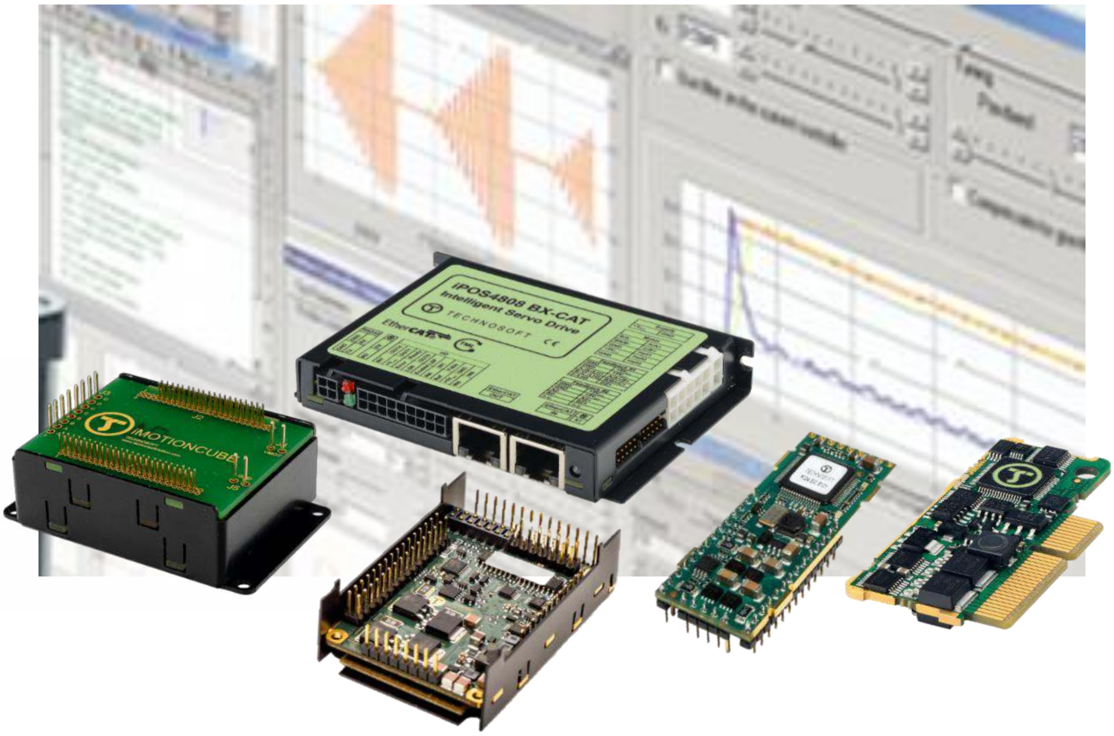
EasyMotion Studio

Quick Setup and Programming Guide

TECHNOSOFT

Your Next Intelligent Move



Technical Reference

Preliminary

Table of contents

1 About EasyMotion Studio

EasyMotion Studio is a software that allows the setup and motion programming of Technosoft intelligent drives and motors. The output of the EasyMotion Studio is a set of setup data and a motion program, which can be downloaded to the drive/motor EEPROM or saved on the PC for later use.

The setup component contains all the information needed to configure and parameterize a Technosoft drive/motor. This information is preserved in the drive/motor EEPROM in the *setup table*. The setup table is copied at power-on into the RAM memory of the drive/motor and is used during runtime.

The motion component contains the motion sequences to do. These are described via a TML (Technosoft Motion Language) program, which is executed by the drives/motors built-in motion controller.

EasyMotion Studio includes a set of evaluation tools like the Data Logger, the Control Panel and the Command Interpreter which help to quickly develop, test, measure and analyse the motion application.

2 Getting Started

A **demo version of EasyMotion Studio** can be downloaded from Technosoft web page (see this link).

The demo version of EasyMotion Studio can be converted to a full version by introducing a valid license number in the menu command **Help | Enter registration info…**

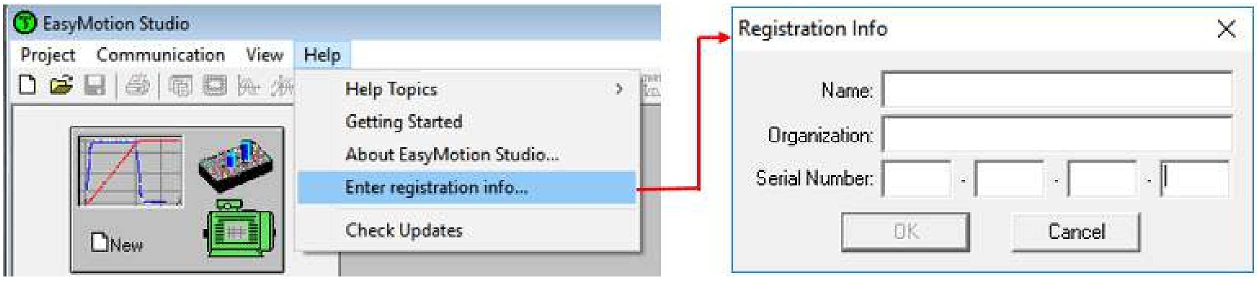


Figure 2.1 - Enter the registration info

After the software is installed, there are 3 major steps in operating the Technosoft drives:

Step 1: **Hardware installation**

The hardware installation is described in details in each drive user manual.

Step 2: **Drive commissioning**

The drive commissioning will be detailed in this document.

Step 3: **Motion programming**

The programming can be done using one of the following options:

- a CANopen master (in case of the CANopen drives);
- an EtherCAT master (in case of the EtherCAT drives);
- the drive's built-in motion controller executing a TML program developed in EasyMotion Studio;
- a TML_LIB motion library for PCs (Windows or Linux)
- a TML_LIB motion library for PLCs
- a distributed control approach which combines the above options, like for example a host calling motion functions programmed on the drives in TML.

This document details the drive programming using the EasyMotion Studio and the Technosoft Motion Language (TML).

The other programming option are detailed in the following user manuals:

- **CANopen Programming** manual - describes the CANopen drives programming;
- **CoE Programming User Manual** - describes the EtherCAT drives programming;
- **TML_LIB User manuals** - describes the project development using the PC / PLC libraries.

The above manuals can be downloaded from our website (link).

3 Establish the communication with the drive

The drive connection to the PC can be done in one of the following ways:

1. Using an RS232 link, directly connected to the PC or via an USB to RS232 adapter;
2. Using the Technosoft Ethernet to RS232 adapter;
3. Using a CAN-bus link, directly connected to the PC through a PC-CAN interface;
4. Using Technosoft Ethernet to CAN adapter
5. Via another drive from the same CAN-bus network, which is connected directly to the PC.

Once the drive connection to the PC is done, the system can be powered ON, to start the commissioning part, using the EasyMotion Studio software.

Use the **Communication | Setup** menu command to check / change the PC communication settings.

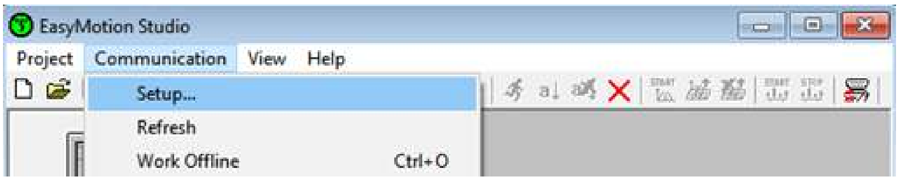
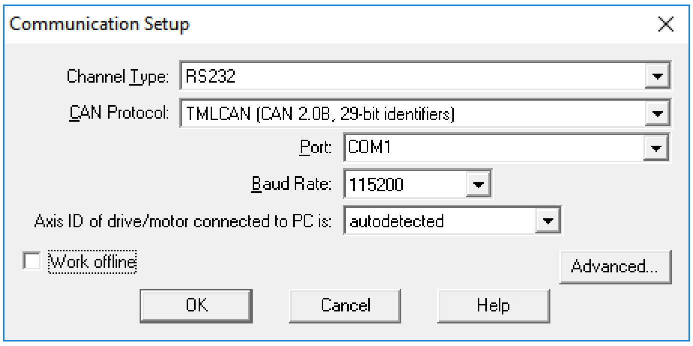


Figure 3.1 - Communication pop-up menu in EasyMotion Studio

The **Communication Setup** dialogue will open, allowing to select and configure the communication between the PC and the Technosoft drives/motors.


Communication Setup

Channel Type: RS232

CAN Protocol: TMLCAN (CAN 2.0B, 29-bit identifiers)

Port: COM1

Baud Rate: 115200

Axis ID of drive/motor connected to PC is: autodetected

□ Work offline

Figure 3.2 - Communication Setup dialogue

The detailed description of the fields inside this dialogue is included in the help topic that opens when the **Help** button from the Communication Setup dialogue is pressed.

After the communication parameters are set, if the **Work offline** option is unmarked and the dialogue is closed using the **OK** button, EasyMotion Studio should display the "**Online**" message in the status bar plus the axis ID of the connected drive/motor and its firmware version. Otherwise the "**Offline**" message is displayed and a communication error message is returned, explaining why the communication failed

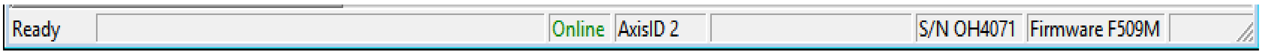

Ready | Online | AxisID 2 | S/N OH4071 | Firmware F509M

Figure 3.3 - Status bar indication when the communication with the drive succeeds

Remarks:

1. When first started, EasyMotion Studio tries to communicate via RS-232 and COM1 with a drive having axis ID=255 (default communication settings). If the drive has a different axis ID, select **Autodetected** option in the **Axis ID of drive/motor connected to PC** field from the **Communication Setup** window.
2. If the drive is connected to the PC through an **USB to RS232 adapter**, is recommended to access the adapter proprieties (in the Windows – **Device Manager** dialogue) and set the **Latency Timer** to **1 ms**. This will improve the communication speed.
3. If this drive is part of a CAN bus network and the PC is serially connected with another drive, use the menu command **Communication | Scan Network** to see the visible drives Axis IDs.

4 Create a new project

EasyMotion Studio starts with an empty window from where the user can create a new project or open a previously created one. The same options are also available in the “Project” pop-up menu.

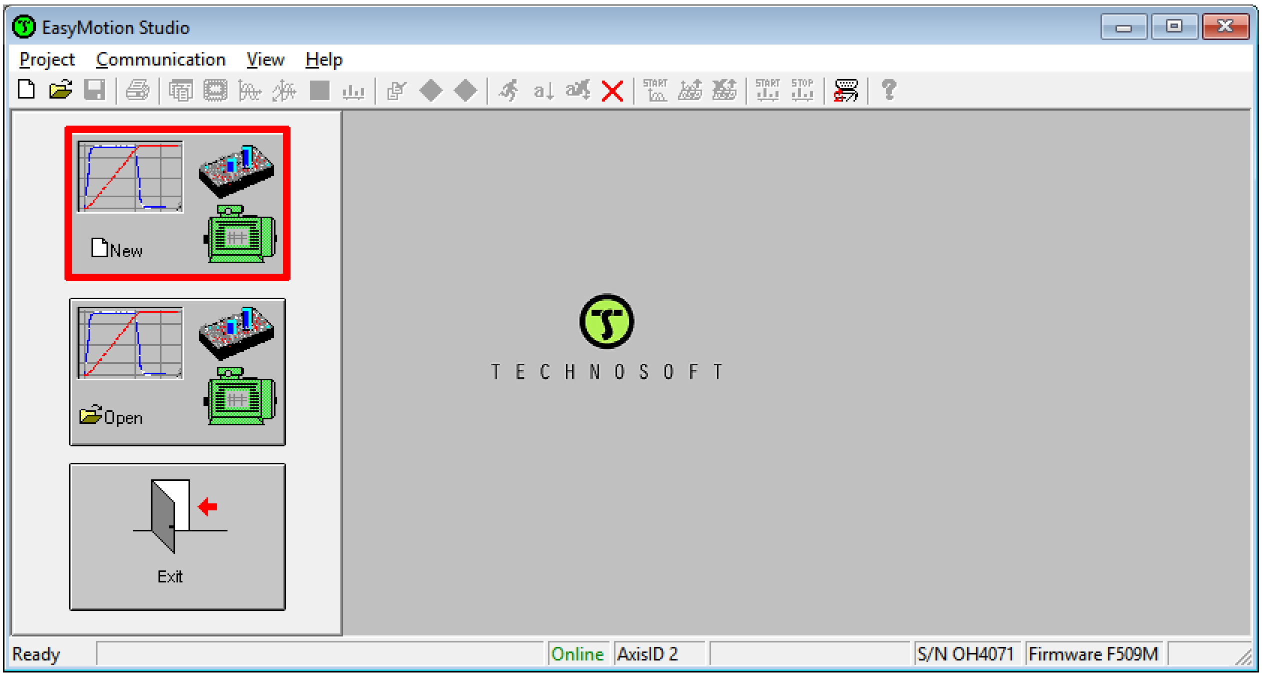
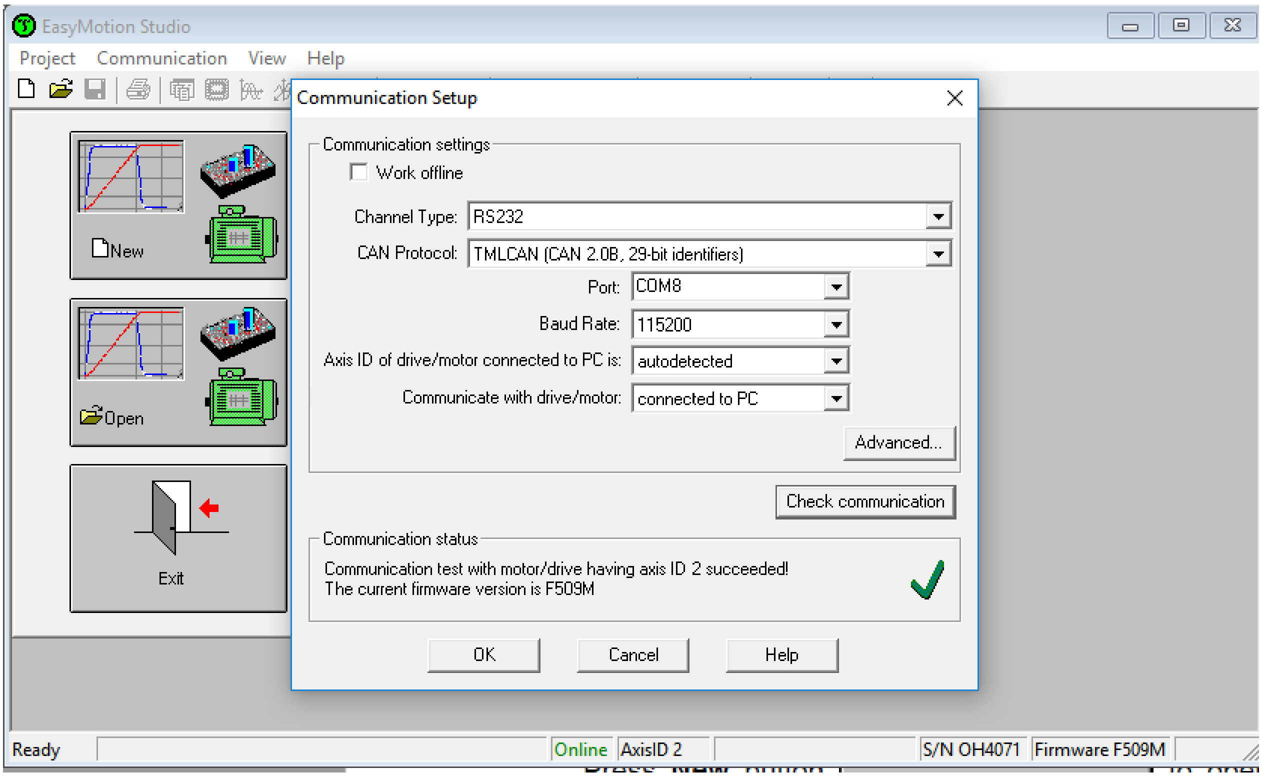


Figure 4.1 - Create a new project

Pressing the **New** button will open a small wizard that allows you to select the axis for which you want to create an application or change the communication settings.



Communication Setup

☐ Work offline

Channel Type: RS232

CAN Protocol: TMLCAN (CAN 2.0B, 29-bit identifiers)

Port: COM8

Baud Rate: 115200

Axis ID of drive/motor connected to PC is: autodetected

Communicate with drive/motor: connected to PC

Communication test with motor/drive having axis ID 2 succeeded!
The current firmware version is F509M ✓

Figure 4.2 - Chose the Axis ID for which the new project will be will be created

If there is not a drive connected to the PC or if you want to start the project being offline, you can check the **Work Offline** option in the above dialogue and continue to the setup, by pressing the **OK** button.

The next step in wizard allows to choose the drive and the motor technology

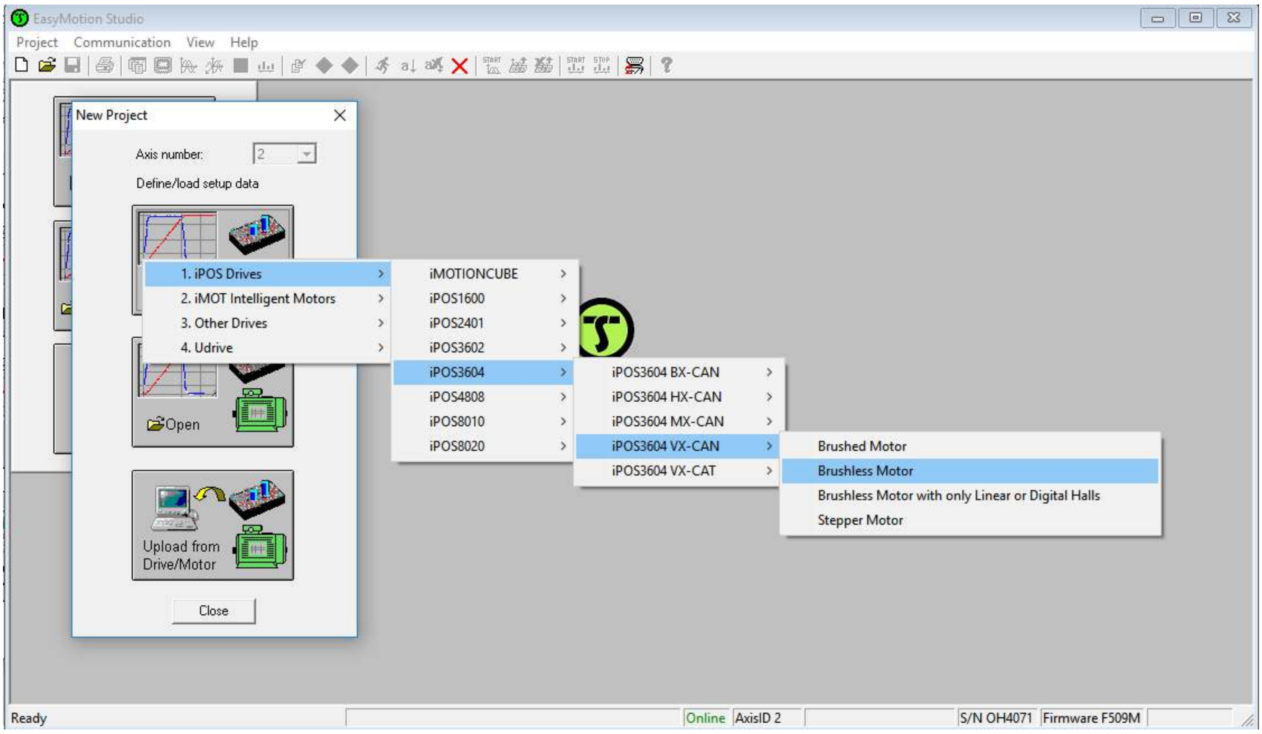
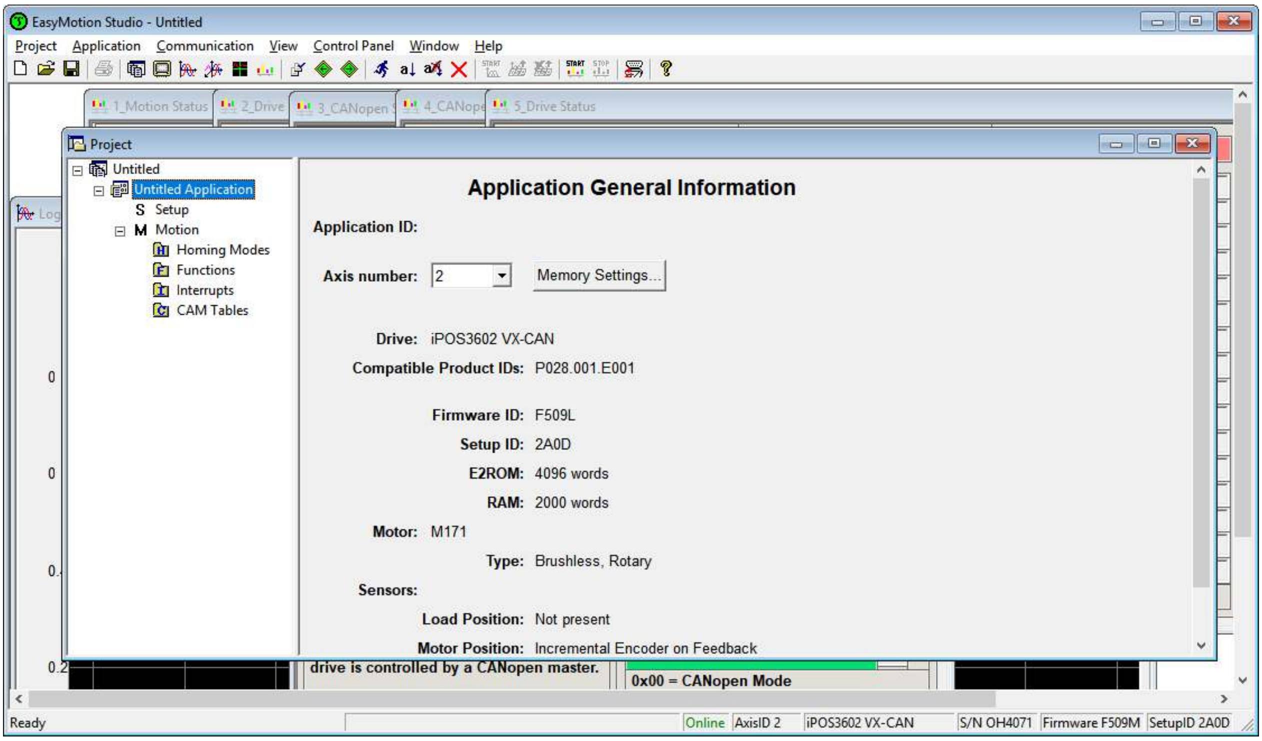


Figure 4.3 - Drive and motor technology selection

When you start a new project, EasyMotion Studio automatically creates a first application. Additional applications can be added later with the New, Duplicate or Insert menu commands under Application menu.

The Project windows contains the project tree (on the left side). The default name for the project and application are “Untitled”.



Application General Information

Application ID:

Axis number: 2

Drive: iPOS3602 VX-CAN

Compatible Product IDs: P028.001.E001

Firmware ID: F509L

Setup ID: 2A0D

E2ROM: 4096 words

RAM: 2000 words

Motor: M171

Type: Brushless, Rotary

Sensors:

Load Position: Not present

Motor Position: Incremental Encoder on Feedback

Figure 4.4 - Project window

The application has 2 components: Setup and Motion. The **Setup** allows to perform/edit the drive and motor setup, while the **Motion**, allows to build the TML program.

5 Drive and Motor setup

In the Project window, select “Setup”, to access the setup options.

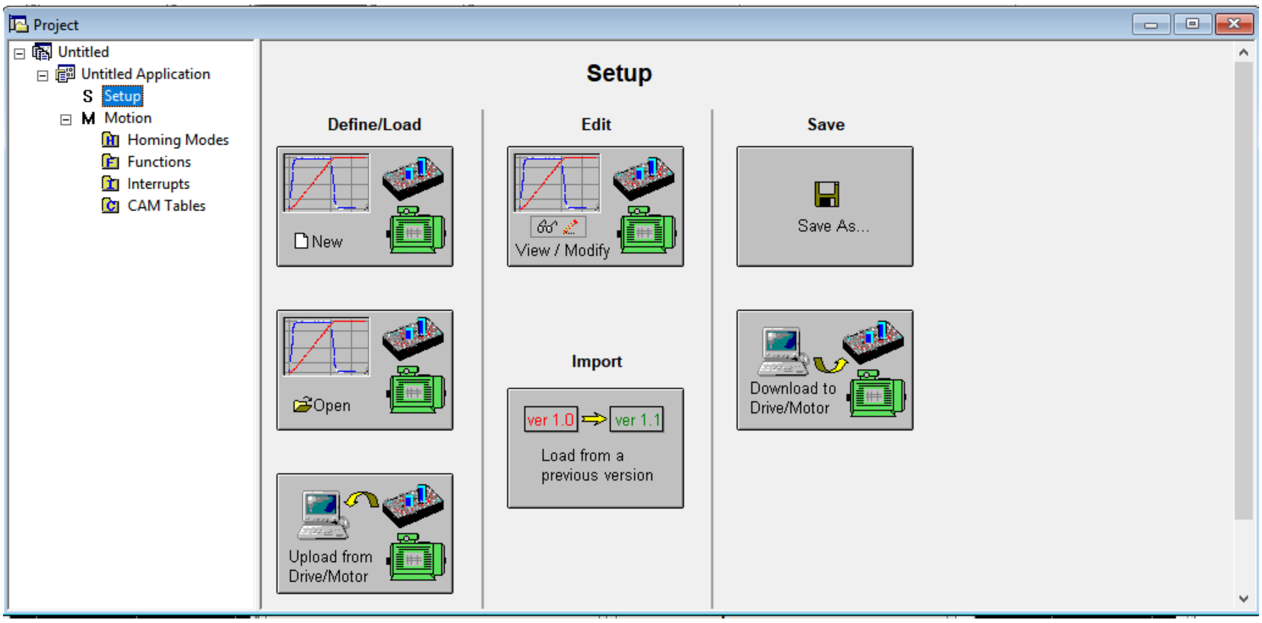
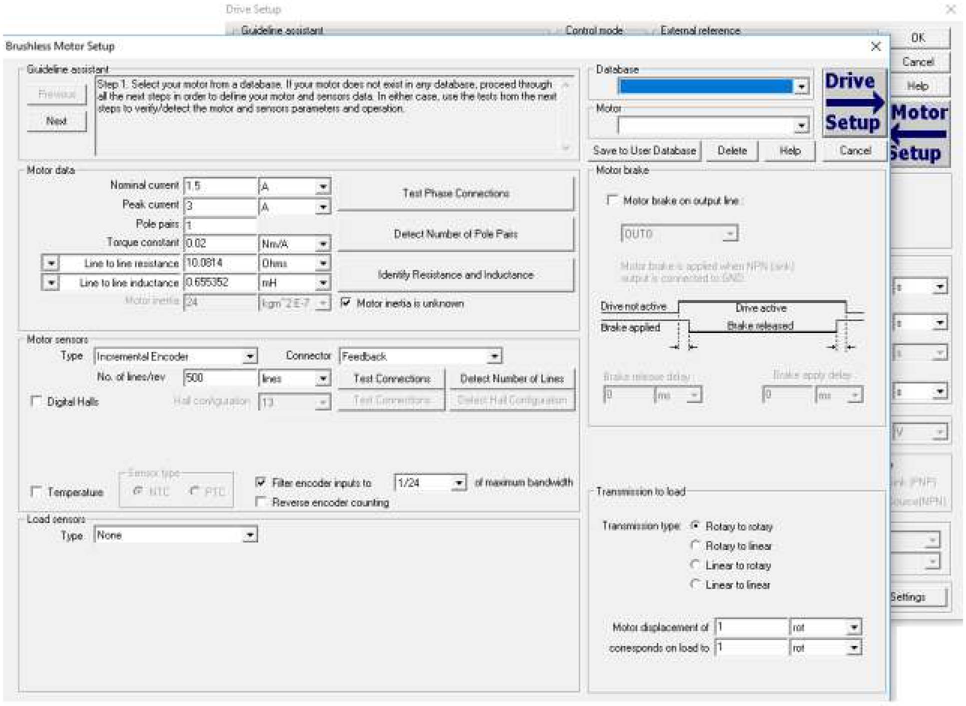


Figure 5.1 - Project window

The **New** button, allows you to change the drive type and/or the motor technology.
The **Open** button offers the possibility to open a previously saved setup.
The **Upload from Drive/Motor** button uploads the setup from the drive memory to the current project.
The **Save** button saves the current setup on the PC, to be used later.
The **Download to Drive/Motor** button downloads the current setup to the drive memory.

Press **View/Modify** button to open the **Motor Setup** and **Drive Setup** dialogs, through which the drive and the motor can be configured.



Guideline assistant: Step 1. Select your motor from a database. If your motor does not exist in any database, proceed through all the next steps in order to define your motor and sensors data. In either case, use the tests from the next steps to verify/detect the motor and sensors parameters and operation.

Motor data:

Nominal current	1.5	A
Peak current	3	A
Pole pairs	1	
Torque constant	0.02	Nm/A
Line to line resistance	10.0814	Ohms
Line to line inductance	0.655352	mH
Motor inertia	24	kgm^2 E-7

☑ Motor inertia is unknown

Motor sensors: Type: Incremental Encoder; Connector: Feedback; No. of lines/rev: 500 lines

□ Digital Halls; Hall configuration: 13

□ Temperature; Sensor type: ● NTC ○ PTC

☑ Filter encoder inputs to 1/24 of maximum bandwidth

□ Reverse encoder counting

Load sensors: Type: None

Database
Motor

Motor brake: □ Motor brake on output line: OUT0

Motor brake is applied when NPN (sink) output is connected to GND

Brake release delay: 0 ms; Brake apply delay: 0 ms

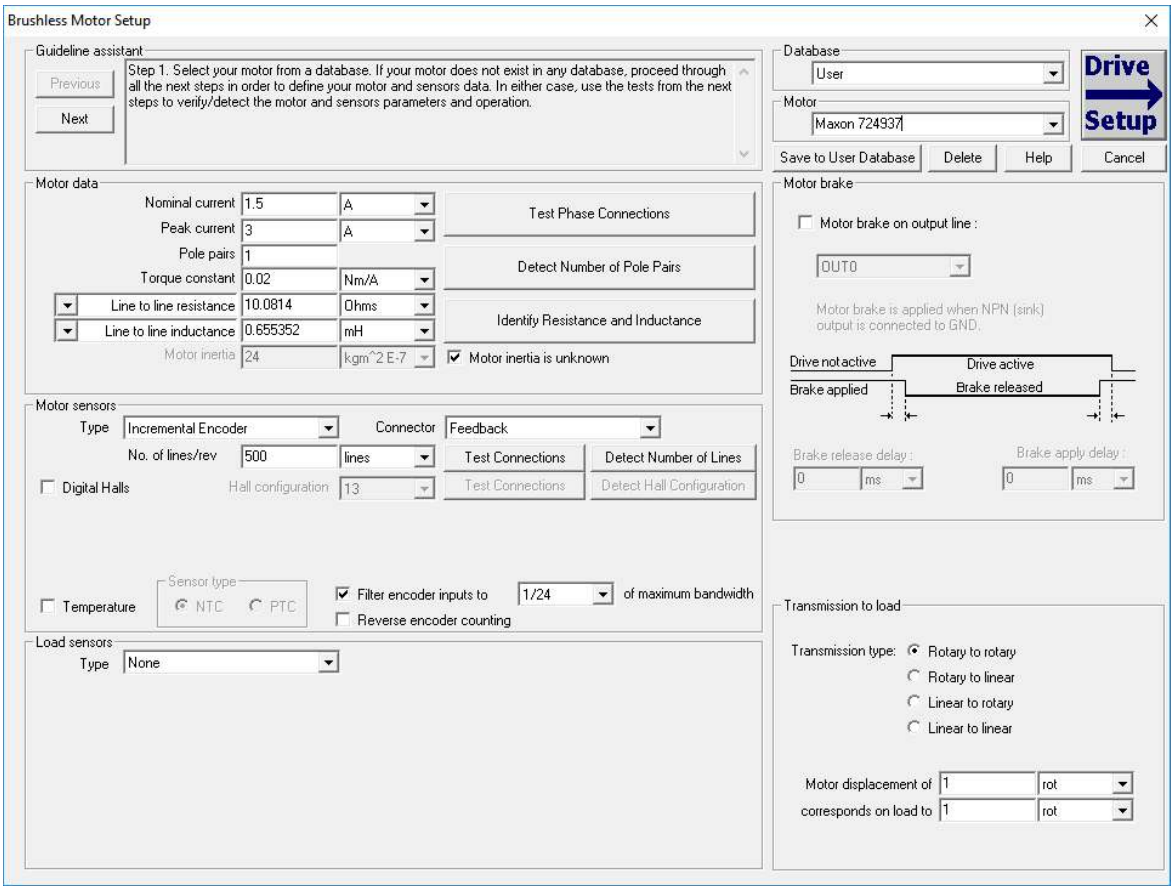
Transmission to load: Transmission type: ● Rotary to rotary ○ Rotary to linear ○ Linear to rotary ○ Linear to linear

Motor displacement of 1 rot corresponds on load to 1 rot

Figure 5.2 – Motor and Drive setup dialogs

5.1 Motor Setup

The **Motor setup** dialog allows to introduce the motor data and the associated sensors parameters. Data introduction is accompanied by a series of tests having as goal to check the connections to the drive and/or to determine or validate a part of the motor and sensors parameters.


Brushless Motor Setup

Guideline assistant: Step 1. Select your motor from a database. If your motor does not exist in any database, proceed through all the next steps in order to define your motor and sensors data. In either case, use the tests from the next steps to verify/detect the motor and sensors parameters and operation.

Database: User
Motor: Maxon 724937

Motor data
Nominal current: 1.5 A
Peak current: 3 A
Pole pairs: 1
Torque constant: 0.02 Nm/A
Line to line resistance: 10.0814 Ohms
Line to line inductance: 0.655352 mH
Motor inertia: 24 kgm^2 E-7
☑ Motor inertia is unknown

Motor sensors
Type: Incremental Encoder
Connector: Feedback
No. of lines/rev: 500 lines
□ Digital Halls
Hall configuration: 13
□ Temperature
Sensor type: ● NTC ○ PTC
☑ Filter encoder inputs to 1/24 of maximum bandwidth
□ Reverse encoder counting

Load sensors
Type: None

Motor brake
□ Motor brake on output line : OUT0
Motor brake is applied when NPN (sink) output is connected to GND.
Brake release delay : 0 ms
Brake apply delay : 0 ms

Transmission to load
Transmission type: ● Rotary to rotary ○ Rotary to linear ○ Linear to rotary ○ Linear to linear
Motor displacement of 1 rot
corresponds on load to 1 rot

Figure 5.3 - Motor setup dialog

The motor and sensor parameters can be found most of the time in the motor / sensor data sheet. Nevertheless for some parameters there are some tests that detects or check the respective parameters value.

The **Load sensors** is available only on for drives with 2 feedback inputs.

The **Motor brake** group box allows to use one of the drives digital inputs to command an electro-mechanical brake and set the brake parameters

The **Transmission to load** group box allows to set the type and ratio of the transmission between the motor and the load. The following options are available:

- rotary to rotary, when there is a gearbox between the motor and the load;
- rotary to linear, when the load is moved through a belt, ball screw, lead screw etc.
- linear to rotary, when there's a system that converts the linear motion of the motor to a rotary movement;
- linear to linear, when a linear motor is used (voice coil motor or brushless linear motor).

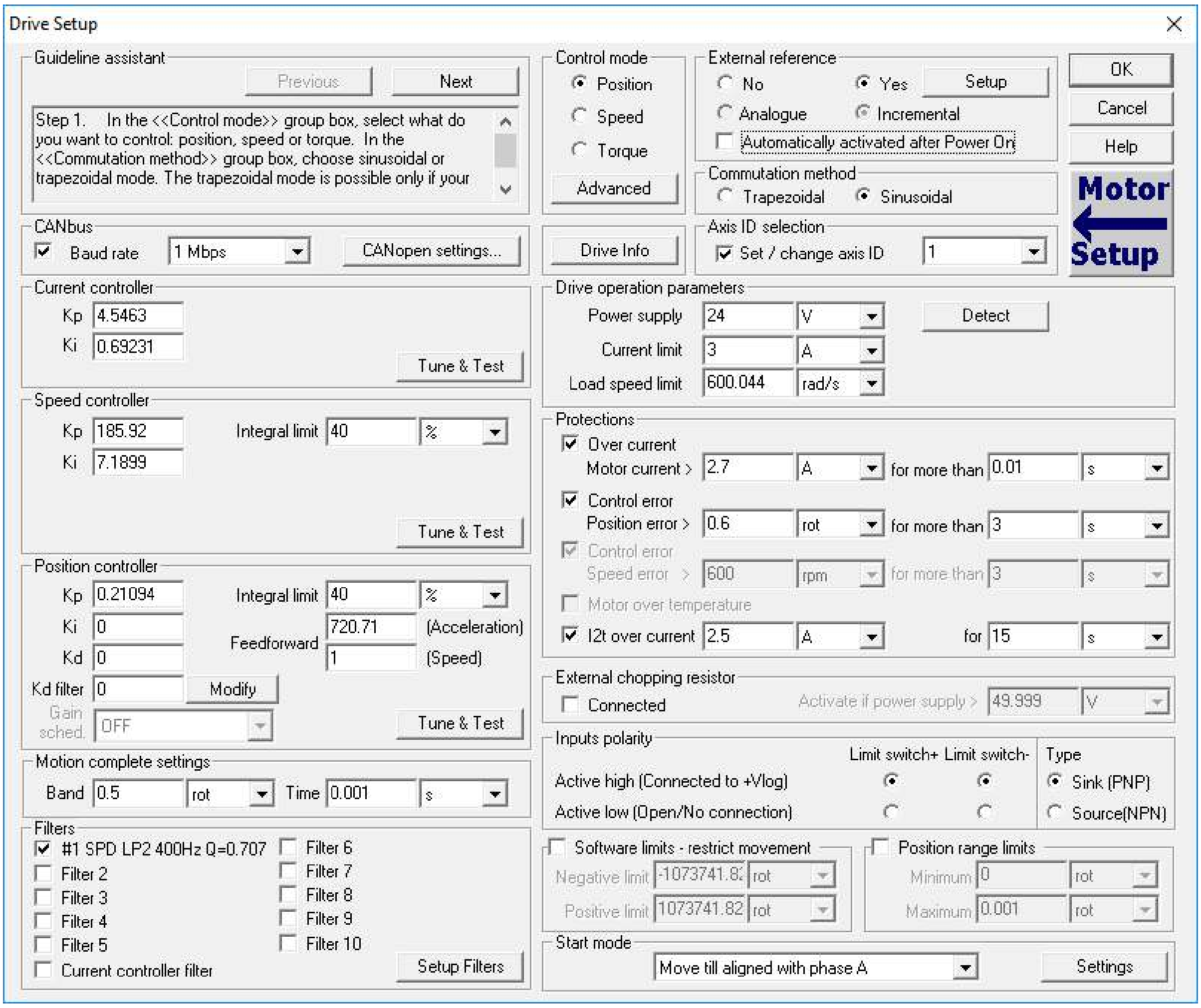
A detailed description of the parameters in the Motor Setup dialogue is included in the help topic that opens when the **Help** button in this dialogue is pressed.

The **Guideline assistant** in the top of the Motor Setup dialogue offer also a step by step guidance, to correctly perform the motor setup.

5.2 Drive Setup

The **Drive setup** dialogue allows to configure and parameterize the drive according to the application specifications.

The **Guideline Assistant** offers a guide through the whole process of introducing and/or checking the drive data. There's also the **Help** button that opens a topic in which all parameters inside the Drive Setup dialogue are detailed.


Drive Setup

Guideline assistant: Step 1. In the <<Control mode>> group box, select what do you want to control: position, speed or torque. In the <<Commutation method>> group box, choose sinusoidal or trapezoidal mode. The trapezoidal mode is possible only if your

Control mode: ● Position ○ Speed ○ Torque

External reference: ○ No ● Yes ○ Analogue ● Incremental; □ Automatically activated after Power On

Commutation method: ○ Trapezoidal ● Sinusoidal

CANbus: ☑ Baud rate 1 Mbps

Axis ID selection: ☑ Set / change axis ID 1

Current controller: Kp 4.5463; Ki 0.69231

Speed controller: Kp 185.92; Ki 7.1899; Integral limit 40 %

Position controller: Kp 0.21094; Ki 0; Kd 0; Kd filter 0; Integral limit 40 %; Feedforward 720.71 (Acceleration), 1 (Speed); Gain sched. OFF

Motion complete settings: Band 0.5 rot; Time 0.001 s

Filters: ☑ #1 SPD LP2 400Hz Q=0.707; □ Filter 2; □ Filter 3; □ Filter 4; □ Filter 5; □ Current controller filter; □ Filter 6; □ Filter 7; □ Filter 8; □ Filter 9; □ Filter 10

Drive operation parameters: Power supply 24 V; Current limit 3 A; Load speed limit 600.044 rad/s

Protections:
☑ Over current: Motor current > 2.7 A for more than 0.01 s
☑ Control error: Position error > 0.6 rot for more than 3 s
☑ Control error: Speed error > 600 rpm for more than 3 s
□ Motor over temperature
☑ I2t over current 2.5 A for 15 s

External chopping resistor: □ Connected; Activate if power supply > 49.999 V

Inputs polarity:

	Limit switch+	Limit switch-
Active high (Connected to +Vlog)	●	●
Active low (Open/No connection)	○	○

Type: ● Sink (PNP) ○ Source(NPN)

□ Software limits - restrict movement: Negative limit -1073741.8 rot; Positive limit 1073741.82 rot

□ Position range limits: Minimum 0 rot; Maximum 0.001 rot

Start mode: Move till aligned with phase A

Figure 5.4 - Drive Setup dialogue

The recommended workflow to configure the drive is:

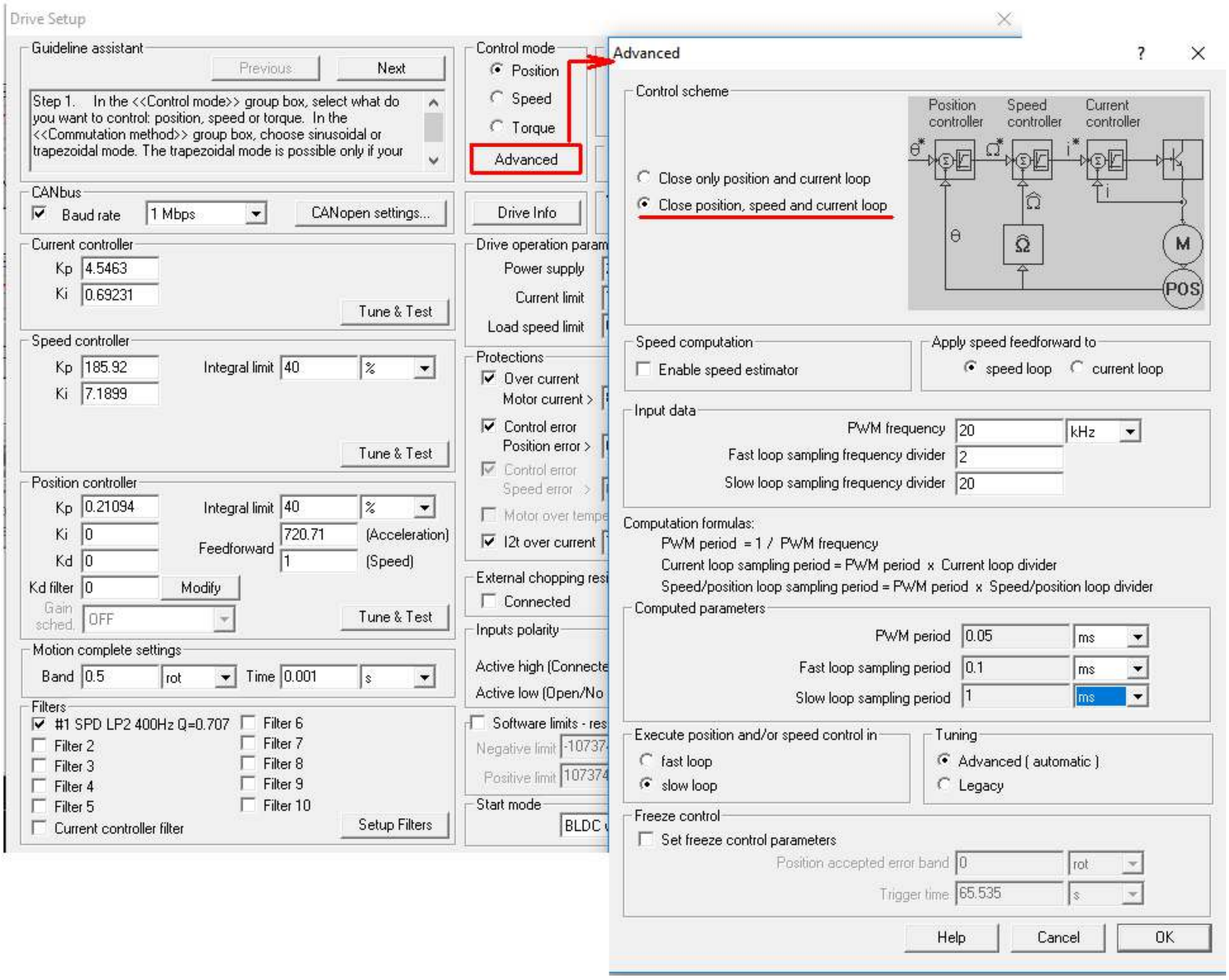
1) Chose the **control mode**
2) Select the **commutation method** for brushless motors equipped with digital Halls;
3) If an **external reference** is used, set its parameters;
4) Set the **drive operation** parameters;
5) **Tune** the active **controllers**;
6) Set the motor **protections**;
7) If a **chopping resistor** is connected to the drive, set the voltage level that activates it;
8) Set the **inputs polarity** and the software limits (if needed);
9) Select the **start mode**;
10) Change the drive **Axis ID** and the **CAN baudrate** (if required).

5.2.1 Control mode selection

The **Control Mode** group box has the following options: **position**, **speed** and **torque**.

- the position mode will activate the current and the position loops, allowing different position profiles.
- the speed mode closes the current and speed loops, allowing speed profiles.
- the torque mode will keep active only the current controller, allowing to control the motor torque.

Some applications require the position and the speed lops to be active at the same time. This control configuration can be enable from the Advanced dialog.



Step 1. In the <<Control mode>> group box, select what do you want to control: position, speed or torque. In the <<Commutation method>> group box, choose sinusoidal or trapezoidal mode. The trapezoidal mode is possible only if your

Control mode: ● Position ○ Speed ○ Torque

CANbus: ☑ Baud rate 1 Mbps

Current controller: Kp 4.5463; Ki 0.69231

Speed controller: Kp 185.92; Ki 7.1899; Integral limit 40 %

Position controller: Kp 0.21094; Ki 0; Kd 0; Kd filter 0; Integral limit 40 %; Feedforward 720.71 (Acceleration), 1 (Speed); Gain sched. OFF

Motion complete settings: Band 0.5 rot; Time 0.001 s

Filters: ☑ #1 SPD LP2 400Hz Q=0.707; ☐ Filter 2; ☐ Filter 3; ☐ Filter 4; ☐ Filter 5; ☐ Current controller filter; ☐ Filter 6; ☐ Filter 7; ☐ Filter 8; ☐ Filter 9; ☐ Filter 10

Drive operation param: Power supply; Current limit; Load speed limit

Protections: ☑ Over current, Motor current >; ☑ Control error, Position error >; ☑ Control error, Speed error >; ☐ Motor over tempe; ☑ I2t over current

External chopping resi: ☐ Connected

Inputs polarity: Active high (Connecte; Active low (Open/No

☐ Software limits - res: Negative limit -10737; Positive limit 107374

Start mode: BLDC

Advanced

Control scheme: ○ Close only position and current loop; ● Close position, speed and current loop

Speed computation: ☐ Enable speed estimator

Apply speed feedforward to: ● speed loop ○ current loop

Input data: PWM frequency 20 kHz; Fast loop sampling frequency divider 2; Slow loop sampling frequency divider 20

Computation formulas:
PWM period = 1 / PWM frequency
Current loop sampling period = PWM period x Current loop divider
Speed/position loop sampling period = PWM period x Speed/position loop divider

Computed parameters: PWM period 0.05 ms; Fast loop sampling period 0.1 ms; Slow loop sampling period 1 ms

Execute position and/or speed control in: ○ fast loop ● slow loop

Tuning: ● Advanced (automatic) ○ Legacy

Freeze control: ☐ Set freeze control parameters; Position accepted error band 0 rot; Trigger time 65.535 s

Figure 5.5 – Control loops configuration in the Advanced window

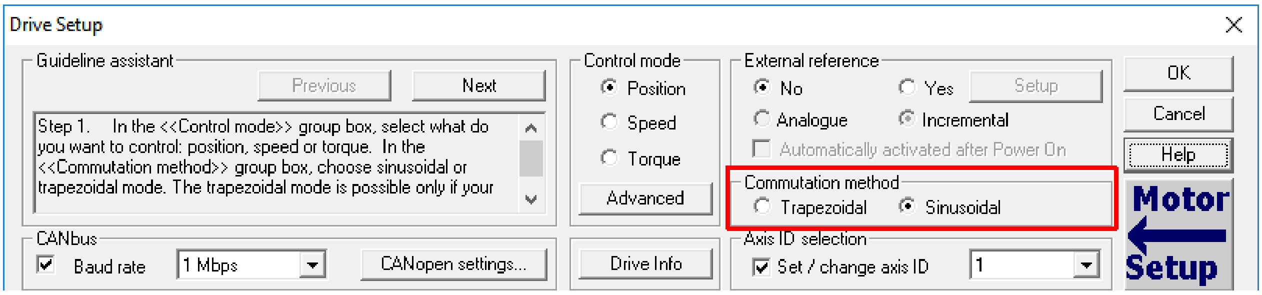
The **Advanced** window allows also to change the PWM frequency and the control loops execution period.

Remarks:

1) For a safe operation is recommended to keep the *PWM frequency* in the 20 kHz … 80 kHz interval;
2) The *fast loop sampling period* default value is 0.1 ms.
3) The *slow loop sampling period* default value is 1 ms

5.2.2 Commutation method options

The **Commutation Method** section allows to choose the strategy used to commutate the power stage transistors.



Drive Setup

Guideline assistant

Step 1. In the <<Control mode>> group box, select what do you want to control: position, speed or torque. In the <<Commutation method>> group box, choose sinusoidal or trapezoidal mode. The trapezoidal mode is possible only if your

Control mode

● Position
○ Speed
○ Torque

External reference

● No ○ Yes
○ Analogue ○ Incremental
□ Automatically activated after Power On

Commutation method

○ Trapezoidal ● Sinusoidal

CANbus

☑ Baud rate 1 Mbps

Axis ID selection

☑ Set / change axis ID 1

Figure 5.6 - Commutation method selection

In the Trapezoidal commutation method the motor is controlled as brushless DC using Hall sensors for commutation. In this mode, the motor currents are rectangular and the back EMF voltages are trapezoidal.

Remark: The trapezoidal mode is possible only if your motor is equipped with digital Hall sensors.

In the **Sinusoidal** commutation method the motor is treated like a permanent magnet synchronous motor and is vector controlled using a field oriented control algorithm. In this mode, the motor works with 3-phase sinusoidal voltages and currents.

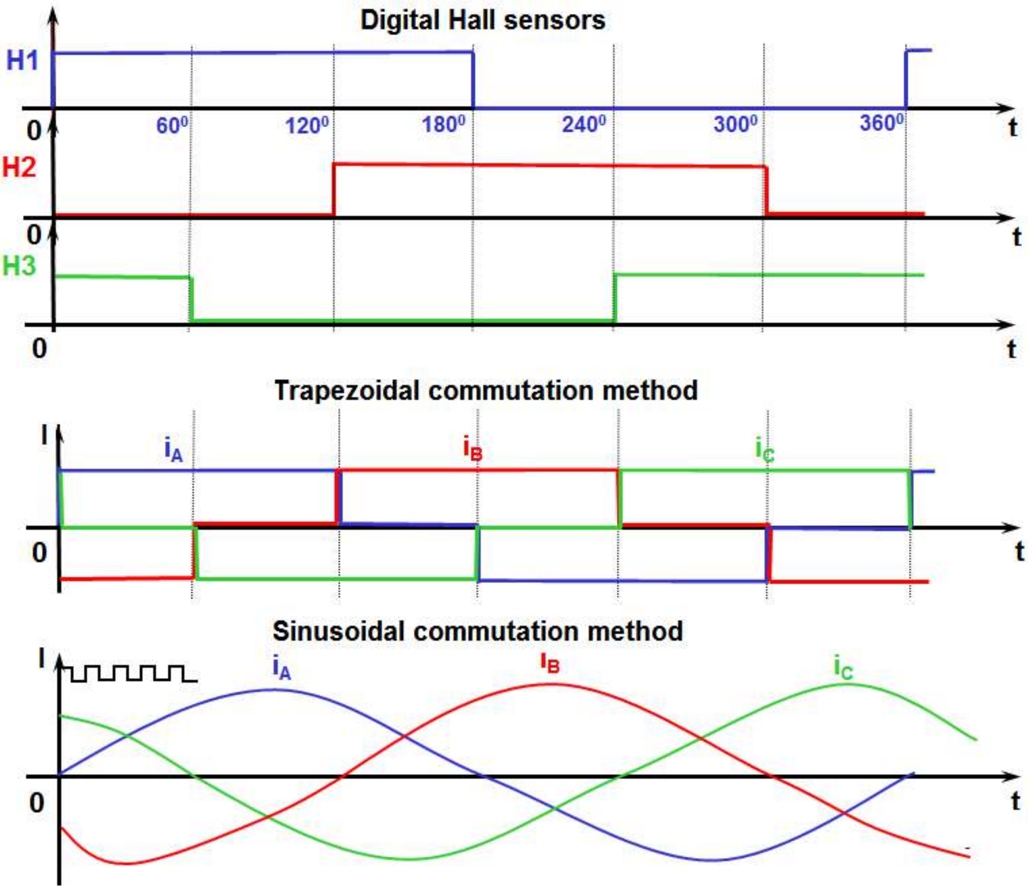
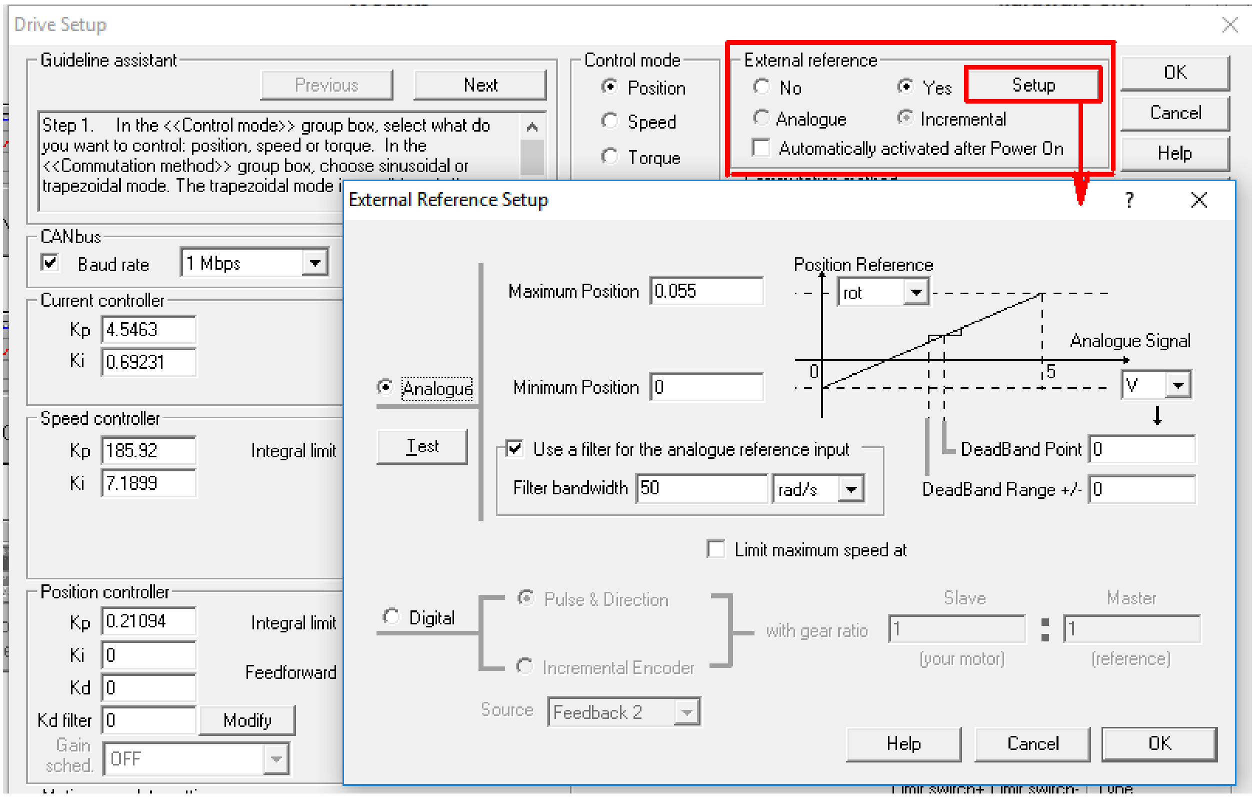


Figure 5.7 - Trapezoidal versus sinusoidal commutation method

The sinusoidal mode offers a smoother operation, especially at low speeds.

5.2.3 External reference

The **External reference** group box allows the parametrization of the external signal (analogue or digital) used to impose the position, the speed or the torque reference,



Drive Setup

Guideline assistant

Step 1. In the <<Control mode>> group box, select what do you want to control: position, speed or torque. In the <<Commutation method>> group box, choose sinusoidal or trapezoidal mode. The trapezoidal mode i

Control mode: ● Position ○ Speed ○ Torque

External reference: ○ No ● Yes ○ Analogue ● Incremental □ Automatically activated after Power On

CANbus: ☑ Baud rate 1 Mbps

Current controller: Kp 4.5463 Ki 0.69231

Speed controller: Kp 185.92 Ki 7.1899

Position controller: Kp 0.21094 Ki 0 Kd 0 Kd filter 0 Gain sched. OFF

External Reference Setup

● Analogue

Maximum Position 0.055

Minimum Position 0

Position Reference rot

Analogue Signal V

☑ Use a filter for the analogue reference input

Filter bandwidth 50 rad/s

DeadBand Point 0

DeadBand Range +/- 0

□ Limit maximum speed at

○ Digital ● Pulse & Direction ○ Incremental Encoder

with gear ratio Slave 1 (your motor) : Master 1 (reference)

Source Feedback 2

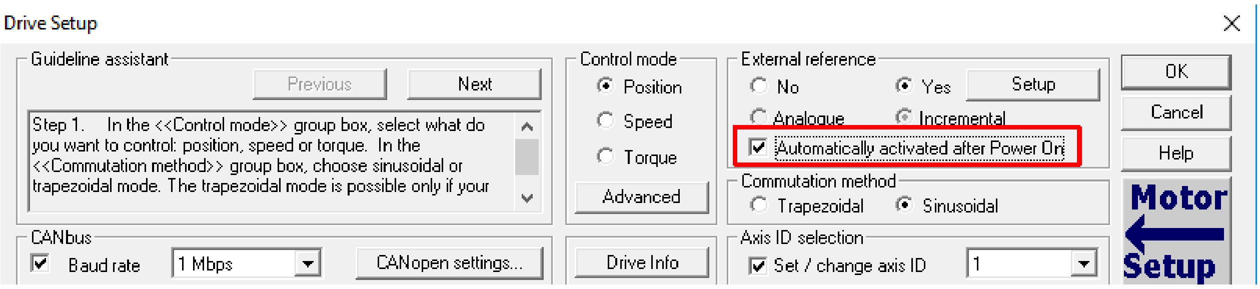
Figure 5.8 - External reference box group & setup dialog

The Analogue reference, uses a 0 V .. 5 V or +/- 10 V voltage signal, to impose the position, the speed or the torque reference, depending on the selected control mode.

The digital reference uses a train of “Pulse & Direction” signals or quadrature A, B signals from an “Incremental encoder”. In both cases, the drive computes the external (master) reference and can follow it with a programmable gear ratio, set through the “Slave” and “Master” parameters.

Remark: In case of the iPOS360x drives, the *Pulse* and *Direction* inputs are shared with the *Encoder A* and *Encoder B* inputs. Hence the digital reference is available only for stepper open loop configurations.

The “**Automatically activated after Power On**” option will be selected if there is no TML program and the drive should enter automatically in the external reference mode, after Powered ON.



Drive Setup

Guideline assistant

Step 1. In the <<Control mode>> group box, select what do you want to control: position, speed or torque. In the <<Commutation method>> group box, choose sinusoidal or trapezoidal mode. The trapezoidal mode is possible only if your

Control mode: ● Position ○ Speed ○ Torque

External reference: ○ No ● Yes ○ Analogue ● Incremental ☑ Automatically activated after Power On

Commutation method: ○ Trapezoidal ● Sinusoidal

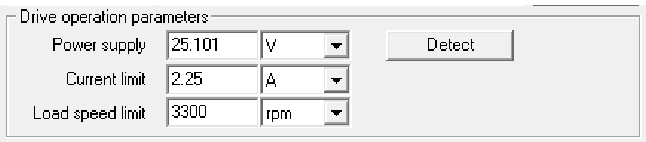
Axis ID selection: ☑ Set / change axis ID 1

CANbus: ☑ Baud rate 1 Mbps

Figure 5.9 - Automatically activated after Power On option

5.2.4 Drive operation parameters

The **Drive operation parameters** section allows to set the power supply voltage, the current and load speed limits.



Drive operation parameters

Power supply	25.101	V
Current limit	2.25	A
Load speed limit	3300	rpm

Figure 5.10 - Drive operation parameters group box

The **Power supply** parameter represents the voltage applied to the motor supply circuit. If this value is unknown, it can be detected automatically by pressing the **Detect** button.

The **Current limit** represents the maximum current that the drive will apply to the motor. This value should be set higher than the motor nominal current, to allow dynamic movements, but smaller than the motor peak current because it may be applied to the motor for a long time.

The **Load speed limit** allows to limit the load speed, with the purpose to protect the mechanical system and the operators in case of a bad command that may lead to a dangerous speed.

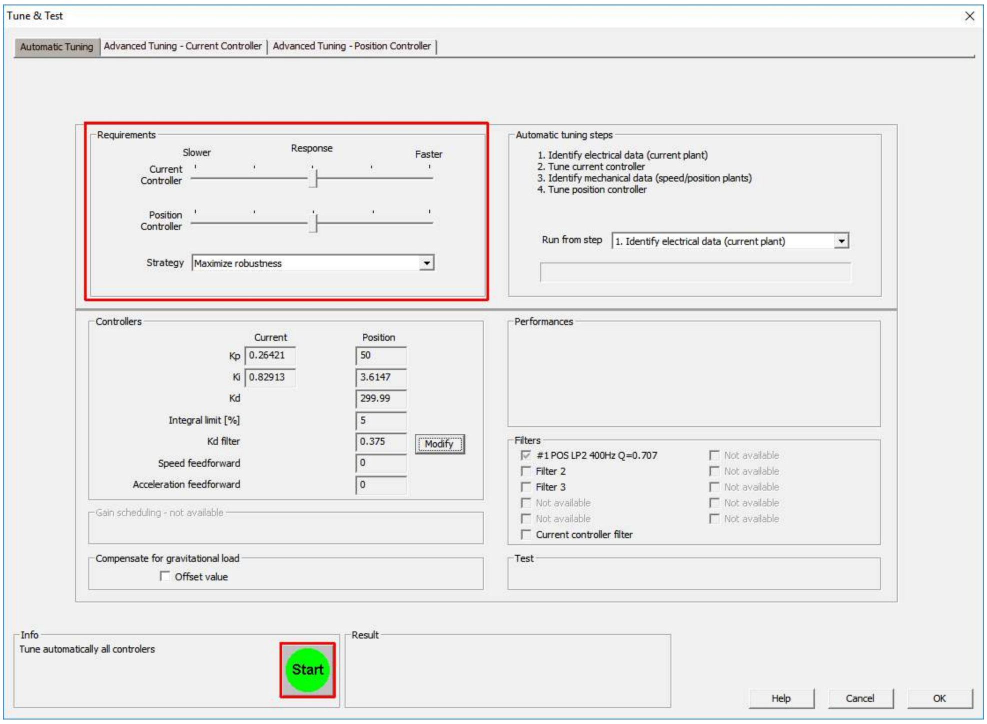
5.2.5 Controllers tuning

The **controllers tuning** is one of the most important parts from the Drive Setup dialogue and should be performed in the following order:

- current controller tuning
- speed controller tuning (if it is active)
- position controller tuning (if it is active)

The automatic tuning procedure performs all needed steps in the needed order no-matter which controller is accessed first. The above recommendation applies only when the user prefers to perform the tuning manually.

The automatic **Tune & Test** window allows to select the tuning strategy and start the automatic tuning procedure.



Tune & Test

Requirements

Strategy: Maximize robustness

Automatic tuning steps

1. Identify electrical data (current plant)
2. Tune current controller
3. Identify mechanical data (speed/position plants)
4. Tune position controller

Run from step: 1. Identify electrical data (current plant)

Controllers

	Current	Position
Kp	0.26421	50
Ki	0.82913	3.6147
Kd		299.99
Integral limit [%]		5
Kd filter		0.375
Speed feedforward		0
Acceleration feedforward		0

Filters

☑ #1 POS LP2 400Hz Q=0.707
☐ Filter 2
☐ Filter 3
☐ Not available
☐ Not available
☐ Current controller filter
☐ Not available
☐ Not available
☐ Not available
☐ Not available
☐ Not available

Compensate for gravitational load

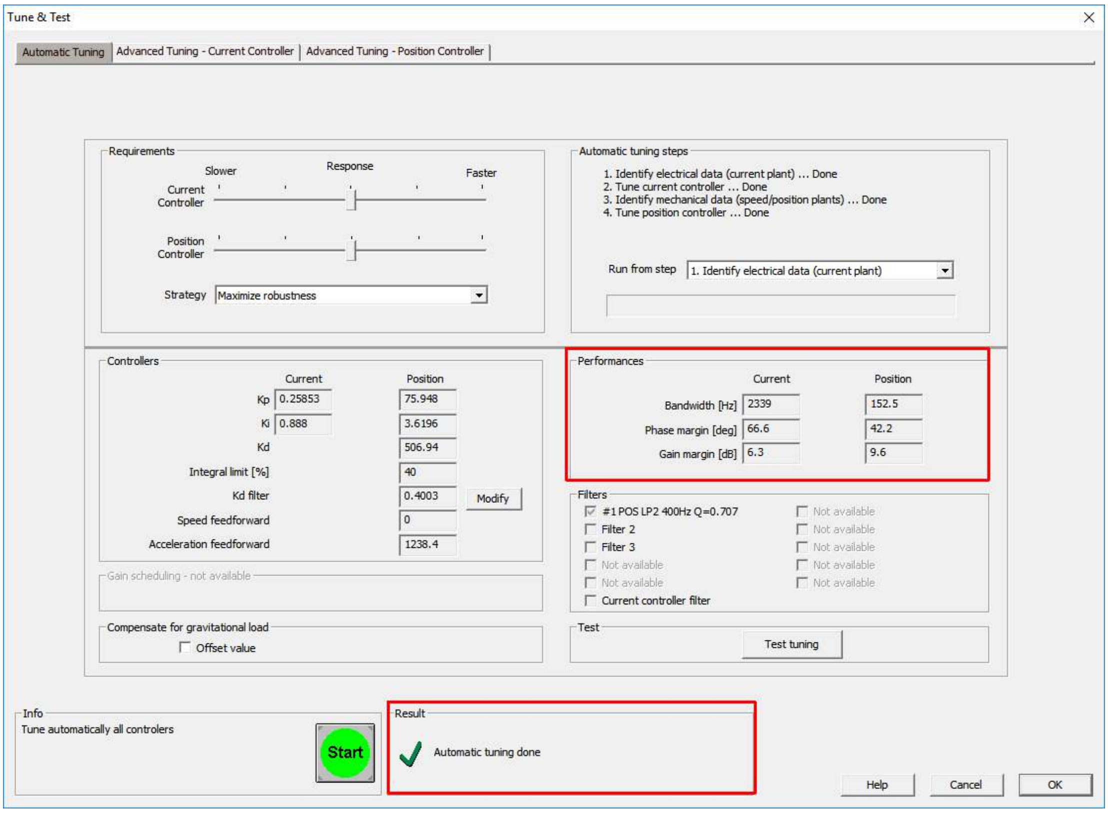
☐ Offset value

Info

Tune automatically all controlers

Figure 5.11 - Automatic Tune & Test dialogue

When the tuning procedure succeeds, the **Tune & Tes**t window will display an OK message in the **Result** box and the controllers bandwidth, phase margin and gain margin in the **Performance** box.



Strategy: Maximize robustness

Automatic tuning steps:
1. Identify electrical data (current plant) ... Done
2. Tune current controller ... Done
3. Identify mechanical data (speed/position plants) ... Done
4. Tune position controller ... Done

Run from step: 1. Identify electrical data (current plant)

	Current	Position
Kp	0.25853	75.948
Ki	0.888	3.6196
Kd		506.94
Integral limit [%]		40
Kd filter		0.4003
Speed feedforward		0
Acceleration feedforward		1238.4

	Current	Position
Bandwidth [Hz]	2339	152.5
Phase margin [deg]	66.6	42.2
Gain margin [dB]	6.3	9.6

Filters:
- ☑ #1 POS LP2 400Hz Q=0.707
- □ Filter 2
- □ Filter 3
- □ Not available
- □ Not available
- □ Current controller filter
- □ Not available
- □ Not available
- □ Not available
- □ Not available
- □ Not available

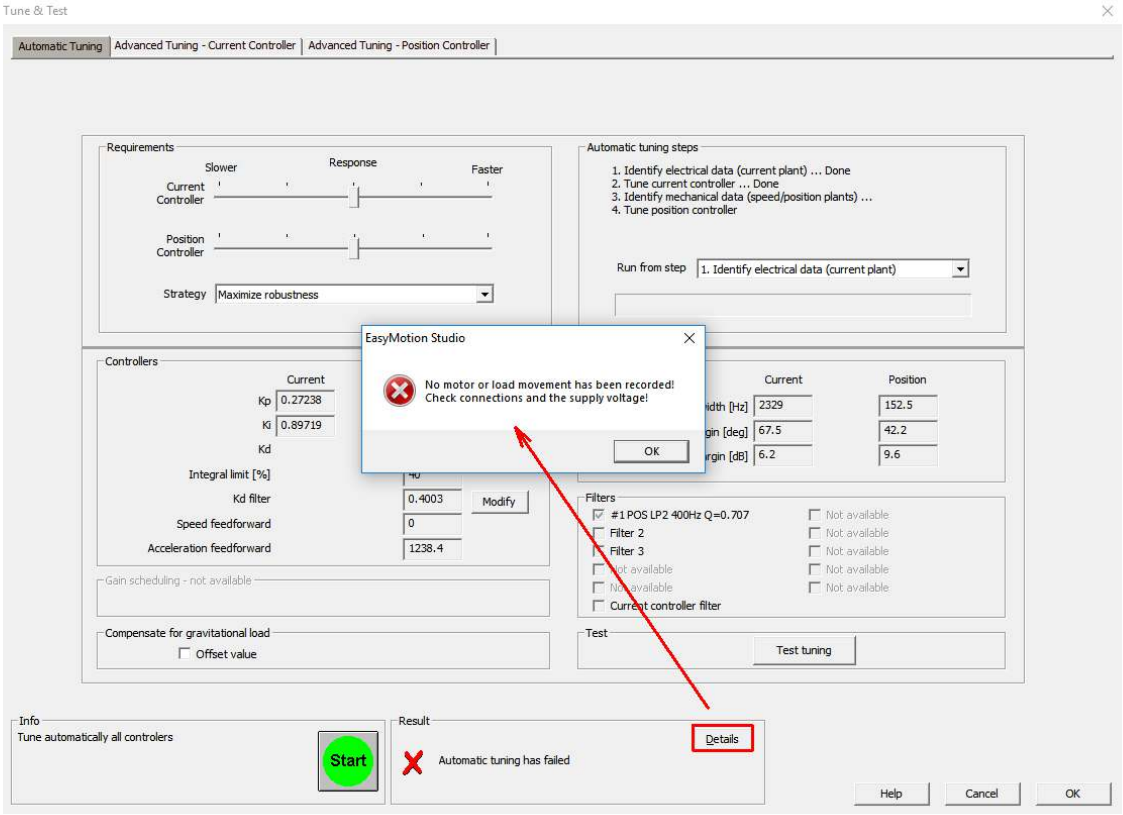
□ Offset value

Tune automatically all controlers

✓ Automatic tuning done

Figure 5.12 - Tune & Test dialogue when the auto-tuning procedure succeeds

If for some reasons the automatically tuning procedure fails, the **Details** button will appear in the **Results** box from the **Tune & Test** window, allowing the customer to check what went wrong.



Strategy: Maximize robustness

Automatic tuning steps:
1. Identify electrical data (current plant) ... Done
2. Tune current controller ... Done
3. Identify mechanical data (speed/position plants) ...
4. Tune position controller

Run from step: 1. Identify electrical data (current plant)

No motor or load movement has been recorded!
Check connections and the supply voltage!

	Current	Position
Kp	0.27238	
Ki	0.89719	
Kd		
Integral limit [%]		40
Kd filter		0.4003
Speed feedforward		0
Acceleration feedforward		1238.4

	Current	Position
...idth [Hz]	2329	152.5
...gin [deg]	67.5	42.2
...rgin [dB]	6.2	9.6

Filters:
- ☑ #1 POS LP2 400Hz Q=0.707
- □ Filter 2
- □ Filter 3
- □ Not available
- □ Not available
- □ Current controller filter
- □ Not available
- □ Not available
- □ Not available
- □ Not available
- □ Not available

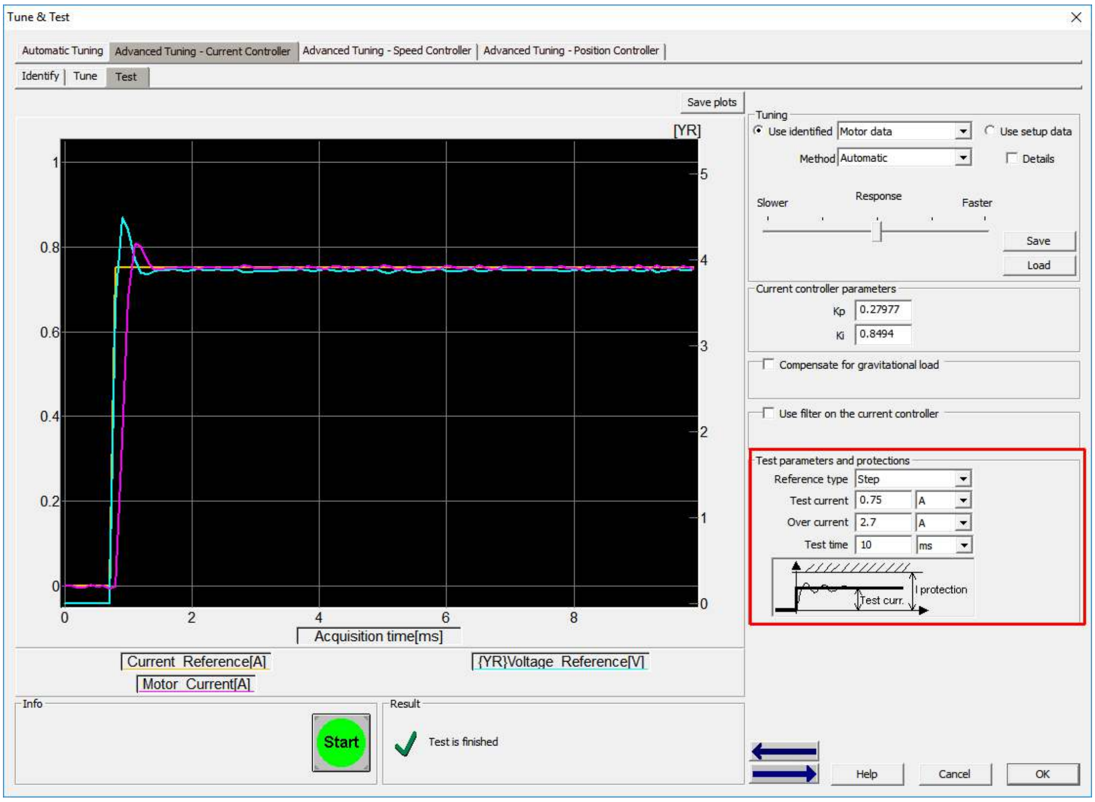
□ Offset value

Tune automatically all controlers

✗ Automatic tuning has failed

Figure 5.13 - Auto-tuning procedure failing example

The **Tune & Test** window allows also to test the controllers performances and adjust (if needed) the tuning parameters. For this, the user needs to select the **Advanced Tuning** tab for the desired controller and go to the **Test** tab, to set the test reference, run it and adjust the parameters (if the case).



Tuning

● Use identified: Motor data ○ Use setup data

Method: Automatic □ Details

Current controller parameters

Kp: 0.27977

Ki: 0.8494

□ Compensate for gravitational load

□ Use filter on the current controller

Test parameters and protections

Reference type: Step

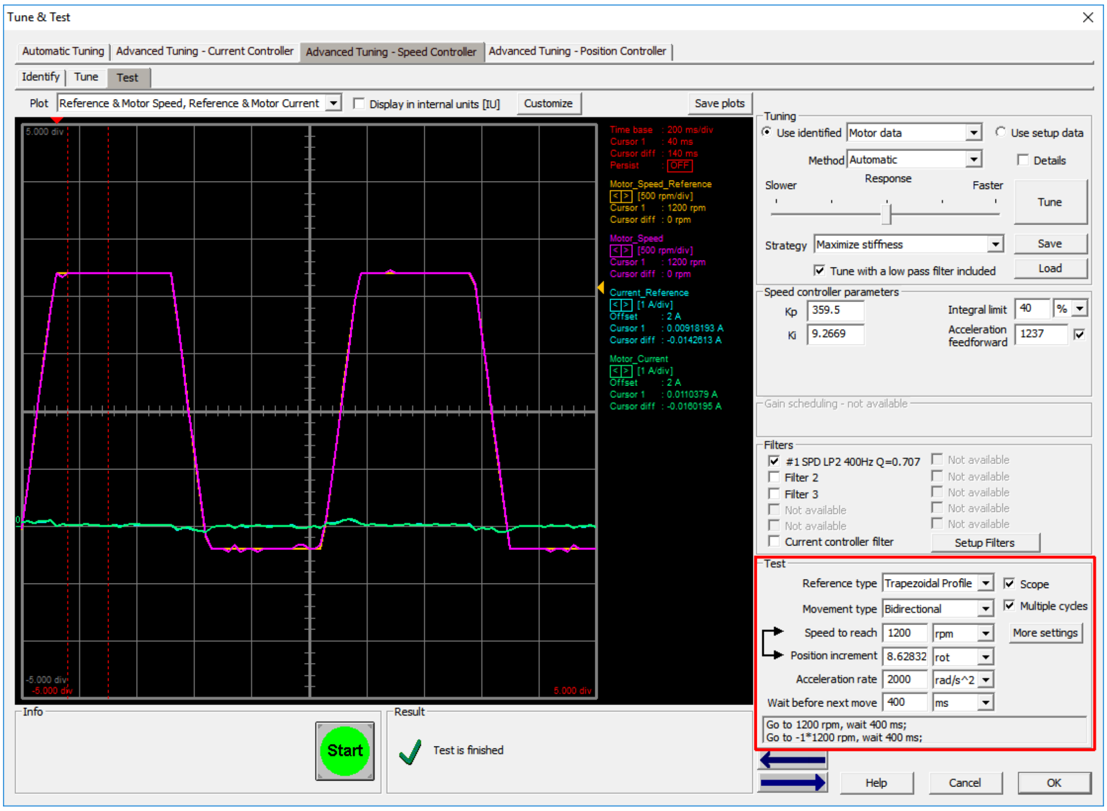
Test current: 0.75 A

Over current: 2.7 A

Test time: 10 ms

✓ Test is finished

Figure 5.14 - Test tab in the Current Controller



Plot: Reference & Motor Speed, Reference & Motor Current □ Display in internal units [IU]

Time base : 200 ms/div
Cursor 1 : 40 ms
Cursor diff : 140 ms
Persist : OFF

Motor_Speed_Reference
[500 rpm/div]
Cursor 1 : 1200 rpm
Cursor diff : 0 rpm

Motor_Speed
[500 rpm/div]
Cursor 1 : 1200 rpm
Cursor diff : 0 rpm

Current_Reference
[1 A/div]
Offset : 2 A
Cursor 1 : 0.00918193 A
Cursor diff : -0.0142613 A

Motor_Current
[1 A/div]
Offset : 2 A
Cursor 1 : 0.0110379 A
Cursor diff : -0.0160195 A

Tuning

● Use identified: Motor data ○ Use setup data

Method: Automatic □ Details

Strategy: Maximize stiffness

☑ Tune with a low pass filter included

Speed controller parameters

Kp: 359.5 Integral limit: 40 %

Ki: 9.2669 Acceleration feedforward: 1237 ☑

Gain scheduling - not available

Filters

☑ #1 SPD LP2 400Hz Q=0.707 □ Not available
□ Filter 2 □ Not available
□ Filter 3 □ Not available
□ Not available □ Not available
□ Not available □ Not available
□ Current controller filter

Test

Reference type: Trapezoidal Profile ☑ Scope

Movement type: Bidirectional ☑ Multiple cycles

Speed to reach: 1200 rpm

Position increment: 8.62832 rot

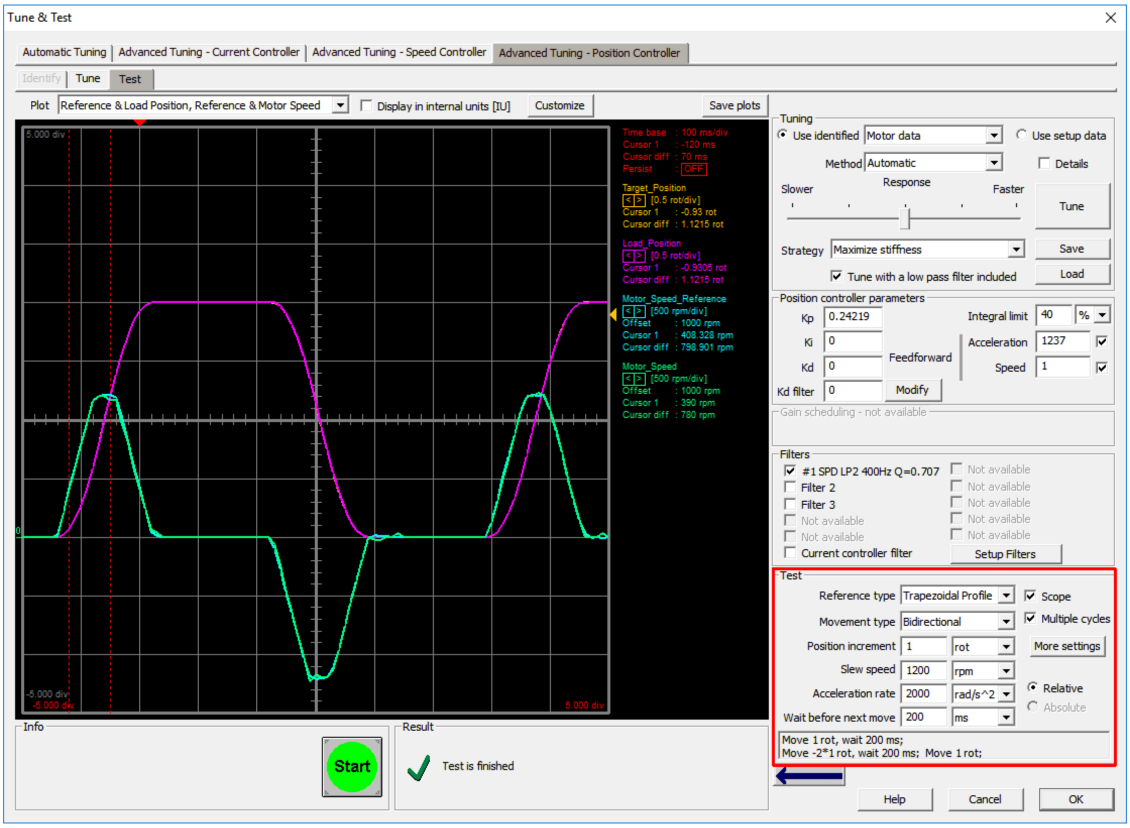
Acceleration rate: 2000 rad/s^2

Wait before next move: 400 ms

Go to 1200 rpm, wait 400 ms;
Go to -1*1200 rpm, wait 400 ms;

✓ Test is finished

Figure 5.15 - Test tab in the Speed Controller



Plot: Reference & Load Position, Reference & Motor Speed
□ Display in internal units [IU]

Time base : 100 ms/div
Cursor 1 : -120 ms
Cursor diff : 70 ms
Persist : OFF

Target_Position
[0.5 rot/div]
Cursor 1 : -0.93 rot
Cursor diff : 1.1215 rot

Load_Position
[0.5 rot/div]
Cursor 1 : -0.9305 rot
Cursor diff : 1.1215 rot

Motor_Speed_Reference
[500 rpm/div]
Offset : 1000 rpm
Cursor 1 : 408.328 rpm
Cursor diff : 798.901 rpm

Motor_Speed
[500 rpm/div]
Offset : 1000 rpm
Cursor 1 : 390 rpm
Cursor diff : 780 rpm

Tuning
● Use identified: Motor data ○ Use setup data
Method: Automatic □ Details
Response: Slower – Faster
Strategy: Maximize stiffness
☑ Tune with a low pass filter included

Position controller parameters
Kp: 0.24219
Ki: 0
Kd: 0
Kd filter: 0
Feedforward
Integral limit: 40 %
Acceleration: 1237 ☑
Speed: 1 ☑

Gain scheduling - not available

Filters
☑ #1 SPD LP2 400Hz Q=0.707
□ Filter 2
□ Filter 3
□ Not available
□ Not available
□ Current controller filter
□ Not available
□ Not available
□ Not available
□ Not available
□ Not available

Test
Reference type: Trapezoidal Profile ☑ Scope
Movement type: Bidirectional ☑ Multiple cycles
Position increment: 1 rot
Slew speed: 1200 rpm
Acceleration rate: 2000 rad/s^2 ● Relative
Wait before next move: 200 ms ○ Absolute

Move 1 rot, wait 200 ms;
Move -2*1 rot, wait 200 ms; Move 1 rot;

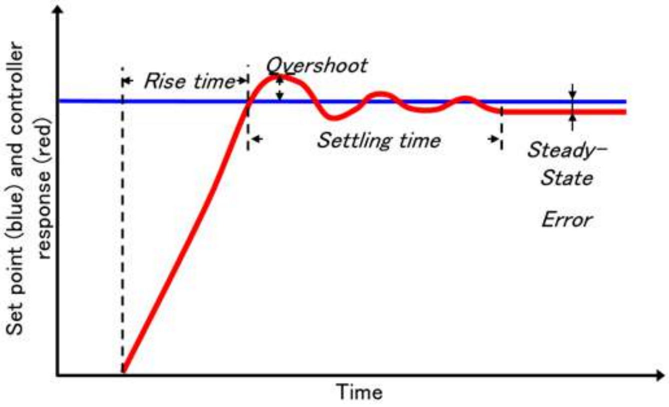
Info

Result
✓ Test is finished

Figure 5.16 - Test tab in the Position Controller

When referring to the response characteristics like overshoot, rise time and steady-state error, the controllers coefficients have the following effects:

- ✓ increasing Kp will result in increased overshoot but with reduced rise time and steady-state error;
- ✓ increasing Ki will result in increased overshoot and slightly increased rise time but will drastically reduce the steady-state error;
- ✓ increasing Kd will result in increased rise time but with reduced overshoot (the steady-state error is not affected by Kd).

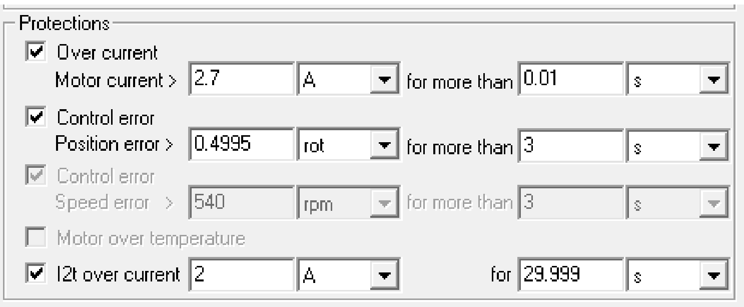


	KP	KI	KD
Rise time	*decrease*	*decrease*	*increase*
Overshoot	*increase*	*increase*	*decrease*
Settling time	*small change*	*increase*	*decrease*
Steady-State Error	*decrease*	*eliminate*	*no effect*

Figure 5.17 - Controllers parameters effect on the overshoot, rise time and steady-state error

5.2.6 Motor protections

The motor protections are adjusted automatically by the auto tuning procedure but if the case, they can also be modified through the **Protections** group box.



Protections

☑ Over current

Motor current > 2.7 A for more than 0.01 s

☑ Control error

Position error > 0.4995 rot for more than 3 s

☑ Control error

Speed error > 540 rpm for more than 3 s

□ Motor over temperature

☑ I2t over current 2 A for 29.999 s

Figure 5.18 - Motor protections

The **over current** protection was designed to protect the motor during the setup phase, when the current controller is not yet tuned. It is defined through a current value and a time interval and triggers if the motor current remains above the over current limit a time interval bigger than the over current time.

The current value should be set higher than the **current limit** parameter but smaller than the motor peak current. The over current time must be set according with the motor data (smaller than the time the motor can support its peak current but long enough, to prevent the current oscillates over and under the protection limit).

The **control error** protection, monitors the position error or the speed error, function on which control mode was selected and triggers when the position/speed error remains above the position/speed error limit a time interval bigger than the position error time.

Note that if the absolute value of the position / speed error goes below the position / speed error limit, the position / speed error time counter is reset.

The **I2t** thermal protection prevents the motor damage due to long time over current operation. Its value is set by selecting a point on the motor I2t curve (should be available in the motor data sheet) and input the I2t over current value and the I2t time in the associated edit boxes. These points together with the nominal motor current “In” define the surface “SI2t” (see figure 5.19 - left).

When the current through the motor exceeds the nominal current (“In”), the drives starts to calculate, at each current loop (fast loop), the difference between the square of the instantaneous current and the square of the nominal current. The value obtained at the current step is added to the value obtained at the previous step, and so on. When the sum is equal or greater than the “SI2t” surface from the first picture, the drive I2t protection is triggered (see figure 5.19 - right).

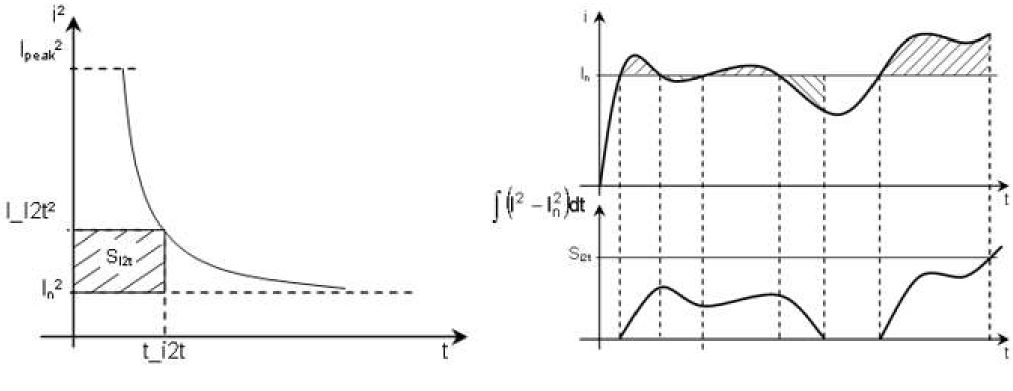


Figure 5.19 - I2t surface and triggering

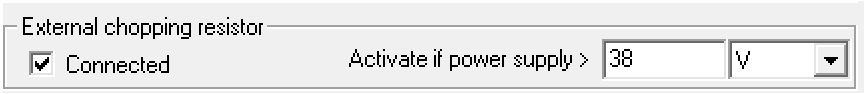
Once triggered, the default behaviour of the I2t protection is to limit the motor current to 90% from the nominal current set in the Motor Setup dialogue. The current reverts back to the normal current limit after the internal protection value drops to 0.

5.2.7 External chopping resistor

When the motor brakes or reverses, it generates energy, causing the bus voltage to increase increase. Depending on the amount of generated energy and how fast it is increasing, the bus voltage can exceed the overvoltage protection and it can even damage the drive or the power supply.

To prevent this unwanted situation, the over-voltage can be limited with a chopping resistor (see the drive user manual for the chopping resistor selection and connection).

The chopping resistor function can be activated by selecting the **Connected** option and setting the voltage threshold (if the supply voltage is higher than the threshold then the chopping circuitry will be active all the time, chopping the supply).



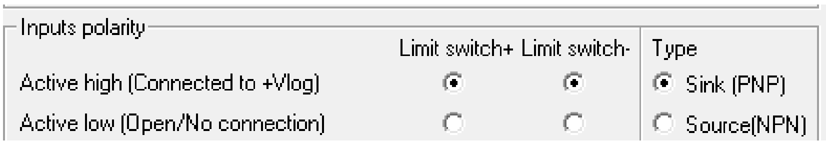
External chopping resistor
☑ Connected Activate if power supply > 38 V

Figure 5.20 - Enable the chopping resistor function and set its voltage threshold

Typically the activation level for the braking circuit must be set slightly above the drive supply voltage value and below its over-voltage protection value.

5.2.8 Inputs polarity and type

The **Inputs polarity** group box can be used to select the active level for the **Enable, Positive Limit Switch** and **Negative Limit Switch** digital inputs.



Inputs polarity

	Limit switch+	Limit switch-	Type
Active high (Connected to +Vlog)	●	●	● Sink (PNP)
Active low (Open/No connection)	○	○	○ Source(NPN)

Figure 5.21 - Inputs polarity and type

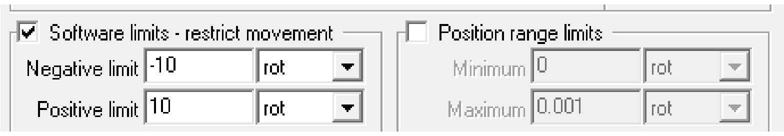
The **Inputs Type** group allows to set the inputs type:

- sink (PNP) – internally pulled-up to +Vlog
- source (NPN) – internally pulled-down to GND

Remark: The configuration options for the inputs depend on the hardware.

5.2.9 Software limit switches and position range limits

The **software limit switches** act like the hardware limit switches. When the a limit is reached then the quick stop profile is automatically executed to stop the motor movement. No movements will be allowed in the respective direction.



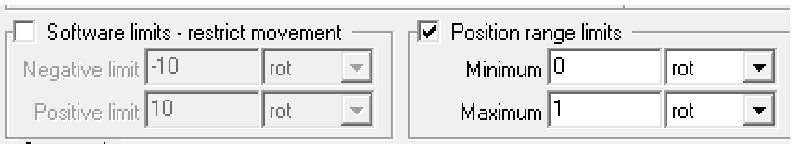
☑ Software limits - restrict movement
Negative limit -10 rot
Positive limit 10 rot

□ Position range limits
Minimum 0 rot
Maximum 0.001 rot

Figure 5.22 - Software limit switches

The **position range limits** group box allows to:

- keep the position indicator between 2 limits (default operation mode)
- get to the target position using only negative or only positive movements
- get to the target using the shortest path



□ Software limits - restrict movement
Negative limit -10 rot
Positive limit 10 rot

☑ Position range limits
Minimum 0 rot
Maximum 1 rot

Figure 5.23 - Position range limits

The position range limits behaviour can be set through the **bits 6** and **7** in the **POSOPTCODE** register, as is described below.

Register name	Bit 7	Bit 6	Description
POSOPTCODE	0	0	Normal positioning (default option): if the position indicator reaches or exceeds one of the position range limits, it wrap automatically to the other end of the range.
	0	1	Positioning only in negative direction: if the target position is higher than the actual position, the axis moves over the min position limit to the target position.
	1	0	Positioning only in positive direction: if the target position is lower than the actual position, the axis moves over the max position limit to the target position.
	1	1	Positioning with the shortest way to the target position. NOTE: If the difference between actual value and target position in a 360° system is 180°, the axis moves in positive direction.

The figure below shows few movement examples (function on the settings of the bits 6 and 7 in the POSOPTCODE register), when the minimum position range limit is set to 0° and the maximum position range limit is set to 360°.

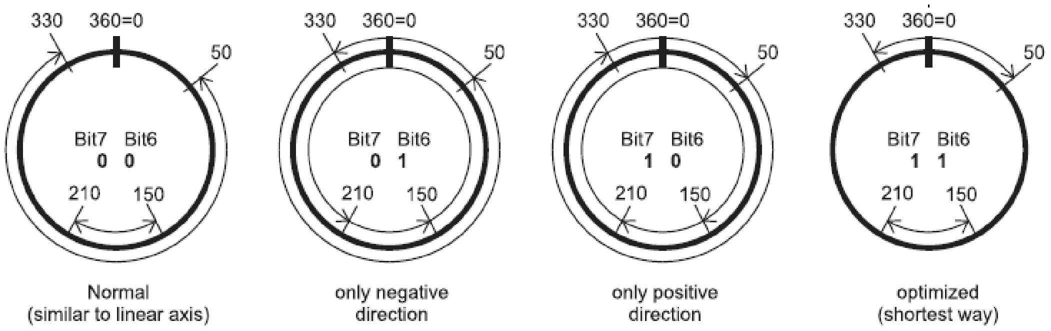


Figure 5.24 - Rotary axis positioning examples

5.2.10 Start mode selection

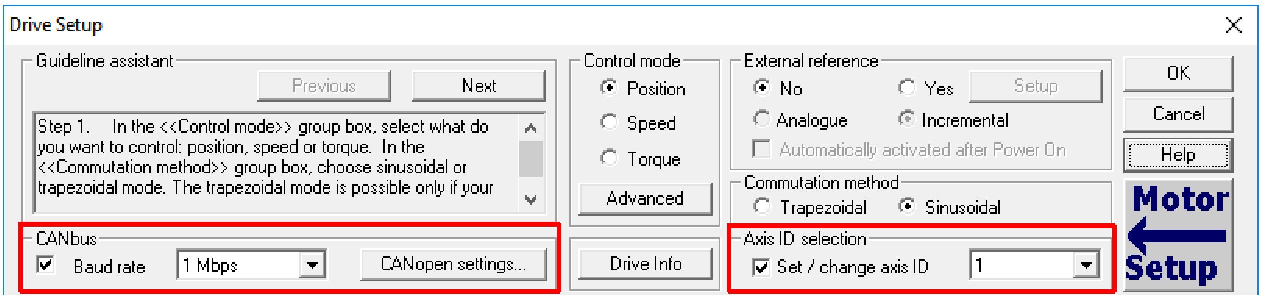
When the motor is controlled using the field oriented control, the position of the rotor must be precisely known. The following start methods are available for incremental encoder configurations:

Move till aligned with phase A
- the motor moves until it is aligned first with phase B, then with phase A.
- this start method can be used for all applications with horizontal loads where the motor can move freely.
BLDC with Hall sensors (available only if the motor is equipped with digital Hall sensors)
- the motor starts in the trapezoidal mode and switches to the sinusoidal mode after the first Hall sensors transition is detected.
- this start mode is recommended for applications with vertical loads, requiring a starting torque close the motor nominal value.
PMSM with Hall sensors (available only if the motor is equipped with digital Hall sensors)
- the motor starts directly in sinusoidal mode and adjusts the rotor position after the first transition of the Hall sensors is detected.
- this start mode that can be used in most of the applications.
Motionless start (available only if there’s an encoder feedback)
- the drive determines the rotor position. The method involves only a very small movement around the initial rotor position and leads to a faster starting time.
- this start mode is recommended when the Hall sensors are not available and the motor/load cannot move freely.

When an absolute position sensor on the motor is present then the default start method is set to **Direct using absolute position sensor** (the motor absolute position is known so the motor control starts directly).

5.2.11 Axis ID selection and CAN bus baudrate

All Technosoft drives are identified with an Axis ID (a number between 1 and 127 in CANopen or between 1 and 255 in TMLCAN) that can be set by hardware (check the drive user manual) or by software, through the **Axis ID selection** group box.



Drive Setup

Guideline assistant

Step 1. In the <<Control mode>> group box, select what do you want to control: position, speed or torque. In the <<Commutation method>> group box, choose sinusoidal or trapezoidal mode. The trapezoidal mode is possible only if your

Control mode: ● Position ○ Speed ○ Torque

External reference: ● No ○ Yes ○ Analogue ● Incremental □ Automatically activated after Power On

Commutation method: ○ Trapezoidal ● Sinusoidal

CANbus: ☑ Baud rate 1 Mbps

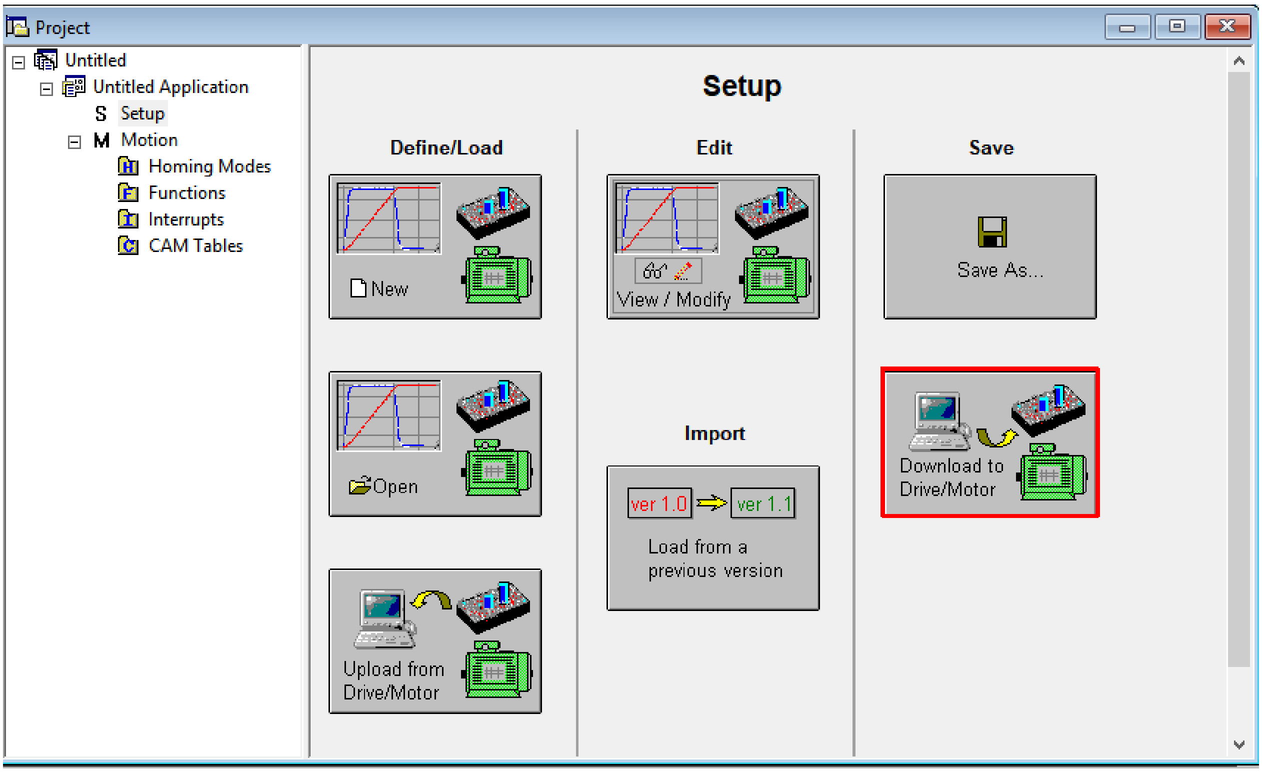
Axis ID selection: ☑ Set / change axis ID 1

Figure 5.25 - Axis ID and CAN bus baudrate selection

The **CANbus** group box allow to set the baudrate for the CAN communication. The default value is **F/W default** (firmware default), which means 500 kbps, but if the case then the following speeds are available: 125 kbps, 250 kbps, 500 kbps and 1 Mbps

5.3 Download the Setup to the drive

After the drive and motor parameters are set, the setup can be downloaded to the drive memory, using the **Download to Drive/Motor** button in the **Setup** section.



Project

- Untitled
  - Untitled Application
    - S Setup
    - M Motion
      - Homing Modes
      - Functions
      - Interrupts
      - CAM Tables

Figure 5.26 - Download the drive and motor setup parameters to the drive memory

Remark: When the Download to Drive/motor button is pressed the configuration parameters are transfered to the non-volatile memory of the drive (EEPROM). The parameters become active after a power cycle (logic power turned off/on) or when the drive is reset.

6 Drive programming

The Technosoft drives can work as standalone devices or as slaves in a CAN (TMLCAN or TechnoCAN), CANopen or EtherCAT network.

The CANopen and EtherCAT protocols are standard. In this cases the Technosoft drives just execute the commands sent by a CANopen/EtherCAT master, if the case, report the errors / emergency situations.

When the drive is used as a standalone device or when the application functionality is distributed between the master and the Technosoft drives, a TML program is required.

The TML program can be developed using the motion programming dialogs available when the **Motion** branch is selected and stored as a main program or as functions. A set of predefined homing methods and interrupts are also available.

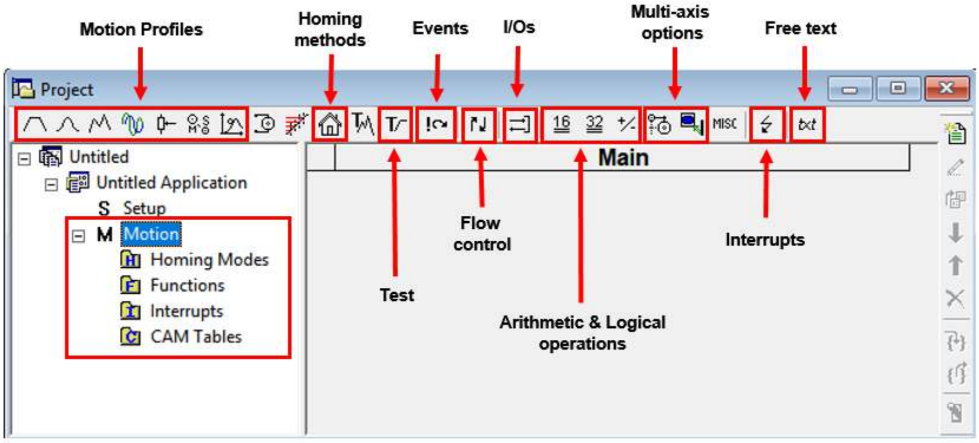
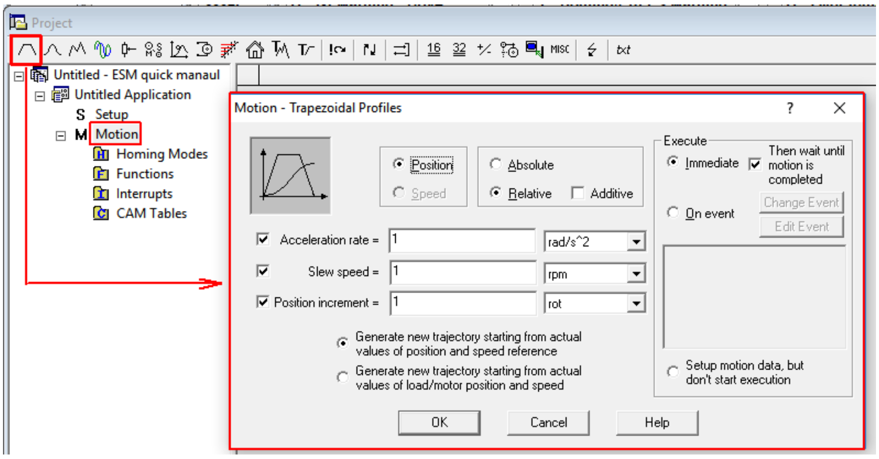


Figure 6.1 - Motion wizard options

The programming with EasyMotion Studio is graphic and can be done through a set of motion wizard dialogues that are opened when the user clicks on the buttons in the Motion branch menu bar.



Motion - Trapezoidal Profiles

● Position ○ Speed

○ Absolute ● Relative □ Additive

Execute: ● Immediate ☑ Then wait until motion is completed; ○ On event (Change Event, Edit Event)

☑ Acceleration rate = 1 rad/s^2

☑ Slew speed = 1 rpm

☑ Position increment = 1 rot

● Generate new trajectory starting from actual values of position and speed reference

○ Generate new trajectory starting from actual values of load/motor position and speed

○ Setup motion data, but don't start execution

OK | Cancel | Help

Figure 6.2 - Example of motion wizard dialogue

The options in the Motion branch are detailed in the **Help** topics***: Help | Help Topics | Application Programming | Motion Programming with built-in Motion Controller***.

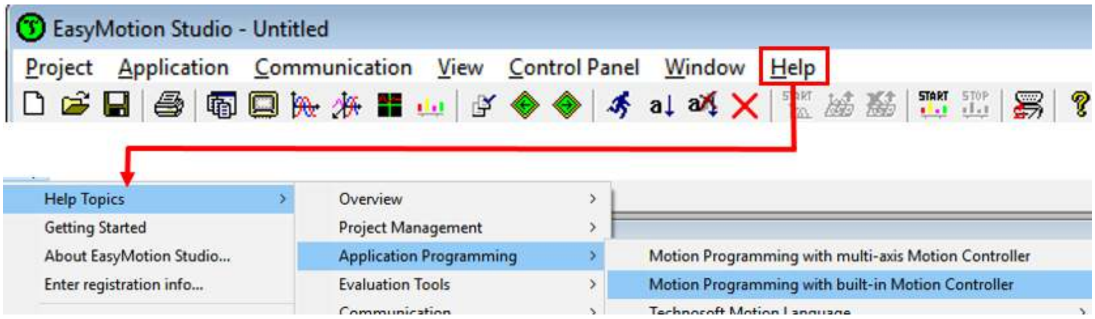

EasyMotion Studio - Untitled
Project
Application
Communication
View
Control Panel
Window
Help
START
STOP
Help Topics
Getting Started
About EasyMotion Studio...
Enter registration info...
Overview
Project Management
Application Programming
Evaluation Tools
Communication
Motion Programming with multi-axis Motion Controller
Motion Programming with built-in Motion Controller
Technosoft Motion Language

Figure 6.3 - Motion wizards help topic

7 Programming example with EasyMotion Studio

For demonstration purpose the drive will be programmed to turn the motor/load 1 rotation, using a trapezoidal position profile, wait 0.5 s and then, rotate -1 rotation, using an S-Curve profile.

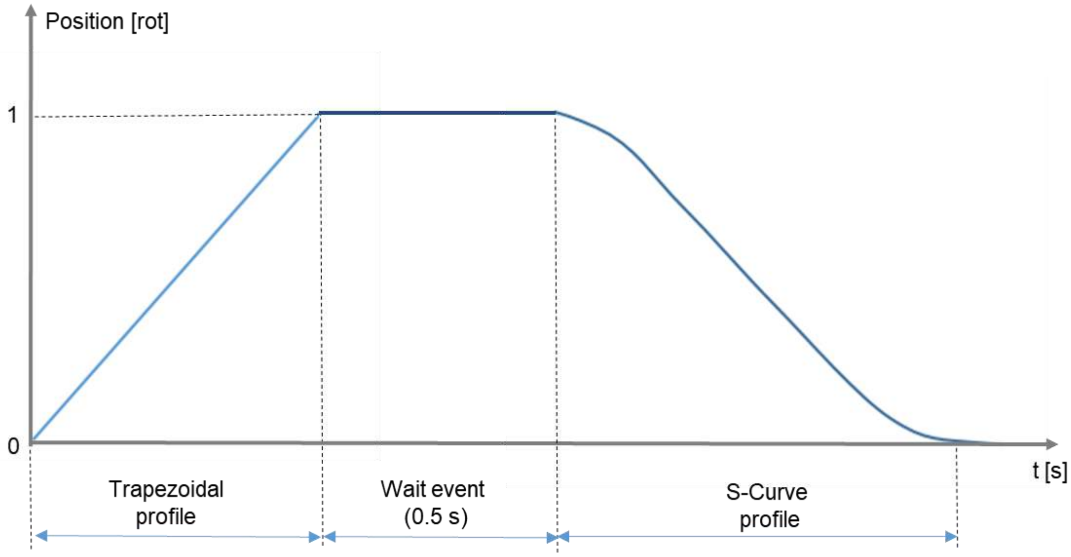
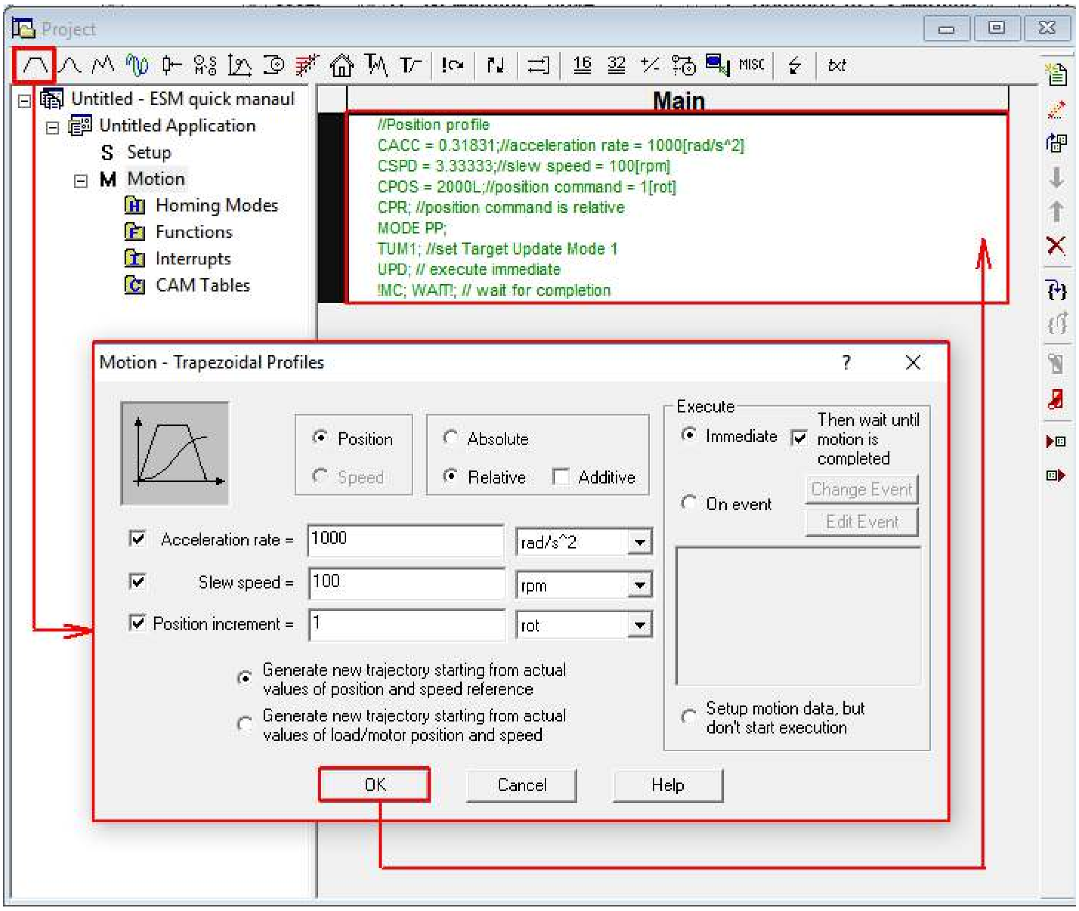


Figure 7.1 - Required movement

The trapezoidal motion profile can be generated through the **Motion – Trapezoidal Profile** motion wizard.



```
//Position profile
CACC = 0.31831;//acceleration rate = 1000[rad/s^2]
CSPD = 3.33333;//slew speed = 100[rpm]
CPOS = 2000L;//position command = 1[rot]
CPR; //position command is relative
MODE PP;
TUM1; //set Target Update Mode 1
UPD; // execute immediate
!MC; WAIT!; // wait for completion
```

● Position ○ Speed

○ Absolute ● Relative □ Additive

Execute: ● Immediate ☑ Then wait until motion is completed; ○ On event; Change Event; Edit Event; ○ Setup motion data, but don't start execution

☑ Acceleration rate = 1000 rad/s^2

☑ Slew speed = 100 rpm

☑ Position increment = 1 rot

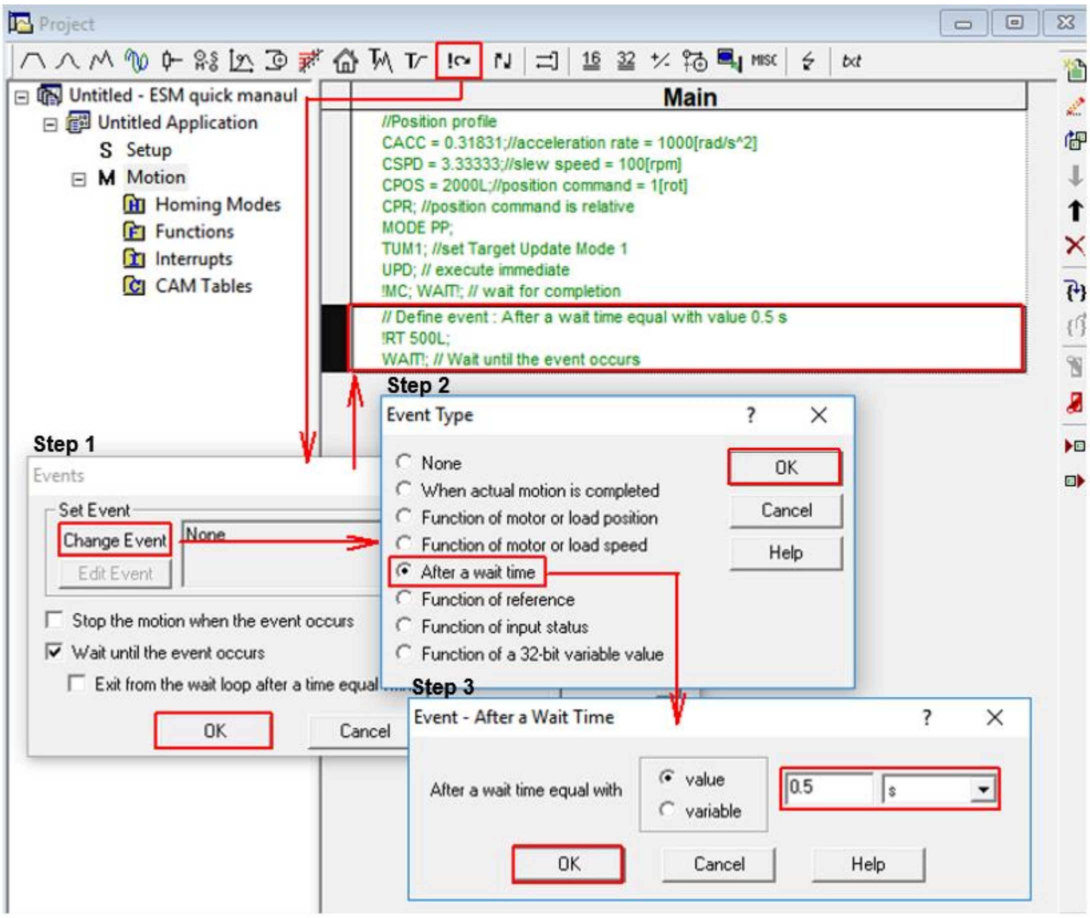
● Generate new trajectory starting from actual values of position and speed reference

○ Generate new trajectory starting from actual values of load/motor position and speed

OK | Cancel | Help

Figure 7.2 - Insert a trapezoidal position profile

The 0.5 s wait condition can be generated using the **Events** wizard.



```
//Position profile
CACC = 0.31831;//acceleration rate = 1000[rad/s^2]
CSPD = 3.33333;//slew speed = 100[rpm]
CPOS = 2000L;//position command = 1[rot]
CPR; //position command is relative
MODE PP;
TUM1; //set Target Update Mode 1
UPD; // execute immediate
!MC; WAIT!; // wait for completion
// Define event : After a wait time equal with value 0.5 s
!RT 500L;
WAIT!; // Wait until the event occurs
```

Step 1 – Events

Set Event: Change Event | Edit Event | None

□ Stop the motion when the event occurs

☑ Wait until the event occurs

□ Exit from the wait loop after a time equal ...

OK | Cancel

Step 2 – Event Type

○ None

○ When actual motion is completed

○ Function of motor or load position

○ Function of motor or load speed

● After a wait time

○ Function of reference

○ Function of input status

○ Function of a 32-bit variable value

OK | Cancel | Help

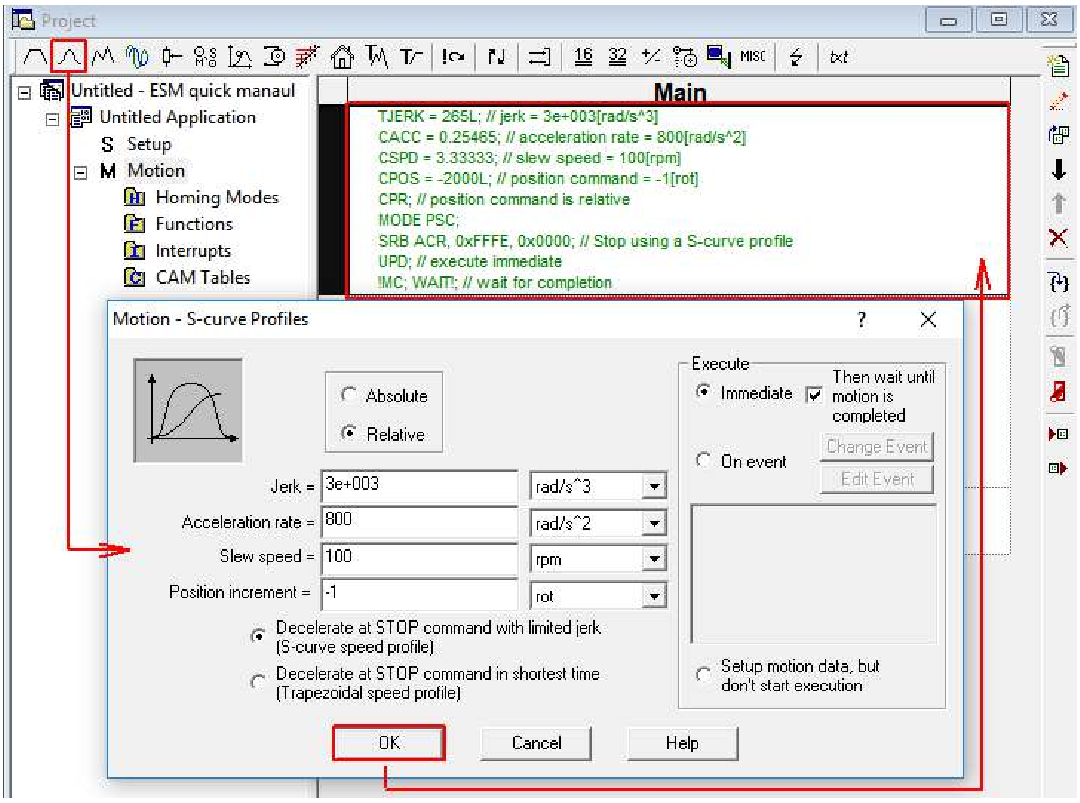
Step 3 – Event - After a Wait Time

After a wait time equal with ● value ○ variable 0.5 s

OK | Cancel | Help

Figure 7.3 - Events dialogue

The S-curve profile will be inserted through the **Motion – S-Curve Profiles** wizard.



```
TJERK = 265L; // jerk = 3e+003[rad/s^3]
CACC = 0.25465; // acceleration rate = 800[rad/s^2]
CSPD = 3.33333; // slew speed = 100[rpm]
CPOS = -2000L; // position command = -1[rot]
CPR; // position command is relative
MODE PSC;
SRB ACR, 0xFFFE, 0x0000; // Stop using a S-curve profile
UPD; // execute immediate
!MC; WAIT!; // wait for completion
```

Motion - S-curve Profiles

○ Absolute

● Relative

Jerk = 3e+003 rad/s^3

Acceleration rate = 800 rad/s^2

Slew speed = 100 rpm

Position increment = -1 rot

● Decelerate at STOP command with limited jerk (S-curve speed profile)

○ Decelerate at STOP command in shortest time (Trapezoidal speed profile)

Execute

● Immediate ☑ Then wait until motion is completed

○ On event — Change Event | Edit Event

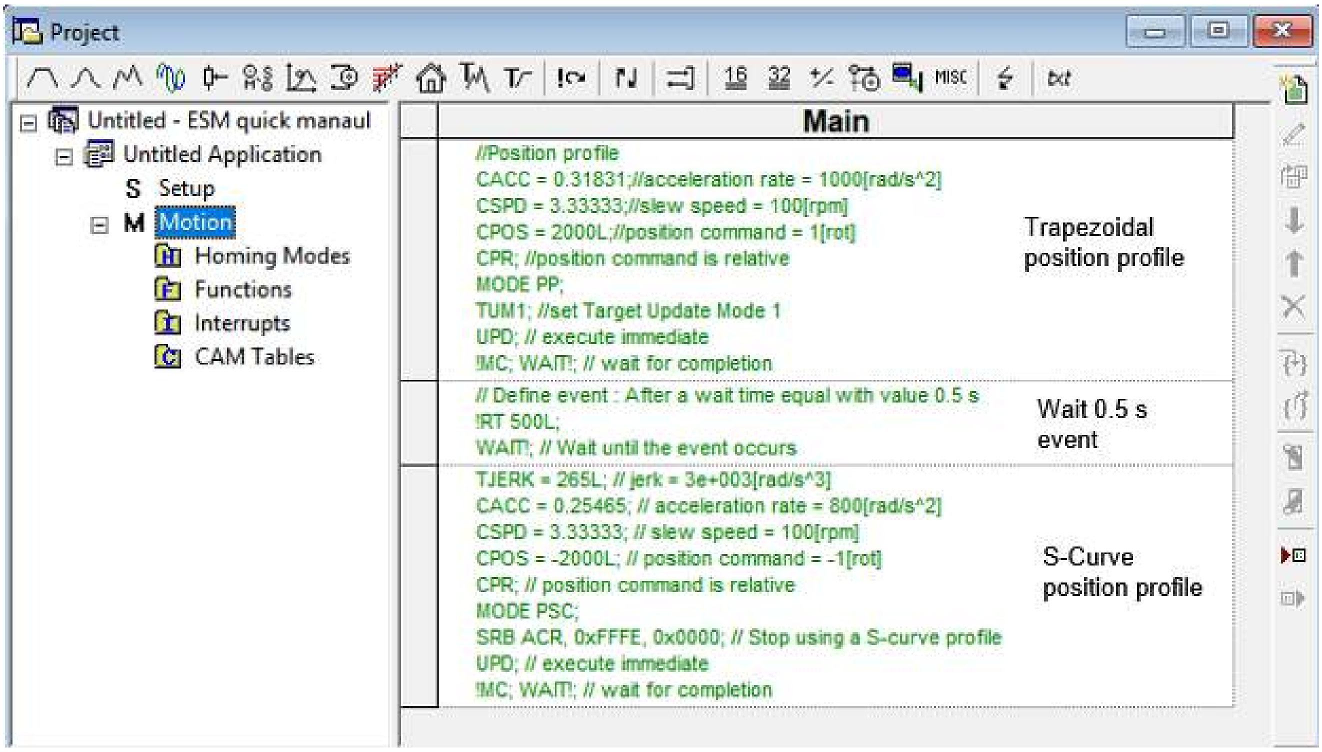
○ Setup motion data, but don't start execution

OK | Cancel | Help

Figure 7.4 - insert an S-curve profile

The parameters inside the motion wizard dialogues, used for above profiles, are explained in the help topics that opens when clicking on the **Help** button from the respective wizard dialogue.

The TML code is generated automatically when a wizard dialogue is closed with the OK button. The picture below shows the TML code generated following above actions.



```
//Position profile
CACC = 0.31831;//acceleration rate = 1000[rad/s^2]
CSPD = 3.33333;//slew speed = 100[rpm]
CPOS = 2000L;//position command = 1[rot]
CPR; //position command is relative
MODE PP;
TUM1; //set Target Update Mode 1
UPD; // execute immediate
!MC; WAIT!; // wait for completion
// Define event : After a wait time equal with value 0.5 s
!RT 500L;
WAIT!; // Wait until the event occurs
TJERK = 265L; // jerk = 3e+003[rad/s^3]
CACC = 0.25465; // acceleration rate = 800[rad/s^2]
CSPD = 3.33333; // slew speed = 100[rpm]
CPOS = -2000L; // position command = -1[rot]
CPR; // position command is relative
MODE PSC;
SRB ACR, 0xFFFE, 0x0000; // Stop using a S-curve profile
UPD; // execute immediate
!MC; WAIT!; // wait for completion
```

Figure 7.5 - Generated TML code

Before running the above TML program, the project must be saved, using the **Save** option in the **Project** menu or the **save** icon in the EasyMotion Studio menu bar.

The TML program can be executed by pressing the blue **running man** in the graphical menu bar.

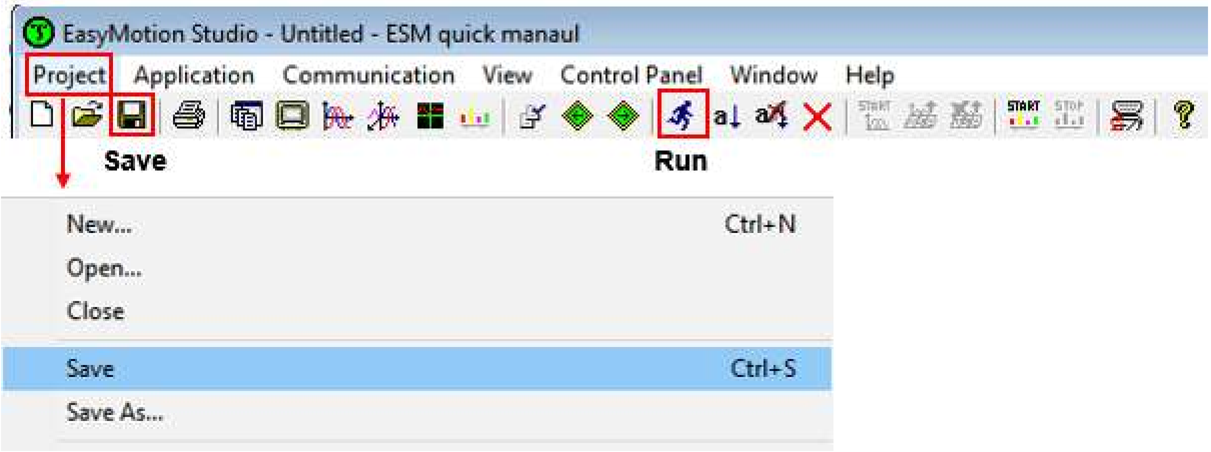


Figure 7.6 – Project “Save” and project “Run” buttons in the graphical menu bar

When the **Run** button is pressed, the drive is reset, the TML application program is downloaded to the drive non-volatile memory and then the drive starts to execute it. The motor should start to move according to the programmed profile (1 rotation in the positive direction, wait 0.5 s and mode 1 rotation in the negative direction).

For more examples see the application notes on our website (link).

8 Motion evaluation

EasyMotion Studio includes a series of **control panels**, a **Logger** and a **Scope** that can be used for the motion evaluation or for the application monitoring, control and debugging. These evaluation tools can be open form the **View** menu or from the **graphical menu bar**.

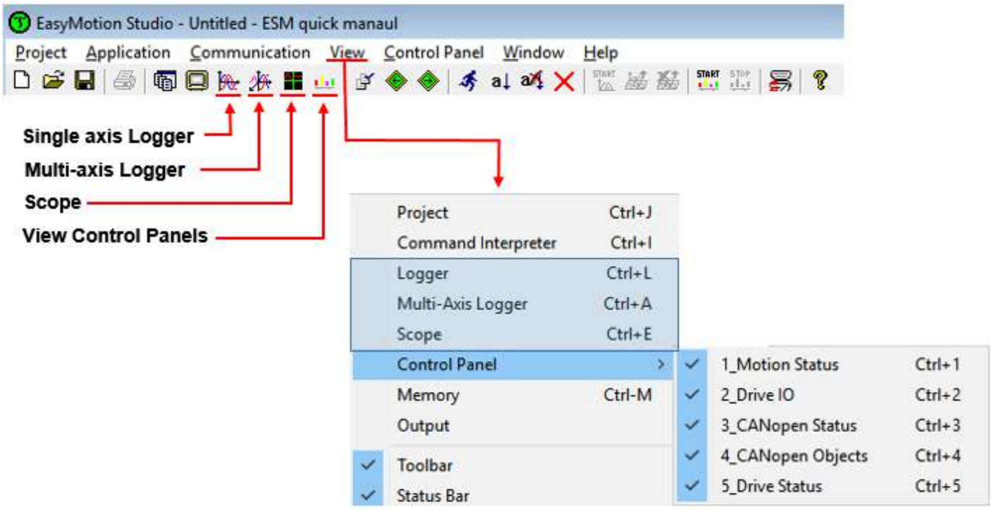


Figure 8.1 - Evaluation tools in EasyMotion Studio

The evaluation tools are described in details in the EasyMotion Studio help topics.

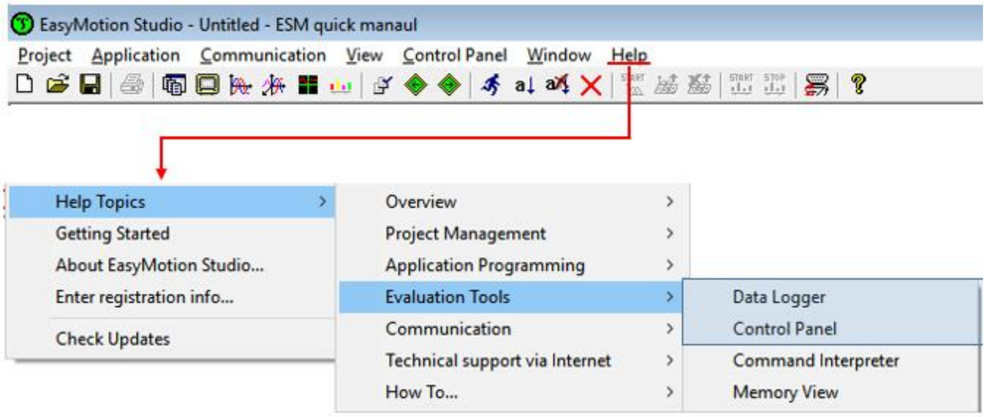
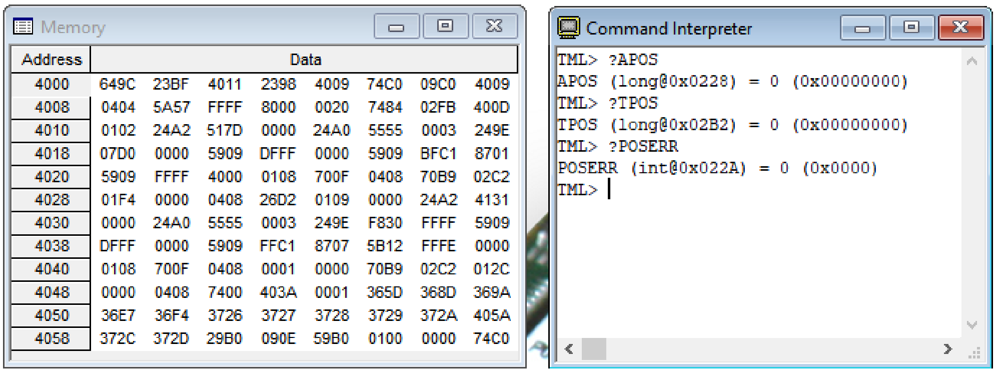

EasyMotion Studio - Untitled - ESM quick manaul
Project Application Communication View Control Panel Window Help
Help Topics
Getting Started
About EasyMotion Studio...
Enter registration info...
Check Updates
Overview
Project Management
Application Programming
Evaluation Tools
Communication
Technical support via Internet
How To...
Data Logger
Control Panel
Command Interpreter
Memory View

Figure 8.2 - Help topics related to the evaluation tools in EasyMotion Studio

Beside the evaluation tools mentioned above, EasyMotion Studio provides also a **Memory View** window, that allows to see the content of all memory locations and a **Command Interpreter** that allows to send online TML commands to the drive. Both tools are described in the EasyMotion Studio help topics.



Address	Data							
4000	649C	23BF	4011	2398	4009	74C0	09C0	4009
4008	0404	5A57	FFFF	8000	0020	7484	02FB	400D
4010	0102	24A2	517D	0000	24A0	5555	0003	249E
4018	07D0	0000	5909	DFFF	0000	5909	BFC1	8701
4020	5909	FFFF	4000	0108	700F	0408	70B9	02C2
4028	01F4	0000	0408	26D2	0109	0000	24A2	4131
4030	0000	24A0	5555	0003	249E	F830	FFFF	5909
4038	DFFF	0000	5909	FFC1	8707	5B12	FFFE	0000
4040	0108	700F	0408	0001	0000	70B9	02C2	012C
4048	0000	0408	7400	403A	0001	365D	368D	369A
4050	36E7	36F4	3726	3727	3728	3729	372A	405A
4058	372C	372D	29B0	090E	59B0	0100	0000	74C0

```
TML> ?APOS
APOS (long@0x0228) = 0 (0x00000000)
TML> ?TPOS
TPOS (long@0x02B2) = 0 (0x00000000)
TML> ?POSERR
POSERR (int@0x022A) = 0 (0x0000)
TML> |
```

Figure 8.3 – "Memory" window (left) and "Command Interpreter" dialogue (right)

8.1 Control panels

The following control panels are available in EasyMotion Studio:

- **Motion Status** - shows the load / target position, the load / target speed and the motor current;
- **Drive IO** - shows the state of the drive digital inputs / outputs and the analogue inputs level;
- **CANopen Status** and **CANopen Objects** - display the value of most important CANopen objects;
- **Drive Status** - displays the bits values in the SRL, SRH, MER, DER and DER2 registers.

The existent control panels can be customized by placing the mouse pointer over a control panel, pressing the right mouse button and select the **Customize** option.

There's also the option to create a new control panel, export it or import it from a file. All these options are available in the **Control Panels** menu.

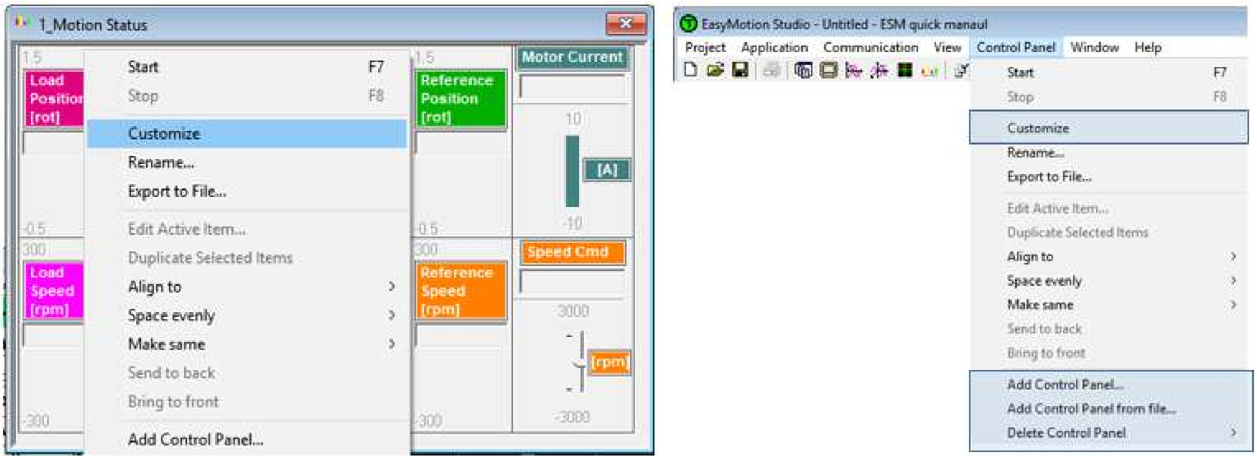


Figure 8.4 - Control panels options

Using the **Motion Status** control panel, to visualize the motion profiles in the example from chapter 7, we will see the curves:

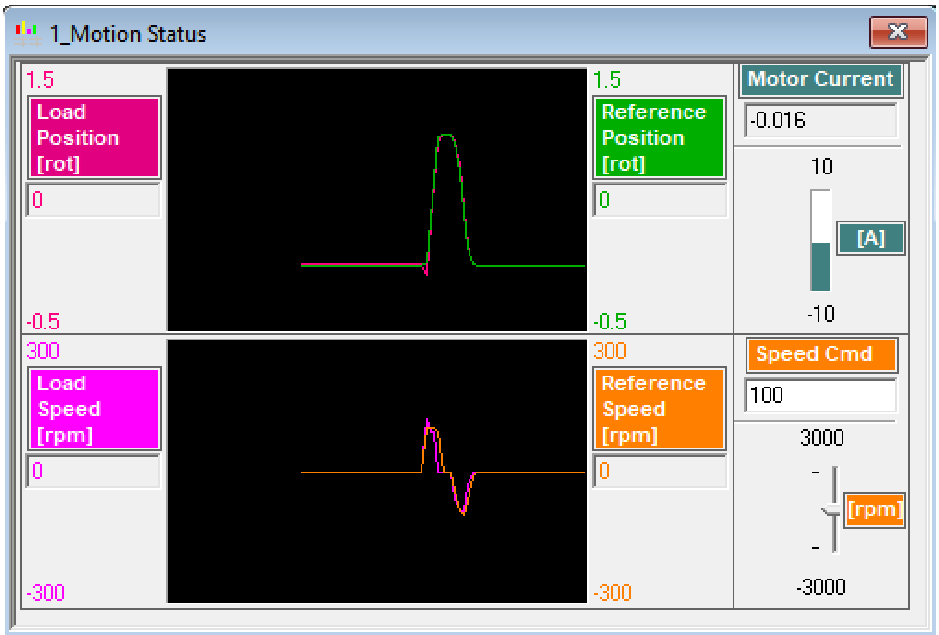


Figure 8.5 - Motion profile evaluation using the Motion Status control panel

The control panels in EasyMotion Studio use the communication line to get the values so the accuracy is not very high, especially if the acquisition is done from a CAN network.

8.2 Logger

The **Data Logger** is an advanced graphical analysis tool that allows to check the evolution of any TML variable, register or parameter.

Using the drive RAM memory to save the points, the logger is very accurate but has the disadvantage that is limited (it can save and plot only a finite number of points).

The Logger can be operate using the buttons in the graphical menu bar or the options in the menu that opens when right clicking over the Logger window area.

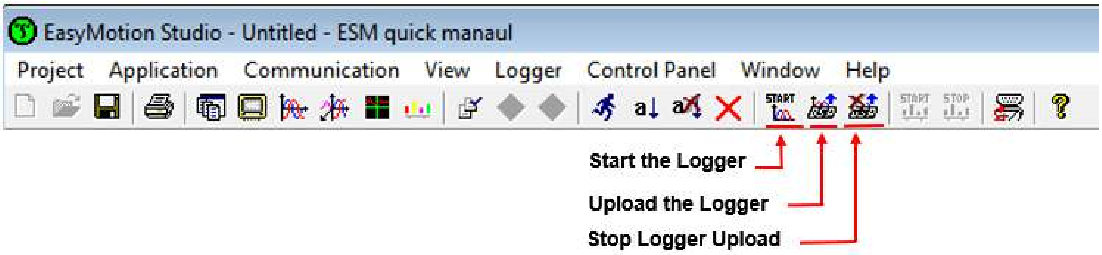

EasyMotion Studio - Untitled - ESM quick manaul
Project Application Communication View Logger Control Panel Window Help
Start the Logger
Upload the Logger
Stop Logger Upload

Figure 8.6 - Logger operation buttons

Remark: The Logger starts automatically when Run the TML application. So, after running the application is enough to press the **Logger Upload** button, to see the system behavior immediately after start-up.

Considering the example in chapter 7, the following plots uploaded, after the motion ends and **Logger Upload** button is pressed.

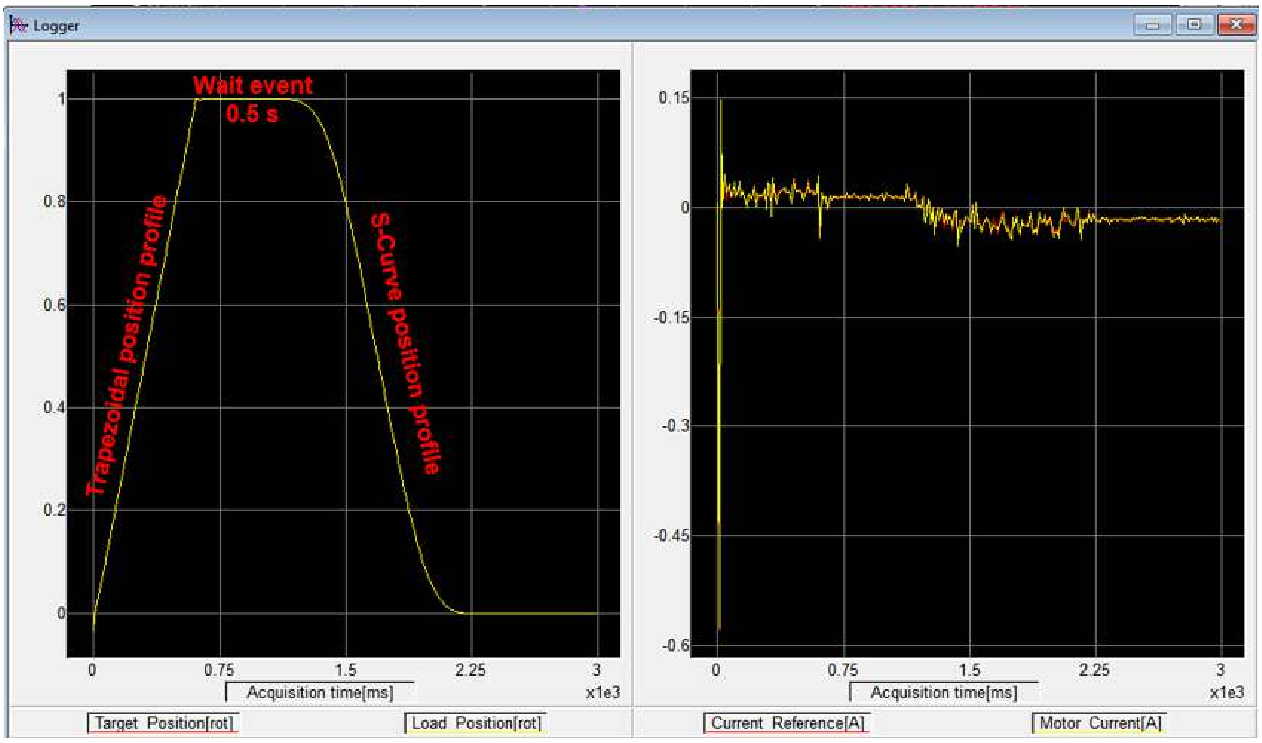


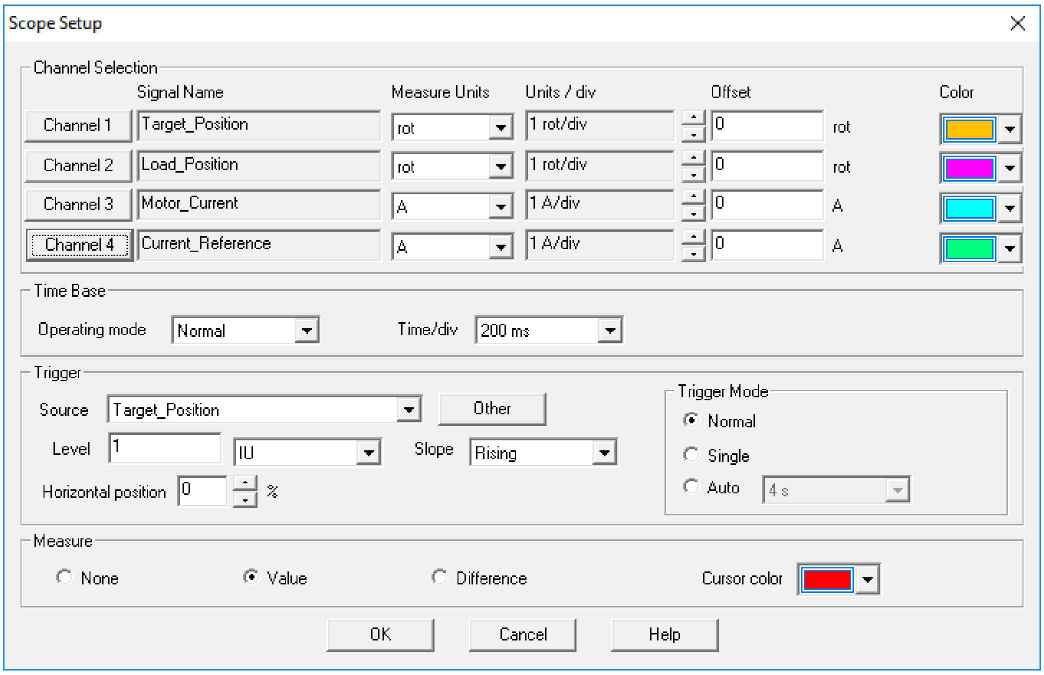
Figure 8.7 - Logger plot for the motion profiles in the example presented in chapter 7

The first plot shows the two motion profiles, separated by the 0.5 s waiting event.
The second plot shows the current reference and the motor current.

Remark: If needed other 2 extra plots can be added (by selecting the **Plot Setup** option in the menu that opens when right clicking over the Logger surface) or the Logger can be set to save other variables too (by selecting the **Variables** option in the same Logger menu).

8.3 Scope

The **Scope** evaluation tool is similar to a 4 channels oscilloscope. It can be set (right click on the Scope surface | Setup) to display (in real time) different information (variable, input/output state, registers etc.).



Scope Setup

Channel Selection

	Signal Name	Measure Units	Units / div	Offset		Color
Channel 1	Target_Position	rot	1 rot/div	0	rot	🟧
Channel 2	Load_Position	rot	1 rot/div	0	rot	🟪
Channel 3	Motor_Current	A	1 A/div	0	A	🟦
Channel 4	Current_Reference	A	1 A/div	0	A	🟩

Time Base

Operating mode: Normal   Time/div: 200 ms

Trigger

Source: Target_Position   Other

Level: 1   IU   Slope: Rising

Horizontal position: 0 %

Trigger Mode

● Normal

○ Single

○ Auto   4 s

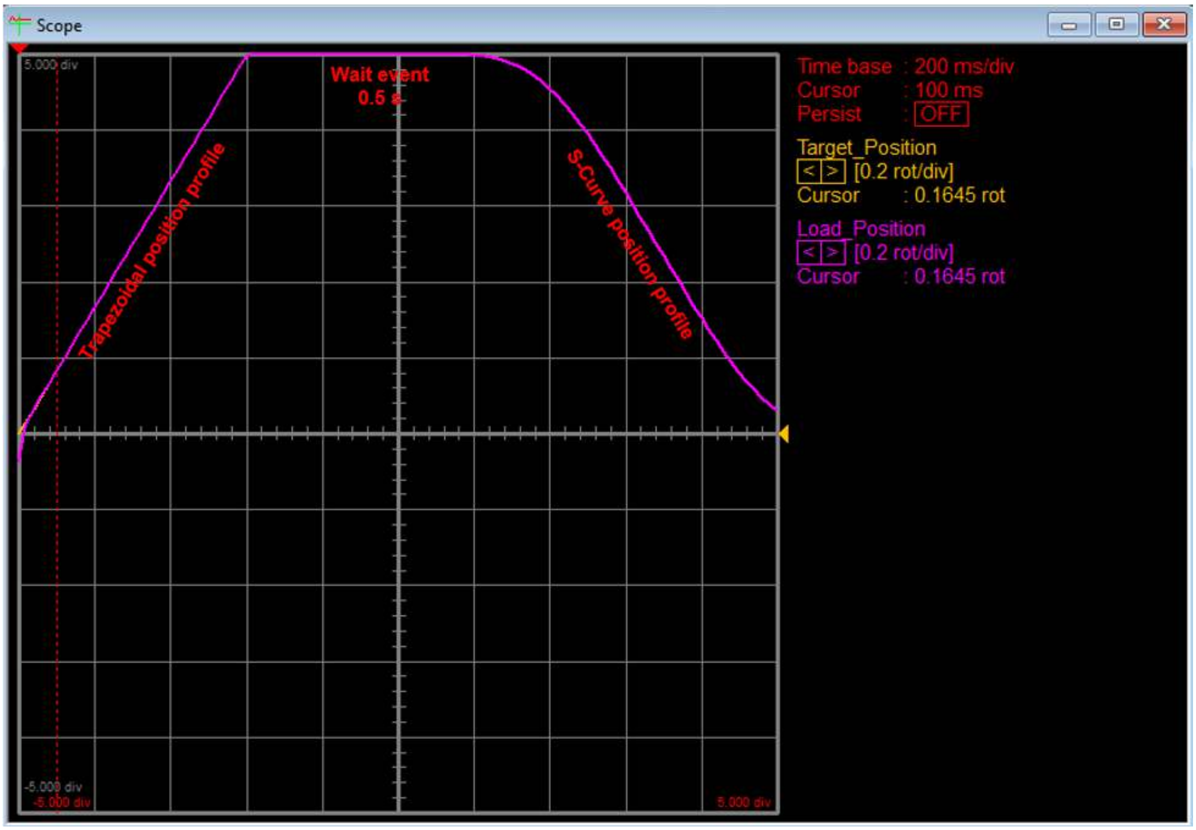
Measure

○ None   ● Value   ○ Difference   Cursor color 🟥

Figure 8.8 - Scope Setup window

The **Help** button in the **Scope Setup** window opens a help topic that describes the Scope use and setting.

Using the Scope to evaluate the motion profiles from chapter 7, the following curve should be displayed.



Time base : 200 ms/div
Cursor : 100 ms
Persist : OFF

Target_Position
< > [0.2 rot/div]
Cursor : 0.1645 rot

Load_Position
< > [0.2 rot/div]
Cursor : 0.1645 rot

Figure 8.9 - Motion evaluation using the Scope

9 Miscellaneous – Project archive / restore

For project storring / transfer purpose EasyMotio Studio includes a **Archive** and **Restore** options.

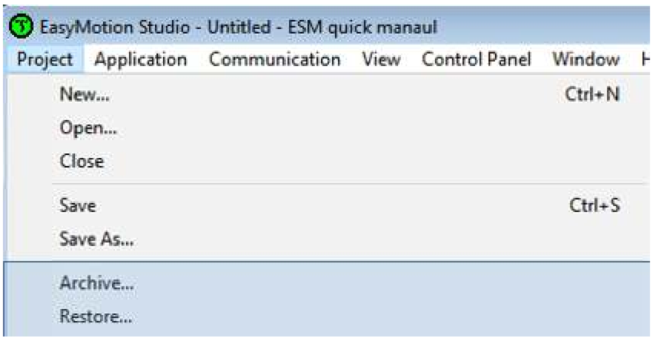
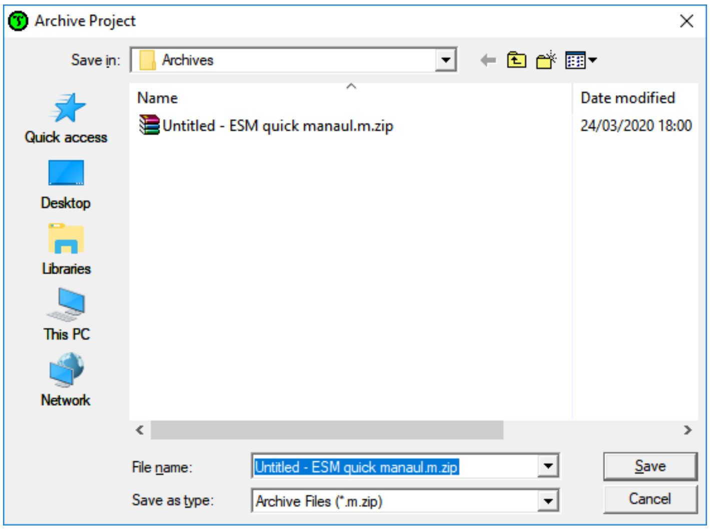


Figure 9.1 – "Archive" and "Restore" options

The **Archive** options creates a special archive (with "m.zip" extension) of the project, including the Logger setting, the control panels and other custom files too. This option is recommended when a project back-up is done or when the project needs to be transferred to other PCs.



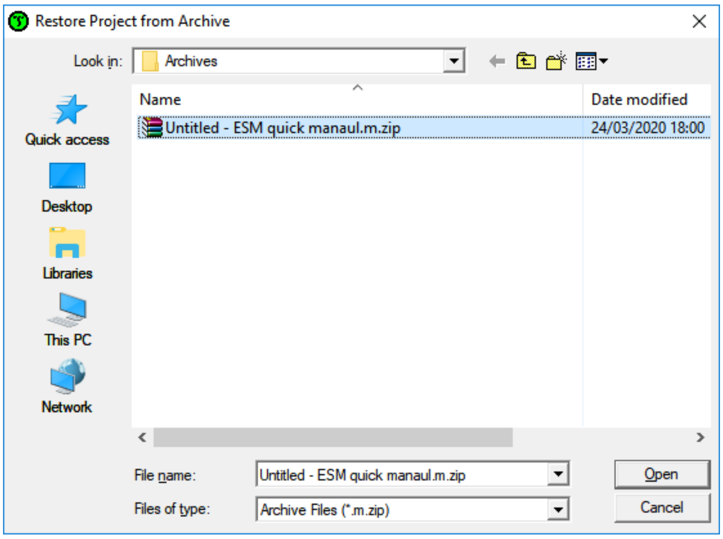
Untitled - ESM quick manaul.m.zip 24/03/2020 18:00

File name: Untitled - ESM quick manaul.m.zip

Save as type: Archive Files (*.m.zip)

Figure 9.2 - Project archiving

The **Restore** option extracts the "*.m.zip" archive content and restores it as a full EasyMotion Studio project. The restored project is identically with the one from where the archive was created.



Untitled - ESM quick manaul.m.zip 24/03/2020 18:00

File name: Untitled - ESM quick manaul.m.zip

Files of type: Archive Files (*.m.zip)

Figure 9.3 - Project restoring

TECHNOSOFT