

Easy Motion

Programming Manual

Retain for future use



Table of Contents

| | |
|--|-----|
| Important information | 4 |
| Documentation structure | 5 |
| Setup - Preliminary recommendations | 6 |
| Launch Easy Motion | 7 |
| General organization of the screen | 8 |
| Organization of the screen | 8 |
| Configuring the devices | 9 |
| Configuring the controller | 10 |
| Configuring the axes | 10 |
| Manual mode | 15 |
| General organization of the screen | 15 |
| Central supervision screen | 16 |
| Toolbar | 17 |
| Teaching positions - "Teach" button | 18 |
| Motion task editor | 19 |
| General organization of the screen | 19 |
| Toolbar | 19 |
| List of accessible function blocks | 20 |
| Function block editing area | 20 |
| The motion sequence | 21 |
| Storing and restoring data | 21 |
| Function blocks | 22 |
| Administrative commands | 43 |
| Multi-axis positioning | 45 |
| Application function blocks | 52 |
| Logic commands | 58 |
| Execution rules | 66 |
| Automatic mode | 80 |
| Central supervision screen | 81 |
| Control panel | 81 |
| Motion sequence | 82 |
| Toolbar | 82 |
| "Run Time" supervision window | 83 |
| Trace function | 84 |
| Mains contactor operation | 85 |
| Scaling | 87 |
| Error management | 88 |
| Application Template parameter table | 91 |
| Axis module interface | 96 |
| Interface Master encoder | 99 |
| Axis parameters | 100 |
| Master encoder parameters | 101 |
| Servo drives configuration | 101 |
| Teach Position Table | 102 |
| Active Errors parameters list | 102 |
| Logged Errors parameters list | 103 |
| Cam Profile parameters | 104 |
| Motion Task table parameters | 105 |
| Flying Shear and Rotary Knife parameters | 107 |

While every precaution has been taken in the preparation of this document, Schneider Electric SA assumes no liability for any omissions or errors it may contain, nor for any damages resulting from the application or use of the information herein.

The products and options described in this document may be changed or modified at any time, either from a technical point of view or in the way they are operated. Their description can in no way be considered contractual.

Important information

PLEASE NOTE

Please read these instructions carefully and examine the equipment in order to familiarize yourself with the device before installing, operating or carrying out any maintenance work on it.

The following special messages that you will come across in this document or on the device are designed to warn you about potential risks or draw your attention to information that will clarify or simplify a procedure.



The addition of this symbol to a "Danger" or "Warning" safety label indicates that there is an electrical risk that will result in injury if the instructions are not followed.



This is a safety warning symbol. It warns you of the potential risk of injury. You must comply with all safety messages that follow this symbol in order to avoid the risk of injury or death.

DANGER

DANGER indicates an imminently hazardous situation which, if not avoided, **will result in** death, serious injury or equipment damage.

WARNING

WARNING indicates a potentially hazardous situation which, if not avoided, **can result in** death, serious injury or equipment damage.

CAUTION

CAUTION indicates a potentially hazardous situation which, if not avoided, **can result in** injury or equipment damage.

PLEASE NOTE:

Only qualified staff are authorized to carry out maintenance work on electrical equipment. Schneider Electric accepts no responsibility for the consequences of using this device. This document does not constitute an instruction manual for inexperienced people.
© 2008 Schneider Electric. All rights reserved.

Documentation structure

Installation Manual

This manual describes:

- How to install the controller
- How to connect the controller

Optional Keypad User's Manual

This manual describes:

- How to install the keypad
- How to connect the keypad
- How to program the controller via the keypad

Easy Motion - Programming Manual

Supplied preinstalled in the Lexium Controller, the application model associated with Easy Motion mode is a user-friendly tool that can be used for:

- Rapid axis configuration
- Use of Manual/Automatic mode
- Creating positioning tasks
- Editing cam profiles
- Backup and recovery of the machine parameters
- Diagnostics of the motion controller and the various axes

This programming manual also contains a table of the parameters that can be accessed via the communication protocols.

Motion Pro - Programming Manual

The Motion Pro Programming Manual is included in the software online help.

This online help describes:

- The software interface
- IEC 1131 programming
- The function libraries (standard functions, motion control functions, application functions)
- The Lexium Controller configuration screens

Modbus, Ethernet, Profibus DP, and DeviceNet manuals

These manuals describe:

- Connection to the bus or network
- Diagnostics
- Software setup
- The protocol communication services

Setup - Preliminary recommendations

DANGER

UNINTENDED EQUIPMENT OPERATION

When using Easy Motion/Application Template:

- check that all movements started are not presenting any risks.
- when the KeyPad is plugged to the Lexium Controller a message appears: "**KeyPad active simultaneous action can be dangerous**". In this case, use of Easy Motion/Application Template can be dangerous for the KeyPad user.

If Easy Motion is unplugged from the Lexium Controller, current movements (Jog included) are not interrupted.

Failure to follow these instructions can result in death or serious injury.

DANGER

UNINTENDED EQUIPMENT OPERATION

Current versions of Easy Motion, Application Template and Key Pad are not compatible with Lexium 15 and SD328. Utilisation of Easy Motion, Application Template and Key Pad is limited to Lexium 05.

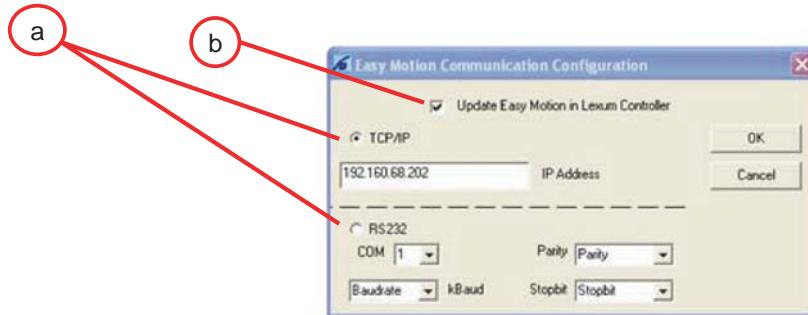
Failure to follow these instructions can result in death or serious injury.

Launch Easy Motion

1 Click on Easy Motion shortcut



2 The following window appears:



a) Connection parameters:

- Choose between TCP/IP or RS232 connection,
- Choose the parameters,
- Click "**OK**".

b)"Update Easy Motion in Lexium Controller": this command has to be selected if the program has changed in the Lexium Controller. It will erase the full memory, download the Application Template and create a boot project. This operation can take several minutes.

Configuring the devices

The parameters required for configuration can be found in the "Configuration mode" screen which can be accessed at any time by clicking the "Config" button located on the left of the Easy Motion graphic interface. Click "Conf Enable" then "Edit Par" to access selection of the various devices.

| Configuration Mode | | V00.01.00 | | | | | | | |
|--------------------|-------------------------------|-----------|------------|----------|-----------|------------|--|--|--|
| Conf Enable | Upload | Download | Edit Par | Read Par | Write Par | Save Drive | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| Editor | Parameter of Axis: 1 | | | | | | | | |
| Auto | Axis Name: 1 | | | | | | | | |
| Manual | Rotary/Linear: 1 | | | | | | | | |
| Config | Modulo: 360 | | | | | | | | |
| | SoftLimit Enable: FALSE | | | | | | | | |
| | SoftLimit Pos: 0 | | | | | | | | |
| | SoftLimit Neg: 0 | | | | | | | | |
| | Ramp Type: trapez | | | | | | | | |
| | User Unit Numerator: 45 | | | | | | | | |
| | User Unit Denominator : 16384 | | | | | | | | |
| | Homing Mode: 23 | | | | | | | | |
| | Homing Speed: 0 | | | | | | | | |
| | KPn: 0 | | | | | | | | |
| Axis Par | TNn: 0 | | | | | | | | |
| | KPp: 0 | | | | | | | | |
| EncoderPar | VelFeedFor: 0 | | | | | | | | |
| | Drive type: 0 | | | | | | | | |
| LMC Par | Firmware: 0 | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| Error Reset | ErrorId: 0 | | Error Msg: | | | | | | |

When the "Configuration Mode" screen is opened, if no modifications have been made to the parameters for the time being, it only shows the general navigation bar (on the left of the screen) and the toolbar specific to this screen. This bar consists of 3 main buttons (Upload, Download and Edit Par) and three masked buttons (Read Par, Write Par and Save Par). The masked buttons only appear if the Edit Par button is selected.

Description of the functions connected with the buttons:

| | | | | | | |
|-------------|--------|----------|----------|----------|-----------|------------|
| Conf Enable | Upload | Download | Edit Par | Read Par | Write Par | Save Drive |
|-------------|--------|----------|----------|----------|-----------|------------|

- **Upload:** Used to read each of the configuration parameters of the servo drives connected to the Lexium Controller on the Motion bus. These parameters are then saved to the controller's non-volatile memory. This function is used to save a given configuration.
- **Download:** Used to restore a configuration saved to the servo drives connected to the Lexium Controller on the Motion bus. This function is used for the immediate transfer of the configuration parameters from the servo drives, necessary during a maintenance operation or duplication of a machine.
- **Edit Par:** Opens the exhaustive list of the axis configuration parameters. This list of parameters is detailed in the next section, List of configuration parameters (see page 11). This button also enables access to the extended function buttons in the configuration toolbar.
- **Read Par:** Used to read previously saved data.
- **Write Par:** Used to write modified data.
- **Save Drive:** Used to save these parameters to the servo drive's non-volatile memory.
Note : You must have performed a "Write Par" before "Save Drive".

Configuring the devices

Configuring the controller

This is the first step in the configuration. To open the Lexium Controller configuration menu click on the "**LMC Par**" button on the left of the screen, in the navigation bar.

Activating this button opens a special window for configuring 8 parameters:

- "Project Name": Enter the name of the project being created.
- "Number of Axis in Project": Enter the number of real axes used in the project.
Note : When the number of axis is modified, it is compulsory to reset the Lexium Controller.
- "Display encoder": This option allows ("True") or prohibits ("False") display of the external encoder parameters.
- "Display VirtAxis": This option allows ("True") or prohibits ("False") display of the virtual axis parameters.
- The "**IP Address**" of the Lexium controller: This address consists of 4 fields to be filled in one after the other.
Note : The Lexium controller must be restarted for the new address to be taken into account.
- The "**IP Mask**" subnet mask: This mask also consists of 4 fields to be filled in one after the other.
- The "**Modbus Address**" of the Lexium controller: Used to display and modify the Modbus address.
- The address of the optional fieldbus, "**Fieldbus Address**": Used to display the address configured using DeviceNet or Profibus DP.
- The "**Date**" and "**Time**": These two fields are used to adjust the internal clock of the Lexium Controller.
- The "**Firmware version**" of the controller. Used to display the software version of the Lexium Controller.

Configuring the axes

Before starting programming the motion tasks in the Lexium Controller, the servo drives connected to the controller must be configured one at a time.

- The servo drive CANopen addresses must correspond to the order of the axes configured on Easy Motion (between 1 and 8)
- The CANopen data rate must correspond to that configured on the Lexium Controller (1 Mbps by default on Easy Motion)

The Lexium Controller can control three types of axis on the Motion Bus:

- Real axes: Up to 8 axes synchronized axes
- Virtual axes: 1 axis with the Easy Motion offer
- A master encoder axis

To open the axis configuration menu, click on the "**Axis Par**" button located on the left of the screen, in the navigation bar

Configuring the devices

List of parameters

Real axes

| Item | Detailed description | Access | Unit | Minimum | Maximum | Default |
|---------------------|--|--------|--------------------|-----------------|------------------------|---------|
| Parameter of axis | Designates the axis whose parameters are being displayed or modified. | R/W | - | 1 | 8 | - |
| Axis Name | Used to name an axis. This field is free format, maximum 11 characters. The name entered appears on the axis display (see page 16). | R/W | - | - | - | - |
| Rotary/Linear | Used to choose between a rotary axis or a linear axis. Rotary: A rotary axis has unlimited travel. The negative [Soft limit neg.] and positive [Soft limit pos.] limit positions are not active in this case. Linear: A linear axis is one whose range of travel is limited by the positions of negative [Soft limit neg.] and positive [Soft limit pos.] software limits. A linear axis performs absolute and relative movements within movement limits that are defined by software limits. A reference point must be defined. | R/W | - | 0 = Rotary | 1 = Linear | 0 |
| Modulo | Used to set the rollover parameters of an axis. When a value other than zero is entered, the axis becomes a rollover axis. The Modulo parameter is used to define the position interval traveled by the axis in one turn. This type of axis requires a referencing movement. Note: This parameter is only active for a rotary axis. | R/W | User unit/turn (1) | 0 | 999999 | 360 |
| Soft Limit Enable | Used to activate or deactivate the software limit that defines a minimum or maximum limit position. When the axis is linear and [Soft Limit Enable] = 1 the range of travel of the axis is limited by the negative [Soft limit neg.] and positive [Soft limit pos.] software limit positions. | R/W | - | 0 = Deactivated | 1 = Activated | 0 |
| Soft Limit Pos | Used to define a maximum software limit position for a linear axis. When the axis is linear and [Soft Limit Enable] = 1 the range of travel of the axis is limited by the negative [Soft limit neg.] and positive [Soft limit pos.] software limit positions. | R/W | User unit (1) | -999999 | 999999 | 0 |
| Soft Limit Neg | Used to define a minimum software limit position for a linear axis. When the axis is linear and [Soft Limit Enable] = 1 the range of travel of the axis is limited by the negative [Soft limit neg.] and positive [Soft limit pos.] software limit positions. | R/W | User unit (1) | -999999 | 999999 | 0 |
| Ramp Type | Trapezoid: The servo drive is subject to constant linear acceleration/deceleration until the target velocity is reached. Sinus²: To reduce jolts, the servo drive is accelerated/decelerated according to a continuous acceleration ramp. The resulting velocity characteristic corresponds to a sinusoidal ² curve. | R/W | - | 0 = Trapezoid | 1 = Sinus ² | 0 |
| User Unit Numerator | This parameter is used to configure the numerator of the scaling factor for the position, velocity and acceleration values. An explanation of how this scaling works is given page 87 | R/W | - | -32768 | 32767 | 45 |

(1) The user units depend on the scaling performed with the [User Unit Numerator] and [User Unit Denominator] parameters.

Configuring the devices

| Item | Detailed description | Access | Unit | Minimum | Maximum | Default |
|-----------------------|---|--------|-------------------|---------------|------------------|----------------|
| User Unit Denominator | This parameter is used to configure the denominator of the scaling factor for the position, velocity and acceleration values. An explanation of how this scaling works is given page 87 | R/W | - | 1 | 2147483647 | 16384 |
| Homing Mode | Definition of the type of homing to be performed. The default value depends on the servo drive used: Lexium 05 range: Positive limit. | R/W | - | 1 | 35 | 18 |
| Homing Speed | Defines the Homing Speed. Lexium 05 parameter: HMn Note: Please refer to the servo drive documentation for more detailed information. | R/W | min ⁻¹ | 1 | 3000 | 60 |
| KPn | Adjustment of the servo drive velocity loop proportional gain. Lexium 05 parameter: KPn Note: Please refer to the servo drive documentation for more detailed information. | R/W | - | Lexium 05: 1 | Lexium 05: 12700 | Lexium 05: - |
| TNn | Adjustment of the servo drive velocity loop integral time. Lexium 05 parameter: TNn Note: Please refer to the servo drive documentation for more detailed information. | R/W | ms | Lexium 05: 1 | Lexium 05: 32767 | Lexium 05: - |
| KPp | Adjustment of the servo drive position loop proportional gain. Lexium 05 parameter: KPp Note: Please refer to the servo drive documentation for more detailed information. | R/W | - | Lexium 05: 20 | Lexium 05: 4950 | Lexium 05: 495 |
| VelFeedFor | Determines the position controller predictive control factor. Predictive control is used to reduce the task of the servo drive position controller. Better adjustment of this factor also makes for easier use of the dynamic range of the servo drive position controller. The most favorable setting (in general 1.0) depends on factors external to the servo drive such as friction, dynamic resistance and rigidity. Lexium 5 parameter: CTRL_KFPp Note: Please refer to the servo drive documentation for more detailed information. | R/W | - | Lexium 05: 0 | Lexium 05: 1100 | Lexium 05: - |
| Drive | Type of servo drive | R/- | - | - | - | - |

Configuring the devices

Virtual axes

| Item | Detailed description | Access | Unit | Minimum | Maximum | Default |
|-----------------------|--|--------|--------------------|-----------------|------------------------|---------|
| Parameter of axis | Designates the axis whose parameters are being displayed or modified. | R/W | - | 9 | 9 | 9 |
| Axis Name | Used to name an axis. This field is free format, maximum 11 characters. The name entered appears on the axis display (see page 16). | R/W | - | - | - | - |
| Rotary/Linear | Used to choose between a rotary axis or a linear axis. Rotary: A rotary axis has unlimited travel. The negative [Soft limit neg.] and positive [Soft limit pos.] limit positions are not active in this case. Linear: A linear axis is one whose range of travel is limited by the positions of negative [Soft limit neg.] and positive [Soft limit pos.] software limits. A linear axis performs absolute and relative movements within movement limits that are defined by software limits. A reference point must be defined. | R/W | - | 0 = Rotary | 1 = Linear | 0 |
| Modulo | Used to set the rollover parameters of an axis. When a value other than zero is entered, the axis becomes a rollover axis. The Modulo parameter is used to define the position interval traveled by the axis in one turn. This type of axis requires a referencing movement. Note: This parameter is only active for a rotary axis. | R/W | User unit/turn (1) | 0 | 999999 | 360 |
| Soft Limit Enable | Used to activate or deactivate the software limit that defines a minimum or maximum limit position. When the axis is linear and [Soft Limit Enable] = 1 the range of travel of the axis is limited by the negative [Soft limit neg.] and positive [Soft limit pos.] software limit positions. | R/W | - | 0 = Deactivated | 1 = Activated | 0 |
| Soft Limit Pos | Used to define a maximum software limit position for a linear axis. When the axis is linear and [Soft Limit Enable] = 1 the range of travel of the axis is limited by the negative [Soft limit neg.] and positive [Soft limit pos.] software limit positions. | R/W | User unit (1) | -999999 | 999999 | 0 |
| Soft Limit Neg | Used to define a minimum software limit position for a linear axis. When the axis is linear and [Soft Limit Enable] = 1 the range of travel of the axis is limited by the negative [Soft limit neg.] and positive [Soft limit pos.] software limit positions. | R/W | User unit (1) | -999999 | 999999 | 0 |
| Ramp Type | Trapezoid: The servo drive is subject to constant linear acceleration/deceleration until the target velocity is reached. Sinus²: To reduce jolts, the servo drive is accelerated/decelerated according to a continuous acceleration ramp. The resulting velocity characteristic corresponds to a sinusoidal ² curve. | R/W | - | 0 = Trapezoid | 1 = Sinus ² | 0 |
| User Unit Numerator | This parameter is used to configure the numerator of the scaling factor for the position, velocity and acceleration values. An explanation of how this scaling works is given page 87 | R/W | - | -32768 | 32767 | 45 |
| User Unit Denominator | This parameter is used to configure the denominator of the scaling factor for the position, velocity and acceleration values. An explanation of how this scaling works is given page 87 | R/W | - | 1 | 2147483647 | 16384 |

(1) The user units depend on the scaling performed with the [User Unit Numerator] and [User Unit Denominator] parameters.

Configuring the devices

External encoder

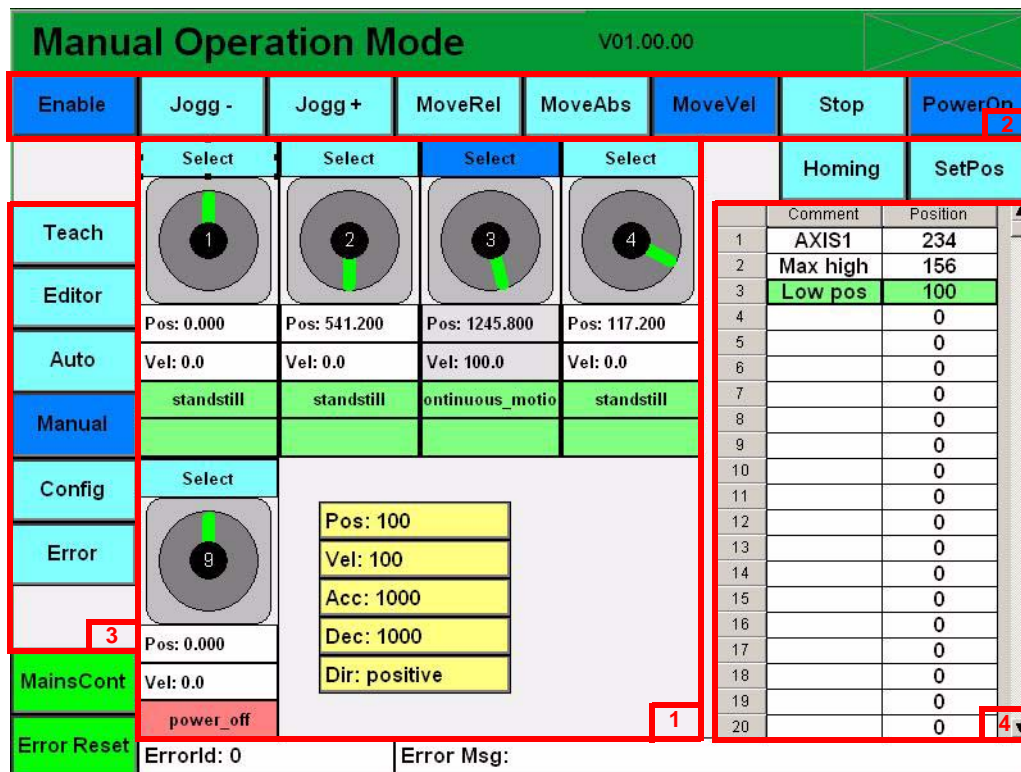
| Item | Detailed description | Access | Unit | Minimum | Maximum | Default |
|-----------------------|--|--------|--------------------|-----------------|---------------|---------|
| Parameter of axis | Designates the axis whose parameters are being displayed or modified. | R/W | - | 10 | 10 | 10 |
| Axis Name | Used to name an axis. This field is free format, maximum 11 characters. The name entered appears on the axis display. | R/W | - | - | - | - |
| Rotary/Linear | Used to choose between a rotary axis or a linear axis. Rotary: A rotary axis has unlimited travel. The negative [Soft limit neg.] and positive [Soft limit pos.] limit positions are not active in this case. Linear: A linear axis is one whose range of travel is limited by the positions of negative [Soft limit neg.] and positive [Soft limit pos.] software limits. A linear axis performs absolute and relative movements within movement limits that are defined by software limits. A reference point must be defined. | R/W | - | 0 = Rotary | 1 = Linear | 0 |
| Modulo | Used to set the rollover parameters of an axis. When a value other than zero is entered, the axis becomes a rollover axis. The Modulo parameter is used to define the position interval traveled by the axis in one turn. This type of axis requires a referencing movement. Note: This parameter is only active for a rotary axis. | R/W | User unit/turn (1) | 0 | 999999 | 360 |
| Soft Limit Enable | Used to activate or deactivate the software limit that defines a minimum or maximum limit position. When the axis is linear and [Soft Limit Enable] = 1 the range of travel of the axis is limited by the negative [Soft limit neg.] and positive [Soft limit pos.] software limit positions. | R/W | - | 0 = Deactivated | 1 = Activated | 0 |
| Soft Limit Pos | Used to define a minimum software limit position for a linear axis. When the axis is linear and [Soft Limit Enable] = 1 the range of travel of the axis is limited by the negative [Soft limit neg.] and positive [Soft limit pos.] software limit positions. | R/W | User unit (1) | -999999 | 999999 | 0 |
| Soft Limit Neg | Used to define a maximum software limit position for a linear axis. When the axis is linear and [Soft Limit Enable] = 1 the range of travel of the axis is limited by the negative [Soft limit neg.] and positive [Soft limit pos.] software limit positions. | R/W | User unit (1) | -999999 | 999999 | 0 |
| User Unit Numerator | This parameter is used to configure the numerator of the scaling factor for the position, velocity and acceleration values. An explanation of how this scaling works is given page 87 | R/W | - | -32768 | 32767 | - |
| User Unit Denominator | This parameter is used to configure the denominator of the scaling factor for the position, velocity and acceleration values. An explanation of how this scaling works is given page 87 | R/W | - | 1 | 2147483647 | - |

(1) The user units depend on the scaling performed with the [User Unit Numerator] and [User Unit Denominator] parameters.

Manual mode

Manual mode is used for individual control of each of the axes connected to the Lexium Controller.

General organization of the screen



The manual mode screen consists of four sections.

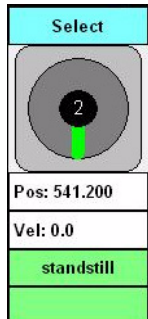
- 1 A central screen for supervising all the connected axes and selecting the active axis.
- 2 The control bar at the top of the screen.
- 3 The navigation bar with the addition of the "Teach" button.
- 4 The table of positions stored via the teach function.

Manual mode

Central supervision screen

Display of the real axes:

The central supervision screen displays each of the real axes connected to the Lexium Controller. The axes are represented by the following visual:



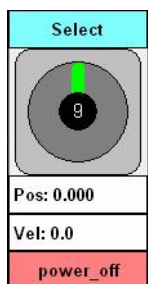
- **"Select" button:** In this mode only one axis can be active at a time. This axis is activated by clicking on the "Select" button.
- **Representation of the motor shaft:** Gives the position of the axis in the turn in real time.
- **"Pos" field:** Gives the absolute position of the axis in real time.
- **"Vel" field:** Gives the current velocity of the axis.
- **Servo drive status field:** Gives the PLC open status diagram status (see page [24](#)) of the axis in real time.
- **Active error field:** Gives the last active error on the axis in real time.

Not more than 4 real axes can be displayed at the same time.
When more than 4 axes are created, click on **"Axis 5-8"** to display the other axes.



Display of the virtual axis:

The central supervision screen displays the virtual axis. The axis is represented by the following visual:



- **"Select" button:** In this mode only one axis can be active at a time. This axis is activated by clicking on the "Select" button.
- **Representation of the motor shaft:** Gives the position of the axis in the turn in real time.
- **"Pos" field:** Gives the absolute position of the axis in real time.
- **"Vel" field:** Gives the current velocity of the axis.
- **Axis status field:** Gives the PLC open status diagram status (see page [24](#)) of the axis in real time.

Manual mode

Control parameters

All the parameters required for executing a movement on the selected axis can be accessed via the following table:

| |
|---------------|
| Pos: 100 |
| Vel: 100 |
| Acc: 1000 |
| Dec: 1000 |
| Dir: positive |

- **"Pos" field:** Gives the position to be reached when executing a movement. This distance can be relative (in the context of executing a relative movement) or absolute (in the context of executing an absolute movement).
- **"Vel" field:** Gives the maximum velocity of the velocity profile generated in the context of the execution of a MoveRel, Move Abs or MoveVel type movement (see Control bar section on page 17).
- **"Acc" field:** Gives the acceleration of the velocity profile generated in the context of the execution of a MoveRel, Move Abs or MoveVel type movement (see Control bar section on page 17).
- **"Dec" field:** Gives the deceleration of the velocity profile generated in the context of the execution of a MoveRel, Move Abs or MoveVel type movement (see Control bar section on page 17).
- **"Dir" field:** Gives the position of the movement. There are 5 possible types of position:
 - -1 = Negative direction (Negative direction is defined as being counterclockwise rotation of the shaft viewed facing the motor shaft).
 - 0 = The shortest (Only for rotary axes. The controller calculates the shortest distance between the current position and the target position. It takes the shortest route).
 - 1 = Positive direction (Positive direction is defined as being clockwise rotation of the shaft viewed facing the motor shaft).
 - 2 = Current direction (Only for rotary axes).
 - 3 = The fastest (Only for rotary axes).

Toolbar

The control bar at the top of the manual mode screen is used to enable, energize and start movements on the selected axis. *

| | | | | | | | | | |
|------------|-------|-------|---------|---------|---------|------|--------|--------|---------|
| Man Enable | Jog - | Jog + | MoveRel | MoveAbs | MoveVel | Stop | Homing | SetPos | PowerOn |
|------------|-------|-------|---------|---------|---------|------|--------|--------|---------|

This bar contains 10 buttons. Each button has a particular function:

- **"Man Enable":** Used to enable manual mode and enable the selected axis.
- **"jog -":** Starts a continuous movement at constant velocity in negative direction. The velocity, acceleration and deceleration of this movement are those entered in the Control parameters table described on page 17.
- **"jog +":** Starts a continuous movement at constant velocity in positive direction. The velocity, acceleration and deceleration of this movement are those entered in the Control parameters table described on page 17.
- **"MoveRel":** Starts a relative movement (eMoveRel type, see page 28) with the target position, maximum velocity, acceleration and deceleration values entered in the Control parameters table described on page 17.
- **"MoveAbs":** Starts an absolute movement (eMoveAbs type, see page 25) with the target position, maximum velocity, acceleration and deceleration values entered in the Control parameters table described on page 17.
- **"MoveVel":** Starts a continuous movement at constant velocity (eMoveVel type, see page 32) with the velocity, acceleration, deceleration, and direction values entered in the Control parameters table described on page 17.
- **"Stop":** stops the current movement.
- **"Homing":** Starts the homing procedure on the selected axis. The homing parameters such as the homing type, velocity and acceleration of this movement are those entered in the Configuration of devices menu described on page 11.
- **"SetPos":** This is used to redefine the current position of the axis (eSetPos block, see page 40). The aim of doing this is, for example, to define a new reference position.
- **"PowerOn":** Enables the power on the selected axis.

Manual mode

Teaching positions - "Teach" button

In EasyMotion it is possible to capture 32 positions using the teach function. This teach function is accessed by clicking on the "Teach" button in the navigation bar.



Four buttons then appear to the left of the position storage table.

Control buttons

- **Enter:** Saves the current position of the axis to the line selected in the table.
- **InsLine:** Inserts a line in the table in the position of the selected line. The following lines are then moved down one row.
- **DelLine:** Deletes the selected line.
- **DelList:** Deletes the list of stored positions.

The table of positions stored via the teach function

Each captured position is stored in the position table located on the right of the "Manual Mode" screen.

| | Comment | Position |
|----|----------|----------|
| 1 | Axis 1 | 100 |
| 2 | High Pos | 200 |
| 3 | Low Pos | 50 |
| 4 | | 0 |
| 5 | | 0 |
| 6 | | 0 |
| 7 | | 0 |
| 8 | | 0 |
| 9 | | 0 |
| 10 | | 0 |
| 11 | | 0 |
| 12 | | 0 |
| 13 | | 0 |
| 14 | | 0 |
| 15 | | 0 |
| 16 | | 0 |
| 17 | | 0 |
| 18 | | 0 |
| 19 | | 0 |
| 20 | | 0 |
| 21 | | 0 |
| 22 | | 0 |
| 23 | | 0 |
| 24 | | 0 |
| 25 | | 0 |
| 26 | | 0 |
| 27 | | 0 |
| 28 | | 0 |
| 29 | | 0 |
| 30 | | 0 |
| 31 | | 0 |
| 32 | | 0 |

The name of the axis appears in this table by default, but only if the name has been defined. The stored position can be renamed or altered by clicking on the selected box and entering the required information.

How are the stored positions used?

The stored positions can be used in the following blocks:

- eMoveAbs
- eMoveRel
- eMoveAdd
- eMoveContAbs
- eMoveContRel

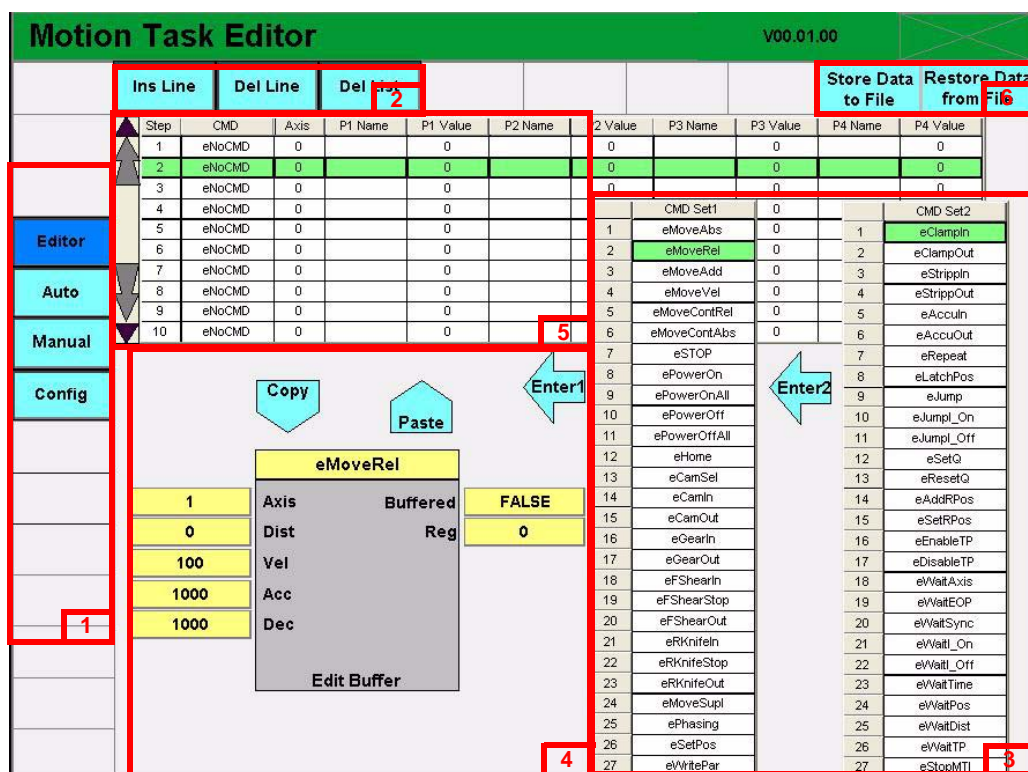
Simply enter the position register number in these blocks in order to use the distance entered as the target distance.

Motion task editor

Motion Task editor mode is used to configure a sequence of function blocks step by step. These can be motion control blocks, axis status management blocks or blocks to manage events linked to this sequence. Each of the accessible blocks is described in the Function Blocks section on page 22. The execution of this sequence is governed by the operating principles described in the Execution rules section on page 66.

The motion sequence editor is accessed by clicking on the "Editor" button in the navigation bar (on the left of the Easy Motion screen).

General organization of the screen



This screen is divided into five parts:

- 1 Navigation bar.
- 2 Toolbar. This is at the top of the screen and contains 3 buttons.
- 3 List of accessible function blocks. On the right of the screen.
- 4 Function block editing area.
- 5 The motion sequence.
- 6 The data store and restore area.

Toolbar

Located at the top of the screen, the motion sequence editor toolbar contains three buttons:

| | | |
|----------|----------|----------|
| Ins Line | Del Line | Del List |
|----------|----------|----------|

- **Ins Line:** Inserts a line above the selected line in the motion sequence table. The following lines are then moved down one number.

Note: If the list is full (i.e. there are 64 blocks in the list) the last block is overwritten.

- **Del Line:** Deletes the selected line.
- **Del List:** Deletes the entire sequence.

Motion task editor

List of accessible function blocks

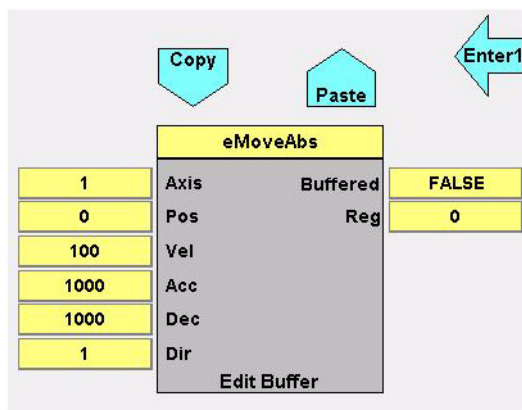
Located on the right of the screen in the special dropdown menu area (see page 19), this is used to access all the function blocks that can be used in the motion sequence (see Function blocks section on page 22).

| | CMD Set1 |
|----|--------------|
| 1 | eMoveAbs |
| 2 | eMoveRel |
| 3 | eMoveAdd |
| 4 | eMoveVel |
| 5 | eMoveContRel |
| 6 | eMoveContAbs |
| 7 | eSTOP |
| 8 | ePowerOn |
| 9 | ePowerOnAll |
| 10 | ePowerOff |
| 11 | ePowerOffAll |
| 12 | eHome |
| 13 | eCamSel |
| 14 | eCamIn |
| 15 | eCamOut |
| 16 | eGearIn |
| 17 | eGearOut |
| 18 | eFShearIn |
| 19 | eFShearStop |
| 20 | eFShearOut |
| 21 | eRKnifeIn |
| 22 | eRKnifeStop |
| 23 | eRKnifeOut |
| 24 | eMoveSupl |
| 25 | ePhasing |
| 26 | eSetPos |
| 27 | eWritePar |

Once a block has been selected in this list, it can be configured in the function block editing area by clicking on the "ENTER" button.

Function block editing area

When the "ENTER" button is pressed, the function block selected in the list is displayed graphically in the function block editing area. In this area the parameters required for each block can be modified. These parameters are described in the Function blocks section on page 22.



The function block editing area for **eMoveAbs** is shown. It features a central grey box labeled **eMoveAbs** with a **Edit Buffer** label at the bottom. To the left of this box is a vertical list of input fields with values: 1, 0, 100, 1000, 1000, and 1. To the right, there are two output fields: **Buffered** with value **FALSE** and **Reg** with value **0**. Above the central box are two buttons: **Copy** and **Paste**. To the right of the **Paste** button is a large blue arrow pointing left, labeled **Enter1**.

Once these parameters have been entered, the block can be pasted into the motion sequence using the "Paste" button. The block is then ready to be executed according to the sequence entered in the motion sequence.

Motion task editor

The motion sequence

Easy Motion allows 64 steps to be entered (divided between all the connected axes) in the motion sequence. Each step in the sequence contains one function block.

This sequence is represented as a value table containing the list of all the parameters for each block. This list can be displayed in the "Motion Task Editor" screen.

| Step | CMD | Axis | P1 Name | P1 Value | P2 Name | P2 Value | P3 Name | P3 Value | P4 Name | P4 Value |
|------|-----------|------|---------|----------|---------|----------|---------|----------|---------|----------|
| 1 | eWait_On | 0 | Input | 1 | | 0 | | 0 | | 0 |
| 2 | eWait_Off | 0 | Input | 1 | | 0 | | 0 | | 0 |
| 3 | eWait_On | 0 | Input | 2 | | 0 | | 0 | | 0 |

Displaying this screen does not provide access to all the parameters in the list, but does enable a block to be selected and:

- It can be edited using the "Copy" button in the function block editing area (see page 20)
- It can be deleted using the "Del Line" button in the toolbar (see page 19)

The complete list of the motion sequence is as shown below. It can be accessed via the Modbus table (see page 91) in order to be edited via the fieldbus.

| Step | Command | Axis | Param. 1 | Val. P1 | Param. 2 | Val. P2 | Param. 3 | Val. P 3 |
|------|---------|-----------|----------|-----------|----------|---------|----------|----------|
| 1 | PowerOn | Conveyor1 | | | | | | |
| 2 | PowerOn | Slave | | | | | | |
| 3 | MoveAbs | Conveyor1 | Position | 23478 | Velocity | 123 | Accel | 23 |
| 4 | GearIn | Slave | Master | Conveyor1 | Ratio | 2.4 | Accel | 45 |
| 5 | | | | | | | | |
| 6 | | | | | | | | |

Storing and restoring data



"Store Data to file": Used to store data in a file

- Step 1: Parameters list files (.SYM) allow to chose the range of parameters to be saved:
 - Complete Application Template parameters [Application Template Parameter.SYM].
 - The motion task table parameters [Motion Task List.SYM].
 - It is also possible to create customized SYM files with Motion Pro.

Default folder: C:\Program Files\Schneider Electric\Motion Control\CoDeSys V2.3\Targets\Schneider Electric\Lexium Controller Vxxxx\Examples\Templates

- Step 2: Chose the directory and the name of the file for data storage
- Step 3: Specify communication parameters for connection with LMC
- Step 4: Start to store

"Restore Data from File": Used to restore data from a file.

- Step 1: Select file with parameter to be written into LMC
- Step 2: Specify communication parameters for connection with LMC
- Step 3: Start restore

Motion task editor

Step by step: How to add a block to the motion sequence

Let's use the example of the eMoveRel block:

Motion Task Editor V00.01.00

| Ins Line | Del Line | Del List | Store Data to File | Restore Data from File | | | | | | |
|----------|----------|----------|--------------------|------------------------|---------|----------|---------|----------|---------|----------|
| Step | CMD | Axis | P1 Name | P1 Value | P2 Name | P2 Value | P3 Name | P3 Value | P4 Name | P4 Value |
| 1 | eNoCMD | 0 | | 0 | | 0 | | 0 | | 0 |
| 2 | eNoCMD | 0 | | 0 | | 0 | | 0 | | 0 |
| 3 | eNoCMD | 0 | | 0 | | 0 | | 0 | | 0 |
| 4 | eNoCMD | 0 | | 0 | | 0 | | 0 | | 0 |
| 5 | eNoCMD | 0 | | 0 | | 0 | | 0 | | 0 |
| 6 | eNoCMD | 0 | | 0 | | 0 | | 0 | | 0 |
| 7 | eNoCMD | 0 | | 0 | | 0 | | 0 | | 0 |
| 8 | eNoCMD | 0 | | 0 | | 0 | | 0 | | 0 |
| 9 | eNoCMD | 0 | | 0 | | 0 | | 0 | | 0 |
| 10 | eNoCMD | 0 | | 0 | | 0 | | 0 | | 0 |

Editor **Auto** **Manual** **Config**

Copy **Paste**

Enter1 **Enter2**

eMoveRel

| | | | |
|------|------|----------|-------|
| 1 | Axis | Buffered | FALSE |
| 0 | Dist | Reg | 0 |
| 100 | Vel | | |
| 1000 | Acc | | |
| 1000 | Dec | | |

Edit Buffer

| CMD Set1 | CMD Set2 |
|-----------------|---------------|
| 1 eMoveAbs | 1 eClampIn |
| 2 eMoveRel | 2 eClampOut |
| 3 eMoveAdd | 3 eStripIn |
| 4 eMoveVel | 4 eStripOut |
| 5 eMoveContRel | 5 eAccuIn |
| 6 eMoveContAbs | 6 eAccuOut |
| 7 eSTOP | 7 eRepeat |
| 8 ePowerOn | 8 eLatchPos |
| 9 ePowerOnAll | 9 eJump |
| 10 ePowerOff | 10 eJump_On |
| 11 ePowerOffAll | 11 eJump_Off |
| 12 eHome | 12 eSetQ |
| 13 eCamSel | 13 eResetQ |
| 14 eCamIn | 14 eAddRPos |
| 15 eCamOut | 15 eSetRPos |
| 16 eGearIn | 16 eEnableTP |
| 17 eGearOut | 17 eDisableTP |
| 18 eFShearIn | 18 eWaitAxis |
| 19 eFShearStop | 19 eWaitEOP |
| 20 eFShearOut | 20 eWaitSync |
| 21 eRKnifeIn | 21 eWait_On |
| 22 eRKnifeStop | 22 eWait_Off |
| 23 eRKnifeOut | 23 eWaitTime |
| 24 eMoveSupl | 24 eWaitPos |
| 25 ePhasing | 25 eWaitDist |
| 26 eSetPos | 26 eWaitTP |
| 27 eWritePar | 27 eStopMTI |

- 1 First of all the block is selected in the list of accessible function blocks.
- 2 Then the block is displayed in the function block editing area by clicking on the **"Enter1"** button.
- 3 The 8 parameters are then entered one at a time.
- 4 Line 2 of the motion sequence is then selected.
- 5 The block is inserted in this line by clicking on the **"Paste"** button.

The next block can be inserted.

Function blocks

Easy Motion enables most of the function blocks in Motion Pro to be accessed. These function blocks comply with the specifications of the PLCOpen standard.

These commands can be of the following types:

- Single axis positioning
- Administrative
- Multi-axis positioning
- Application functions
- Execution conditions

Motion task editor

Easy Motion/Motion Pro/PLCopen equivalence table

This table indicates the equivalent names for the following functions:

N.B.:

The Easy Motion functions use the blocks that are available with Motion Pro. Motion Pro is based on the SoftMotion library developed by 3S. This library runs in the CoDeSys environment. It complies with the PLCopen recommendations.

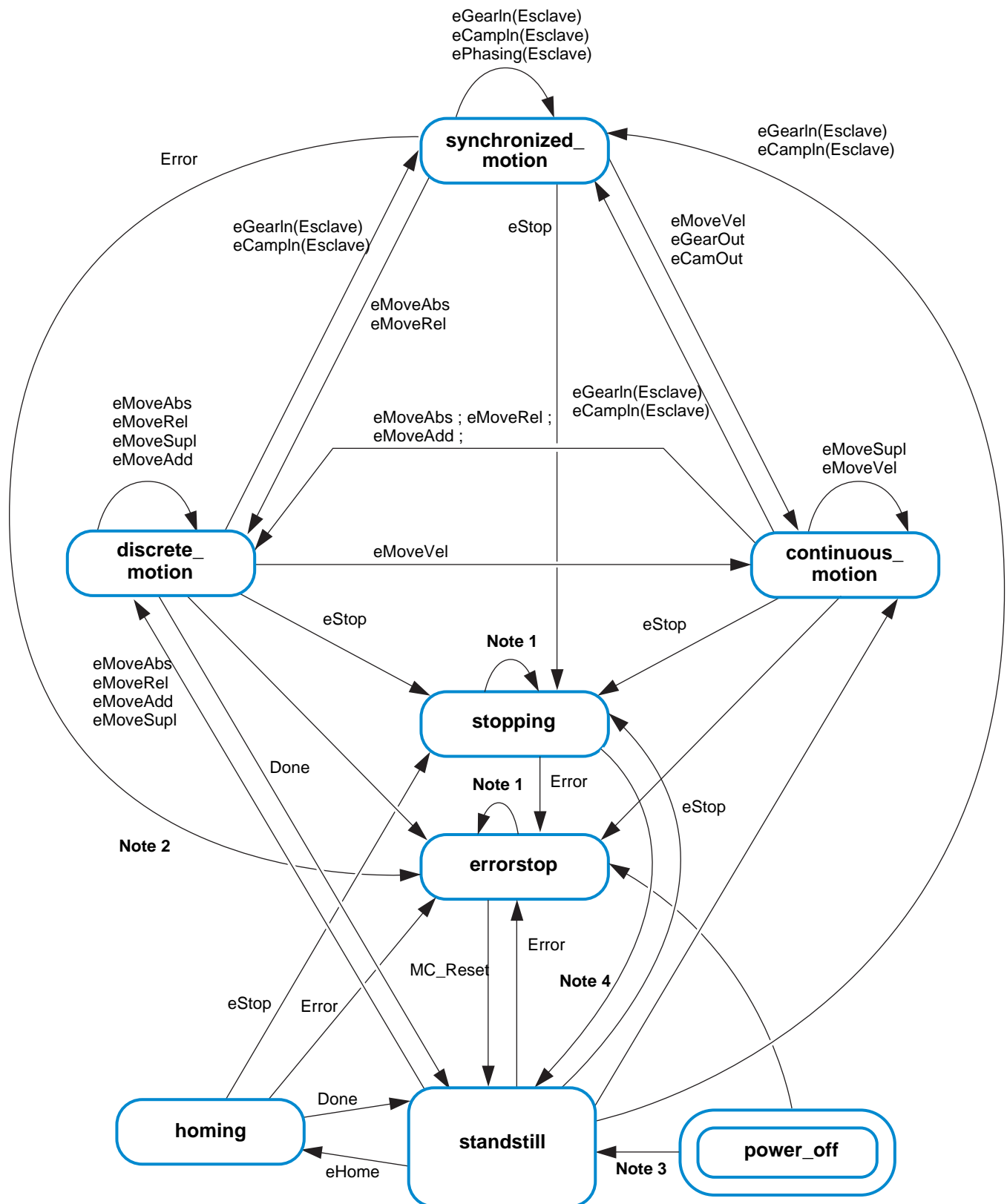
PLCopen is a structural and functional description of positioning functions. This description has been drawn up by a group of companies made up of the main suppliers on the position control market.

| Easy Motion | Motion Pro | PLCopen |
|--------------|----------------------------|---------------------|
| eMoveAbs | MC_MoveAbsolute | MC_MoveAbsolute |
| eMoveRel | MC_MoveRelative | MC_MoveRelative |
| eMoveAdd | MC_MoveAdditive | MC_MoveAdditive |
| eMoveVel | MC_MoveVelocity | MC_MoveVelocity |
| eMoveContRel | SMC_MoveContinuousAbsolute | |
| eMoveContAbs | SMC_MoveContinuousRelative | |
| eStop | MC_Stop | |
| ePowerOn | MC_Power | MC_Power |
| ePowerOnAll | | |
| ePowerOff | | |
| ePowerOffAll | | |
| eHome | MC_PassiveHome | MC_PassiveHome |
| eCamSel | MC_CamTableSelect | MC_CamTableSelect |
| eCamIn | MC_CamIn | MC_CamIn |
| eCamOut | MC_CamOut | MC_CamOut |
| eGearIn | MC_GearIn | MC_GearIn |
| eGearOut | MC_GearOut | MC_GearOut |
| eFShearIn | | |
| eFShearStop | | |
| eFShearOut | | |
| eRKnifeIn | | |
| eRKnifeStop | | |
| eRKnifeOut | | |
| eMoveSupl | MC_MoveSuperimposed | MC_MoveSuperimposed |
| ePhasing | MC_Phasing | MC_Phasing |
| eSetPos | MC_SetPosition | |
| eRepeat | | |
| eLatchPos | | |
| eJump | | |
| eJumpI_On | | |
| eJumpI_Off | | |
| eSetQ | | |
| eResetQ | | |
| eSetRPos | | |
| eAddRPos | | |
| eEnableTP | MC_TouchProbe | |
| eDisableTP | | |
| eWaitAxis | | |
| eWaitEOP | | |
| eWaitSync | | |
| eWaitI_On | | |
| eWaitI_Off | | |
| eWaitTime | | |
| eWaitPos | | |
| eWaitDist | | |

Motion task editor

Status diagram

These commands follow the status diagram of the PLCopen standard



Note 1: In this ErrorStop or Stopping state, all the function blocks can be called, even if they are not executed. Only the Error Reset button or an error can generate the transition to the StandStill or ErrorStop states respectively.

Note 2: Power on = TRUE and there is an active error on the axis.

Note 3: Power on = TRUE and there is no active error on the axis.

Note 4: eStop performed AND NO eStop in progress.

Motion task editor

Managing the execution time : "TimeOut"

Each block in the sequence of motion tasks must be executed within a limited period. This period is defined by the "TimeOut" value entered for each of the blocks. If this value is 0, there is no limitation on the execution time of the block. If this value is positive (For example X ms), the block must be executed in X milliseconds. After this period, the Lexium Controller generates an error message "MTI Timeout Error" (See page [88](#))

Single axis positioning commands

The single axis motion control function blocks are defined in the PLCopen standard for coordinated axes, i.e., those that do not require a link with one another. These function blocks execute simple positioning commands such as relative or absolute movements, etc.

eMoveAbs

| eMoveAbs | | | |
|-------------|------|----------|-------|
| 1 | Axis | Buffered | FALSE |
| 0 | Pos | Reg | 0 |
| 100 | Vel | | |
| 1000 | Acc | | |
| 1000 | Dec | | |
| 1 | Dir | | |
| Edit Buffer | | | |

This command creates a movement of the axis to an absolute position following a trapezoid or Sinus² path. The choice of the type of path is connected with the parameters of the axes (see Parameters of real axes on page [11](#) or Parameters of virtual axes on page [13](#)).

The shape of the profile is defined by the velocity, acceleration and deceleration values given as the parameters of the function block. Depending on the limits of the linear axis, this distance can be positive or negative (if the axis is defined from -2000 to 2000, for example). A negative distance will be rejected if the axis is positive only, and vice versa.

In the case of a rotary axis, the direction parameter indicates the direction of rotation. This parameter is not used for a linear axis. If the axis is rotary, this position cannot be negative.

The Reg parameter is used to specify whether the position to be reached is that entered in the Pos parameter (in this case, Reg = 0) or a position in the table of positions stored via the teach function (see page [18](#)). In this case, Reg will take the value of the index in the table. The "Buffered" parameter is used to link the current movement to another movement on the same axis without using eWaitAxis function (see page [67](#) for an example). A sequence can contain a maximum of 8 buffered movements.

| Parameter | Identifier | Function | Adjustment range/Units |
|------------|------------|---------------------------------|--|
| Axis | Axis | Axis number | 1 to 9 |
| P1 | Pos | Position | User units |
| P2 | Vel | Velocity | User units/Sec |
| P3 | Acc | Acceleration | User units/sec^2 |
| P4 | Dec | Deceleration | User units/sec^2 |
| P5 | Dir | Direction (Rotary axis only) | 0 = shortest |
| | | | 1 = positive direction |
| | | | 2 = current direction |
| | | | 3 = fastest |
| | | | -1 = negative direction |
| P6 | Reg | Position register index | 0 = Position of P1 |
| | | | 1..max = Position of teach function register |
| Sequencing | Buffered | Sequencing of movements | 0 = Deactivated |
| | | | 1 = Activated |

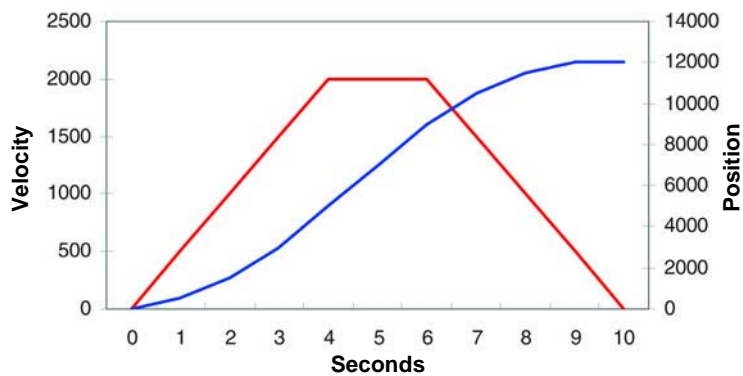
Motion task editor

Step by step: Examples of movement

Linear axis with a trapezoid motion profile.

| | |
|----------|-------|
| Axis | 1 |
| Pos | 12000 |
| Vel | 2000 |
| Acc | 500 |
| Dec | 500 |
| Dir | N/A |
| Reg | 0 |
| Buffered | 0 |

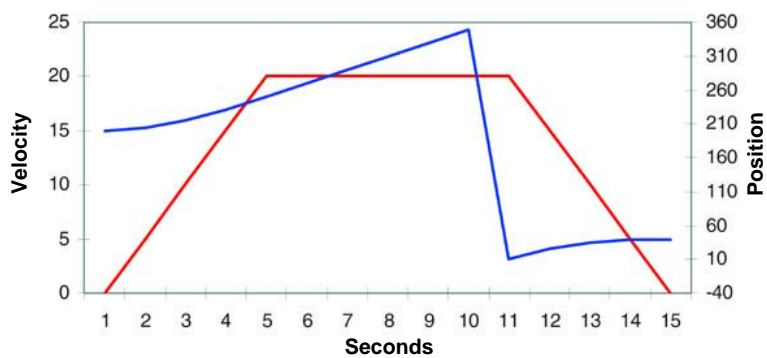
The above configuration gives the following behavior:



Rotary axis (360° rollover) with a trapezoid motion profile and positive movement (the axis is already positioned at 200°).

| | |
|----------|----|
| Axis | 1 |
| Pos | 40 |
| Vel | 20 |
| Acc | 5 |
| Dec | 5 |
| Dir | 1 |
| Reg | 0 |
| Buffered | 0 |

The above configuration gives the following behavior:

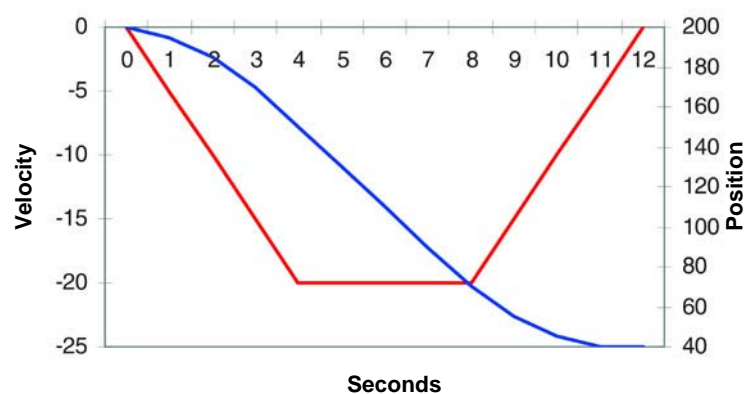


Motion task editor

Rotary axis (360° rollover) with a trapezoid motion profile and shortest movement (the axis is already positioned at 200°).

| | |
|----------|----|
| Axis | 1 |
| Pos | 40 |
| Vel | 20 |
| Acc | 5 |
| Dec | 5 |
| Dir | 0 |
| Reg | 0 |
| Buffered | 0 |

The above configuration gives the following behavior:



Motion task editor

eMoveRel

| | | | |
|-------------|------|----------|-------|
| eMoveRel | | | |
| 1 | Axis | Buffered | FALSE |
| 0 | Dist | Reg | 0 |
| 100 | Vel | | |
| 1000 | Acc | | |
| 1000 | Dec | | |
| Edit Buffer | | | |

This command creates a movement of the axis for a distance relative to its current position following a trapezoid or Sinus² path.

The choice of the type of path is connected with the parameters of the axes (see Parameters of real axes on page [11](#) or Parameters of virtual axes on page [13](#)).

The shape of the profile is defined by the velocity, acceleration and deceleration parameters given as the parameters of the function block. This distance can be positive or negative. In the case of a linear axis, the position must be within the limits of the axis.

In the case of a rotary axis, if it has no end limit, and if the relative distance exceeds the rollover, the axis will perform several turns to reach the position.

The Reg parameter is used to specify whether the position to be reached is that entered in the Pos parameter (in this case, Reg = 0) or a position in the table of positions stored via the teach function (see page [18](#)). In this case, Reg will take the value of the index in the table. The "Buffered" parameter is used to link the current movement to another movement on the same axis without using eWaitAxis function (see page [67](#) for an example). A sequence can contain a maximum of 8 movements.

| Parameter | Identifier | Function | Adjustment range/Units |
|------------|------------|-------------------------|--|
| Axis | Axis | Axis number | 1 to 9 |
| P1 | Dist | Distance | User units |
| P2 | Vel | Velocity | User units/Sec |
| P3 | Acc | Acceleration | User units/sec^2 |
| P4 | Dec | Deceleration | User units/sec^2 |
| P6 | Reg | Position register index | 0 = Position of P1 |
| | | | 1..max = Position of teach function register |
| Sequencing | Buffered | Sequencing of movements | 0 = Deactivated |
| | | | 1 = Activated |

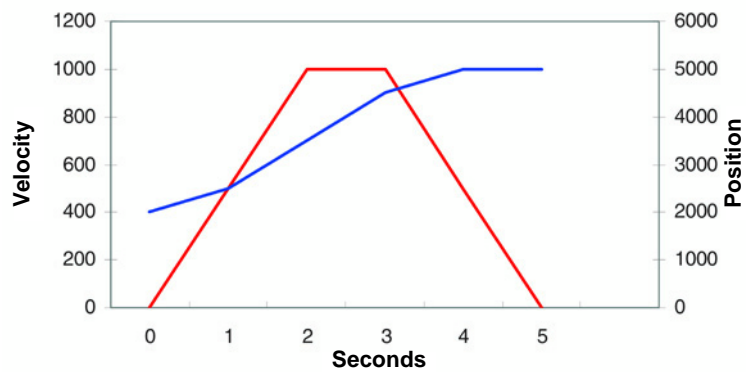
Motion task editor

Step by step: Examples of movement

Linear axis with a trapezoid positive motion profile. The axis is already at position 2000.

| | |
|----------|------|
| Axis | 1 |
| Pos | 3000 |
| Vel | 1000 |
| Acc | 500 |
| Dec | 500 |
| Reg | 0 |
| Buffered | 0 |

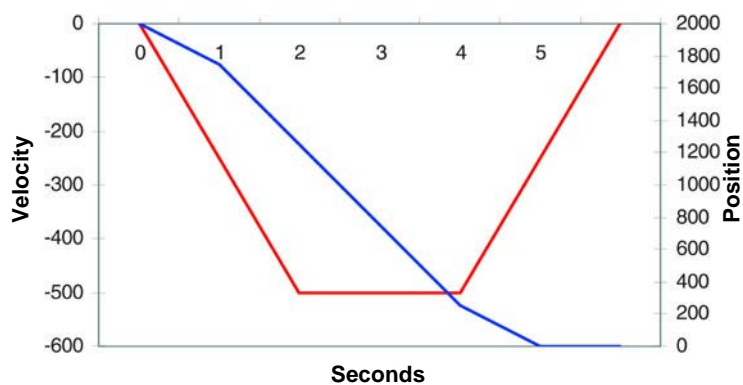
The above configuration gives the following behavior:



Linear axis with a trapezoid negative motion profile. The axis is already at position 2000.

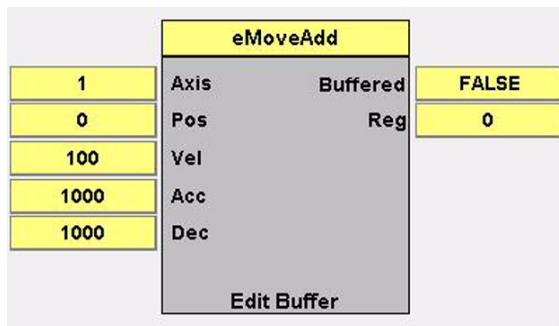
| | |
|----------|-------|
| Axis | 1 |
| Pos | -2000 |
| Vel | 500 |
| Acc | 250 |
| Dec | 250 |
| Reg | 0 |
| Buffered | 0 |

The above configuration gives the following behavior:



Motion task editor

eMoveAdd



This command adds a distance relative to target position of the axis, following a trapezoid or Sinus² path, to the current movement of the axis. There are two possibilities:

- The axis is in "discrete_motion" state: The axis performs a coordinated axis movement. The distance set in the eMoveAdd function block is added to the target position of the function currently operating on the axis. The velocity of the axis is thus that set in the eMoveAdd block. This function is useful you wish to add a distance to a movement without interrupting that movement.
- The axis is in "standstill" state: The axis is therefore stationary. The additional distance to be traveled is measured from the momentary position.

The choice of the type of path is connected with the parameters of the axes (see Parameters of real axes on page 11 or Parameters of virtual axes on page 13).

The profile is defined by the velocity, acceleration and deceleration parameters given as the parameters of the function block. This distance can be positive or negative. In the case of a linear axis, the target position must be within the limits of the axis.

In the case of a rotary axis, if it has no end limit, and if the target distance exceeds the rollover, the axis will perform several turns to reach the position.

The Reg parameter is used to specify whether the position to be reached is that entered in the Pos parameter (in this case, Reg = 0) or a position in the table of positions stored via the teach function (see page 18). In this case, Reg will take the value of the index in the table. The "Buffered" parameter is used to link the current movement to another movement on the same axis without using eWaitAxis function (see page 67 for an exemple). A sequence can contain a maximum of 8 movements.

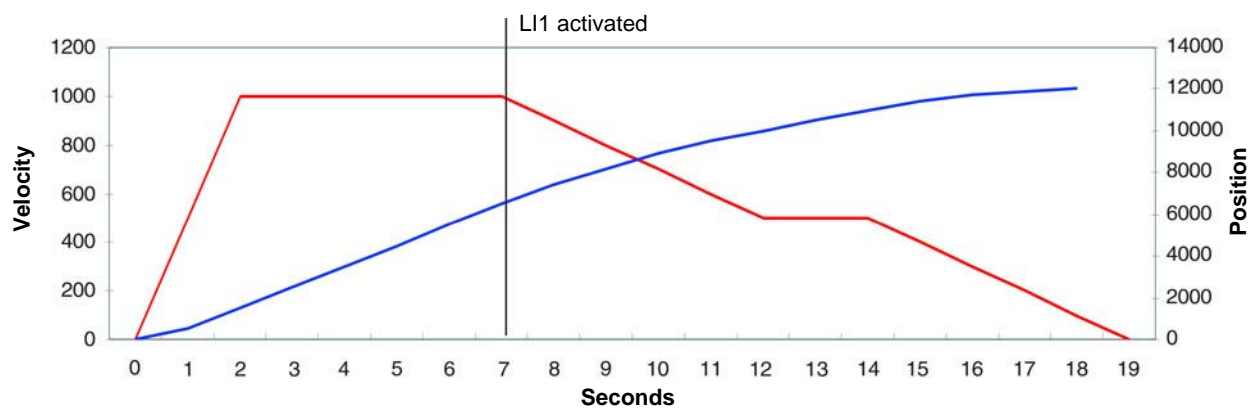
| Parameter | Identifier | Function | Adjustment range/Units |
|------------|------------|-------------------------|--|
| Axis | Axis | Axis number | 1 to 9 |
| P1 | Pos | Position | User units |
| P2 | Vel | Velocity | User units/Sec |
| P3 | Acc | Acceleration | User units/sec^2 |
| P4 | Dec | Deceleration | User units/sec^2 |
| P6 | Reg | Position register index | 0 = Position of P1 |
| | | | 1..max = Position of teach function register |
| Sequencing | Buffered | Sequencing of movements | 0 = Deactivated |
| | | | 1 = Activated |

Motion task editor

Step by step: Examples of movement

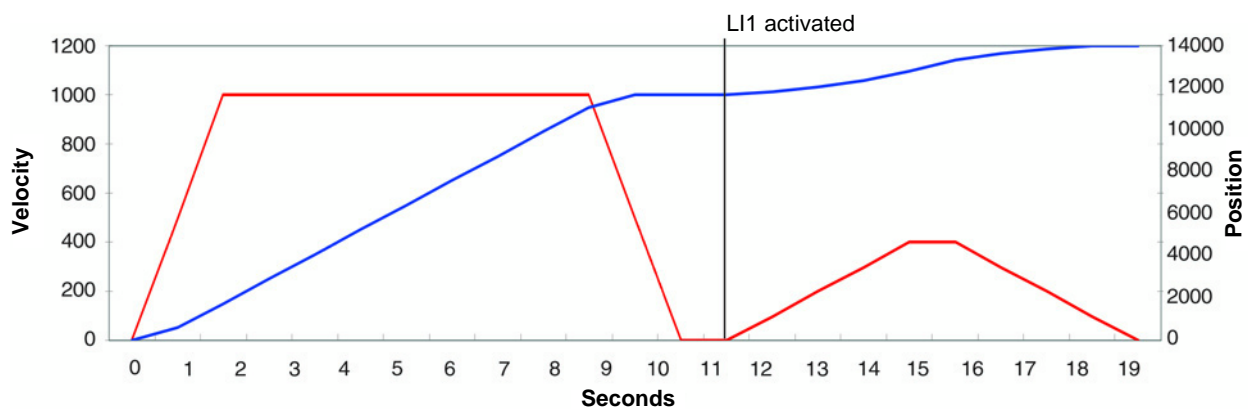
Linear axis with a trapezoid positive motion profile. The target position of the axis is 10000. It is still moving when 2000 is added to the position with a lower velocity on the rising edge of an input.

| Step | Command | Axis | Param.1 | Val. P1 | Param. 2 | Val. P2 | Param. 3 | Val. P3 | Param. 4 | Val. P4 | Param. 5 | Val. P5 | Param. 6 | Val. P6 | Param. 7 | Val. P7 |
|------|-----------|------|---------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| STEP | CMD | Axis | P1 Name | P1 Value | P2 Name | P2 Value | P3 Name | P3 Value | P4 Name | P4 Value | P5 Name | P5 Value | P6 Name | P6 Value | P7 Name | P7 Value |
| 1 | PowerOn | 1 | | | | | | | | | | | | | | |
| 2 | eMoveAbs | 1 | Pos | 10000 | Vel | 1000 | Acc | 500 | Dec | 500 | Dir | 1 | Reg | 0 | Buffered | 0 |
| 3 | eWaitl_On | 0 | Input | 1 | | | | | | | | | | | | |
| 4 | eMoveAdd | 1 | Pos | 2000 | Vel | 500 | Acc | 100 | Dec | 100 | | | Reg | 0 | Buffered | 0 |
| 5 | eWaitAxis | 1 | | | | | | | | | | | | | | |
| 6 | | | | | | | | | | | | | | | | |



Linear axis with a trapezoid positive motion profile. The target position of the axis is 2. It is stationary when 2000 is added to the position with a lower velocity on the rising edge of an input.

| Step | Command | Axis | Param.1 | Val. P1 | Param. 2 | Val. P2 | Param. 3 | Val. P3 | Param. 4 | Val. P4 | Param. 5 | Val. P5 | Param. 6 | Val. P6 | Param. 7 | Val. P7 |
|------|-----------|------|---------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| STEP | CMD | Axis | P1 Name | P1 Value | P2 Name | P2 Value | P3 Name | P3 Value | P4 Name | P4 Value | P5 Name | P5 Value | P6 Name | P6 Value | P7 Name | P7 Value |
| 1 | PowerOn | 1 | | | | | | | | | | | | | | |
| 2 | eMoveAbs | 1 | Pos | 12 | Vel | 1000 | Acc | 500 | Dec | 500 | Dir | 1 | Reg | 0 | Buffered | 0 |
| 3 | eWaitl_On | 0 | Input | 1 | | | | | | | | | | | | |
| 4 | eMoveAdd | 1 | Pos | 2000 | Vel | 500 | Acc | 100 | Dec | 100 | | | Reg | 0 | Buffered | 0 |
| 5 | eWaitAxis | 1 | | | | | | | | | | | | | | |
| 6 | | | | | | | | | | | | | | | | |



Motion task editor

eMoveVel

| eMoveVel | | |
|-------------|------|----------------|
| 1 | Axis | Buffered FALSE |
| 100 | Vel | |
| 1000 | Acc | |
| 1000 | Dec | |
| 1 | Dir | |
| Edit Buffer | | |

This command involves an infinite movement of the axis at the given velocity.

To reach this velocity, the servo drive follows a profile defined by the velocity, acceleration and deceleration parameters given in the function block.

The configured velocity must always be positive.

The direction of the movement is given by the Dir parameter.

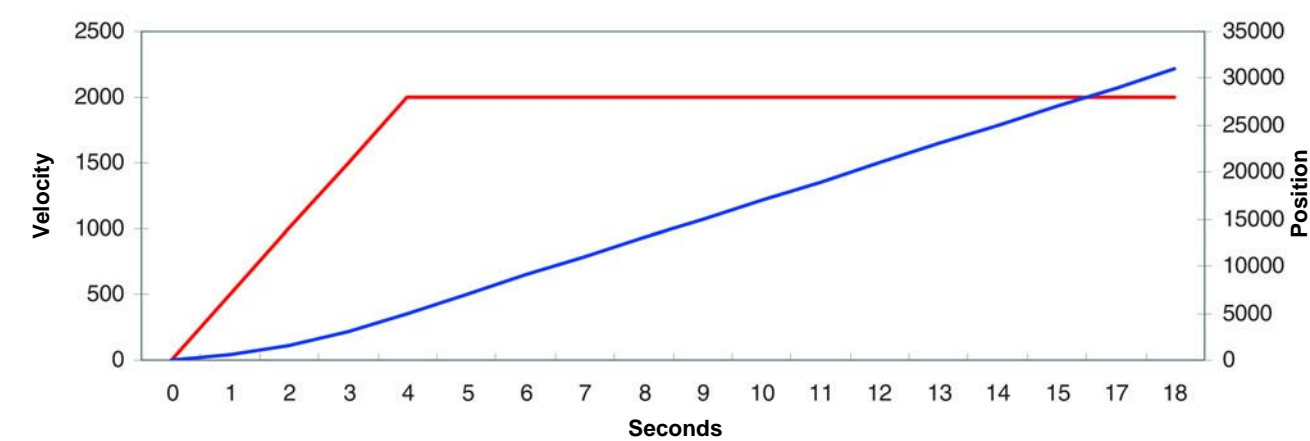
| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|---------------------------------|-------------------------|
| Axis | Axis | Axis number | 1 to 9 |
| P1 | Vel | Velocity | User units/Sec |
| P2 | Acc | Acceleration | User units/sec^2 |
| P3 | Dec | Deceleration | User units/sec^2 |
| P4 | Dir | Direction (Rotary axis only) | 0 = shortest |
| | | | 1 = positive direction |
| | | | 2 = current direction |
| | | | 3 = fastest |
| | | | -1 = negative direction |

Motion task editor

Step by step: Examples of movement

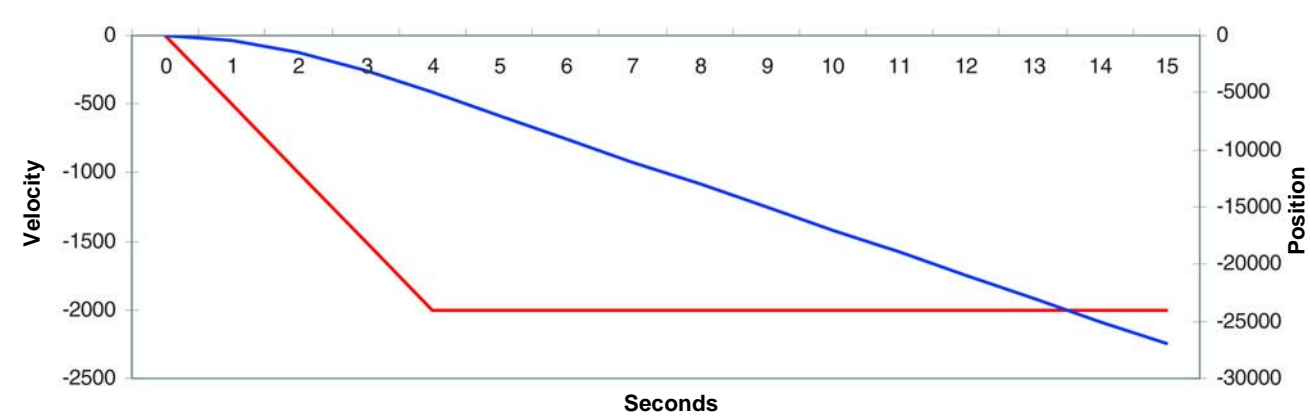
Linear axis with trapezoid profile. Infinite movement at a constant velocity of 2000 units/second in positive direction.

| | |
|------|------|
| Axis | 1 |
| Vel | 2000 |
| Acc | 500 |
| Dec | 500 |
| Dir | 1 |



Linear axis with trapezoid profile. Infinite movement at a constant velocity of 2000 units/second in negative direction.

| | |
|------|------|
| Axis | 1 |
| Vel | 2000 |
| Acc | 500 |
| Dec | 500 |
| Dir | -1 |



Motion task editor

eMoveContAbs

| eMoveContAbs | | | |
|--------------|--------|----------|-------|
| 1 | Axis | Buffered | FALSE |
| 0 | Pos | | |
| 100 | Vel | Reg | 0 |
| 0 | EndVel | | |
| 1000 | Acc | | |
| 1000 | Dec | | |
| Edit Buffer | | | |

This command creates a movement of the axis to an absolute position following a trapezoid or Sinus² path. The choice of the type of path is connected with the parameters of the axes (see Parameters of real axes on page 11 or Parameters of virtual axes on page 13).

The profile is defined by the velocity, acceleration and deceleration values given as the parameters of the function block.

This distance can be positive or negative, depending on the limits of the linear axis. If the axis is defined from -2000 to 2000, for example. A negative distance will be rejected if the axis is positive only, and vice versa.

Unlike the eMoveAbs block, the final velocity parameter ("EndVel") defines the velocity of the axis to the target position of the movement. This parameter is used to create movement sequences (see the Sequencing movements section, on page 67)

The Reg parameter is to be used to specify whether the position to be reached is that entered in the Pos parameter (in this case, Reg = 0) or a position in the table of positions stored via the teach function (see page 18). In this case, Reg will take the value of the index in the table. The "Buffered" parameter is used to link the current movement to another movement on the same axis without using eWaitAxis function (see page 67 for an example). A sequence can contain a maximum of 8 movements.

⚠ CAUTION

Depending on the command (route required to take the axis at the final velocity < distance between the starting position and the final position), the axis may perform an initial movement in the opposite direction.

Failure to follow this instruction can result in equipment damage.

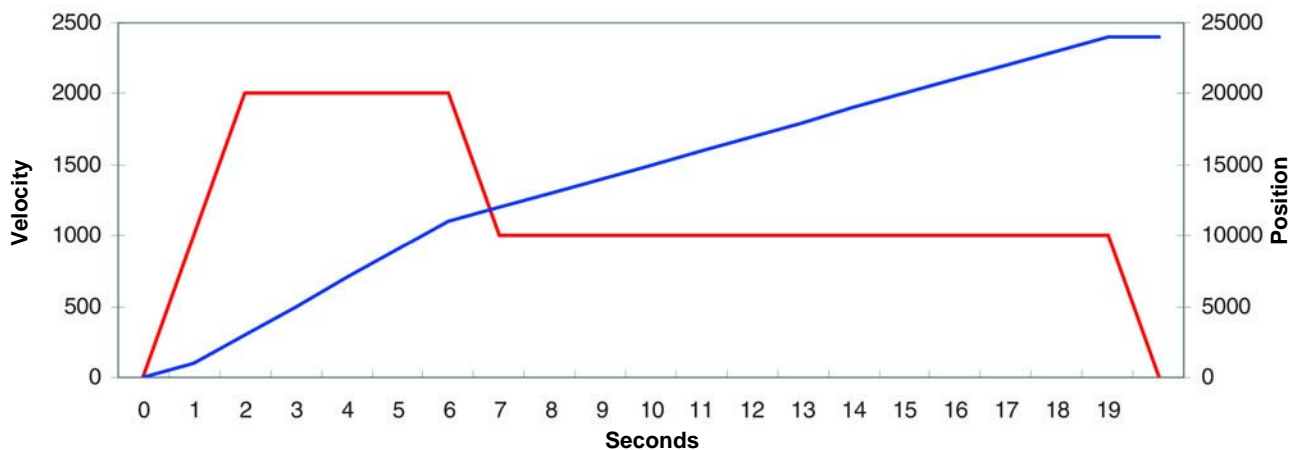
| Parameter | Identifier | Function | Adjustment range/Units |
|------------|------------|-------------------------|--|
| Axis | Axis | Axis number | 1 to 9 |
| P1 | Pos | Position | |
| P2 | Vel | Velocity | User units/Sec |
| P3 | EndVel | Final velocity | User units/Sec |
| P4 | Acc | Acceleration | User units/sec^2 |
| P5 | Dec | Deceleration | User units/sec^2 |
| P6 | Reg | Position register index | 0 = Position of P1 |
| | | | 1..max = Position of teach function register |
| Sequencing | Buffered | Sequencing of movements | 0 = Deactivated |
| | | | 1 = Activated |

Motion task editor

Step by step: Examples of movements

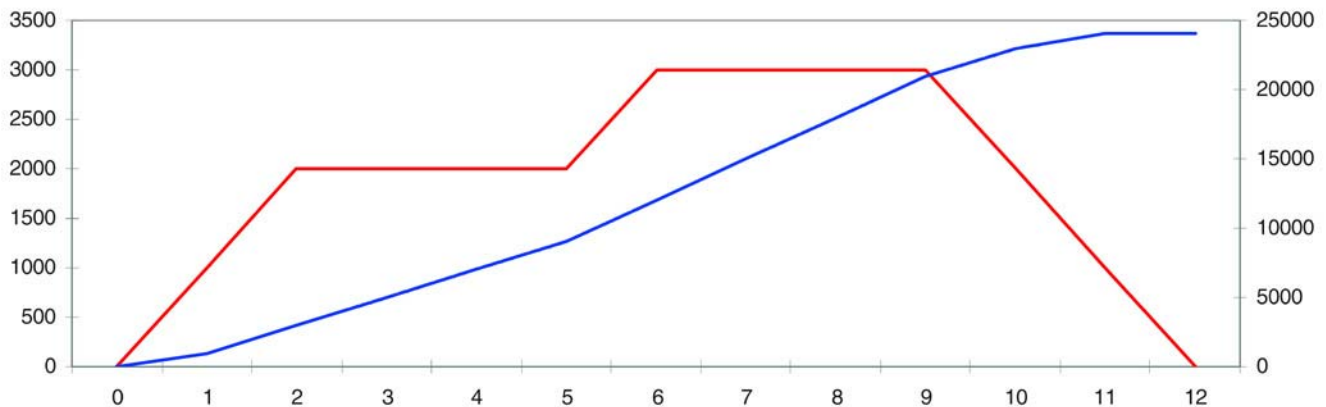
Linear axis with trapezoid profile. Absolute movement to position 12000 with a velocity of 2000, followed by an absolute movement to position 24000 with a velocity of 1000.

| Step | Command | Axis | Param.1 | Val. P1 | Param. 2 | Val. P2 | Param. 3 | Val. P3 | Param. 4 | Val. P4 | Param. 5 | Val. P5 | Param. 6 | Val. P6 | Param. 7 | Val. P7 |
|------|------------------|------|---------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| STEP | CMD | Axis | P1 Name | P1 Value | P2 Name | P2 Value | P3 Name | P3 Value | P4 Name | P4 Value | P5 Name | P5 Value | P6 Name | P6 Value | P7 Name | P7 Value |
| 1 | PowerOn | 1 | | | | | | | | | | | | | | |
| 2 | eMoveCont Abs | 1 | Pos | 12000 | Vel | 2000 | EndVel | 1000 | Acc | 1000 | Dec | 1000 | Reg | 0 | Buffered | 0 |
| 3 | eWaitAxis | 1 | | | | | | | | | | | | | | |
| 4 | eMoveAbs | 1 | Pos | 24000 | Vel | 1000 | Acc | 1000 | Dec | 1000 | Dir | 0 | Reg | 0 | Buffered | 0 |
| 5 | eWaitAxis | 1 | | | | | | | | | | | | | | |



Linear axis with trapezoid profile. Absolute movement to position 12000 with a velocity of 2000, followed by an absolute movement to position 24000 with a velocity of 3000.

| Step | Command | Axis | Param.1 | Val. P1 | Param. 2 | Val. P2 | Param. 3 | Val. P3 | Param. 4 | Val. P4 | Param. 5 | Val. P5 | Param. 6 | Val. P6 | Param. 7 | Val. P7 |
|------|------------------|------|---------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| STEP | CMD | Axis | P1 Name | P1 Value | P2 Name | P2 Value | P3 Name | P3 Value | P4 Name | P4 Value | P5 Name | P5 Value | P6 Name | P6 Value | P7 Name | P7 Value |
| 1 | PowerOn | 1 | | | | | | | | | | | | | | |
| 2 | eMoveCont Abs | 1 | Pos | 12000 | Vel | 2000 | EndVel | 3000 | Acc | 1000 | Dec | 1000 | Reg | 0 | Buffered | 0 |
| 3 | eWaitAxis | 1 | | | | | | | | | | | | | | |
| 4 | eMoveAbs | 1 | Pos | 24000 | Vel | 3000 | Acc | 1000 | Dec | 1000 | Dir | 0 | Reg | 0 | Buffered | 0 |
| 5 | eWaitAxis | 1 | | | | | | | | | | | | | | |



Motion task editor

eMoveContRel

| eMoveContRel | | | |
|--------------|--------|----------|-------|
| 1 | Axis | Buffered | FALSE |
| 0 | Dist | Dir | 1 |
| 100 | Vel | Reg | 0 |
| 0 | EndVel | | |
| 1000 | Acc | | |
| 1000 | Dec | | |
| Edit Buffer | | | |

This command creates a movement of the axis to a distance relative to its current position following a trapezoid or Sinus² path. The choice of the type of path is connected with the parameters of the axes (see Parameters of real axes on page 11 or Parameters of virtual axes on page 13).

The profile is defined by the velocity, acceleration and deceleration values given as the parameters of the function block.

This distance can be positive or negative. In the case of a linear axis, the target position must be within the limits of the axis. In the case of a rotary axis, if it has no end limit, and if the relative distance exceeds the rollover, the axis will perform several turns to reach the target position.

Unlike the eMoveRel block, the final velocity parameter ("EndVel") defines the velocity of the axis to the target position of the movement. This parameter is used to create movement sequences (see the Sequencing movements section, on page 67).

The Reg parameter is used to specify whether the position to be reached is that entered in the Pos parameter (in this case, Reg = 0) or a position in the table of positions stored via the teach function (see page 18). In this case, Reg will take the value of the index in the table. The "Buffered" parameter is used to link the current movement to another movement on the same axis without using eWaitAxis function (see page 67 for an example). A sequence can contain a maximum of 8 movements.

Caution: Depending on the command (route required to take the axis at the final velocity < distance between the starting position and the final position), the axis may perform an initial movement in the opposite direction.

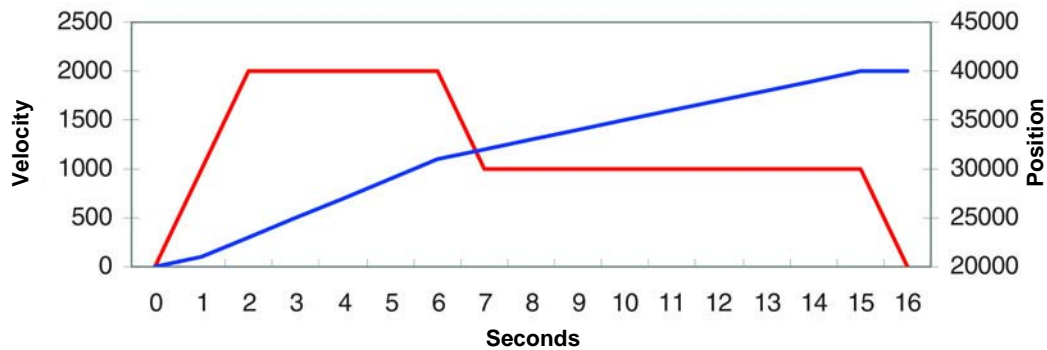
| Parameter | Identifier | Function | Adjustment range/Units |
|------------|------------|---------------------------------|--|
| Axis | Axis | Axis number | 1 to 91 to 9 |
| P1 | Dist | Distance | |
| P2 | Vel | Velocity | User units/Sec |
| P3 | EndVel | Final velocity | User units/Sec |
| P4 | Acc | Acceleration | User units/sec^2 |
| P5 | Dec | Deceleration | User units/sec^2 |
| P6 | Reg | Position register index | 0 = Position of P1 |
| | | | 1..max = Position of teach function register |
| P7 | Dir | Direction (Rotary axis only) | 0 = shortest |
| | | | 1 = positive direction |
| | | | 2 = current direction |
| | | | 3 = fastest |
| | | | -1 = negative direction |
| Sequencing | Buffered | Sequencing of movements | 0 = Deactivated |
| | | | 1 = Activated |

Motion task editor

Step by step: Examples of movements

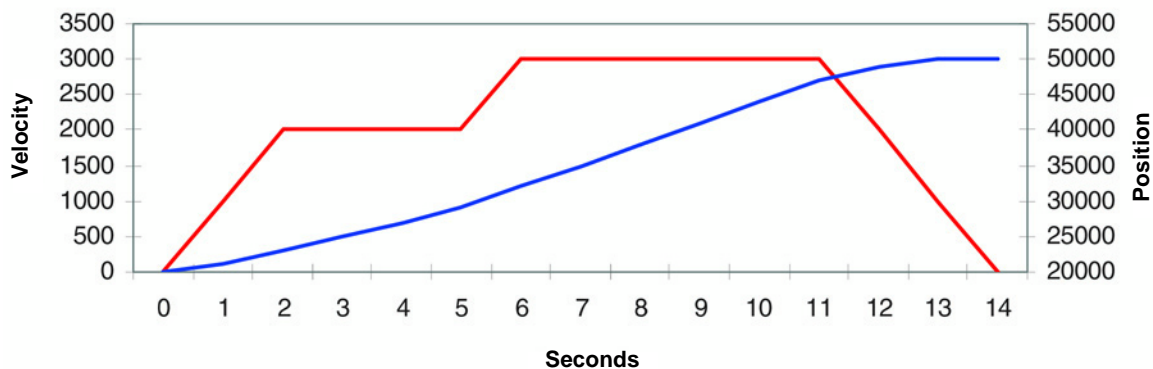
Linear axis with trapezoid profile. The axis is at position 20000. Relative movement to position 12000 with a velocity of 2000, followed by an absolute movement to position 26000 with a velocity of 1000.

| Step | Command | Axis | Param. 1 | Val. P1 | Param. 2 | Val. P2 | Param. 3 | Val. P3 | Param. 4 | Val. P4 | Param. 5 | Val. P5 | Param. 6 | Val. P6 |
|------|--------------|------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| STEP | CMD | Axis | P1 Name | P1 Value | P2 Name | P2 Value | P3 Name | P3 Value | P4 Name | P4 Value | P5 Name | P5 Value | P6 Name | P6 Value |
| 1 | PowerOn | 1 | | | | | | | | | | | | |
| 2 | eMoveContRel | 1 | Pos | 12000 | Vel | 2000 | EndVel | 1000 | Acc | 1000 | Dec | 1000 | Reg | 0 |
| 3 | eWaitAxis | 1 | | 1 | | | | | | | | | | |
| 4 | eMoveAbs | 1 | Pos | 40000 | Vel | 1000 | Acc | 1000 | Dec | 1000 | Dir | 0 | Reg | 0 |
| 5 | eWaitAxis | 1 | | | | | | | | | | | | |



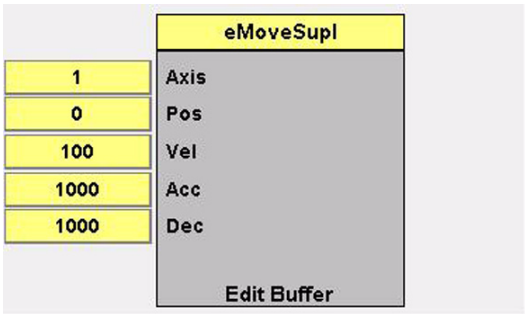
Linear axis with trapezoid profile. The axis is at position 20000. Relative movement to position 12000 with a velocity of 2000, followed by an absolute movement to position 26000 with a velocity of 3000.

| Step | Command | Axis | Param. 1 | Val. P1 | Param. 2 | Val. P2 | Param. 3 | Val. P3 | Param. 4 | Val. P4 | Param. 5 | Val. P5 | Param. 6 | Val. P6 |
|------|--------------|------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| STEP | CMD | Axis | P1 Name | P1 Value | P2 Name | P2 Value | P3 Name | P3 Value | P4 Name | P4 Value | P5 Name | P5 Value | P6 Name | P6 Value |
| 1 | PowerOn | 1 | | | | | | | | | | | | |
| 2 | eMoveContRel | 1 | Pos | 29000 | Vel | 2000 | EndVel | 3000 | Acc | 1000 | Dec | 1000 | Reg | 0 |
| 3 | eWaitAxis | 1 | | 1 | | | | | | | | | | |
| 4 | eMoveAbs | 1 | Pos | 50000 | Vel | 3000 | Acc | 1000 | Dec | 1000 | Dir | 0 | Reg | 0 |
| 5 | eWaitAxis | 1 | | | | | | | | | | | | |



Motion task editor

eMoveSupl



This command has two modes of operation, which depend on the status (see PLCopen status diagram on page 24) of the axis:

- The axis is in "discrete_motion" state: The axis performs a coordinated axis movement. The distance set in the eMoveSupl function block is added to the target position of the function currently operating on the axis. This new target position (i.e., the position of the movement plus that which has been added on) is then reached after the same period of time.
- The axis is in "continuous_motion" state: The axis performs an infinite movement. The additional distance to be traveled is measured from the momentary position. This function block is very useful for correcting a delay on the fly.
- The axis is in "Synchronised_motion" state: The axis performs a synchronized movement. The additional distance to be traveled is measured from the momentary position. This function block is very useful for correcting a delay on the fly.

This command is always executed in addition to another current command. It does not interrupt a current initial command.

The velocity, acceleration and deceleration values must be considered as being relative and thus independent of the current movement.

Important:
eMoveSupl must be executed after an initial movement. If the initial command is interrupted by a new command, the *eMoveSupl* command is interrupted.

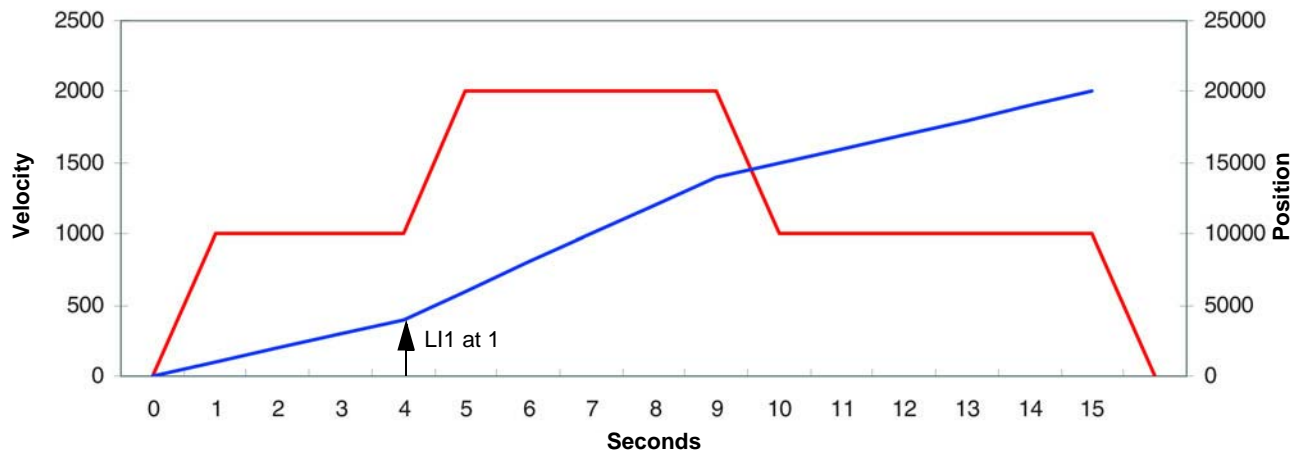
| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|--------------|------------------------|
| Axis | Axis | Axis number | 1 to 9 |
| P1 | Pos | Position | |
| P2 | Vel | Velocity | User units/Sec |
| P3 | Acc | Acceleration | User units/sec^2 |
| P4 | Dec | Deceleration | User units/sec^2 |

Motion task editor

Step by step: Examples of movements

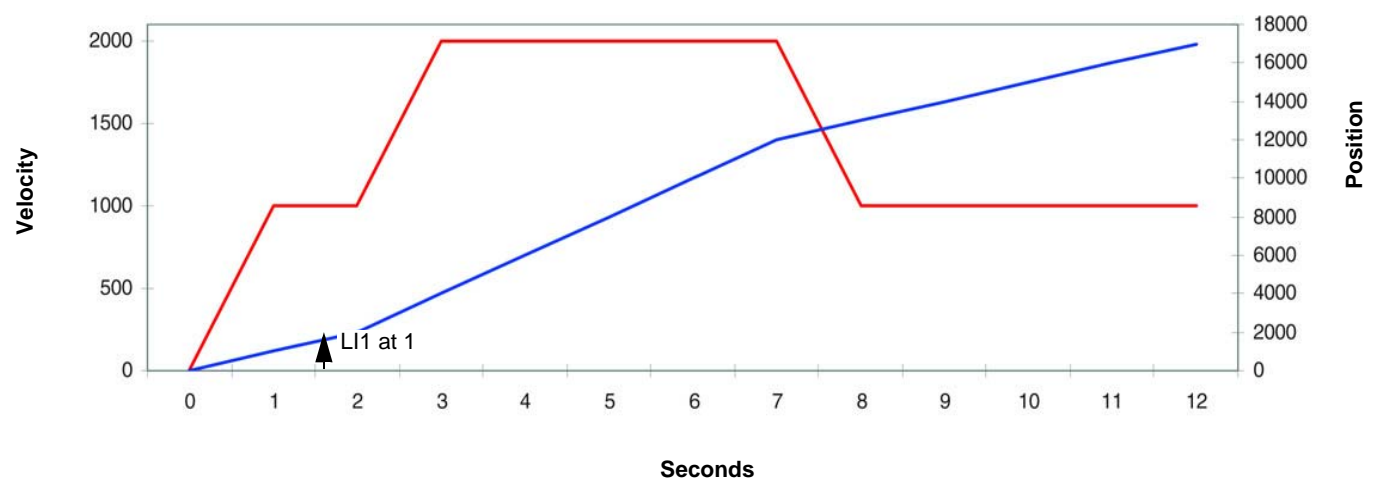
Linear axis with trapezoid profile. The axis performs an absolute movement to position 15000 with a velocity of 1000. Enabling an input triggers an additional movement of 5000.

| Step | Command | Axis | Param. 1 | Val. P1 | Param. 2 | Val. P2 | Param. 3 | Val. P3 | Param. 4 | Val. P4 | Param. 5 | Val. P5 | Param. 6 | Val. P6 |
|------|-----------|------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| STEP | CMD | Axis | P1 Name | P1 Value | P2 Name | P2 Value | P3 Name | P3 Value | P4 Name | P4 Value | P5 Name | P5 Value | P6 Name | P6 Value |
| 1 | PowerOn | 1 | | | | | | | | | | | | |
| 2 | eMoveAbs | 1 | Pos | 15000 | Vel | 1000 | Acc | 1000 | Dec | 1000 | Dir | 0 | Reg | 0 |
| 3 | eWaitl_On | 0 | Input | 1 | | | | | | | | | | |
| 4 | eMoveSupl | 1 | Pos | 5000 | Vel | 1000 | Acc | 1000 | Dec | 1000 | | | | |
| 5 | eWaitAxis | 1 | | | | | | | | | | | | |



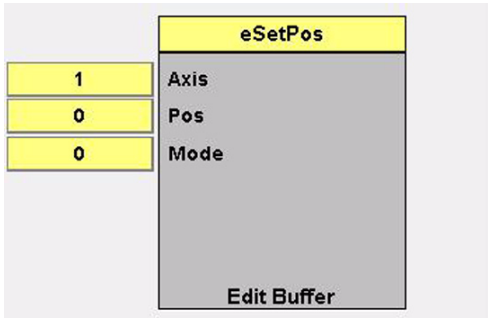
Linear axis with trapezoid profile. The axis performs an infinite movement at a velocity of 1000. Enabling an input triggers an additional movement of 5000.

| Step | Command | Axis | Param. 1 | Val. P1 | Param. 2 | Val. P2 | Param. 3 | Val. P3 | Param. 4 | Val. P4 |
|------|-----------|------|----------|----------|----------|----------|----------|----------|----------|----------|
| STEP | CMD | Axis | P1 Name | P1 Value | P2 Name | P2 Value | P3 Name | P3 Value | P4 Name | P4 Value |
| 1 | PowerOn | 1 | | | | | | | | |
| 2 | eMoveVel | 1 | Vel | 1000 | Vel | 2000 | Acc | 1000 | Dec | 1000 |
| 3 | eWaitl_On | 0 | Input | 1 | | | | | | |
| 4 | eMoveSupl | 1 | Pos | 5000 | Vel | 1000 | Acc | 1000 | Dec | 1000 |
| 5 | eWaitAxis | 1 | | | | | | | | |



Motion task editor

eSetPos



This command is used to redefine the position of the axis zero point. For example in order to define a new reference position.

This command can be executed during the execution of another positioning function.

There are two possible modes:

- Absolute: The position defined in P1 becomes the current position of the axis.
- Relative: The current position is shifted by the value entered in P1.

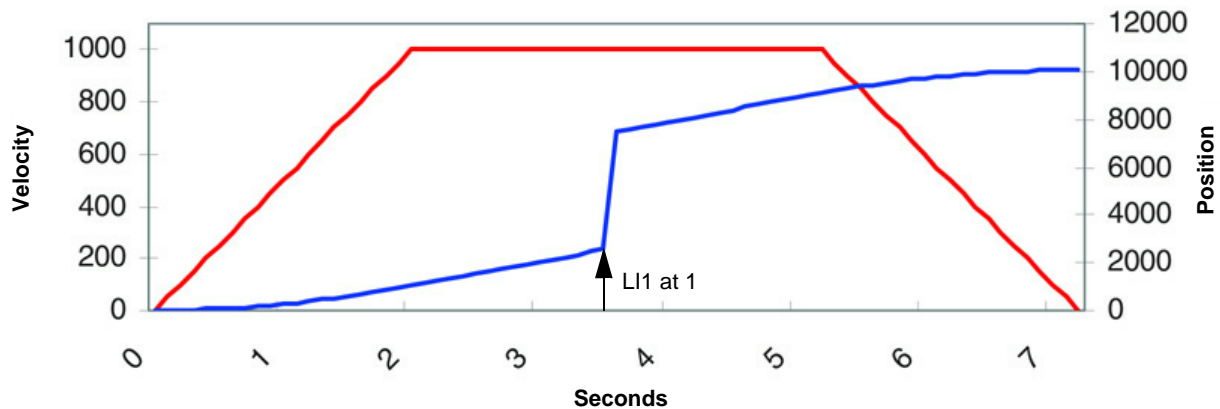
| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|-------------|------------------------|
| Axis | Axis | Axis number | 1 to 10 |
| P1 | Pos | Position | |
| P2 | Mode | Mode | 0 = Absolute |
| | | | 1 = Relative |

Motion task editor

Step by step: Examples of movements

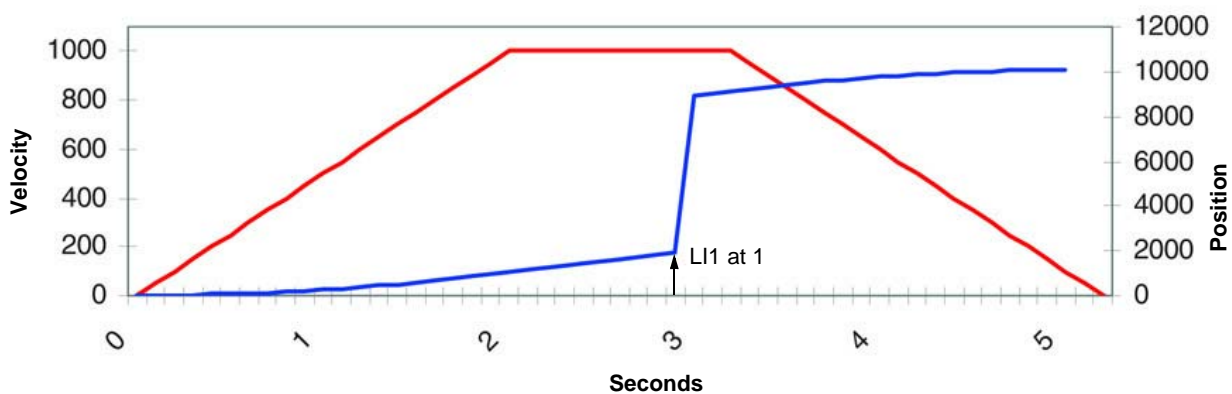
Linear axis with trapezoid profile. The axis performs an absolute movement to position 10000 with a velocity of 1000. Enabling an input triggers a redefinition of the current position as 7500.

| Step | Command | Axis | Param. 1 | Val. P1 | Param. 2 | Val. P2 | Param. 3 | Val. P3 | Param. 4 | Val. P4 | Param. 5 | Val. P5 | Param. 6 |
|------|-----------|------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| STEP | CMD | Axis | P1 Name | P1 Value | P2 Name | P2 Value | P3 Name | P3 Value | P4 Name | P4 Value | P5 Name | P5 Value | P6 Name |
| 1 | PowerOn | 1 | | | | | | | | | | | |
| 2 | eMoveAbs | 1 | Pos | 10000 | Vel | 1000 | Acc | 500 | Dec | 500 | Dir | 0 | Reg |
| 3 | eWaitI_On | 0 | Input | 1 | | | | | | | | | |
| 4 | eSetpos | 1 | Pos | 7500 | Mode | 0 | | | | | | | |
| 5 | eWaitAxis | 1 | | | | | | | | | | | |



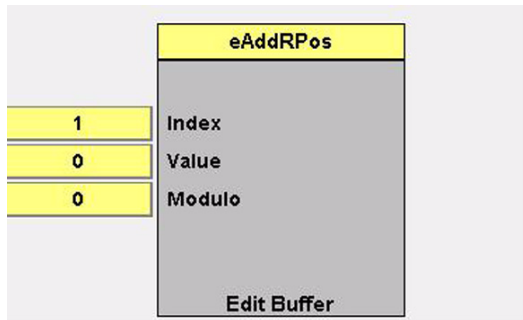
Linear axis with trapezoid profile. The axis performs an absolute movement to position 10000 with a velocity of 1000. Enabling an input triggers the addition of 7000 to the current position.

| Step | Command | Axis | Param. 1 | Val. P1 | Param. 2 | Val. P2 | Param. 3 | Val. P3 | Param. 4 | Val. P4 | Param. 5 | Val. P5 | Param. 6 |
|------|-----------|------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| STEP | CMD | Axis | P1 Name | P1 Value | P2 Name | P2 Value | P3 Name | P3 Value | P4 Name | P4 Value | P5 Name | P5 Value | P6 Name |
| 1 | PowerOn | 1 | | | | | | | | | | | |
| 2 | eMoveAbs | 1 | Pos | 10000 | Vel | 1000 | Acc | 500 | Dec | 500 | Dir | 0 | Reg |
| 3 | eWaitI_On | 0 | Input | 1 | | | | | | | | | |
| 4 | eSetpos | 1 | Pos | 7000 | Mode | 1 | | | | | | | |
| 5 | eWaitAxis | 1 | | | | | | | | | | | |



Motion task editor

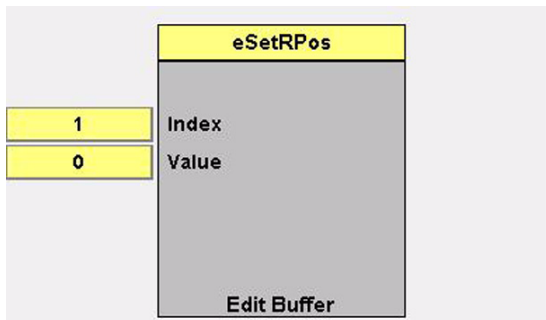
eAddRPos



This command adds the value defined in P2 to a position stored in the position register defined in P1. For rotary axes, the value given in P3 must be equal to the rollover value.

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|-------------------------------------|------------------------|
| P1 | Index | Register number | 1 to 32 |
| P2 | Value | Value of the position to be saved | User units |
| P3 | Modulo | Value of the rollover (rotary axis) | User units |

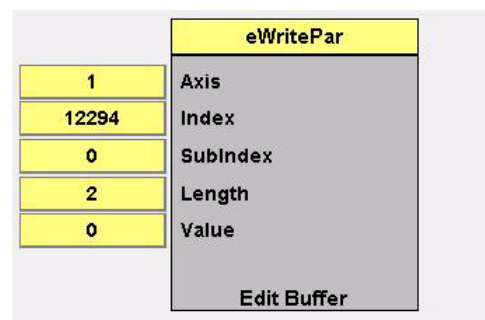
eSetRPos



This command writes the value of P2 to the P1 position register.

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|-----------------------------------|------------------------|
| P1 | Index | Register number | 1 to 32 |
| P2 | Value | Value of the position to be saved | User units |

eWritePar



This command is used to write a parameter to a servo drive using its CANopen address.

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|-------------------------------------|------------------------|
| Axe | Axis | Axis number | 1 to 9 |
| P1 | Index | Parameter CANopen address index | decimal |
| P2 | SubIndex | Parameter CANopen address sub-index | decimal |
| P3 | Length | Parameter length | bytes |
| P4 | Value | Value to be written | |

Motion task editor

Administrative commands

eStop

| | |
|---------|------|
| eSTOP | |
| 1 | Axis |
| 1000 | Dec |
| TimeOut | |

This command brakes the axis to zero velocity. **It cannot be interrupted and has to be followed by a eWaitAxis function.**

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|--------------|------------------------|
| Axis | Axis | Axis number | 1 to 9 |
| P1 | Dec | Deceleration | Units/sec^2 |

ePowerOn

| | |
|-------------|------|
| ePowerOn | |
| 1 | Axis |
| Edit Buffer | |

This command enables the power stage of the servo drive designated by the Axis parameter.

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|-------------|------------------------|
| Axis | Axis | Axis number | 1 to 9 |

ePowerOnAll

| |
|-------------|
| ePowerOnAll |
| Edit Buffer |

This command enables the power stage of all configured servo drives.

It does not require the entry of any parameters.

Motion task editor

ePowerOff

The screenshot shows the 'ePowerOff' command editor. It has a yellow header bar with the text 'ePowerOff'. Below the header, there is a yellow box containing the number '1' next to the label 'Axis'. At the bottom of the editor is a large grey rectangular area labeled 'Edit Buffer'.

This command disables the power stage of the servo drive designated by the Axis parameter.

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|-------------|------------------------|
| Axis | Axis | Axis number | 1 to 9 |

ePowerOffAll

The screenshot shows the 'ePowerOffAll' command editor. It has a yellow header bar with the text 'ePowerOffAll'. Below the header is a large grey rectangular area labeled 'Edit Buffer'.

This command disables the power stage of all configured servo drives.

It does not require the entry of any parameters.

eHome

The screenshot shows the 'eHome' command editor. It has a yellow header bar with the text 'eHome'. Below the header, there are three yellow boxes containing the values '1', '0', and '200' next to the labels 'Axis', 'Pos', and 'Dec' respectively. At the bottom of the editor is a large grey rectangular area labeled 'Edit Buffer'.

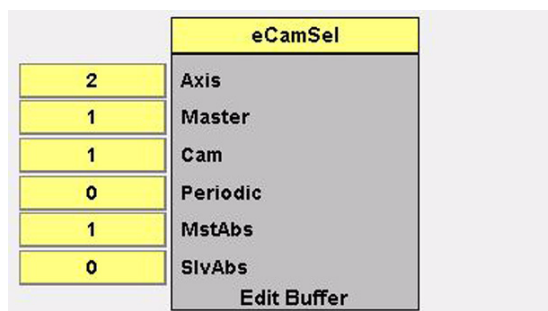
This command starts a homing operation on the axis designated in P1. The homing method used is that configured by the "**Homing Mode**" axis parameter, see page [12](#).

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|--------------|------------------------|
| Axis | Axis | Axis number | 1 to 9 |
| P1 | Pos | Position | |
| P2 | Dec | Deceleration | User units/sec^2 |

Motion task editor

Multi-axis positioning

eCamSel



This command selects and loads a cam profile to be used with an *eCamIn* command. It allows up to 8 cams to be entered.

The first two parameters are used to choose, respectively:

- The axis on which the cam will be executed, or Slave axis.
- The reference axis of this cam, or Master axis.

Parameter P3 (Periodic cam) determines whether the profile must be executed again when the position of the master goes outside the range of the cam.

- **P3 = 0 Deactivated:** When the master goes beyond the profile end position (*xEnd*), the *eCamIn* block provides the end of profile information (*EndOfProfile*). The slave is kept in its last programmed position. It should be noted that if the master re-enters the profile definition area, the slave will be controlled in accordance with the defined profile.
- **If P3 = 1 Activated:** When the master goes beyond the profile end position, the slave returns to the start of the cam profile cycle.

Parameter P4 (Absolute master position) determines the starting point of the cam according to the current position of the master

- **P4 = 1 Absolute master position:** The cam starts at the point on the profile corresponding to the current position of the master. This point may therefore be in the middle of the cam profile. In this case, as soon as the *eCamIn* block is executed, the slave will move to the position defined by the profile. If this point is located outside the cam profile definition area, an error is indicated.
- **P4 = 0 Relative master position:** The cam is started at the current position of the master, i.e., in this case, the zero point of the profile is the current position of the master. Cam profiles for which the master definition area (range from *xStart* to *xEnd*) does not contain 0 cannot be used with this mode as an error is indicated.
- Parameter P5 (Absolute slave position) determines the behavior of the slave according to the value of the *eCamIn* function *StartMode* input.

| CamIn.StartMode | absolute | relative | Ramp_In | Ramp_In_pos | Ramp_In_neg |
|-------------------------|----------|----------|------------------|----------------------|----------------------|
| Absolute slave position | absolute | relative | Rampe_In absolue | Rampe_In_pos absolue | Rampe_In_neg absolue |
| Relative slave position | relative | relative | Rampe_In relativ | Rampe_In_pos relativ | Rampe_In_neg relativ |

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|---------------------------------|------------------------------|
| Axis | Axis | Axis number | 1 to 9 |
| P1 | Master | Master axis number | 1-8 Real axes |
| | | | 9 Virtual axis |
| | | | 10 Master encoder |
| P2 | Cam | Cam profile number | 1 to 8 |
| P3 | Periodic | Periodicity of the cam | 1 = periodic cam |
| | | | 0 = non-periodic cam |
| P4 | MstAbs | Position of the absolute master | 1 = Absolute master position |
| | | | 0 = Relative master position |
| P5 | SlvAbs | Position of the absolute slave | 1 = Absolute slave position |
| | | | 0 = Relative slave position |

Motion task editor

Entering the cam points

The cam points are entered by clicking on the additional "Data" button in the navigation bar on the left of the screen. The following window is used for entering 16 points per cam.

| Number of Cam Points: 3 | | | | |
|-------------------------|-----|----------|----|----|
| XStart: 0 | | XEnd 360 | | |
| Point | dX | dY | dV | dA |
| 0 | 0 | 0 | 1 | 0 |
| 1 | 100 | 100 | 1 | 0 |
| 2 | 360 | 0 | 1 | 0 |
| 3 | 360 | 360 | 1 | 0 |
| 4 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | |
| 6 | | | | |
| 7 | | | | |
| 8 | | | | |
| 9 | | | | |
| 10 | | | | |
| 11 | | | | |
| 12 | | | | |
| 13 | | | | |
| 14 | | | | |
| 15 | | | | |

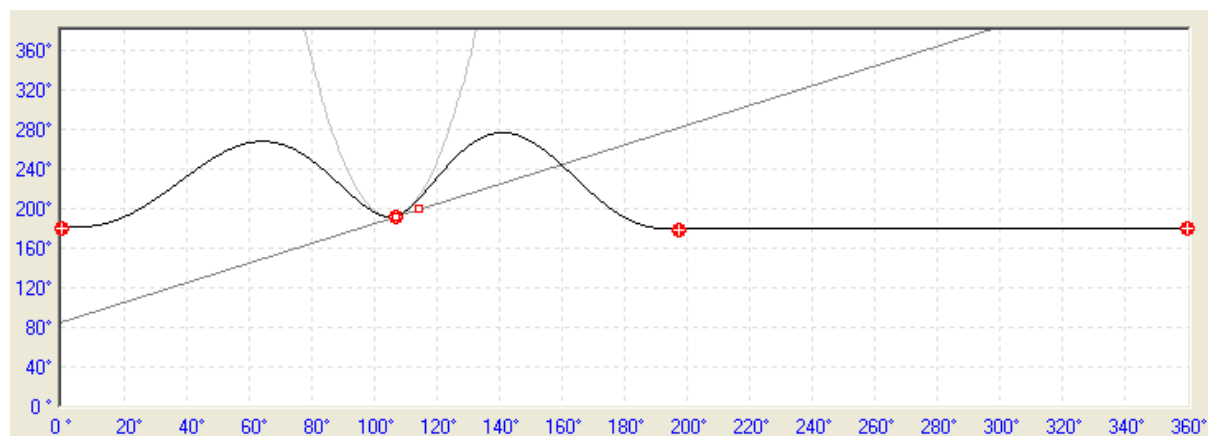
- **"Number of Cam Points"**: Number of points in the Cam profile.
- **"XStart"**: Start position.
- **"XEnd"**: End position.

Each point is characterized by its coordinates on the cam profile (dX and dY): The curve between two points is calculated using the "spline curve" interpolation method. These curves are used to define complex curves in their entirety, using a limited number of points. Each point on a spline curve is defined, in addition to its coordinates, by its tangential velocity dV and its local acceleration value dA (the first and the second position derivatives respectively).

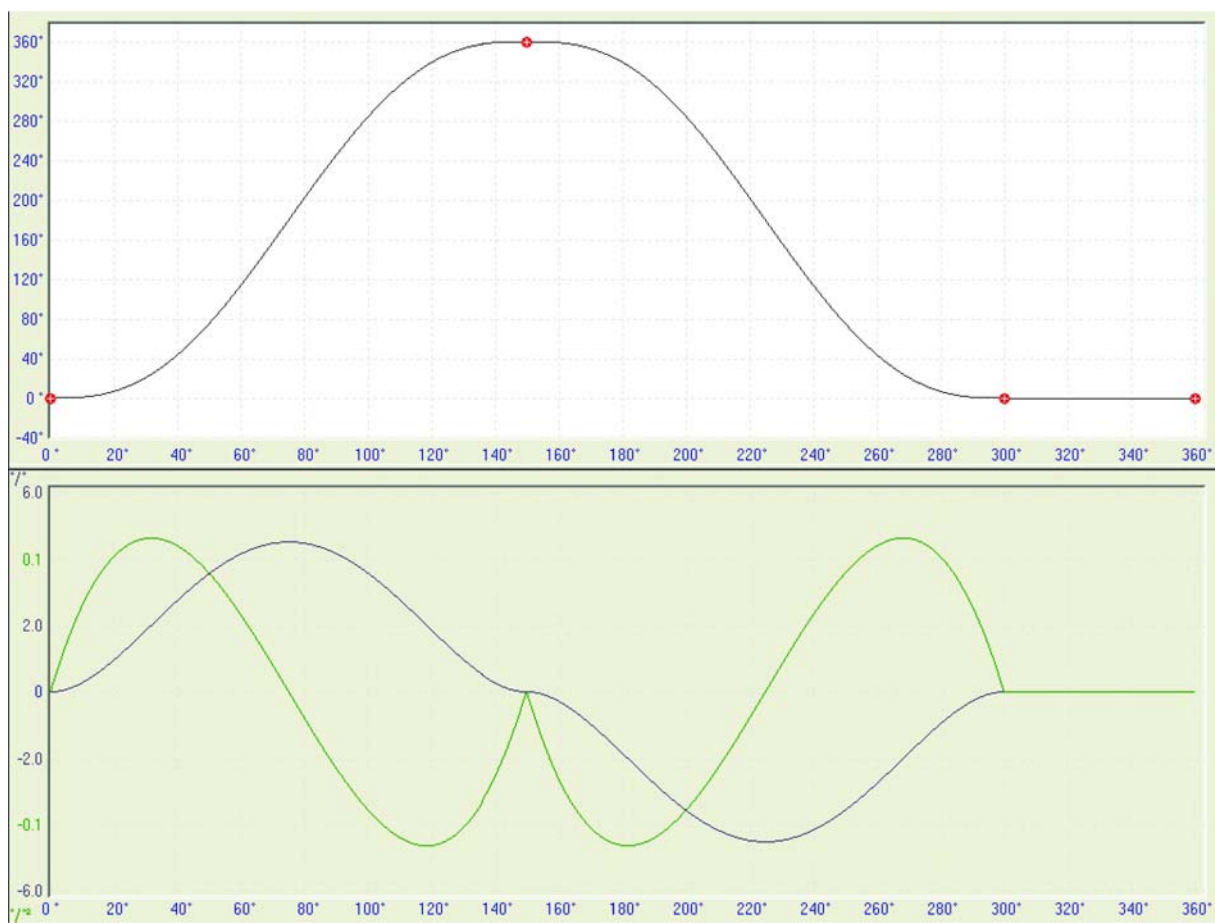
- dX: The vertical axis of the point on the cam profile. A complete cam profile is by definition between 0° and 360°
- dY: The horizontal axis of the point gives the distance between the slave axis and its home position. This is also between 0° and 360°.
- dV: The local velocity at the point. This gives the angle of the profile at this point.
- dA: The local acceleration at the point. This gives the curvature of the profile.

Motion task editor

Examples of cams



Example of a profile with velocity $dv = 1$ and acceleration $dA = 0.5$ at point 2.



Example of a profile with velocity $dv = 0$ and acceleration $dA = 0$ at point 2.

Motion task editor

eCamIn

| eCamIn | | | |
|-------------|--------|---------|---|
| 2 | Axis | | |
| 1 | Master | MOffset | 0 |
| 100 | Vel | SOffset | 0 |
| 1000 | Acc | MScale | 1 |
| 1000 | Dec | SScale | 1 |
| 0 | SMode | Cam | 1 |
| Edit Buffer | | | |

This command links a slave axis to a master axis. It can be symbolized by a mechanical clutch in opposition to the eCamOut function. The slave then executes the cam profile given in P10 by the value of "Cam".

P2 "Vel", P3 "Acc" and P4 "Dec" are characteristic parameters of the additional acceleration ramp for the Ramp_In, Ramp_In_Pos and Ramp_In_Neg modes described below.

Parameter P5 "SMode" gives the initial characteristics of the movement of the slave axis:

- **Absolute:** The cam profile is started in absolute position in relation to the current position of the slave axis. If the target position of the cam is different from the position of the slave, this results in a position jump.
- **Relative:** The cam profile is started in relative position in relation to the current position. This choice is only relevant if the cam activation position is position 0, otherwise this results in a position jump.
- **Ramp_In:** It is possible to compensate for any jumps by following an additional motion profile to reach the cam target position. This profile is configured using the P2, P3 and P4 data, and is calculated as quickly as possible.
- **Ramp_In_Pos:** In this case, an additional motion profile is activated but it is only executed in positive direction. Only for rotary axes.
- **Ramp_In_Neg:** In this case, an additional motion profile is activated but it is only executed in negative direction. Only for rotary axes.

Parameter P6 "MOffset" is used to enter a delay or an advance on the position of the master axis. This parameter is used for precise adjustment of the link between the master and the slave.

Parameter P7 "SOffset" is used to enter a delay or an advance on the position of the slave axis. This parameter is used for precise adjustment of the link between the master and the slave.

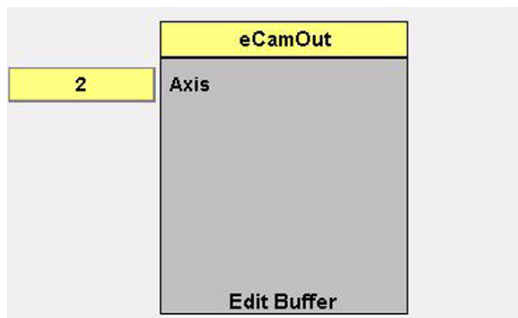
Parameter P8 "MScale" is used for scaling the position of the master axis. This parameter is used for precise adjustment of the link between the master and the slave.

Parameter P9 "SScale" is used for scaling the position of the slave axis. This parameter is used for precise adjustment of the link between the master and the slave.

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|-----------------------|------------------------|
| Axis | Axis | Axis number | 1 to 9 |
| P1 | Master | Master axis number | 1-8 Real axes |
| | | | 9 Virtual axis |
| | | | 10 Master encoder |
| P2 | Vel | Velocity | User units/Sec |
| P3 | Acc | Acceleration | User units/sec^2 |
| P4 | Dec | Deceleration | User units/sec^2 |
| P5 | SMode | Start Mode | 0: Absolute |
| | | | 1: Relative |
| | | | 2: Ramp_In |
| | | | 3: Ramp_In_Pos |
| | | | 4: Ramp_In_Neg |
| P6 | MOffset | Master offset | User units |
| P7 | SOffset | Slave offset | User units |
| P8 | MScale | Scaling of the master | Default value: 1.0 |
| P9 | SScale | Scaling of the slave | Default value: 1.0 |
| P10 | Cam | Cam number | 1 to 8 |

Motion task editor

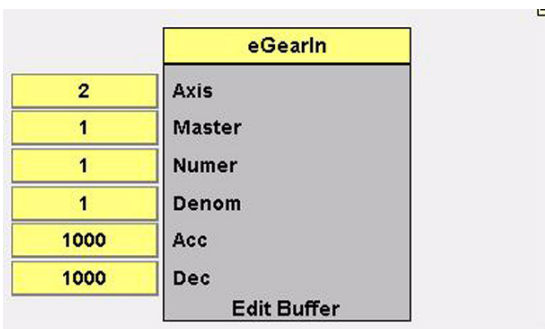
eCamOut



This command uncouples a slave axis from a master axis. If the slave axis is rotating at this moment, it will execute an infinite movement at constant velocity to its current velocity.

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|-------------|------------------------|
| Axis | Axis | Axis number | 1 to 9 |

eGearIn



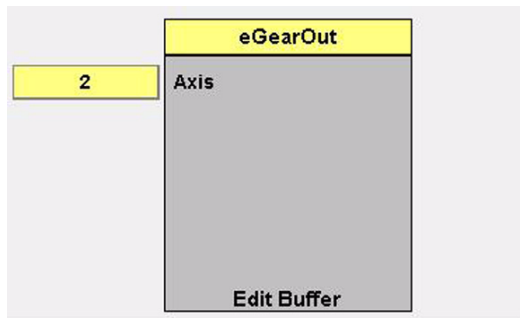
This command links a slave axis to a master axis via electronic gearing. The relationship between the velocity of the slave and that of the master is calculated using the quotient of the input parameters P2 "Numer" and P3 "Denom". These are the numerator and denominator of this ratio respectively.

The slave axis may need to accelerate or brake in order to reach the target velocity. Parameters P3 "Acc" and P4 "Dec" give the acceleration and deceleration values of such a movement.

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|--------------------|------------------------|
| Axis | Axis | Axis number | 1 to 91 to 9 |
| P1 | Master | Master axis number | 1-8 Real axes |
| | | | 9 Virtual axis |
| | | | 10 Master encoder |
| P2 | Numer | Numerator | |
| P3 | Denom | Denominator | |
| P4 | Acc | Acceleration | User units/sec^2 |
| P5 | Dec | Deceleration | User units/sec^2 |

Motion task editor

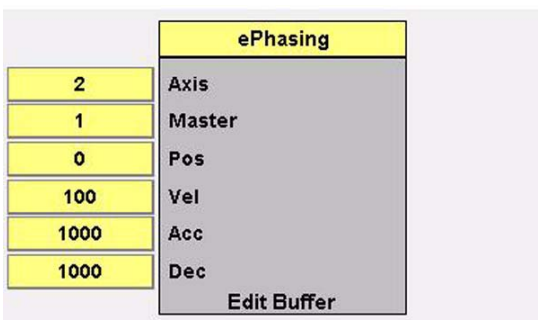
eGearOut



This command uncouples a slave axis from a master axis. If the slave axis is rotating at this moment, it will execute an infinite movement at constant velocity to its current velocity.

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|-------------|------------------------|
| Axis | Axis | Axis number | 1 to 9 |

ePhasing



This command links a slave axis to a master axis in phase. In other words both axes rotate at the same velocity.

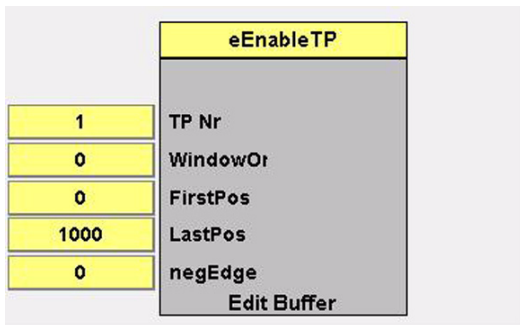
Their respective positions are given by parameter P2 "Pos". Therefore if this parameter is zero, both axes not only rotate at the same velocity, but their positions are also identical at all times.

In order to bring the axes into phase, the slave axis may have to follow an additional motion profile defined by the P3 "Vel", P4 "Acc" and P5 "Dec" parameters.

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|--------------------|------------------------|
| Axis | Axis | Axis number | 1 to 9 |
| P1 | Master | Master axis number | 1-8 Real axes |
| | | | 9 Virtual axis |
| | | | 10 Master encoder |
| P2 | Pos | Position | User units |
| P3 | Vel | Velocity | User units/Sec |
| P4 | Acc | Acceleration | User units/sec^2 |
| P5 | Dec | Deceleration | User units/sec^2 |

Motion task editor

eEnableTP



This function triggers the operation of one of the two Lexium Controller fast capture inputs. This block requires the following parameters to be entered:

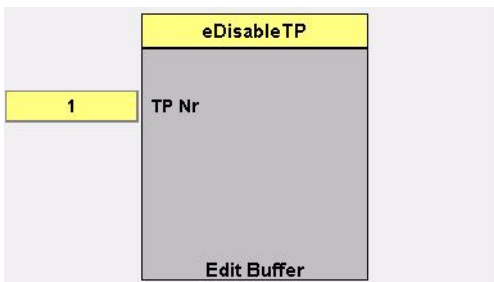
Parameter P1 "TP Nr" designates the number of the selected fast capture input. There are two capture inputs available on the controller.

Parameter P2 "Window On" activates the position window in which the fast input is activated. If this parameter is zero, the capture input will always be enabled. The limits of this window are defined by parameters P3 "FirstPos" and P4 "LastPos", which give the start and end of activation position values of the fast capture input respectively.

Parameter P5 "negEdge" is used to reverse the polarity of the capture input. By default, the position capture is carried out on a rising edge (negEdge=0). If this parameter is enabled (negEdge = 1) the position capture is carried out on a falling edge of the input.

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|-----------------------------|------------------------|
| Axis | Axis | Axis number | 1 to 9 |
| P1 | TP Nr | Number of the capture input | 1 or 2 |
| P2 | WindowOn | Activity window | 1-8 Real axes |
| | | | 9 Virtual axis |
| | | | 10 Master encoder |
| P3 | FirstPos | Starting position | User units |
| P4 | LastPos | Final position | User units |
| P5 | negEdge | Polarity of capture | 0 = rising edge |
| | | | 1 = falling edge |

eDisableTP



This function is used to deactivate one of the two Lexium Controller fast capture inputs.

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|-----------------------------|------------------------|
| P1 | TP Nr | Number of the capture input | 1 or 2 |

Motion task editor

Application function blocks

eFShearIn

| eFShearIn | | | |
|-----------|-------------|-------------|---|
| 2 | Axis | | |
| 1 | Master | UseNextTP | 0 |
| 360 | LengthToCut | TP_StdInput | 0 |
| 0 | TeachIndex | WSSelect | 0 |
| 0 | SynOff | q_InSync | 0 |
| 0 | Immed.Cut | q_WSPosReac | 0 |
| | | TimeOut | 0 |

This command is used to launch the "Flying Shear" application function block. eStop is not accepted after a eFShearIn. eFShearStop or eFShearOut has to be used.

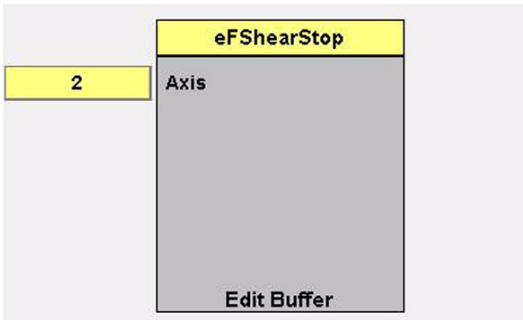
The flying shear parameters are entered by clicking on the additional "Parameter" button in the navigation bar on the left of the screen.

Please refer to the **Motion Pro - Programming manual** for more information on application function block parameters.

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|-------------|---|-------------------------|
| Axis | Axis | Axis number | 1 to 9 |
| P1 | Master | Master axis number | 1-8 Real axes |
| | | | 9 Virtual axis |
| | | | 10 Master encoder |
| P2 | LengthToCut | Length to cut if TeachIndex=0 | User Units |
| P3 | TeachIndex | Index of Teach List to be used as the length to cut. Allows to modify length to cut on the fly | 0 = LengthToCut from P2 |
| | | | 1 to 32 |
| P4 | SynOff | Input used to trigger SynOff | Input 0-10 |
| | | | 0 = not used |
| P5 | Immed.Cut | Input used to trigger Immed.Cut | Input 0-10 |
| | | | 0 = not used |
| P6 | UseNextTP | Input used to trigger UseNextTP | Input 0-10 |
| | | | 0 = not used |
| P7 | TP_StdInput | Input used to trigger TP_StdInput | Input 0-10 |
| | | | 0 = not used |
| P8 | WSSelect | Input used to trigger WSSelect | Input 0-10 |
| | | | 0 = always ColdStart |
| P9 | q_InSync | Output for q_InSync | OutPut 0-8 |
| | | | 0 = not used |
| P10 | q_WSPosRea | Output for q_WSPosRea | OutPut 0-8 |
| | | | 0 = not used |

Motion task editor

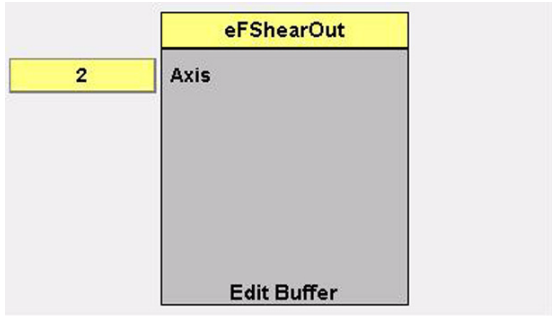
eFShearStop



eFShearStop stops the execution (i_xExecute to FALSE) of the Flying Shear application function bloc. Slave move to rest position.

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|-------------------|------------------------|
| Axis | Axis | Slave axis number | 1 to 9 |

eFShearOut



eFShearOut disables (i_xEnable to FALSE) the Flying Shear application function bloc. Slave is stopped and do not move to rest position.

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|-------------------|------------------------|
| Axis | Axis | Slave axis number | 1 to 9 |

Motion task editor

eRKnifeIn

| eRKnifeIn | | | |
|-----------|-------------|-------------|---|
| 2 | Axis | | |
| 1 | Master | UseNextTP | 0 |
| 360 | LengthToCut | TP_StdInput | 0 |
| 0 | TeachIndex | WSSelect | 0 |
| | | q_InSync | 0 |
| 0 | Immed.Cut | q_WSPosReac | 0 |
| | | TimeOut | 0 |

This command launches the "Rotary Knife" application function block. eStop is not accepted after a eRKnifeIn. eRKnifeStop or eRKnifeOut has to be used.

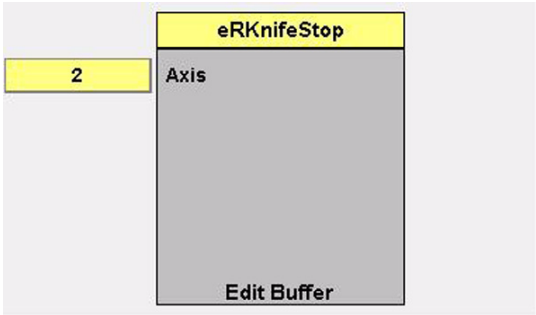
The rotary knife parameters are entered by clicking on the additional "Parameter" button in the navigation bar on the left of the screen.

Please refer to the **Motion Pro - Programming manual** for more information on application function block parameters.

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|-------------|---|--|
| Axis | Axis | Axis number | 1 to 9 |
| P1 | Master | Master axis number | 1-8 Real axes 9 Virtual axis 10 Master encoder |
| P2 | LengthToCut | Length to cut if TeachIndex=0 | User Units |
| P3 | TeachIndex | Index of Teach List to be used as the length to cut. Allows to modify length to cut on the fly | 0 = LengthToCut from P2 1 to 32 |
| P5 | Immed.Cut | Input used to trigger Immed.Cut | Input 0-10 0 = not used |
| P6 | UseNextTP | Input used to trigger UseNextTP | Input 0-10 0 = not used |
| P7 | TP_StdInput | Input used to trigger TP_StdInput | Input 0-10 0 = not used |
| P8 | WSSelect | Input used to trigger WSSelect | Input 0-10 0 = always ColdStart |
| P9 | q_InSync | Output for q_InSync | OutPut 0-8 0 = not used |
| P10 | q_WSPosRea | Output for q_WSPosRea | OutPut 0-8 0 = not used |

Motion task editor

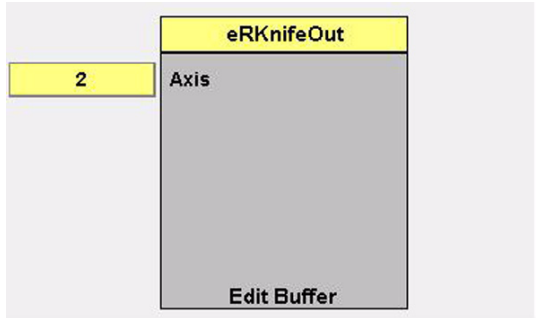
eRKnifeStop



eRKnifeStop stops the execution (i_xExecute to FALSE) of the Rotary Knife application function bloc. Slave move to rest position.

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|-------------------|------------------------|
| Axis | Axis | Slave axis number | 1 to 9 |

eRKnifeOut



eRKnifeOut disables (i_xEnable to FALSE) the Rotary Knife application function bloc. Slave is stopped and do not move to rest position.

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|-------------------|------------------------|
| Axis | Axis | Slave axis number | 1 to 9 |

Motion task editor

eClampIn

| | eClampIn | | |
|---|------------|-------------|---|
| 1 | Axis | | |
| 0 | Close | q_Closed | 0 |
| 0 | Open | q_Busy | 0 |
| 0 | StdTPInput | q_InFeed | 0 |
| 0 | Teach | q_OutFeed | 0 |
| 0 | TeachParam | q_BehindPos | 0 |
| | | TimeOut | 0 |

This command is used to launch the "eClampIn" application function block used for clamping with torque limiting.

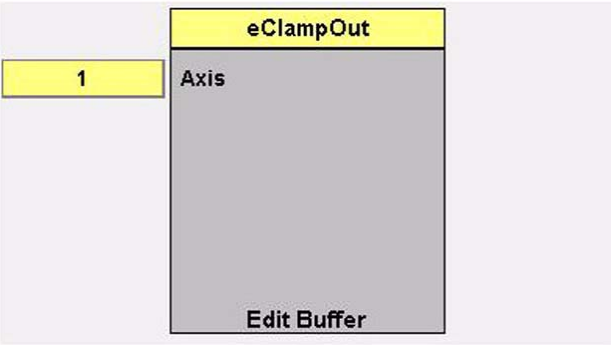
These block parameters are entered by clicking on the additional "Parameter" button of the navigation bar on the left of the screen.

Please refer to the **Motion Pro - Programming manual** for more information on application function block parameters.

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|-------------|----------------------------------|--|
| Axis | Axis | Slave axis number | 1 to 9 |
| P1 | Close | Input used to trigger Close | Input 0-10 0 = not used |
| P2 | Open | Input used to trigger Open | Input 0-10 0 = not used |
| P3 | StdTPInput | Input used to trigger StdTPInput | Input 0-10 0 = not used |
| P4 | Teach | Input used to trigger Teach | Input 0-10 0 = not used |
| P5 | TeachParam | Teach parameter | 0 = no action 1 = Out feed position 2 = In feed position 3 = Near position 4 = Open position |
| P6 | q_Closed | Output used for q_Closed | OutPut 0-8 0 = not used |
| P7 | q_Busy | Output used for q_Busy | OutPut 0-8 0 = not used |
| P8 | q_OutFeed | Output used for q_OutFeed | OutPut 0-8 0 = not used |
| P9 | q_Closed | Output used for q_Closed | OutPut 0-8 0 = not used |
| P10 | q_BehindPos | Output used for q_BehindPos | OutPut 0-8 0 = not used |

Motion task editor

eClampOut



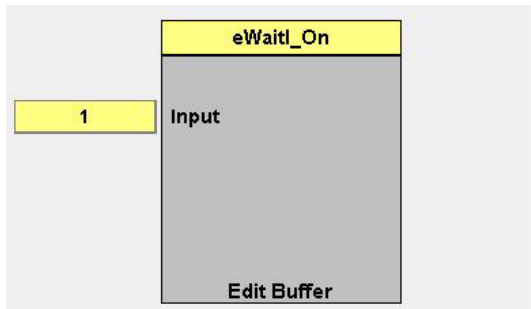
This command stops the "eClampIn" application function block.

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|-------------------|------------------------|
| Axis | Axis | Slave axis number | 1 to 9 |

Motion task editor

Logic commands

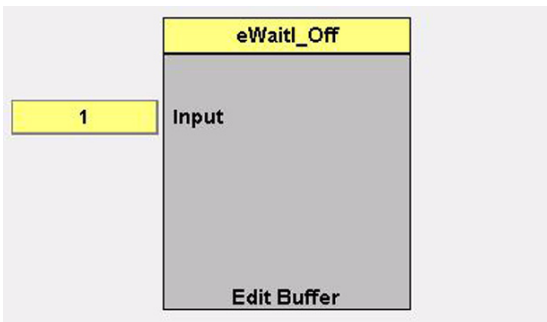
eWaitl_On



This command waits for a Lexium Controller logic input to change to state 1

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|--------------------------|------------------------|
| P1 | Input | Number of the input used | 1 to 8 |

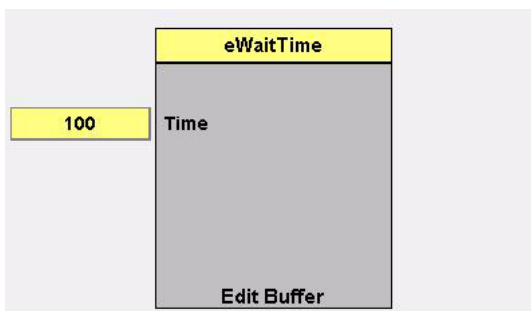
eWaitl_Off



This command waits for a Lexium Controller logic input to change to state 0

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|--------------------------|------------------------|
| P1 | Input | Number of the input used | 1 to 8 |

eWaitTime

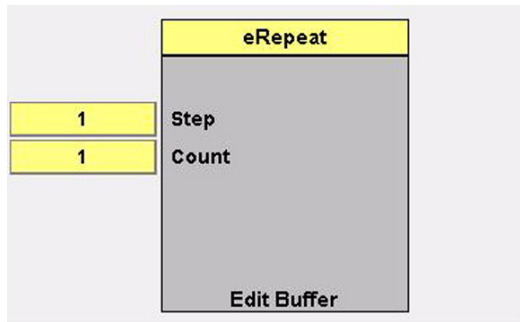


This command executes a time delay of a defined duration.

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|----------------------------|------------------------|
| P1 | Time | Duration of the time delay | 1-100000 ms |

Motion task editor

eRepeat

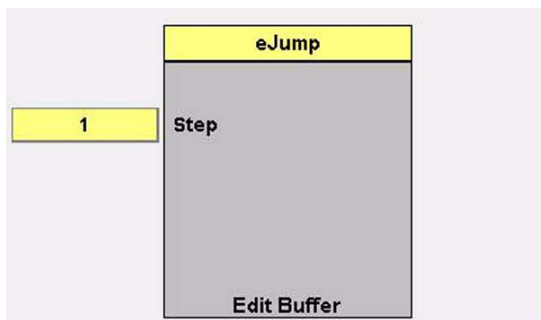


This command repeats one or more steps in the motion sequence table. The step from which the table must be executed again must be less than the step of the *eRepeat* command. There can be up to 8 nested repetitions.

Parameter P2 "Count" is used to limit the number of repetitions of the command.

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|---|------------------------|
| P1 | Step | Step from which the sequence must be repeated | 1 to 75 |
| P2 | Count | Number of repetitions | |

eJump



This command executes an unconditional jump to the step defined in P1 "Step".

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|--------------------------------|------------------------|
| P1 | Step | Number of the destination step | 1 to 75 |

Motion task editor

eJumpl_On

The screenshot shows the configuration interface for the 'eJumpl_On' command. It features a yellow header bar with the command name. Below this is a grey rectangular area containing two labels: 'Input' and 'Step'. To the left of the 'Input' label is a yellow box containing the value '1'. Similarly, to the left of the 'Step' label is a yellow box containing the value '1'. At the bottom of the grey area is an 'Edit Buffer' label.

This Command executes a conditional jump to a step number (P2 "Step"). If the input selected in P1 "Input" is at state 1, the jump is executed. Otherwise the next step is executed.

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|--------------------------------|------------------------|
| P1 | Input | Number of the input used | 1 to 8 |
| P2 | Step | Number of the destination step | 1 to 75 |

eJumpl_Off

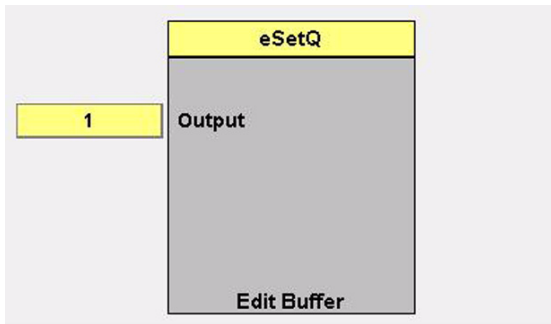
The screenshot shows the configuration interface for the 'eJumpl_Off' command. It features a yellow header bar with the command name. Below this is a grey rectangular area containing two labels: 'Input' and 'Step'. To the left of the 'Input' label is a yellow box containing the value '1'. Similarly, to the left of the 'Step' label is a yellow box containing the value '1'. At the bottom of the grey area is an 'Edit Buffer' label.

This Command executes a conditional jump to a step number (P2 "Step"). If the input selected in P1 "Input" is at state 0, the jump is executed. Otherwise the next step is executed.

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|--------------------------------|------------------------|
| P1 | Input | Number of the input used | 1 to 8 |
| P2 | Step | Number of the destination step | 1 to 75 |

Motion task editor

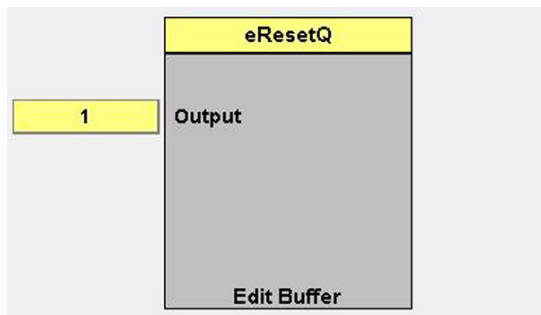
eSetQ



This command forces the selected logic output to state 1.

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|---------------------------|------------------------|
| P1 | Output | Number of the output used | 1 to 8 |

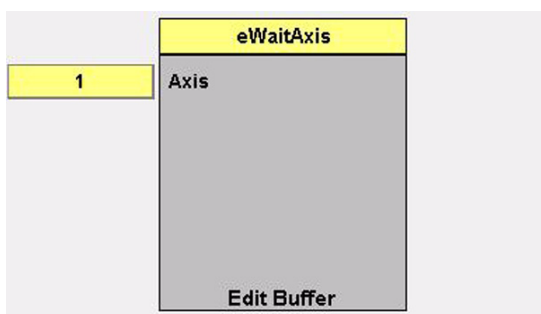
eResetQ



This command forces the selected logic output to state 0.

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|---------------------------|------------------------|
| P1 | Output | Number of the output used | 1 to 8 |

eWaitAxis

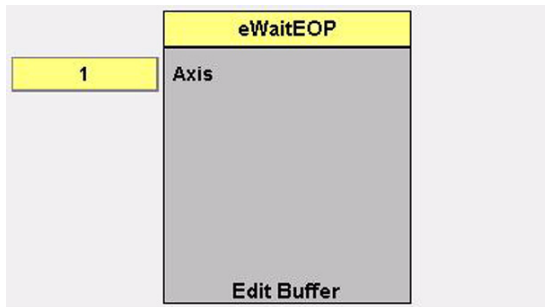


This command waits for the last movement command executed on the axis to be completed (see page [65](#)).

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|-------------|------------------------|
| Axis | Axis | Axis number | 1 to 9 |

Motion task editor

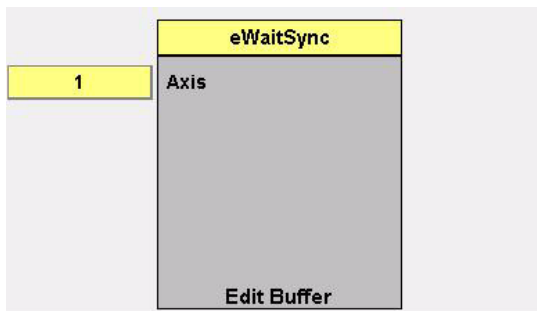
eWaitEOP



This command waits for the end of execution of a cam profile (EndOfProfile). This is the end of a *eCamIn* command (see page [65](#)).

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|-------------|------------------------|
| Axis | Axis | Axis number | 1 to 9 |

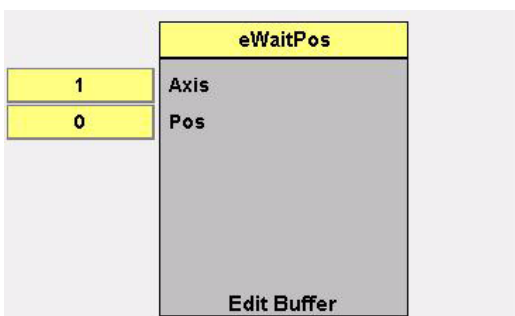
eWaitSync



This command waits for the "InSync" (Synchronized) state of an *eCamIn*, *eGearIn*, *eFShearIn* or *eRKnifeIn* command (see page [65](#)).

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|-------------|------------------------|
| Axis | Axis | Axis number | 1 to 9 |

eWaitPos



This command waits for the position of an axis to be greater than/equal to P1 "Pos" (see page [65](#)).

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|---------------------------|---------------------------------|
| Axis | Axis | Axis number | 1 to 9 |
| P1 | Pos | Position | User units |
| P2 | Range | Allows to define interval | 0 = upper or equal 1 = lower |

Motion task editor

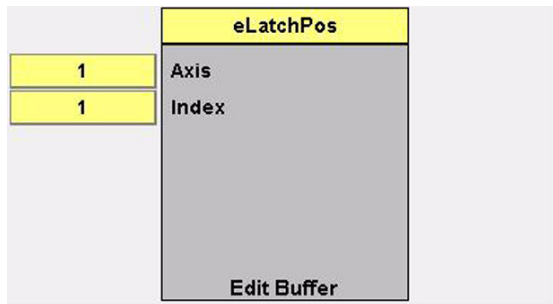
eWaitDist



This command waits for the distance traveled by the axis to be greater than distance P1, in both directions (see page [65](#)).

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|-------------|------------------------|
| Axis | Axis | Axis number | 1 to 9 |
| P1 | Dist | Distance | User units |

eLatchPos

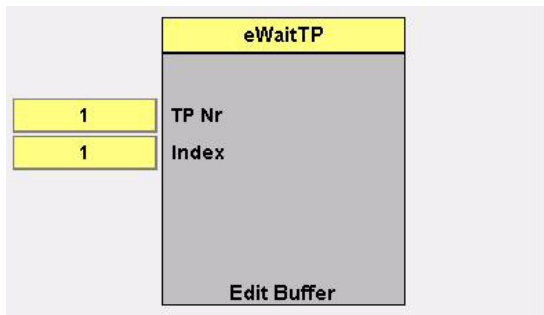


This command stores the position of the axis in the position register using the teach function.

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|-------------------|------------------------|
| Axis | Axis | Axis number | 1 to 9 |
| P1 | Index | Position register | 1 to 32 |

Motion task editor

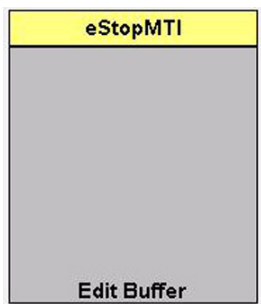
eWaitTP



This command is used to wait for a Touch Probe input. The position is stored in the index (see page 65).

| Parameter | Identifier | Function | Adjustment range/Units |
|-----------|------------|---------------------------|------------------------|
| Axis | Axis | Axis number | 1 to 9 |
| P1 | TP Nr | Capture input number | 1 to 2 |
| P2 | Index | Teach function list index | 1 to 32 |

eStopMTI



This command is used to stop execution of the motion sequence.

Motion task editor

This table describes in which situation the eWait functions should be used. Some functions have the same behavior and can be used equally.

| | eWaitAxis | eWaitEOP | eWaitSync | eWaitPos | eWaitDist | eWaitTP |
|---------------|------------|----------|-----------|------------|------------|---------|
| eMoveAbs, | Yes | No | No | Yes | Yes | No |
| eMoveRel, | Yes | No | No | Yes | Yes | No |
| eMoveAdd, | Yes | No | No | Yes | Yes | No |
| eMoveVel, | Yes | No | No | Yes | Yes | No |
| eMoveContRel, | Yes | No | No | Yes | Yes | No |
| eMoveContAbs, | Yes | No | No | Yes | Yes | No |
| eSTOP, | Compulsory | No | No | Compulsory | Compulsory | No |
| ePowerOn, | No | No | No | No | No | No |
| ePowerOnAll, | No | No | No | No | No | No |
| ePowerOff, | No | No | No | No | No | No |
| ePowerOffAll, | No | No | No | No | No | No |
| eHome, | Yes | No | No | Yes | Yes | No |
| eCamSel, | No | No | No | No | No | No |
| eCamIn, | Yes | Yes | Yes | Yes | Yes | No |
| eCamOut, | No | No | No | No | No | No |
| eGearIn, | No | No | Yes | No | No | No |
| eGearOut, | No | No | No | No | No | No |
| eFShearIn, | No | Yes | Yes | No | No | No |
| eFShearOut, | No | No | No | No | No | No |
| eFShearStop, | Yes | No | No | Yes | Yes | No |
| eRKnifeIn, | No | Yes | Yes | No | No | No |
| eRKnifeOut, | No | No | No | No | No | No |
| eRKnifeStop, | Yes | No | No | Yes | Yes | No |
| eMoveSupl, | Yes | No | No | Yes | Yes | No |
| ePhasing, | Yes | No | No | Yes | Yes | No |
| eSetPos, | No | No | No | No | No | No |
| eWritePar, | No | No | No | No | No | No |
| eClampIn, | Yes | No | No | Yes | No | No |
| eClampOut, | No | No | No | No | No | No |
| eEnableTP | No | No | No | No | No | Yes |

Motion task editor

Execution rules

General rules for executing commands using the motion task table interpreter

| | CMD | Axis | Name1 | Value1 |
|----|-------------|------|-------|--------|
| 1 | ePowerOnAll | 0 | | 0 |
| 2 | eMoveAbs | 1 | Pos | 90 |
| 3 | eMoveAbs | 2 | Pos | 90 |
| 4 | eMoveAbs | 3 | Pos | 90 |
| 5 | eWaitAxis | 1 | | 0 |
| 6 | eWaitAxis | 2 | | 0 |
| 7 | eWaitAxis | 3 | | 0 |
| 8 | eMoveAbs | 1 | Pos | 270 |
| 9 | eMoveAbs | 2 | Pos | 180 |
| 10 | eMoveAbs | 3 | Pos | 180 |
| 11 | eWaitAxis | 1 | | 0 |
| 12 | eWaitAxis | 2 | | 0 |

The interpreter executes all commands in ascending step number order in the same cycle. This means that two consecutive instructions in the table will be executed simultaneously. Only the following conditions stop the execution of the table:

- The next command is zero (*eNoCmd*). This is usually the end of the list, then the table starts again automatically from the beginning.
- A wait command is executed (*eWait●●●* type) and the next command is not a wait command. If a number of wait commands are placed one after the other, they are all executed in the same cycle time. The interpreter stops at the last wait command and waits for the missing condition.

These stopping rules are described in detail in the next section, Examples of tables, on page [76](#).

Motion task editor

Sequencing of movements

There are various ways of sequencing movements. Depending on the requirements of the application, the movements must be executed in one of the following ways:

- **Point-to-point:** The axis moves to each target position with zero velocity and the positions follow on one after the other.
- **Interruptive:** The axis executes a standard movement and must, due to a reason connected with the sequence of the machine, change position.
- **Linked to previous position:** The axis arrives at the first target position with the velocity of the first movement.
- **Linked to next position:** The axis arrives at the first target position with the velocity of the second movement.

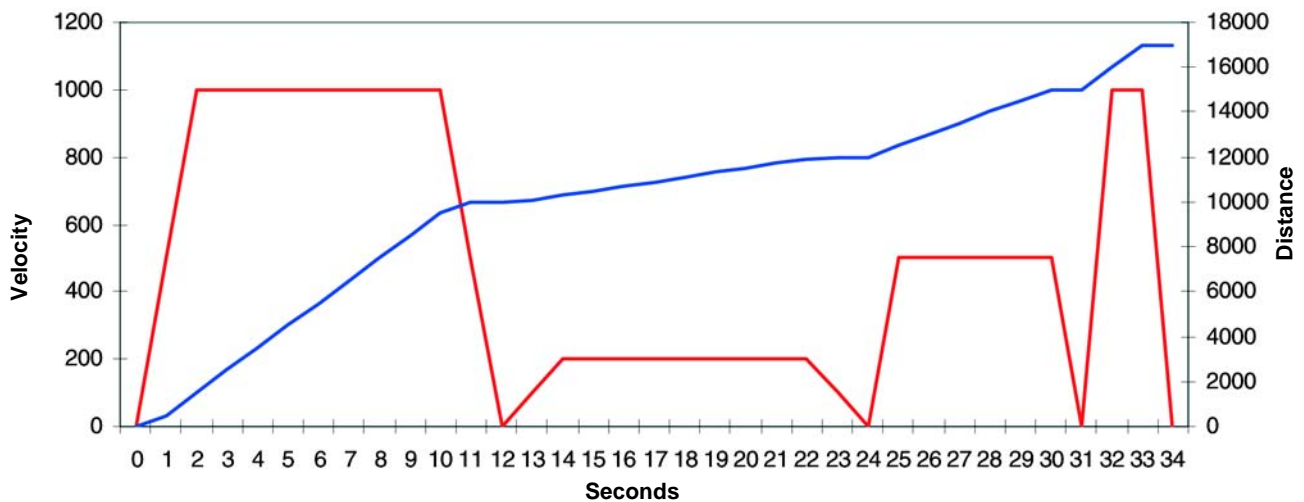
Step by step: Point-to-Point sequencing

In Point-to-point mode the movements of the axis can be linked together immediately, or this may be dependent on time or a logic input.

Linear axis with a trapezoid motion profile. Positions linked together immediately.

The parameter P7 "Buffered" function in the eMoveAbs block is used to avoid eWaitAxis between each eMoveAbs.

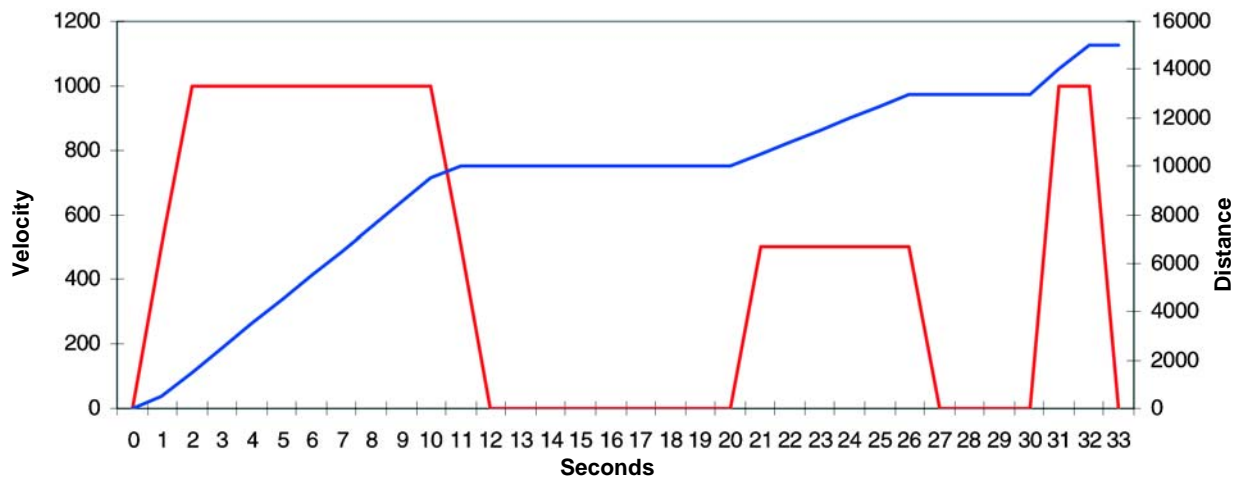
| Step | Command | Axis | Param.1 | Val. P1 | Param. 2 | Val. P2 | Param. 3 | Val. P3 | Param. 4 | Val. P4 | Param. 5 | Val. P5 | Param. 6 | Val. P6 | Param. 7 | Val. P7 |
|------|------------|------|---------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| STEP | CMD | Axis | P1 Name | P1 Value | P2 Name | P2 Value | P3 Name | P3 Value | P4 Name | P4 Value | P5 Name | P5 Value | P6 Name | P6 Value | P7 Name | P7 Value |
| 1 | Power On | 1 | | | | | | | | | | | | | | |
| 2 | eMove Abs | 1 | Pos | 10000 | Vel | 1000 | Acc | 500 | Dec | 500 | Dir | 1 | Reg | 0 | Buffer | 0 |
| 3 | eMove Abs | 1 | Pos | 2000 | Vel | 200 | Acc | 100 | Dec | 100 | Dir | 1 | Reg | 0 | Buffer | 1 |
| 4 | eMove Abs | 1 | Pos | 3000 | Vel | 500 | Acc | 500 | Dec | 500 | Dir | 1 | Reg | 0 | Buffer | 1 |
| 5 | eMove Abs | 1 | Pos | 2000 | Vel | 1000 | Acc | 1000 | Dec | 500 | Dir | 1 | Reg | 0 | Buffer | 1 |
| 6 | eWait Axis | 1 | | | | | | | | | | | | | | |



Motion task editor

Linear axis with a trapezoid motion profile. Positions dependent on time.

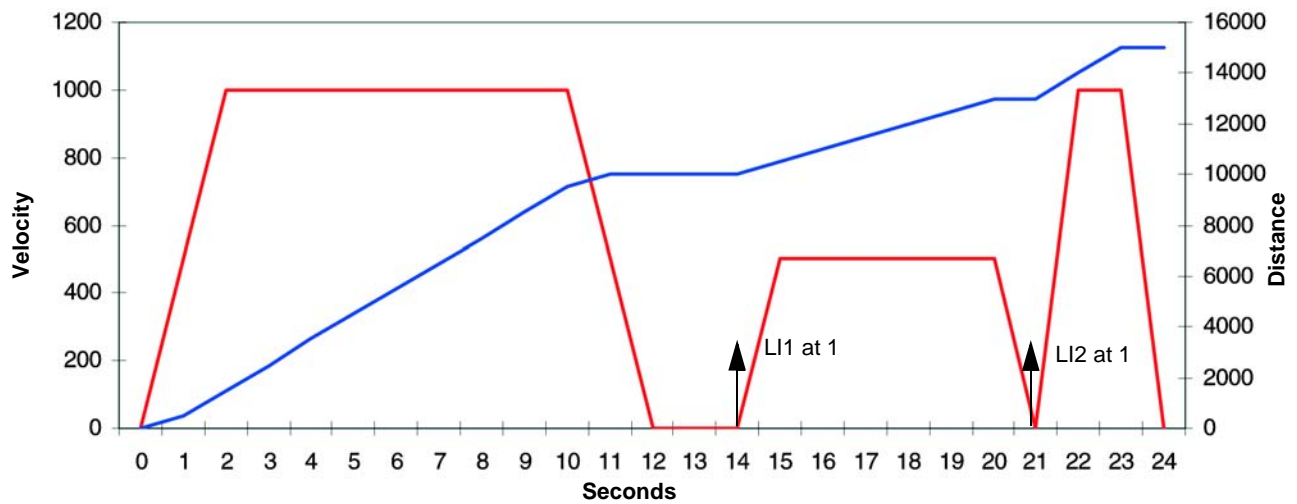
| Step | Command | Axis | Param.1 | Val. P1 | Param. 2 | Val. P2 | Param. 3 | Val. P3 | Param. 4 | Val. P4 | Param. 5 | Val. P5 | Param. 6 | Val. P6 | Param. 7 | Val. P7 |
|------|------------|------|---------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| STEP | CMD | Axis | P1 Name | P1 Value | P2 Name | P2 Value | P3 Name | P3 Value | P4 Name | P4 Value | P5 Name | P5 Value | P6 Name | P6 Value | P7 Name | P7 Value |
| 1 | Power On | 1 | | | | | | | | | | | | | | |
| 2 | eMove Abs | 1 | Pos | 10000 | Vel | 1000 | Acc | 500 | Dec | 500 | Dir | 1 | Reg | 0 | Buffer | 0 |
| 3 | eWait Time | 0 | | | Time | 20 | | | | | | | | | | |
| 4 | eMove Abs | 1 | Pos | 3000 | Vel | 500 | Acc | 500 | Dec | 500 | Dir | 1 | Reg | 0 | Buffer | 0 |
| 5 | eWait Time | 0 | | | Time | 10 | | | | | | | | | | |
| 6 | eMove Abs | 1 | Pos | 2000 | Vel | 1000 | Acc | 1000 | Dec | 500 | Dir | 1 | Reg | 0 | Buffer | 0 |
| 7 | eWait Axis | 1 | | | | | | | | | | | | | | |



Motion task editor

Linear axis with a trapezoid motion profile. Positions dependent on a logic input.

| Step | Command | Axis | Param.1 | Val. P1 | Param. 2 | Val. P2 | Param. 3 | Val. P3 | Param. 4 | Val. P4 | Param. 5 | Val. P5 | Param. 6 | Val. P6 | Param. 7 | Val. P7 |
|------|---------------|------|---------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| STEP | CMD | Axis | P1 Name | P1 Value | P2 Name | P2 Value | P3 Name | P3 Value | P4 Name | P4 Value | P5 Name | P5 Value | P6 Name | P6 Value | P7 Name | P7 Value |
| 1 | Power On | 1 | | | | | | | | | | | | | | |
| 2 | eMove Abs | 1 | Pos | 10000 | Vel | 1000 | Acc | 500 | Dec | 500 | Dir | 1 | Reg | 0 | Buffer | 0 |
| 3 | eWaitL _On | 0 | | | LI | 1 | | | | | | | | | | |
| 4 | eMove Abs | 1 | Pos | 3000 | Vel | 500 | Acc | 500 | Dec | 500 | Dir | 1 | Reg | 0 | Buffer | 0 |
| 5 | eWaitL _On | 0 | | | LI | 2 | | | | | | | | | | |
| 6 | eMove Abs | 1 | Pos | 2000 | Vel | 1000 | Acc | 1000 | Dec | 500 | Dir | 1 | Reg | 0 | Buffer | 0 |
| 7 | eWait Axis | 1 | | | | | | | | | | | | | | |



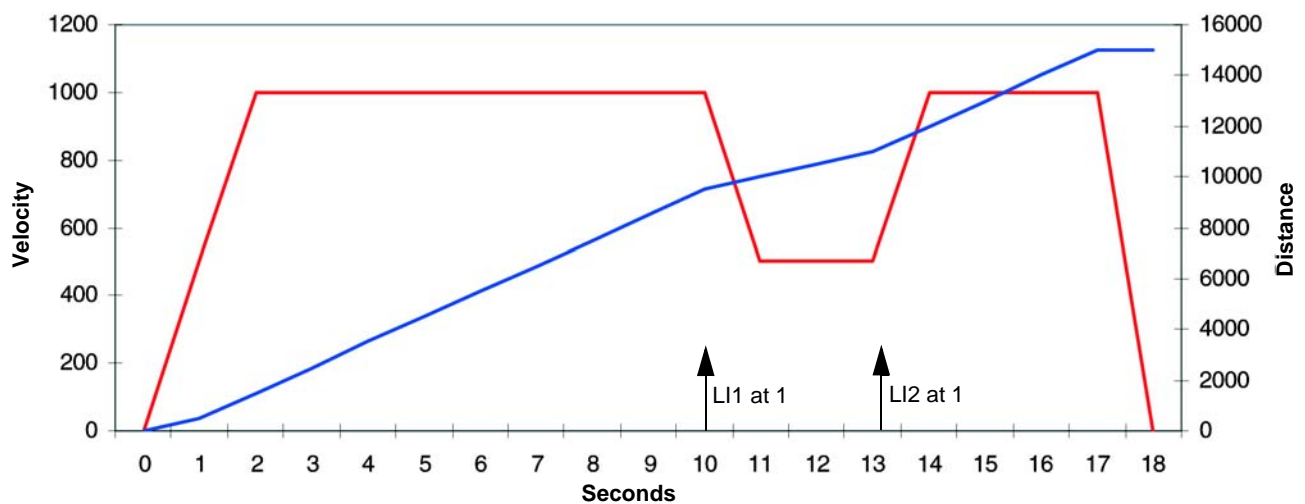
Motion task editor

Step by step: Interruptive sequencing

In interruptive mode the movements of the axis can be interrupted immediately by a new movement. This is the default mode of the Lexium Controller.

Linear axis with a trapezoid motion profile. Interrupts dependent on time.

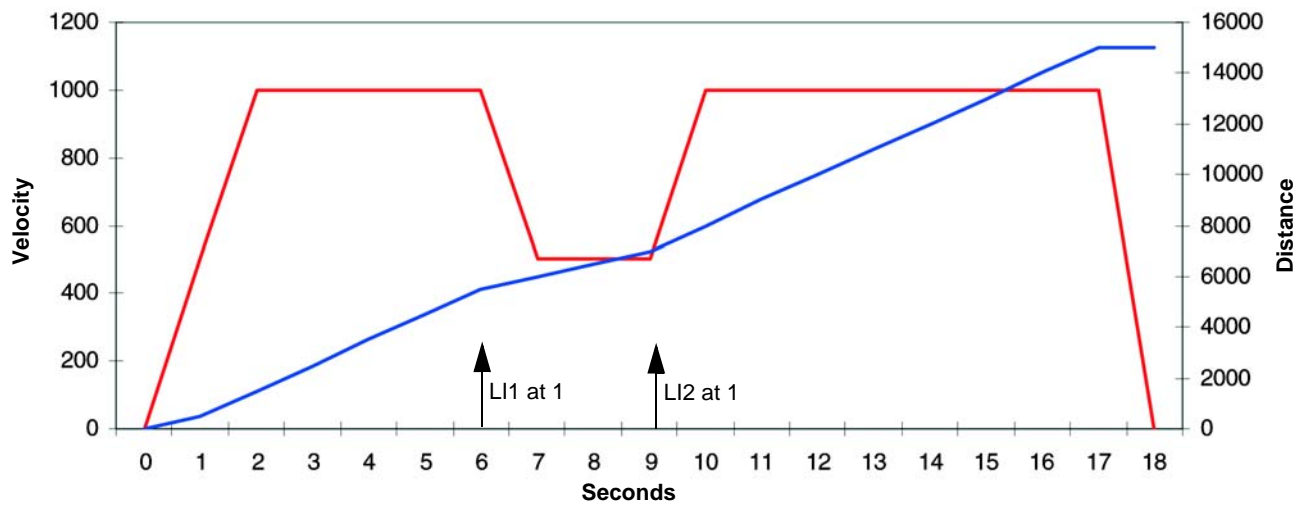
| Step | Command | Axis | Param.1 | Val. P1 | Param. 2 | Val. P2 | Param. 3 | Val. P3 | Param. 4 | Val. P4 | Param. 5 | Val. P5 | Param. 6 | Val. P6 | Param. 7 | Val. P7 |
|------|------------|------|---------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| STEP | CMD | Axis | P1 Name | P1 Value | P2 Name | P2 Value | P3 Name | P3 Value | P4 Name | P4 Value | P5 Name | P5 Value | P6 Name | P6 Value | P7 Name | P7 Value |
| 1 | Power On | 1 | | | | | | | | | | | | | | |
| 2 | eMove Vel | 1 | Vel | 1000 | Acc | 500 | Dec | 500 | | | | | | | | |
| 3 | eWait Time | 0 | | | Time | 10 | | | | | | | | | | |
| 4 | eMove Vel | 1 | Pos | 3000 | Vel | 500 | Acc | 500 | Dec | 500 | Dir | 1 | Reg | 0 | Buffer | 0 |
| 5 | eWait Time | 0 | | | Time | 3 | | | | | | | | | | |
| 6 | eMove Abs | 1 | Pos | 2000 | Vel | 1000 | Acc | 1000 | Dec | 500 | Dir | 1 | Reg | 0 | Buffer | 0 |
| 7 | eWait Axis | 1 | | | | | | | | | | | | | | |



Motion task editor

Linear axis with a trapezoid motion profile. Interrupts dependent on a logic input.

| Step | Command | Axis | Param.1 | Val. P1 | Param. 2 | Val. P2 | Param. 3 | Val. P3 | Param. 4 | Val. P4 | Param. 5 | Val. P5 | Param. 6 | Val. P6 | Param. 7 | Val. P7 |
|------|------------|------|---------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| STEP | CMD | Axis | P1 Name | P1 Value | P2 Name | P2 Value | P3 Name | P3 Value | P4 Name | P4 Value | P5 Name | P5 Value | P6 Name | P6 Value | P7 Name | P7 Value |
| 1 | Power On | 1 | | | | | | | | | | | | | | |
| 2 | eMove Rel | 1 | Pos | 10000 | Vel | 1000 | Acc | 500 | Dec | 500 | Dir | 1 | Reg | 0 | Buffer | 0 |
| 3 | eWaitI _On | 0 | | | LI | 1 | | | | | | | | | | |
| 4 | eMove Rel | 1 | Pos | 3000 | Vel | 500 | Acc | 500 | Dec | 500 | Dir | 1 | Reg | 0 | Buffer | 0 |
| 5 | eWaitI _On | 0 | | | LI | 2 | | | | | | | | | | |
| 6 | eMove Rel | 1 | Pos | 8000 | Vel | 1000 | Acc | 1000 | Dec | 1000 | Dir | 1 | Reg | 0 | Buffer | 0 |
| 7 | eWait Axis | 1 | | | | | | | | | | | | | | |

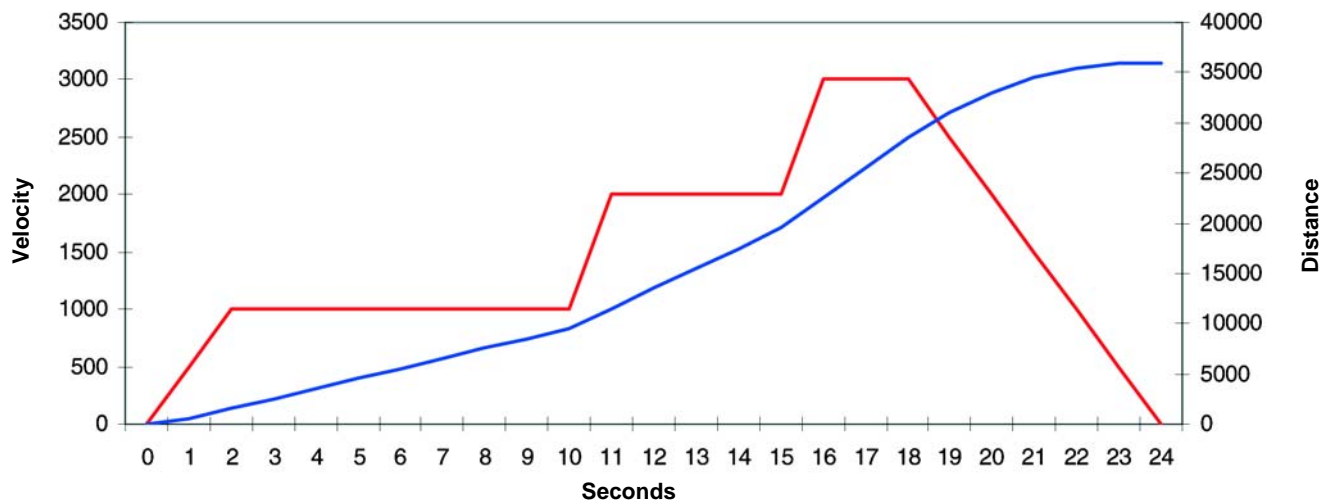


Motion task editor

Step by step: "Linked to previous position" sequencing

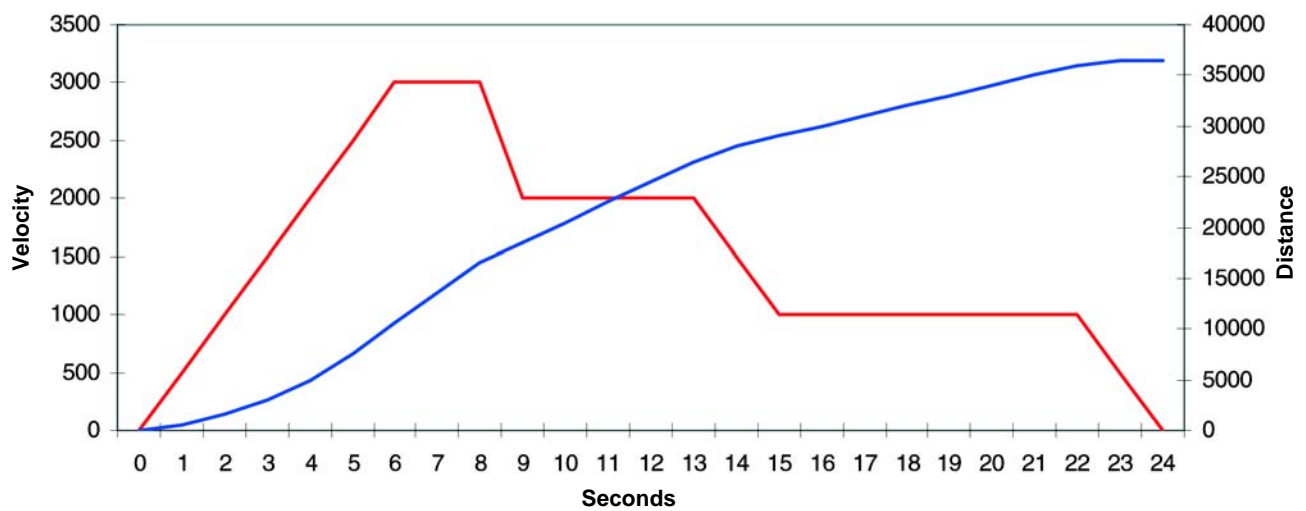
In Linked to previous position mode the movements of the axis follow on one after the other and the velocity of movement N to the target position is equal to velocity V_N of this movement.

| Step | Command | Axis | Param.1 | Val. P1 | Param. 2 | Val. P2 | Param. 3 | Val. P3 | Param. 4 | Val. P4 | Param. 5 | Val. P5 | Param. 6 | Val. P6 | Param. 7 | Val. P7 |
|------|---------------|------|---------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| STEP | CMD | Axis | P1 Name | P1 Value | P2 Name | P2 Value | P3 Name | P3 Value | P4 Name | P4 Value | P5 Name | P5 Value | P6 Name | P6 Value | P7 Name | P7 Value |
| 1 | Power On | 1 | | | | | | | | | | | | | | |
| 2 | eMove ContAbs | 1 | Pos | 9500 | Vel | 1000 | EndVel | 1000 | Acc | 500 | Dec | 500 | Reg | 0 | Buffer | 0 |
| 3 | eMove ContAbs | 1 | Pos | 19500 | Vel | 2000 | EndVel | 2000 | Acc | 1000 | Dec | 1000 | Reg | 0 | Buffer | 1 |
| 4 | eMove Abs | 1 | Pos | 36000 | Vel | 3000 | Acc | 1000 | Dec | 500 | Dir | 1 | Reg | 0 | Buffer | 1 |
| 5 | eWait Axis | 1 | | | | | | | | | | | | | | |



Motion task editor

| Step | Command | Axis | Param.1 | Val. P1 | Param. 2 | Val. P2 | Param. 3 | Val. P3 | Param. 4 | Val. P4 | Param. 5 | Val. P5 | Param. 6 | Val. P6 | Param. 7 | Val. P7 |
|------|---------------|------|---------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| STEP | CMD | Axis | P1 Name | P1 Value | P2 Name | P2 Value | P3 Name | P3 Value | P4 Name | P4 Value | P5 Name | P5 Value | P6 Name | P6 Value | P7 Name | P7 Value |
| 1 | Power On | 1 | | | | | | | | | | | | | | |
| 2 | eMove ContAbs | 1 | Pos | 16500 | Vel | 3000 | EndVel | 3000 | Acc | 500 | Dec | 500 | Reg | 0 | Buffer | 0 |
| 3 | eMove ContAbs | 1 | Pos | 26500 | Vel | 2000 | EndVel | 3000 | Acc | 1000 | Dec | 1000 | Reg | 0 | Buffer | 1 |
| 4 | eMove Abs | 1 | Pos | 36500 | Vel | 1000 | Acc | 1000 | Dec | 500 | Dir | 1 | Reg | 0 | Buffer | 1 |
| 5 | eWait Axis | 1 | | | | | | | | | | | | | | |

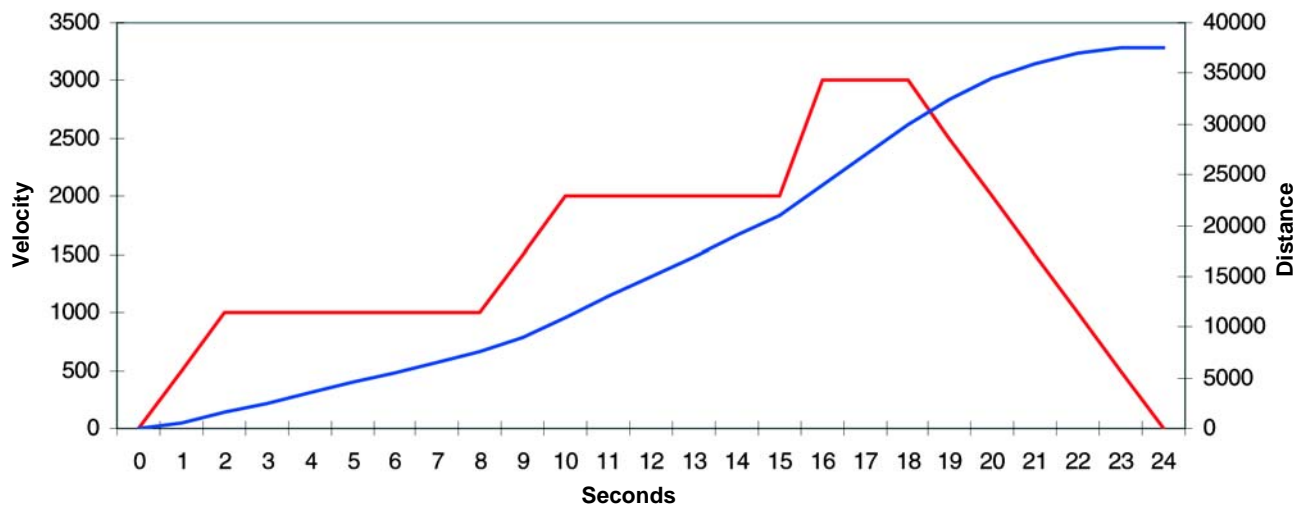


Motion task editor

Step by step: "Linked to next position" sequencing

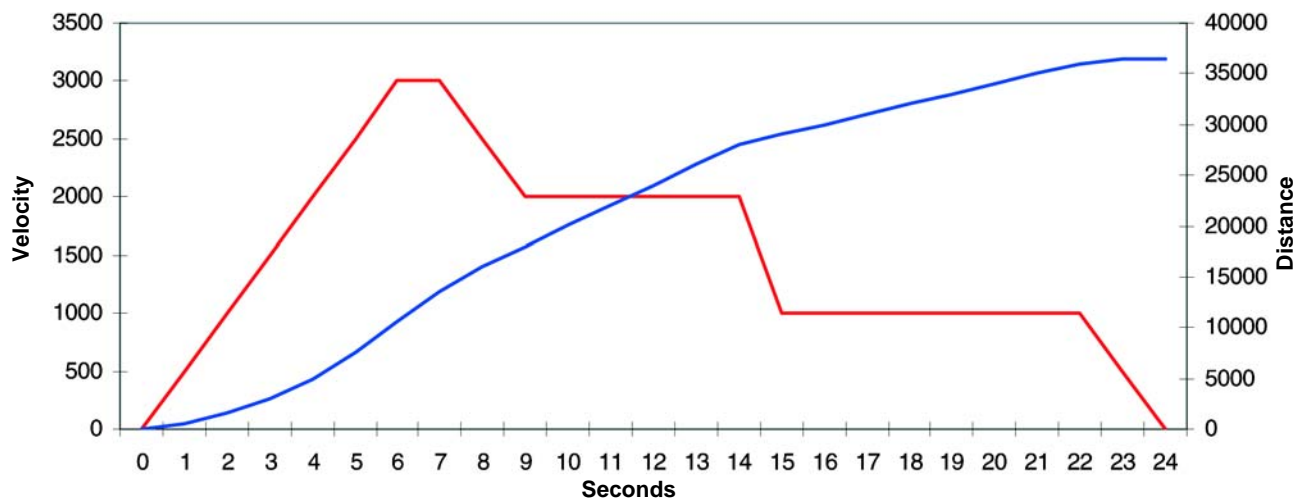
In Linked to next position mode the movements of the axis follow on one after the other and the velocity of movement N to the target position is equal to velocity V_{N+1} of the next movement.

| Step | Command | Axis | Param.1 | Val. P1 | Param. 2 | Val. P2 | Param. 3 | Val. P3 | Param. 4 | Val. P4 | Param. 5 | Val. P5 | Param. 6 | Val. P6 | Param. 7 | Val. P7 |
|------|---------------|------|---------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| STEP | CMD | Axis | P1 Name | P1 Value | P2 Name | P2 Value | P3 Name | P3 Value | P4 Name | P4 Value | P5 Name | P5 Value | P6 Name | P6 Value | P7 Name | P7 Value |
| 1 | Power On | 1 | | | | | | | | | | | | | | |
| 2 | eMove ContAbs | 1 | Pos | 11000 | Vel | 1000 | EndVel | 2000 | Acc | 500 | Dec | 500 | Reg | 0 | Buffer | 0 |
| 3 | eMove ContAbs | 1 | Pos | 21000 | Vel | 2000 | EndVel | 3000 | Acc | 1000 | Dec | 1000 | Reg | 0 | Buffer | 1 |
| 4 | eMove Abs | 1 | Pos | 37500 | Vel | 3000 | Acc | 1000 | Dec | 500 | Dir | 1 | Reg | 0 | Buffer | 1 |
| 5 | eWait Axis | 1 | | | | | | | | | | | | | | |



Motion task editor

| Step | Command | Axis | Param.1 | Val. P1 | Param. 2 | Val. P2 | Param. 3 | Val. P3 | Param. 4 | Val. P4 | Param. 5 | Val. P5 | Param. 6 | Val. P6 | Param. 7 | Val. P7 |
|------|---------------|------|---------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| STEP | CMD | Axis | P1 Name | P1 Value | P2 Name | P2 Value | P3 Name | P3 Value | P4 Name | P4 Value | P5 Name | P5 Value | P6 Name | P6 Value | P7 Name | P7 Value |
| 1 | Power On | 1 | | | | | | | | | | | | | | |
| 2 | eMove ContAbs | 1 | Pos | 18000 | Vel | 3000 | EndVel | 2000 | Acc | 500 | Dec | 500 | Reg | 0 | Buffer | 0 |
| 3 | eMove ContAbs | 1 | Pos | 29000 | Vel | 2000 | EndVel | 1000 | Acc | 1000 | Dec | 1000 | Reg | 0 | Buffer | 1 |
| 4 | eMove Abs | 1 | Pos | 36500 | Vel | 1000 | Acc | 1000 | Dec | 500 | Dir | 1 | Reg | 0 | Buffer | 1 |
| 5 | eWait Axis | 1 | | | | | | | | | | | | | | |



Motion task editor

Examples of tables

Example 1:

When input 1 changes to state 1, axes 1, 2 and 3 start their movements in parallel. When all the axes have reached their target positions, output 1 changes to state 1.

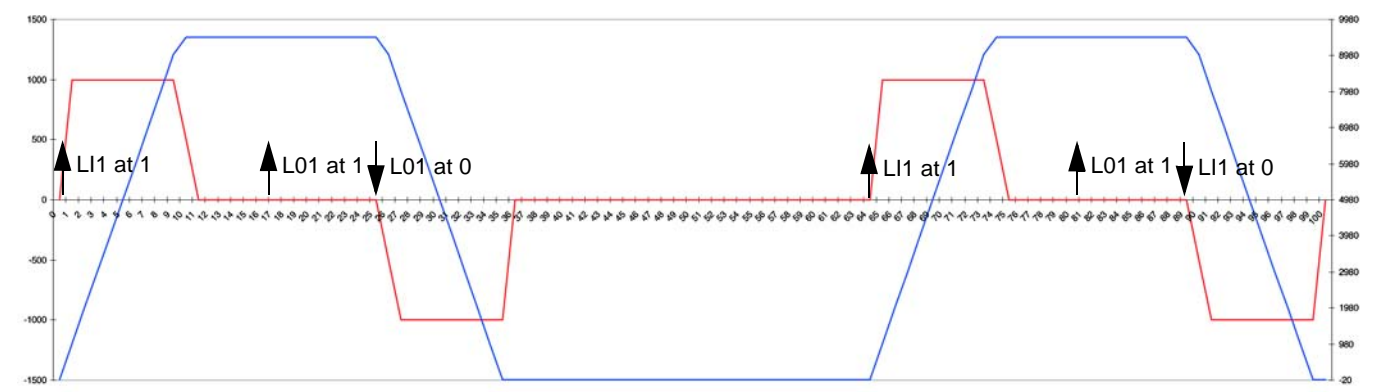
When input 1 changes to state 0, all the axes return to zero, but one after the other.

When axis 3 reaches zero, the output is reset, the motion task table returns to step 2, and the cycle restarts.

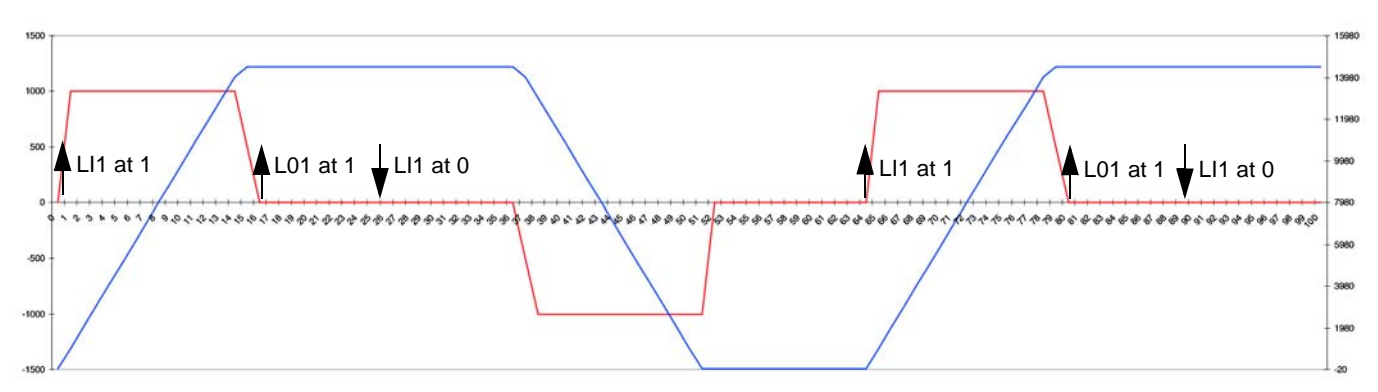
| Command | Axis | Param. 1 | Val. P1 | Param. 2 | Val. P2 | Param. 3 | Val. P3 | Param. 4 | Val. P4 | Param. 5 | Val. P5 | Param. 6 | Val. P6 | Param. 7 | Val. P7 |
|-------------|------|----------|---------|----------|---------|----------|---------|----------|---------|----------|---------|----------|---------|----------|---------|
| CMD | Axis | Par1 | Value1 | Par2 | Value2 | Par3 | Value3 | Par4 | Value4 | Par5 | Value5 | Par6 | Value6 | Par7 | Value7 |
| Power OnAll | 0 | | | | | | | | | | | | | | |
| Waitl_On | | Input | 1 | | | | | | | | | | | | |
| MoveAbs | 1 | Pos | 9500 | Vel | 1000 | Acc | 1000 | Dec | 500 | Dir | 1 | Reg | 0 | Buffered | 0 |
| MoveAbs | 2 | Pos | 14500 | Vel | 1000 | Acc | 1000 | Dec | 500 | Dir | 1 | Reg | 0 | Buffered | 0 |
| MoveAbs | 3 | Pos | -10000 | Vel | 1000 | Acc | 1000 | Dec | 500 | Dir | 1 | Reg | 0 | Buffered | 0 |
| WaitAxis | 1 | | | | | | | | | | | | | | |
| WaitAxis | 2 | | | | | | | | | | | | | | |
| WaitAxis | 3 | | | | | | | | | | | | | | |
| SetQ | | Output | 1 | | | | | | | | | | | | |
| Waitl_Off | | Input | 1 | | | | | | | | | | | | |
| MoveAbs | 1 | Pos | 0 | Vel | 1000 | Acc | 500 | Dec | 1000 | Dir | 1 | Reg | 0 | Buffered | 0 |
| WaitAxis | 1 | | | | | | | | | | | | | | |
| MoveAbs | 2 | Pos | 0 | Vel | 1000 | Acc | 500 | Dec | 1000 | Dir | 1 | Reg | 0 | Buffered | 0 |
| WaitAxis | 2 | | | | | | | | | | | | | | |
| MoveAbs | 3 | Pos | 0 | Vel | 1000 | Acc | 500 | Dec | 1000 | Dir | 1 | Reg | 0 | Buffered | 0 |
| WaitAxis | 3 | | | | | | | | | | | | | | |
| ResetQ | | Output | 1 | | | | | | | | | | | | |
| Jump | | Step | 2 | | | | | | | | | | | | |

Motion task editor

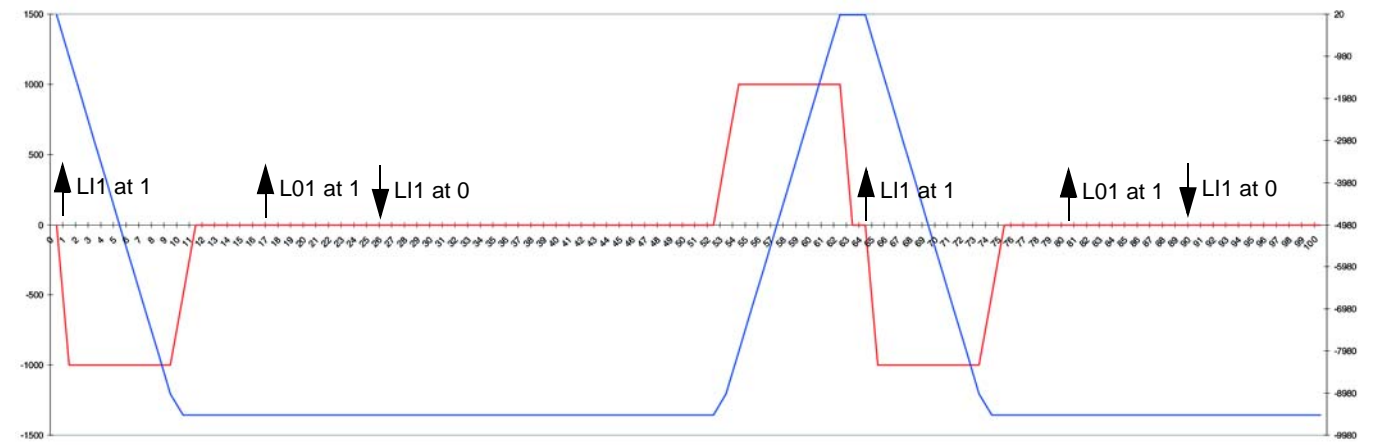
Axis 1:



Axis 2:



Axis 3:



Motion task editor

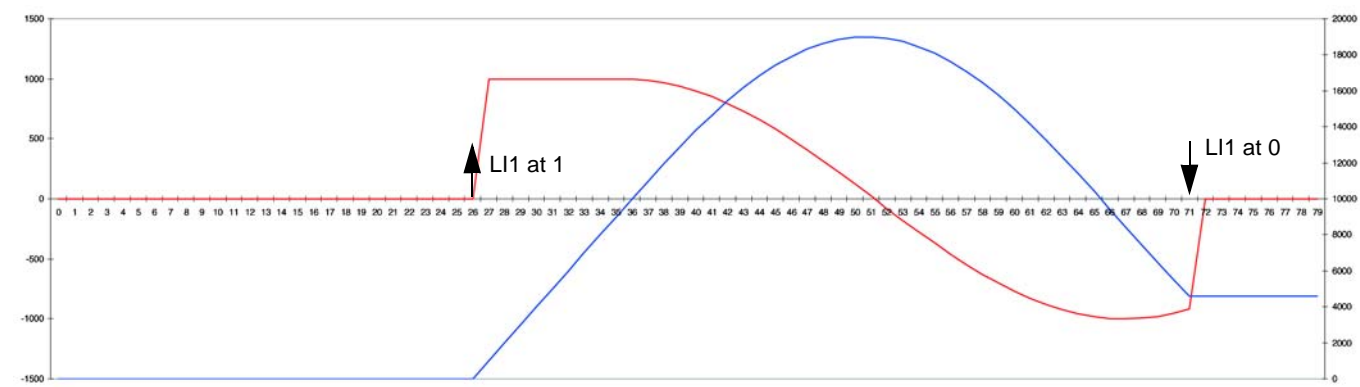
Example 2:

When Input 1 becomes true, User Cam 1 is loaded and a Camming for Axis 2 is started.
As soon as Axis 2 is synchronized, Axis 1 is started with a MoveVel.
The two axes remain synchronized until Input 1 becomes false.
Then Camming is aborted and both Axes are stopped.

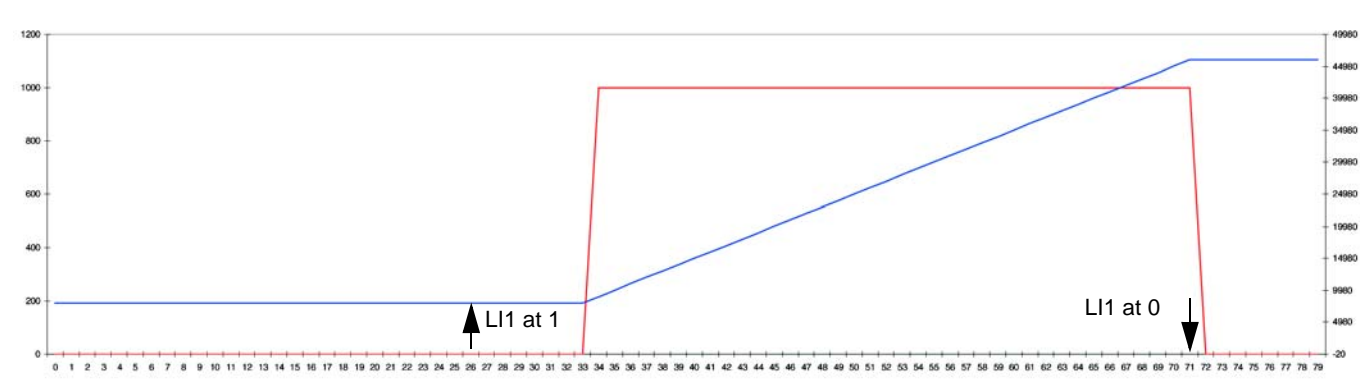
| Step | Command | Axis | Param. 1 | Val. P1 | Param. 2 | Val. P2 | Param. 3 | Val. P3 | Param. 4 | Val. P4 | Param. 5 | Val. P5 | Param. 6 | Val. P6 | Param. 7 | Val. P7 |
|------|-----------|------|----------|---------|----------|---------|-----------|---------|----------|---------|----------|---------|----------|---------|----------|---------|
| Step | CMD | Axis | Par1 | Value1 | Par2 | Value2 | Par3 | Value3 | Par4 | Value4 | Par5 | Value5 | Par6 | Value6 | Par7 | Value7 |
| 1 | PowerOn | 1 | | | | | | | | | | | | | | |
| 2 | PowerOn | 2 | | | | | | | | | | | | | | |
| 3 | WaitI_On | | Input | 1 | | | | | | | | | | | | |
| 4 | CamSel | 2 | Master | 1 | Cam | 1 | Perio Cam | 1 | MstAbs | 1 | | | | | | |
| 5 | CamIn | 2 | Master | 1 | Vel | 1000 | Acc | 1000 | Dec | 1000 | | | | | | |
| 6 | WaitSync | 2 | | | | | | | | | | | | | | |
| 7 | MoveVel | 1 | Vel | 2000 | Acc | 1000 | Dec | 1000 | | | | | | | | |
| 8 | WaitAxis | 1 | | | | | | | | | | | | | | |
| 9 | WaitI_Off | | Input | 1 | | | | | | | | | | | | |
| 10 | Camout | 2 | | | | | | | | | | | | | | |
| 11 | Stop | 2 | Dec | 2000 | | | | | | | | | | | | |
| 12 | WaitAxis | 2 | | | | | | | | | | | | | | |
| 13 | Stop | 1 | Dec | 2000 | | | | | | | | | | | | |
| 14 | WaitAxis | 1 | | | | | | | | | | | | | | |
| 15 | Jump | | Step | 3 | | | | | | | | | | | | |
| 16 | 0 | | | | | | | | | | | | | | | |

Motion task editor

Axis 1:



Axis 2:



Automatic mode

Manual mode enables the motion sequence programmed in the motion sequence editor (see page 15) to be checked step by step.

This mode is the continuous operating mode of the Lexium Controller. It is used to execute the motion sequence continuously.

General organization of the screen

Automatic Operation Mode

DT#1970-01-01-00:00

V1.001

Auto Enable

Start

Step

Reset

StartStep: 1

CycleTime

1

Axis 1

Pos: 73662.305

Vel: 0.0

eNoCMD

power_off

No Error

2

Axis 2

Pos: 24900.389

Vel: 0.0

eNoCMD

power_off

No Error

9

Virt Axis

Pos: 0.000

Vel: 0.0

eNoCMD

power_off

No Error

Editor

Auto

Manual

Config

Error List

Logg List

Trace

| Step | CMD | Axis | P1 Name | P1 Value | P2 Name | P2 Value | P3 Name | P3 Value | P4 Name | P4 Value | P5 Nam |
|------|--------|------|---------|----------|---------|----------|---------|----------|---------|----------|--------|
| 1 | eNoCMD | 0 | | 0 | | 0 | | 0 | | 0 | |
| 2 | eNoCMD | 0 | | 0 | | 0 | | 0 | | 0 | |
| 3 | eNoCMD | 0 | | 0 | | 0 | | 0 | | 0 | |
| 4 | eNoCMD | 0 | | 0 | | 0 | | 0 | | 0 | |
| 5 | eNoCMD | 0 | | 0 | | 0 | | 0 | | 0 | |
| 6 | eNoCMD | 0 | | 0 | | 0 | | 0 | | 0 | |
| 7 | eNoCMD | 0 | | 0 | | 0 | | 0 | | 0 | |
| 8 | eNoCMD | 0 | | 0 | | 0 | | 0 | | 0 | |
| 9 | eNoCMD | 0 | | 0 | | 0 | | 0 | | 0 | |
| 10 | eNoCMD | 0 | | 0 | | 0 | | 0 | | 0 | |

Force IQs

Main Power

Error Reset

Inputs

Outputs

Force Mask

Force Mask

ErrorId: 0





Error Msg:

The automatic mode screen consists of five sections:

- A central screen for supervising all the connected axes
- A control panel on the right of the screen used for supervising the states of the I/O
- The motion task sequence
- The toolbar at the top of the screen
- The navigation bar

Automatic mode

Central supervision screen

| | | | |
|--|--|---|---|
|  Feeder |  Lifter |  Stripper |  Knife |
| Pos: 84.872 | Pos: 83.232 | Pos: 83.232 | Pos: 306.497 |
| Vel: 412.0 | Vel: 408.0 | Vel: 408.0 | Vel: 0.0 |
| eMoveRel | eMoveRel | eMoveRel | eNoCMD |
| discrete_motion | discrete_motion | discrete_motion | standstill |
| No Error | No Error | No Error | No Error |

The central supervision screen is used to display the behavior of the axes during the execution of the sequence. For each axis it gives:

- **Its position:** The position of the axis is updated in real time.
- **Its velocity:** The velocity of the axis is updated in real time.
- **The function block being executed on the axis:** This information is important for checking the synchronization of the sequence.
- **The axis status:** Enables the behavior of the axis to be located in the status diagram (see page 24).
- The presence of errors on the axis.

Control panel

| | Inputs | | | | | | | | | | Outputs | | | | | | | |
|---------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Force Enable |  |  |  |  |  |  |  |  | | |  |  |  |  |  |  |  |  |

Used to display and force the state of the Lexium Controller I/O.

To force I/O click on "**Force IQs**".

"**Force mask**" allows to select I/O to force. Then forcing is done by clicking on the I/O.

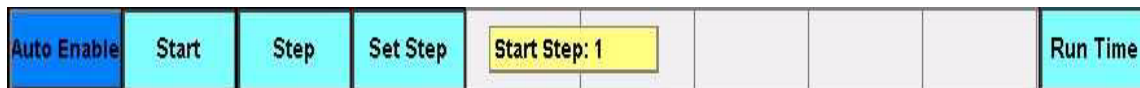
Automatic mode

Motion sequence

| Step | CMD | Axis | P1 Name | P1 Value | P2 Name | P2 Value | P3 Name | P4 Value |
|------|-----------|------|---------|----------|---------|----------|---------|----------|
| 10 | eMoveAbs | 3 | Pos | 0 | Vel | 1000 | Acc | 1000 |
| 11 | eWaitAxis | 1 | | 0 | | 0 | | 0 |
| 12 | eWaitAxis | 2 | | 0 | | 0 | | 0 |
| 13 | eWaitAxis | 3 | | 0 | | 0 | | 0 |
| 14 | eMoveRel | 1 | Pos | 360 | Vel | 1000 | Acc | 1000 |
| 15 | eMoveRel | 2 | Pos | 450 | Vel | 1000 | Acc | 1000 |
| 16 | eMoveRel | 3 | Pos | 540 | Vel | 1000 | Acc | 1000 |
| 17 | eWaitAxis | 1 | | 0 | | 0 | | 0 |
| 18 | eWaitAxis | 2 | | 0 | | 0 | | 0 |
| 19 | eWaitAxis | 3 | | 0 | | 0 | | 0 |

This table enables the progress of the sequence to be monitored step by step in real time. The active step is indicated by a green background.

Toolbar



The toolbar contains five buttons and one field:

- **"Auto Enable"**: This button activates the execution of the motion sequence.
- **"Start"**: This button starts the automatic execution of the motion task sequence. The execution corresponds to the normal operation of the machine.
- **"Step"**: This button starts step by step execution of the motion sequence.
- **"Set Step"**: This button is used to return to the first step in the execution of the sequence.
- **"StartStep"**: This field is used to select to the first step in the execution of the sequence.
- **"Run time"**: This button is used to open the specific window for supervising the Lexium Controller cycle time.

Automatic mode

"Run Time" supervision window

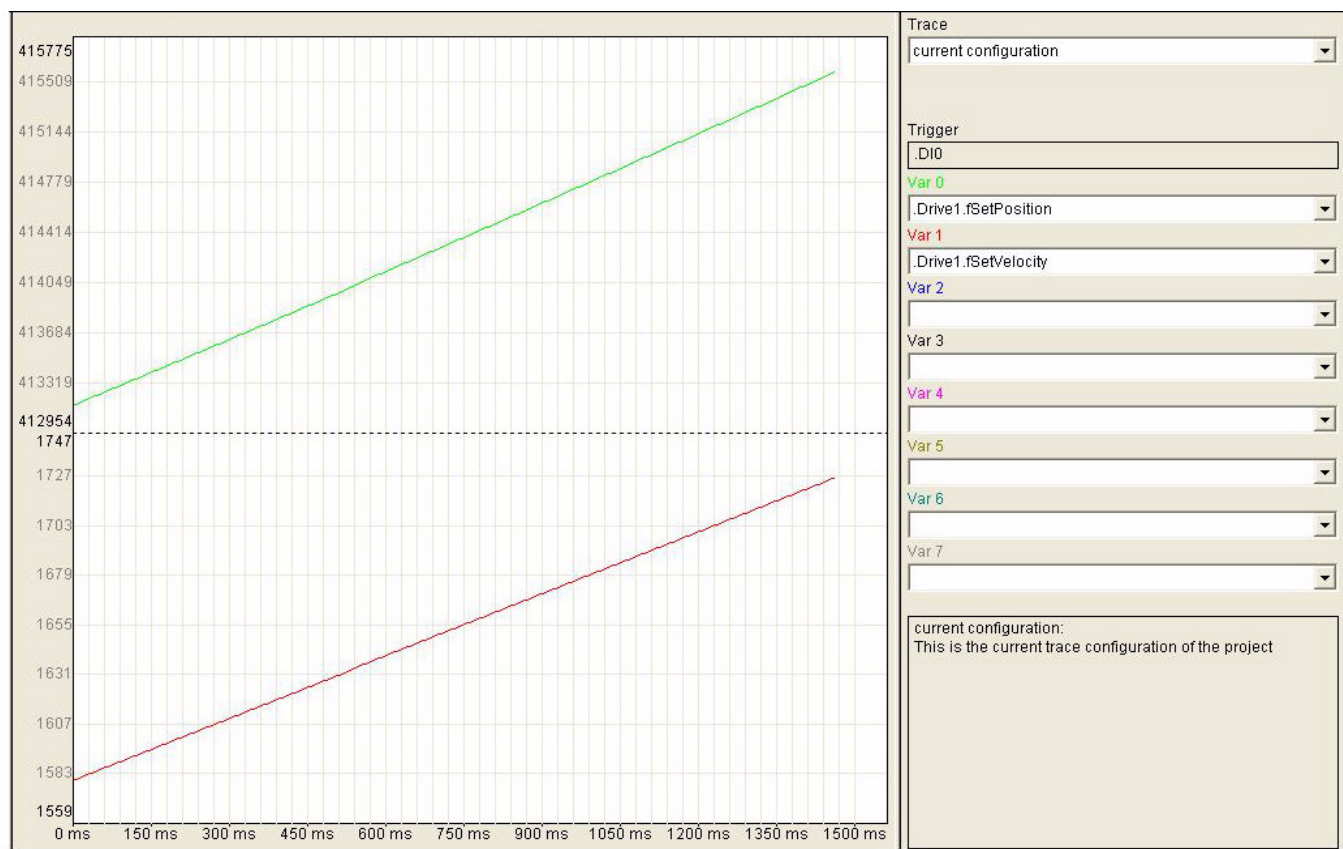
| Motion Task Run Time in micro seconds |
|--|
| Start |
| Run Time: 0 |
| Max Run Time: 0 |

This screen consists of six fields:

- **"Run Time"**: Gives the current cycle time in μs .
- **"Max Run Time"**: Gives the maximum cycle time in μs .

Trace function

This function is accessed by pressing "Trace".



This function is used to read any 8 parameters simultaneously.
These parameters can be selected from a list of active parameters using: "Var 0, Var 1, Var 2", etc.
The parameter-setting actions and options are accessed by right-clicking the mouse:



Please refer to the Motion Pro - Programming manual for more information on the Trace function parameters.

Mains contactor operation

Easy Motion offers the option of controlling the mains contactor that makes the current upstream of the servo drives. The purpose of this function is to control the contactor (gMB_xMainsContactor output) and monitor its status (activated or not) via the feedback contact (NO).

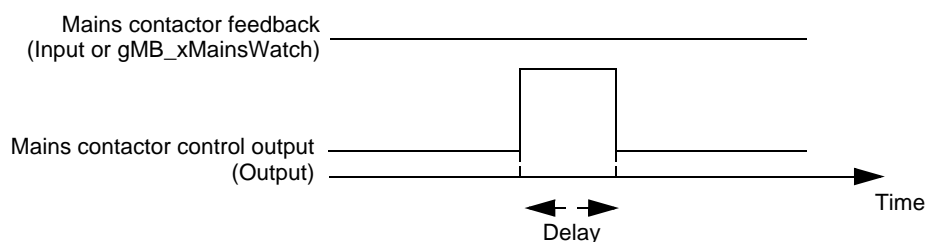
Three types of operation are available. They can be configured via the *MainsWatch Mode* parameter which can be accessed in the **Easy Motion Config/LMC Par** menu.

- If *MainsWatch Mode* = 0, mains contactor feedback monitoring is deactivated.
- If *MainsWatch Mode* = 1, the function is activated, and the *gMB_xMainsWatch* internal variable is monitored. This mode can be used to add software conditions to the feedback contact monitoring.
- If *MainsWatch Mode* = 2, the function is activated, and mains contactor feedback monitoring is associated with a logic input Llx.

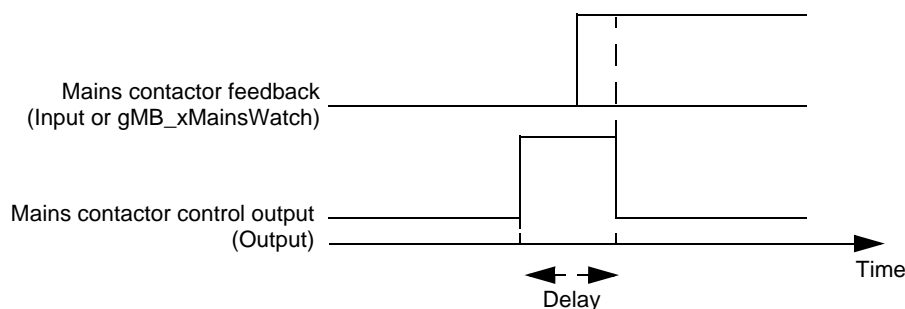
When the function is activated, the start sequence is as follows:

After **Easy Motion** has been started and the drives initialized, the mains contactor control output changes to 1.

If the mains contactor feedback does not change to 1 (Llx logic input or *gMB_xMainsWatch* variable) within a predefined time, or if a Reaction 1 (R1) type error is active, the mains contactor control output changes back to 0, the application cannot start and *Mains Control Error* is displayed.



If the mains contactor feedback switches to 1 before the end of the delay and no Reaction 1 (R1) type error is active, the application can start.



During operation, if the mains contactor feedback changes to 0 (open circuit), if the mains contactor control output is forced to 0 or if a Reaction 1 (R1) type error appears, the system shuts down and *Mains Control Error* is displayed.

Note : The *Main Power* button can also be used to trigger a Reaction 1 (R1) type error. When it is acknowledged by *Error Reset* the axes will be reinitialized.

Default value:

The mains contactor feedback is assigned to the Lexium Controller input 8.

Mains contactor monitoring is assigned to the Lexium Controller output 8.

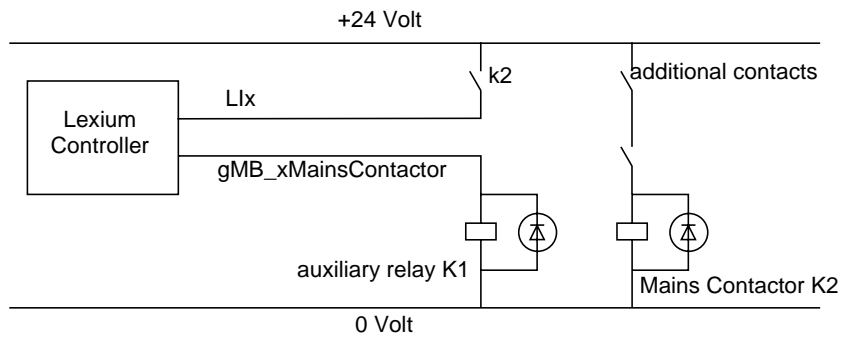
The delay is configured as 1 second.

Control of the mains contactor

These parameters can be configured when the **Application Template** is used with **Motion Pro**.

Please refer to the *Easy Motion - Programming Manual - Application Template parameter table* documentation for more information.

Example of wiring the mains contactor with a mains contactor auxiliary contact wired on a logic input (Llx)



Scaling

For the user to be able to program his distances, positions (degrees, μm , etc) as well as the velocities, accelerations and decelerations using units that are consistent with those used in mechanical engineering, a scaling coefficient has to be defined for each axis.

This coefficient takes the form of a ratio of 2 parameters (integers):

- User Unit Numerator (N)
- User Unit Denominator (D)

There are two possible ways to determine the ratio of N/D

a) Whether it is possible to measure or find out precisely the distance traveled by the axis for one motor revolution.

$$\frac{N}{D} = X \times \frac{1}{\text{INC}}$$

X is the distance traveled for one motor revolution. This distance must be expressed in the unit that the user wishes to use.

b) If various devices such as a gearbox, ball screw, and cog wheels make up the drive chain, I can determine my N/D ratio using the ratios of these devices.

$$\frac{N}{D} = Y \times \frac{\text{OUT}}{\text{IN}} \times \frac{1}{\text{INC}}$$

Y is the distance traveled for one revolution of the ball screw or cog wheel. This distance must be expressed in the unit that the user wishes to use.

OUT: Is the number of revolutions at the gearbox output for IN input revolutions.

INC: Is the number of increments per motor revolution.

In the case of the LEX05 with BSH motor **INC** = 131,072

With a BSH, **INC** = 131,072.

With other types of motor, the maximum value of **INC** can vary (Max = 1,054,478)

The speed should always be expressed in Units/sec

The acceleration and deceleration should always be expressed in units/sec²

Important note:

For users of Motion Pro/CoDeSys, who also use the Application Template, it is preferable to use the Application Template configuration screens or the graphic display terminal to set these parameters.

In this case, the Motion Pro/CoDeSys configuration screens must not be used.

Error management

Active Error List

DT#1970-01-01-00:00

V1.001

<

When `gMB_xErrorHandlerEnable = 1`, faults are routinely saved in 2 lists:

- **Active Error list:** accessible in the **Easy Motion Error List** menu.
This list details the active errors and is deleted after an *Error Reset*.
It is limited to 16 inputs. If more than 16 errors are active, only the first 16 errors are listed.
- **Logger List:** accessible in the **Easy Motion Logg List** menu.
This list contains the error history. It lists all errors, even after they have been acknowledged.
It is deleted after a *Del Logger*.
It is limited to 32 inputs and operates on the FIFO principle (First In, First Out).

If the error is still active, after an *Error Reset* it is automatically rewritten to *Active Error List* and it appears twice in *Logger List*.

On the graphic display terminal, only the most recent error is displayed.

There are 2 types of error:

- Errors specific to **Codesys - Soft Motion**. For more information about these errors, please refer to the *Codesys - Soft Motion* documentation.
- Errors specific to the **Application Template** with a type 40xx ID.

Error management

Description of the errors specific to the **Application Template**:

There are 3 classes of fault, each causing a different reaction:

Reaction 1 (R1): Faults causing stopping and disabling (PowerOff) of all axes and resetting of the mains contactor.

Reaction 2 (R2): Faults causing stopping and disabling (PowerOff) of all axes.

Reaction 3 (R3): Faults causing stopping without disabling of all axes.

| ID | Reaction | Error message | Meaning | Corrective action |
|------|----------|---------------------------|---|---|
| 4001 | 1 | Emergency Stop active | The emergency stop bit has been activated | Change the <i>gMB_xEmergencyStop</i> bit to FALSE This bit can only be controlled via the <i>Application Template parameter table</i> . Please refer to the <i>Easy Motion - Programming Manual - Application Template parameter table</i> documentation for more information. |
| 4002 | 1 | Mains Control Error | Error affecting the mains contactor function or a Reaction 1 (R1) type error is active | Check the mains contactor mechanism If no physical mains contactor mechanism has been fitted, set the <i>MainsWatch Mode</i> parameter to 0. This parameter can be accessed in the Easy Motion Config/LMC Par menu Please refer to the <i>Easy Motion - Programming Manual - Mains Contactor operation</i> documentation for more information about how the mains contactor works |
| 4003 | 3 | Drive Number out of range | The axis number is invalid | The axis number must be between 1 and 10 Please refer to the <i>Easy Motion - Programming Manual - Application Template parameter table</i> documentation for more information |
| 4004 | 3 | MTI Parameter Error | The parameters entered in the motion sequence are not correct | Please refer to the <i>Easy Motion - Programming Manual - Application Template parameter table</i> documentation for more information about the parameter value ranges |
| 4005 | 3 | MTI Unknown Command | Command not recognized in the motion sequence | Please refer to the <i>Easy Motion - Programming Manual - Application Template parameter table</i> documentation for more information about the motion sequence commands |
| 4006 | 3 | MTI Timeout Error | A timeout can be configured in each motion task. This error is triggered when this timeout has elapsed. | Modify the programming in the motion sequence |
| 4007 | 3 | MTI Repeat Error | The number of repeats (eRepeat) is more than 8 | Modify the programming in the motion sequence |
| 4008 | 3 | Teach List Index Error | The teach function table index is invalid | The index must be an integer between 1 and 32 |

Error management

| ID | Reaction | Error message | Meaning | Corrective action |
|------|----------|---------------------------------|---|--|
| 4009 | 3 | Invalid User Cam | The number of the selected Cam profile, the number of points or the value of certain parameters are invalid | Please refer to the <i>Easy Motion - Programming Manual - Application Template parameter table</i> documentation for more information about the Cam profile value ranges |
| 4010 | 3 | Axis Parameter Error | The axis parameters are inconsistent with those saved during the last session | This error always appears the first time Easy Motion is used. Use the <i>Read Par</i> command in the Easy Motion Config menu to update the axis parameters |
| 4011 | 1 | Error during startup | Problem during servo drive initialization | Check communication with the servo drives (numbers of the nodes and baudrate) This error appears if the number of axes configured is higher than the number of axes that are actually connected In this case, correct the number of axis and reset the Lexium Controller |
| 4015 | 3 | Axis Name: Unknown CMD | An invalid command has been sent to the axis. | Please refer to the <i>Easy Motion - Programming Manual - Application Template parameter table</i> documentation for more information about valid axis commands |
| 4016 | 2 | Axis Name: Axis in Error Stop | A motion command has been sent to an axis in ErrorStop mode (PLCopen state) | Please refer to the PLCopen state chart in the <i>Easy Motion - Programming Manual</i> documentation for more information |
| 4017 | 3 | Axis Name: Axis in wrong State | A motion command has been sent to an axis in PowerOff mode (PLCopen state) | Please refer to the PLCopen state chart in the <i>Easy Motion - Programming Manual</i> documentation for more information |
| 4018 | 3 | Axis Name: CMD not allowed | An eMoveSupl command has been sent to a stopped axis | The eMoveSupl command must be sent to a moving axis Please refer to the PLCopen state chart in the <i>Easy Motion - Programming Manual</i> documentation for more information |
| 4019 | 3 | Axis Name: Invalid Axis Number | The axis number is invalid | The axis number must be between 1 and 10 Please refer to the <i>Easy Motion - Programming Manual - Application Template parameter table</i> documentation for more information |
| 4020 | 3 | Axis Name: Buffer size exceeded | The number of movements stored in memory is more than 8 | Modify the programming in the motion sequence |
| 4026 | 3 | Read Drive Parameter Error | Error when reading the parameters on a servo drive | Check communication with the servo drives |
| 4027 | 3 | Write Drive Parameter Error | Error when writing the parameters to a servo drive | Check communication with the servo drives |
| 4028 | 3 | Save Drive Parameter Error | Error when saving the parameters to non-volatile memory on a servo drive | Check communication with the servo drives |
| 4029 | 3 | Upload Drive Parameter Error | Error when uploading a configuration from a servo drive | Check communication with the servo drives |
| 4030 | 3 | Download Drive Parameter Error | Error when downloading a configuration to a servo drive | Check communication with the servo drives |

Application Template parameter table

Parameters and commands accessible via the Application Template can be managed by global variables. Those variables are accessible via Ethernet, Modbus or directly with Motion Pro. It allows to use the functionalities of the application template without using the Graphical User Interface.

| Parameter Name | Address | Type | Default Value | Range | Description | Access |
|----------------------------|---------|---------------|---------------|----------------------------------|--|--------|
| SYSTEM | | | | | | |
| gMB_sUserProjectName | %MW0 | STRING[19] | - | - | Project name defined by User in configuration screen | R / W |
| gMB_xDisplayEncoder | %MW20 | BOOL | 0 | 0 / 1 | Display Encoder in manual and automatic modes | R / W |
| gMB_xDisplayVirtAxis | %MW21 | BOOL | 0 | 0 / 1 | Display virtual Axis in manual and automatic modes | R / W |
| gMB_xForceEnable | %MW17 | BOOL | | 0 / 1 | Enables forcing of I/O | R / W |
| gMB_dwMC_Input_Image | %MW22 | WORD | - | 8# 00000000 TO 8# 11111111 | Image of local Logical inputs (LI1 to LI8) | R |
| gMB_dwMC_Input_ForceValue | %MW23 | WORD | 0 | 8# 00000000 TO 8# 11111111 | Force values of HW inputs | R |
| gMB_dwMC_Input_ForceMask | %MW24 | WORD | 16# FFFF | 16# 0000 TO 16# FFFF | Force mask of local logical inputs 0 : not allowed to force 1 : allowed to force | R / W |
| gMB_dwMC_Output_Image | %MW25 | WORD | | 8# 00000000 TO 8# 11111111 | Image of Logical outputs (LO1 to LO8) | R |
| gMB_dwMC_Output_ForceValue | %MW26 | WORD | 0 | 8# 00000000 TO 8# 11111111 | Force values of local logical outputs | R |
| gMB_dwMC_Output_ForceMask | %MW27 | WORD | 16# FFFF | 16# 0000 TO 16# FFFF | Force mask of local logical outputs =0 not allowed to force 1 : allowed to force | R / W |
| gMB_xBatteryOK | %MW28 | BOOL | | 0 / 1 | Battery status 1 : battery ok 0 : battery in default | R |
| gMB_iTPLVersionLow | %MW29 | WORD | 0 | | Application Template Version, low index | R |
| gMB_iTPLVersionHigh | %MW30 | WORD | 1 | | Application Template Version, high index | R |
| gMB_sTPLVersion | %MW31 | STRING[11] | V1.0.0 | | Complete Application Template Version, string format | R |
| gMB_dtMC_SystemTime | %MW42 | DATE_AND_TIME | | | System Timer Format : DATE_AND_TIME#1996-05-06-15:36:30 | R / W |
| gB_dtC_SetSystemTime | %MW44 | DATE_AND_TIME | | | Set value of system watch | R / W |
| gMB_timSystemTime | %MW46 | TIME | | | Format : t#12h34m15s | R / W |

Application Template parameter table

| Parameter Name | Address | Type | Default Value | Range | Description | Access |
|-----------------------------|---------|------|---------------|-------|--|--------|
| ERROR HANDLING | | | | | | |
| gMB_xEmergencyStop | %MW50 | BOOL | 0 | 0 / 1 | Emergency Stop, 1 : generates error category 1 It is not a safety feature | R / W |
| gMB_xErrorReset | %MW51 | BOOL | 0 | 0 / 1 | Global Error reset (reset whole active error list) | W |
| gMB_xErrorReset1 | %MW52 | BOOL | 0 | 0 / 1 | Error reset for Axis1 (axis module reset) (drive + program) | W |
| gMB_xErrorReset2 | %MW53 | BOOL | 0 | 0 / 1 | Error reset for Axis2 | W |
| gMB_xErrorReset3 | %MW54 | BOOL | 0 | 0 / 1 | Error reset for Axis3 | W |
| gMB_xErrorReset4 | %MW55 | BOOL | 0 | 0 / 1 | Error reset for Axis4 | W |
| gMB_xErrorReset5 | %MW56 | BOOL | 0 | 0 / 1 | Error reset for Axis5 | W |
| gMB_xErrorReset6 | %MW57 | BOOL | 0 | 0 / 1 | Error reset for Axis6 | W |
| gMB_xErrorReset7 | %MW58 | BOOL | 0 | 0 / 1 | Error reset for Axis7 | W |
| gMB_xErrorReset8 | %MW59 | BOOL | 0 | 0 / 1 | Error reset for Axis8 | W |
| gMB_xErrorReset9 | %MW60 | BOOL | 0 | 0 / 1 | Error reset for virtual Axis | W |
| gMB_xErrorHandlerEnable | %MW61 | BOOL | 1 | 0 / 1 | 1 : Enable Error Handler - active errors list - error logger list - error reactions | R / W |
| gMB_xError | %MW62 | BOOL | 0 | 0 / 1 | Status global Error active (any kind of error) | R |
| gMB_xErrorReaction1 | %MW63 | BOOL | 0 | 0 / 1 | Status Error reaction1 active | R |
| gMB_xErrorReaction2 | %MW64 | BOOL | 0 | 0 / 1 | Status Error reaction10 active | R |
| gMB_xErrorReaction3 | %MW65 | BOOL | 0 | 0 / 1 | Status Error reaction100 active | R |
| gMB_xBlinkSlow | %MW66 | BOOL | 0 | 0 / 1 | Slow blink frequency (1 sec on / 0.5 off) | R |
| gMB_xBlinkFast | %MW67 | BOOL | 0 | 0 / 1 | Fast blink frequency (0.5 sec on / 0.25 off) | R |
| gMB_iNumberOfActiveErrors | %MW4443 | INT | | | Active errors in the list | R |
| gMB_xLoggerDelete | %MW4828 | BOOL | | 0 / 1 | Delete logger list | W |
| gMB_iNumberOfLogListEntries | %MW4829 | INT | | | Number of entries in the logger list | R |

Application Template parameter table

| Parameter Name | Address | Type | Default Value | Range | Description | Access |
|----------------------------|---------|--------------|---------------|---------|--|--------|
| MANUAL MODE | | | | | | |
| gMB_xTeachEnter | %MW98 | BOOL | 0 | 0 / 1 | Enter selected axis actual position to teach list register | R / W |
| gMB_xTeachDeleteList | %MW99 | BOOL | 0 | 0 / 1 | Delete COMPLETE teachlist | R / W |
| gMB_xTeachInsertLine | %MW96 | BOOL | | 0 / 1 | Insert line, moves all following lines one line up | W |
| gMB_xTeachDeleteLine | %MW97 | BOOL | | 0 / 1 | Delete actual line | W |
| gMB_uiTeachListIndex | %MW4185 | UINT | 1 | 1 to 32 | Index of teachlist | R / W |
| gMB_Manual_iAxisNo | %MW100 | UINT | 0 | 1 to 10 | Selected Axis number for manual operation - 1 to 8 for real axis - 9 for virtual axis | R / W |
| gMB_Manual_xJoggForw | %MW101 | BOOL | 0 | 0 / 1 | Input jogging forward (in manual mode) | R / W |
| gMB_Manual_xJoggBack | %MW102 | BOOL | 0 | 0 / 1 | Input jogging backward (in manual mode) | R / W |
| gMB_Manual_xMoveRelStart | %MW103 | BOOL | 0 | 0 / 1 | Input start of relative movement (in manual mode) | R / W |
| gMB_Manual_xMoveVelStart | %MW104 | BOOL | 0 | 0 / 1 | Input start of velocity movement (in manual mode) | R / W |
| gMB_Manual_xMoveAbsStart | %MW105 | BOOL | 0 | 0 / 1 | Input start of absolute movement (in manual mode) | R / W |
| gMB_Manual_xHomeStart | %MW106 | BOOL | 0 | 0 / 1 | Input start of homing (in manual mode) | R / W |
| gMB_Manual_xStop | %MW107 | BOOL | 0 | 0 / 1 | Input stop (in manual mode) | R / W |
| gMB_Manual_xPowerOn | %MW108 | BOOL | 0 | 0 / 1 | Input Power On (in manual mode) This input closes position and velocity loop on the motor. | R / W |
| gMB_Manual_xSetPos | %MW109 | BOOL | 0 | 0 / 1 | Input Set axis Position. The axis position is settled to gMB_Manual_rPosition (in manual mode) | R / W |
| gMB_Manual_rVel | %MW110 | REAL | 100 | | Velocity for manual movement | R / W |
| gMB_Manual_rAcc | %MW112 | REAL | 1000 | | Acceleration for manual movement | R / W |
| gMB_Manual_rDec | %MW114 | REAL | 1000 | | Deceleration for manual movement | R / W |
| gMB_Manual_rPosition | %MW116 | REAL | 100 | | Position for manual movement | R / W |
| gMB_Manual_eDirection | %MW118 | MC_Direction | 0 | -1 to 3 | Direction for manual movement - 1 : negative 0 : shortest 1 : positive 2 : previous 3 : fastest | R / W |
| gMB_Manual_xJoggForwActive | %MW122 | BOOL | 0 | 0 / 1 | Status Jog forward active (in manual mode) | W |
| gMB_Manual_xJoggBackActive | %MW123 | BOOL | 0 | 0 / 1 | Status Jog backward active (in manual mode) | W |
| gMB_Manual_xMoveRelActive | %MW124 | BOOL | 0 | 0 / 1 | Status relative move active (in manual mode) | W |
| gMB_Manual_xMoveAbsActive | %MW125 | BOOL | 0 | 0 / 1 | Status absolute move active (in manual mode) | W |
| gMB_Manual_xHomeActive | %MW126 | BOOL | 0 | 0 / 1 | Status homing move active (in manual mode) | W |
| gMB_Manual_xMoveVelActive | %MW127 | BOOL | 0 | 0 / 1 | Status velocity move active (in manual mode) | W |
| gMB_Manual_xSetPosActive | %MW128 | BOOL | 0 | 0 / 1 | Status set axis position active (in manual mode) | W |

Application Template parameter table

| Parameter Name | Address | Type | Default Value | Range | Description | Access |
|---------------------------|---------|------|---------------|---------|---|--------|
| CONFIGURATION MODE | | | | | | |
| gMB_iConfigAxisNumber | %MW130 | INT | 0 | 1 to 10 | Selected Axis Number for Configuration mode - 1 to 8 for real axis - 0 for virtual axis | R / W |
| gMB_xDownLoadPar | %MW131 | BOOL | 0 | 0 / 1 | Input Download drive configuration Controller to drive | W |
| gMB_xDownLoadActive | %MW132 | BOOL | 0 | 0 / 1 | Status Download Controller to drive (feedback of gMB_xDownLoadPar) | R |
| gMB_xUpLoadPar | %MW133 | BOOL | 0 | 0 / 1 | Input Upload drive configuration Drive to Controller | R / W |
| gMB_xUpLoadActive | %MW134 | BOOL | 0 | 0 / 1 | Status Upload Drive to Controller Status (feedback of gMB_xUpLoadPar _) | R |
| gMB_xUpDownLoadYes | %MW135 | BOOL | 0 | 0 / 1 | Input acknowledge Up-DownLoad (wait for gMB_xDownLoadActive) | R/W |
| gMB_xUpDownLoadBusy | %MW136 | BOOL | 0 | 0 / 1 | Status Upload/download drive configuration in progress | R |
| gMB_xUpDownLoadError | %MW137 | BOOL | 0 | 0 / 1 | Status Upload/download drive configuration error (reseted with global error reset) | R |
| gMB_xEditParameter | %MW138 | BOOL | 0 | 0 / 1 | Input Edit parameters defined by : gMB_xAxisParEditSel and gMB_iConfigAxisNumber parameters gMB_xEncParEditSel gMB_xLMCParEditSel | R/W |
| gMB_xEditParameterActive | %MW139 | BOOL | 0 | 0 / 1 | Status edit axis parameters active feedback. | R |
| gMB_xConfigWritePar | %MW140 | BOOL | 0 | 0 / 1 | Input write axes Parameter (all parameters selected (encoder, axis, controller) | R / W |
| gMB_xConfigReadPar | %MW141 | BOOL | 0 | 0 / 1 | Input start read axes Parameter | R / W |
| gMB_xConfigSavePar | %MW142 | BOOL | 0 | 0 / 1 | Input start save drive Parameter (in non Volatil memory) | R / W |
| gMB_xWriteParActive | %MW143 | BOOL | 0 | 0 / 1 | Status write axes Parameters active | R |
| gMB_xReadParActive | %MW144 | BOOL | 0 | 0 / 1 | Status read axes Parameters active | R |
| gMB_xSaveParActive | %MW145 | BOOL | 0 | 0 / 1 | Status save drive Parameters active | R |
| gMB_xAxisParEditSel | %MW146 | BOOL | 0 | 0 / 1 | Input select Edit Axis Parameters. Axis number is given by gMB_iConfigAxisNumber | R / W |
| gMB_xEncParEditSel | %MW147 | BOOL | 0 | 0 / 1 | Input select Edit Master Encoder Parameters | R / W |
| gMB_xLMCParEditSel | %MW148 | BOOL | 0 | 0 / 1 | Input select Edit Lexium Controller Parameters | R / W |
| gMB_xAxisParEditActive | %MW149 | BOOL | 0 | 0 / 1 | Status Edit Axis Parameters active | R |
| gMB_xEncParEditActive | %MW150 | BOOL | 0 | 0 / 1 | Status Edit Master Enc Parameters active | R |
| gMB_xLMCParEditActive | %MW151 | BOOL | 0 | 0 / 1 | Status Edit Lexium Controller Parameters active | R |

Application Template parameter table

| Parameter Name | Address | Type | Default Value | Range | Description | Access |
|----------------------------|---------|-----------------|---------------|-----------|---|--------|
| MAINS CONTACTOR | | | | | | |
| gMB_iMainsWatchMode | %MW19 | WORD | 0 | 0 / 1 / 2 | Selection of Mains Watch feedback mode 0 : feed back is Simulated, 1 : the internal gMB_xMainsWatch variable is used to simulate the feedback 2 : a logical input (Lix) is used. The number of the input is defined by gMB_iMainsWatchInPut | R / W |
| gMB_xMainsWatch | %MW155 | BOOL | 0 | 0 / 1 | Only needed If gMB_xMainsWatchMode=1 Simulate the Feedback contact from Mains Contactor It has to be false and switch to true within gMB_timPowerOnDelay | R / W |
| gMB_timPowerOnDelay | %MW156 | TIME | 1s | | Power on delay | R / W |
| gMB_xMainsContactor | %MW161 | BOOL | 0 | 0 / 1 | Status Mains Contactor | R |
| gMB_byMainsContactorOutPut | %MW162 | BYTE | 8 | 0 to 8 | Number of Lexium Controller HW output used for Mains Contactor 0 for no output used 1 to 8 number of output used | R / W |
| gMB_byMainsWatchInPut | %MW163 | BYTE | 8 | 0 to 8 | Only needed If gMB_xMainsWatchMode=2 It has to be false and switch to true within gMB_timPowerOnDelay Number of Lexium Controller HW input used for Mains Contactor feedback | R / W |
| OPERATIONAL MODES | | | | | | |
| gMB_xManualOn | %MW180 | BOOL | 0 | 0 / 1 | Operation mode Manual select | R / W |
| gMB_xAutoOn | %MW181 | BOOL | 0 | 0 / 1 | Operation mode Automatic select | R / W |
| gMB_xCommissOn | %MW182 | BOOL | 0 | 0 / 1 | Operation mode Configuration select | R / W |
| gMB_xManActive | %MW183 | BOOL | 0 | 0 / 1 | Operation mode Manual active | R |
| gMB_xAutoActive | %MW184 | BOOL | 0 | 0 / 1 | Operation mode Automatic active | R |
| gMB_xConfigActive | %MW185 | BOOL | 0 | 0 / 1 | Operation mode Configuration active | R |
| AUTOMATIC MODE | | | | | | |
| gMB_xMTIStart | %MW190 | BOOL | 0 | | Start Motion Task table execution at step defined by gMB_uiMTIStartStep | R / W |
| gMB_xMTIReset | %MW191 | BOOL | 0 | | Reset Motion Task table to step defined by gMB_uiMTIStartStep | R / W |
| gMB_xMTISingleStep | %MW192 | BOOL | 0 | | Single Step Motion Task table execution | R / W |
| gMB_uiMTIStartStep | %MW193 | UINT | 1 | 1 to 64 | Defines the Start Step of Motion Task table | R / W |
| gMB_iMTIActualStep | %MW196 | INT | 0 | 1 to 64 | Actual step of Motion Task table | R |
| gMB_eMTIActualCMD | %MW197 | TPL_CommandType | - | - | Actual command of Motion Task table. See list of commands | R |
| gMB_iNumberOfAxes | %MW299 | INT | 2 | 1 to 8 | Number of real Axes in Application Template | R / W |

Application Template parameter table

Axis module interface

This interface allows, through an array of words, to control the status and to send move commands for each axis independently. It contains all the parameters needed for every type of commands (**eCMD**, see list below)
Parameters are taken into account and the command is executed as soon as **eCMD** is different than 0.

Command names and numbers (**eCMD**):

| Command Nbr | Command Name | Command Nbr | Command Name | Command Nbr | Command Name |
|-------------|---------------|-------------|--------------|-------------|--------------|
| 1 | eMoveAbs, | 12 | eHome, | 23 | eRKnifeStop, |
| 2 | eMoveRel, | 13 | eCamSel, | 24 | eMoveSupl, |
| 3 | eMoveAdd, | 14 | eCamIn, | 25 | ePhasing, |
| 4 | eMoveVel, | 15 | eCamOut, | 26 | eSetPos, |
| 5 | eMoveContRel, | 16 | eGearIn, | 27 | eWritePar, |
| 6 | eMoveContAbs, | 17 | eGearOut, | 28 | eClampIn, |
| 7 | eSTOP, | 18 | eFShearIn, | 29 | eClampOut, |
| 8 | ePowerOn, | 19 | eFShearOut, | 30 | eStrippIn, |
| 9 | ePowerOnAll, | 20 | eFShearStop, | 31 | eStrippOut, |
| 10 | ePowerOff, | 21 | eRKnifeIn, | 32 | eAccuIn, |
| 11 | ePowerOffAll, | 22 | eRKnifeOut, | 33 | eAccuOut, |

For further information regarding the parameters and the behaviour of the functions see page [23](#).

The interface is shown as an array of parameters. Each configured axis has its own array.

| Variable Name | Address | Data Type |
|----------------------------|---------|--|
| gMB_astAxisModuleInterface | %MW300 | ARRAY [1..gc_iMaxNumerOfAxes] OF TPL_AxisModuleInterfaceType |

Length of the array : 106 words / 212 bytes

| Axis | Axis default name | Array address |
|--------------|-------------------|---------------|
| Real axis 1 | Axis1 | %MW300 |
| Real axis 2 | Axis2 | %MW406 |
| Real axis 3 | Axis3 | %MW512 |
| Real axis 4 | Axis4 | %MW618 |
| Real axis 5 | Axis5 | %MW724 |
| Real axis 6 | Axis6 | %MW830 |
| Real axis 7 | Axis7 | %MW936 |
| Real axis 8 | Axis8 | %MW1042 |
| Virtual axis | VirtAxis | %MW1148 |

The following array gives:

All the parameters of the axis module interface

The offset to add to the address of the first byte

For instance, address of **byNumberOfOutput2** for axis 6 is:

%MW(830+4+2) = **%MB836**

eDirection for axis 4 is:

%MW(618+14) = **%MW632**

Application Template parameter table

| Offset (byte) | Var Type | LL Byte + 0 | LH Byte +1 | HL Byte +2 | HH Byte +3 |
|---------------|----------|---|--|--|---|
| 0 | Input | iMasterAxisNumber: Master axis number Is used only if eCMD function requires a master (ex: eGearIn, eCamIn...). | | eCMD <ul style="list-style-type: none">Type of function to executeCan be either the name or the number of the command (see list before) | |
| 4 | Input | xBuffer eCMD is memorised and will be executed after the previous command. Used for sequencing of movement | byNumberOfOutput1 Lexium Controller's output number allocated to the first output of eCMD function | byNumberOfOutput2 Lexium Controller's output number allocated to the second output of eCMD function | byNumberOfOutput3 Lexium Controller's output number allocated to the third output of eCMD function |
| 8 | Input | rPosition position parameter of eCMD function | | | |
| 12 | Input | rVelocity velocity parameter of eCMD function | | | |
| 16 | Input | rEndVelocity For eMoveContAbs or eMoveContRel commands, allows to define the end velocity | | | |
| 20 | Input | rAcceleration acceleration parameter of eCMD function | | | |
| 24 | Input | rDeceleration deceleration parameter of eCMD function | | | |
| 28 | Input | eDirection direction parameter of eCMD function | | iUserCamNumber number of the cam profile (see eCamIn command) | |
| 32 | Input | rMasterOffset see eCamIn command | | | |
| 36 | Input | rSlaveOffset see eCamIn command | | | |
| 40 | Input | rMasterScaling see eCamIn command | | | |
| 44 | Input | rSlaveScaling voir commande eCamIn | | | |
| 48 | Input | iCamStartMode see eCamIn command | | xPeriodicCam see eCamSel command | xMasterAbsolute see eCamSel command |
| 52 | Input | xSlaveAbsolute see eCamSel command | reserved | iGearNumerator numerator of the ratio for eGearIn command | |
| 56 | Input | uiGearDenominator denominator of the ratio for eGearIn command | | xWSSelect starting mode selection (warm/cold) for the application function blocks (See application function blocs commands) | byNumberOfInput1 Lexium Controller's input number allocated to the first input of eCMD function |
| 60 | Input | byNumberOfInput2 Lexium Controller's input number allocated to the second input of eCMD function | byNumberOfInput3 Lexium Controller's input number allocated to the third input of eCMD function | byNumberOfInput4 Lexium Controller's input number allocated to the fourth input of eCMD function | byNumberOfInput5 Lexium Controller's input number allocated to the fifth input of eCMD function |
| 64 | Input | byNumberOfInput6 Lexium Controller's input number allocated to the sixth input of eCMD function | byNumberOfOutput4 Lexium Controller's output number allocated to the fourth output of eCMD function | byNumberOfOutput5 Lexium Controller's output number allocated to the fifth output of eCMD function | byNumberOfOutput6 Lexium Controller's output number allocated to the sixth output of eCMD function |
| 68 | Input | rLengthToCut see eFlyingShearIn and eRotaryKnife commands | | | |
| 72 | Input | rErrorStopDec see eFlyingShearIn and eRotaryKnife commands | | | |
| 76 | Input | rSetPosPos position parameter of eSetPos function (see eSetPos page 40) | | | |

Application Template parameter table

| Offset (byte) | Var Type | LL Byte + 0 | LH Byte +1 | HL Byte +2 | HH Byte +3 |
|---------------|----------|---|---|---|---|
| 80 | Input | xSetPosMode mode of eSetPos command (see command eSetPos page 40) | xWindowOnly validation of the TPx capture in a determined window only | reserved | reserved |
| 84 | Input | rFirstPosition starting position for the TPx capture window | | | |
| 88 | Input | rLastPosition final position for the TPx capture window | | | |
| 92 | Input | iTPNumber Touch probe number (TPx) | | reserved | reserved |
| 96 | Output | xActive = 1: a command is active | reserved | eActiveCMD number of the active command | |
| 100 | Output | xCMDDone = 1: eCMD command is done | xParallelCMDDone end of execution of parallel command | xHomeOk = 1: homing is done | xSynchron =1: the slave is synchronised with the master |
| 104 | Output | xEndOfProfile = 1: CAM profile is done (see eCamIn command) | xError = 1: axis is in error | iErrorID identification number of the error | |
| 108 | Output | sErrorMsg error message | | | |
| 112 | Output | | | | |
| 116 | Output | | | | |
| 120 | Output | | | | |
| 124 | Output | | | | |
| 128 | Output | | | | |
| 132 | Output | | | | |
| 136 | Output | | | | |
| 140 | Output | | | | |
| 144 | Output | | | | |
| 148 | Output | reserved | | nAxisState axis status in PLCopen diagram | |
| 152 | Output | rSetPosition axis current position | | | |
| 156 | Output | rSetVelocity axis current velocity | | | |
| 160 | Output | xPowerOn = 1: power is validated on the axis. | reserved | reserved | reserved |
| 164 | Output | rRecordedPosition TPx captured position | | | |
| 168 | Output | pAxisRefPtr • pointer to Axis ref • allows to easily access to Axis ref parameters | | | |
| 172 | Output | pAxisParPtr • pointer to axis parameters • allows to easily access to axis parameters | | | |
| 176 | Input | iTPSens TPx's polarity of capture (rising edge or falling edge) | | reserved | |
| 180 | Input | stCANParameter write a parameter in a servo drive using its CANopen address. Used with eWritePar command (see page 41). See details of this structure below | | | |
| 188 | Input | pAdrOfInput inputs' module address (=0 for local inputs) can be used for CANopen I/O extensions | | | |
| 192 | Input | pAdrOfOutput outputs' module address (=0 for local outputs) can be used for CANopen I/O extensions | | | |

Application Template parameter table

| Offset (byte) | Var Type | LL Byte + 0 | LH Byte +1 | HL Byte +2 | HH Byte +3 |
|---------------|----------|-------------|------------|------------|------------|
| 196 | | reserved | | reserved | |
| 200 | | reserved | | reserved | |
| 204 | | reserved | | reserved | |
| 208 | | reserved | | reserved | |

Details of stCANParameter array:

| Offset (byte) | LL Byte + 0 | LH Byte +1 | HL Byte +2 | HH Byte +3 |
|---------------|---|------------|---|--|
| 180 | wIndex: Index of the CANopen parameter to write | | bySubIndex Sub-index of the CANopen parameter | byLength Length of the parameter |
| 184 | dwValue value to write | | | |

Interface Master encoder

This interface allows, through an array of words, to control the status and to send commands to the master encoder.

| Variable Name | Address | Data Type |
|------------------------|------------|--------------------------|
| gMB_stEncoderInterface | AT %MW1254 | TPL_EncoderInterfaceType |

Size of TPL_EncoderInterfaceType Structure: 44 Bytes = 22 Words

| Offset (bytes) | LL Byte + 0 | LH Byte +1 | HL Byte +2 | HH Byte +3 |
|-----------------------|---|--|--|--|
| 0 | rFirstPosition starting position for the TPx capture window | | | |
| 4 | rLastPosition final position for the TPx capture window | | | |
| 8 | iTPNumber Touch probe number (TPx) | | i iTPSens TPx's polarity of capture (rising edge or falling edge) | |
| 12 | rRecordedPosition TPx captured position | | | |
| 16 | rSetPosPos position parameter of eSetPos function (see eSetPos page 40) | | | |
| 20 | xSetPosMode > mode of eSetPos command (see command eSetPos page 40) | xWindowOnly validation of the TPx capture in a determined window only | xActive = 1: a command is active | xError = 1: axis is in error |
| 24 | iErrorID identification number of the error | | reserved | |
| 28 | pAxisRefPtr <ul style="list-style-type: none">• pointer to Axis ref• allows to easily access to Axis ref parameters | | | |
| 32 | pAxisParPtr <ul style="list-style-type: none">• pointer to axis parameters• allows to easily access to axis parameters | | | |
| 36 | rPosition current position | | | |
| 40 | rVelocity current velocity | | | |

Application Template parameter table

Axis parameters

This array is used for reel or virtual axes configuration parameters (see page 9).
Each axis has its own array.

| Variable Name | Address | Data Type |
|----------------------|------------|---------------------------------------|
| gMB_astAxisParameter | AT %MW1276 | ARRAY [1..9] OF TPL_AxisParameterType |

Length of the array : 36 words / 72 bytes

| Axis | Axis default name | Array address |
|----------------|-------------------|---------------|
| Real axis 1 | Axis1 | %MW 1276 |
| Real axis 2 | Axis2 | %MW 1312 |
| Real axis 3 | Axis3 | %MW 1348 |
| Real axis 4 | Axis4 | %MW 1384 |
| Real axis 5 | Axis5 | %MW 1420 |
| Real axis 6 | Axis6 | %MW 1456 |
| Real axis 7 | Axis7 | %MW 1492 |
| Real axis 8 | Axis8 | %MW 1528 |
| Virtual axis | VirtAxis | %MW 1564 |
| Master encoder | Master | %MW 1600 |

| Offset | LL Byte + 0 | LH Byte +1 | HL Byte +2 | HH Byte +3 |
|--------|--|------------|---|------------|
| | SAxisName axis name | | | |
| 12 | iMovementType axis type: rotary (= 0) or linear (=1) axis | | reserved | reserved |
| 16 | rModulo modulo of the axis | | | |
| 20 | xSoftLimitEnable software limit enable | reserved | reserved | reserved |
| 24 | rSoftLimitPos maximum software limit position for a linear axis | | | |
| 28 | rSoftLimitNeg minimum software limit position for a linear axis | | | |
| 32 | eRampType ramp type (trapezoid or sinus) | | iNum numerator of the scaling factor | |
| 36 | dwDenom denominator of the scaling factor | | | |
| 40 | uiHMethod homing method | | uiHMn homing speed | |
| 44 | uiKPn velocity loop proportional gain | | uiTNn velocity loop integral time | |
| 48 | uiKPP position loop proportional gain | | uiKFPP predictive control factor (VelFeedFor) | |
| 52 | sType type of servo drive | | | |
| 56 | | | | |
| 60 | | | | |
| 64 | | | | |
| 68 | reserved | | wChecksum checksum calculation, used by the application to control the consistency | |

Application Template parameter table

Master encoder parameters

This array is used for master encoder configuration parameters (see page 14)

| Variable Name | Address | Data Type |
|------------------------|------------|-------------------------------|
| gMB_stEncoderParameter | AT %MW1600 | TPL_ TPL_EncoderParameterType |

Length of array: 24 words / 48 bytes

| Offset | LL Byte + 0 | LH Byte +1 | HL Byte +2 | HH Byte +3 |
|--------|--|---|---|------------|
| 0 | S <i>AxisName</i> axis name | | | |
| 4 | | | | |
| 8 | | | | |
| 12 | i <i>MovementType</i> axis type: rotary (= 0) or linear (=1) axis | | b <i>CodeType</i> type of code used for the encoder: BIN: Binary code GRAY: Gray code | Réservé |
| 16 | r <i>Modulo</i> modulo of the axis | | | |
| 20 | x <i>SoftLimitEnable</i> software limit enable | b <i>EncoderType</i> type of encoder: INC: Incremental encoder SSI: SSI encoder | w <i>HeaderBits</i> number of header bits | |
| 24 | r <i>SoftLimitPos</i> maximum software limit position for a linear axis | | | |
| 28 | r <i>SoftLimitNeg</i> minimum software limit position for a linear axis | | | |
| 32 | i <i>Num</i> numerator of the scaling factor | | w <i>StatusBits</i> number of frame bits for the status | |
| 36 | dw <i>Denom</i> denominator of the scaling factor | | | |
| 40 | w <i>SSiSingleTurnBits</i> number of frame bits for the position in the turn | | w <i>SSiMultiTurnBits</i> number of frame bits for the turn number | |
| 44 | w <i>Parity</i> used to activate or deactivate the odd or even parity bit | | w <i>Checksum</i> checksum calculation, used by the application to control the consistency | |

Servo drives configuration

Lexium Controller can save the complete configuration of all the servo drives connected. (see page 7)

Each servo drive's configuration is stored in 300 words / 600 bytes array.

| Variable Name | Address | Data Type |
|---------------------|------------|--|
| gMB_aaLXMPParameter | AT %MW1624 | ARRAY [1..8] OF TypeParameterSetVar_SM |

| Axis | Axis default name | Array address |
|-------------|-------------------|---------------|
| Real axis 1 | Axis1 | %MW 1624 |
| Real axis 2 | Axis2 | %MW 1944 |
| Real axis 3 | Axis3 | %MW 2264 |
| Real axis 4 | Axis4 | %MW 2584 |
| Real axis 5 | Axis5 | %MW 2904 |
| Real axis 6 | Axis6 | %MW 3224 |
| Real axis 7 | Axis7 | %MW 3544 |
| Real axis 8 | Axis8 | %MW 3864 |

Application Template parameter table

Teach Position Table

The values stored in the teach position table with the 'Teach' button (see page 18) are accessible (in write or read mode)

It is possible to store 32 positions, each with a comment (12 digits)

| Variable Name | Address | Data Type |
|------------------|------------|-------------------------------|
| gMB_astTeachList | AT %MW4186 | ARRAY[1..32] OF TPL_TeachType |

The description below gives access to the arrays of each stored position.

| Line | Start address | Line | Start address | Line | Start address | Line | Start address |
|------|---------------|------|---------------|------|---------------|------|---------------|
| 1 | %MW 4186 | 9 | %MW 4250 | 17 | %MW 4314 | 25 | %MW 4378 |
| 2 | %MW 4194 | 10 | %MW 4258 | 18 | %MW 4322 | 26 | %MW 4386 |
| 3 | %MW 4202 | 11 | %MW 4266 | 19 | %MW 4330 | 27 | %MW 4394 |
| 4 | %MW 4210 | 12 | %MW 4274 | 20 | %MW 4338 | 28 | %MW 4402 |
| 5 | %MW 4218 | 13 | %MW 4282 | 21 | %MW 4346 | 29 | %MW 4410 |
| 6 | %MW 4226 | 14 | %MW 4290 | 22 | %MW 4354 | 30 | %MW 4418 |
| 7 | %MW 4234 | 15 | %MW 4298 | 23 | %MW 4362 | 31 | %MW 4426 |
| 8 | %MW 4242 | 16 | %MW 4306 | 24 | %MW 4370 | 32 | %MW 4434 |

Array length (for each position) : 16 bytes / 8 words

Array content :

| Offset(byte) | LL Byte + 0 | LH Byte + 1 | HL Byte + 2 | HH Byte + 3 |
|---------------|--|-------------|-------------|-------------|
| 0 | sSource linked comment, Axis name by default | | | |
| 4 | | | | |
| 8 | | | | |
| 12 | rPosition stored position | | | |

Active Errors parameters list

Each active error in the Lexium Controller is stored with its parameters in an array.
A maximum of 16 errors can be active simultaneously

| Variable Name | Address | Data Type |
|------------------|------------|-------------------------------|
| gMB_astErrorList | AT %MW4444 | ARRAY[1..16] OF TPL_ErrorType |

The description below gives access to the array of each active error

Error arrays addresses :

| Error | Array address | Error | Array address |
|------------|---------------|-------------|---------------|
| Error N° 1 | %MW 4444 | Error N° 9 | %MW 4636 |
| Error N° 2 | %MW 4468 | Error N° 10 | %MW 4660 |
| Error N° 3 | %MW 4492 | Error N° 11 | %MW 4684 |
| Error N° 4 | %MW 4516 | Error N° 12 | %MW 4708 |
| Error N° 5 | %MW 4540 | Error N° 13 | %MW 4732 |
| Error N° 6 | %MW 4564 | Error N° 14 | %MW 4756 |
| Error N° 7 | %MW 4588 | Error N° 15 | %MW 4780 |
| Error N° 8 | %MW 4612 | Error N° 16 | %MW 4804 |

Array length: 48 bytes / 24 words

Application Template parameter table

Array content:

| Offset | LL Byte + 0 | LH Byte +1 | HL Byte +2 | HH Byte +3 |
|--------|--|------------|--|------------|
| 0 | iErrorId error identification number | | wErrorReaction reaction type | |
| 4 | timErrorTime date and time when error appeared | | | |
| 8 | sErrorMsg Error Message | | | |
| 12 | | | | |
| 16 | | | | |
| 20 | | | | |
| 24 | | | | |
| 28 | | | | |
| 32 | | | | |
| 36 | | | | |
| 40 | | | | |
| 44 | | | | |

For more details concerning the error management see page [90](#).

Logged Errors parameters list

The description below gives access to the array of each active error

| Variable Name | Address | Data Type |
|-------------------|------------|--------------------------------|
| gMB_astLoggerList | AT %MW4830 | ARRAY[1..32] OF TPL_ ErrorType |

Error arrays addresses:

| Error | Array address | Error | Array address |
|-------------|---------------|-------------|---------------|
| Error N° 1 | %MW 4830 | Error N° 17 | %MW 5214 |
| Error N° 2 | %MW 4854 | Error N° 18 | %MW 5238 |
| Error N° 3 | %MW 4878 | Error N° 19 | %MW 5262 |
| Error N° 4 | %MW 4902 | Error N° 20 | %MW 5286 |
| Error N° 5 | %MW 4926 | Error N° 21 | %MW 5310 |
| Error N° 6 | %MW 4950 | Error N° 22 | %MW 5334 |
| Error N° 7 | %MW 4974 | Error N° 23 | %MW 5358 |
| Error N° 8 | %MW 4998 | Error N° 24 | %MW 5382 |
| Error N° 9 | %MW 5022 | Error N° 25 | %MW 5406 |
| Error N° 10 | %MW 5046 | Error N° 26 | %MW 5430 |
| Error N° 11 | %MW 5070 | Error N° 27 | %MW 5454 |
| Error N° 12 | %MW 5094 | Error N° 28 | %MW 5478 |
| Error N° 13 | %MW 5118 | Error N° 29 | %MW 5502 |
| Error N° 14 | %MW 5142 | Error N° 30 | %MW 5526 |
| Error N° 15 | %MW 5166 | Error N° 31 | %MW 5550 |
| Error N° 16 | %MW 5190 | Error N° 32 | %MW 5574 |

Array length: 48 bytes / 24 words

The array structure is exactly the same than for the active error list.

For more details concerning the error management see page [90](#).

Application Template parameter table

Cam Profile parameters

The cam profile parameters created with the Application Template (see page 46) are stored in predefined arrays.

| Variable Name | Address | Data Type |
|----------------|-------------|--------------------------------|
| gMB_astUserCam | AT %MW 5598 | ARRAY[1..8] OF TPL_UserCamType |

The description below gives access to the parameters of each cam profile.

Array address for each profile:

| Cam Profile | Array address | Cam Profile | Array address |
|-------------|---------------|-------------|---------------|
| CAM1 | %MW 5598 | CAM5 | %MW 6350 |
| CAM2 | %MW 5786 | CAM6 | %MW 6538 |
| CAM3 | %MW 5974 | CAM7 | %MW 6726 |
| CAM4 | %MW 6162 | CAM8 | %MW 6914 |

Array length: 376 bytes / 188 words

Array content:

| Offset | LL Byte + 0 | LH Byte +1 | HL Byte +2 | HH Byte +3 |
|--------|--|--|--|------------|
| 0 | wCamStructID cam profile identification | | reserved | reserved |
| 4 | xStart cam master start position | | | |
| 8 | xEnd cam master end position | | | |
| 12 | byType define cam profile type = 3 by default for polynomial profile between 2 points See MC_CAM_REF structure description in Motion Pro On line Help | byVarType used if byType = 1 or = 2 See MC_CAM_REF structure description in Motion Pro On line Help | nElements cam profile number of points | |
| 16 | byInterpolationQuality interpolation quality = 1 : linear (default) = 2 : cubic See MC_CAM_REF structure description in Motion Pro On line Help | reserved | reserved | reserved |
| 20 | pce | | | |
| 24 | nTappets number of configured cam switches | | reserved | reserved |
| > | reserved | reserved | | |
| 28 | pt pointer to cam switch structure. See SMC_CAMTappet structure description in Motion Pro On line Help | | | |
| 32 | dwTappetActiveBits | | | |
| 36 | strCAMName Cam profile name | | | |
| 116 | reserved | reserved | bChangedOnline | reserved |
| 120 | dx[0] first point, master position | | | |
| 124 | dy[0] first point, Slave position | | | |
| 128 | dv[0] first point, local velocity at the point. Gives the angle of the profile at this point. | | | |

Application Template parameter table

| Offset | LL Byte + 0 | LH Byte +1 | HL Byte +2 | HH Byte +3 |
|--------|--|------------|------------|------------|
| 132 | dA[0] first point, local acceleration at the point. Gives the curvature of the profile at this point | | | |
| 136 | dX[1] second point, master position | | | |
| 140 | dY[1] second point, Slave position | | | |
| 144 | dV[1] second point, local velocity at the point. Gives the angle of the profile at this point. | | | |
| 148 | dA[1] second point, local acceleration at the point. Gives the curvature of the profile at this point | | | |
| . | . | . | . | . |
| 360 | dX[15] sixteenth point, master position | | | |
| 364 | dY[15] sixteenth point, Slave position | | | |
| 368 | dV[15] sixteenth point, local velocity at the point. Gives the angle of the profile at this point. | | | |
| 372 | dA[15] sixteenth point, local acceleration at the point. Gives the curvature of the profile at this point | | | |

Motion Task table parameters

The application template allows configuring a motion task table (see page 19)
The description below gives access to all the parameters of this motion task table

| Variable Name | Address | Data Type |
|----------------|------------|------------------------------------|
| gMB_astMTIList | AT %MW7600 | ARRAY[1..75] OF TPL_MotionTaskType |

Each Motion task is described by an array with a command and parameters.
Motion Task array addresses:

| Line | Address | Line | Address | Line | Address | Line | Address |
|------|----------|------|-----------|------|-----------|------|-----------|
| 1 | %MW 7600 | 17 | %MW 8976 | 33 | %MW 10352 | 49 | %MW 11728 |
| 2 | %MW 7686 | 18 | %MW 9062 | 34 | %MW 10438 | 50 | %MW 11814 |
| 3 | %MW 7772 | 19 | %MW 9148 | 35 | %MW 10524 | 51 | %MW 11900 |
| 4 | %MW 7858 | 20 | %MW 9234 | 36 | %MW 10610 | 52 | %MW 11986 |
| 5 | %MW 7944 | 21 | %MW 9320 | 37 | %MW 10696 | 53 | %MW 12072 |
| 6 | %MW 8030 | 22 | %MW 9406 | 38 | %MW 10782 | 54 | %MW 12158 |
| 7 | %MW 8116 | 23 | %MW 9492 | 39 | %MW 10868 | 55 | %MW 12244 |
| 8 | %MW 8202 | 24 | %MW 9578 | 40 | %MW 10954 | 56 | %MW 12330 |
| 9 | %MW 8288 | 25 | %MW 9664 | 41 | %MW 11040 | 57 | %MW 12416 |
| 10 | %MW 8374 | 26 | %MW 9750 | 42 | %MW 11126 | 58 | %MW 12502 |
| 11 | %MW 8460 | 27 | %MW 9836 | 43 | %MW 11212 | 59 | %MW 12588 |
| 12 | %MW 8546 | 28 | %MW 9922 | 44 | %MW 11298 | 60 | %MW 12674 |
| 13 | %MW 8632 | 29 | %MW 10008 | 45 | %MW 11384 | 61 | %MW 12760 |
| 14 | %MW 8718 | 30 | %MW 10094 | 46 | %MW 11470 | 62 | %MW 12846 |
| 15 | %MW 8804 | 31 | %MW 10180 | 47 | %MW 11556 | 63 | %MW 12932 |
| 16 | %MW 8890 | 32 | %MW 10266 | 48 | %MW 11642 | 64 | %MW 13018 |

Application Template parameter table

Array length: 172 bytes / 86 words

Array content:

| Offset | LL Byte + 0 | LH Byte +1 | HL Byte +2 | HH Byte +3 |
|--------|--|------------|---|------------|
| 0 | iStepNr Line number | | eCommand number of executed command See blow this array, the list of Motion commands and numbers | |
| 4 | xBuffer enables buffer mode | reserved | iAxisNumber axis number | |
| 8 | diTimeOut execution time out (=0, no time out) | | | |
| 12 | sName_P1 First parameter name | | | |
| 24 | rValue_P1 First parameter value | | | |
| 28 | sName_P2 Second parameter name | | | |
| 40 | rValue_P2 Second parameter value | | | |
| 44 | sName_P3 | | | |
| 56 | rValue_P3 | | | |
| 60 | sName_P4 | | | |
| 72 | rValue_P4 | | | |
| 76 | sName_P5 | | | |
| 88 | rValue_P5 | | | |
| 92 | sName_P6 | | | |
| 104 | rValue_P6 | | | |
| 108 | sName_P7 | | | |
| 120 | rValue_P7 | | | |
| 124 | sName_P8 | | | |
| 136 | rValue_P8 | | | |
| 140 | sName_P9 | | | |
| 152 | rValue_P9 | | | |
| 156 | sName_P10 | | | |
| 168 | rValue_P10 | | | |

Application Template parameter table

Command names and numbers:

| Command Nbr | Command Name | Command Nbr | Command Name | Command Nbr | Command Name |
|-------------|---------------|-------------|--------------|-------------|--------------|
| 1 | eMoveAbs, | 12 | eHome, | 23 | eRKnifeStop, |
| 2 | eMoveRel, | 13 | eCamSel, | 24 | eMoveSupl, |
| 3 | eMoveAdd, | 14 | eCamIn, | 25 | ePhasing, |
| 4 | eMoveVel, | 15 | eCamOut, | 26 | eSetPos, |
| 5 | eMoveContRel, | 16 | eGearIn, | 27 | eWritePar, |
| 6 | eMoveContAbs, | 17 | eGearOut, | 28 | eClampIn, |
| 7 | eSTOP, | 18 | eFShearIn, | 29 | eClampOut, |
| 8 | ePowerOn, | 19 | eFShearOut, | 30 | eStrippIn, |
| 9 | ePowerOnAll, | 20 | eFShearStop, | 31 | eStrippOut, |
| 10 | ePowerOff, | 21 | eRKnifeIn, | 32 | eAccuIn, |
| 11 | ePowerOffAll, | 22 | eRKnifeOut, | 33 | eAccuOut, |

For further information regarding the parameters and the behavior of the functions see page [23](#).

Flying Shear and Rotary Knife parameters

The array below gives for each configured axis the addresses of the parameters of Flying shear and Rotary Knife application function blocks.

| Variable Name | Address | Data Type |
|-----------------------------|------------|---|
| gMB_astFlyingShearParameter | AT %MW7102 | ARRAY[1..9] OF AFB_ST_FlyingShearParameter |

Array length: 108 bytes = 54 words

Array content:

| Offset | LL Byte + 0 | LH Byte +1 | HL Byte +2 | HH Byte +3 |
|--------|-------------------|------------|------------|------------|
| 0 | rMaxSlidePosition | | | |
| 4 | rMinSlidePosition | | | |
| 8 | rRestPos | | | |
| 12 | rMaxBackmoveVel | | | |
| 16 | rMaxBackmoveAcc | | | |
| 20 | rSyncEndPos | | | |
| 24 | rStopDec | | | |
| 28 | diWS_Mode | | | |
| 32 | rWS_Window | | | |
| 36 | rWS_Vel | | | |
| 40 | rWS_AccDec | | | |
| 44 | iTP_Number | | xTP_Mode | reserved |
| 48 | rTP_SP | | | |
| 52 | rTP_Window | | | |
| 56 | diOpMode | | reserved | reserved |
| 60 | rKnifePerimeter | | | |
| 64 | diMaxNumMissedTP | | | |
| 68 | rm | | | |
| 72 | rCAMOffset | | | |
| 76 | xOffsetMode | reserved | reserved | reserved |
| 80 | rOffsetStartPos | | | |
| 84 | rOffsetReachedPos | | | |
| 88 | rSlavePosition | | | |
| 92 | rMasterPosition | | | |
| 96 | xInValidTP | reserved | reserved | reserved |
| 100 | rMinTP_SP | | | |
| 104 | rMasterStartPhase | | | |

For more details about behavior and variable signification, see on line help.

