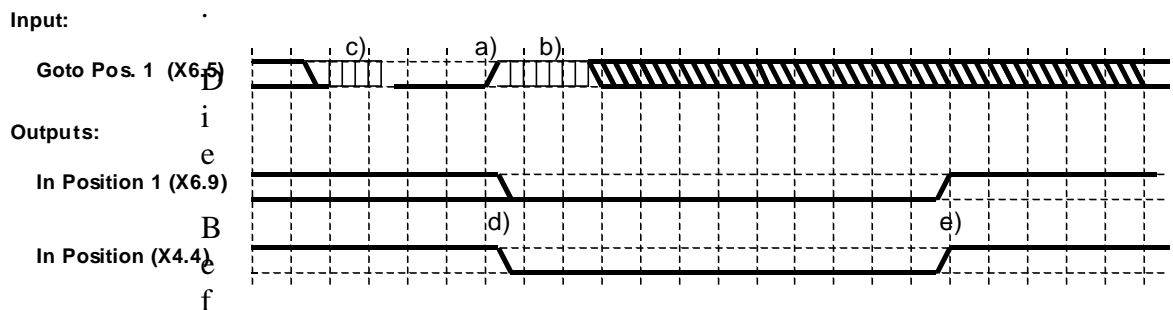
Through selective setting and resetting of the control signals the state transitions of the controller's operating state machine can be controlled:

- The controller software starts working, when the 24Vdc logic supply is turned on.
- When the controller firmware has started and no error condition has been detected, the */Error* output is set.
- By clearing the *Enable* signal the operating state *Locked A* can be left. The *Enable* signal may have been cleared already at power on.
- Through setting the */Stop* signal (inverted emergency stop input), the state *Locked B* can be left and the controller changes to the *Disabled* state. The */Stop* signal may have been set before or with the 24Vdc logic supply.
- Before the transition to the *Enabled* state, the motor supply voltage (mostly 72Vdc) has to be applied. The motor supply can be turned on together with the 24Vdc logic supply.
- By setting the *Enable* signal the *Disabled* state is left and the *Enabled* state entered. The motor is powered on.

- g) As soon as the controller is in *Enabled* state, the */Brake* output is turned on and thus the brake released (if there's a brake connected).
- h) When in operating state *Enabled* the *Home* input is set, the controller changes to the *Homing* state. The motor performs its homing procedure.
- i) At the end of the homing procedure the *In Position* output becomes active.
- j) If there are no further warnings pending, the */Warning* output is set to logic level 1 after homing. The warning 'motor not homed' is not active anymore.
- k) After successfully finishing of the homing procedure the *Home* input can be cleared. The controller comes back to *Enabled* state and is now ready for executing MPC motion commands.

5.5 Command Call and Feedback with E1100-MP Controller

For each of the eight programmable commands on an E1100-MP controller, there is a designated input (*Goto Position 1 .. Goto Position 8*) and a corresponding output (*In Position 1 .. In Position 8*). The commands have to be started through rising edges on the corresponding inputs. When the command has successfully executed, the appropriate output becomes active (the *Press* command is handled specially, see command description). Execution of commands is possible in *Enabled* state only. The motor must be homed before.



- a) The rising edge on the input *Goto Position 1* starts execution of command no. 1.
- b) After detection of the rising edge, the controller firmware does not read out the state of the respective command trigger input for the definable *Debounce Filter Time*.
- c) After detection of a falling edge on a *Goto Position* input, the input is not sampled for the duration of the *Debounce Filter Time*. The next positive edge can not be detected before the filter has timed out.
- d) As soon a new command has started, all *In Position* outputs are cleared.
- e) When the *Actual Position* reaches the *Target Position*, the *In Position* output X4.4 becomes active. If the respective command is of type *Absolute Move* or *Relative Move*, the command specific *In Position* outputs on X6 is activated at the same time. If the

command is of type *Press*, the command specific output is active, when the motor applies the defined press force.

The commands are started through rising edges. A signal pulse is therefore sufficient to call a command. The pulse length has to be at least 0.5ms.

New commands can be called, before the former command has finished. The new command starts immediatly (no queuing).

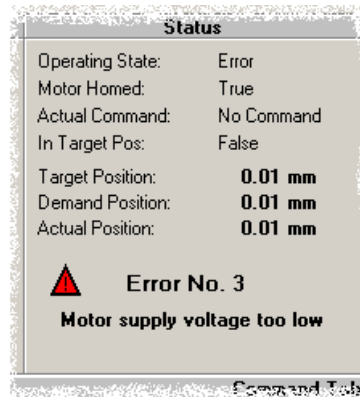
The same *Relative Move* command often has to be executed several times consecutively (e.g. destacking applications). Between two calls a minimal delay time of twice the *Debounce Filter Time* is necessary then. So it is guaranteed, that all commands are executed (without any further time lag). Since by the execution of *Relative Move* commands the current *Target Position* is incremented (not the *Actual Position* value!), no drift phenomenas can occur, even after thousands of *Relative Move* commands.

5.6 Command Call and Feedback with E1100-MT Controller

When an MT-type controller is used, the way commands are called and the kind of the feedback depend on the configuration setting. The corresponding explanations can be found in the chapter 'MPC Configurator Settings: IO Interface'. With the default setup the behavior of an MT controller is exactly the same as on a MP-type controller.

5.7 Error Acknowledge

When the controller is in *Error* state, the output */Error* is cleared, which can be detected by the overlaid control system (PLC, IPC). The reason for the transition to *Error* state is displayed as clear text message on the status panel of the MPC Configurator.



With a **falling edge on the *Enable* signal** an attempt to leave the *Error* state is made. If there are no further error conditions pending when the falling edge is detected, the *Error* state can be left. If the reason which let the controller switch to *Error* state is still effective, or a new error became effective, the *Error* state can not be left. If the error acknowledge try was successful, the */Error* output is set active again.

5.8 Emergency Stop Behavior

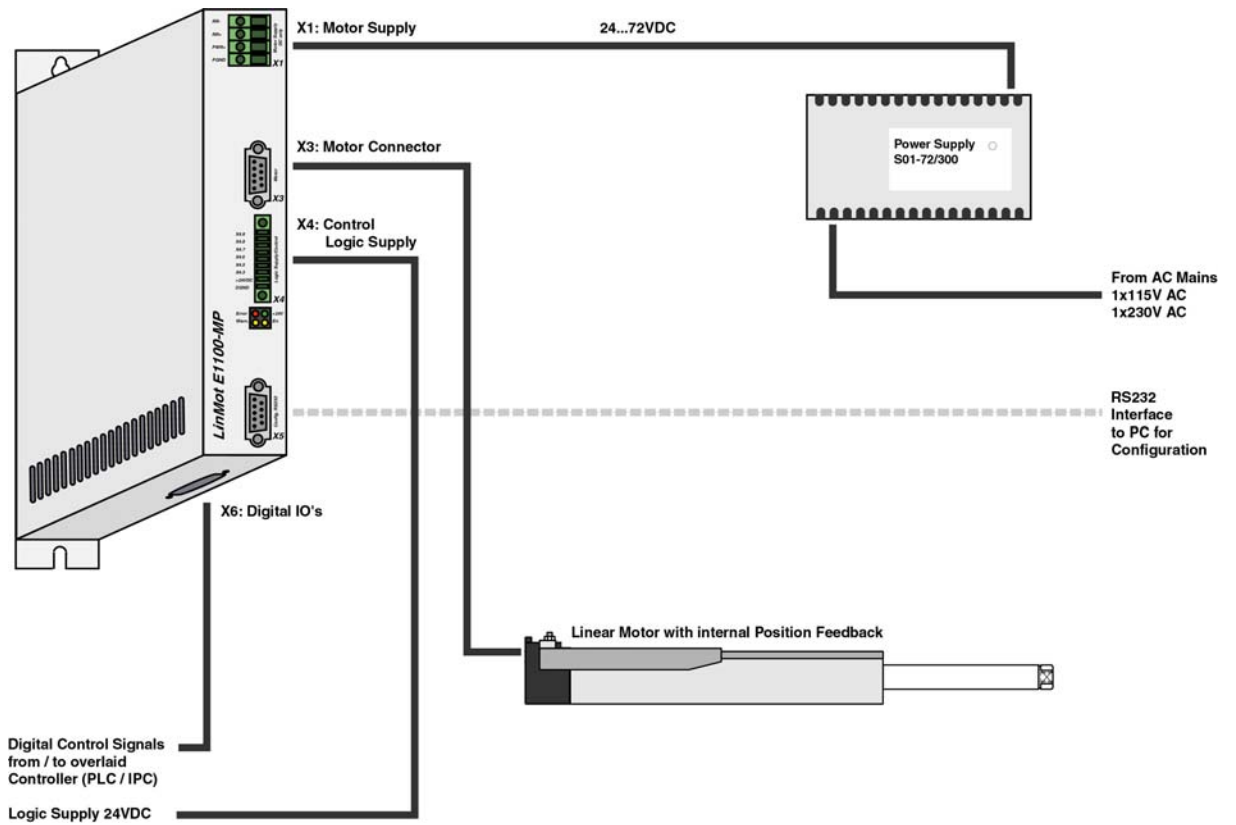
Clearing the */Stop* input signal initiates the emergency stop behavior. If the motor was enabled when the */Stop* signal disappeared (operating state *Enabled* or *Homing*), the motor decelerates until the actual velocity is zero (operating state *Stopping*). Afterwards the power stage will be switched off and controller changes to operating state *Locked A*. The operating state *Locked A* can be left by clearing the *Enable* signal (acknowledge of the emergency stop). Also in the emergency stop case, the *Enable* signal has the control over the power stage. Clearing the *Enable* signal means immediately turning off the power stage. If an active deceleration of the motor is required, the *Enable* signal has to remain active until the motor has stopped (operating state *Stopping*, see state machine diagram).

A. MPC Controller Types

As part of the MPC system a MPC Controller is needed. The different controller types differ in their power stage (maximal motor current) and their functionality:

Controller	E1100-MP	E1100-MP-HC	E1100-MT	E1100-MT-HC
No. of commands	8	8	256	256
Max. motor current	4 A	15 A	4 A	15 A
Motor supply voltage	72 V	72 V	72 V	72V
Ext. position sensor option	No	No	Yes	Yes
Command type <i>Absolute Move</i>	Yes	Yes	Yes	Yes
Command type <i>Relative Move</i>	Yes	Yes	Yes	Yes
Command type <i>Press</i>	Yes	Yes	Yes	Yes

B. Connector Location on the MPC Controller



Detailed informations about the controller hardware can be found in the corresponding installation guide. The installation guide is delivered with the controller and can be downloaded from www.LinMot.com.

Notes:

Smart solutions are...



NTI AG
LinMot
Haerdlistrasse 15
CH-8957 Spreitenbach
Switzerland

Tel.: +41 (0)56 419 91 91
Fax: +41 (0)56 419 91 92
office@LinMot.com
www.LinMot.com

LinMot Inc.
N2444 Broad Street
Delavan
WI 53115
USA

Phone: +1-877-546-3270
+1-262-728-22699
Fax: +1-800-463-8708
officeUS@LinMot.com
www.LinMot.com

LinMot® products are available from more than 80 Distributors worldwide.
For the distribution nearest you, visit <http://www.LinMot.com>

