



SAFETY	PAGE 2
INTRODUCTION	PAGE 3
SOME USEFUL CONCEPTS	PAGE 4
DETAILS OF METAL WORK HARDWARE SYSTEM	PAGE 8
MWDRIVE SOFTWARE	PAGE 16
PARAMETER SETUP MENU	PAGE 19
PROGRAM MENU	PAGE 30
LIST OF COMMANDS	PAGE 46
TEST AND JOG MENU	PAGE 55
APPLICATION EXAMPLES	PAGE 58
REFERENCES	PAGE 69

# SAFETY

The e.drive by Metal Work must be handled, installed and used by a qualified fitter specialising in programmable drives for motors and related problems.

## CE MARKING

The product described in this manual complies with the Low Voltage Directive 2006/95/EC and the EMC Directive 2004/108/EC, as amended.

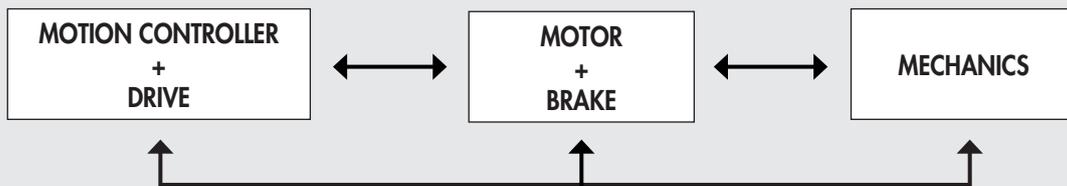
-  **FORCED VENTILATION, AUTOMATIC CURRENT REDUCTION**  
Forced ventilation can be required depending on the application conditions of use. The heatsink temperature must not exceed 65°C. However, it is advisable not to exceed the operating limits shown in the Technical Specifications. In the stall condition, the drive automatically reduces the current value to 60% of the set value.
-  **ELECTROMAGNETIC INTERFERENCE!**  
The part relating to the drive and the power supply devices (transformer, condenser, fuses, motor, rectifier bridge and cables) contribute to generate EMC noise. In order to comply with the applicable standards (EN 61800-3), it is strictly recommended to adopt the following precautions:
  - proper shielding of control signal cables, motor and power supply;
  - suitable grounding of components parts, using cable as short and with the less inductance as possible;
  - using transformers with a metal shield between primary and secondary winding. The use of auto-transformers for accident prevention or other purposes is not admitted.
  - using appropriate mains filters, especially as close as possible to the electrical panel power input. The use of a double-cell filter (e.g. Schaffner FN660-10-06) is recommended.
  - using clamp-on ferrites on the various cables, if deemed necessary.
-  **WARNING: FIRE HAZARD, BURNS DUE TO HEAT AND RESIDUAL VOLTAGE!**  
Given the currents and the voltages involved, the part relating to the drive can heat up considerably. The drive temperature is controlled by suitable internal sensors that generate three different types of alarm. The drive is designed to be installed at a certain distance from other devices to allow the circulation of air. The minimum distance to allow heat dissipation is 5 cm. Do not obstruct the ventilation grilles so as to ensure correct dissipation of the heat generated.  
Depending on how the user has created the power stage supply circuit, a waiting time may be required after powering off the equipment. The connectors must not be removed and/or fitted with the equipment turned on. After switching off, wait a few minutes to allow for some of the components to cool down.
-  **MECHANICAL HAZARD!**  
The mechanical forces at play in an actuator are of such an extent as to cause damage to property or injury to persons. The speed, acceleration, dimensions and positions of the axis are controlled by electronic systems (a combination of HW and SW tools), which, although of a high quality standard, may be subject to breakdowns and malfunctions.
-  **CAUTION: RISK OF UNWANTED MOVEMENTS OR FAILED STOP!**  
The mechanical brake cannot be considered a safety device.  
The motion controller incorporated in the drive does not guarantee that the axis does not move at all, even in the absence of logical run conditions. Unwanted start-up cannot be excluded completely.
-  **SAFETY**  
The use in safety components for safety functions (EN 60204-1) or with application methods that can entail a risk of damage when a circuit board fails is prohibited. The installation in machines or equipment subject to one or more EC directives, without having declared that they comply with these directives, is also prohibited.

## INTRODUCTION

Aiming at offering an increasingly comprehensive and efficient customer service, Metal Work offers an easy-to-use programmable e.drive with the best price-to-quality ratio.

The e.drive has been designed and developed to facilitate the control of electric STEPPER motors with two poles and two, four or eight wires. It can be used immediately by everyone, i.e. those who do not possess specific technical skills in using motors, drives and logic-programmable devices. A brief description is provided, however, of the basic concepts and notions concerning the movement of electric axes.

In general, a positioning system consists of various devices that are physically and logically interconnected.



*Typical motion control system*

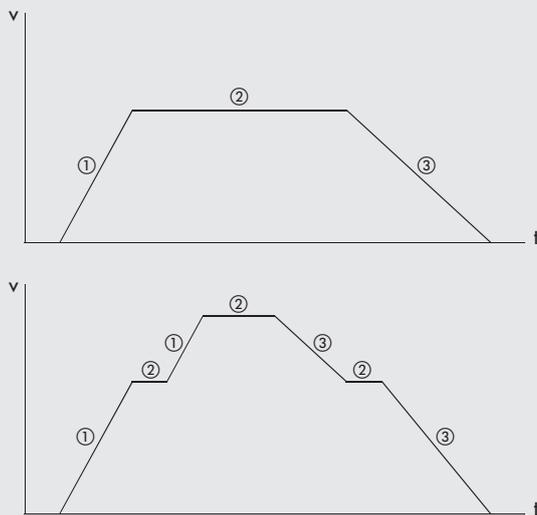
1. **MOTION CONTROLLER:** it is used to give the desired movement to the linear axis (electric actuator). The physical concepts typically involved in the kinematics are: speed (m/s), acceleration (m/s<sup>2</sup>), deceleration (m/s<sup>2</sup>), position (m), torque (Nm). The motion programming logic is expressed via the occurrence of software and hardware events, which are in turn controlled by software-based instructions created by the designer and translated into the desired movement. Other useful functions are made available to manage alarms and outbound signals. This type of controller is not designed to boost synchronisation with other kinematic mechanisms and thus it is not recommended for interpolations or X- or Y-axis precision machining.
2. **DRIVE:** the electric drive can be considered a special device that converts input electric power into output mechanical power, through the use of a motor. This conversion generally requires the use of power electronics by following a specific function, known as control function. In this way, you can command the electric motor to operate as desired to achieve a preset aim. In practice, there are two different families of drive, depending on the type of motor (BRUSHLESS or STEPPER).  
The e.drive incorporates in one device both the functions described, in terms of logic and power.
3. **MOTOR:** BRUSHLESS motors are supplied complete with the drive designed specifically for that particular type of motor. STEPPER motors are generally controlled by drives that are not dedicated but have compatible electrical features. They can be equipped with or without holding brake. The torque and speed of the motor with the drive must be such as to move the axis under the worst possible conditions. These details are generally specified when sizing the system for a particular application.
4. **ENCODER:** an angular position transducer (encoder) is an electromechanical device capable of converting the angular position of its rotary axis into short electrical pulses that need to be processed by a signal analysis circuit in the form of digital numerical signals. There are different types of encoder, but the following two categories are the most commonly used:
  - Incremental encoders. These encoders allow simple circuits to read and display the speed and acceleration of the axis concerned, but not the instant position.
  - Absolute encoders. These encoders allow an appropriate decoding circuit to decode and display at any time the angular position of the axis concerned. Axis movement data (direction, speed and acceleration) is the result of the processing of the axis's absolute position over time.
5. **MECHANICAL PART:** this component transforms the rotary movement of the motor into a typically linear movement, e.g. in a classic electric actuator, a spiral/worm screw coupling converts the rotary movement of the motor into a translation applied to a load.

# SOME USEFUL CONCEPTS

## LAW OF MOTION

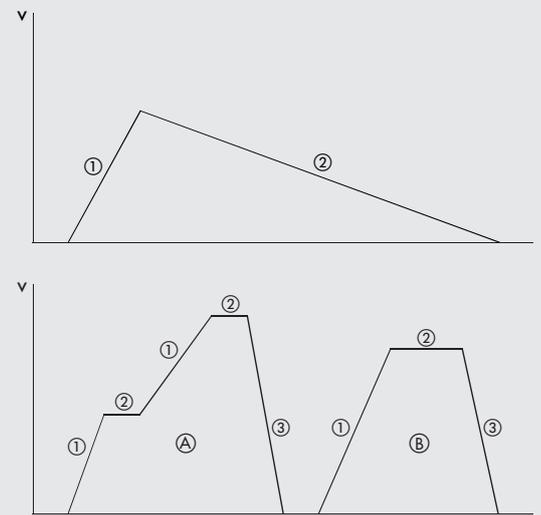
- **Kinematics:** is the study of the motion of objects (space, time, speed and acceleration).
- **Dynamics:** is the study of the causes of motion, such as forces and torques.
- **Statics:** is the study of the balance of objects. It considers the conditions that allow objects to remain still (or prevent movement).
- **Trajectory:** is the geometrical place formed by the positions acquired by the centre of mass of a moving body. In classical mechanics, it is in general a continuous curve formed through space described by a reference system.
- **Spatial reference system:** is a set of all the objects in respect of which the movement takes place with the same characteristics. It is represented schematically with a set of three Cartesian axes. It consists of three straight lines ( $x, y, z$ ), each of which is perpendicular to the other two. The straight lines have a common point ( $O$ ), called the origin of the axes. The position of a point  $P$  with respect to this set of Cartesian axes is known if its three coordinates are known.
- **Law of motion:** is the relationship between position and time. It can be expressed using a table, a chart or a mathematical formula. Below is a graphic illustration of some of the most frequently used laws of motion, using speed-time diagrams.

### Types of laws of motion



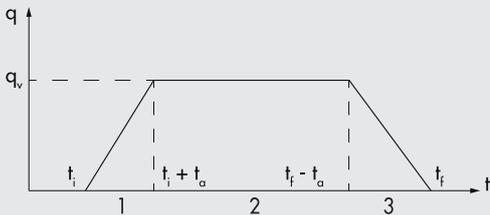
① Acceleration      ② Constant speed phase      ③ Deceleration

### Law of triangular motion and multiple laws (A, B)



Ⓐ First law of motion      Ⓑ Second law of motion

A process widely used in industrial practice to generate a trajectory consists of planning a linear position profile joined at the start and end of the trajectory with parabolic sections. The resulting speed profile has the typical trapezoidal trend.



The trajectory is, therefore, divided into three parts:

1. Constant acceleration, ramp speed, parabolic position;
2. Null acceleration, constant speed, linear position;
3. Constant deceleration, ramp speed, parabolic position.

Typically, acceleration time (phase 1) equals deceleration time (phase 3). This gives a symmetric trajectory with respect to the average time between the start and end time.

Clearly, it must be  $t_a \leq (t_f - t_i)/2$ .

In applications requiring fast and instant handling, high-performance motors and drives must be used.

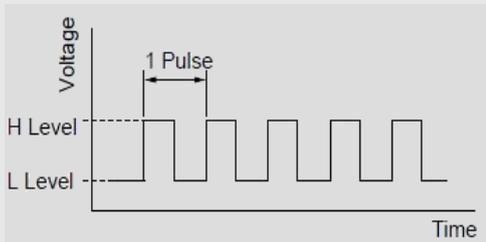
In order to make the motion of the actuator smoother and less abrupt, the concept of "jerk" has been introduced.

This parameter indicates the acceleration variation over time and is used to connect different phases of the motion. A better response from the positioning system can be obtained by adopting these tips.

These considerations are noticeable not only in terms of pure performance but also to preserve the lifecycle and efficiency of the entire system.

## STEP OR PULSE

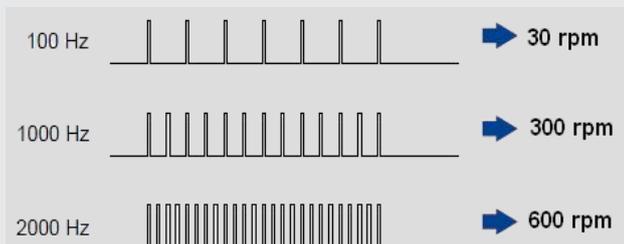
This is a digital electrical system that repeatedly switches between ON and OFF. Each ON/OFF is a pulse.



The angle of rotation of the motor is proportional to the number of pulses provided by the drive.



The pulse frequency (number of pulses per second measured in Hertz "Hz") determines the motor rotation speed.



Example: Calculation of rotation speed  $N$  [rpm] =  $\frac{\text{Frequency} \times 60}{\text{pulse/rev}}$

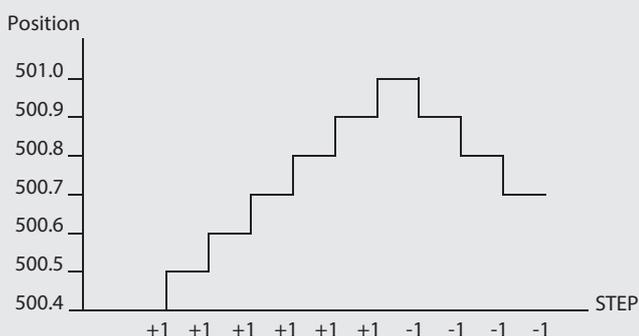
## RESOLUTION

When working by steps or pulses, the actuator stroke is divided into a finite number of positions, the result being that the user has only discrete values for positioning. The different positions can be reached by giving the correct number of steps.

Resolution depends on two factors: the screw pitch (for screw-driven electric axes) or the pulley pitch diameter (for belt-driven electric axes), which define the feed by rotation of the motor shaft (with mechanical transmission ratio, if any) and motor indexing derived from the combination of motor + drive.

Indexing means the number of positions in which the shaft stops, when making one revolution.

For example, a pitch-10 screw (10 mm step forward per rev) driven by a motor/drive assembly with 400 stop positions/rev has a final resolution of 10/400, i.e. 0.025 mm/step. As can be seen, the resolution is always very high in systems with long pitch and low indexing.



Concept of step and resolution

## ACCURACY

The accuracy of the system depends on numerous factors, such as mechanical backlash, failure due to the load and operating temperature. With reference to the previous example, when using a STEPPER motor without any loss of step, the maximum failure will be  $1.8^\circ$  with a maximum load that is resistant and opposed to the motion of the actuator.

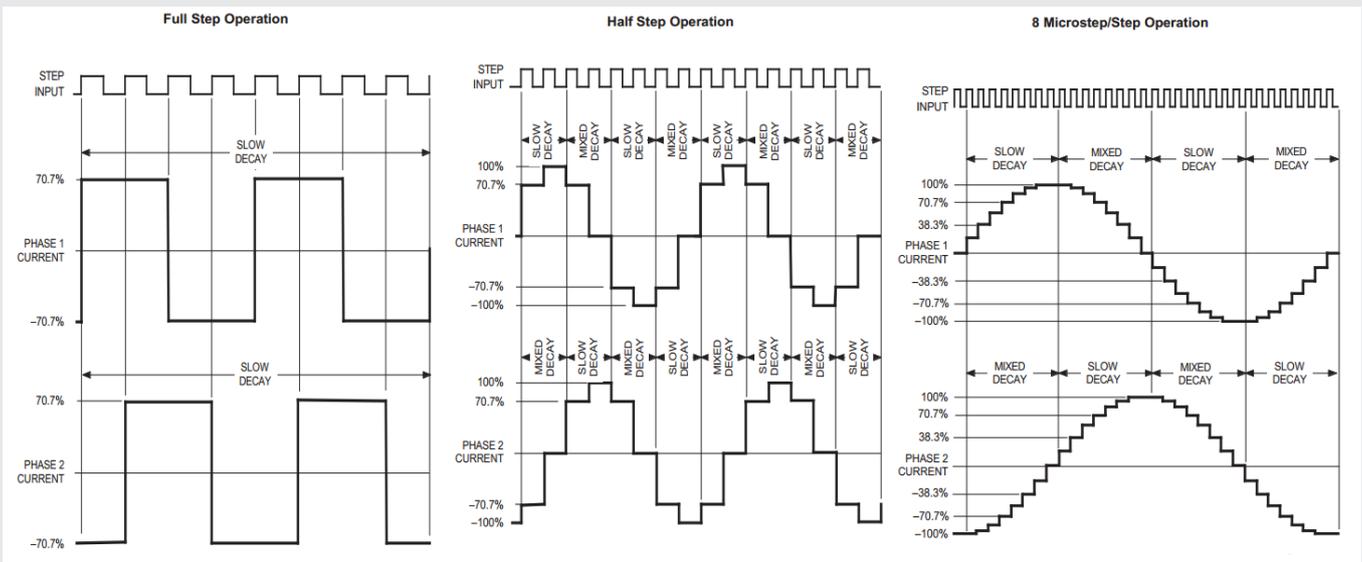
With a 10 mm screw pitch, this corresponds to 0.025 mm, but the value is generally much lower. If the load moves in accordance with the motion of the actuator, the failure will be the same magnitude but in the opposite direction. Accuracy is often little affected by the indexing set on the drive.

## MICROSTEPPING

With microstepping control, the motor shaft can be moved by a fraction of a step that is less than the rated step, generally known as "full step". The fractions can be  $1/2$ ,  $1/4$ ,  $1/8$ ,  $1/16$  of a step. The result is a considerable increase in motor resolution. Moreover, this type of control reduces motor resonance, a phenomenon that depends on the applied load, usually occurring at low speeds. A typical motor has a  $1.8^\circ$  step angle, which corresponds to 200 steps per revolution. 200 impulses supplied to the drive will cause the motor to run 200 steps if a "full step" technique is used. In practice, the microstepping technique provides, via an increased number of electrical impulses, a signal that is as close as possible to the ideal signal to ensure that the motor covers the 200 steps per revolution more efficiently. In addition to increasing overall system resolution, the result is improved supply of current and hence the ability of the system to apply a constant torque during motion that takes place at a constant speed rate.

No. of IMPULSES	STEP MODE
200	Full step
400	Half step
800	1/4 step
1600	1/8 step
3200	1/16 step

The figure below shows the differences between a full step signal, a half step signal, a microstepping signal and a 1/8 signal. The ideal signal that would be the best possible signal to send to the motor is represented by a sinusoid waveform.



## OPEN LOOP VS CLOSED LOOP POSITIONING SYSTEMS

The axis reaches the required position because the motion controller provides the drive with a train of pulses (steps), the number of which is determined by dividing the desired value by the resolution of the system.

For example, for a 200 mm travel, with a 0.025 resolution per step,  $200/0.025$ , i.e. 8000 steps, will be sent to the appropriate direction of the drive. These pulses "leave" the motion controller and are directed to the drive, which converts them into appropriate currents to generate the rotation of the motor for the exact angle needed to cover the 200mm required.

In practice, two control modes are used: the open-loop systems and the closed-loop systems.

In an **open-loop** system, which is typically used with STEPPER motors without an encoder, the motor is supplied with such currents as to make it rotate by the desired angle, but the position reached is not controlled in real time. The position is actually reached when the driving force of the motor is greater than the resistive load. If any problems occur, the system does not react appropriately for correction.

In a **closed-loop** system, which is typically used with BRUSHLESS motors with an encoder, the motor rotates until the set position is reached.

The position is read in real time by a transducer, which can be an encoder arranged coaxially to the motor.

Drive torque is applied to the motor until the desired position is reached.

The position is not fixed; it varies according to the resistive load.

The closed-loop system provides the following advantages:

- Incorrect positioning diagnostics;
- Energy saving efficiency, resulting from modulation of torque according to load resistance;
- Reduced resonance and noise.

The e.drive controller allows diagnostics also on STEPPER motors, thanks to the possibility of connecting and managing an encoder mounted on the axis (directly on the motor or in line with the axis).

## BRAKE

A non-electrically powered motor does not provide any torque and hence it cannot hold a load. Unpredictable or hazardous effects may, therefore, occur due to the fact that the load moves freely. In this case, a motor with an electrically-controlled holding brake can be used. Typically, this brake features a static torque (with the actuator at a standstill) greater than the rated torque of the motor thus ensuring the correct locking. The control logics are devised in such a way as to lock the axis in the absence of supply voltage. To free the axis, you need to release the brake. In order to guarantee a longer lifecycle of the brake, it is advisable to activate it only when the motor is at a standstill, as it is a holding brake. The brake release command can be given by the e.drive.

# DETAILS OF METAL WORK HARDWARE SYSTEM

## e.drive SPECIFICATIONS

Logic stage parameters		Min	Max
Supply voltage *	VDC	18	30
Supply current **	mA		200
Encoder supply voltage	VDC	4.75	5.25
Encoder supply current	mA	-	500
Encoder channel frequency	kHz	DC	100
Sensor supply voltage	VDC	9.6	10.4
Sensor supply current	mA	-	20
External device supply voltage	VDC	18	30
External device supply current	mA	-	1000
Output current *** 1-6	mA	-	400
Output current 7	mA	-	600
Maximum voltage per output	VDC	18	30
Protection class	-	IP 20	
Operating temperature	°C	0	50
Relative humidity (Non-condensing)		10%	85%
Dimensions L x H x P	mm	148x99x50.5	
Weight	g	790	

Specifications power stage		Min	Max
Drive supply	VDC	20	55
Motor phase current (peak)	A	1	6
Motor inductance	mH	1	12

Table 1

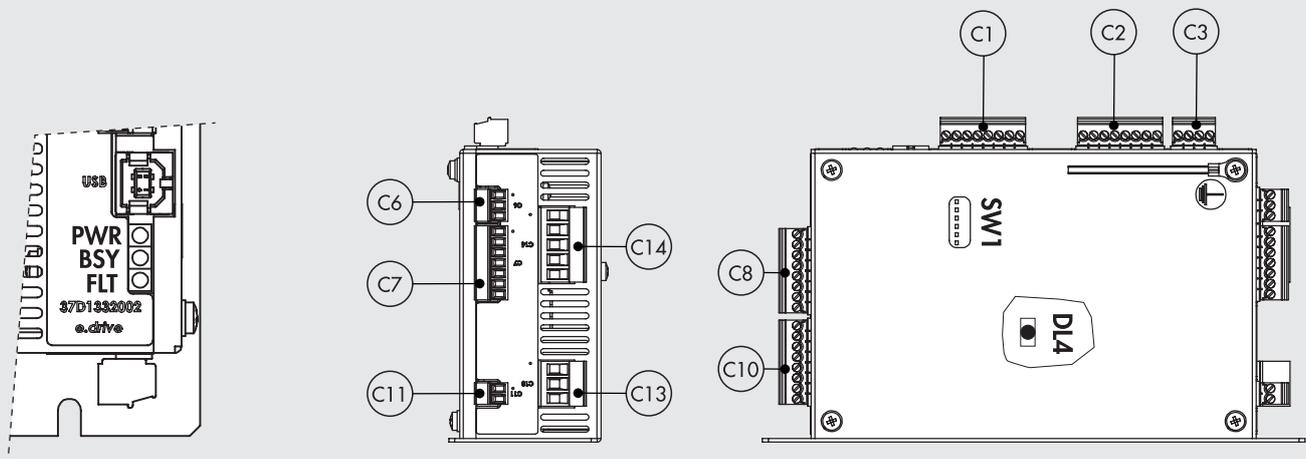
\* Below the 18V, the motion parameters are saved and the system locks up; while above the 30V, permanent electric failures may occur. The recommended operating range is 21.6 - 26.4V.

\*\* Without the loads powered at 24V

\*\*\* The max. current per output is 400 mA, for up to 4 simultaneously energised outputs, and 300 mA for more than 4 simultaneously energised outputs.

## CONTROL BOARD LAYOUT

The plan below highlights the elements of interest to the user. It shows the numbering of the connectors, signal LED lights and the setting dip-switch.



Plan view of the control board with the main references

## DIP-SWITCH SETTINGS

For some settings, the control board is equipped with internal DIP switches, called SW1 (see control board plan).

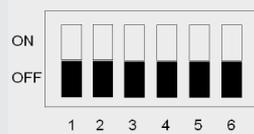
Ref	Parameter	Position	Default
DIP1	Foreseen to be connected to RS485 (termination inserting).	ON = termination enabled OFF = termination disabled	OFF = disabled
DIP2	Defines the 24V or 5V encoder input (A). Refer to encoder user manual or technical specifications.	ON = encoder powered at 5V OFF = encoder powered at 24V	OFF = 24V
DIP3	Defines the 24V or 5V encoder input (B). Refer to encoder user manual or technical specifications.	ON = encoder powered at 5V OFF = encoder powered at 24V	OFF = 24V
DIP4	Defines the 24V or 5V encoder input (C). Refer to encoder user manual or technical specifications.	ON = encoder powered at 5V OFF = encoder powered at 24V	OFF = 24V
DIP5	Board Firmware update.	ON = Updating OFF = Run	OFF = Run
DIP6	Reserved for future upgrade.		OFF

Table 2

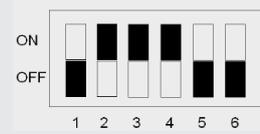
 The incorrect setting of the supply voltage may result in damage or even a breakdown of the encoder.



e.drive in UPDATE MODE pin 5 ON



24VDC encoder connection  
- PUSH-PULL encoder  
- OPEN COLLECTOR encoder  
All pins OFF



5VDC encoder connection  
- LINE DRIVER encoder  
pins 2-3-4 ON

## KEY TO Leds

e.drive has three Leds to indicate the current state of the device.  
There is also an internal Led, DL4, to display the correct operation of the microprocessor.  
Refer to the plan view of the control board for the layout.

Led	State	Description
DL1 (Fault)	 <b>OFF</b>	No FAULT
	 <b>ON (Red)</b>	Not used
	 <b>FLASHING (Red)</b>	Fault
DL2 (Enable motor driver)	 <b>OFF</b>	Drive disabled
	 <b>ON (Yellow)</b>	Drive enabled or CPU "locked" for FW update
	 <b>FLASHING (Yellow)</b>	Not used
DL3 (Power-ON)	 <b>OFF</b>	Board powered off or faulty
	 <b>ON (Green)</b>	CPU "locked"
	 <b>FLASHING (Green)</b>	Board powered on
DL4	 <b>OFF</b>	Board powered off or faulty
	 <b>ON (Yellow)</b>	CPU "locked" for FW update
	 <b>FLASHING (Yellow)</b>	Board powered on

Table 3

## ENCODER CONNECTION

If an encoder is used, e.g. for motion diagnostics, special care must be taken with the wiring.

Encoder models powered at 5VDC can be connected directly to e.drive.

The corresponding connector supplies the encoder with maximum 500 mA current.

Encoder models powered at 24VDC require an external power supply.

Before connecting the encoder to the e.drive, make sure the encoder voltage is the correct level.

Refer to Table 3 of page 8 for the correct matching of switches on the dip-switch.

The hardware of these switches must be configured with the boards powered off.

Two connection modes are provided, according to the interface of the signals coming from the encoder.

- Line-driver encoder outputs: these are differential outputs, i.e. they use 2 wires for each signal. This type is recommended when encoder and e.drive are placed at a long distance (indicatively >2 m) and in electrically noisy environments.
- Single-ended encoder outputs: these are open-collector outputs that use a single wire for each signal. This category includes push-pull encoder outputs.

The connection for the three different types of encoder is illustrated in the figures below.



Wiring of a line-driver encoder powered at 5VDC.



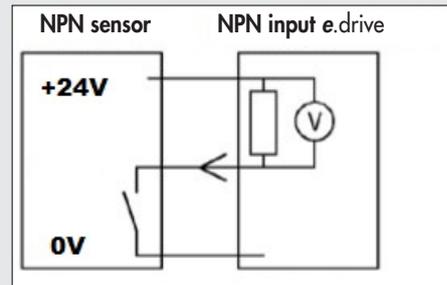
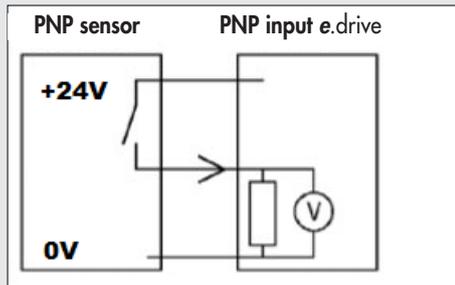
Wiring of a push-pull encoder powered at 24VDC.



Wiring of an NPN open-collector encoder powered at 24VDC.

## DIGITAL INPUT CONNECTION

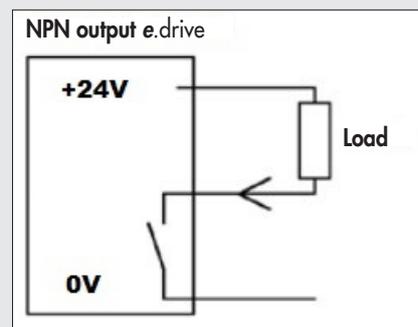
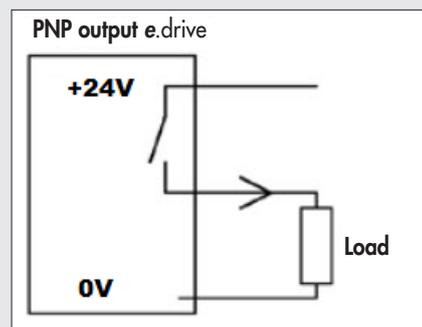
The e.drive has 14 24VDC digital inputs that can be used for such connections as limit switches, proximity switches and contacts in general. Current consumption of each input is 2 mA and software filtering techniques are provided to prevent the occurrence of false signals due to disturbances. Filtering parameters are entirely software configurable. The international standards for this type of signals are two: PNP or NPN. The configuration of each input can be set by the software as well. Below are the wiring diagrams for the two described types.



Example of connection of a sensor to a PNP input (left) and NPN input (right, on request).

## DIGITAL OUTPUT CONNECTION

The e.drive has 7 24VDC digital outputs that can be used for such connections as actuator control, solenoids, lamps and valves. The internal electronics are provided with PNP outputs. On request, they can be provided with NPN outputs or even software programmable PNP-NPN outputs. The voltage applied to the load is substantially the supply voltage, which must be less than 26.4VDC. An internal protection is already provided for inductive loads as long as the inductive energy is less than 50 mJ. Special care must be taken if the associated inductance exceeds 0.4 H. Current and voltage specifications are the same for each channel in both modes: PNP and NPN.



Example of connection of load to a PNP (left) and a NPN digital output (right, on request).

## ANALOGUE INPUT

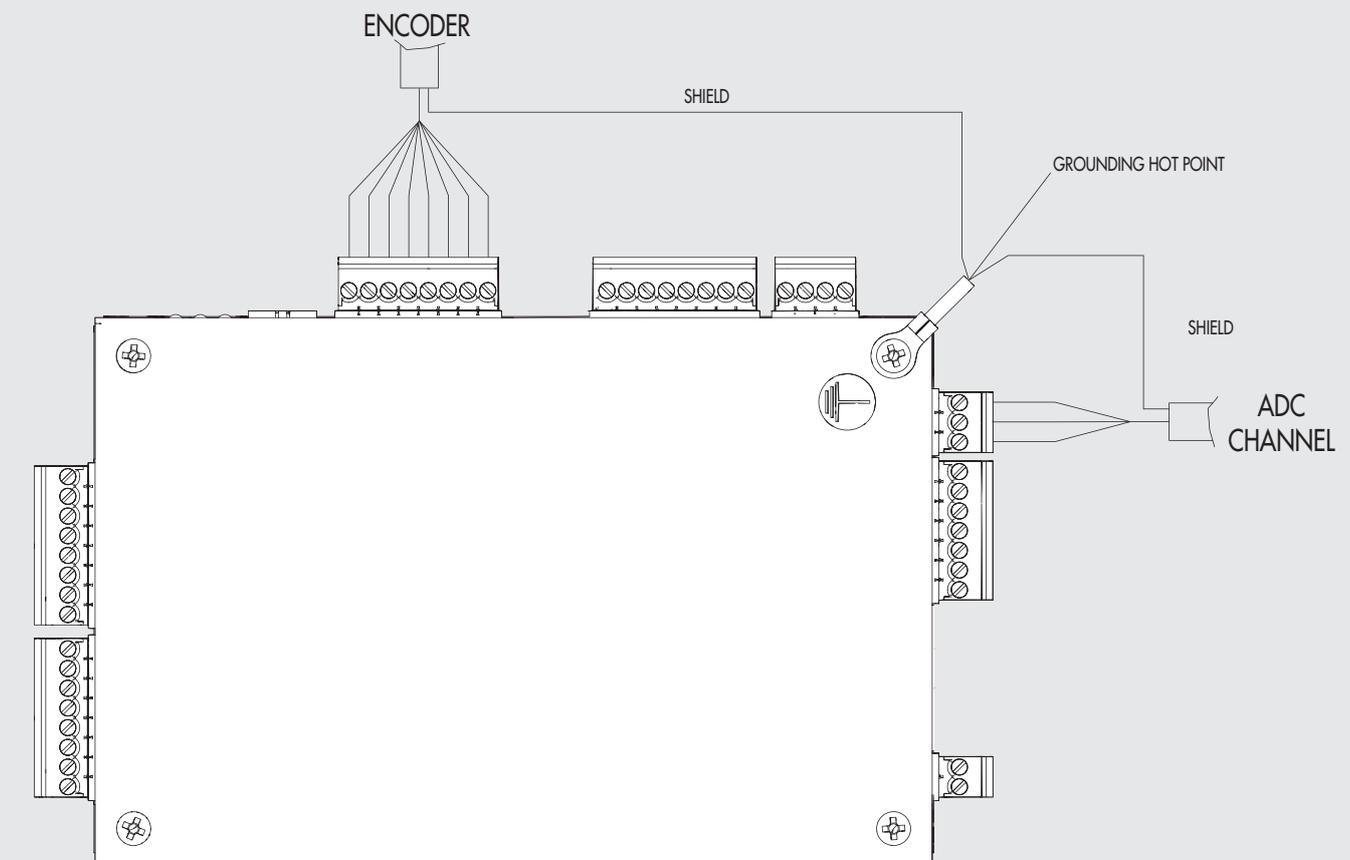
The analogue input is used to acquire voltage signals only in the range of 0 - 10V positive values with the following characteristics:

- 10 bit resolution.
- Circuit time constant = 10  $\mu$ s.
- Input impedance = 6.6 k $\Omega$ .
- The following precautions must be taken when using the analogue input:
  - Use a shielded cable with the shield connected to the grounding hot point provided on the e.drive casing.
  - Take great care when laying the cable, avoid routing it near cables carrying interference signals, such as motor drives, cables with strong currents and/or inductive loads in general.
  - With low input voltage values, typically a few mV, the background noise could generate incorrect readings.
  - In case of disturbed signals, it is appropriate to install the software filters available in the management software.
  - When using these filters, it must be borne in mind that they affect the response speed of the channel. The "heavier" the filter, the lower the system response.
  - With the filter at maximum efficiency, the delay is approximately 200 ms.

## GROUNDING

The e.drive must be grounded by connecting the grounding cable to the casing.

The grounding is to be considered as functional grounding (i.e. one to increase immunity to equipment interference) and not as safety grounding. See figure below.



## CONNECTORS

### C1 ENCODER CONNECTOR

Pin	Cable cross-section	Type of terminal	Description
1	0.25 mm <sup>2</sup>	Screw clamping	5VDC 500 mA
2	0.25 mm <sup>2</sup>	Screw clamping	Phase A + encoder
3	0.25 mm <sup>2</sup>	Screw clamping	Phase A - encoder
4	0.25 mm <sup>2</sup>	Screw clamping	Phase B + encoder
5	0.25 mm <sup>2</sup>	Screw clamping	Phase B - encoder
6	0.25 mm <sup>2</sup>	Screw clamping	Phase Z + encoder
7	0.25 mm <sup>2</sup>	Screw clamping	Phase Z - encoder
8	0.25 mm <sup>2</sup>	Screw clamping	0V

### C2 CONNECTOR

Pin	Cable cross-section	Type of terminal	Description
1	0.25 mm <sup>2</sup>	Screw clamping	Analogue output 0 - 10 VDC 20 mA
2	-	-	Not used
3	-	-	Not used
4	-	-	Not used
5	-	-	Not used
6	0.25 mm <sup>2</sup>	Screw clamping	06 PNP output
7	0.25 mm <sup>2</sup>	Screw clamping	05 PNP output
8	0.25 mm <sup>2</sup>	Screw clamping	Analogue input 2 0 - 10 VDC

### C3 CONNECTOR FOR BRAKE

Pin	Cable cross-section	Type of terminal	Description
1	0.25 mm <sup>2</sup>	Screw clamping	+24VDC for external devices
2	0.25 mm <sup>2</sup>	Screw clamping	7 PNP output (0.6 A)
3	0.25 mm <sup>2</sup>	Screw clamping	Input 1 NPN-PNP programmable
4	0.25 mm <sup>2</sup>	Screw clamping	0V for external devices

### C6 ANALOGUE INPUT CONNECTOR

Pin	Cable cross-section	Type of terminal	Description
1	0.25 mm <sup>2</sup>	Screw clamping	+10VDC 20 mA for external devices
2	0.25 mm <sup>2</sup>	Screw clamping	Analogue input 1 0 - 10VDC
3	0.25 mm <sup>2</sup>	Screw clamping	0V for external devices

### C7 CONNECTOR - OUTPUTS 1-4

Pin	Cable cross-section	Type of terminal	Description
1	0.25 mm <sup>2</sup>	Screw clamping	0V for external devices
2	0.25 mm <sup>2</sup>	Screw clamping	04 PNP output
3	0.25 mm <sup>2</sup>	Screw clamping	03 PNP output
4	0.25 mm <sup>2</sup>	Screw clamping	02 PNP output
5	0.25 mm <sup>2</sup>	Screw clamping	01 PNP output
6	-	-	Not used
7	-	-	Not used

### C8 CONNECTOR - INPUTS 8-13

Pin	Cable cross-section	Type of terminal	Description
1	0.25 mm <sup>2</sup>	Screw clamping	+24VDC for external devices
2	0.25 mm <sup>2</sup>	Screw clamping	Input 08 NPN-PNP programmable
3	0.25 mm <sup>2</sup>	Screw clamping	Input 09 NPN-PNP programmable
4	0.25 mm <sup>2</sup>	Screw clamping	Input 10 NPN-PNP programmable
5	0.25 mm <sup>2</sup>	Screw clamping	Input 11 NPN-PNP programmable
6	0.25 mm <sup>2</sup>	Screw clamping	Input 12 NPN-PNP programmable
7	0.25 mm <sup>2</sup>	Screw clamping	Input 13 NPN-PNP programmable
8	0.25 mm <sup>2</sup>	Screw clamping	0V for external devices

### C10 CONNECTOR FOR INPUTS 1-7

Pin	Cable cross-section	Type of terminal	Description
1	0.25 mm <sup>2</sup>	Screw clamping	+24VDC for external devices
2	0.25 mm <sup>2</sup>	Screw clamping	Input 01 NPN-PNP programmable
3	0.25 mm <sup>2</sup>	Screw clamping	Input 02 NPN-PNP programmable
4	0.25 mm <sup>2</sup>	Screw clamping	Input 03 NPN-PNP programmable
5	0.25 mm <sup>2</sup>	Screw clamping	Input 04 NPN-PNP programmable
6	0.25 mm <sup>2</sup>	Screw clamping	Input 05 NPN-PNP programmable
7	0.25 mm <sup>2</sup>	Screw clamping	Input 06 NPN-PNP programmable
8	0.25 mm <sup>2</sup>	Screw clamping	Input 07 NPN-PNP programmable

### C11 LOGIC SUPPLY CONNECTOR

Pin	Cable cross-section	Type of terminal	Description
1	0.25 mm <sup>2</sup>	Screw clamping	Board power supply +
2	0.25 mm <sup>2</sup>	Screw clamping	Board power supply -

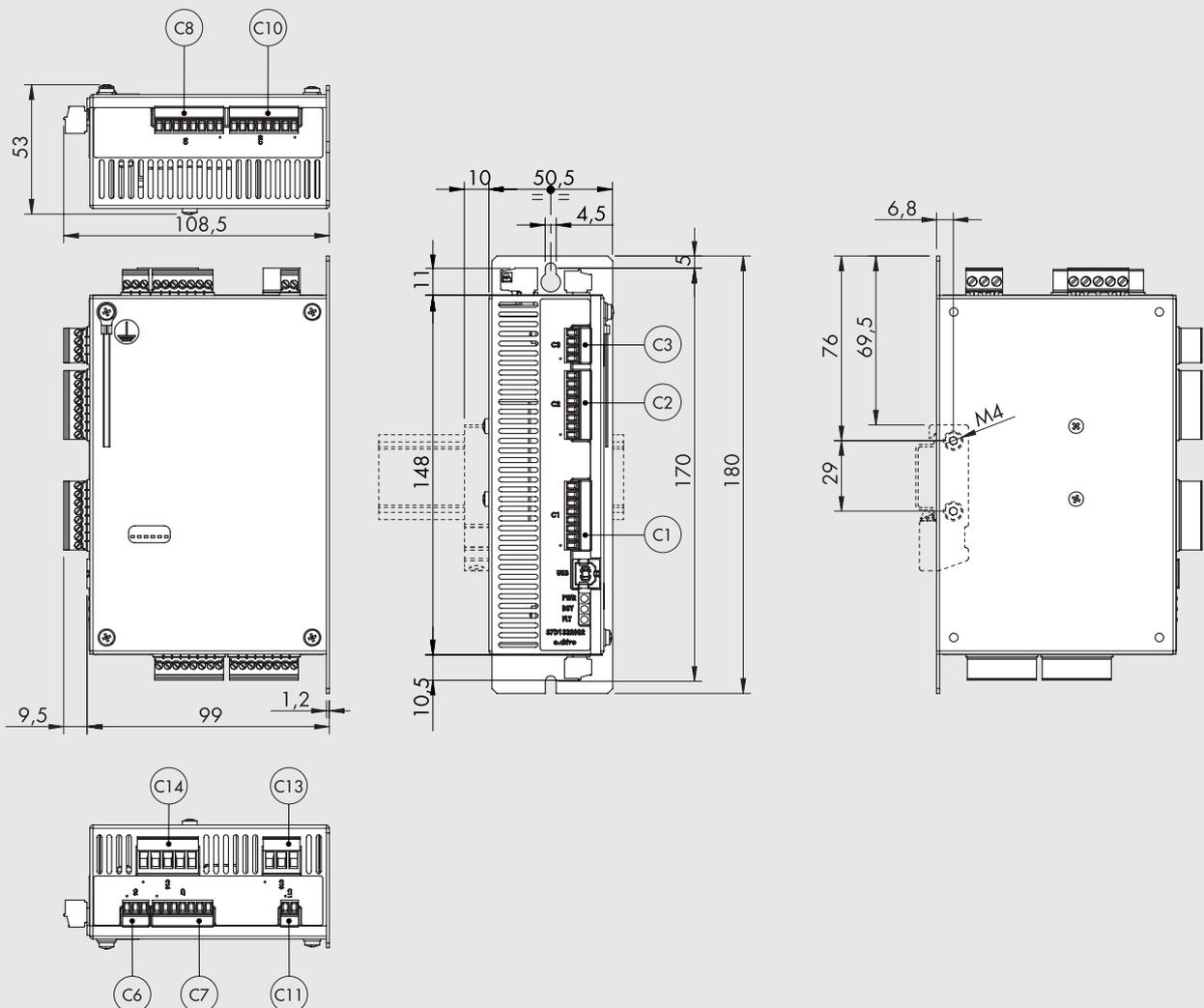
### C13 POWER SUPPLY CONNECTOR

Pin	Cable cross-section	Type of terminal	Description
1	2.5 mm <sup>2</sup>	Screw clamping	SHD
2	2.5 mm <sup>2</sup>	Screw clamping	Drive power supply +
3	2.5 mm <sup>2</sup>	Screw clamping	Drive power supply -

### C14 MOTOR PHASE CONNECTOR

Pin	Cable cross-section	Type of terminal	Description
1	2.5 mm <sup>2</sup>	Screw clamping	A -
2	2.5 mm <sup>2</sup>	Screw clamping	B -
3	2.5 mm <sup>2</sup>	Screw clamping	B +
4	2.5 mm <sup>2</sup>	Screw clamping	A +
5	2.5 mm <sup>2</sup>	Screw clamping	SHD

### MODULE MECHANICAL PROPERTIES



# MWDRIVE SOFTWARE

## GENERAL

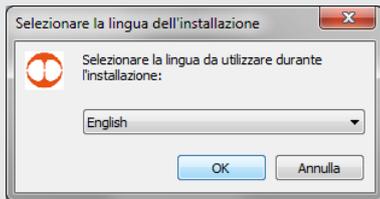
MWDRIVE is the software environment that Metal Work supplies for motion programming. It is compatible with MS Windows-based operating systems (Windows XP up to Windows 10) and has the distinctive feature of combining ease of use with flexibility and operating capacity. MWDRIVE is used to generate a file that is then transferred into the e.motion board to perform the working cycle. It can be downloaded from the Metal Work website, under "Tools & Manuals – Elektro programs", together with the example of configuration with the drives included in the general catalogue and the examples of applications.

## MINIMUM SYSTEM REQUIREMENTS

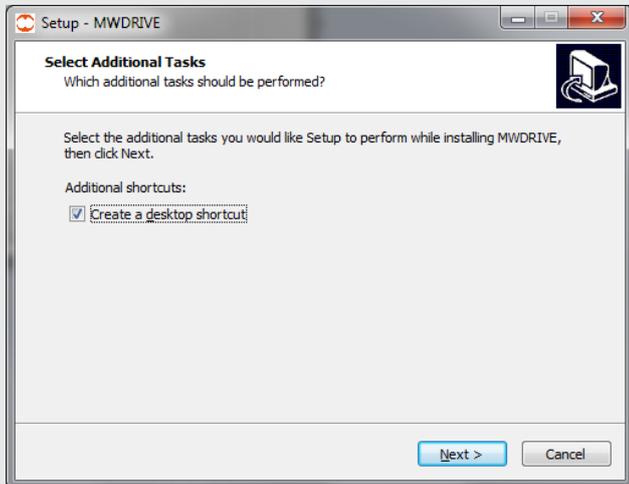
- Microsoft Windows Xp Sp3 or higher
- Microsoft .NET Framework 4.0 ([www.microsoft.com/en-us/download/details.aspx?id=17718](http://www.microsoft.com/en-us/download/details.aspx?id=17718))
- Screen resolution 1024x768
- Driver Stm32 Virtual Com Port ([www.st.com/web/en/catalog/tools/PF257938](http://www.st.com/web/en/catalog/tools/PF257938))

## INSTALLATION

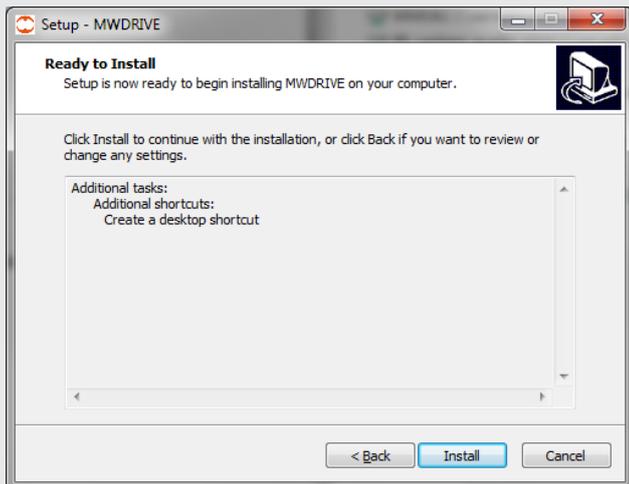
The installation of the software is simple and consists of a few steps. When executing the installation file, you are prompted to select the language to use during both installation and use. The available languages are Italian and English.



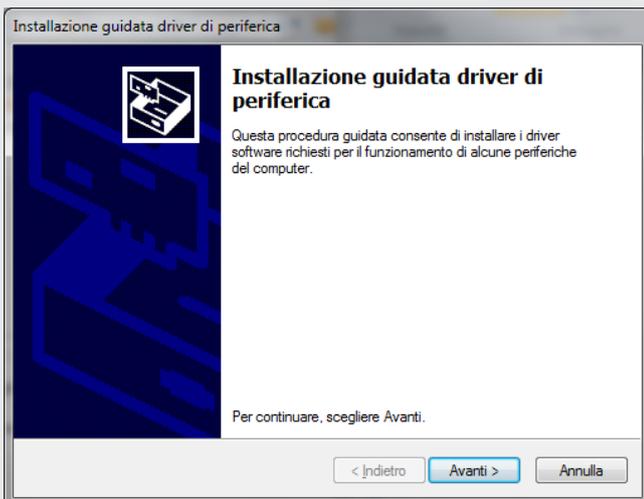
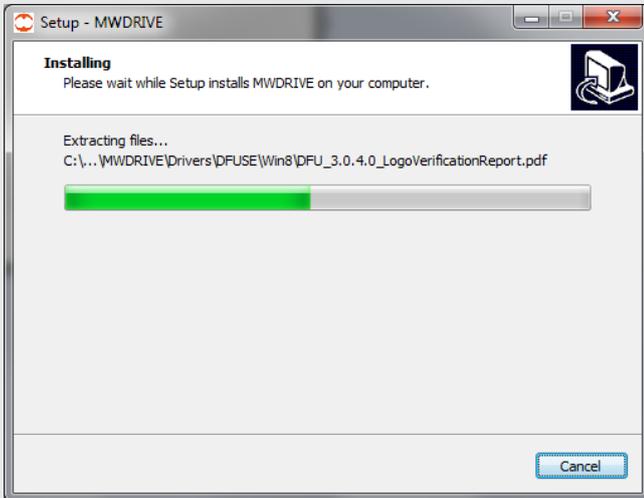
Select the desired language and click "OK".



Then click "Next".



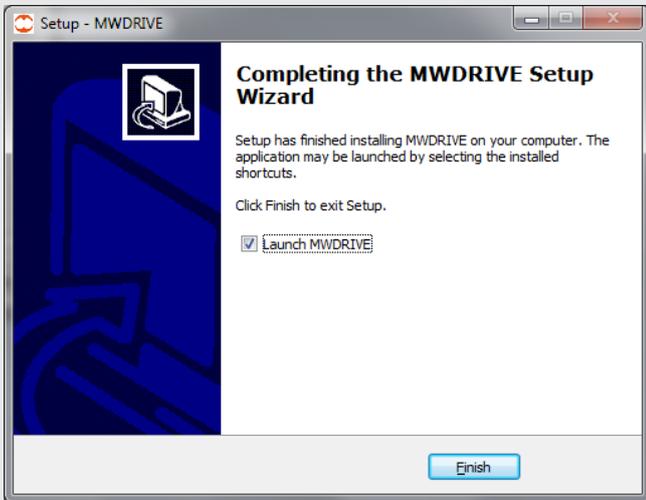
Then click "Install".



Click “Avanti” to install the virtual serial port driver needed to communicate with the device.



Click “Fine” to exit the peripheral installation.



Program installed.  
Tick the "Start MWDRIVE" checkbox to launch the program.

## MAIN MENU

When the program has been launched, the following page is displayed at the start:



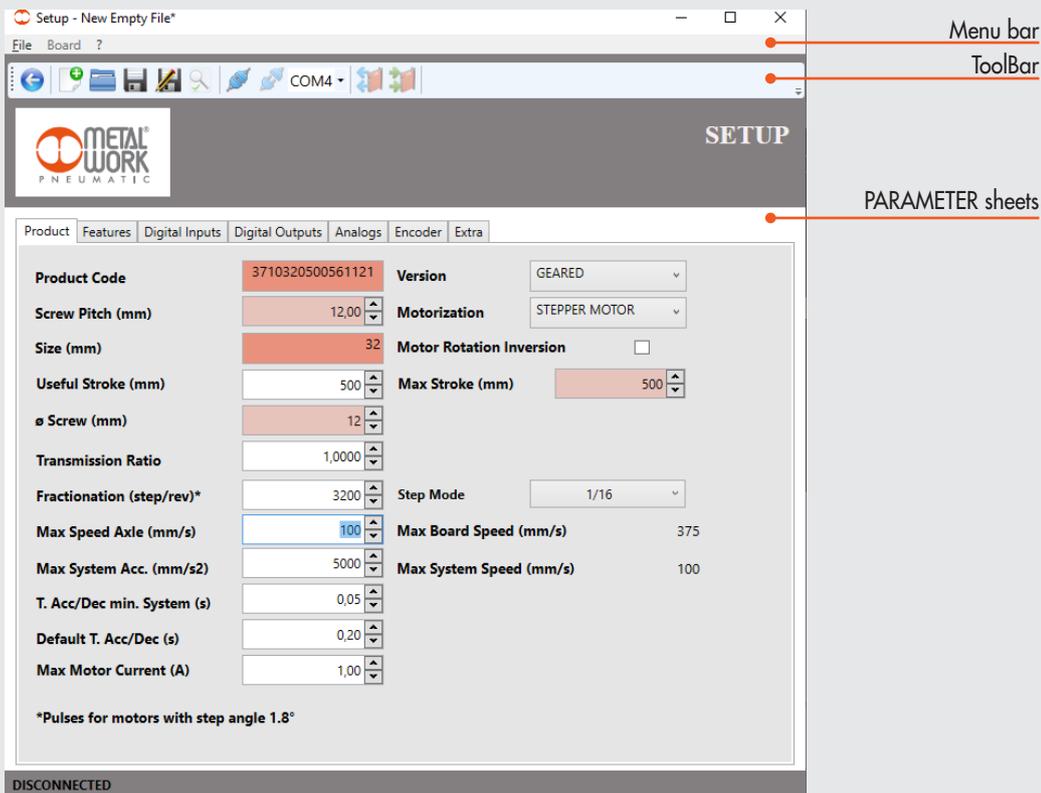
**SETUP:** allows you to set up the actuator, drive and function parameters.  
**PROGRAM:** allows you to create the work program.  
**TEST JOG:** allows you to perform motion tests on the actuator and collect diagnostic information.

## PARAMETER SETUP MENU

### SETUP

The **Setup** window is divided into the following sections.

By entering the Product Code (in the case of a standard product), which is stamped on the electric actuator supplied, the software automatically completes the fields relating to the mechanical characteristics of the actuator (pink fields). The section relating to indexing and drive characteristics are to be completed by the user.



Menu bar

ToolBar

PARAMETER sheets

Product	Features	Digital Inputs	Digital Outputs	Analogs	Encoder	Extra
Product Code	3710320500561121	Version	GEARED			
Screw Pitch (mm)	12,00	Motorization	STEPPER MOTOR			
Size (mm)	32	Motor Rotation Inversion	<input type="checkbox"/>			
Useful Stroke (mm)	500	Max Stroke (mm)	500			
ø Screw (mm)	12					
Transmission Ratio	1,0000	Step Mode	1/16			
Fractionation (step/rev)*	3200	Max Board Speed (mm/s)	375			
Max Speed Axle (mm/s)	100	Max System Speed (mm/s)	100			
Max System Acc. (mm/s <sup>2</sup> )	5000					
T. Acc/Dec min. System (s)	0,05					
Default T. Acc/Dec (s)	0,20					
Max Motor Current (A)	1,00					

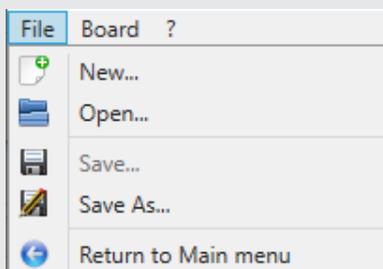
\*Pulses for motors with step angle 1.8°

DISCONNECTED

If there is no standard or special product code, the fields can be customised by the user.

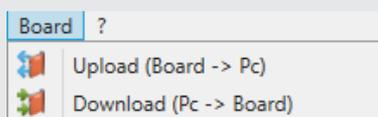
### MENU BAR

The menu bar can be used to perform some operations on the parameter file.



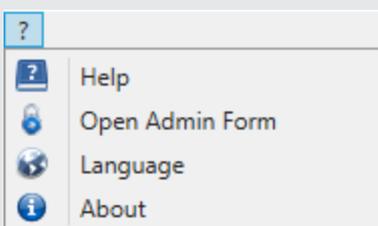
File	Board	?

The **"File"** menu allows you to create, open and save the parameter file. You can return to the home page by selecting "Return to main menu".



Board	?

The **"Board"** menu can be selected if there is a connection to the e.drive board. It allows you to upload or download the parameter file.

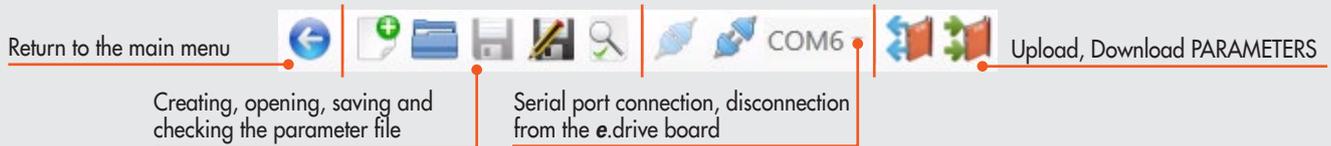


?

The **"?"** menu allows you to view the manual, update the board firmware, change the PC software language and view MWDRIIVE software version information.

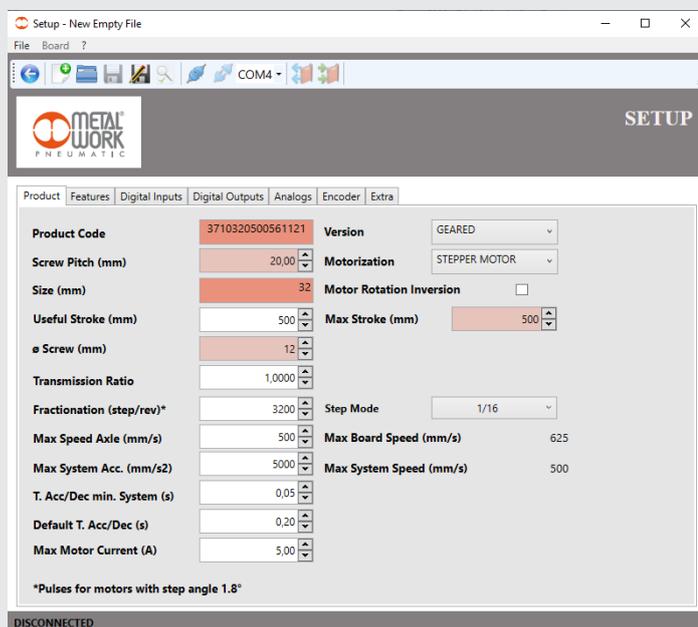
## TOOLBAR

The ToolBar is divided into several sections and can be used to perform some operations that can also be accessed from the Menu Bar. Some icons can only be selected under certain conditions (such as online mode and debugging in progress).



## PRODUCT DATA SHEET

The parameters are divided into several sections that group together individual functions.



The Product data sheet contains actuator, motor and drive setup parameters.

Name	Field description
Product code	Actuator code. If standard, some fields (coloured differently) are completed automatically
Screw pitch / Primitive (mm)	Axis feed per motor rev with gear ratio 1 : 1
Size (mm)	Size of actuator (info box)
Useful stroke (mm)	Usable actuator stroke: useful stroke < maximum stroke
Max stroke (mm)	Total actuator stroke
Screw ø (mm)	Nominal screw diameter (info box), not present if orthogonal motor
Max motor current (A)	Maximum current that can be supplied to the motor. The maximum absolute value is 6A
Transmission ratio	Gear ratio between drive shaft and actuator input shaft. Decimal number (es. 0.3333 - 1.0000)
Resolution (step/rev)	Number of pulses per revolution: automatically set according to the selected microstepping mode.
Step Mode	Microstepping mode to be used. Full Step, Half Step, 1/4, 1/8, 1/16 step.
Max speed axle (mm/s)	Maximum speed reachable by the axis: see mechanical properties in the actuator catalogue
Max board speed (mm/ s)	Maximum board speed: it depends on the actuator characteristics and the indexing
Max system acc. (mm/s²)	Maximum axis acceleration: see mechanical properties in the actuator catalogue
Max system speed (mm/s)	Maximum speed reachable by the system: minimum between axis max. speed and board max. speed
T. acc/dec min system (s)	Minimum acceleration/deceleration time for the system: it defines the minimum usable time
Default T. acc/dec (s)	Acceleration/deceleration time proposed as default
Version	Motor version (in line, geared, orthogonal)
Motorization	Type of motor used (STEPPER, BRUSHLESS)
Motor rotation inversion	Reverses the direction of rotation of the motor

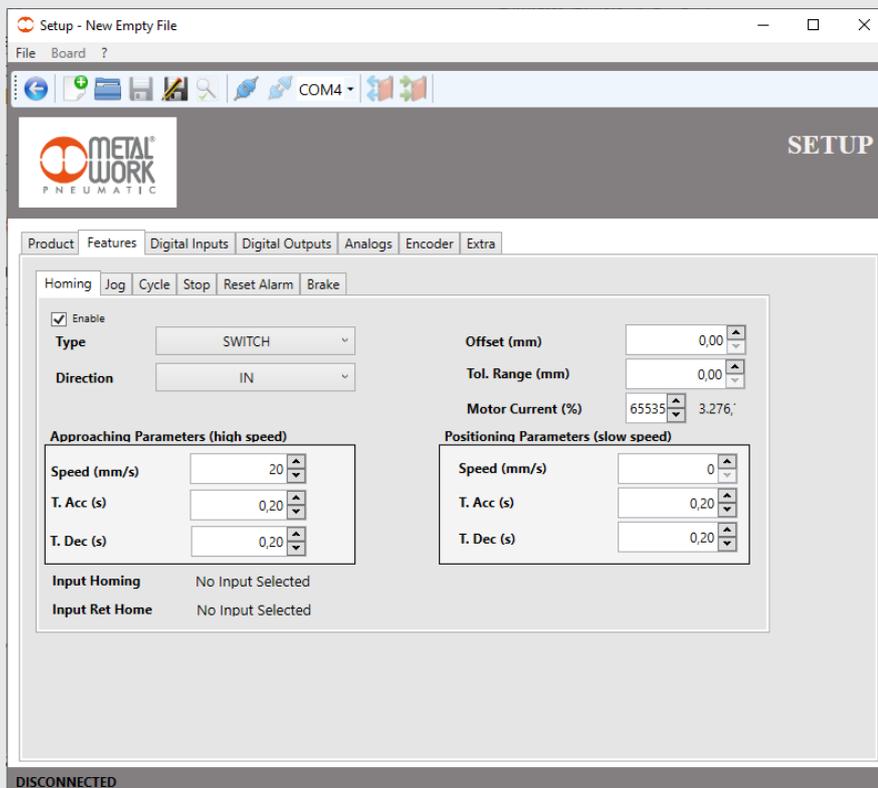
**N.B.:** The decimal separator in numeric fields depends on the international settings of the PC used.

## FEATURES

The “Features” page is used to enable and set up some characteristics and settings that can be assigned to the board inputs/outputs.

### Homing

The “Homing” function is used to enable and configure the zero search, i.e. the point that generally corresponds to the occurrence of an event after which the position is set to 0 mm. Zero search generally takes place on system start-up, before starting the work cycle. In some cases, it is used to carry out a step-loss test.



Zeroing can be performed in different ways: MECHANICAL STOP, LIMIT SWITCH, MECHANICAL STOP+ENCODER, LIMIT SWITCH+ENCODER.

**MECHANICAL STOP:** the actuator moves in the specified direction (IN or OUT) at a settable speed, which is usually quite high, until it reaches an end of the axis and goes against the mechanical stop, which is known as “mechanical stop”.

**LIMIT SWITCH:** the actuator moves in the specified direction (IN or OUT) at a settable speed, until the limit switch sensor on the actuator activates. At this point, it reverses its direction and moves again at a settable speed, which is usually lower than the previous speed, until the sensor deactivates.

**MECHANICAL STOP+ENCODER:** The axis follows the same procedure as the one for stop under homing. It then reverses the direction of movement and slowly moves to search the encoder index pulse.

**LIMIT SWITCH+ENCODER:** the axis follows the same procedure as the one for limit switch under homing. It then reverses the direction of movement and slowly moves to encoder index pulse.

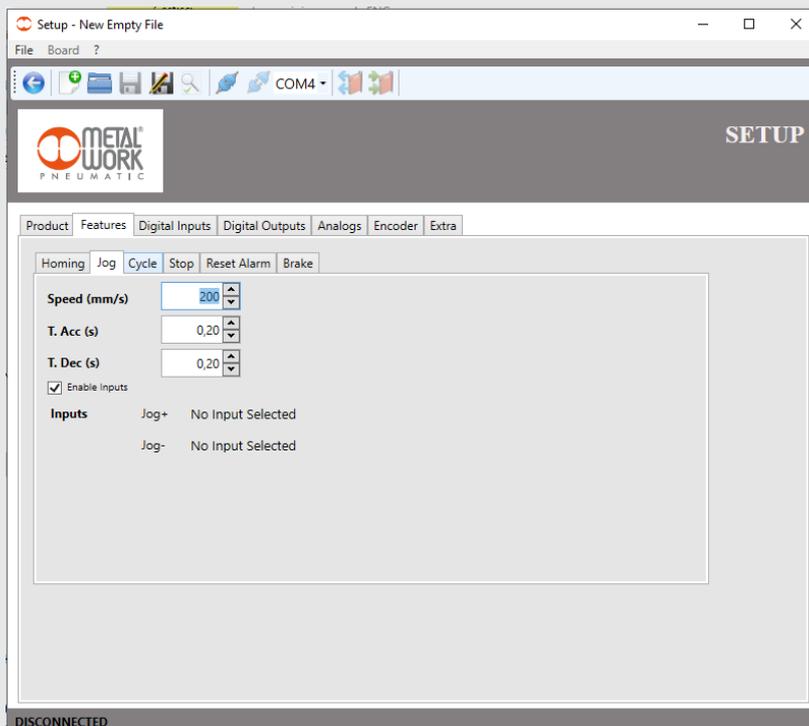
The parameters referring both to the approach speed (high speed) and the positioning speed (low speed) during zeroing can be set by the user, together with the relevant decelerations and accelerations. impostabili dall'utente unitamente alle relative decelerazioni e accelerazioni.

**IMPORTANT!** To activate the **MECHANICAL STOP + ENCODER** and **LIMIT SWITCH + ENCODER** mode, an Encoder with Index must be enabled in the Encoder page.

Other parameters displayed in this page:

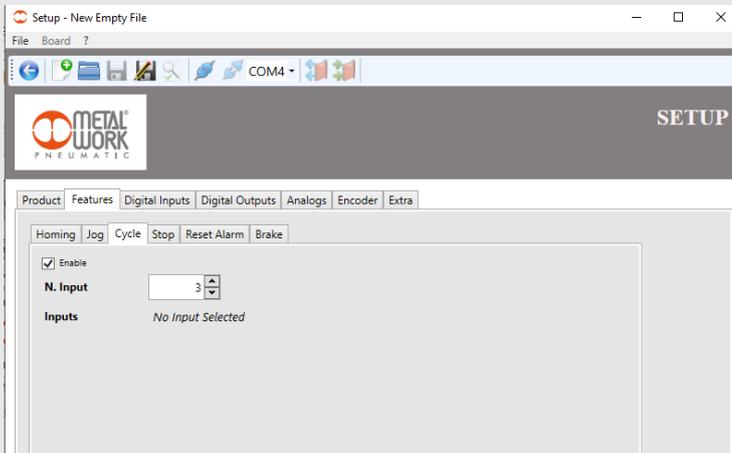
Type	Zeroing mode
<b>Direction</b>	Motor direction IN or OUT
<b>Offset</b>	Offset (mm) to be reached after zero searching. The permissible stroke reduces consequently.
<b>Tolerance</b>	Tolerance (mm) for the calculation of the step loss during homing. It is disabled if you set 0.00 mm. If enabled, the control works regardless of whether there is an encoder connected and/or enabled. In fact this control doesn't consider pulses coming from the encoder, it whereas makes a comparison between pulses supplied during actual homing procedure with the ones supplied during the previous zeroing. It's easy to understand that this control takes sense from the second procedure performed without switching off the e.drive control board. This control is particularly useful in open loop systems, typically with motors without encoders.
<b>Input HOMING</b>	It's the digital input where to connect the signal coming from the micro switch if you want to perform the homing procedure SWITCH or SWITCH + ENCODER.
<b>Input RET HOME</b>	It's the digital input provided to let the e.drive execute the homing procedure in automatic mode without the cycle execution. The homing procedure is performed as defined in feature tab.
<b>Motor current (%)</b>	Sets the percentage of current to be used during zeroing. It must be noted that in STEPPER motors the torque is inversely proportional to the speed. For this reason, it is advisable to reduce the current during zeroing. The theoretical calculation may not be complied with due to mechanical fractions, acceleration ramps, etc. It is advisable to find the optimal value by proceeding by trial and error.

## Jog



The "Jog" function can be used to set speed, acceleration and deceleration parameters relating to the jog motion of the actuator. The jog mode is used to allow the manual speed movement of the actuator, without position control. It is generally used during the first commissioning stages or to check the operation of the actuator following critical operating conditions. There's also the possibility to assign two digital inputs to be used as Jog+ or Jog-. Refer to the "Digital input" sheet for the setting of these inputs.

## Cycle



The “**Cycle**” function is used to set the number of digital inputs for the selection of the cycle to run. Refer to the programming menu below to fill in the cycle fields and to the “**Digital input**” function for the definition of digital inputs.

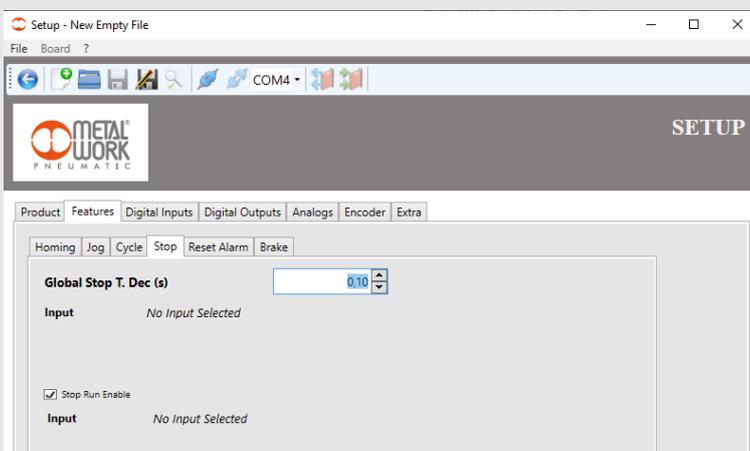
The number of executable cycles is equal to 2 inputs, and the cycle is selected using the various possible combinations according to binary logics. The cycle to run must be selected prior to the START of the program.

For example: using “**Input no.**” 2, four cycles can be set, as shown in the table below.

SEL2, SEL1	# Cycle
0,0	Cycle 0
0,1	Cycle 1
1,0	Cycle 2
1,1	Cycle 3

**! IMPORTANT!** If this function is disabled, only “Cycle 0” can be run, which corresponds to the condition in which all the digital inputs for cycle selection are in logic state 0.

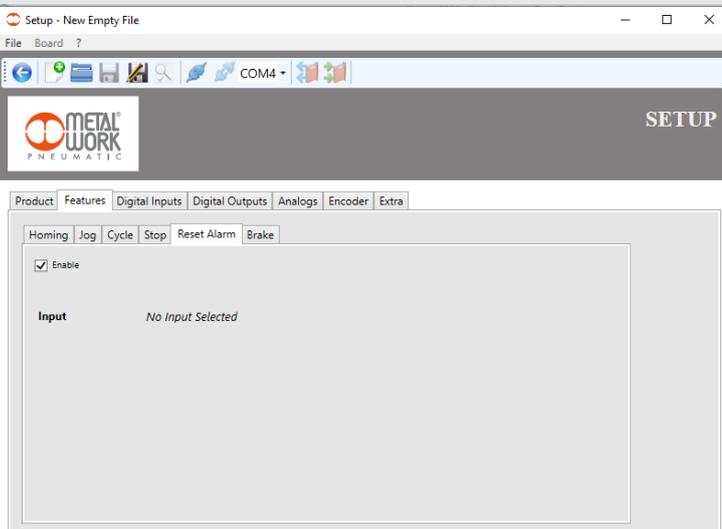
## Stop



The “**Stop**” function is used to set the deceleration time when you wish to stop the cycle before its natural end. It is also possible to enable an additional stop mode, called Run Stop. Refer to the “**Digital input**” function for the definition of the relevant input.

Type	Description
Global Stop	“Sudden stop”: the motor stops at the set deceleration time. Run Homing to resume operation.
Run Stop	“Scheduled stop”: execution of current instruction in the cycle is completed and the system remains on hold. On the next restart, the system restarts the cycle from the next instruction.

## Reset Alarm

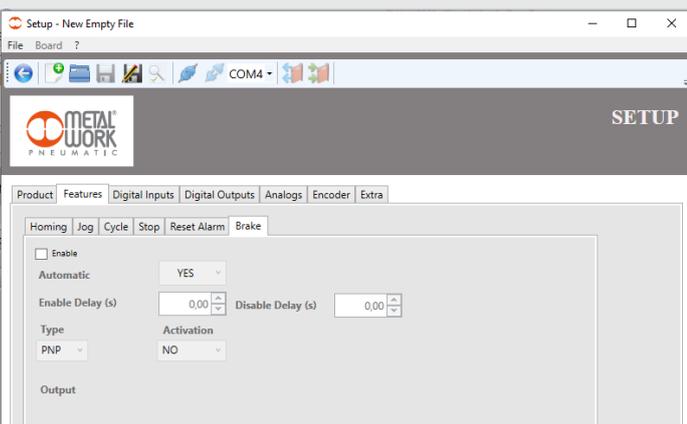


The “**Reset Alarm**” function is used to enable a digital input for use as reset from an alarm condition. Refer to the “**Digital Input**” function for the definition of this input.

Most drives come with an output to indicate when a condition (generally called ALARM) that can generate an alarm has occurred in the system. This condition disables any movement of the motor.

A typical application envisages the connection of a button to the digital input of the e.drive.

## Brake



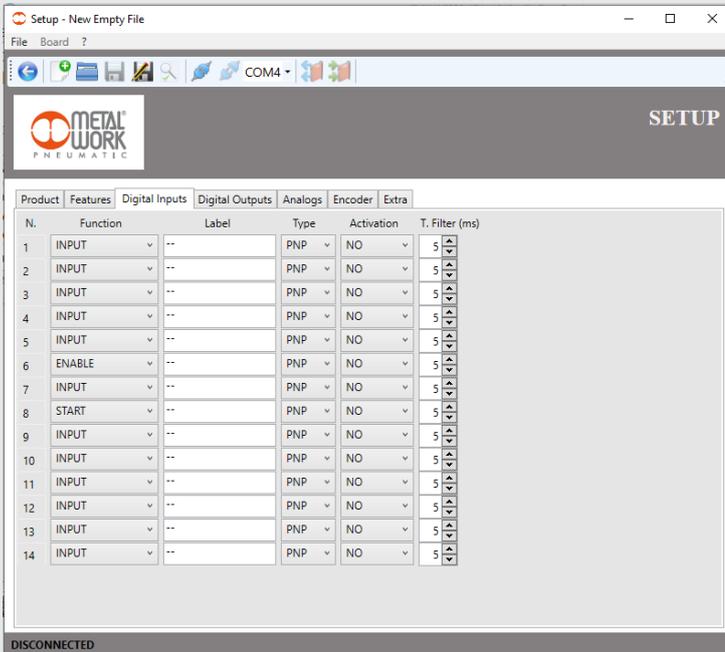
The **Brake** function is used to enable control of any holding brake installed on the motor used.

Parameter	Description
<b>Enables</b>	Enables/Disables automatic brake control
<b>Energ. delay T (s)</b>	Brake energising delay
<b>De-energ. delay T (s)</b>	Brake de-energising delay
<b>Type</b>	Type of PNP brake output
<b>Activation</b>	NO/NC brake output activation

**⚠ IMPORTANT!** The brake must be connected to the dedicated OUT07 digital output, in consideration of the greater amount of current supply required to ensure correct operation of the holding brake.

If the brake function is disabled, the digital output can be configured for generic functions.

## DIGITAL INPUTS



The “**Digital Input**” function can be used to configure digital inputs for connection to the board using connectors C3, C8 and C10. Each input has the following configurable features:

Type	Description
<b>No.</b>	Input identification number
<b>Function</b>	Input-associable function
<b>Comment</b>	Description of the input
<b>Type</b>	Type of PNP, NPN input
<b>Activation</b>	NO, NC activation (normally open, normally closed)
<b>Filter</b>	Filter in ms to filter out any interference

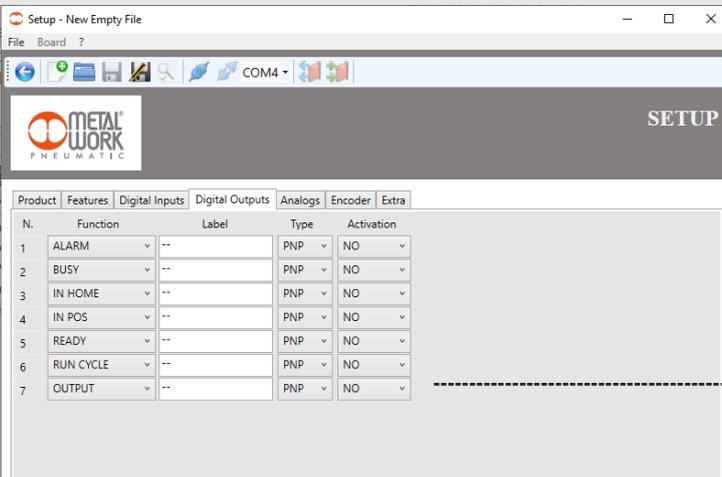
**⚠ IMPORTANT!** Once the configuration has been transferred to the board, the comments are only saved on file using the standard save procedure provided by the operating system.

Each digital input can be associated to a “**Function**”. Following activation of the selected input, the corresponding function is called up. A generic function, called “**INPUT**”, is also provided in addition to the standard functions required by a motion controller. This input is considered as an ON/OFF switch to be used when running a cycle as a simple test, as a condition for accessing or exiting cyclical nested instructions or as a “**trigger**” to run a particular cycle sequence.

Other functions are available depending on the operations that are enabled:

Function	Description
<b>ENABLE</b>	Enables the system by providing torque to the motor
<b>ALARM</b>	Alarm input
<b>HOMING</b>	Zero switch signal
<b>JOG-</b>	Performs Jog- motion (MANUAL ON input required)
<b>JOG+</b>	Performs Jog+ motion (MANUAL ON input required)
<b>MANUAL</b>	Enables the manual mode to perform jog commands
<b>RST ALARM</b>	Resets internal and external alarms
<b>SEL1, SEL2, SELn</b>	Cycle selection input (binary combination – see Cycle under FEATURES)
<b>START</b>	Starts a selected cycle
<b>STOP GLOBAL</b>	Executes a global stop (see Stop under FEATURES)
<b>STOP RUN</b>	Stops current instruction without completing the cycle (see Stop under FEATURES)
<b>INPUT</b>	Free generic inputs available
<b>RET HOME</b>	Executes the homing cycle defined in homing feature

## DIGITAL OUTPUTS



→ "Special" 600 mA output to control the holding brake.

The "Digital Output" function can be used to configure digital outputs connectable to the board using connectors C2, C3, C7 and C9. Each output has the following configurable features:

Type	Description
No.	Output identification number
Function	Output-associable function
Comment	Description of the output
Type	Type of PNP output
Actuation	NO, NC actuation (normally open, normally closed)

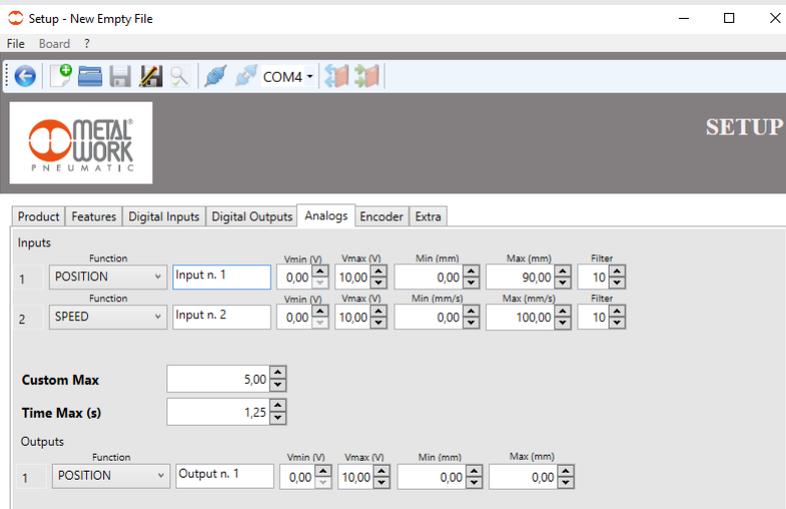
**IMPORTANT!** Once the configuration has been transferred to the board, the comments are only saved on file using the standard save procedure provided by the operating system.

Each digital output can be associated to a "Function". Following activation of the selected input, the corresponding function is called up. A generic function, called "OUTPUT", is also provided in addition to the standard functions required by a motion controller. This output is considered as an ON/OFF switch to be used when running a cycle as a simple test, as a condition for accessing or exiting cyclical nested instructions or as a "trigger" to run a particular cycle sequence.

Other functions are available depending on the operations enabled:

Functions	Description
BUSY	The system is enabled
ALARM	Alarm output (generally towards a logic device)
IN HOME	Zeroing executed correctly
IN POS	Following a movement, it performs a consistency check of the actual position achieved
READY	The system is ready to perform the axis movement
RUN CYCLE	The cycle selected with the selection inputs (see DIGITAL INPUTS) is in progress

## ANALOGUE BOARDS



The “Analogue” board is used to configure the two analogue inputs and the analogue output on the e.drive. Each input only accepts voltage signals (0-10VDC) and has the following configurable features:

Type	Description
<b>ID</b>	Analogue input identification number
<b>Function</b>	Physical variable to be associated to the input signal
<b>Comment</b>	Generic description
<b>Vmin (V)</b>	Minimum input voltage
<b>Vmax (V)</b>	Maximum input voltage
<b>Filter</b>	Input filter to avoid interference (the greater the filter, the greater the acquisition delay)
<b>Min (meas. unit)*</b>	Minimum value related to the selected function
<b>Max (meas. unit)*</b>	Maximum value related to the selected function

\***(meas. unit)** measurement unit related to the selected function, then: POSITION (mm) / SPEED (mm/s); TIME (s); CUSTOM (- -)

**⚠ IMPORTANT!** Once the configuration has been transferred to the board, the comments are only saved on file using the standard save procedure provided by the operating system.

The physical variable associated to the analogue input variation can be selected between POSITION, SPEED, TIME, and a CUSTOM one. Function choice is due to the hardware physically connected and to user needs.

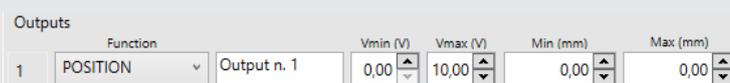
**POSITION** and **SPEED** perform a real-time control of the movement and they must be used with the MOVE\_CNT instruction. See page 66 current manual for a usage example of this two features.

**TIME:** it's the maximum time settable for using instruction WAIT\_TIME. See instruction details for a usage example.

**CUSTOM:** generic function selectable. Through the TEST\_AIN instruction it's possible retrieving information coming from the hardware connected to the selected analogue input. See instruction detail for a usage example.

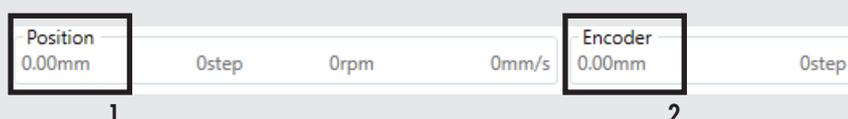
### ANALOGUE OUTPUT

The analogue output on this board at pin 1 of the C2 acts as an analogue position transducer incorporated with the following specifications: 0-10VDC

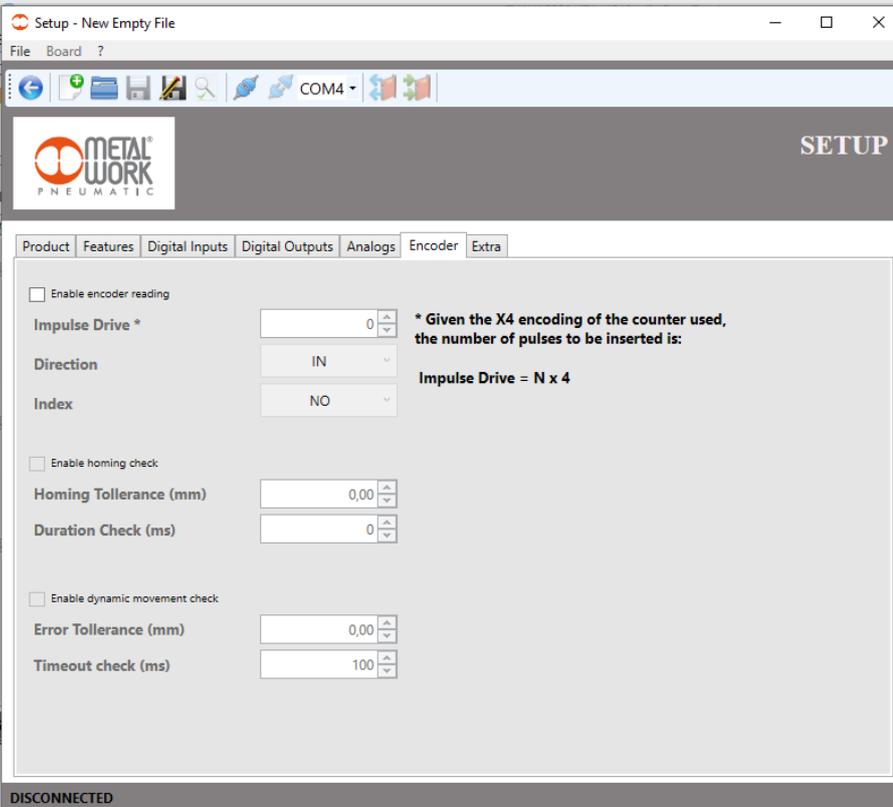


The Product board contains the Useful Stroke (mm) field. The operation of the analogue output envisages two different operating modes.

1. If a simple stepper motor without an encoder is connected, the analogue output will resume the calculated theoretical position. This means that, if the motor misses the step, the system is likely to be in disagreement with the actual position of the motor and hence the position of the cylinder.
2. If a stepper motor is connected or enabled with an encoder or an optional encoder is mounted on the mechanical part but it is correctly configured in the encoder board, the analogue output will send the actual position read by the encoder. In the event of a mechanical blockage or a missed step, the actual position of the motor or the cylinder will be determined.



## ENCODER



Whenever the system is supplied with a rotary position transducer, named Encoder, is possible to use it simply to retrieve real – time information about position, or to check the position itself. The picture below shows how to refer to position in Test/Jog menu.



**POSITION:** is the commanded position and is calculated by the movement instruction.

**ENCODER:** position read by angular transducer usually provided directly on the motor (brushless) or externally applied on the motor rotary shaft (stepper).

### ENABLE ENCODER READING

Type	Description
<b>Impulse Drive</b>	Encoder pulses number for every motor revolution. *Given the X4 encoding (multiplying factor) of the counter used, the pulses' total number to insert is: Encoder pulses = N x 4. N is the pulses' number on encoder datasheet.
<b>Direction</b>	Motor direction (IN/OUT). Pulses' counting and transducing in mm/revolution is performer clockwise IN or counter clockwise OUT compliance with motor shaft rotation.
<b>Index</b>	Z encoder zero pulses (YES / NO). If wired can be used during homing procedure both with mechanical stop and with micro. Once ended homing procedure as defi ned in Setup Features a positive movement being executed until the Z pulse is detected. That's the new origin rotation system that will rotate in agreement with zero motor shaft.

### ENABLE HOMING CHECK

It's a position control performed through the encoder, expressed in mm, made when the motor is not moving and only when the homing procedure is performing. That procedure ends well only if this control is passed when the duration check time lapsed. Consequently the "state flag" HOME switches ON and is acquired the rights for executing an Automatic movement. Otherwise the system detects an error and goes into an alarm condition signaled by "position error" alarm.

When the motor is stopped and after position reference has been zeroed, once 200 ms elapsed if Encoder position is  $> 0.06$  mm or  $< -0.06$  mm an error is pointed out, otherwise the homing procedure ends well.

Type	Description
<b>Homing Tolerance (mm)</b>	Tolerance for position control. Given by the subtraction between position commanded and position read. Tolerance (mm) = Position (mm) – Encoder (mm)
<b>Duration Check (ms)</b>	During of tolerance control.

### ENABLE DYNAMIC MOVEMENT CHECK

It's a position control performed through the encoder, expressed in mm, made only during a motor movement, except for the homing procedure. Once any movement instruction is given, this control checks that the difference between commanded position and real – time position does not go over the edge for more than the timeout. Then a "position error" alarm is activated, otherwise if the error over the edge defined by the tolerance lasts less than the timeout the movement ends without any problem.

\*\*Due to delay of movement introduced by brushless motor drives is strictly suggested not to use this control with brushless motors.

When the motor starts moving the cyclic control performs a constant monitoring that the difference between the two positions does not stay over 0.2 mm for more than 100 ms.

This control is not functioning during homing procedure. di anomalia segnalata dall'allarme "position error".

Type	Description
<b>Error tolerance (mm)</b>	Tolerance for position control. Given by the subtraction between position commanded and position read. Tolerance (mm) = Position (mm) – Encoder (mm)
<b>Timeout check (ms)</b>	Timeout of tolerance cyclic control.

### EXTRA

Product | Features | Digital Inputs | Digital Outputs | Analogs | Encoder | Extra

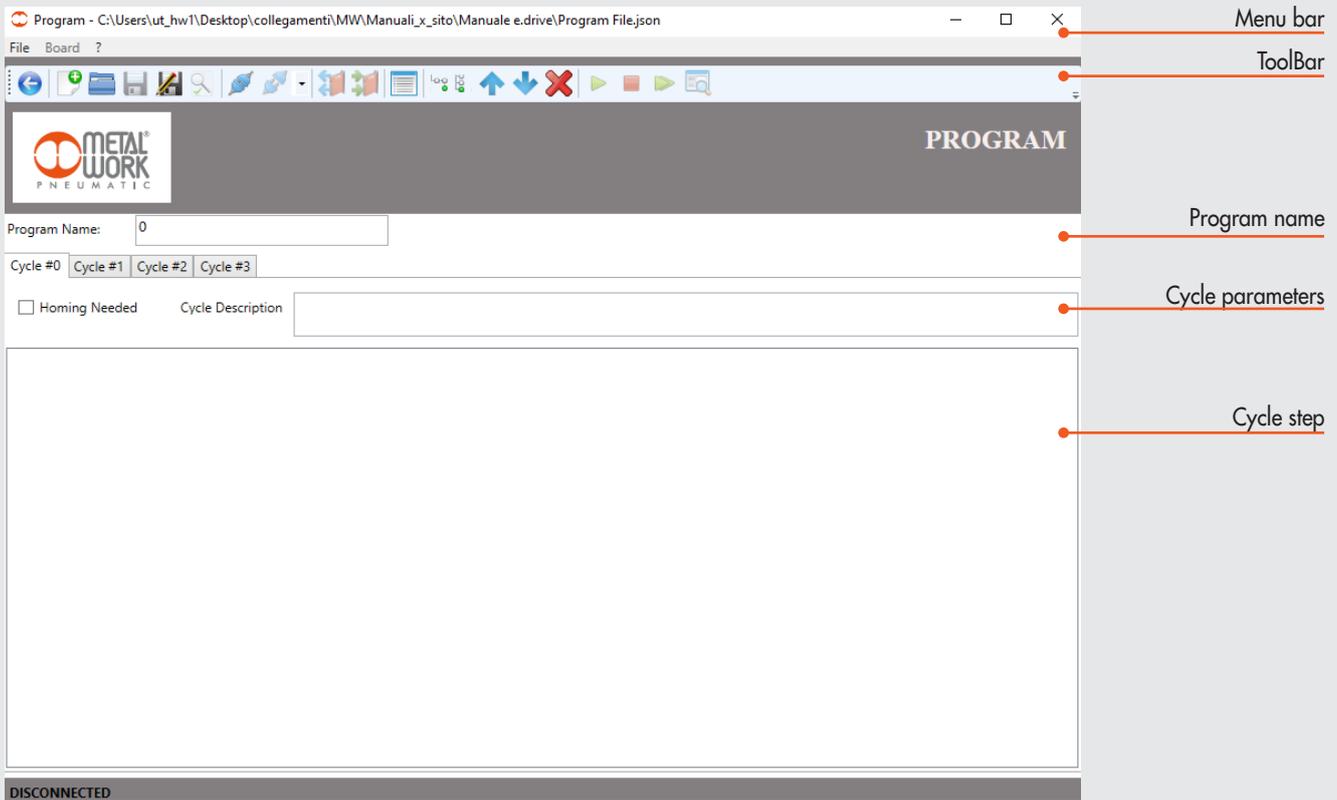
<b>Motor Enable Delay (s)</b>	<input type="text" value="0,20"/>
<b>Motor Inversion Delay (s)</b>	<input type="text" value="0,20"/>
<b>Stop Jump (s)</b>	<input type="text" value="0,10"/>
<b>PowerFail Stop Time (s)</b>	<input type="text" value="0,10"/>

The "Extra" function page contains some additional settings that allow greater operating flexibility of the entire system.

Type	Description
<b>Motor enable delay (s)</b>	Time after which the motor is enabled
<b>Motor reverse delay (s)</b>	Time after which the motor reverses its direction of rotation
<b>Stop Jump (s)</b>	Time when using the JUMP function during the cycle
<b>Stop PowerFail (s)</b>	Time allowed to the motor to stop in the event of PowerFail

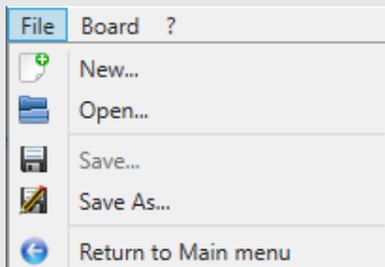
# PROGRAM MENU

The PROGRAM menu can be used to create and configure the work program. The work program consists of a series of cycles, each comprising commands, called Steps. A step can be a motion control, a test control, a digital input control, a repeat control, etc. The Program window is divided into the following sections:

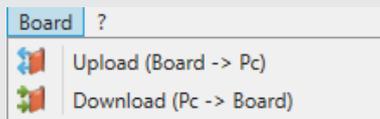


## MENU BAR

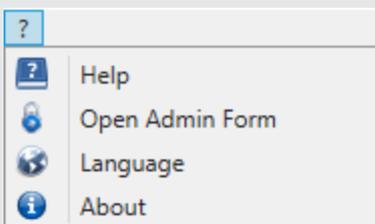
The menu bar allows you to execute some operations on the program file.



The "File" lets you create, open and save the program file. The item "Return to main menu" lets you return to the main display.



The "Board" menu can be selected if a connection is active with the e.drive board. It lets you upload the program currently present on the board or download the program.



The "?" menu lets you display the manual, through admin form lets you update the firmware or download data to the board, exchange the MWDRIVE software language and display current software version data.

## TOOLBAR

The **ToolBar** is divided into subsections and can be used to perform operations accessible from the Menu Bar as well. Some icons can only be selected under certain conditions (online mode, debugging in progress, etc.)



## PROGRAM NAME

You can assign a name of maximum 30 characters to the program created.

Program Name:

## CYCLE PARAMETERS

The program consists of cycles. In the "Cycle Description" zone, you can add a brief comment regarding the cycle.

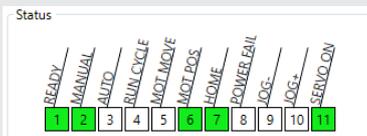
**IMPORTANT!** Once the configuration has been transferred to the board, the comments are only saved on file using the standard save procedure provided by the operating system.

Cycle #0	Cycle #1	Cycle #2	Cycle #3	Cycle #4	Cycle #5	Cycle #6	Cycle #7
<input type="checkbox"/> Homing Needed	Cycle Description Performs homing, advances 50mm, waits 1 second and steps back 50mm						

A convention commonly used in the practice consists in defining the first cycles as the "Homing cycle" or "Zeroing cycle".

We suggest setting the 0 cycle as follows:

- the "Homing needed" box unchecked;
  - the cycle containing only the MOVE-HOME instruction forces zeroing according to the mode set in the Homing tab described in the Setup menu.
- In this way, there being no input for the selection of the activated cycle, the motor will execute zeroing on first cycle start. Upon completion of the zeroing procedure, the HOME condition is valid.



This condition is a "software state" that makes it possible to execute any other cycle with the "Homing needed" box checked. In the absence of this condition, the cycle with this check box ticked can't be executed.

## CYCLE STEP

The Cycle Step zone contains the steps of the cycle to be run. Execution is the Top-Down type, i.e. the execution of steps takes place from top to bottom. There are some conditional steps, which are only executed if a logic test has been passed. Conditional jumps between different sequences within the same program can also be executed. See the section titled Step List for details of all the steps available and their configuration.

<b>SET_HOME</b>	Label									
<b>WAIT_TIME</b>	Time (s)	Label								
	1,00									
<b>MOVE_REL</b>	Position (mm)	Tot. time(s)	Speed (mm/s)	T. Acc (s)	T. Dec (s)	Current (A)	Smooth	Label		
	20,00	1,00	25,00	0,20	0,20	5,00	0			
<b>WAIT_TIME</b>	Time (s)	Label								
	1,00									
<b>MOVE_REL</b>	Position (mm)	Tot. time(s)	Speed (mm/s)	T. Acc (s)	T. Dec (s)	Current (A)	Smooth	Label		
	-20,00	1,00	25,00	0,20	0,20	5,00	0			
<b>RESTART</b>	Label									

## CREATING A PROGRAM

To create a new work program you can either select the File menu -> New File or use the shortcut button in the ToolBar.



A new empty program will be created with a number of cycles corresponding to 2<sup>n</sup> digital inputs defined during the configuration of parameters relative to the Cycle function.

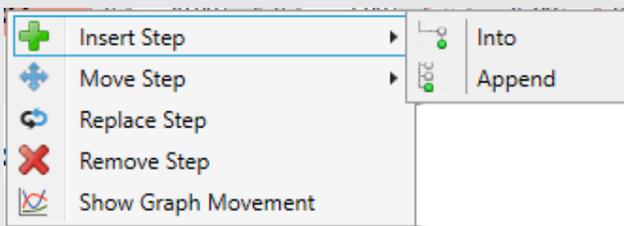
## ENTERING SIMPLE STEPS

There are two ways to add a step:

1. Select either the "Insert Step - Into" or the "Insert Step - Append" button using the toolbar.



2. Click the right mouse button in the Cycle Step section and select Insert Step -> Into or Insert Step -> Append



The difference between "Insert Step - Into" and "Insert Step - Append" is the possibility of entering nested steps. This is useful, for instance, when you wish to execute a motion step and simultaneously run a test on an input; or when conditional jump steps are used.

For further details on nested steps, please refer to the section titled "Enter Nested Step".

**N.B.:** The entry of the first step will always be the Append type, as there are no further steps present, irrespective of the type of entry selected.

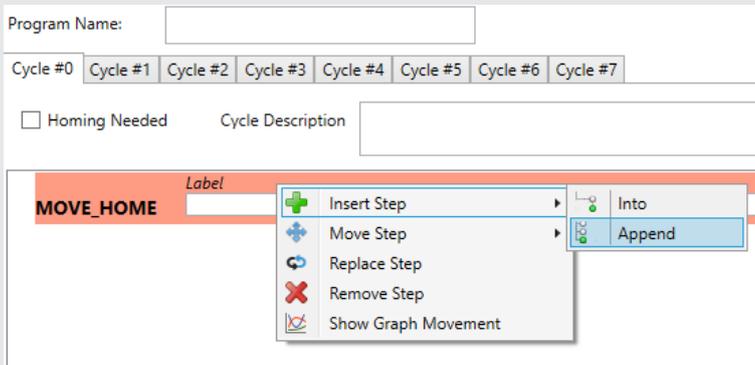
When the type of entry has been selected, a page in which you can select the type of command to assign is displayed divided into subsections. The various types of steps are displayed by category to the left of the page. If you select the desired category, the dropdown "Command Type" menu appears to the right.

Click OK to enter the step or Cancel to cancel the entry.

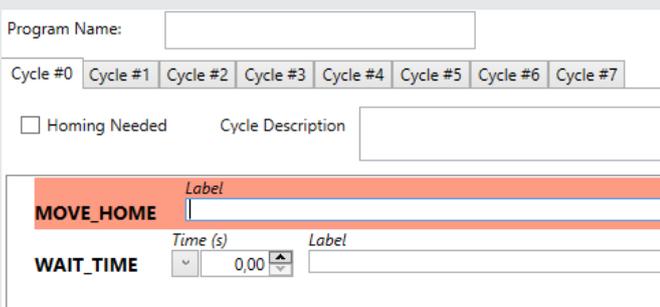
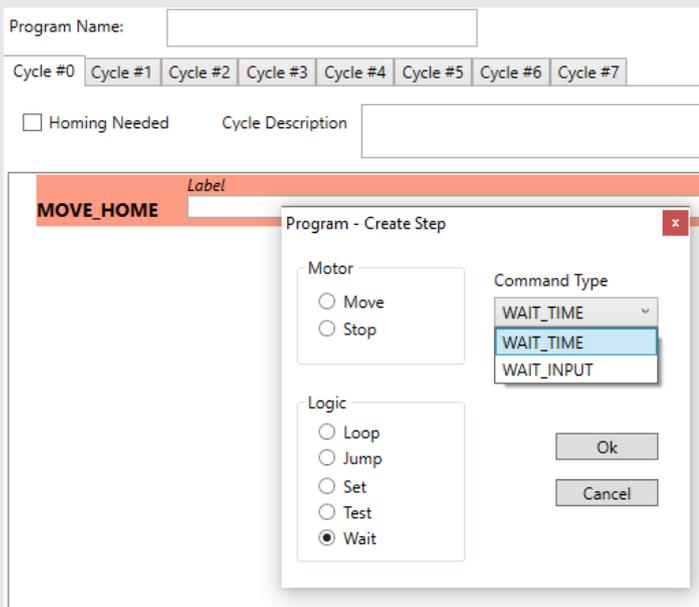
The selected step then appears in the selected cycle.

To enter another step next to the one just added, you can simply repeat the procedure, by selecting an Append entry.

Enter the WAIT\_TIME step to enter a temporal pause after MOVE\_HOME.



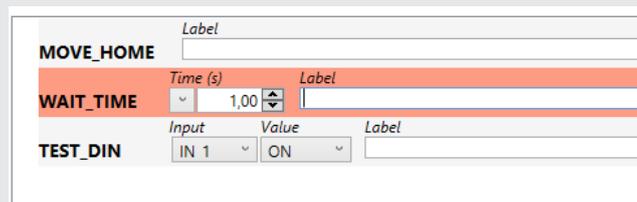
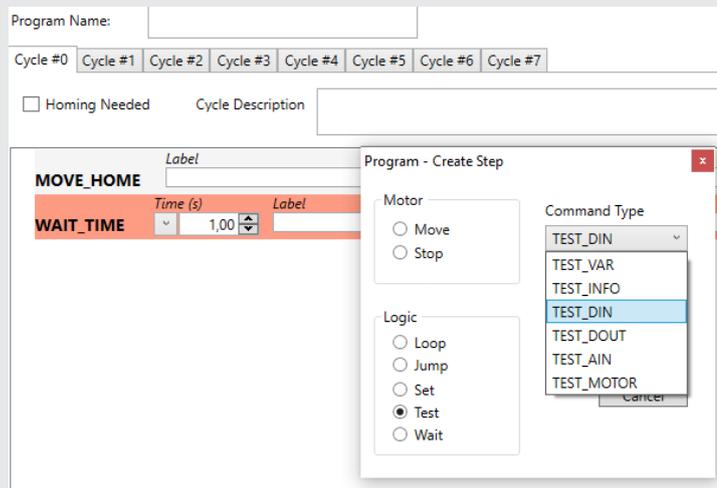
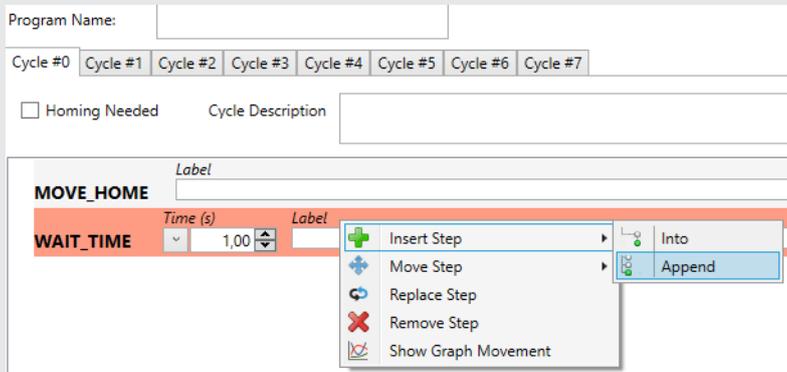
After clicking OK, the new WAIT\_TIME step will be Append to the step list.



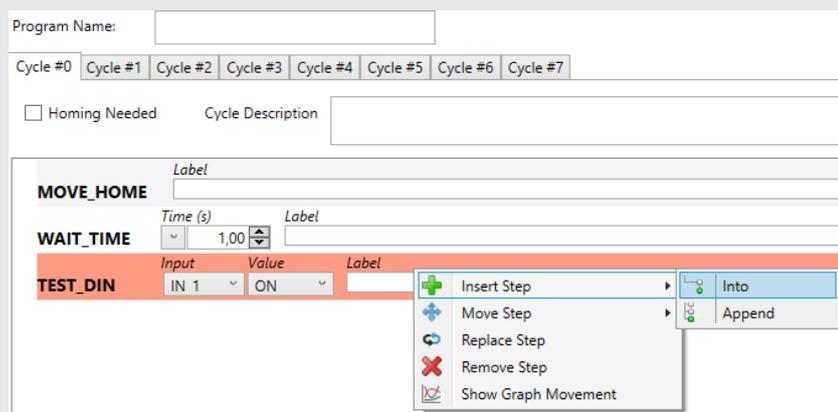
When the wait-time step has been lined up after the zeroing step, you can modify the time parameter.

## ENTERING NESTED STEPS

Some steps accept the execution of additional nested steps, such as the TEST\_DIN step to test the digital input state. To enter this step, you simply add it to the existing ones, as shown in the figure.

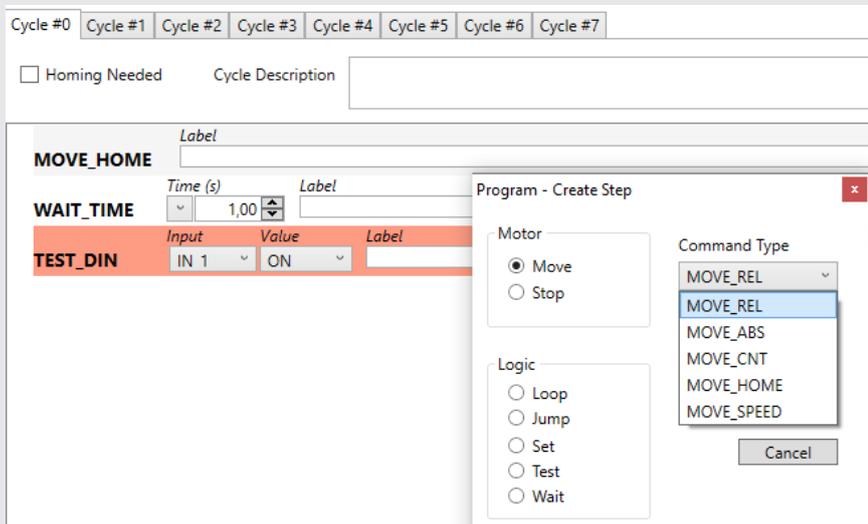


Once entered, you need to change the digital input you want to test (Input) and its logic state (Value). At this point, you need to enter a nested step, the execution of which depends on the positive result of the test. For example, if you want to set the positioning in front of the selected input, you can proceed as follows:

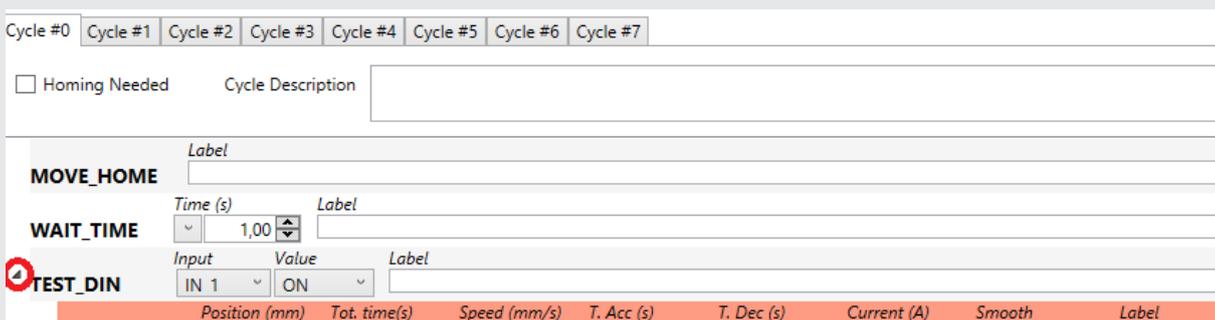


Select the "TEST\_DIN" step and click the right mouse button to select the "Insert Step -> Into".

The select the "MOVE\_REL" command.

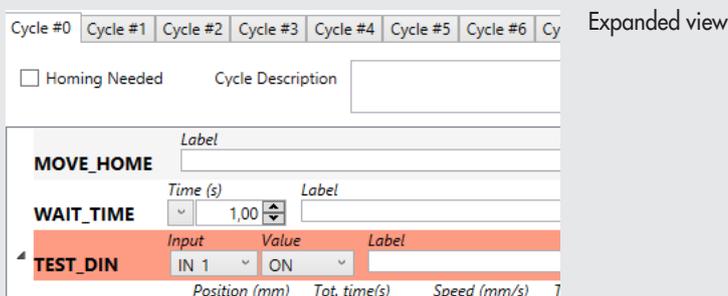


When you click OK, the step will be nested inside the "TEST\_DIN" step.

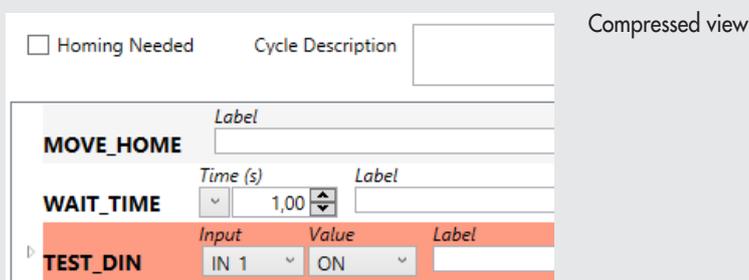


The steps (father) containing nested steps (son) are easily recognisable thanks to a triangle to the left of the "father" step (circled red in the figure). The "son" steps are lined up using a tab stop (indentation) and can be identified intuitively just by looking at the alignment. Each tab space corresponds to a father-son hierarchy (branch). There is no limit to the number of possible hierarchies.

**N.B.:** The triangle to the left of the father steps can be used to expand or compress the list of son steps, so as to facilitate a quick reading of the program.



Expanded view



Compressed view

At this point, you can personalise the parameters of the "MOVE\_REL" step.

<b>MOVE_HOME</b>	Label							
<b>WAIT_TIME</b>	Time (s)	Label						
<b>TEST_DIN</b>	Input	Value	Label					
	IN 1	ON						
<b>MOVE_REL</b>	Position (mm)	Tot. time(s)	Speed (mm/s)	T. Acc (s)	T. Dec (s)	Current (A)	Smooth	Label
	100,00	1,20	100,00	0,20	0,20	4,00	0	

In summary, the newly entered cycle will:

1. perform the zeroing (MOVE\_HOME).
2. wait 1 second (WAIT\_TIME).
3. check that input 1 of the board is turned to ON (TEST\_DIN).
4. if input 1 is turned to ON, it will move by 100 mm (MOVE\_REL) in m in. 1.0 s, with 0.5 s acceleration and deceleration time.
5. if not, there will be no movement.

## MOVING STEPS

Two options are provided to move one step before or after another:

1. Select the step to be moved and operate the buttons "Move Step" "Up" or "Move Step" "Down".
2. Select the step to be moved with the mouse and click the right button to select MoveStep->Up or MoveStep->Down.



Cycle #0 Cycle #1 Cycle #2 Cycle #3 Cycle #4 Cycle #5 Cycle #6 Cycle #7

Homing Needed      Cycle Description

<b>MOVE_HOME</b>	Label							
<b>WAIT_TIME</b>	Time (s)	Label						
<b>TEST_DIN</b>	Input	Value	Label					
	IN 1	ON						
<b>MOVE_REL</b>	Position (mm)	Tot. tin	Speed (mm/s)	T. Acc (s)	T. Dec (s)	Current (A)	Smooth	Label
	100,00	1,20	100,00	0,20	0,20	4,00	0	

- + Insert Step
- + Move Step
  - ↑ Up
  - ↓ Down
- ↻ Replace Step
- ✗ Remove Step
- 📊 Show Graph Movement

More precisely, the "WAIT\_TIME" step has been moved below the "TEST\_DIN", test, and the following result is obtained:

Homing Needed      Cycle Description

<b>MOVE_HOME</b>	Label							
<b>TEST_DIN</b>	Input	Value	Label					
	IN 1	ON						
<b>MOVE_REL</b>	Position (mm)	Tot. time(s)	Speed (mm/s)	T. Acc (s)	T. Dec (s)	Current (A)	Smooth	Label
	100,00	1,20	100,00	0,20	0,20	4,00	0	
<b>WAIT_TIME</b>	Time (s)	Label						
	1,00							

In this way, the 1.0 s wait time starts after the movement has been performed.

**N.B.:** The steps are moved within the same father-son branch. With this command, it is not possible to move steps between different father-son branches. If you wish to move or copy steps between different branches, please refer to the Copy Step section.

## REMOVING STEPS

Proceed in either of the following ways to remove a step:

1. Select the stop to be removed and use the toolbar with the **“Remove Step”** button.
2. Use the right mouse button and click the step to remove by selecting **“Remove Step”**



Cycle #0 Cycle #1 Cycle #2 Cycle #3 Cycle #4 Cycle #5 Cycle #6 Cycle #7

Homing Needed    Cycle Description

Label

**MOVE\_HOME**

Input Value Label

TEST\_DIN IN 1 ON

Position (mm) Tot. time(s) Speed (mm/s) T. Acc (s) T. Dec (s)

MOVE\_REL 100,00 1,20 100,00 0,20 0,20

Time (s) Label

**WAIT\_TIME** 1,00

- Insert Step
- Move Step
- Replace Step
- Remove Step**
- Show Graph Movement

You are prompted to confirm removal of the step. If you confirm, the step will be removed.

Cycle #0 Cycle #1 Cycle #2 Cycle #3 Cycle #4 Cycle #5 Cycle #6 Cycle #7

Homing Needed    Cycle Description

Label

**MOVE\_HOME**

Input Value Label

TEST\_DIN IN 1 ON

Position (mm) Tot. time(s) Speed (mm/s) T. Acc (s) T. Dec (s) Current (A)

MOVE\_REL 100,00 1,20 100,00 0,20 0,20

Time (s) Label

**WAIT\_TIME** 1,00

Remove

 Do you want to remove the current step?

Si No

The following result is obtained:

Cycle #0 Cycle #1 Cycle #2 Cycle #3 Cycle #4 Cycle #5 Cycle #6 Cycle #7

Homing Needed    Cycle Description

Label

**MOVE\_HOME**

Input Value Label

TEST\_DIN IN 1 ON

Position (mm) Tot. time(s) Speed (mm/s) T. Acc (s) T. Dec (s) Current (A) Smooth Label

MOVE\_REL 100,00 1,20 100,00 0,20 0,20 4,00 0

## REPLACING STEPS

You can replace one step when, for example, you wish to change a test logic condition for the execution of a next instruction. To do this, proceed as follows:

Select the Step to be replaced by clicking the right mouse button on "Replace Step".

Program Name:

Cycle #0 Cycle #1 Cycle #2 Cycle #3 Cycle #4 Cycle #5 Cycle #6 Cycle #7

Homing Needed      Cycle Description

Label	Input	Value	Label
MOVE_HOME			
TEST_DIN	IN 1	ON	
MOVE_REL		Position (mm) 100,00	Acc (s) 0,20

- Insert Step
- Move Step
- Replace Step
- Remove Step
- Show Graph Movement

In this case, you wish to replace the step associated with the digital input logic test with one associated with the analogue input logic test.

Program Name:

Cycle #0 Cycle #1 Cycle #2 Cycle #3 Cycle #4 Cycle #5 Cycle #6 Cycle #7

Homing Needed      Cycle Description

Label	Input	Value	Label
MOVE_HOME			
TEST_DIN	IN 1	ON	
MOVE_REL		Position (mm) 100,00      Tot. time(s) 1,20	

Program - Create Step

Motor

Move

Stop

Logic

Loop

Jump

Set

Test

Wait

Command Type

TEST\_VAR

TEST\_INFO

TEST\_DIN

TEST\_DOUT

TEST\_AIN

TEST\_MOTOR

Cancel

The following result is obtained:

Program Name:

Cycle #0 Cycle #1 Cycle #2 Cycle #3 Cycle #4 Cycle #5 Cycle #6 Cycle #7

Homing Needed      Cycle Description

Label	Input	Condition	Value	Label
MOVE_HOME				
TEST_AIN	AN1	=	0,00	
MOVE_REL		Position (mm)	Tot. time(s)	Speed (mm/s)
		100,00	1,20	100,00
				T. Acc (s)
				0,20
				T. Dec (s)
				0,20
				Current (A)
				4,00
				Sn

## COPYING STEPS

If you want to copy a step, this can be done by dragging the step and holding down the Ctrl key on the keypad simultaneously. For example, in the following program you want to perform the same identical test on input 1 and input 2.

Program Name:

Cycle #0 | Cycle #1 | Cycle #2 | Cycle #3 | Cycle #4 | Cycle #5 | Cycle #6 | Cycle #7

Homing Needed    Cycle Description

Label	Input	Value	Label
<b>MOVE_HOME</b>			
<b>TEST_DIN</b>	IN 1	ON	
MOVE_REL	Position (mm)	Tot. time(s)	Speed (mm/s)
	100,00	1,20	100,00
			T. Acc (s)
			0,20
			T. Dec (s)
			0,20
			Current (A)
			4,00
			Smooth
			0
			Label

Select the step to be copied, i.e. TEST\_DIN in this case.

Label	Input	Value	Label
<b>MOVE_HOME</b>			
<b>TEST_DIN</b>	IN 1	ON	
MOVE_REL	Position (mm)	Tot. time(s)	Speed (mm/s)
	100,00	1,20	100,00

Drag the step down to the bottom of the list or to the left, holding down the left mouse button, until the mouse pointer highlights the displacement in progress. The area marked in red is where the displacement mode is activated.

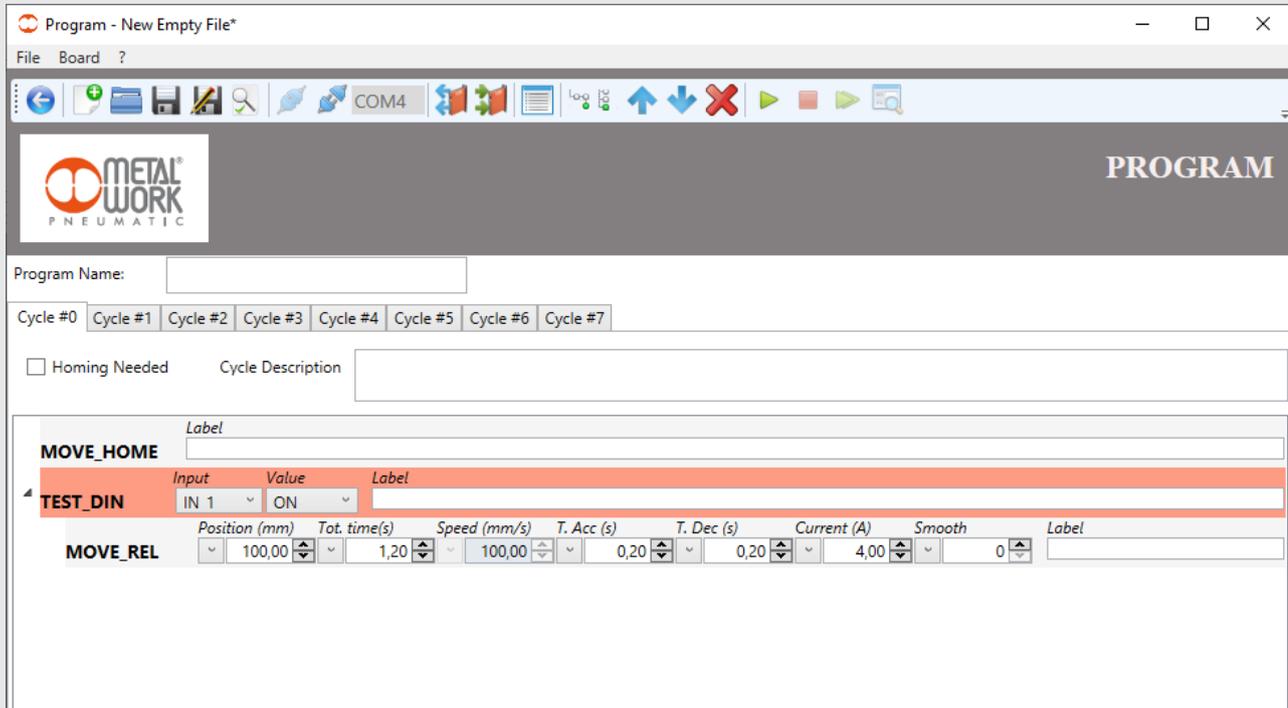
Cycle #0 | Cycle #1 | Cycle #2 | Cycle #3 | Cycle #4 | Cycle #5 | Cycle #6 | Cycle #7

Homing Needed    Cycle Description

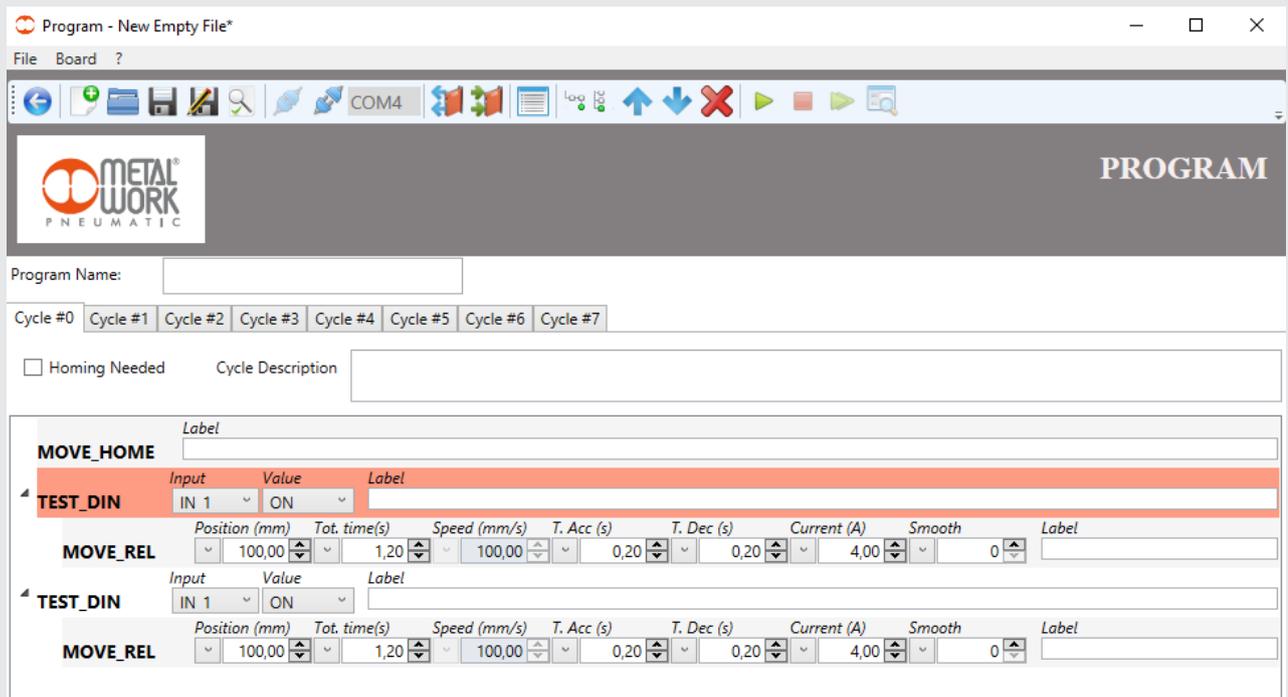
Label	Input	Value	Label
<b>MOVE_HOME</b>			
<b>TEST_DIN</b>	IN 1	ON	
MOVE_REL	Position (mm)	Tot. time(s)	Speed (mm/s)
	100,00	1,20	100,00
			T. Acc (s)
			0,20
			T. Dec (s)
			0,20
			Current (A)
			4,00
			Smooth
			0
			Label

CONNECTED

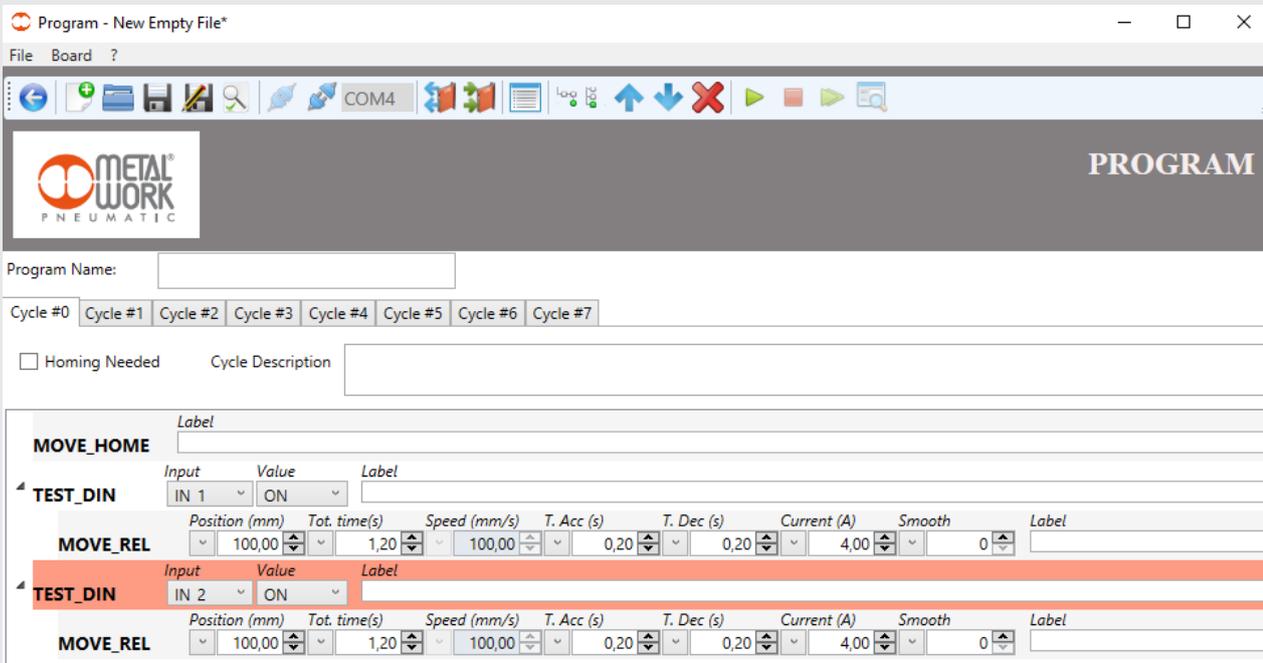
Without releasing the left mouse button and holding the Ctrl key down, the symbol “+” appears on the mouse pointer indicating the copy mode.



At this point, if you release the mouse left button, the copy is made and the step is lined up at the bottom of the step list.

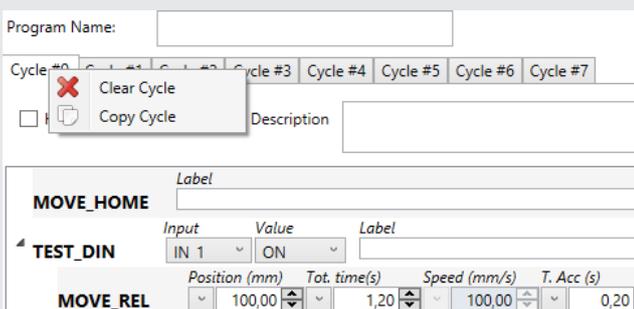


You only need to modify the "Input" parameter associated with the new "TEST\_DIN" instruction just enabled. The final result is:

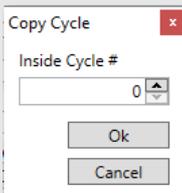


## COPY/CLEAR CYCLE

You can copy a whole cycle or remove all the steps in a cycle. Click the right mouse button on the cycle identification tag and select "Copy Cycle" or "Clear Cycle".



If you select "Copy Cycle", a dialogue box appears prompting you to specify the copy target cycle.



 **IMPORTANT!** The target cycle will be overwritten!

If you confirm, the cycle will be copied.

Cycle #0 | Cycle #1 | Cycle #2 | Cycle #3 | Cycle #4 | Cycle #5 | Cycle #6 | Cycle #7

Homing Needed    Cycle Description

---

**MOVE\_HOME**    Label

**TEST\_DIN**    Input: IN 1    Value: ON    Label

Input:    Value:    Label:

If you select "Clear Cycle", all the steps in the current cycle will be cleared.

Program Name:

Cycle #0 | Cycle #1 | Cycle #2 | Cycle #3 | Cycle #4 | Cycle #5 | Cycle #6 | Cycle #7

Clear Cycle

Copy Cycle    Cycle Description

If you confirm, the cycle will be cleared.

Cycle #0 | Cycle #1 | Cycle #2 | Cycle #3 | Cycle #4 | Cycle #5 | Cycle #6 | Cycle #7

Homing Needed    Cycle Description



## USING VARIABLES

In some step parameters, you can use variables, which are useful in case of repeated operations and/or to simplify program writing and maintainability.

Select the "OpenList" icon to display the variables in the ToolBar.



A dialogue box will be displayed with the list of variables.

Variable Name	Value
VAR1	0,00
VAR2	0,00
VAR3	0,00
VAR4	0,00
VAR5	0,00
VAR6	0,00
VAR7	0,00
VAR8	0,00
VAR9	0,00
VAR10	0,00
VAR11	0,00
VAR12	0,00
VAR13	0,00
VAR14	0,00
VAR15	0,00
VAR16	0,00
VAR17	0,00
VAR18	0,00
VAR19	0,00
VAR20	0,00
VAR21	0,00
VAR22	0,00

You can modify the name assigned by default to the variable by clicking the left column and the initialisation value by clicking the right column. This will be used to initialise the variable on first program start-up.

Variable Name	Value
TOTAL REPITIONS	10,00
CYCLE COUNTER	0,00
VAR3	0,00
VAR4	0,00
VAR5	0,00

Some instructions allow you to use one of the variables illustrated above as a parameter. To do this, you must access the VAR submenu.

Instruction	Counter	Label
REPEAT	1	
MOVE_REL	VAR	
WAIT_TIME	1,00	

Then select the desired variable.

Instruction	Variable	Condition	Value	Label
SET_VAR	CYCLE COUNTER	=	1,00	
WAIT_TIME	1,00			

All the variables are "global type": they can be viewed and available at any point of the program. They can be used to perform operations in different cycles simultaneously. This means that a modification to a variable in a cycle also applies to another cycle ("global variables").

**N.B.:** If a variable is used in multiple cycles, it will be initialised at the start of the first cycle only!  
Check for initialisations in the various cycles, if necessary.

Instruction	Counter	Label
REPEAT	TOTAL REPITIONS	
MOVE_REL	20,00	
SET_VAR	CYCLE COUNTER	
WAIT_TIME	1,00	

Two variables are used in this example: one for counting the cycles executed and the other to define the number of cycles to execute. To change the number of cycles to execute, simply change the value of the TOTAL REPS variable. The CYCLE COUNTER variable could be used, for example, to perform tests on the number of cycles executed.

## PROGRAM DEBUGGING

The Debug function can be used to check the execution of the step-by-step program, by monitoring the board inputs and outputs and the defined variables.

Select the "Start Debug" icon in the ToolBar to enable the Debug function.

Two dialogue boxes are displayed, one to monitor the variables, the other to check the inputs and outputs of the control board.

Program Name: 0

Cycle #0 Cycle #1 Cycle #2 Cycle #3 Cycle #4 Cycle #5 Cycle #6 Cycle #7

Homing Needed Cycle Description

**MOVE\_HOME**

Variable Condition Value Label

**SET\_VAR** TOTAL REPETITIONS = 0,00

Position (mm) Tot. time(s) Speed (mm/s) T. Acc (s) T. Dec (s) Current (A) Smooth Label

**MOVE\_REL** 100,00 2,00 55,56 0,20 0,20 6,00 0

Time (s) Label

**WAIT\_TIME** 1,00

Variable Condition Value Label

**SET\_VAR** CYCLE COUNTER + 1,00

**Variables L...**

TOTAL REPETITIONS	10,00
CYCLE COUNTER	0,00
VAR3	0,00
VAR4	0,00
VAR5	0,00
VAR6	0,00
VAR7	0,00
VAR8	0,00
VAR9	0,00
VAR10	0,00
VAR11	0,00
VAR12	0,00
VAR13	0,00

**Board Test**

Digital Inputs: 1 START, 2 STOP GLOBAL, 3 METHOME, 4 INPUT, 5 ENABLE, 6 TEST ALARM, 7 INPUT, 8 BELT, 9 SEL2, 10 INPUT, 11 INPUT, 12 INPUT, 13 INPUT, 14 HOMING

Digital Outputs: 1 ALARM, 2 IN HOME, 3 RUN CYCLE, 4 OUTPUT, 5 OUTPUT, 6 OUTPUT, 7 OUTPUT

Status: 1 READY, 2 MANUAL, 3 AUTO, 4 RUN CYCLE, 5 MOT MOVE, 6 MOT POS, 7 HOME, 8 POWER ENL, 9 LOG, 10 LOG+, 11 SERVO ON

Alarm: 1 PARAM, 2 PROGRAM, 3 MOT ERR, 4 EXTERNAL, 5 POS ERR, 6 CALIBRAT, 7 W DOG, 8 SHORT CUT, 9 ALARM, 10 EXT V. AN1, 11 EXT V. AN2

Analog Input: IN 1: POSITION 0.00 V (0.00) 0.00V 10.00V; IN 2: SPEED 0.00 V (0.00) 0.00V 10.00V; OUT 1: POSIZIONE 0.00 V (0.00) 0.00V 10.00V

Debug Status: Cycle: 0 Status: Wait Program

CONNECTED

**N.B.:** On starting the debug session, the current program is automatically overwritten.

The e-drive starts up on AUTO, and thus the cycle is selected and launched using the digital inputs connected to the board, as in a typical application of use.

Program Name: 0

Cycle #0 Cycle #1 Cycle #2 Cycle #3 Cycle #4 Cycle #5 Cycle #6 Cycle #7

Homing Needed Cycle Description

**MOVE\_HOME**

Variable Condition Value Label

**SET\_VAR** TOTAL REPETITIONS = 0,00

Position (mm) Tot. time(s) Speed (mm/s) T. Acc (s) T. Dec (s) Current (A) Smooth Label

**MOVE\_REL** 100,00 2,00 55,56 0,20 0,20 6,00 0

Time (s) Label

**WAIT\_TIME** 1,00

Variable Condition Value Label

**SET\_VAR** CYCLE COUNTER + 1,00

**Variables L...**

TOTAL REPETITIONS	10,00
CYCLE COUNTER	0,00
VAR3	0,00
VAR4	0,00
VAR5	0,00
VAR6	0,00
VAR7	0,00
VAR8	0,00
VAR9	0,00
VAR10	0,00
VAR11	0,00
VAR12	0,00
VAR13	0,00

**Test Schede**

Input Digitali: 1 HOMING, 2 METHOME, 3 START, 4 TEST ALARM, 5 ENABLE, 6 MANUAL, 7 MOT MOVE, 8 MOT POS, 9 LOG, 10 SEL2, 11 STOP-BLAN, 12 INPUT, 13 INPUT, 14 INPUT

Output Digitali: 1 ALARM, 2 BUSY, 3 IN-HOME, 4 IN-POS, 5 RUN CYCLE, 6 OUTPUT, 7 OUTPUT

Stato: 1 READY, 2 MANUAL, 3 AUTO, 4 RUN CYCLE, 5 MOT MOVE, 6 MOT POS, 7 HOME, 8 POWER ENL, 9 LOG, 10 LOG+, 11 SERVO ON

Allarmi: 1 PARAM, 2 PROGRAM, 3 MOT ERR, 4 EXTERNAL, 5 POS ERR, 6 CALIBRAT, 7 W DOG, 8 SHORT CUT, 9 ALARM, 10 EXT V. AN1, 11 EXT V. AN2

Ingressi Analogici: IN 1: POSIZIONE 0.00 V (0.00) 0.00V 10.00V; IN 2: VELOCITA 0.00 V (0.00) 0.00V 10.00V; OUT 1: POSIZIONE 0.00 V (0.00) 0.00V 10.00V

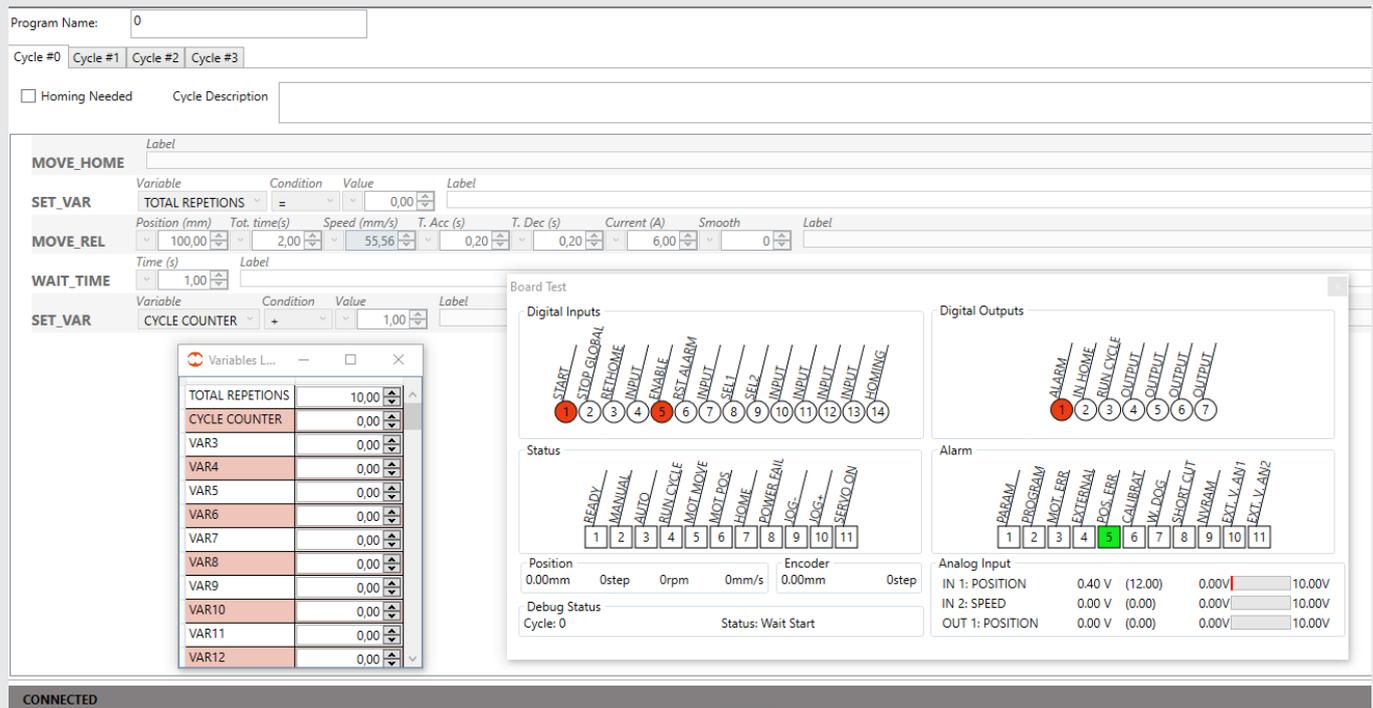
Stato Debug: Ciclo: 0 Stato: Attesa Start Step

CONNECTED

The "Wait Start" is displayed when the "Test Board" dialogue box is enabled, i.e. the board awaits the Start Cycle signal. Once the cycle to execute has been selected via the inputs (if configured) and the start command has been given, the board selects the first step to execute, awaits the start given by the user and then continues with the execution of the next step.

Press "Start Step" to start execution. The cycle performs the first step, moves to the next one available and returns to the Wait Start mode. The "Test Board" dialogue box displays the status of digital inputs and outputs in real time, while the "Variable List" dialogue box shows the updated status of the variables.

You can close each of the dialogue boxes by clicking [x] on the top right to close. To reopen it, just click the "Open Test Window" key in the Toolbar.



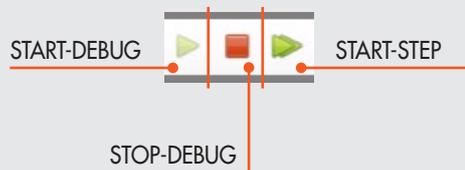
The screenshot shows the software interface with the following components:

- Program Name:** 0
- Cycle #0:** Cycle #1, Cycle #2, Cycle #3
- MOVE\_HOME:** Variable: TOTAL REPETIONS, Condition: =, Value: 0,00
- MOVE\_REL:** Position (mm): 100,00; Tot. time(s): 2,00; Speed (mm/s): 55,56; T. Acc (s): 0,20; T. Dec (s): 0,20; Current (A): 6,00; Smooth: 0
- WAIT\_TIME:** Time (s): 1,00
- SET\_VAR:** Variable: CYCLE COUNTER, Condition: +, Value: 1,00
- Variables List:**

TOTAL REPETIONS	10,00
CYCLE COUNTER	0,00
VAR3	0,00
VAR4	0,00
VAR5	0,00
VAR6	0,00
VAR7	0,00
VAR8	0,00
VAR9	0,00
VAR10	0,00
VAR11	0,00
VAR12	0,00
- Board Test:**
  - Digital Inputs:** START, STOP GLOBAL, BETHOME, INPUT, ENABLE, TEST ALARM, INPUT, SEL2, INPUT, INPUT, INPUT, INPUT, HOMING
  - Digital Outputs:** ALARM, IN HOME, RUN CYCLE, OUTPUT, OUTPUT, OUTPUT
  - Status:** READY, MANUAL, AUTO, RUN CYCLE, MOT MOVE, MOT POS, HOME, POWER FAIL, LOG+, SERVO ON
  - Alarm:** PARAM, PROGRAM, MOT ERR, EXTERNAL, POS ERR, CALIBRAT, W. DOG, SHORT CUT, NVRAM, EXT.V. AN1, EXT.V. AN2
  - Analog Input:** IN 1: POSITION (0.40 V (12.00)), IN 2: SPEED (0.00 V (0.00)), OUT 1: POSITION (0.00 V (0.00))
  - Debug Status:** Cycle: 0, Status: Wait Start

**N.B.:** Not all the steps can be paused. The nested steps are a clear example.

You can stop debugging at any time by clicking "Stop Debug" in the Toolbar.



# LIST OF COMMANDS

## MOTION STEP COMMANDS (MOVE UNIT)

### MOVE\_REL

	Position (mm)	Tot. time(s)	Speed (mm/s)	T. Acc (s)	T. Dec (s)	Current (A)	Smooth	Label
MOVE_REL	100,00	2,00	55,56	0,20	0,20	6	0	

This command can be used to position the actuator.

Parameters:

- **Position (mm)**

The movement expressed in millimetres to be performed by the actuator can be positive or negative, depending on the desired direction of movement. The maximum value cannot be greater than the actuator stroke set under the parameters.

The button  gives access to a menu in which you can set this parameter.

1. VAR = lets you select a variable to be assigned to the parameter.

2. CONST = (default) lets you enter manually the value to be assigned to the parameter.

3. AN = lets you select an analogue input to be assigned to the parameter (for further details, please refer to section entitled Analogue Input Configuration in Parameter Setup).

- **Tot Time (s)**

The total time in seconds allowed to complete the movement, including acceleration and deceleration times.

This button  gives access to a menu in which you can set this parameter.

1. VAR = lets you select a variable to be assigned to the parameter.

2. CONST = (default) lets you enter manually the value to be assigned to the parameter.

- **Speed (mm/s)**

The maximum movement speed calculated automatically depending on the configuration of the loaded parameters.

The maximum value cannot be greater than the maximum system speed set in the parameter setup menu.

- **T. Acc (s)**

The time it takes to reach the maximum speed, expressed in seconds. It cannot be less than the minimum system acceleration time set under the parameters.

The button  gives access to a menu in which you can set this parameter.

1. VAR = lets you select a variable to be assigned to the parameter.

2. CONST = (default) lets you enter manually the value to be assigned to the parameter.

- **T. Dec (s)**

The time it takes to reach the set position, expressed in seconds. It cannot be less than the minimum system deceleration time set under the parameters.

The button  gives access to a menu in which you can set this parameter.

1. VAR = lets you select a variable to be assigned to the parameter.

2. CONST = (default) lets you enter manually the value to be assigned to the parameter.

- **Current (A):** the maximum current that can be supplied for the specific instruction. It allows the use of the motor with reduced current by decreasing motor heating and increasing energy savings. For details, reference is made to the Product section.

- **Smooth (%):** parameter settable in the range 1-5, which introduces the jerk function between the acceleration/deceleration phase and the constant speed phase.

- **Comment (string):** a text string to comment on the step.

**N.B.:** The comments are not transferred to the e.drive, board. They are only saved on a computer file. When uploading the program from the board, the comments field is void. If the comments are worthy of note, we recommend saving the file to your computer before closing the PC software.

This is not a locking step and hence it may contain nested steps that are executed during execution of the movement.

The step is executed until the set position is reached or the motor is stopped.

## MOVE\_ABS

	Position (mm)	Speed (mm/s)	T. Acc (s)	T. Dec (s)	Current (A)	Smooth	Label
MOVE_ABS	100,00	60,00	0,20	0,20	6,00	0	

This command can be used for absolute actuator positioning, i.e. with respect to the zero position defined in the Homing function page.

Parameters:

- Position (mm)**  
 The absolute movement expressed in millimetres to be performed by the actuator. The maximum value cannot be greater than the actuator stroke set under the parameters.  
 The button  gives access to a menu in which you can set this parameter.
  - VAR = lets you select a variable to be assigned to the parameter.
  - CONST = (default) lets you enter manually the value to be assigned to the parameter.
  - AN = lets you select an analogue input to be assigned to the parameter (for further details, please refer to section entitled Analogue Input Configuration in Parameter Setup).
- Speed (mm/s)**  
 The maximum movement speed. The maximum value cannot be greater than the maximum system speed set in the parameter setup menu.  
 The button  gives access to a menu in which you can set this parameter.
  - VAR = lets you select a variable to be assigned to the parameter.
  - CONST = (default) lets you enter manually the value to be assigned to the parameter.
  - AN = lets you select an analogue input to be assigned to the parameter (for further details, please refer to section entitled Analogue Input Configuration in Parameter Setup).
- T. Acc (s)**  
 The time it takes to reach the maximum speed, expressed in seconds. It cannot be less than the minimum system acceleration time set under the parameters.  
 The button  gives access to a menu in which you can set this parameter.
  - VAR = lets you select a variable to be assigned to the parameter.
  - CONST = (default) lets you enter manually the value to be assigned to the parameter.
- T. Dec (s)**  
 The time it takes to reach the set position, expressed in seconds. It cannot be less than the minimum system deceleration time set under the parameters.  
 The button  gives access to a menu in which you can set this parameter.
  - VAR = lets you select a variable to be assigned to the parameter.
  - CONST = (default) lets you enter manually the value to be assigned to the parameter.
- Current (A)**: the maximum current that can be supplied for the specific instruction. It allows the use of the motor with reduced current by decreasing motor heating and increasing energy savings. For details, reference is made to the Product section.
- Smooth (%)**: parameter settable in the range 1-5, which introduces the jerk function between the acceleration/deceleration phase and the constant speed phase.
- Comment (string)**: a text string to comment on the step.

**N.B.:** The comments are not transferred to the e.drive, board. They are only saved on a computer file. When uploading the program from the board, the comments field is void. If the comments are worthy of note, we recommend saving the file to your computer before closing the PC software.

This is not a locking step and hence it may contain nested steps that are executed during execution of the movement.  
 The step is executed until the set position is reached or the motor is stopped.

## MOVE\_CNT

	Position (mm)	Speed (mm/s)	T. Acc (s)	T. Dec (s)	Current (A)	Smooth	Label
MOVE_CNT	50,00	10,00	0,20	0,20	6,00	0	

This command can be used for absolute and continuous actuator positioning. The actuator closely follows the change in the input analogue signal, according to the "Analogue" parameter setting function configured in the Setup menu.

Parameters:

- **Position (mm)**

The absolute movement expressed in millimetres to be covered by the actuator. The maximum value cannot be greater than the actuator stroke set under the parameters.

The button  gives access to a parameter setting menu.

1. VAR = lets you select a variable to be assigned to the parameter.

2. CONST = (default) lets you enter manually the value to be assigned to the parameter.

3. AN = lets you select an analogue input to be assigned to the parameter (for further details, please refer to section entitled Analogue Input Configuration in Parameter Setup).

- **Speed (mm/s)**

The maximum movement speed. The maximum value cannot be greater than the maximum system speed set in the parameter setup menu.

The button  gives access to a menu in which you can set this parameter

1. VAR = lets you select a variable to be assigned to the parameter.

2. CONST = (default) lets you enter manually the value to be assigned to the parameter.

3. AN = lets you select an analogue input to be assigned to the parameter (for further details, please refer to section entitled Analogue Input Configuration in Parameter Setup).

- **T. Acc (s)**

The time it takes to reach the maximum speed, expressed in seconds. It cannot be less than the minimum system acceleration time set under the parameters.

The button  gives access to a menu in which you can set this parameter.

1. VAR = lets you select a variable to be assigned to the parameter.

2. CONST = (default) lets you enter manually the value to be assigned to the parameter.

- **T. Dec (s)**

The time it takes to reach the set position, expressed in seconds. It cannot be less than the minimum system deceleration time set under the parameters.

The button  gives access to a menu in which you can set this parameter.

1. VAR = lets you select a variable to be assigned to the parameter.

2. CONST = (default) lets you enter manually the value to be assigned to the parameter.

- **Current (A)**: the maximum current that can be supplied for the specific instruction. It allows the use of the motor with reduced current by decreasing motor heating and increasing energy savings. For details, reference is made to the Product section.

- **Smooth (%)**: parameter settable in the range 1-5, which introduces the jerk function between the acceleration/deceleration phase and the constant speed phase.

- **Comment (string)**: a text string to comment on the step.

**N.B.:** The comments are not transferred to the e.drive, board. They are only saved on a computer file. When uploading the program from the board, the comments field is void. If the comments are worthy of note, we recommend saving the file to your computer before closing the PC software.

This is not a locking step and hence it may contain nested steps that are executed during execution of the movement.

The step is executed until the motor is stopped (STOP\_MOTOR).

## MOVE\_HOME

	Label
MOVE_HOME	

This command is used to perform homing (generally used in the 0 cycle), according to the parameter configuration set in the setup menu. This instruction is mandatory when "Request Homing" is checked in the header of one or more cycles.

Parameters:

- **Comment (string)**: a text string to comment on the step.

**N.B.:** The comments are not transferred to the e.drive board. They are only saved on a computer file. When uploading the program from the board, the comments field is void. If the comments are worthy of note, we recommend saving the file to your computer before closing the PC software.

This is not a locking step and hence it may contain nested steps that are executed during execution of the movement. The step is executed until the set position is reached or the motor is stopped.

## MOVE\_SPEED

	Speed (mm/s)	T. Acc (s)	T. Dec (s)	Current (A)	Smooth	Label
MOVE_SPEED	100,00	0,20	0,20	6,00	0	

This command can be used to perform a continuous movement at the set speed. The movement is stopped automatically when the maximum stroke is reached.

Parameters:

- **Speed (mm/s)**  
The maximum movement speed. The maximum value cannot be greater than the maximum speed set in the setup menu parameters. The button  gives access to a menu in which you can set this parameter.
  1. VAR = lets you select a variable to be assigned to the parameter.
  2. CONST = (default) lets you enter manually the value to be assigned to the parameter.
  3. AN = lets you select an analogue input to be assigned to the parameter (for further details, please refer to section entitled Analogue Input Configuration in Parameter Setup).
- **T. Acc (s)**  
The time it takes to reach the maximum speed, expressed in seconds. It cannot be less than the minimum system acceleration time set under the parameters. The button  gives access to a menu in which you can set this parameter.
  1. VAR = lets you select a variable to be assigned to the parameter.
  2. CONST = (default) lets you enter manually the value to be assigned to the parameter.
- **T. Dec (s)**  
The time it takes to reach the set position, expressed in seconds. It cannot be less than the minimum system deceleration time set under the parameters. The button  gives access to a menu in which you can set this parameter.
  1. VAR = lets you select a variable to be assigned to the parameter.
  2. CONST = (default) lets you enter manually the value to be assigned to the parameter.
- **Current (A)**: the maximum current that can be supplied for the specific instruction. It allows the use of the motor with reduced current by decreasing motor heating and increasing energy savings. For details, reference is made to the Product section.
- **Smooth (%)**: parameter settable in the range 1-5, which introduces the jerk function between the acceleration/deceleration phase and the constant speed phase.
- **Comment (string)**: a text string to comment on the step.

**N.B.:** The comments are not transferred to the e.drive, board. They are only saved on a computer file. When uploading the program from the board, the comments field is void. If the comments are worthy of note, we recommend saving the file to your computer before closing the PC software.

This is not a locking step and hence it may contain nested steps that are executed during execution of the movement. The step is executed until the set position is reached or the motor is stopped.

## STOP COMMAND (STOP GROUP)

### STOP\_MOTOR

	T. Dec (s)	Label
STOP_MOTOR	0,15	

This command stops the motor with the set parameters. If the step is nested into a step of the MOVE group, it forces exit from the MOVE step, by moving to the next step when the last nested MOVE step has been reached.

Parameters:

- **T. Dec (s)**  
The time it takes to reach the set position, expressed in seconds. It cannot be less than the minimum system deceleration time set under the parameters. The button  gives access to a menu in which you can set this parameter.
  1. VAR = lets you select a variable to be assigned to the parameter.
  2. CONST = (default) lets you enter manually the value to be assigned to the parameter.
- **Comment (string)**: a text string to comment on the step.

**N.B.:** The comments are not transferred to the e.drive board. They are only saved on a computer file. When uploading the program from the board, the comments field is void. If the comments are worthy of note, we recommend saving the file to your computer before closing the PC software.

This step cannot have nested steps.

## REPEAT COMMAND (LOOP GROUP)

### REPEAT

	Counter	Label
REPEAT	<input type="text" value="1"/>	<input type="text"/>

This command performs a repetition of any nested steps.

Parameters:

- **Count (numeric)**  
Number of nested step repetitions. When all the set repetitions have been performed, the system moves to the next non-nested instruction. The button  gives access to a menu in which you can set this parameter.
  1. VAR = lets you select a variable to be assigned to the parameter.
  2. CONST = (default) lets you enter manually the value to be assigned to the parameter.
- **Comment (string)**  
A text string to comment on the step.

**N.B.:** The comments are not transferred to the e.drive board. They are only saved on a computer file. When uploading the program from the board, the comments field is void. If the comments are worthy of note, we recommend saving the file to your computer before closing the PC software.

This is not a locking step and hence it may contain nested steps that are executed during the loop cycle. The step will be executed until the counter has reached the set count has been reached.

## LOOP COMMAND (LOOP GROUP)

### LOOP

	Label
LOOP	<input type="text"/>

This command performs an infinite repetition of any nested steps.

Parameters:

- **Comment (string)**  
A text string to comment on the step.

**N.B.:** The comments are not transferred to the e.drive board. They are only saved on a computer file. When uploading the program from the board, the comments field is void. If the comments are worthy of note, we recommend saving the file to your computer before closing the PC software.

This is a locking step and hence it may contain nested steps that are executed during the loop cycle. The step will be executed continuously until a STOP command is received.

## JUMP COMMAND (JUMP GROUP)

### RESTART

	Label
RESTART	<input type="text"/>

The cycle restarts from the first step.

Parameters:

- **Comment (string)**  
A text string to comment on the step.

**N.B.:** The comments are not transferred to the e.drive board. They are only saved on a computer file. When uploading the program from the board, the comments field is void. If the comments are worthy of note, we recommend saving the file to your computer before closing the PC software.

This step cannot have nested steps.

## SET COMMAND (SET GROUP)

### SET\_OUTPUT

	Output	Status	Label
SET_OUTPUT	OUT 1	ON	

The command sets a digital output to the specified status.

Parameters:

- **Output**  
The logic output of the board to be set
- **Value (ON, OFF)**  
The value you wish to set: output enabled (ON) or output disabled (OFF)
- **Comment (string)**  
A text string to comment on the step.

**N.B.:** The comments are not transferred to the e.drive board. They are only saved on a computer file. When uploading the program from the board, the comments field is void. If the comments are worthy of note, we recommend saving the file to your computer before closing the PC software.

This step cannot have nested steps.

### SET\_VAR

	Variable	Condition	Value	Label
SET_VAR	VAR3	=	0,00	

This command sets the value of a variable.

Parameters:

- **Variable (name of variable)**  
The variable to be set.
- **Condition** (= equal to, + add, - deduct, x multiply, : divide). The type of condition to be used.
- **Value (numeric)**  
The value to be set on the variable.  
Via the  button, you can access a parameter setting menu.
  1. VAR = lets you select a variable to be assigned to the parameter.
  2. CONST = (default) lets you enter manually the value to be assigned to the parameter.
- **Comment (string)**  
A text string to comment on the step.

**N.B.:** The comments are not transferred to the e.motion board. They are only saved on a computer file. When uploading the program from the board, the comments field is void. If the comments are worthy of note, we recommend saving the file to your computer before closing the PC software.

This step cannot have nested steps.

### SET\_HOME

	Label
SET_HOME	

This command sets the reference position.

It enables the "HOME" status flag and lets you execute, though temporarily, cycles for which the "Request Homing" box must be checked.

Parameters:

- **Comment (string)**  
A text string to comment on the step.

**N.B.:** The comments are not transferred to the e.drive board. They are only saved on a computer file. When uploading the program from the board, the comments field is void. If the comments are worthy of note, we recommend saving the file to your computer before closing the PC software.

This step cannot have nested steps.

### SET\_CURRENT

	Current (A)	Label
SET_CURRENT	2,00	

The command sets the maximum load admitted during the movement.

Parameters:

- **Current (A)**  
The value of the current to be set.  
Via the  button, you can access a parameter setting menu.
  1. VAR = lets you select a variable to be assigned to the parameter.
  2. CONST = (default) lets you enter manually the value to be assigned to the parameter.
  3. AN = (default) lets you enter manually the value to be assigned to the parameter.
- **Comment (string)**  
A text string to comment on the step.

**N.B.:** The comments are not transferred to the e.drive board. They are only saved on a computer file. When uploading the program from the board, the comments field is void. If the comments are worthy of note, we recommend saving the file to your computer before closing the PC software.

## STATE COMMANDS (TEST GROUP)

### TEST\_VAR

	Variable	Condition	Value	Label
TEST_VAR	VAR3	=	0,00	

This command performs a test on a previously set variable.

Parameters:

- **Variable (name of variable)**  
The variable to be set.
- **Condition** (!= other than, = equal to, > greater than, >= greater than or equal to, < less than, <= less than or equal to, >=< included, <> not included). The type of condition to be used.
- **Value (numeric)**  
The value to be compared with the variable or the minimum value in the case of a condition that uses a range of values.  
Via the  button, you can access a parameter setting menu.
  1. VAR = lets you select a variable to be assigned to the parameter.
  2. CONST = (default) lets you enter manually the value to be assigned to the parameter.
- **Value 2 (numeric)**  
Optional, to be used only to specify the maximum value in the case of a condition using a range of values.  
Via the  button, you can access a parameter setting menu.
  1. VAR = lets you select a variable to be assigned to the parameter.
  2. CONST = (default) lets you enter manually the value to be assigned to the parameter.
- **Comment (string)**  
A text string to comment on the step.

**N.B.:** The comments are not transferred to the e.drive board. They are only saved on a computer file. When uploading the program from the board, the comments field is void. If the comments are worthy of note, we recommend saving the file to your computer before closing the PC software.

This is not a locking step for the following steps not nested, but it may contain nested steps that are executed only if the test has been passed.

### TEST\_INFO

	Info	Condition	Value	Label
TEST_INFO	POSITION	=	0,00	

This command is used to perform a test on information available by the system during execution of a cycle.

Parameters:

- **Info** (Position, Speed, Force). The type of information to be tested.
- **Condition** (!= other than, = equal to, > greater than, >= greater than or equal to, < less than, <= less than or equal to, >=< included, <> not included). The type of condition to be used.
- **Value (numeric)**  
The value to be compared with the variable or the minimum value in the case of a condition that uses a range of values.  
Via the  button, you can access a parameter setting menu.
  1. VAR = lets you select a variable to be assigned to the parameter.
  2. CONST = (default) lets you enter manually the value to be assigned to the parameter.
- **Value 2 (numeric)**  
Optional, to be used only to specify the maximum value in the case of a condition using a range of values.  
Via the  button, you can access a parameter setting menu.
  1. VAR = lets you select a variable to be assigned to the parameter.
  2. CONST = (default) lets you enter manually the value to be assigned to the parameter.
- **Comment (string)**  
A text string to comment on the step.

**N.B.:** The comments are not transferred to the e.drive board. They are only saved on a computer file. When uploading the program from the board, the comments field is void. If the comments are worthy of note, we recommend saving the file to your computer before closing the PC software.

This is not a locking step for the following steps not nested, but it may contain nested steps that are executed only if the test has been passed.

## TEST\_DIN

	Input	Value	Label
TEST_DIN	IN 1	ON	

This command is used to perform a test on a digital input of the board.

Parameters:

- **Input**  
The digital input of the board to be tested.
- **Value (ON, OFF)**  
The value to be tested, input enabled (ON) or input disabled (OFF)
- **Comment (string)**  
A text string to comment on the step.

**N.B.:** The comments are not transferred to the e.drive board. They are only saved on a computer file. When uploading the program from the board, the comments field is void. If the comments are worthy of note, we recommend saving the file to your computer before closing the PC software.

This is not a locking step for the following steps not nested, but it may contain nested steps that are executed only if the test has been passed.

## TEST\_DOUT

	Output	Value	Label
TEST_DOUT	OUT 1	ON	

This command is used to perform a test on a digital output of the board.

Parameters:

- **Output**  
The digital output of the board to be tested.
- **Value (ON, OFF)**  
The value to be tested, input enabled (ON) or input disabled (OFF)
- **Comment (string)**  
A text string to comment on the step.

**N.B.:** The comments are not transferred to the e.drive board. They are only saved on a computer file. When uploading the program from the board, the comments field is void. If the comments are worthy of note, we recommend saving the file to your computer before closing the PC software.

This is not a locking step for the following steps not nested, but it may contain nested steps that are executed only if the test has been passed.

## TEST\_AIN

	Input	Condition	Value	Label
TEST_AIN	AN1	=	0,00	

This command is used to perform a test on the analogue input of the board.

Parameters:

- **Input** (AN1 analogue input 1, AN2 analogue input 2). The analogue input of the board to be tested.
- **Condition** (!= other than, = equal to, > greater than, >= greater than or equal to, < less than, <= less than or equal to, >=< included, <> not included). The type of condition to be used.
- **Value (numeric)**  
The value to be compared with the variable or the minimum value in the case of a condition that uses a range of values.  
Via the  button, you can access a parameter setting menu.
  1. VAR = lets you select a variable to be assigned to the parameter
  2. CONST = (default) lets you enter manually the value to be assigned to the parameter.
- **Value 2 (numeric)**  
Optional, to be used only to specify the maximum value in the case of a condition using a range of values.  
Via the  button, you can access a parameter setting menu.
  1. VAR = lets you select a variable to be assigned to the parameter.
  2. CONST = (default) lets you enter manually the value to be assigned to the parameter.
- **Comment (string)**  
A text string to comment on the step.

**N.B.:** The comments are not transferred to the e.drive board. They are only saved on a computer file. When uploading the program from the board, the comments field is void. If the comments are worthy of note, we recommend saving the file to your computer before closing the PC software.

This is not a locking step for the following steps not nested, but it may contain nested steps that are executed only if the test has been passed.

## WAIT COMMANDS (WAIT GROUP)

### WAIT\_TIME

	Time (s)	Label
WAIT_TIME	0,00	

This command is used to execute a delay for the number of set seconds.

Parameters:

- **Time (s)**  
Number of delay seconds.  
Via the  button, you can access a parameter setting menu.
  1. VAR = lets you select a variable to be assigned to the parameter.
  2. CONST = (default) lets you enter manually the value to be assigned to the parameter.
- **Comment (string)**  
A text string to comment on the step.

**N.B.:** The comments are not transferred to the e.drive board. They are only saved on a computer file. When uploading the program from the board, the comments field is void. If the comments are worthy of note, we recommend saving the file to your computer before closing the PC software.

This is a locking step but it may contain nested steps that will be executed when the set delay time has elapsed.

### WAIT\_INPUT

	Input	Status	Label
WAIT_INPUT	IN 1	ON	

The command activates a delay after which the selected input is like the one set.

Parameters:

- **Input**  
The logic input of the board to be tested.
- **Status (ON, OFF)**  
The status to be tested, input enabled (ON) or input disabled (OFF).
- **Comment (string)**  
A text string to comment on the step.

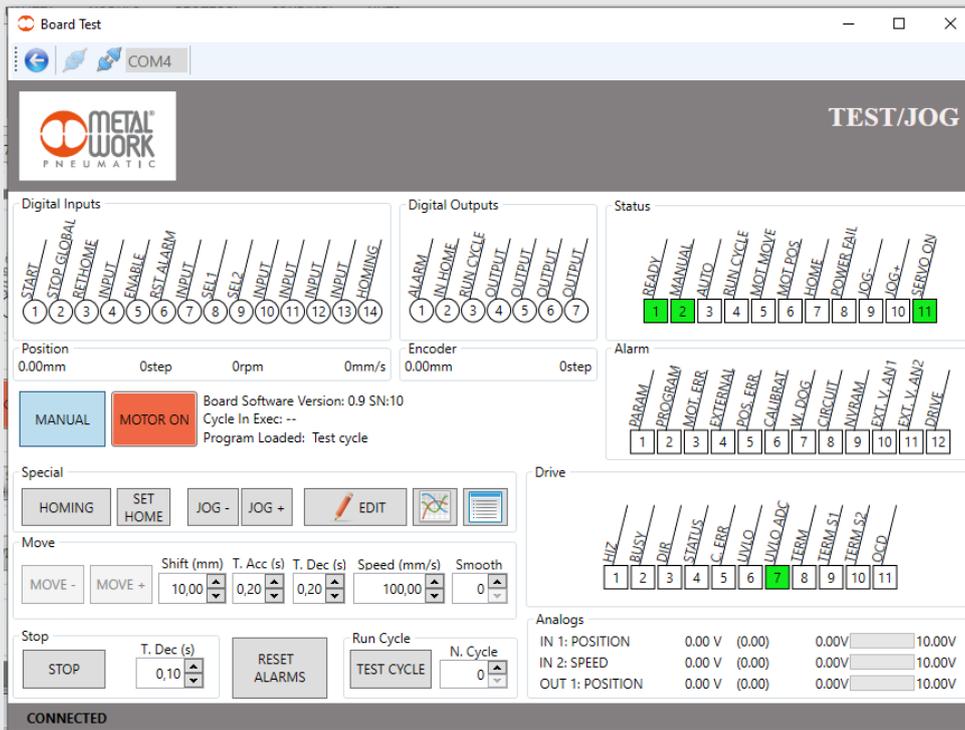
**N.B.:** The comments are not transferred to the e.drive board. They are only saved on a computer file. When uploading the program from the board, the comments field is void. If the comments are worthy of note, we recommend saving the file to your computer before closing the PC software.

This is a locking step but it may contain nested steps that will be executed until the selected input is like the one set.

## TEST AND JOG MENU

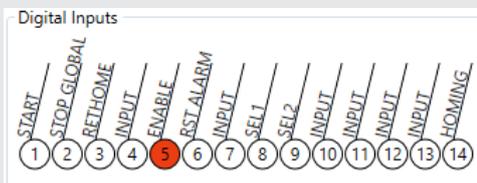
The Test and Jog menu can be used to perform tests on the e.drive board, check the status of digital inputs and outputs and monitor the presence of any alarms.

**N.B.:** In order to use the various functions, the board must be connected to the PC. The parameters and program file must be uploaded for a correct display of the functions associated with the inputs and to perform the tests on the set cycles.



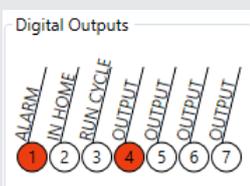
### Digital inputs

The Digital Input window displays the status of inputs in real time. If an input is enabled (ON), the corresponding indicator light (LED) turns red. The names of the functions associated to the inputs are those configured in the setup menu.



### Digital outputs

The Digital Input window displays the status of outputs in real time. If an output is enabled (ON), the corresponding indicator light (LED) turns red. The names of the functions associated to the outputs are those configured in the setup menu.



If an output is configured as a simple OUTPUT, you can force activation to ON or OFF just by clicking it.

## General status

In the Status window you can check the board status flags.



Name	State
READY	System ready for movements
MANUAL	System on manual mode (executes commands from PC)
AUTO	System on automatic mode (executes commands from input)
RUN_CYCLE	Cycle in progress
MOT_MOVE	Axis moving
MOT_POS	The motor has reached the required position
HOME	Zeroing performed
POWER_FAIL	System power failure
JOG-	Axis performing negative jog
JOG+	Axis performing positive jog
SERVO_ON	Servo system activated

## Alarms

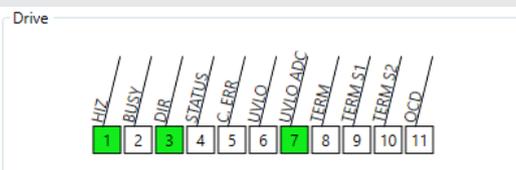
In the Alarms window you can check the flags of the various alarms on the board.



Name	State
PARAM	Error in the Parameters
PROGRAM	Error in the program
MOT.ERR	Motor failure
EXTERNAL	External alarm
POS. ERR.	Position error
CALIBRAT	Error in analogue input/output HW calibration
W. DOG.	Watch dog error
CIRCUIT	Digital output short-circuiting
NVRAM	NVRAM error. The board is switched off and it is not able to save in time the parameters on the eeprom.
EXT. V. AN1	Analogue output 1 overpowered
EXT. V. AN2	Analogue output 2 overpowered
DRIVE	Drive alarm

## Drive

In the Alarm page, you can check the flags of the various power-related alarms.



Name	State
HIZ	High impedance (no torque)
BUSY	Drive running (in command mode only)
DIR	Direction of rotation
STATUS	Drive status
C. ERR.	Command error sent to the drive
UVLO	Under-voltage or reset event
UVLO ADC	AD converter under-voltage
TERM	Temperature warning
TERM S1	Temperature alarm: transistor off
TERM S2	Temperature alarm: device off
OCD	Overcurrent: current higher than the set threshold

## Analogue

The Analogue Input window displays the status of analogue inputs according to their configuration.

Analogs			
IN 1: POSITION	0.00 V (0.00)	0.00V	10.00V
IN 2: SPEED	0.00 V (0.00)	0.00V	10.00V
OUT 1: POSITION	0.00 V (0.00)	0.00V	10.00V

The figure below shows the name of the analogue input "IN 1: POSITION", the value read in real time is "0.01 V", the number in brackets (0.50) is the conversion of the analogue value into the engineering value based on the parameter setting defined during setup.

## GENERAL INFORMATION

The Position window displays the current position of the actuator, the number of steps controlled by the drive, the speed in real time and the number of encoder pulses detected.

Position	8477step	0rpm	0mm/s	Encoder	0.00mm	0step
----------	----------	------	-------	---------	--------	-------

An area displaying the software version, the current cycle and the name of the program loaded on the board is also provided.

Board Software Version: 0.9 SN:10  
 Cycle In Exec: --  
 Program Loaded: Test cycle

In addition to the various displays, buttons are also provided for performing tests:

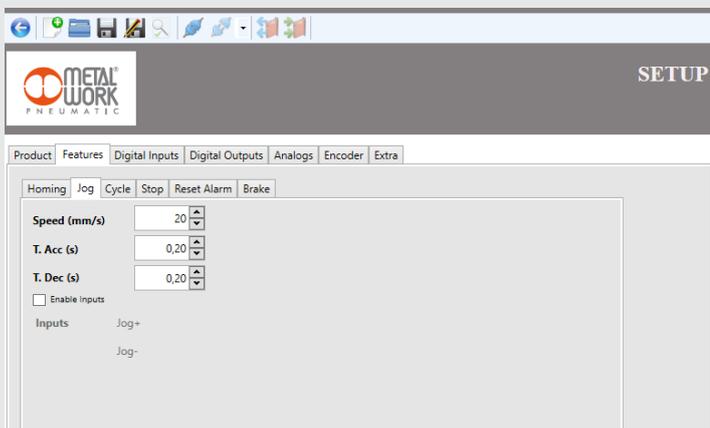
This can be used to switch from the Auto to the Manual mode.  
 The tests can only be performed in Manual mode.

Enables / Disables the motor

Special

In the "Special" window, you can perform Zeroing by clicking HOMING, execute movements using the JOG- and JOG+ functions, set the zero value via the SET HOME button and real-time display of both the motion diagram and update the variables used in the current cycle.

These movements use the parameters defined during setup. Via the MODIFY button, you can change both the homing mode and the jog motion parameters.



Using the  you return to the previous dialogue box.  
 The new parameters are automatically sent to the board.

Special

The MOVE-/MOVE+ buttons can be used to perform movements, by specifying the stroke, the acceleration and deceleration time and the speed.

Stop

The STOP button stops the motor, by specifying the desired deceleration time.

Resets alarms.

Run Cycle

Starts the specified cycle.

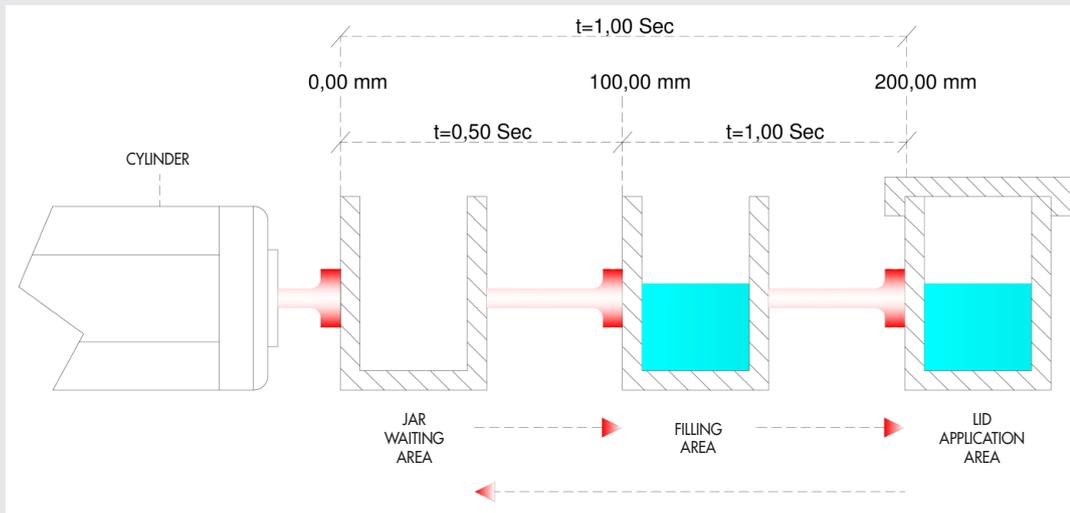
# APPLICATION EXAMPLES

This chapter illustrates some sample cycles.

## EXAMPLE 1: Simple positioner

The application relates to the movement of jars. The global system deals with positioning a jar, filling it and closing it using an airtight lid, followed by final positioning for packaging. Each pack can contain up to 10 jars. The e-drive receives a digital input signal and positions the jar in the filling area, then in the lid application area, and brings the actuator back to its initial position. The cycle is repeated ten times.

The system can be summarised in the following layout:



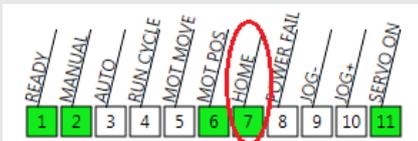
Two cycles are created. In the first one, the axis is zeroed, while in the second a work cycle is implemented. As to the execution sequence of the various cycles, please refer to the section entitled "Setup->Cycle".

Upon completion of the parameter setting in the setup section in the MWDRIVE software, the first operation in the PROGRAM dialogue box consists of entering the Program Name: 10-pcs packaging.

The comment "Zeroing cycle" is entered as a description of **Cycle #0**.

The cycle consists of one MOVE\_HOME instruction, and the relevant comment is: Homing Switch.

**It is important to check the box "Request Homing" in the cycle type. Only on completion of zeroing, the HOME "status flag" activates.**



This flag is the essential condition for running the cycle when the Request Homing box has been checked.

Program Name:

Cycle #0	Cycle #1	Cycle #2	Cycle #3	Cycle #4	Cycle #5	Cycle #6	Cycle #7
----------	----------	----------	----------	----------	----------	----------	----------

Homing Needed      Cycle Description:

<b>MOVE_HOME</b>	Label: <input type="text" value="Performs reset"/>
------------------	--

Work **Ciclo #1** is now implemented.

Check the Request Homing box to ensure the execution of the cycle starting from a specific, unambiguous reference position: the axis zero. The description of the cycle in this case is: cycle for filling and packaging 10 pieces.

Once the cycle start command has been given, the start movement signal is represented by a piece detection photoelectric cell. It was chosen to connect the sensor to digital input no. 12 of the board, hence the first step to enter is WAIT\_INPUT, with the following specifications: IN 12 as the Input, ON as the Status, Wait Jar as the Comment. A mandatory safety condition is that there should be no jars in the other two positions, i.e. in the filling and the lid application areas. A piece detection photoelectric cell is installed in each position. It is therefore necessary to test the other two sensors connected to inputs IN 13 and IN 14.

Program Name:								
Cycle #0	Cycle #1	Cycle #2	Cycle #3	Cycle #4	Cycle #5	Cycle #6	Cycle #7	
<input checked="" type="checkbox"/>	Homing Needed		Cycle Description					Filling and packaging cycle 10 pcs
<b>WAIT_INPUT</b>	Input	Status	Label					
	IN 12	ON	Waiting for jar					
	Time (s)	Label						
	0,50	Waiting for filling						
<b>TEST_DIN</b>	Input	Value	Label					
	IN 14	OFF	Check plug application area not occupied					

When the presence of the jar has been ascertained, it needs to be positioned in the filling area.

Enter step MOVE\_REL with the following specifications: Position = 100.00 mm, Total time = 0.50 s, Acc.T = Dec.T = 0.25 s.

Program Name:								
Cycle #0	Cycle #1	Cycle #2	Cycle #3	Cycle #4	Cycle #5	Cycle #6	Cycle #7	
<input checked="" type="checkbox"/>	Homing Needed		Cycle Description					Filling and packaging cycle 10 pcs
<b>WAIT_INPUT</b>	Input	Status	Label					
	IN 12	ON	Waiting for jar					
	Time (s)	Label						
	0,50	Waiting for filling						
<b>TEST_DIN</b>	Input	Value	Label					
	IN 14	OFF	Check plug application area not occupied					
<b>MOVE_REL</b>	Position (mm)	Tot. time(s)	Speed (mm/s)	T. Acc (s)	T. Dec (s)	Current (A)	Smooth	Label
	100,00	2,20	50,00	0,20	0,20	6,00	0	Arrival in the filling area

In order to allow the system to fill the jar, it is necessary to set the pause time.

The next step is WAIT\_TIME: Time = 0.50 s, Comment: Wait Filling.

Program Name:								
Cycle #0	Cycle #1	Cycle #2	Cycle #3	Cycle #4	Cycle #5	Cycle #6	Cycle #7	
<input checked="" type="checkbox"/>	Homing Needed		Cycle Description					Filling and packaging cycle 10 pcs
<b>WAIT_INPUT</b>	Input	Status	Label					
	IN 12	ON	Waiting for jar					
	Time (s)	Label						
	0,50	Waiting for filling						
<b>TEST_DIN</b>	Input	Value	Label					
	IN 14	OFF	Check plug application area not occupied					
<b>MOVE_REL</b>	Position (mm)	Tot. time(s)	Speed (mm/s)	T. Acc (s)	T. Dec (s)	Current (A)	Smooth	Label
	100,00	2,20	50,00	0,20	0,20	6,00	0	Arrival in the filling area
<b>TEST_DIN</b>	Input	Value	Label					
	IN 13	OFF	Check plug area not occupied					
<b>WAIT_TIME</b>	Time (s)	Label						
	0,50	Waiting for filling						

Once the jar is full, it must be positioned in the delivery area where the lid is applied.

Since the jar is no longer empty, the speed must obviously be slowed down to prevent the contents from spilling out.

The MOVE\_REL command is inserted with the following specifications: Position = 100.00 mm, Tot Time = 1.00 s, Acc T. = Dec T. = 0.50 s and comment: Arrival at Lid Area. A pause time must be set to allow time to the system to apply the lid.

The following step is therefore WAIT\_TIME: Time = 0.50 s, Comment: Wait Lid Application.

Program Name:								
Cycle #0	Cycle #1	Cycle #2	Cycle #3	Cycle #4	Cycle #5	Cycle #6	Cycle #7	
<input checked="" type="checkbox"/>	Homing Needed		Cycle Description					Filling and packaging cycle 10 pcs
<b>WAIT_INPUT</b>	Input	Status	Label					
	IN 12	ON	Waiting for jar					
<b>TEST_DIN</b>	Input	Value	Label					
	IN 13	OFF	Check plug area not occupied					
<b>TEST_DIN</b>	Input	Value	Label					
	IN 14	OFF	Check plug application area not occupied					
<b>MOVE_REL</b>	Position (mm)	Tot. time(s)	Speed (mm/s)	T. Acc (s)	T. Dec (s)	Current (A)	Smooth	Label
	100,00	2,20	50,00	0,20	0,20	6,00	0	Arrival in the filling area
<b>TEST_DIN</b>	Input	Value	Label					
	IN 14	OFF	Check plug application area not occupied					
<b>WAIT_TIME</b>	Time (s)	Label						
	0,50	Waiting for filling						
<b>MOVE_REL</b>	Position (mm)	Tot. time(s)	Speed (mm/s)	T. Acc (s)	T. Dec (s)	Current (A)	Smooth	Label
	100,00	4,20	25,00	0,20	0,20	6,00	0	Arrival in the stopper application area
<b>WAIT_TIME</b>	Time (s)	Label						
	0,50	Waiting for cap application						

At this point the jar is removed by an actuator installed outside the system. Certain conditions must be met before returning to the initial position. The IN 14 digital input needs to be turned off, while IN 13 and IN 12 must already be turned off. To return to the initial position, you only need to enter a further MOVE\_REL-associated step with the following specifications: Position = -200.00 mm, Tot Time = 1.00 s, Acc T. = Dec T. = 0.50 s, Comment = Return to Jar Wait zone.

Program Name:

Cycle #0 | Cycle #1 | Cycle #2 | Cycle #3 | Cycle #4 | Cycle #5 | Cycle #6 | Cycle #7

Homing Needed    Cycle Description: Filling and packaging cycle 10 pcs

	Input	Value	Label
<b>TEST_DIN</b>	IN 13	OFF	Check plug area not occupied
<b>MOVE_REL</b>	Position (mm)	100,00	Tot. time(s) 2,20
	Speed (mm/s)	50,00	T. Acc (s) 0,20
	T. Dec (s)	0,20	Current (A) 6,00
	Smooth	0	Label: Arrival in the filling area
<b>TEST_DIN</b>	IN 14	OFF	Check plug application area not occupied
<b>WAIT_TIME</b>	Time (s)	0,50	Label: Waiting for filling
<b>MOVE_REL</b>	Position (mm)	100,00	Tot. time(s) 4,20
	Speed (mm/s)	25,00	T. Acc (s) 0,20
	T. Dec (s)	0,20	Current (A) 6,00
	Smooth	0	Label: Arrival in the stopper application area
<b>WAIT_TIME</b>	Time (s)	0,50	Label: Waiting for cap application
<b>WAIT_INPUT</b>	IN 14	ON	Label: Waiting for jar removal from an external actuator
<b>TEST_DIN</b>	IN 12	OFF	Label: Check position for start unoccupied cycle
<b>TEST_DIN</b>	IN 13	OFF	Label: Check filling area not occupied
<b>MOVE_REL</b>	Position (mm)	-200,00	Tot. time(s) 2,20
	Speed (mm/s)	100,00	T. Acc (s) 0,20
	T. Dec (s)	0,20	Current (A) 6,00
	Smooth	0	Label: Return to initial position

Now the system plans the execution 10 repetitions of this cycle. First insert the repetition command through the usual procedure: click last record entered with the right mouse button and select the menu item: InsertStep->Enqueue.

Program Name:

Cycle #0 | Cycle #1 | Cycle #2 | Cycle #3 | Cycle #4 | Cycle #5 | Cycle #6 | Cycle #7

Homing Needed    Cycle Description: Filling and packaging cycle 10 pcs

	Position (mm)	Tot. time(s)	Speed (mm/s)	T. Acc (s)	T. Dec (s)	Current (A)	Smooth	Label
<b>MOVE_REL</b>	100,00	2,20	50,00	0,20	0,20	6,00	0	Arrival in the filling area
<b>TEST_DIN</b>	IN 14	OFF	Label: Check plug application area not occupied					
<b>WAIT_TIME</b>	Time (s)	0,50	Label: Waiting for filling					
<b>MOVE_REL</b>	100,00	4,20	25,00	0,20	0,20	6,00	0	Arrival in the stopper application area
<b>WAIT_TIME</b>	Time (s)	0,50	Label: Waiting for cap application					
<b>WAIT_INPUT</b>	IN 14	ON	Label: Waiting for jar removal from an external actuator					
<b>TEST_DIN</b>	IN 12	OFF	Label: Check position for start unoccupied cycle					
<b>TEST_DIN</b>	IN 13	OFF	Label: Check filling area not occupied					
<b>MOVE_REL</b>	-200,00	2,20	100,00	0,20	0,20	6,00	0	Return to initial position
<b>REPEAT</b>	Counter	10	Label: Work cycle repetition					



All the steps present will be inserted into the REPEAT command one at a time. Select the first step from the top with the left mouse button and drag it to the REPEAT record.  
When you release the mouse button, the step will be moved inside the REPEAT record.

Program Name:

Cycle #0 Cycle #1 Cycle #2 Cycle #3 Cycle #4 Cycle #5 Cycle #6 Cycle #7

Homing Needed    Cycle Description: Filling and packaging cycle 10 pcs

Input	Value	Label
TEST_DIN	IN 14 OFF	Check plug application area not occupied
WAIT_TIME	0,50	Waiting for filling
MOVE_REL	Position (mm): 100,00    Tot. time(s): 4,20    Speed (mm/s): 25,00    T. Acc (s): 0,20    T. Dec (s): 0,20    Current (A): 6,00    Smooth: 0	Arrival in the stopper application area
WAIT_TIME	0,50	Waiting for cap application
WAIT_INPUT	IN 14 ON	Waiting for jar removal from an external actuator
TEST_DIN	IN 12 OFF	Check position for start unoccupied cycle
TEST_DIN	IN 13 OFF	Check filling area not occupied
MOVE_REL	Position (mm): -200,00    Tot. time(s): 2,20    Speed (mm/s): 100,00    T. Acc (s): 0,20    T. Dec (s): 0,20    Current (A): 6,00    Smooth: 0	Return to initial position
REPEAT	Counter: 10	Work cycle repetition
WAIT_INPUT	IN 12 ON	Waiting for jar

Execute this operation for all the steps up to MOVE\_REL instruction, always starting from the top step, so as to maintain the logical order of execution of the cycle.

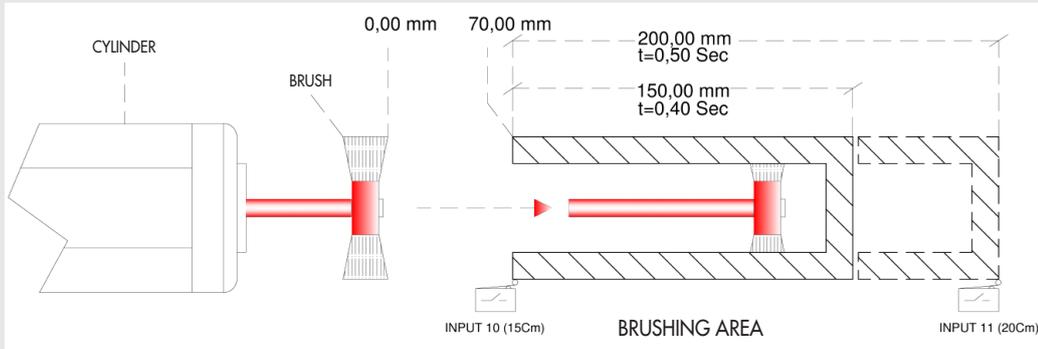
Input	Value	Label
TEST_DIN	IN 13 OFF	Check plug area not occupied
MOVE_REL	Position (mm): 100,00    Tot. time(s): 2,20    Speed (mm/s): 50,00    T. Acc (s): 0,20    T. Dec (s): 0,20    Current (A): 6,00    Smooth: 0	Arrival in the filling area
TEST_DIN	IN 14 OFF	Check plug application area not occupied
WAIT_TIME	0,50	Waiting for filling
MOVE_REL	Position (mm): 100,00    Tot. time(s): 4,20    Speed (mm/s): 25,00    T. Acc (s): 0,20    T. Dec (s): 0,20    Current (A): 6,00    Smooth: 0	Arrival in the stopper application area
WAIT_TIME	0,50	Waiting for cap application
WAIT_INPUT	IN 14 ON	Waiting for jar removal from an external actuator
TEST_DIN	IN 12 OFF	Check position for start unoccupied cycle
TEST_DIN	IN 13 OFF	Check filling area not occupied
MOVE_REL	Position (mm): -200,00    Tot. time(s): 2,20    Speed (mm/s): 100,00    T. Acc (s): 0,20    T. Dec (s): 0,20    Current (A): 6,00    Smooth: 0	Return to initial position

The final result is illustrated in the figure.



## EXAMPLE 2: Brushing unit

This application concerns the brushing of objects of a different height. Two digital inputs are provided. Their activation depends on the object being handled. Input IN 10 is activated when the object is at least 150 mm high and input IN 11 when it is at least 200 mm high. The e.drive must provide two sets of instructions to be adapted to the selected input.

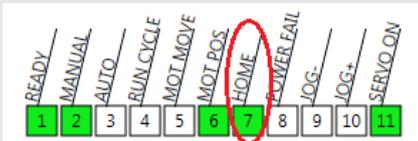


Two cycles are created: one for the zeroing, one for actual operation. Please refer to the Setup->Cycle section for further details on the execution sequence of the various cycles.

For Cycle #0, the comment "Zeroing cycle" is inserted as a description.

The cycle consists of a single MOVE\_HOME instruction, whose comment is: Execute Zeroing.

**In this type of cycle, it is very important not to check the Request Homing box. The "HOME flag" status activates only on completion of zeroing.**



This flag is a mandatory condition for executing the cycle when the Request Homing box has been checked.

Program Name: Test cycle

Cycle #0 Cycle #1 Cycle #2 Cycle #3 Cycle #4 Cycle #5 Cycle #6 Cycle #7

Homing Needed Cycle Description Zero cycle

Label	Performs reset
MOVE_HOME	

The work cycle: Cycle #1, is now implemented.

The Request Homing box must be checked to ensure the execution of the cycle starting from a specific, unambiguous reference position: the axis zero.

The cycle description here is: Cycle for the Brushing Unit.

The first operation is to move to the Brushing Unit area, which is 70.00 mm distant from the zero.

The MOVE\_REL command is enabled with the following specifications: Position = 70.00 mm, Tot Time = 0.40 s, Acc T. = Dec T. = 0.20 s.

Comment: Arrival at Brushing Unit area.

Program Name: 0

Cycle #0 Cycle #1 Cycle #2 Cycle #3 Cycle #4 Cycle #5 Cycle #6 Cycle #7

Homing Needed Cycle Description

	Position (mm)	Tot. time(s)	Speed (mm/s)	T. Acc (s)	T. Dec (s)	Current (A)	Smooth	Label
MOVE_REL	70,00	0,40	350,00	0,20	0,20	6,00	0	Arrival in the filling area
WAIT_INPUT	IN 10	ON						Waiting input 15cm height

When the digital input IN 10 activates, it means that the photoelectric cell has detected the presence of an object.

Step WAIT\_INPUT is used precisely for this operation. A test is performed on digital input IN 11 via the TEST\_DIN step to select which of the two possible objects are to be brushed.

Program Name: 0

Cycle #0 Cycle #1 Cycle #2 Cycle #3 Cycle #4 Cycle #5 Cycle #6 Cycle #7

Homing Needed    Cycle Description

	Position (mm)	Tot. time(s)	Speed (mm/s)	T. Acc (s)	T. Dec (s)	Current (A)	Smooth	Label
<b>MOVE_REL</b>	70,00	0,40	350,00	0,20	0,20	6,00	0	Arrival in the filling area
<i>Input                      Status                      Label</i>								
<b>WAIT_INPUT</b>	IN 10	ON	Waiting input 15cm height					
<i>Input                      Value                      Label</i>								
<b>TEST_DIN</b>	IN 11	ON	20cm height control					

If the test result on IN 11 is positive, the brushing unit requests a dozen passes to remove any burrs on the object of a height greater than or equal to 200 mm. A REPEAT step, in which the brushing cycle is run, is added to the TEST\_DIN step.

Considering the number of required repetitions, 10 is entered in the Counter field, with the comment: Execute Brushing Unit.

Program Name: 0

Cycle #0 Cycle #1 Cycle #2 Cycle #3 Cycle #4 Cycle #5 Cycle #6 Cycle #7

Homing Needed    Cycle Description

	Position (mm)	Tot. time(s)	Speed (mm/s)	T. Acc (s)	T. Dec (s)	Current (A)	Smooth	Label
<b>MOVE_REL</b>	70,00	0,40	350,00	0,20	0,20	6,00	0	Arrival in the filling area
<i>Input                      Status                      Label</i>								
<b>WAIT_INPUT</b>	IN 10	ON	Waiting input 15cm height					
<i>Input                      Value                      Label</i>								
<b>TEST_DIN</b>	IN 11	ON	20cm height control					
<i>Counter                      Label</i>								
<b>REPEAT</b>		10	Performs brushing					

Two MOVE\_REL instructions are provided in sequence for the brushing cycle, with Position = 200.00 mm for the first instruction and Position = -200.00 mm for the second, with Tot Time = 0.5 s and Acc T. = Dec T. = 0.10 s for both. The instructions are then dragged inside the REPEAT instruction.

Program Name: 0

Cycle #0 Cycle #1 Cycle #2 Cycle #3 Cycle #4 Cycle #5 Cycle #6 Cycle #7

Homing Needed    Cycle Description

	Position (mm)	Tot. time(s)	Speed (mm/s)	T. Acc (s)	T. Dec (s)	Current (A)	Smooth	Label
<b>MOVE_REL</b>	70,00	0,40	350,00	0,20	0,20	6,00	0	Arrival in the filling area
<i>Input                      Status                      Label</i>								
<b>WAIT_INPUT</b>	IN 10	ON	Waiting input 15cm height					
<i>Input                      Value                      Label</i>								
<b>TEST_DIN</b>	IN 11	ON	20cm height control					
<i>Counter                      Label</i>								
<b>REPEAT</b>		10	Performs brushing					
<i>Position (mm)    Tot. time(s)    Speed (mm/s)    T. Acc (s)    T. Dec (s)    Current (A)    Smooth    Label</i>								
<b>MOVE_REL</b>	200,00	1,00	250,00	0,20	0,20	6,00	0	
<b>MOVE_REL</b>	-200,00	1,00	250,00	0,20	0,20	6,00	0	

If the test result on IN 11 is negative, the brushing unit requests a dozen passes to remove any burrs on the object of a height greater than or equal to 150 mm and less than 200 mm. Considering that the instruction sequence is the one just enabled, you can enter the second part of the cycle with the Copy-Paste function, by simply selecting the "father" instruction, clicking the left mouse button and simultaneously holding the keyboard Ctrl button down.

Program Name: 0

Cycle #0 Cycle #1 Cycle #2 Cycle #3 Cycle #4 Cycle #5 Cycle #6 Cycle #7

Homing Needed    Cycle Description

	Position (mm)	Tot. time(s)	Speed (mm/s)	T. Acc (s)	T. Dec (s)	Current (A)	Smooth	Label
<b>MOVE_REL</b>	70,00	0,40	350,00	0,20	0,20	6,00	0	Arrival in the filling area
<b>WAIT_INPUT</b>	Input: IN 10	Status: ON	Label: Waiting input 15cm height					
<b>TEST_DIN</b>	Input: IN 11	Value: ON	Label: 20cm height control					
<b>REPEAT</b>	Counter: 10	Label: Performs brushing						
<b>MOVE_REL</b>	200,00	1,00	250,00	0,20	0,20	6,00	0	
<b>MOVE_REL</b>	-200,00	1,00	250,00	0,20	0,20	6,00	0	
<b>TEST_DIN</b>	Input: IN 11	Value: ON	Label: 20cm height control					

At this point, you only need to modify the just added TEST\_DIN instruction.  
 In the MOVE\_REL instructions, simply change the parameters related to the stroke to be covered (Position = 150.00 mm) and the speed at which the movement must be performed. To change the speed parameter, being a position movement, you only need to change the total movement execution time: Tot Time = 0.40 s.

	Position (mm)	Tot. time(s)	Speed (mm/s)	T. Acc (s)	T. Dec (s)	Current (A)	Smooth	Label
<b>MOVE_REL</b>	70,00	0,40	350,00	0,20	0,20	6,00	0	Arrival in the filling area
<b>WAIT_INPUT</b>	Input: IN 10	Status: ON	Label: Waiting input 15cm height					
<b>TEST_DIN</b>	Input: IN 11	Value: ON	Label: 20cm height control					
<b>REPEAT</b>	Counter: 10	Label: Performs brushing						
<b>MOVE_REL</b>	200,00	1,00	250,00	0,20	0,20	6,00	0	
<b>MOVE_REL</b>	-200,00	1,00	250,00	0,20	0,20	6,00	0	
<b>TEST_DIN</b>	Input: IN 11	Value: ON	Label: 20cm height control					
<b>REPEAT</b>	Counter: 10	Label: Performs brushing						
<b>MOVE_REL</b>	150,00	0,80	250,00	0,20	0,20	6,00	0	
<b>MOVE_REL</b>	-150,00	0,80	250,00	0,20	0,20	6,00	0	

The cycle is completed when the object is removed, i.e. the digital input IN 10 is OFF.  
 If you select the last TEST\_DIN step and click Enter with the right mouse button, the WAIT\_INPUT step is enqueued by setting as digital input IN 10, Status = OFF and Comment: Wait exit.

Program Name: 0

Cycle #0 Cycle #1 Cycle #2 Cycle #3 Cycle #4 Cycle #5 Cycle #6 Cycle #7

Homing Needed    Cycle Description

<b>WAIT_INPUT</b>	Input: IN 10	Status: ON	Label: Waiting input 15cm height					
<b>TEST_DIN</b>	Input: IN 11	Value: ON	Label: 20cm height control					
<b>REPEAT</b>	Counter: 10	Label: Performs brushing						
<b>MOVE_REL</b>	200,00	1,00	250,00	0,20	0,20	6,00	0	
<b>MOVE_REL</b>	-200,00	1,00	250,00	0,20	0,20	6,00	0	
<b>TEST_DIN</b>	Input: IN 11	Value: ON	Label: 20cm height control					
<b>REPEAT</b>	Counter: 10	Label: Performs brushing						
<b>MOVE_REL</b>	150,00	0,80	250,00	0,20	0,20	6,00	0	
<b>MOVE_REL</b>	-150,00	0,80	250,00	0,20	0,20	6,00	0	
<b>WAIT_INPUT</b>	Input: IN 10	Status: OFF	Label: Waiting for exit					

The last step is related to the return to the initial position, i.e. the product wait position. You only need to enter a MOVE\_REL instruction with the following characteristics: Position = -70.00 mm, Tot Time = 0.25 s, Acc T. = Dec T. = 0.10 s and Comment: Return to Wait Position.

Program Name: 0

Cycle #0 Cycle #1 Cycle #2 Cycle #3 Cycle #4 Cycle #5 Cycle #6 Cycle #7

Homing Needed      Cycle Description

Input	Value	Label
TEST_DIN	IN 11	ON
		20cm height control
Counter		
	10	Performs brushing
REPEAT		
MOVE_REL	Position (mm)	Tot. time(s)
	200,00	1,00
	Speed (mm/s)	T. Acc (s)
	250,00	0,20
	T. Dec (s)	Current (A)
	0,20	6,00
	Smooth	Label
	0	
MOVE_REL	Position (mm)	Tot. time(s)
	-200,00	1,00
	Speed (mm/s)	T. Acc (s)
	250,00	0,20
	T. Dec (s)	Current (A)
	0,20	6,00
	Smooth	Label
	0	
Input		
TEST_DIN	IN 11	ON
		20cm height control
Counter		
	10	Performs brushing
REPEAT		
MOVE_REL	Position (mm)	Tot. time(s)
	150,00	0,80
	Speed (mm/s)	T. Acc (s)
	250,00	0,20
	T. Dec (s)	Current (A)
	0,20	6,00
	Smooth	Label
	0	
MOVE_REL	Position (mm)	Tot. time(s)
	-150,00	0,80
	Speed (mm/s)	T. Acc (s)
	250,00	0,20
	T. Dec (s)	Current (A)
	0,20	6,00
	Smooth	Label
	0	
Input		
WAIT_INPUT	IN 10	OFF
		Waiting for exit
MOVE_REL		
	Position (mm)	Tot. time(s)
	-70,00	0,4
	Speed (mm/s)	T. Acc (s)
	350,00	0,20
	T. Dec (s)	Current (A)
	0,20	6,00
	Smooth	Label
	0	

The end result is illustrated in the figure.

### EXAMPLE 3: Humidity control

This application concerns the control of the humidity level in a room. An analogue input (0-10 V) is provided. It is connected to a sensor that measures the humidity in the room, which must be kept at around 60%. If there is a change in humidity, the system reacts by opening or closing the window in the room to ensure the exchange of air.

The intention is to activate the MOVE\_CNT instruction so as to use the analogue input of the humidity sensor as the feedback for controlling the opening of the window.

First of all, the correct parameters for acquiring an analogue signal must be set in the SETUP dialogue box. In the Analogue box, analogue input 1 is configured with Position function by setting the following values: Vmin = 6.00 V and Vmax = 10 V.

Inputs	Function	Input	Vmin (V)	Vmax (V)	Min (mm)	Max (mm)	Filter
1	POSITION	Input n. 1	0,00	10,00	0,00	300,00	10
2	SPEED	Input n. 2	0,00	10,00	0,00	10,00	10

When a signal falling within the set range (6.00/10.00 V) is measured, this is converted in real time into an analogue command of the actuator position between 0.00 mm and 300.00 mm (maximum stroke of the actuator defined in the Setup parameters).

Two types of cycle are provided in the PROGRAM dialogue box: a zeroing cycle and a work cycle.

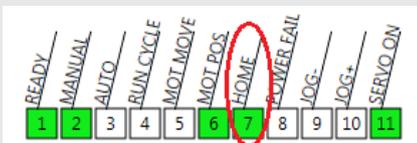
Please refer to the Setup->Cycle for details on the sequence of execution of the various cycles.

Program Name: Humidity Control.

For **Cycle #0** the comment "zeroing cycle" is inserted as a description.

The cycle consists of a single MOVE\_HOME instruction, whose comment is: Execute Zeroing.

**In this type of cycle, it is very important not to check the Request Homing box. The HOME flag status comes on only upon completion of zeroing.**



**This flag is a mandatory condition for executing the cycle when the Request Homing box has been checked.**

Program Name:

Cycle #0 | Cycle #1 | Cycle #2 | Cycle #3 | Cycle #4 | Cycle #5 | Cycle #6 | Cycle #7

Homing Needed    Cycle Description:

Label	Comment
MOVE_HOME	Performs reset

The work cycle: Cycle #1 is now implemented.

The Request Homing box must be checked to ensure the execution of the cycle starting from a specific, unambiguous reference position: the axis zero. The cycle description here is: Actuator control for room humidity.

The MOVE\_REL command is enabled with the following specifications:

Position = AN1, after AN has been selected in the left drop-down menu, Speed = 10.00 mm/s, T.Dec = T.Acc = 0.50 s and Comment: Humidity Control.

Program Name:

Cycle #0 | Cycle #1 | Cycle #2 | Cycle #3 | Cycle #4 | Cycle #5 | Cycle #6 | Cycle #7

Homing Needed    Cycle Description:

MOVE_CNT	Position (mm)	Speed (mm/s)	T. Acc (s)	T. Dec (s)	Current (A)	Smooth	Label
	AN1	10,00	0,20	0,20	6,00	0	Humidity control

The MOVE\_CNT is a locking command and can be terminated by a STOP\_GLOBAL digital input.

## REFERENCES

<http://www.metalwork.it/eng/elektro/index.html>

<http://copperhilltech.com/a-comprehensible-guide-to-servo-motor-sizing/>