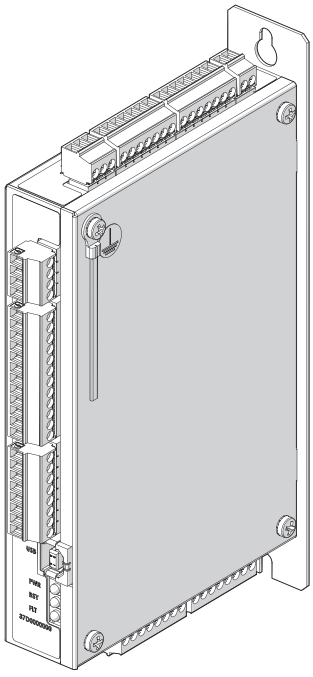
METAL WORK PNEUMATIC



MANUAL *e*.motion

EN

EN

INDEX

EN

SAFETY

The following must be borne in mind when using a component of Metal Work ***e***.motion system:

- ⚠ **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 ELECTROCUTION!**
  Voltage and current in the electric system of the module are below the risk threshold levels and can be considered in the ELV class.
  The module is free from internal galvanic insulation and hence external hazardous voltages can propagate unexpectedly.
  The module does not galvanically insulate the various components. The voltages of interlinked drives often exceed the ELV threshold.
  Hazardous voltages can be present even when an electrical wire comes into contact with external hazardous voltages.

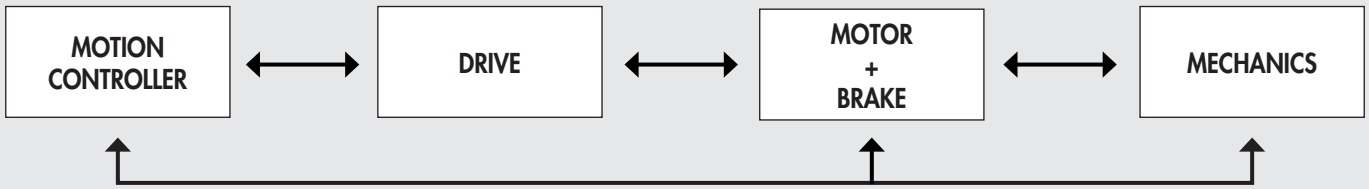
- ⚠ **CAUTION: RISK OF FIRE AND BURNS!**
  The ***e***.motion module does not protect the motor, drive or mechanical components against overload (e.g. excessive force).

- ⚠ **CAUTION: RISK OF UNWANTED MOVEMENTS OR FAILED STOP!**
  The mechanical brake cannot be considered a safety device.
  The motion controller does not guarantee that the axis does not move, even in the absence of logical running conditions.
  Unwanted start of the actuation cannot be ruled out completely.

EN

INTRODUCTION

As part of its policy aimed at providing an increasingly comprehensive and efficient customer service, Metal Work offers an easy-to-run and cost-effective programmable motion controller.
e.motion was designed and developed to facilitate the use of all sizes of electric actuators equipped with motors that can be controlled by a pulse train, irrespective of whether they are the BRUSHLESS or STEPPER type.
The module needs to communicate with the drive supplied with the motor and is equipped with an intuitive software. 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^2), deceleration (m/s^2), 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). Any type of drive provided with Direction and Step inputs can be used with ***e***.motion.
3. **MOTOR**: motors can be either BRUSHLESS or STEPPER and are supplied with the type of drive specifically designed for that motor. 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.

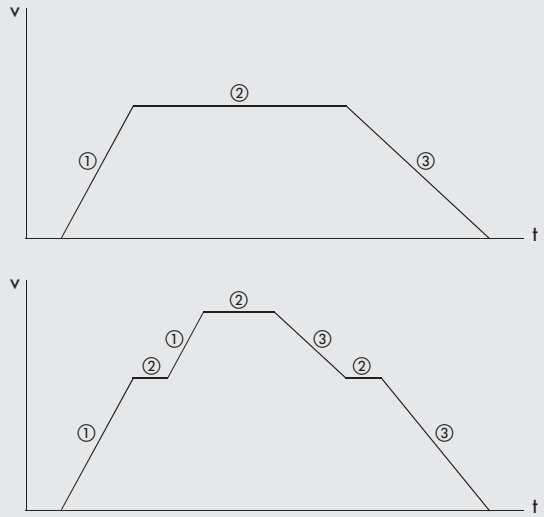
SOME USEFUL CONCEPTS

LAW OF MOTION

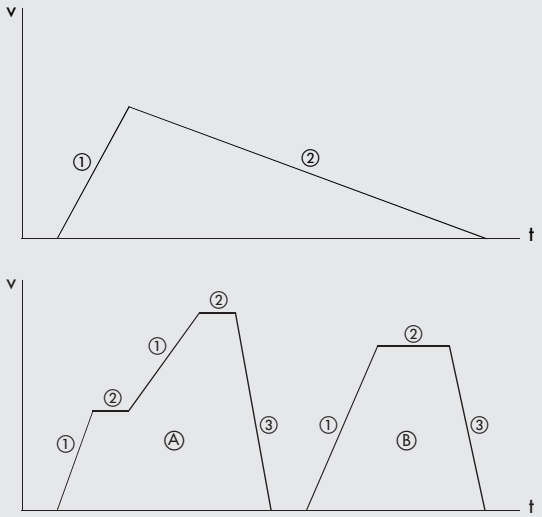
- **Kinematics**: is the study of the motion of objects (space, time, speed and acceleration).
- **Dynamics**: is the study of the causes of motion, such as forces and torques.
- **Statics**: is the study of the balance of objects. It considers the conditions that allow objects to remain still (or prevent 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

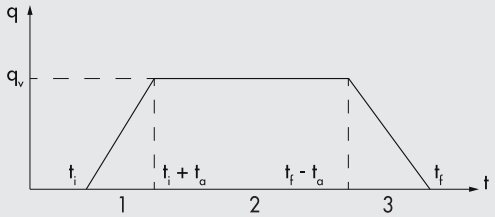


Law of triangular motion and multiple laws Ⓐ, Ⓑ



① Acceleration ② Constant speed phase ③ Deceleration Ⓐ 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.



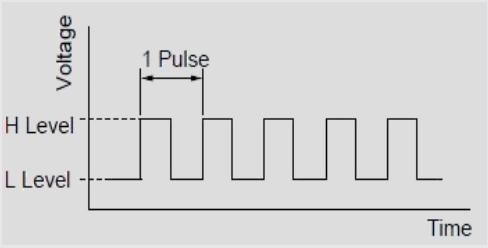
The trajectory is, therefore, divided into three parts:
1. Constant acceleration, ramp speed, parabolic position;
2. Null acceleration, constant speed, linear position;
3. Constant deceleration, ramp speed, parabolic position.

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.

EN

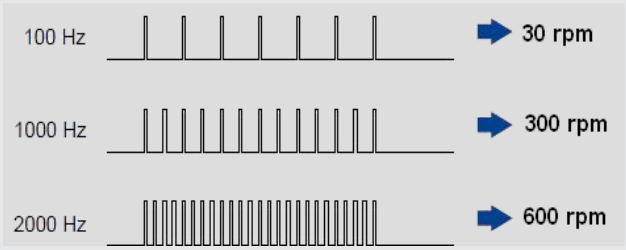
STEP OR PULSE

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.

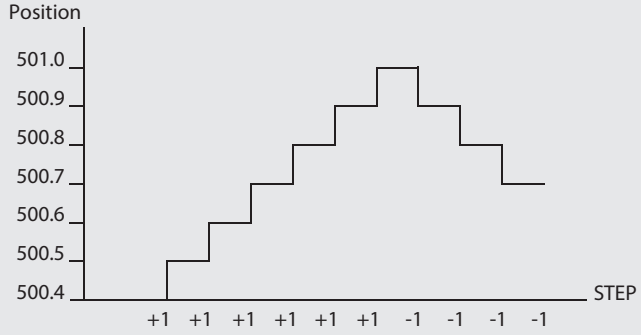
The pulse frequency (number of pulses per second measured in Hertz "Hz") determines the motor rotation speed.



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

EN

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° 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.
In the case of BRUSHLESS motors, accuracy depends greatly on the encoder resolution and the internal setting parameters (e.g. the PID control).

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 200mm 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 peak torque in BRUSHLESS motors can be 3-4 times the rated torque.
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**.motion 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. Brake release can be controlled from the drive or the **e**.motion controller.

e.motion CONTROLLER SPECIFICATIONS

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÷14	mA	-	400
Output current 15	mA	-	600
Maximum voltage per output	VDC	18	30
Protection class	-	IP 20	
Operating temperature	°C	0	50
Relative humidity (Non-condensing)		10%	90%
Dimensions L x H x P	mm	180x99x30	
Weight	g	460	

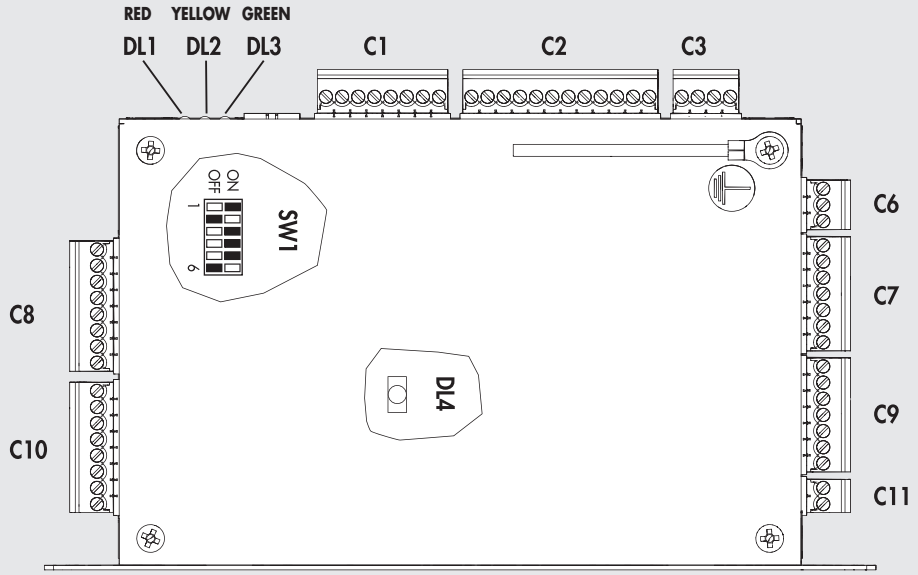
Table 1

* 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.

EN

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

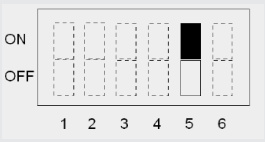
DIP-SWITCH SETTINGS

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

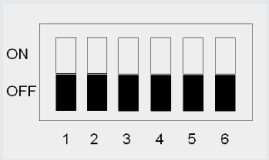
Ref	Parameter	Position	Default
DIP1	Foreseen to be connected to RS485 (termination inserting).	ON = termination enabled OFF = termination disabled	OFF = disabled
DIP2	Defines the 24V or 5V encoder input (A). Refer to encoder user manual or technical specifications.	ON = encoder powered at 5V OFF = encoder 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 (Z). 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.motion 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

EN

KEY TO Leds

e.motion 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.

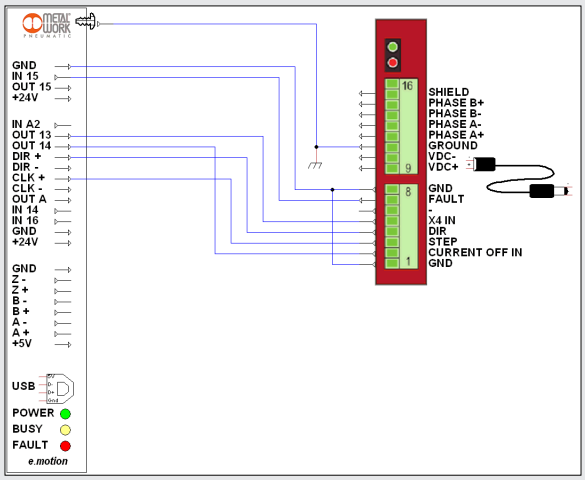
Led	State	Description
DL1 (Fault)	○ **OFF**	No FAULT
	● **ON** (Red)	Not used
	✺ **FLASHING** (Red)	Fault
DL2 (Enable motor driver)	○ **OFF**	Drive disabled
	● **ON** (Yellow)	Drive enabled or CPU "locked" for FW update
	✺ **FLASHING** (Yellow)	Not used
DL3 (Power-ON)	○ **OFF**	Board powered off or faulty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

Table 3

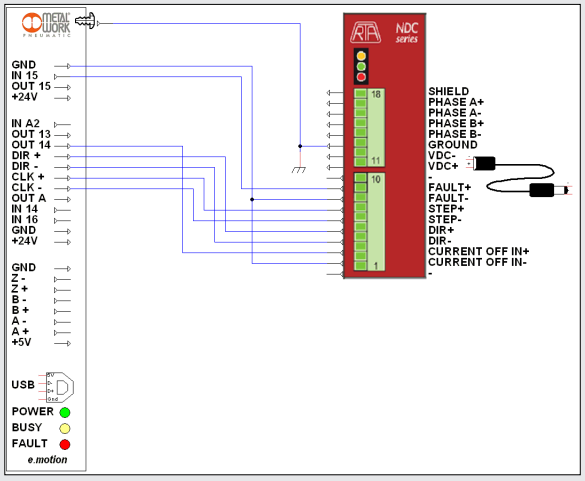
CONNECTION TO DRIVES FOR STEPPER MOTOR CONTROL

Below are several links and drives dedicated to STEPPER motor control, which are included in the Metal Work catalogue. The Dir and Step output signals generated by ***e***.motion are high-speed signals and, though they are a differential type, precautions must be taken to avoid problems.
Any electrical interference may generate step disturbance signals that lead to position errors and/or axis vibration, thus demanding greater energy consumption and increasing the generation of heat, thereby reducing the lifecycle of the electrical and mechanical system.
In order to prevent these disturbances, it is advisable:

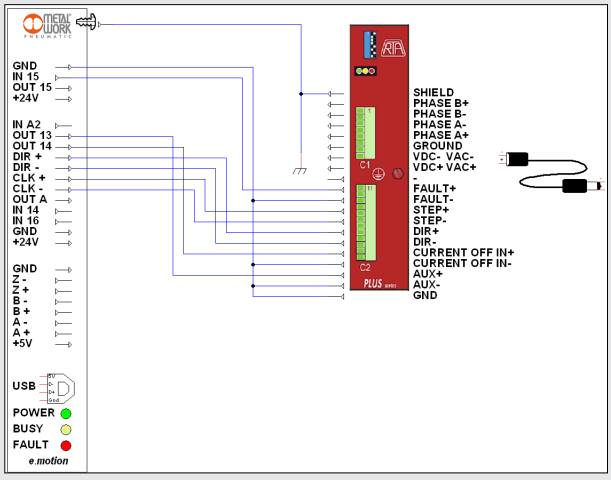
1. to connect the signals using shielded and twisted cables;
2. to earth the sheath at one end of the wiring, that of the ***e***.motion;
3. not to place the signal cables close to power cables (e.g. near motor cables).



Wiring with "RTA CSD 94" drive, code 37D1222000.
N.B.: *For details on configurations/settings, please refer to the corresponding manual.*



Wiring with "RTA NDC 96" drive, code 37D1332000.
N.B.: *For details on configurations/settings, please refer to the corresponding manual.*



Wiring with "RTA PLUS A4" drive, code 37D1442000 or "RTA PLUS B7" code 37D1552000.
N.B.: *For details on configurations/settings, please refer to the corresponding manual.*

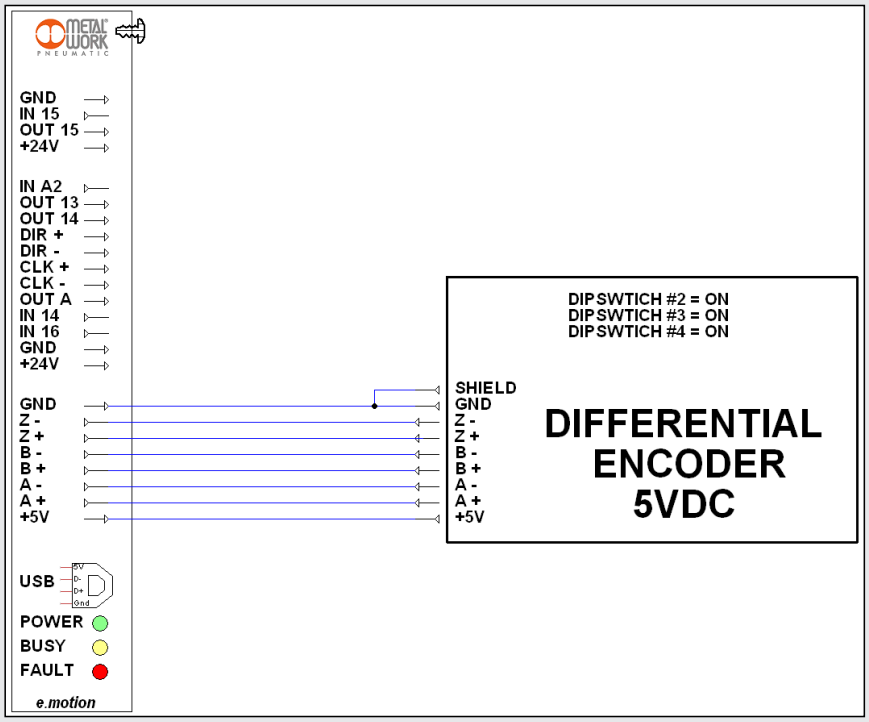
EN

ENCODER CONNECTION

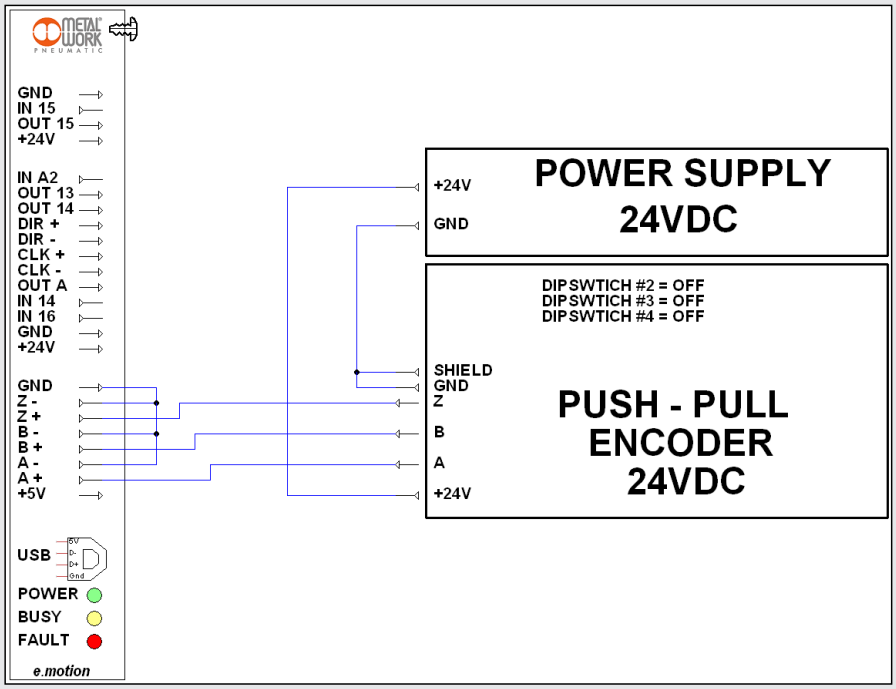
If an encoder is used, e.g. for motion diagnostics, special care must be taken with the wiring.
Encoder models powered at 5VDC can be connected directly to ***e***.motion.
The corresponding connector supplies the encoder with maximum 500 mA current.
Encoder models powered at 24VDC require an external power supply.
Before connecting the encoder to the ***e***.motion, 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 an ***e***.motion 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.

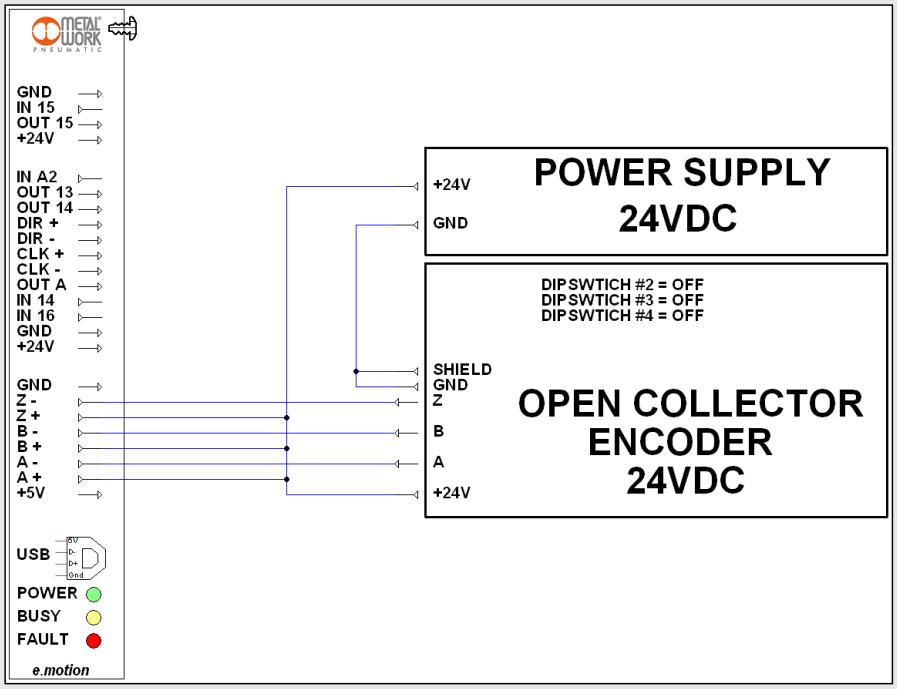
The connection for the three different types of encoder is illustrated in the figures below.



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



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

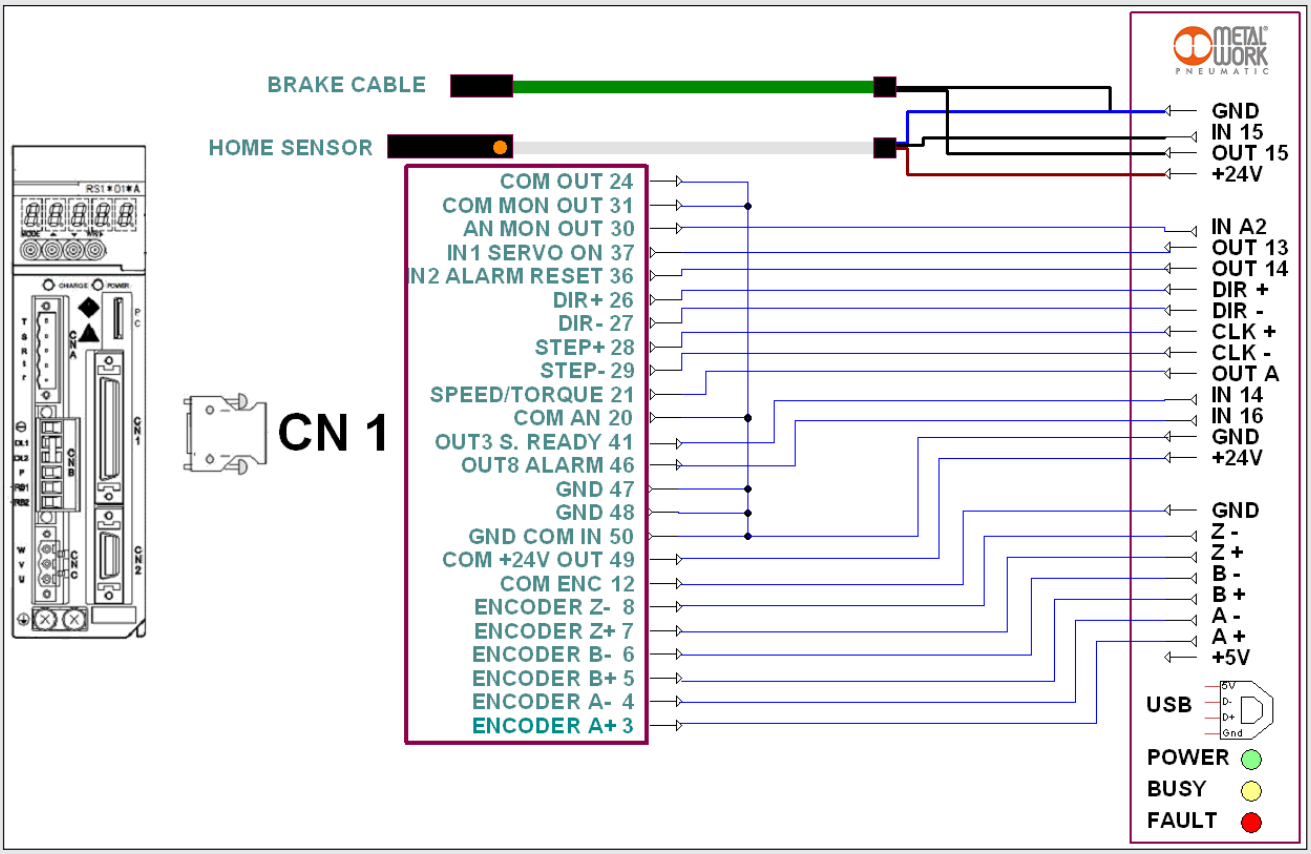


Wiring of an NPN open-collector encoder powered at 24VDC.

EN

CONNECTION TO A DRIVE "SANYO DENKI" FOR BRUSHLESS MOTOR CONTROL

The connection to a "Sanyo Denki RS1A01" code 37D2200000, "Sanyo Denki RS1A03" code 37D2400000, or "Sanyo Denki RS3A03" code 37D2400008 drive designed for controlling a Sanyo Denki BRUSHLESS motor, is illustrated below.
Dir driver and Step driver signals from the ***e***.motion are high-speed signals and, though they are the differential type, precautions must be taken to avoid problems.
Any electrical interference may generate step disturbance signals that lead to position errors and/or axis vibration, thus demanding greater energy consumption and increasing the generation of heat, thereby reducing the lifecycle of the electrical and mechanical system.
In order to prevent these disturbances, it is recommended:
1. to connect the signals using shielded and twisted cables;
2. to earth the sheath at one end of the wiring, that of the ***e***.motion;
3. not to place the signal cables close to power cables (e.g. near motor cables).

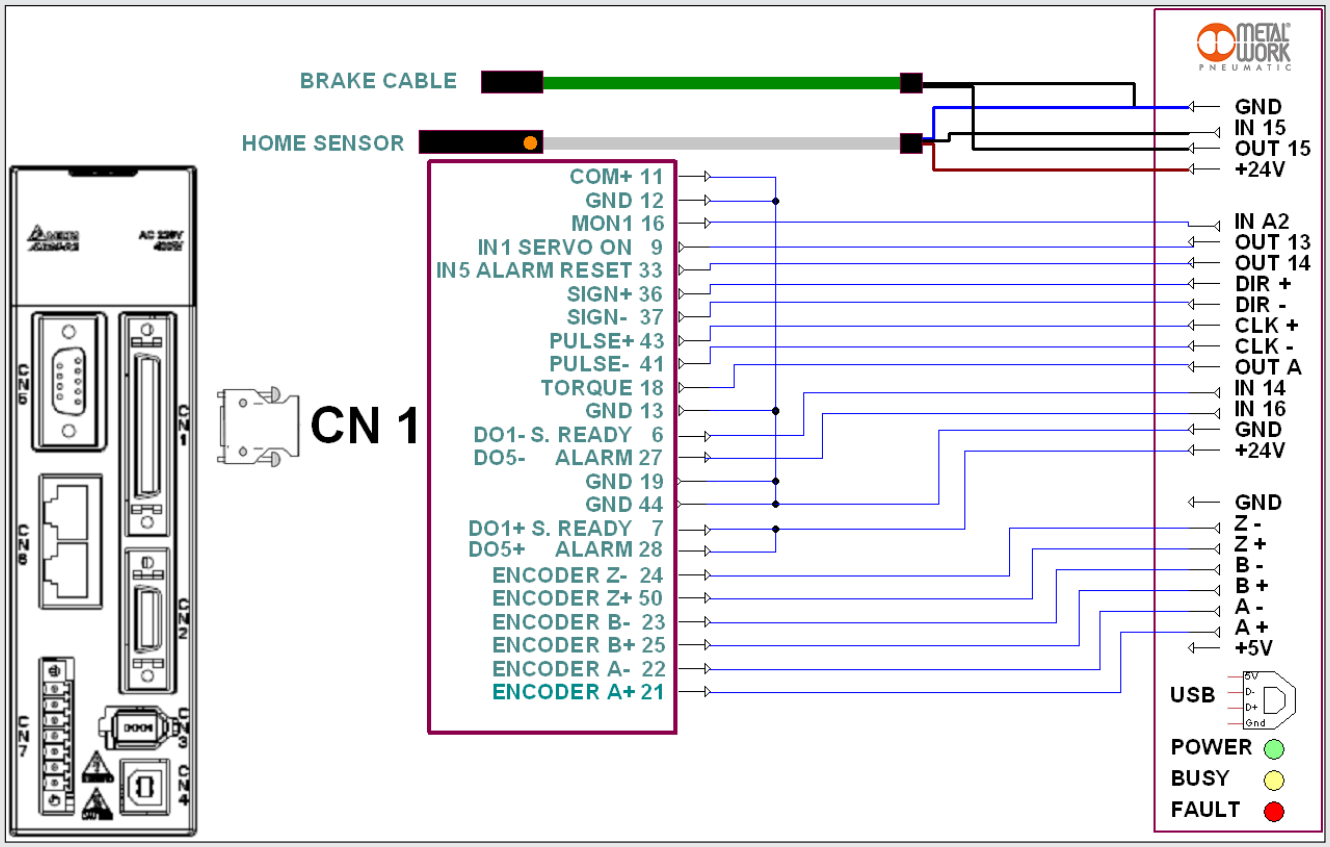


Example of the connection of ***e***.motion *to a "Sanyo Denki RS1___ or RS3___" driver using a cable code 37C2510000 to control a BRUSHLESS motor with brake.*

EN

CONNECTION TO A DRIVE "DELTA" FOR BRUSHLESS MOTOR CONTROL

The connection to a "Delta ASDA-A2-0221-M" code 37D2200001, "Delta ASDA-A2-0421-M" code 37D2300000 or "Delta ASDA-A2-3043-M" code 37D2600001 drive designed for controlling a Delta BRUSHLESS motor, is illustrated below.
Dir driver and Step driver signals from the ***e***.motion are high-speed signals and, though they are the differential type, precautions must be taken to avoid problems.
Any electrical interference may generate step disturbance signals that lead to position errors and/or axis vibration, thus demanding greater energy consumption and increasing the generation of heat, thereby reducing the lifecycle of the electrical and mechanical system.
In order to prevent these disturbances, it is recommended:
1. to connect the signals using shielded and twisted cables;
2. to earth the sheath at one end of the wiring, that of the ***e***.motion;
3. not to place the signal cables close to power cables (e.g. near motor cables).



Example of the connection of ***e***.motion *to a "Delta ASDA-A2_" driver using a cable code 37C2510001 to control a BRUSHLESS motor with brake.*

EN

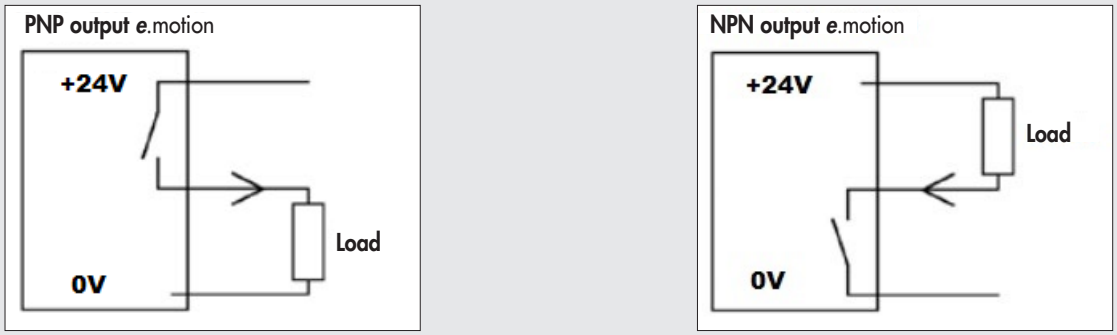
DIGITAL INPUT CONNECTION

The ***e***.motion has 16 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***.motion has 15 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).

EN

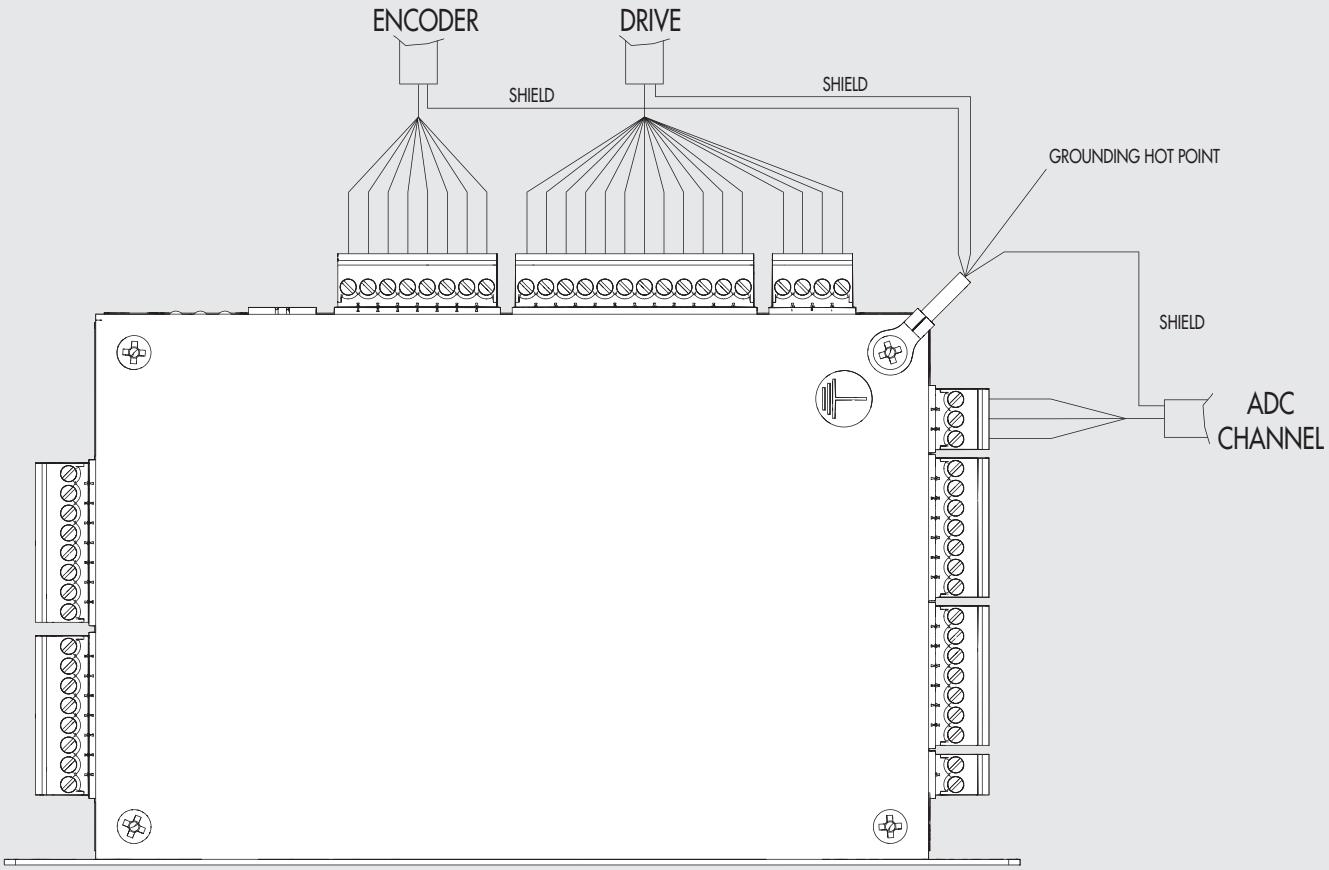
ANALOGUE INPUT

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 µs.
- Input impedance = 6.6 kOhm.
- 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***.motion 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***.motion 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.



EN

CONNECTORS

C1 ENCODER CONNECTOR

Pin	Cable cross-section	Type of terminal	Description
1	0.25 mm²	Screw clamping	5VDC 500 mA
2	0.25 mm²	Screw clamping	Phase A + encoder
3	0.25 mm²	Screw clamping	Phase A - encoder
4	0.25 mm²	Screw clamping	Phase B + encoder
5	0.25 mm²	Screw clamping	Phase B - encoder
6	0.25 mm²	Screw clamping	Phase Z + encoder
7	0.25 mm²	Screw clamping	Phase Z - encoder
8	0.25 mm²	Screw clamping	0V

C2 DRIVE CONNECTOR

Pin	Cable cross-section	Type of terminal	Description
1	0.25 mm²	Screw clamping	+24VDC for external devices
2	0.25 mm²	Screw clamping	0V for external devices
3	0.25 mm²	Screw clamping	Input 16 NPN-PNP programmable
4	0.25 mm²	Screw clamping	Input 14 NPN-PNP programmable
5	0.25 mm²	Screw clamping	Analogue output 0-10 VDC 20 mA
6	0.25 mm²	Screw clamping	Negative differential output STEP (5 V LINE DRIVER)
7	0.25 mm²	Screw clamping	Positive differential output STEP (5 V LINE DRIVER)
8	0.25 mm²	Screw clamping	Negative differential output DIRECTION (5 V LINE DRIVER)
9	0.25 mm²	Screw clamping	Positive differential output DIRECTION (5 V LINE DRIVER)
10	0.25 mm²	Screw clamping	14 PNP output
11	0.25 mm²	Screw clamping	13 PNP output
12	0.25 mm²	Screw clamping	Analogue input 2 0 - 10 VDC

C3 CONNECTOR FOR BRAKE

Pin	Cable cross-section	Type of terminal	Description
1	0.25 mm²	Screw clamping	+24VDC for external devices
2	0.25 mm²	Screw clamping	15 PNP output (0.6 A)
3	0.25 mm²	Screw clamping	Input 15 NPN-PNP programmable
4	0.25 mm²	Screw clamping	0V for external devices

C6 ANALOGUE INPUT CONNECTOR

Pin	Cable cross-section	Type of terminal	Description
1	0.25 mm²	Screw clamping	+10VDC 20 mA for external devices
2	0.25 mm²	Screw clamping	Analogue input 1 0 - 10VDC
3	0.25 mm²	Screw clamping	0V for external devices

C7 CONNECTOR - OUTPUTS 7-12

Pin	Cable cross-section	Type of terminal	Description
1	0.25 mm²	Screw clamping	0V for external devices
2	0.25 mm²	Screw clamping	12 PNP output
3	0.25 mm²	Screw clamping	11 PNP output
4	0.25 mm²	Screw clamping	10 PNP output
5	0.25 mm²	Screw clamping	09 PNP output
6	0.25 mm²	Screw clamping	08 PNP output
7	0.25 mm²	Serraggio a vite	07 PNP output

C8 CONNECTOR - INPUTS 8-13

Pin	Cable cross-section	Type of terminal	Description
1	0.25 mm²	Screw clamping	+24VDC for external devices
2	0.25 mm²	Screw clamping	Input 08 NPN-PNP programmable
3	0.25 mm²	Screw clamping	Input 09 NPN-PNP programmable
4	0.25 mm²	Screw clamping	Input 10 NPN-PNP programmable
5	0.25 mm²	Screw clamping	Input 11 NPN-PNP programmable
6	0.25 mm²	Screw clamping	Input 12 NPN-PNP programmable
7	0.25 mm²	Screw clamping	Input 13 NPN-PNP programmable
8	0.25 mm²	Screw clamping	0V for external devices

C9 CONNECTOR FOR OUTPUTS 1-6

Pin	Cable cross-section	Type of terminal	Description
1	0.25 mm²	Screw clamping	06 PNP output
2	0.25 mm²	Screw clamping	05 PNP output
3	0.25 mm²	Screw clamping	04 PNP output
4	0.25 mm²	Screw clamping	03 PNP output
5	0.25 mm²	Screw clamping	02 PNP output
6	0.25 mm²	Screw clamping	01 PNP output
7	0.25 mm²	Serraggio a vite	+24VDC for external devices

EN

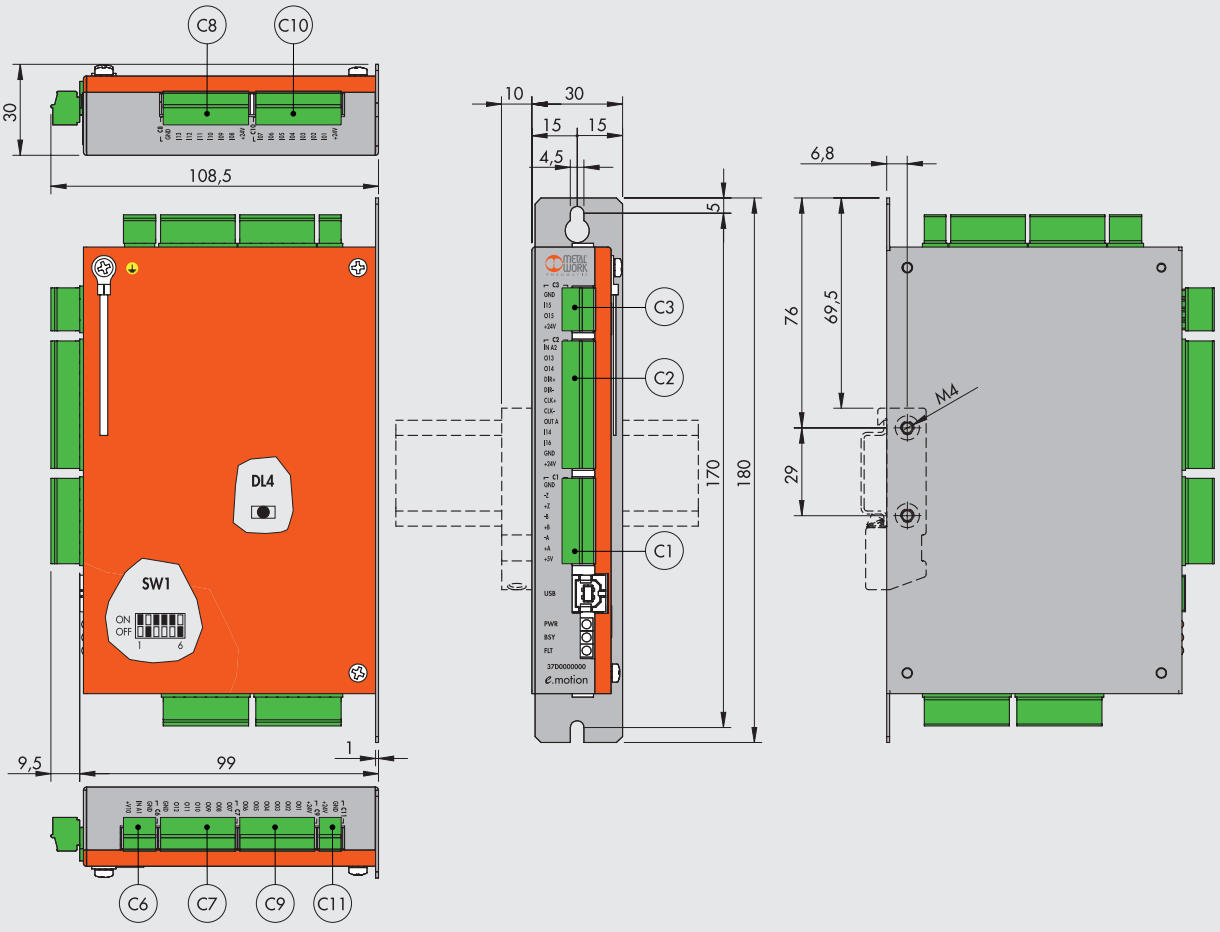
C10 CONNECTOR FOR INPUTS 1-7

Pin	Cable cross-section	Type of terminal	Description
1	0.25 mm²	Screw clamping	+24VDC for external devices
2	0.25 mm²	Screw clamping	Input 01 NPN-PNP programmable
3	0.25 mm²	Screw clamping	Input 02 NPN-PNP programmable
4	0.25 mm²	Screw clamping	Input 03 NPN-PNP programmable
5	0.25 mm²	Screw clamping	Input 04 NPN-PNP programmable
6	0.25 mm²	Screw clamping	Input 05 NPN-PNP programmable
7	0.25 mm²	Screw clamping	Input 06 NPN-PNP programmable
8	0.25 mm²	Screw clamping	Input 07 NPN-PNP programmable

C11 CONNECTOR FOR POWER SUPPLY

Pin	Cable cross-section	Type of terminal	Description
1	0.25 mm²	Screw clamping	Board power supply +
2	0.25 mm²	Screw clamping	Board power supply -

MODULE MECHANICAL PROPERTIES



EN

MWPOS SOFTWARE

GENERAL

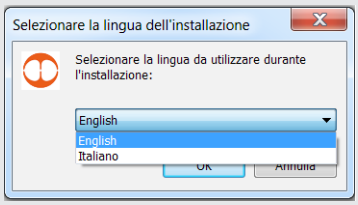
MWPOS is the software environment that Metal Work supplies for motion programming. It is compatible with MS Windows-based operating systems (Windows XP up to Windows 10) and has the distinctive feature of combining ease of use with flexibility and operating capacity. MWPOS is used to generate a file 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

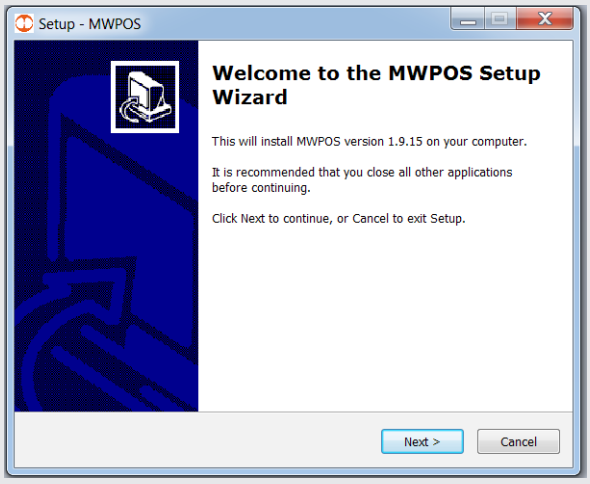
- Microsoft Windows Xp Sp3 or higher
- Microsoft .NET Framework 4.0 (www.microsoft.com/en-us/download/details.aspx?id=17718)
- Screen resolution 1024x768
- Driver Stm32 Virtual Com Port (www.st.com/web/en/catalog/tools/PF257938)

INSTALLATION

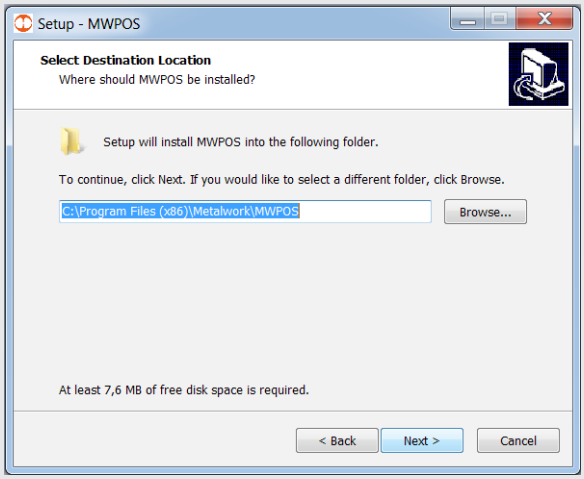
The installation of the software is simple and consists of a few steps.
When executing the installation file, you are prompted to select the language to use during both installation and use.
The available languages are Italian and English.



Select the desired language and click "**OK**".

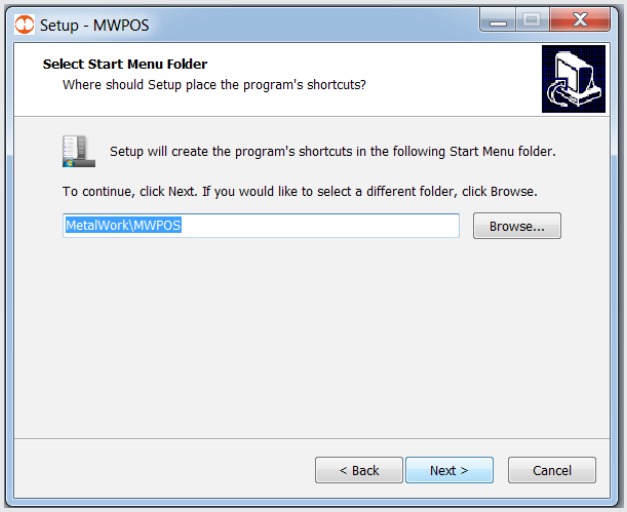


Then click "**Next**".

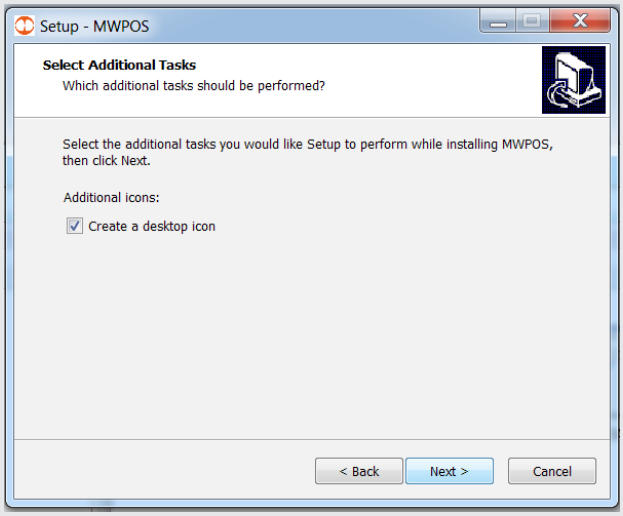


Select the destination folder and click "**Next**".

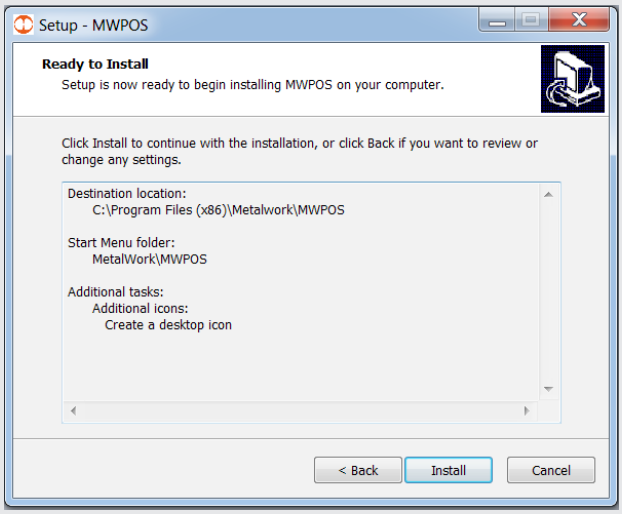
EN



Select the name of the folder in the Start menu and click "**Next**".

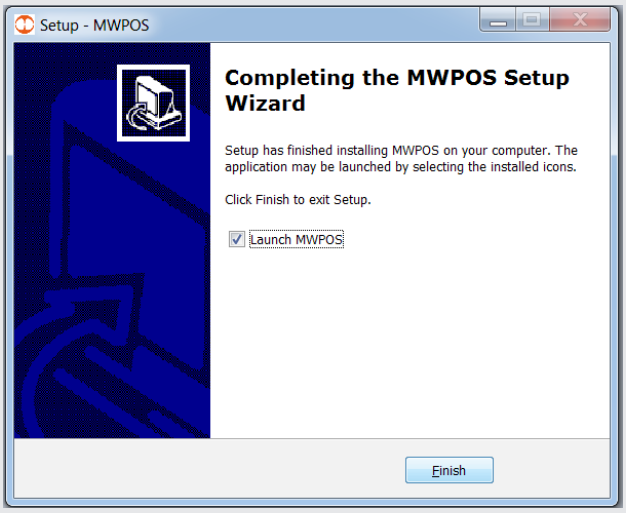


Tick the "**Create a desktop icon**" checkbox if you want to create a desktop shortcut.



Click "**Install**" to install the program.

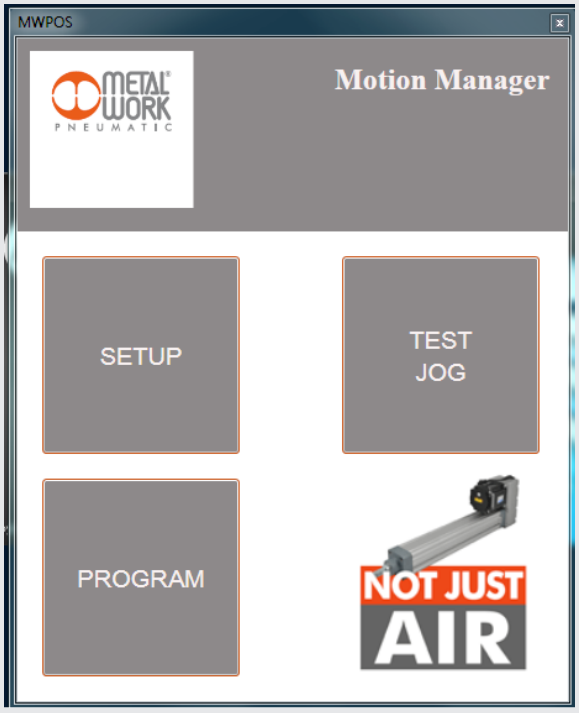
EN



Program installed.
Tick the "**Start MWPOS**" checkbox to launch the program.

MAIN MENU

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

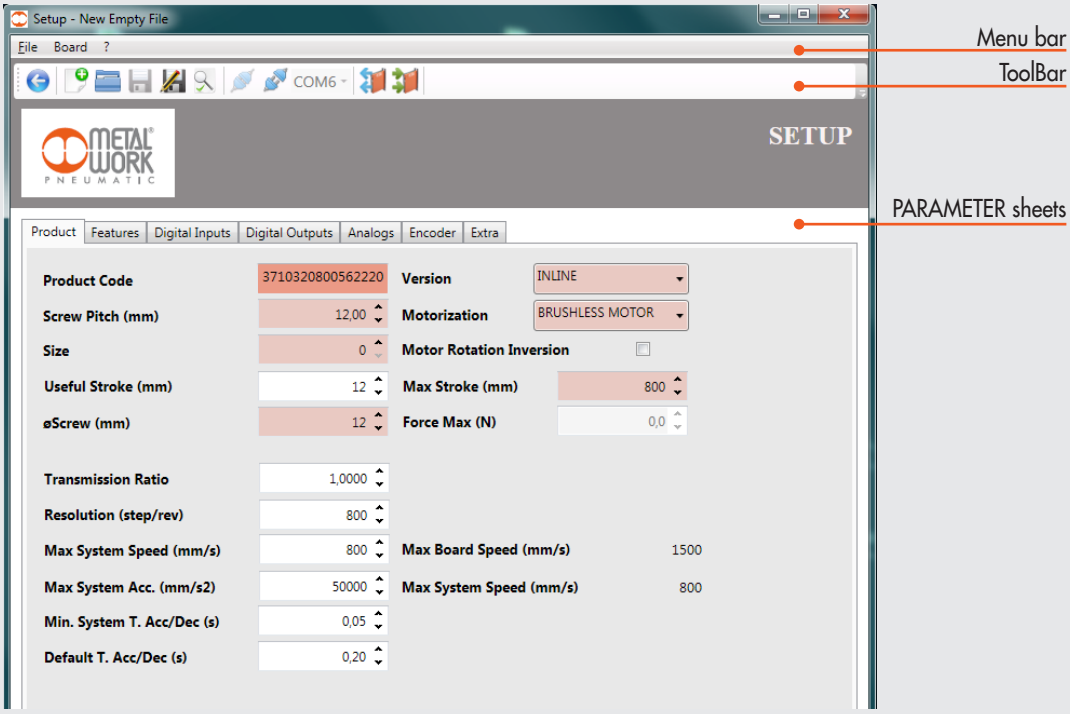


SETUP: allows you to set up the actuator, drive and function parameters.
PROGRAM: allows you to create the work program.
TEST JOG: allows you to perform motion tests on the actuator and collect diagnostic information.

EN

SETUP

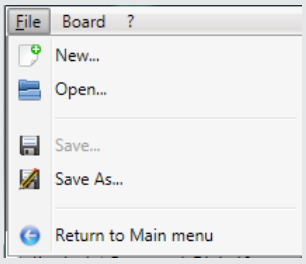
The **Setup** window is divided into the following sections:
By entering the Product Code (in the case of a standard product), which is stamped on the electric actuator supplied, the software automatically completes the fields relating to the mechanical characteristics of the actuator (pink fields).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



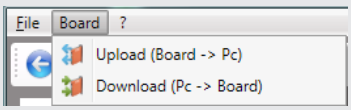
If there is no standard or special product code, the fields can be customised by the user.

MENU BAR

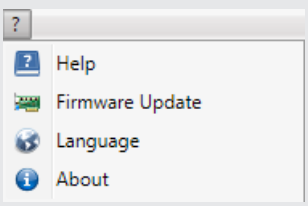
The menu bar can be used to perform some operations on the parameter file.



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".



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

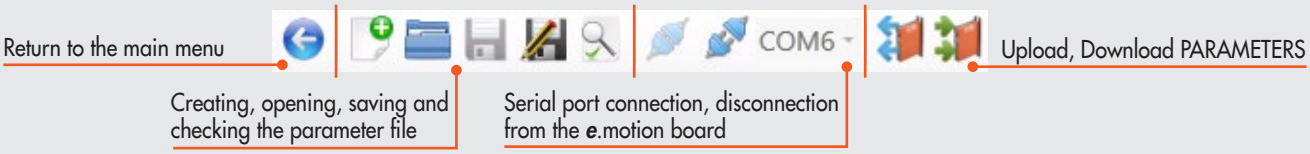


The "**?**" menu allows you to view the manual, update the board firmware, change the PC software language and view MWPOS software version information.

EN

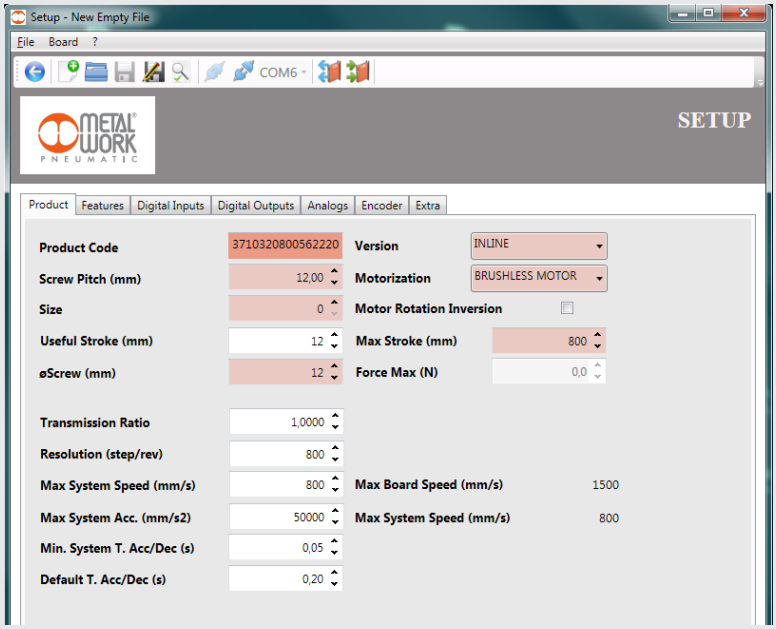
TOOLBAR

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/rev)	Axis feed per motor rev with gear ratio1:1
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 force (N)	Maximum axial load acting on actuator
Transmission ratio	Gear ratio between drive shaft and actuator input shaft. Decimal number (e.g. 0.3333 – 1.0000)
Resolution (step/rev)	Number of pulses per revolution: see characteristics and settings of the drive used
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

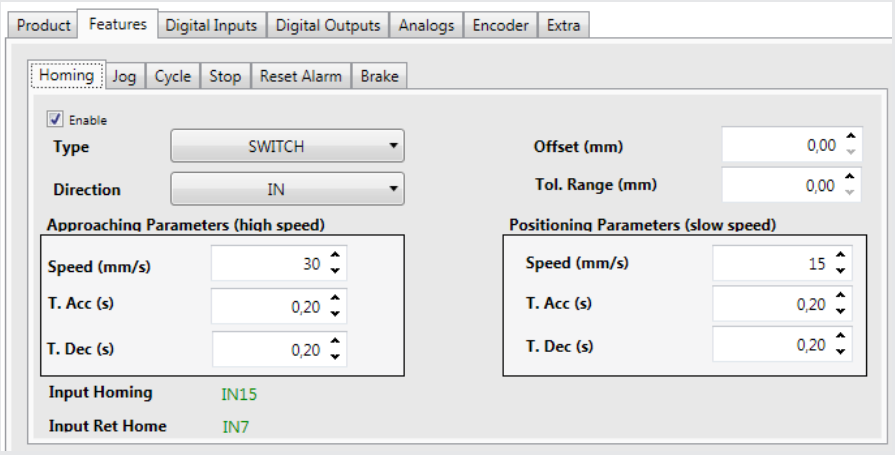
N.B.: The decimal separator in numeric fields depends on the international settings of the PC used.

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.

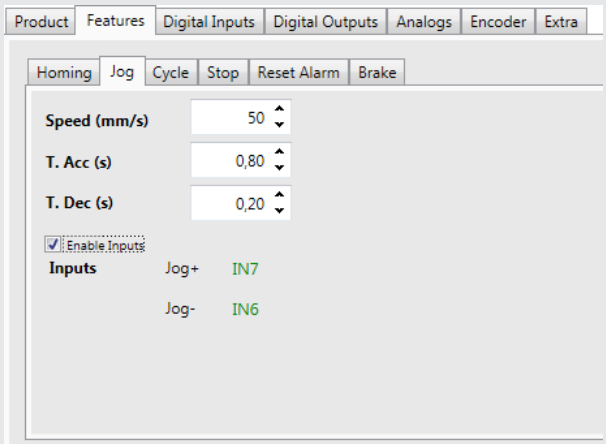
⚠ **IMPORTANT!** To activate the **MECHANICAL STOP +ENCODER** and **LIMIT SWITCH+ENCODER** mode, an Encoder with Index must be enabled in the Encoder page.

Other parameters displayed in this page:

Type	Zeroing mode
Direction	Motor direction IN or OUT
Offset	Offset (mm) to be reached after zero searching. The permissable stroke reduces consequently.
Tolerance	Tolerance (mm) for the calculation of the step loss during homing. It is disabled if you set 0.00 mm.
	If enabled it functions with both motors type (stepper and brushless) even if thre's no Encoder connected and 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.motion control board.
	It is very useful in open – loop systems, typically made of stepper motor without encoder.
Input HOMING	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.
Input RET HOME	It's the digital input provided to let the e.motion execute the homing procedure in automatic mode without the cycle execution.
	The homing procedure is performed as defined in feature tab.

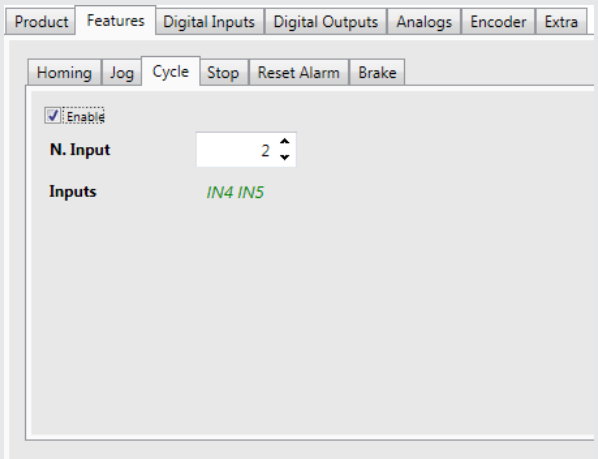
EN

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.

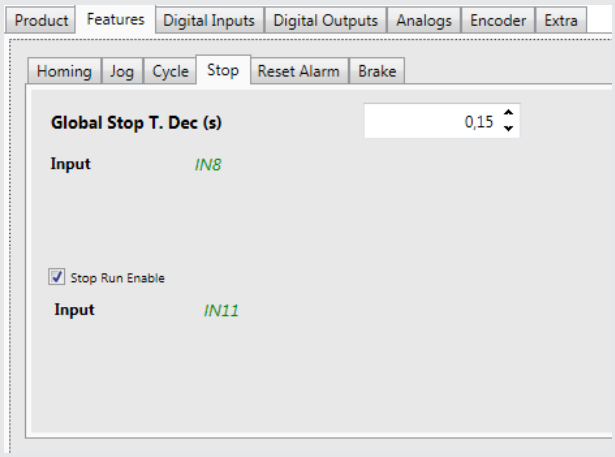
For example: using "**Input no.**" 2, four cycles can be set, as shown in the table below.

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.

EN

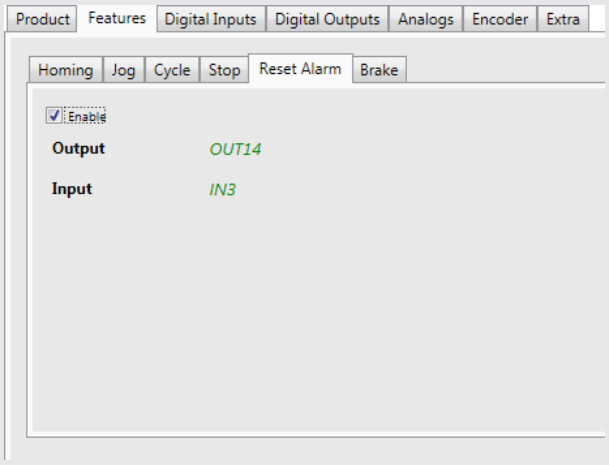
Stop



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



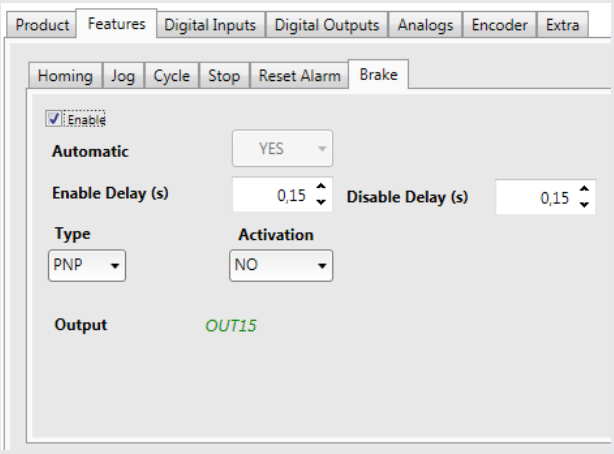
The "**Reset Alarm**" function is used to enable a digital input for use as reset from an alarm condition. Refer to the "**Digital Input**" function for the definition of this input.

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 digital input is also provided on the drive to allow you to restore the condition of movement authorisation after the cause of the alarm has been removed (generally called ALARM RESET). Refer to the "**Digital Output**" function for the definition of this output.

A typical application envisages the connection of a button to the digital input of the **e**.motion, whereas for restoring the end of the alarm condition, a direct connection with the specific drive input is envisaged from the digital output.

EN

Brake



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

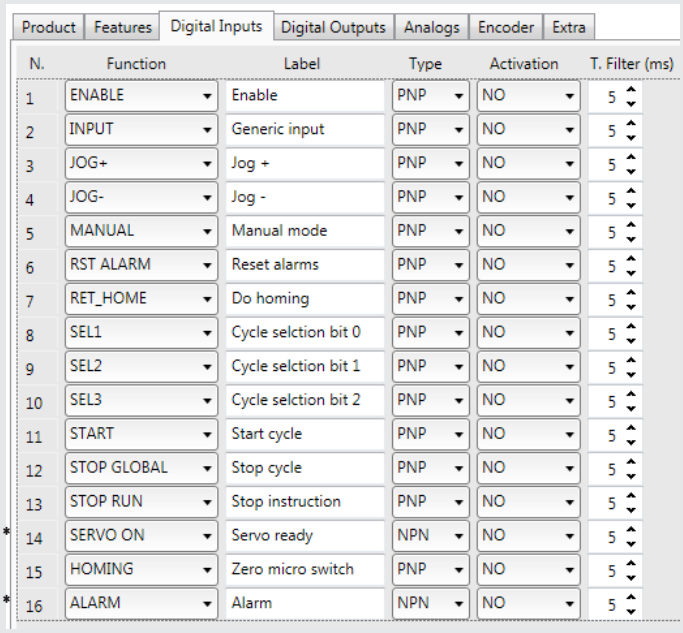
Parameter	Description
Automatic	Enables/Disables automatic brake control
Energ. delay T (s)	Brake energising delay
De-energ. delay T (s)	Brake de-energising delay
Type	Type of PNP brake output
Activation	NO/NC brake output activation

⚠ **IMPORTANT!** The brake must be connected to the dedicated OUT15 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.

EN

DIGITAL INPUTS



	N.	Function	Label	Type	Activation	T. Filter (ms)
	1	ENABLE	Enable	PNP	NO	5
	2	INPUT	Generic input	PNP	NO	5
	3	JOG+	Jog +	PNP	NO	5
	4	JOG-	Jog -	PNP	NO	5
	5	MANUAL	Manual mode	PNP	NO	5
	6	RST ALARM	Reset alarms	PNP	NO	5
	7	RET_HOME	Do homing	PNP	NO	5
	8	SEL1	Cycle selction bit 0	PNP	NO	5
	9	SEL2	Cycle selction bit 1	PNP	NO	5
	10	SEL3	Cycle selction bit 2	PNP	NO	5
	11	START	Start cycle	PNP	NO	5
	12	STOP GLOBAL	Stop cycle	PNP	NO	5
	13	STOP RUN	Stop instruction	PNP	NO	5
*	14	SERVO ON	Servo ready	NPN	NO	5
	15	HOMING	Zero micro switch	PNP	NO	5
*	16	ALARM	Alarm	NPN	NO	5

* Default inputs to be maintained in the positions shown and in NPN type when connecting the ***e***.motion to "Sanyo Denki RS1A0_" driver using cable code 37C2510000.
Default inputs to be manteined in the position shown but in PNP type when connecting the e.motion to "Delta ASDA-A2" driver using cable 37C2510001.

The "**Digital Input**" function can be used to configure digital inputs for connection to the board using connectors C2, C3, C8 and C10.
Each input has the following configurable features:

Type	Description
No.	Input identification number
Function	Input-associable function
Comment	Description of the input
Type	Type of PNP, NPN input
Activation	NO, NC activation (normally open, normally closed)
Filter	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
ENABLE	Enables the system by activating the EN_MOTOR output (see "DIGITAL OUTPUT" function)
ALARM *	Alarm input
HOMING	Zero switch signal
JOG-	Performs Jog- motion (MANUAL ON input required)
JOG+	Performs Jog+ motion (MANUAL ON input required)
MANUAL	Enables the manual mode to perform jog commands
RST ALARM	Resets internal and external alarms, if the RST_ALARM output is configured.
SEL1, SEL2, SELn	Cycle selection input (binary combination – see Cycle under FEATURES)
SERVO ON *	From the drive in response to actual enabling
START	Starts a selected cycle
STOP GLOBAL	Executes a global stop (see Stop under FEATURES)
STOP RUN	Stops current instruction without completing the cycle (see Stop under FEATURES)
INPUT	Free generic inputs available
RET HOME	Executes the homing cycle defined in homing feature

DIGITAL OUTPUTS

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

N.	Function	Label	Type	Activation
1	READY	--	PNP	NO
2	ALARM	--	PNP	NO
3	OUTPUT	--	PNP	NO
4	OUTPUT	--	PNP	NO
5	OUTPUT	--	PNP	NO
6	OUTPUT	--	PNP	NO
7	OUTPUT	--	PNP	NO
8	OUTPUT	--	PNP	NO
9	OUTPUT	--	PNP	NO
10	RUN CYCLE	--	PNP	NO
11	IN HOME	--	PNP	NO
12	IN POS	--	PNP	NO
*13	EN_MOTOR	--	PNP	NO
*14	RST ALARM	--	PNP	NO
15	OUTPUT	--	PNP	NO

---> "Special" 600 mA output to control the holding brake.

* Predefined outputs to be maintained in the positions shown, when the ***e***.motion is connected to "Sanyo Denki RS1A0_" driver with cable code 37C2510000 or when it's connected to "Delta ASDA-A2" driver with cable code 37C2510001.

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.

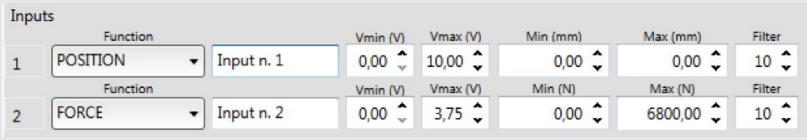
Each digital output can be associated to a "**Function**". Following activation of the selected input, the corresponding function is called up.
A generic function, called "**OUTPUT**", is also provided in addition to the standard functions required by a motion controller.
This output is considered as 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
RST ALARM *	Resets the external alarms if the RST_ALARM input is configured
EN_MOTOR *	Drive enabling output activated with the ENABLE input (see DIGITAL INPUTS)

EN

ANALOGUE INPUTS



The "**Analogue Input**" function can be used to configure two analogue inputs provided on the **e**.motion controller, which accept voltage signals only (0 ÷ 10VDC). Each input has the following configurable features.

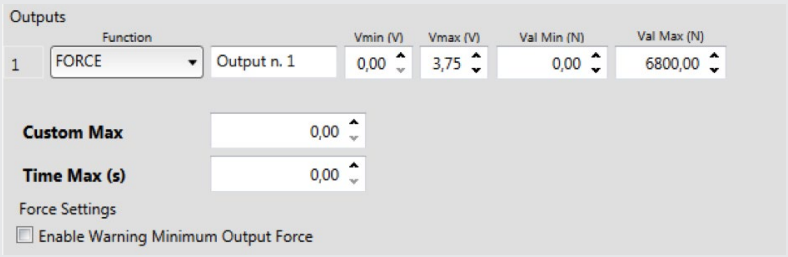
Type	Description
ID	Analogue input identification number
Function	Physical variable to be associated to the input signal
Comment	Generic description
Vmin (V)	Minimum input voltage
Vmax (V)	Maximum input voltage
Filter	Input filter to avoid interference (the greater the filter, the greater the acquisition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

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.

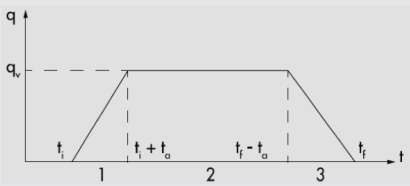
The physical variable associable to the analogue input variation can be selected between POSITION, SPEED, TIME, FORCE, and a CUSTOM one. Function choice is due to the hardware physically connected and to user needs.
POSITION and **SPEED** perform a real – time control of the movement and they must be used with the MOVE_CNT. instruction. See page 68 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.
FORCE: information could come from any force transducer (correctly configured) or from the analogue output of a brushless servo drive (correctly configured). Usually the drive analogue output gives information about the torque percentage necessary to perform actual movement.
Through simple formulas the user can translate the analogue signal coming from the drive (0 ÷ 10VDC) in the FORCE feature: $\mathbf{F = 2\pi \frac{C}{p}\ [\frac{Nm}{m}]}$
Where **C** is the torque and **p** is the screw pitch of the cylinder used.

Using a force transducer like a load cell the user can translate the analogue signal coming from the cell itself (0 ÷ 10VDC) in the FORCE feature through the following formula: $\mathbf{F = m \cdot g\ [\ kg \cdot \frac{m}{s^2}\ ]}$ Where **m** is the mass of the load moving and **g** is the gravitational constant.



The analogue output can be parameterized, (0 ÷ 10VDC), to perform the torque limitation of brushless servo drive correctly configured.
These drives usually accept an analogue input signal that limits the maximum torque that drive gives to the motor during the real – time movement.
Relationship between maximum load admitted and torque given by the drive is exactly as described for analogue input.
Instructions using this limitations are: MOVE_ABS; MOVE_REL; MOVE_CNT; MOVE_SPEED; SET_FORCE
See instructions details for example using.

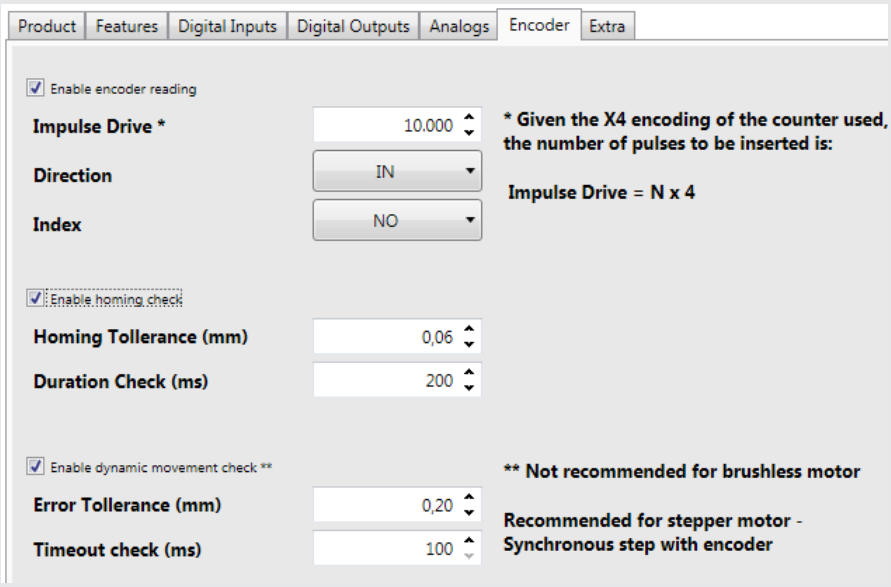
Enable Warning Minimum Output Force: selecting this option the torque limiting through analogue output is enabled, it means the Force (N) field in the abovementioned instructions cannot be set to 0,00 N anymore.
Every movement instruction must have a Force(N) limit value.



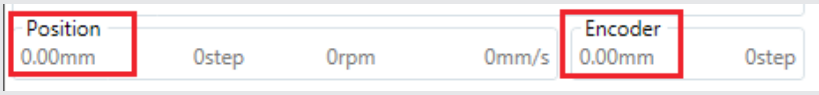
N.B.: This value will be the limit value that e.motion execute during the constant phase of the typical trapezoidal speed profile view (phase 2).
During acceleration and deceleration (phases 1 and 3) the limit will always be the Force Max (N) set in the product mask.
It's assumed that the speed profile should always be like the one reported in the figure below.

EN

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
Impulse Drive	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.
Direction	Motor direction (IN/OUT). Pulses' counting and transducing in mm/revolution is performer clockwise IN or counter clockwise OUT compliance with motor shaft rotation.
Index	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 defined 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.

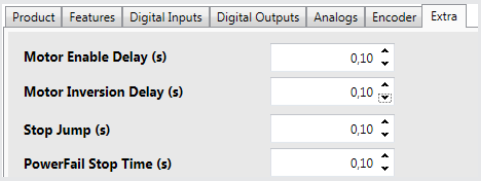
ENABLE HOMING CHECK
It's a position control performed through the encoder, expressed in mm, made when the motor is not moving and only when the homing procedure is performing. That procedure ends well only if this control is passed when the duration check time lapsed. Consequently the "state flag" HOME switches ON and is acquired the right s for executing an Automatic movement. Otherwise the system detects an error and goes into an alarm condition signaled by "position error" alarm.
Referring to the Encoder feature described by the picture above, 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
Homing Tolerance (mm)	tolerance for position control. Given by the subtraction between position commanded and position read.
	Tolerance (mm) = Position (mm) – Encoder (mm)
Duration Check (ms)	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.
Referring to the Encoder feature described by the picture above, 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.

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

EXTRA



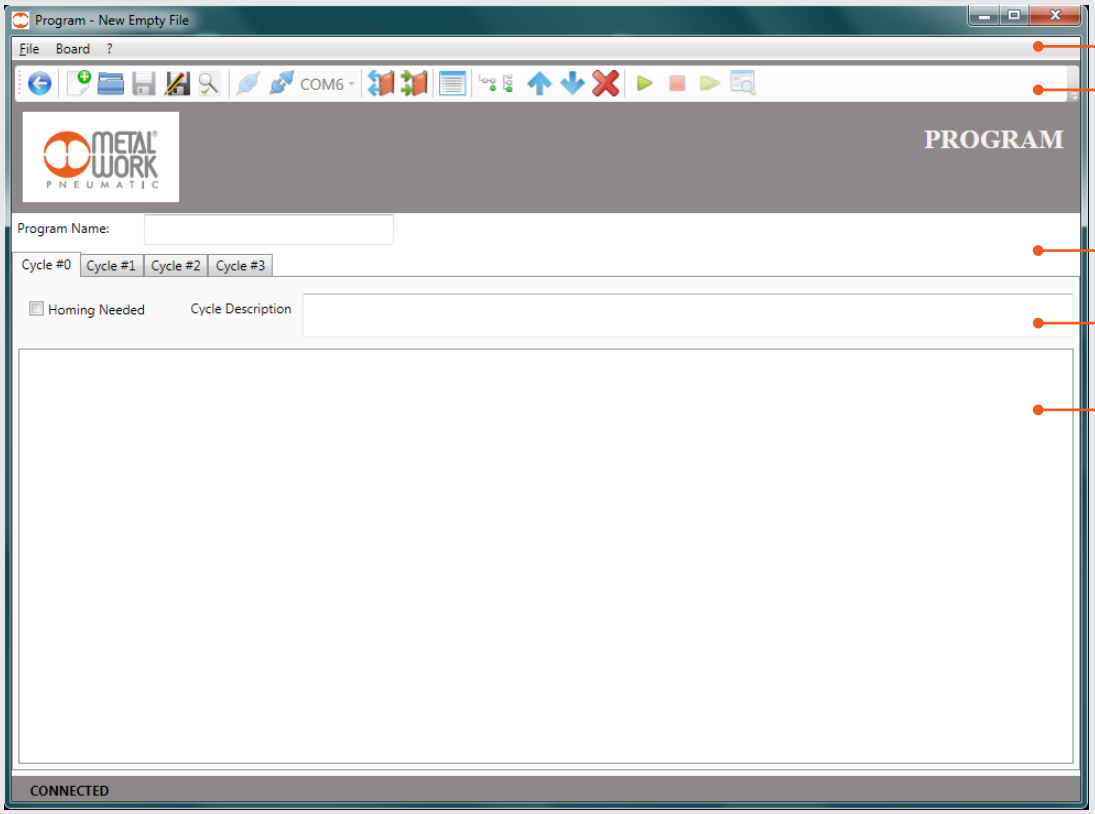
The "**Extra**" function page contains some additional settings that allow greater operating flexibility of the entire system.

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

EN

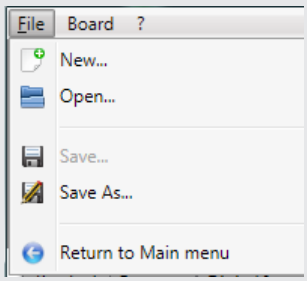
PROGRAM MENU

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:

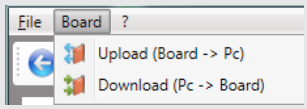


MENU BAR

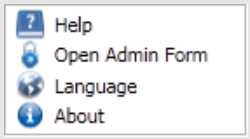
The menu bar allows you to execute some operations on the program file.



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**.motion board.
It lets you upload the program currently present on the board or download the program.

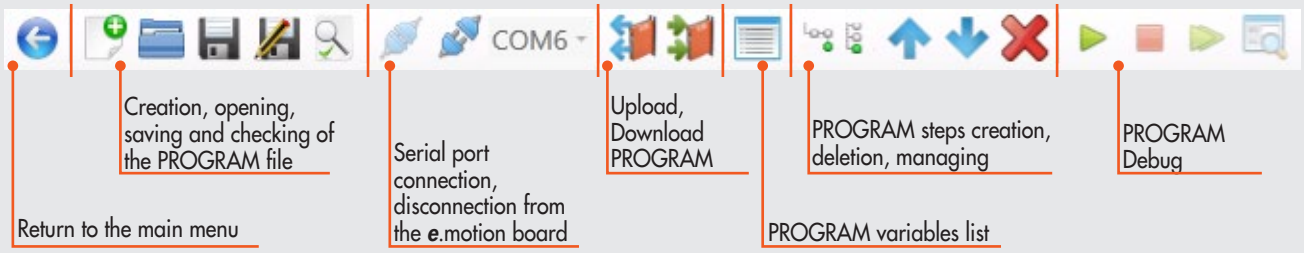


The "**?**" menu lets you display the manual, through admin form lets you update the firmware or download data to the board, exchange the MWPOS software language and display current software version data.

EN

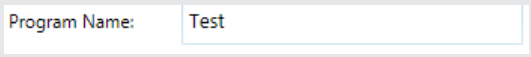
TOOLBAR

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

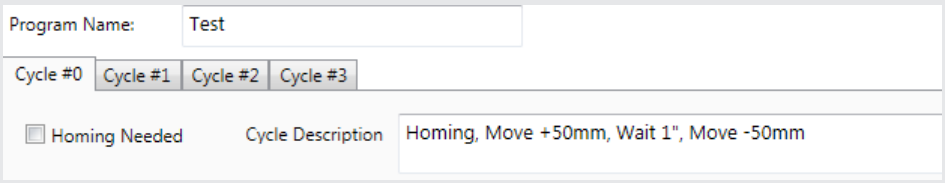
You can assign a name of maximum 30 characters to the program created.



CYCLE PARAMETERS

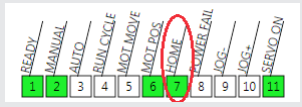
The program consists of cycles. In the "Cycle Description" zone, you can add a brief comment regarding the cycle.

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.



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.

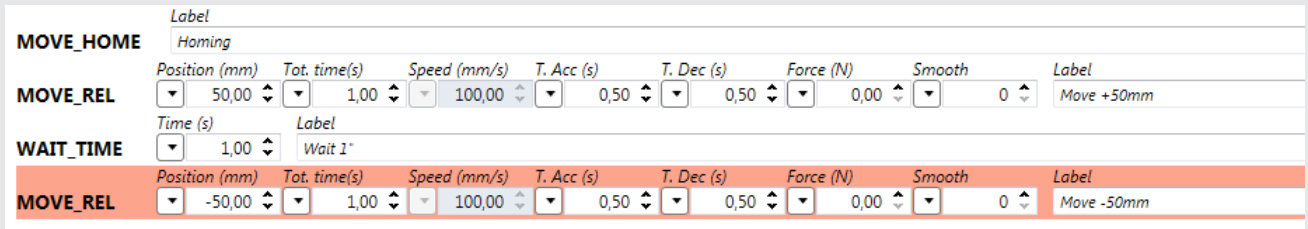
In this way, there being no input for the selection of the activated cycle, the motor will execute zeroing on first cycle start.
Upon completion of the zeroing procedure, the HOME condition is valid.



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.



EN

CREATING A PROGRAM

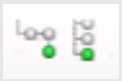
To create a new work program you can either select the File menu -> New File or use the shortcut button in the ToolBar.

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

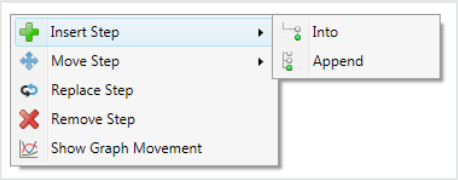
ENTERING SIMPLE STEPS

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

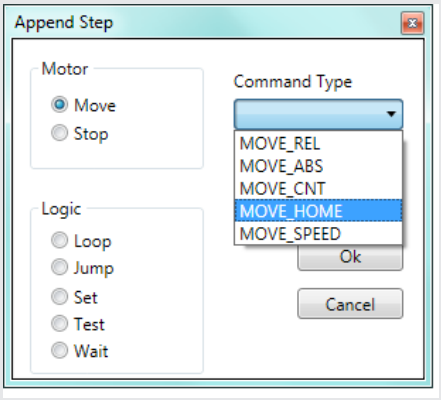


The difference between "Insert Step - Into" and "Insert Step - Append" is the possibility of entering nested steps. This is useful, for instance, when you wish to execute a motion step and simultaneously run a test on an input; or when conditional jump steps are used.

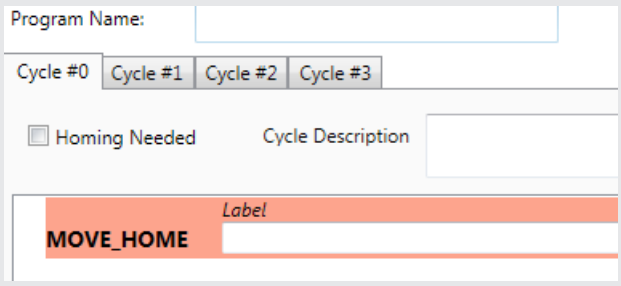
For further details on nested steps, please refer to the section titled "Enter Nested Step".

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.



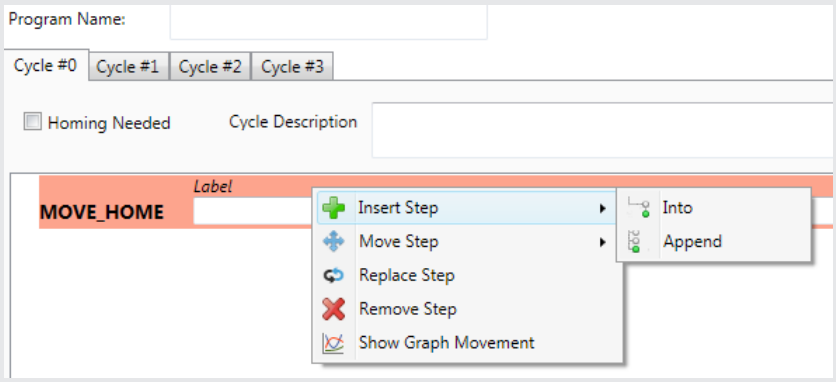
Click OK to enter the step or Cancel to cancel the entry.



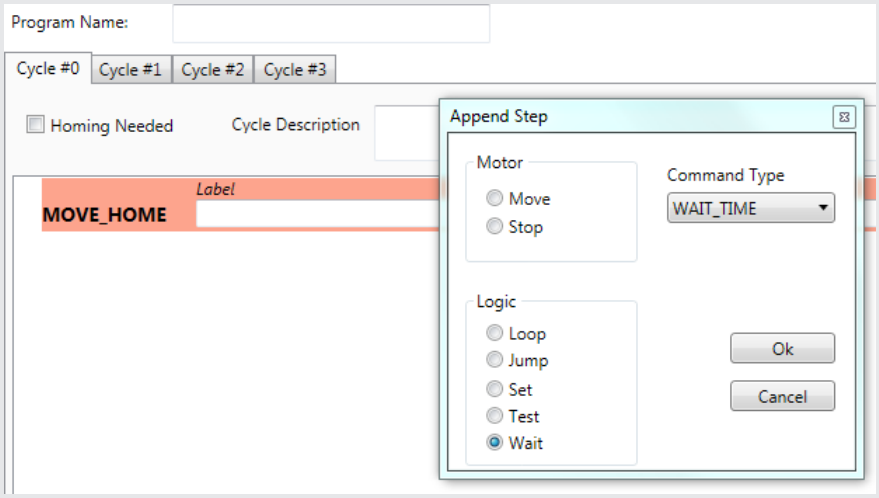
The selected step then appears in the selected cycle.

EN

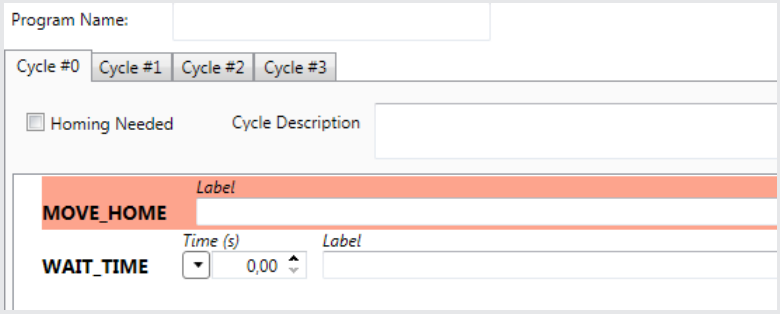
To enter another step next to the one just added, you can simply repeat the procedure, by selecting an Append entry.



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.

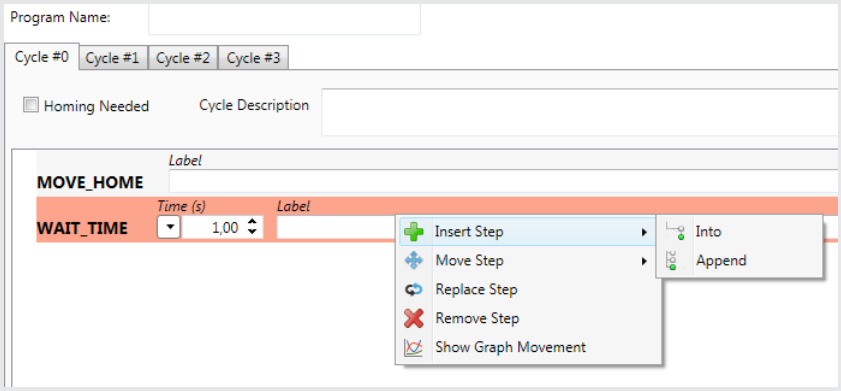


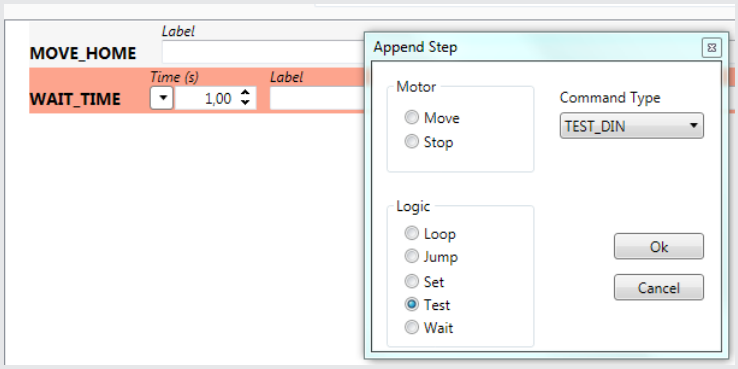
When the wait-time step has been lined up after the zeroing step, you can modify the time parameter.

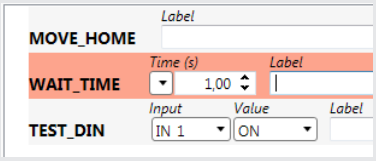
EN

ENTERING NESTED STEPS

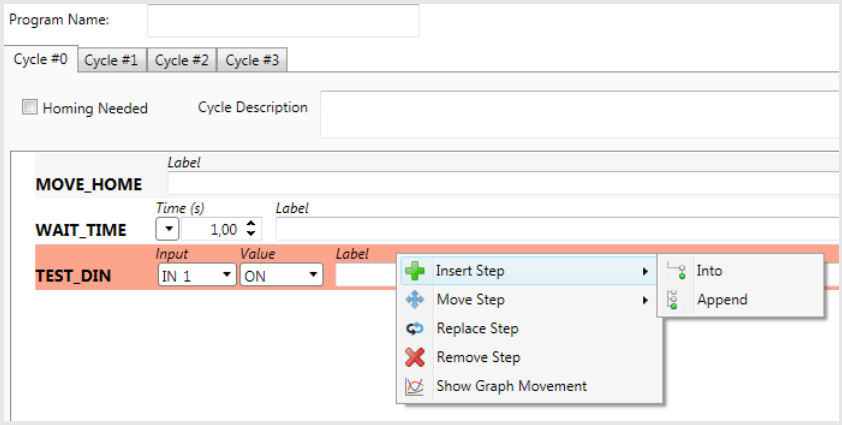
Some steps accept the execution of additional nested steps, such as the TEST_DIN step to test the digital input state.
To enter this step, you simply add it to the existing ones, as shown in the figure.



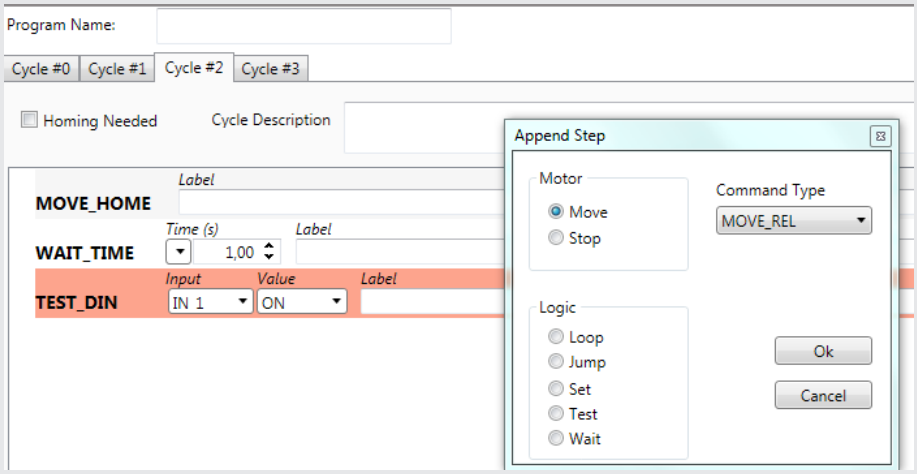




Once entered, you need to change the digital input you want to test (Input) and its logic state (Value).
At this point, you need to enter a nested step, the execution of which depends on the positive result of the test.
For example, if you want to set the positioning in front of the selected input, you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

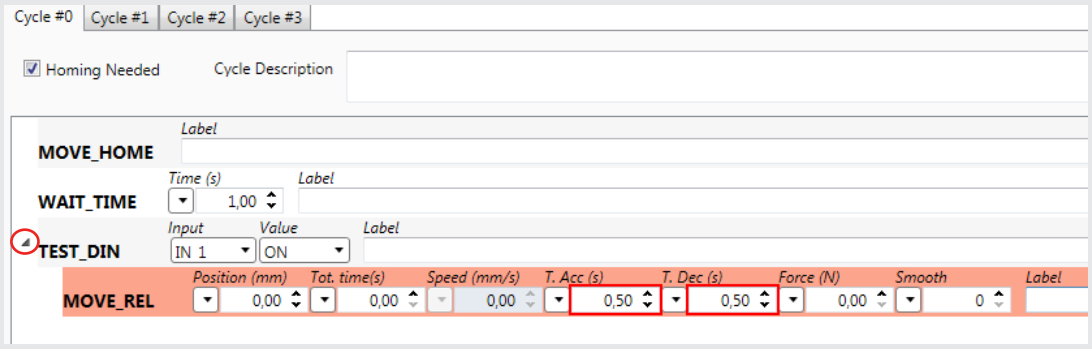


Select the "**TEST_DIN**" step and click the right mouse button to select the "**Insert Step -> Into**".



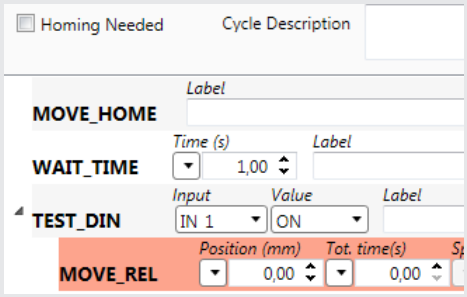
The select the "**MOVE_REL**" command.

When you click OK, the step will be nested inside the "**TEST_DIN**" step.

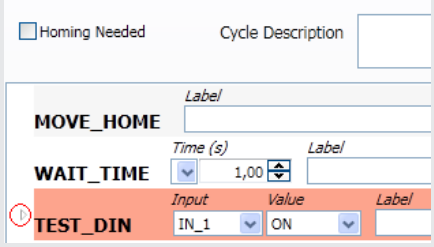


The steps (father) containing nested steps (son) are easily recognisable thanks to a triangle to the left of the "father" step (circled red in the figure).
The "son" steps are lined up using a tab stop (indentation) and can be identified intuitively just by looking at the alignment.
Each tab space corresponds 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.



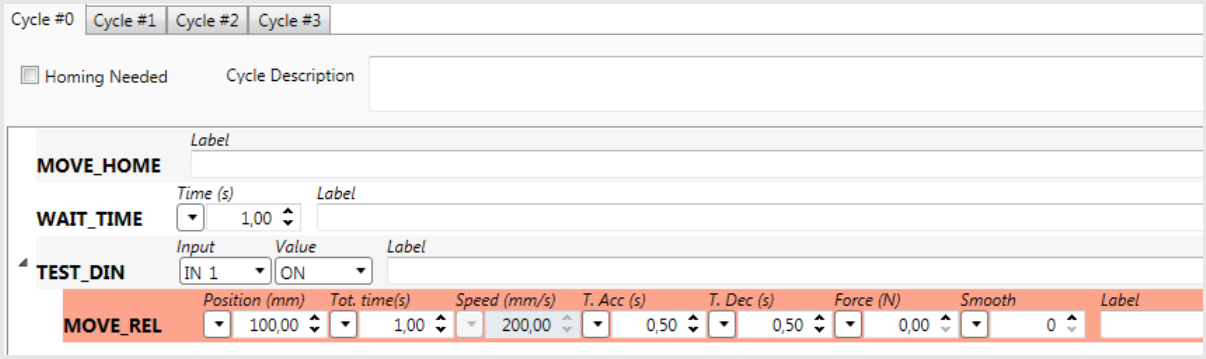
Expanded view



Compressed view

EN

At this point, you can personalise the parameters of the "**MOVE_REL**" step.



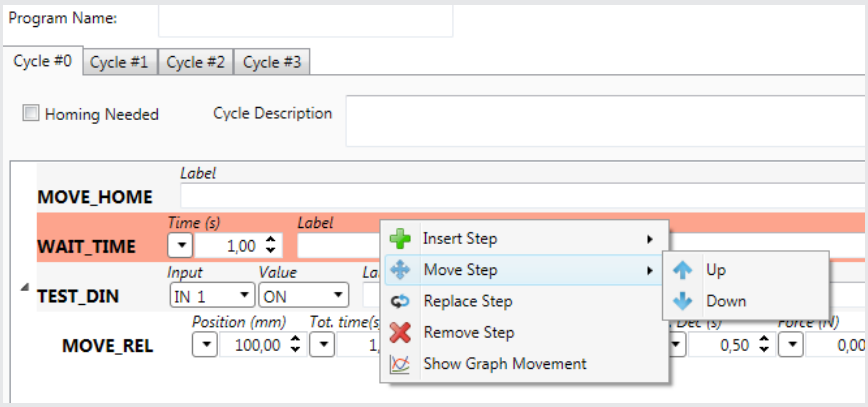
In summary, the newly entered cycle will:
1. perform the zeroing (MOVE_HOME).
2. wait 1 second (WAIT_TIME).
3. check that input 1 of the board is turned to ON (TEST_DIN).
4. if input 1 is turned to ON, it will move by 100 mm (MOVE_REL) in m in. 1.0 s, with 0.5 s acceleration and deceleration time
5. if not, there will be no movement.

MOVING STEPS

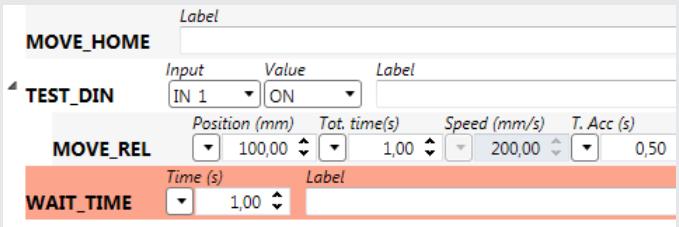
Two options are provided to move one step before or after another:

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.



More precisely, the "**WAIT_TIME**" step has been moved below the "**TEST_DIN**", test, and the following result is obtained:



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

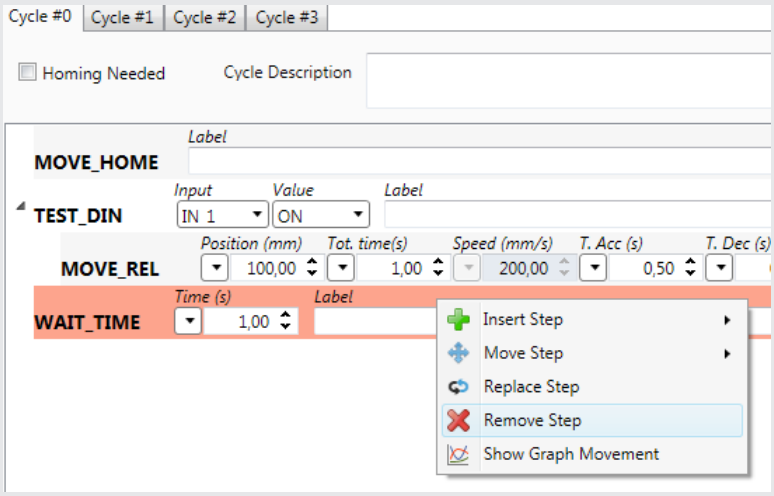
N.B.: The steps are moved within the same father-son branch. With this command, it is not possible to move steps between different father-son branches. If you wish to move or copy steps between different branches, please refer to the Copy Step section.

REMOVING STEPS

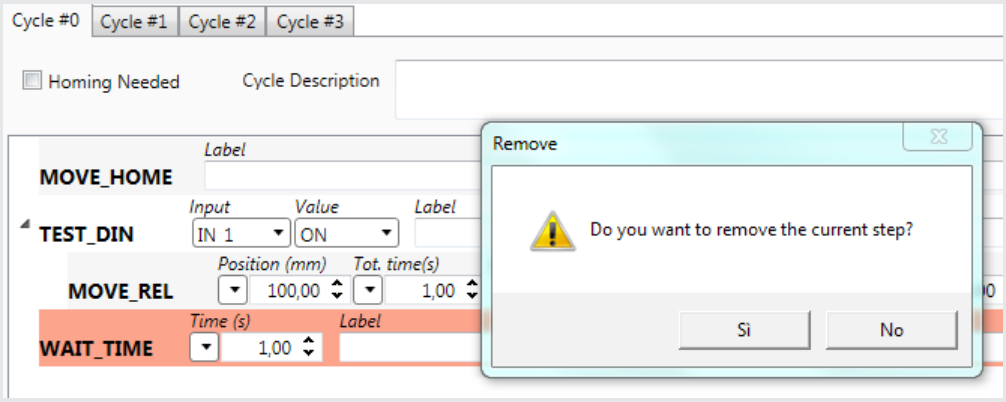
Proceed in either of the following ways to remove a step:

1. Select the stop to be removed and use the toolbar with the "**Remove Step**" button. 

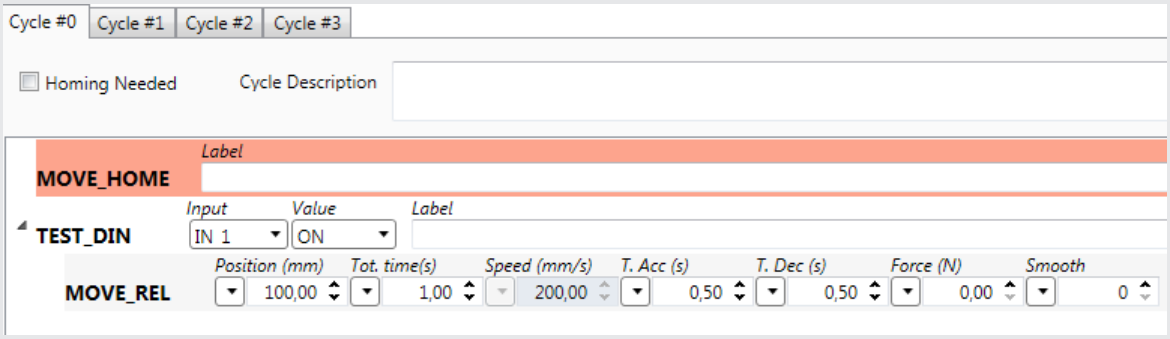
2. Use the right mouse button and click the step to remove by selecting "**Remove Step**"



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



The following result is obtained:

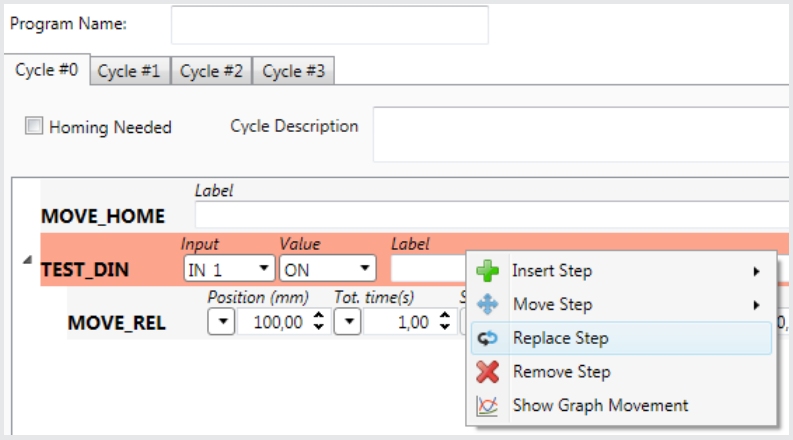


EN

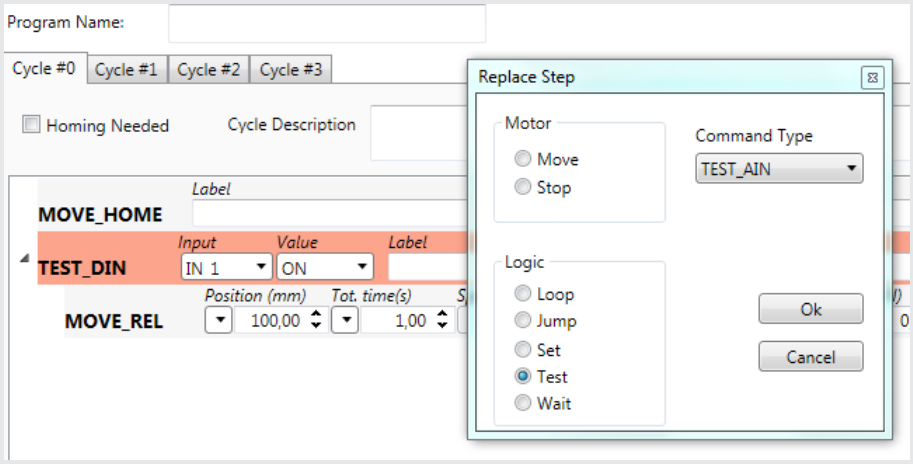
REPLACING STEPS

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:

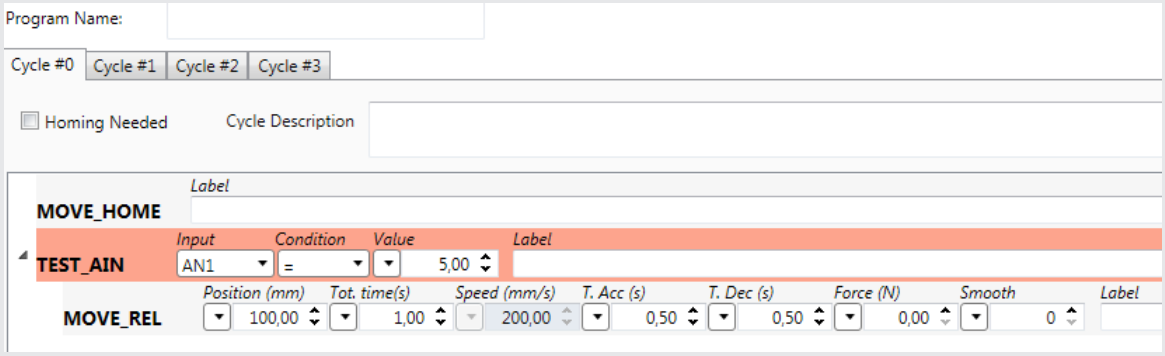
Select the Step to be replaced by clicking the right mouse button on "**Replace Step**".



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.



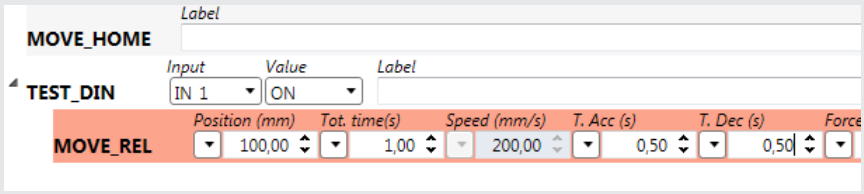
The following result is obtained:



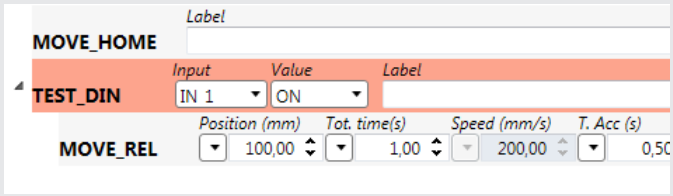
EN

COPYING STEPS

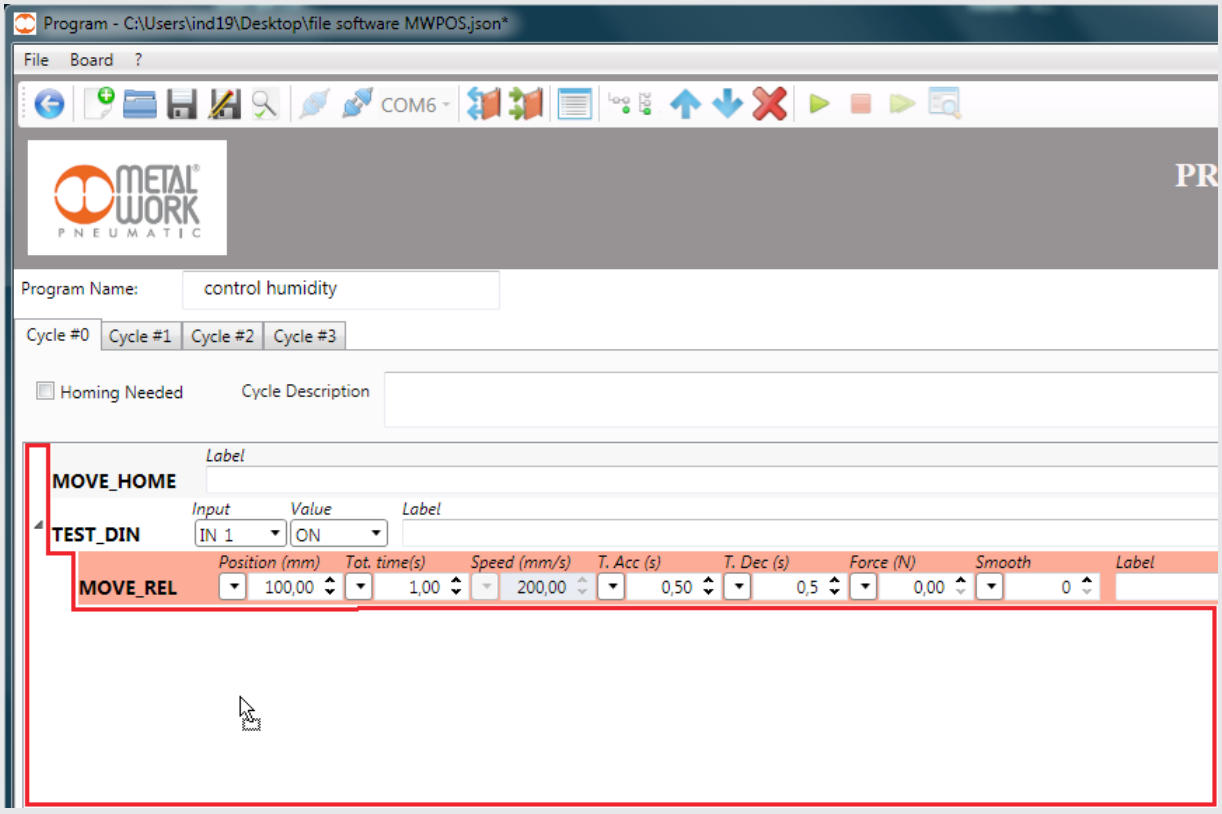
If you want to copy a step, this can be done by dragging the step and holding down the Ctrl key on the keypad simultaneously. For example, in the following program you want to perform the same identical test on input 1 and input 2.



Select the step to be copied, i.e. TEST_DIN in this case.

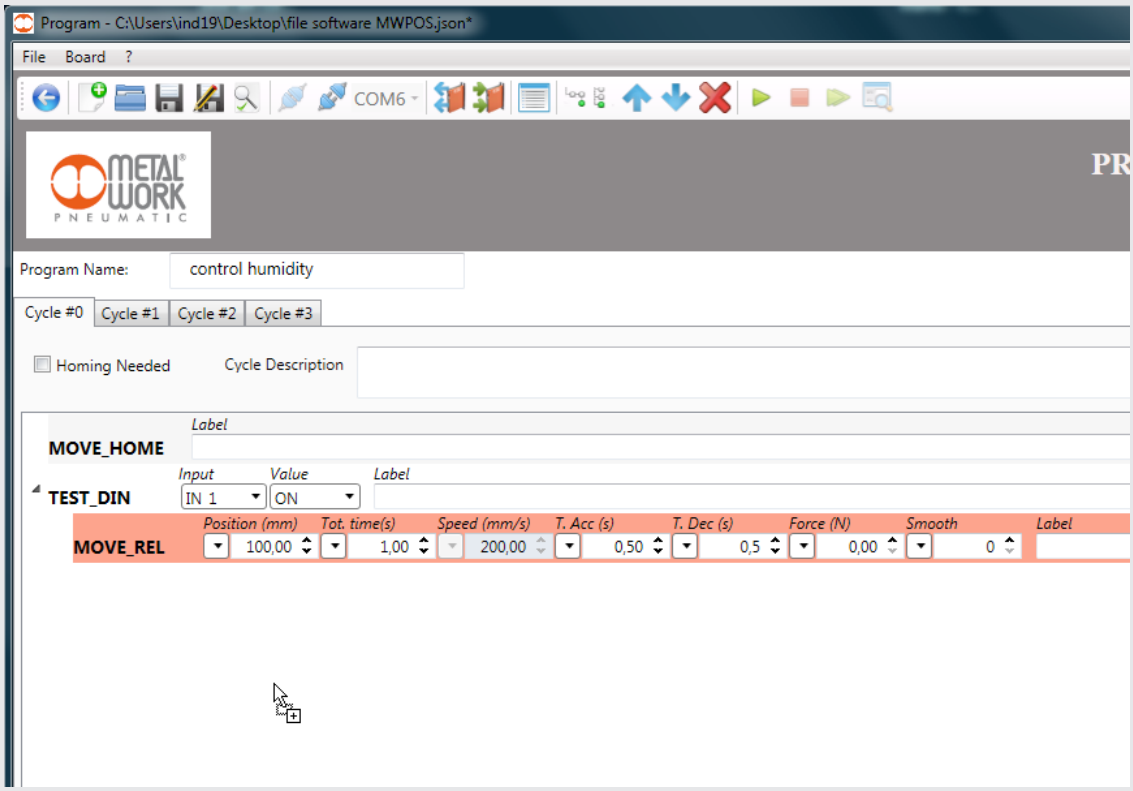


Drag the step down to the bottom of the list or to the left, holding down the left mouse button, until the mouse pointer highlights the displacement in progress. The area marked in red is where the displacement mode is activated.

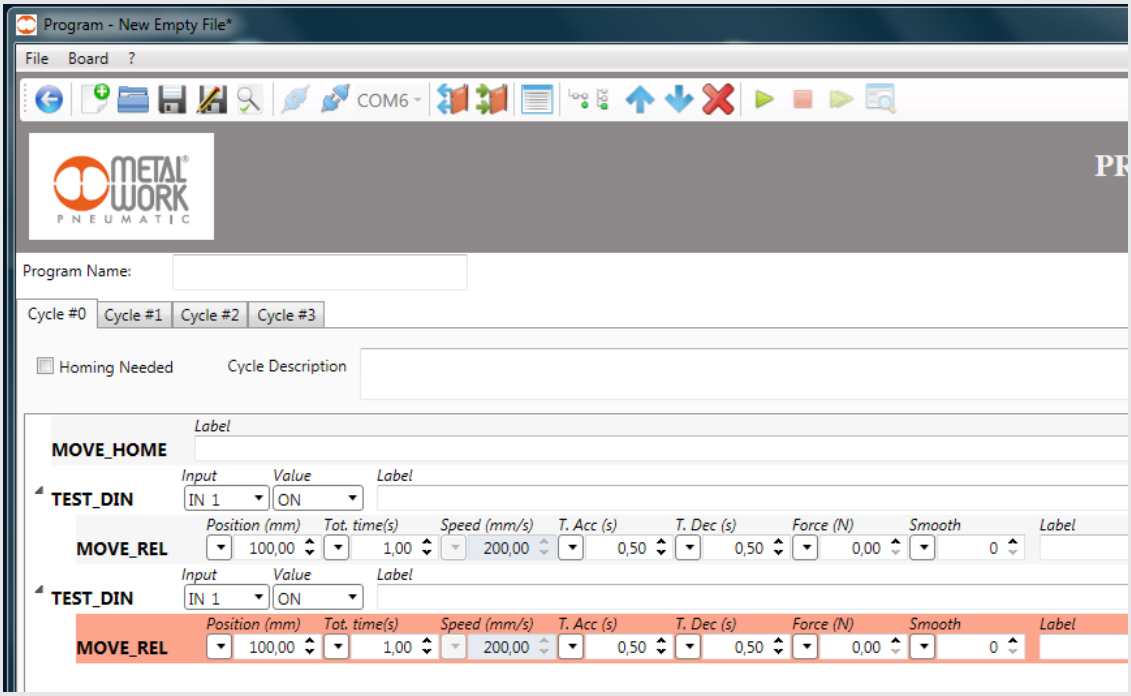


EN

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

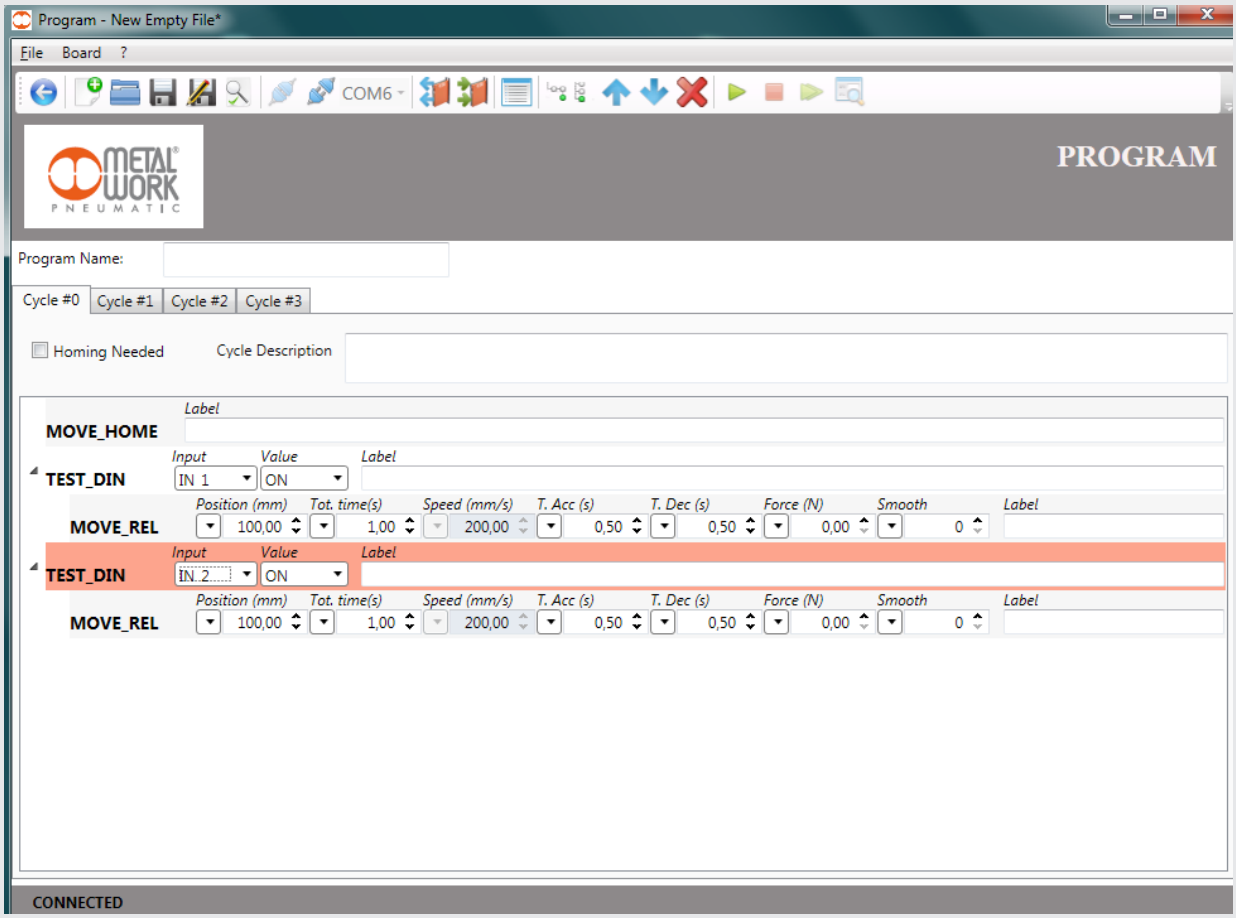


At this point, if you release the mouse left button, the copy is made and the step is lined up at the bottom of the step list.



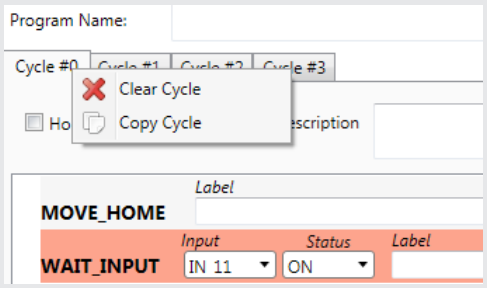
EN

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

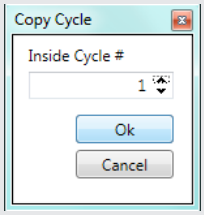


COPY/CLEAR CYCLE

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".



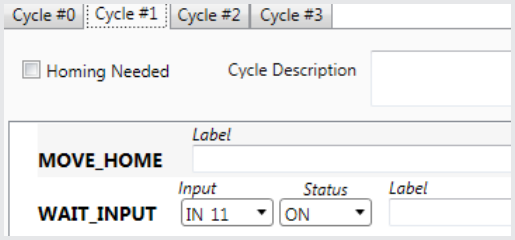
If you select "Copy Cycle", a dialogue box appears prompting you to specify the copy target cycle.



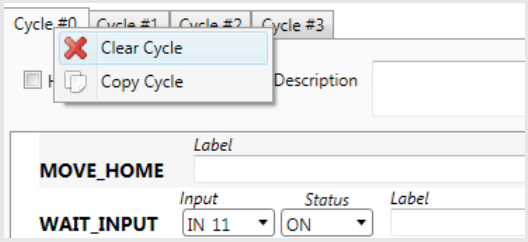
⚠ **IMPORTANT!** The target cycle will be overwritten!

EN

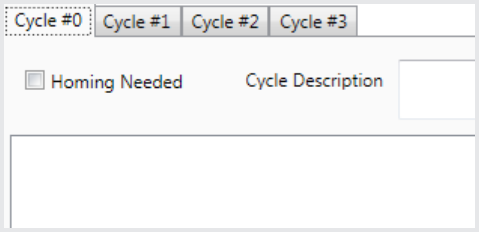
If you confirm, the cycle will be copied.



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



If you confirm, the cycle will be cleared.

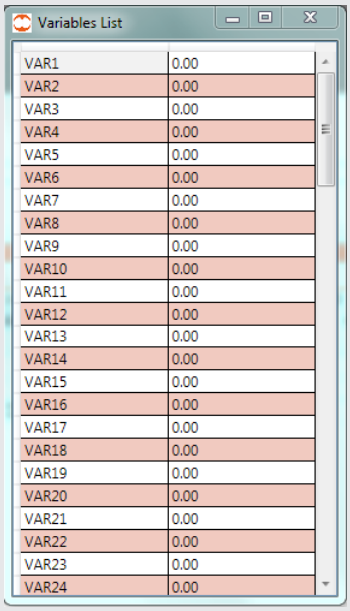


USING VARIABLES

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

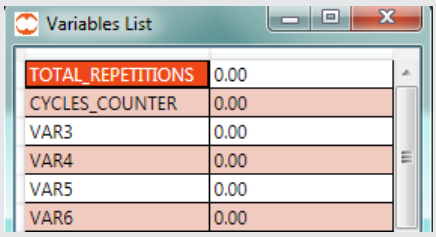
Select the "OpenList" icon to display the variables in the ToolBar. 

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

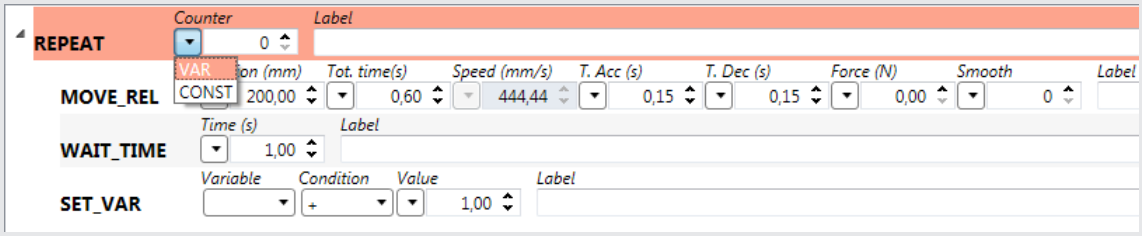


EN

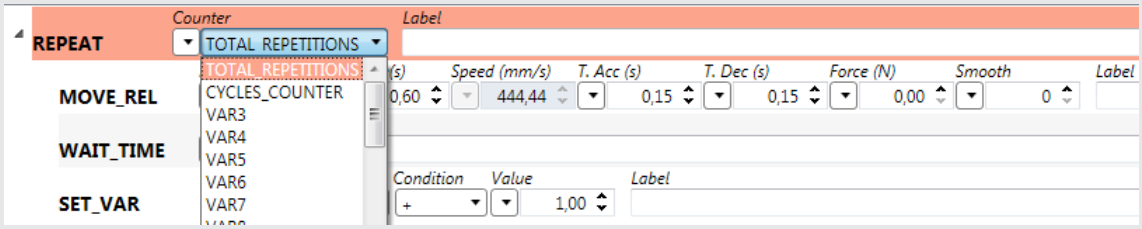
You can modify the name assigned by default to the variable by clicking the left column and the initialisation value by clicking the right column.
This will be used to initialise the variable on first program start-up.



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

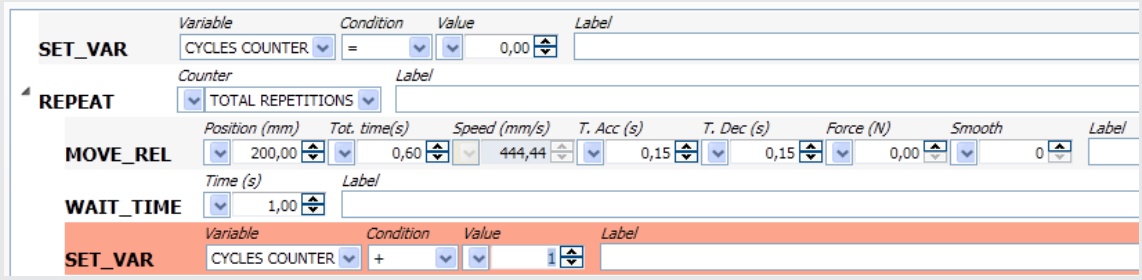


Then select the desired variable.



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.

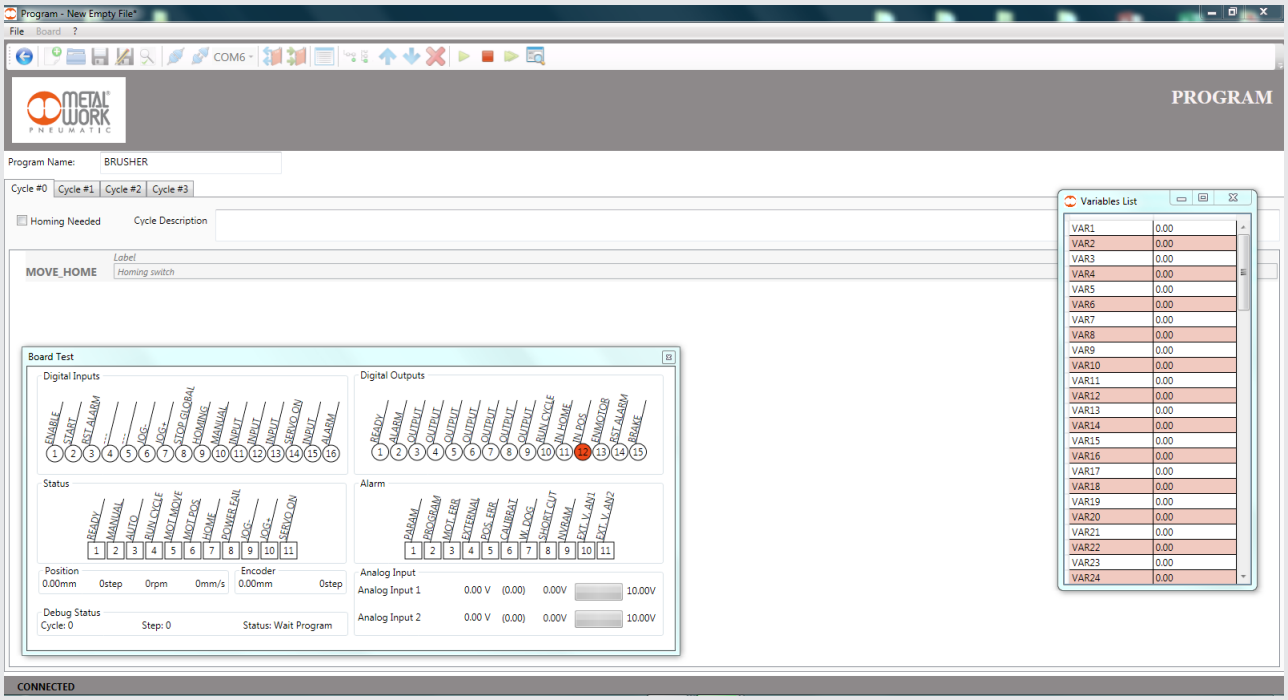


Two variables are used in this example: one for counting the cycles executed and the other to define the number of cycles to execute.
To change the number of cycles to execute, simply change the value of the TOTAL REPS variable.
The CYCLE COUNTER variable could be used, for example, to perform tests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

EN

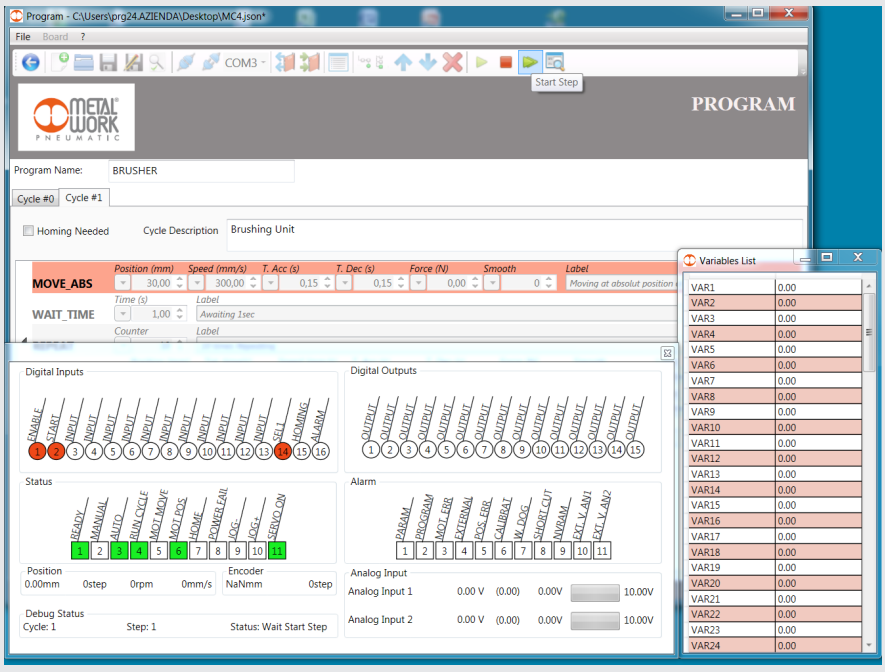
PROGRAM DEBUGGING

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.

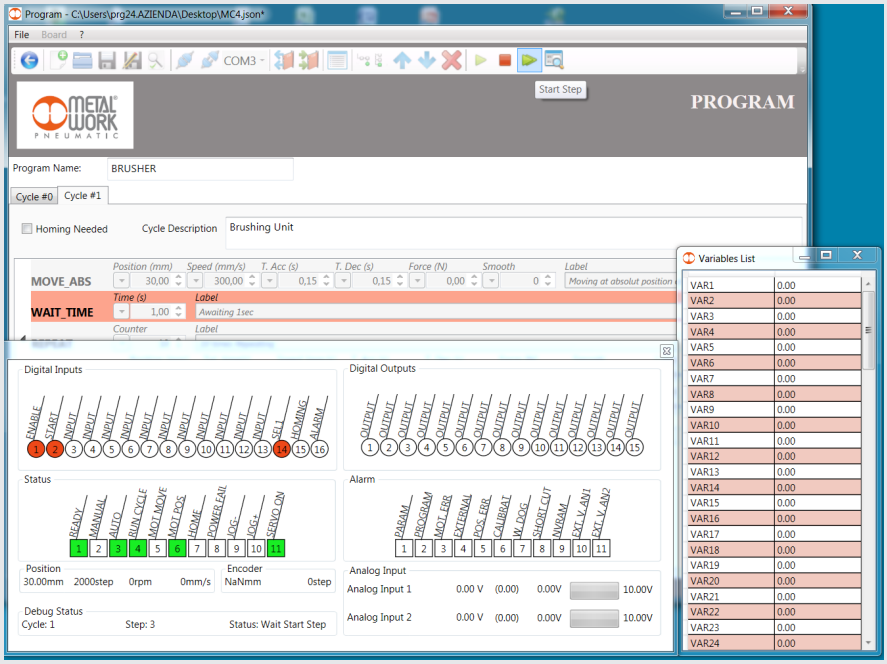


N.B.: On starting the debug session, the current program is automatically overwritten.
The **e**.motion starts up on AUTO, and thus the cycle is selected and launched using the digital inputs connected to the board, as in a typical application of use.

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.

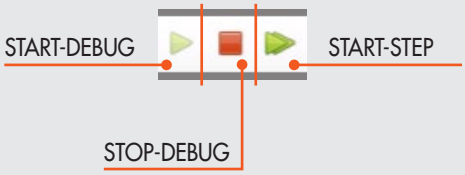


Press “Start Step” to start execution. The cycle performs the first step, moves to the next one available and returns to the Wait Start mode.
The “Test Board” dialogue box displays the status of digital inputs and outputs in real time, while the “Variable List” dialogue box shows the updated status of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



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.

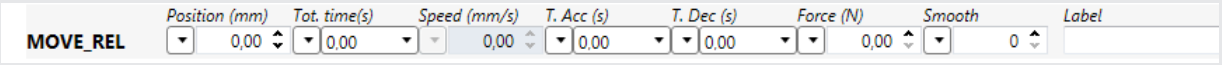


EN

LIST OF COMMANDS

MOTION STEP COMMANDS (MOVE UNIT)

MOVE_REL



This command can be used to position the actuator.

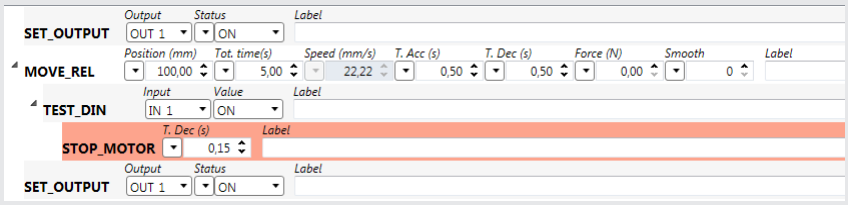
Parameters:

- **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.
- **Acc T. (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.
- **Dec T. (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.
- **Force (N)**: maximum load admitted during the movement at constant speed. See Analogs feature for detailed explanation.
- **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***.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.

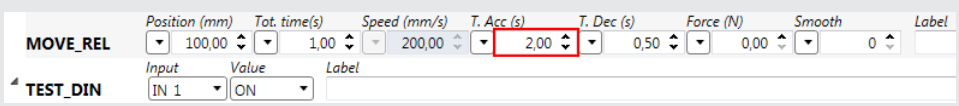
This is not a locking step and hence it may contain nested steps that are executed during execution of the movement.
The step is executed until the set position is reached or the motor is stopped.

For example:



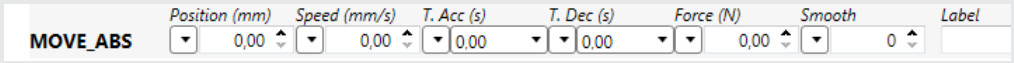
In the example, output 1 is set to OFF and then a 100.00 mm movement is performed in 5.0 seconds. If during the movement, input 7 turns to ON, the motor is stopped with a deceleration time of 0.15 s. The cycle then continues with the next step and hence sets output 1 to ON.

The MOVE_REL step performs an auto-test to check the correct setting of parameters. In case of an error, the parameter identified as wrong will be highlighted in the red box and you can move the mouse pointer on it to display a message that explains the type of error.
The red signal disappears once the wrong parameter has been corrected.



EN

MOVE_ABS



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.
  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).
- **AccT. (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.
- **DecT. (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.
- **Force (N)**: maximum load admitted during the movement at constant speed. See Analogs feature for detailed explanation.
- **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***.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 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.

EN

MOVE_CNT

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.

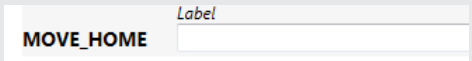
Parameters:

- **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).
- **AccT. (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.
- **Dec. T. (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.
- **Force (N)**: maximum load admitted during the movement at constant speed. See Analogs feature for detailed explanation.
- **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***.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 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



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:

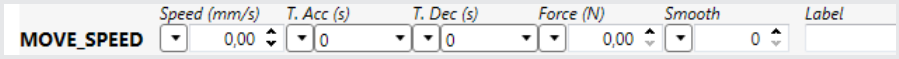
- **Comment (string)**: a text string to comment on the step.

N.B.: The comments are not transferred to the ***e***.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 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.

EN

MOVE_SPEED



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:

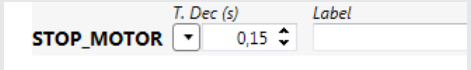
- **Speed (mm/s)**
  The maximum movement speed. The maximum value cannot be greater than the maximum speed set in the setup menu parameters.
  The button gives access to a menu in which you can set this parameter.
  1. VAR = lets you select a variable to be assigned to the parameter.
  2. CONST = (default) 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).
- **Acc T. (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.
- **Dec T. (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.
- **Force (N)**: maximum load admitted during the movement at constant speed. See Analogs feature for detailed explanation.
- **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***.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 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



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

Parameters:

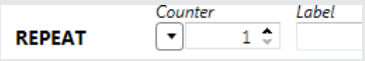
- **Dec. T. (s)**
  The time it takes to reach the set position, expressed in seconds. It cannot be less than the minimum system deceleration time set under the parameters.
  The button gives access to a menu in which you can set this parameter.
  1. VAR = lets you select a variable to be assigned to 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***.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.

REPEAT COMMAND (LOOP GROUP)

REPEAT



This command performs a repetition of any nested steps.

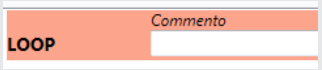
Parameters:
- **Count (numeric)**
  Number of nested step repetitions. When all the set repetitions have been performed, the system moves to the next non-nested instruction.
  The button gives access to a menu in which you can set this parameter.
  1. VAR = lets you select a variable to be assigned to 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***.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 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



This command performs an infinite repetition of any nested steps.

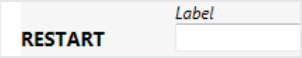
Parameters:
- **Comment (string)**
  A text string to comment on the step.

N.B.: The comments are not transferred to the ***e***.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 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



The cycle restarts from the first step.

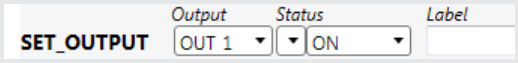
Parameters:
- **Comment (string)**
  A text string to comment on the step.

N.B.: The comments are not transferred to the ***e***.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 COMMAND (SET GROUP)

SET_OUTPUT



The command sets a digital output to the specified status.

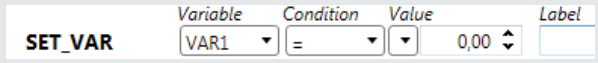
Parameters:

- **Output**
  The logic output of the board to be set.
- **Value (ON, OFF)**
  The value you with 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***.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_VAR



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

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***.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_FORCE

The command sets the maximum load admitted during the movement. This load is converted into an analogue value according the calibration set in the Analogs Mask Parameters:

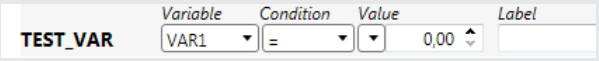
- **Force (N)**
  The value to be set for the Force.
  Via the button, you can access a parameter setting menu.
  1. VAR = lets you select a variable to be assigned to the parameter.
  2. CONST = (default) lets you enter manually the value to be assigned to the parameter.
  3. AN = (default) lets you 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.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.

EN

STATE COMMANDS (TEST GROUP)

TEST_VAR



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.
  3. VAR = lets you select a variable to be assigned to the parameter.
  4. 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***.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 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.

EN

TEST_INFO

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***.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 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

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***.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 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

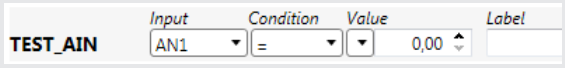
This command is used to perform a test on a digital output of the board.

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***.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 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



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.

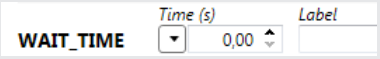
N.B.: The comments are not transferred to the ***e***.motion board. They are only saved on a computer file. When uploading the program from the board, the comments field is void. If the comments are worthy of note, we recommend saving the file to your computer before closing the PC software.

This 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.

EN

WAIT COMMANDS (WAIT GROUP)

WAIT_TIME



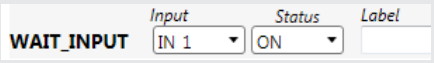
This command is used to execute a delay for the number of set seconds.

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

WAIT_INPUT



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.

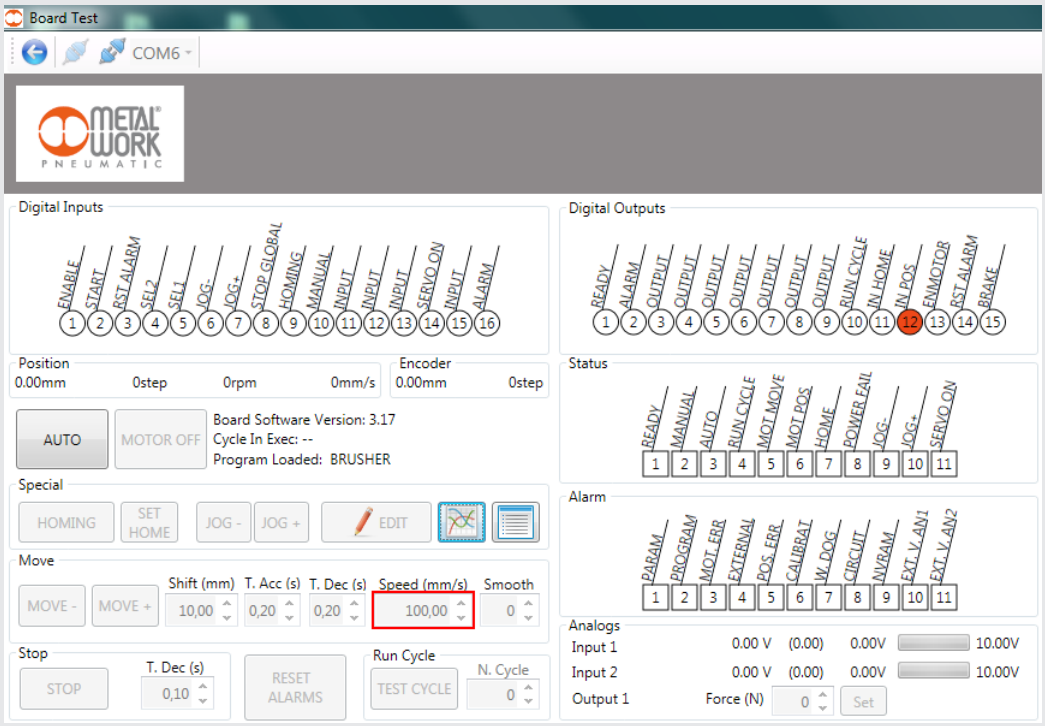
N.B.: The comments are not transferred to the ***e***.motion board. They are only saved on a computer file. When uploading the program from the board, the comments field is void. If the comments are worthy of note, we recommend saving the file to your computer before closing the PC software.

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

EN

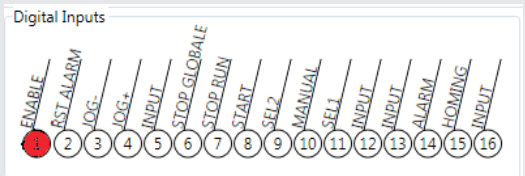
The Test and Jog menu can be used to perform tests on the ***e***.motion board, check the status of digital inputs and outputs and monitor the presence of any alarms.

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.



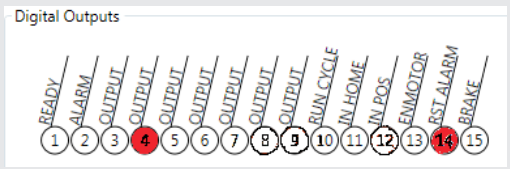
Digital inputs

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.

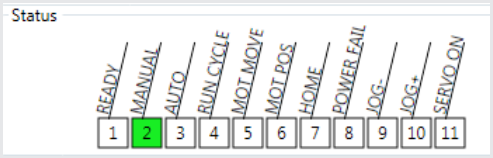


If an output is configured as a simple OUTPUT, you can force activation to ON or OFF just by clicking it.

EN

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

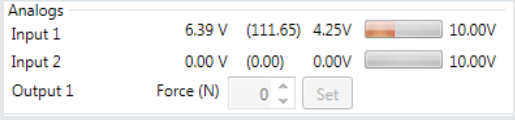
Alarms

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

Analogue inputs

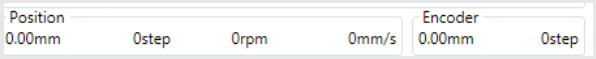
The Analogue Input window displays the status of analogue inputs according to their configuration.



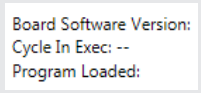
The figure below shows the name of the analogue input “Input 1”, the value read in real time is “6.39V”, the number in brackets is the conversion of the analogue value into the engineering value based on the parameter setting defined during setup.

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.

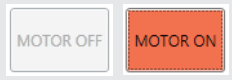


An area displaying the software version, the current cycle and the name of the program loaded on the board is also provided.



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

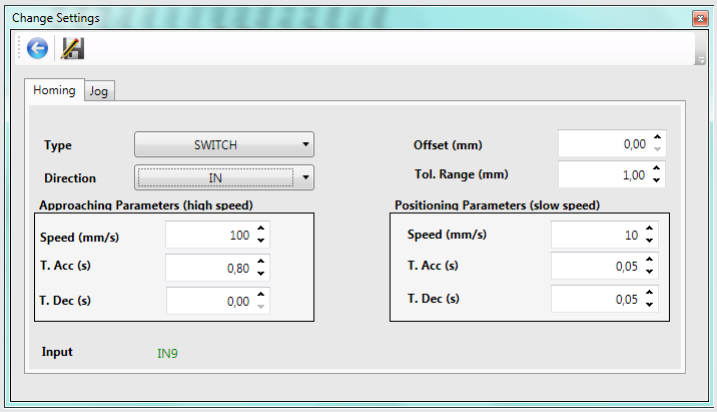
This can be used to switch from the Auto to the Manual mode.
The tests can only be performed in Manual mode.



Enables / Disables the motor

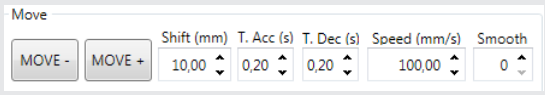
In the "Special" window, you can perform Zeroing by clicking HOMING, execute movements using the JOG- and JOG+ functions, set the zero value via the SET HOME button and real-time display of both the motion diagram and update the variables used in the current cycle.

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

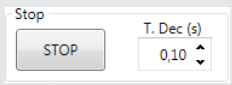


Using the  you return to the previous dialogue box.

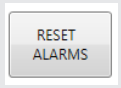
The new parameters are automatically sent to the board.



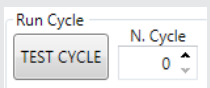
The MOVE-/MOVE+ buttons can be used to perform movements, by specifying the stroke, the acceleration and deceleration time and the speed.



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



Resets alarms



Starts the specified cycle

EN

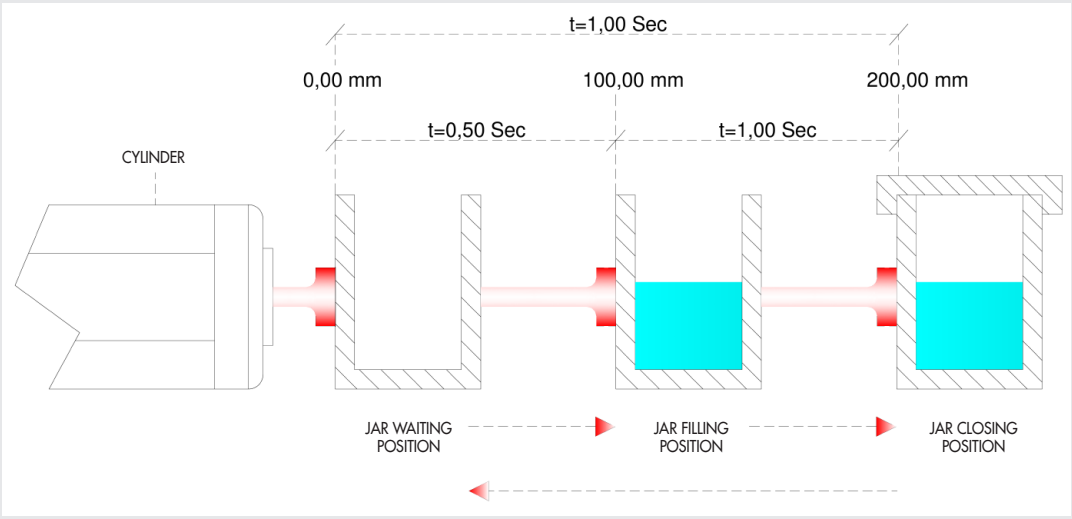
APPLICATION EXAMPLES

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***.motion 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.

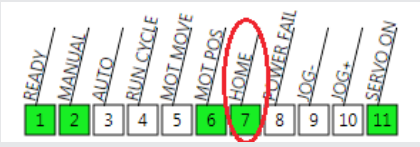
The system can be summarised in the following layout:



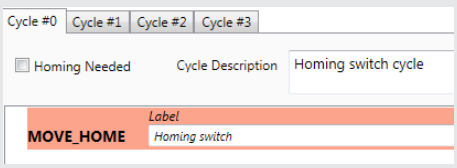
Two cycles are created. In the first one, the axis is zeroed, while in the second a work cycle is implemented. As to the execution sequence of the various cycles, please refer to the section entitled "Setup->Cycle".

Upon completion of the parameter setting in the setup section in the MWPOS 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.

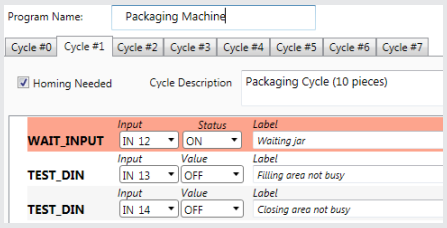


This flag is the essential condition for running the cycle when the Request Homing box has been checked.

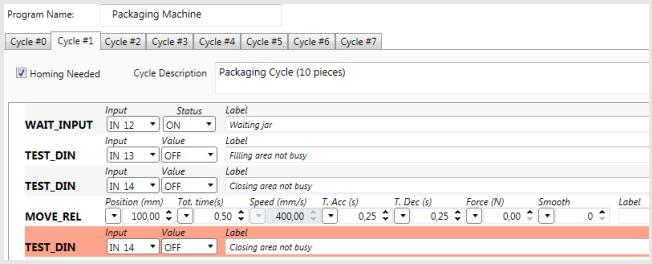


EN

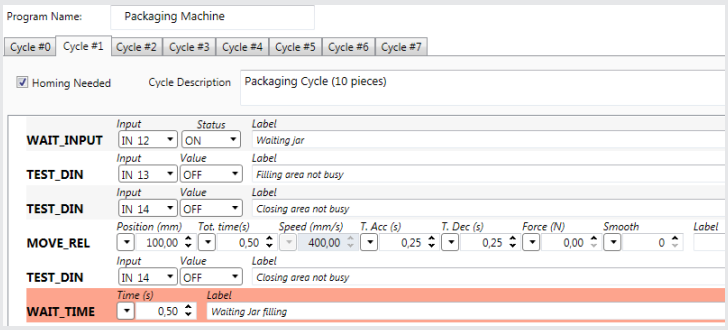
Work **Cycle #1** is now implemented.
Check the Request Homing box to ensure the execution of the cycle starting from a specific, unambiguous reference position: the axis zero.
The description of the cycle in this case is: cycle for filling and packaging 10 pieces.
Once the cycle start command has been 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.



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

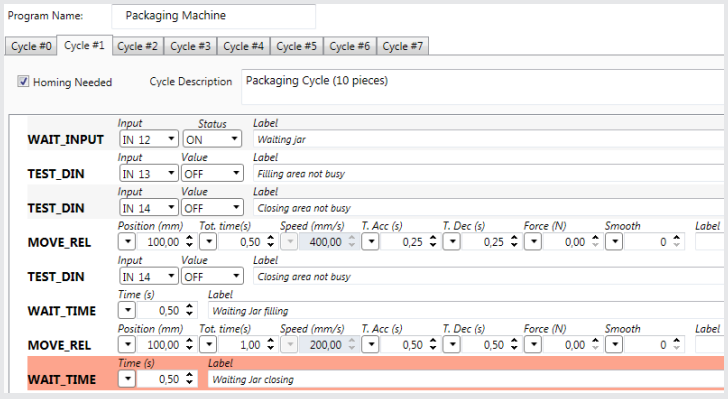


In order to allow the system to fill the jar, it is necessary to set the pause time.
The next step is WAIT_TIME: Time = 0.50 s, Comment: Wait Filling.

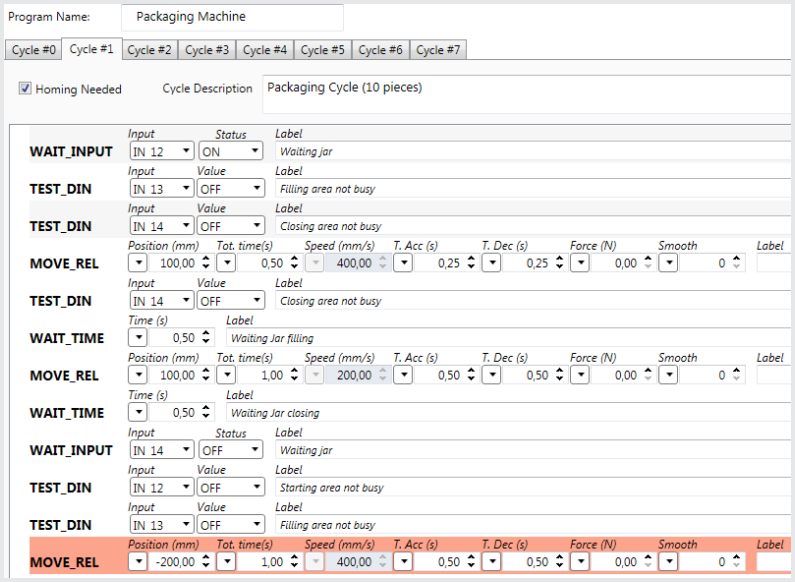


Once the jar is full, it must be positioned in the delivery area where the lid is applied.
Since the jar is no longer empty, the speed must obviously be slowed down to prevent the contents from spilling out.

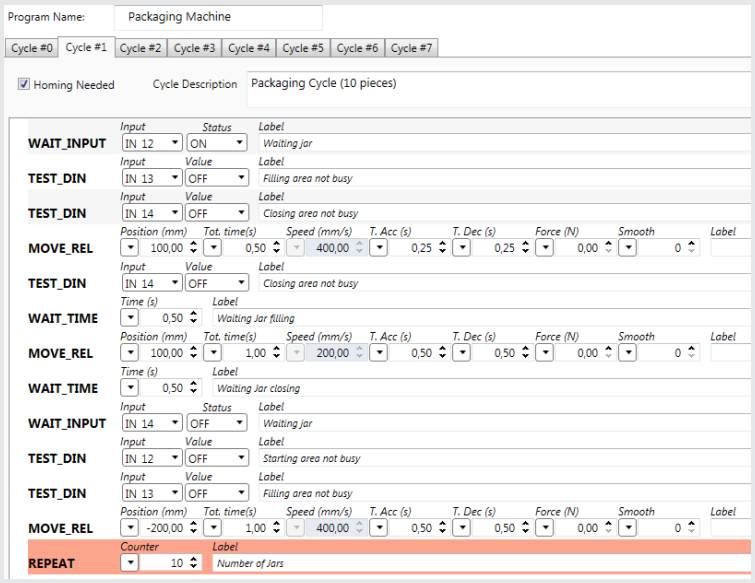
The MOVE_REL command is inserted with the following specifications: Position = 100.00 mm, Tot Time= 1.00 s, Acc T. = Dec T. = 0.50 s and comment: Arrival at Lid Area. A pause time must be set to allow time to the system to apply the lid.
The following step is therefore WAIT_TIME: Time = 0.50 s, Comment: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



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.

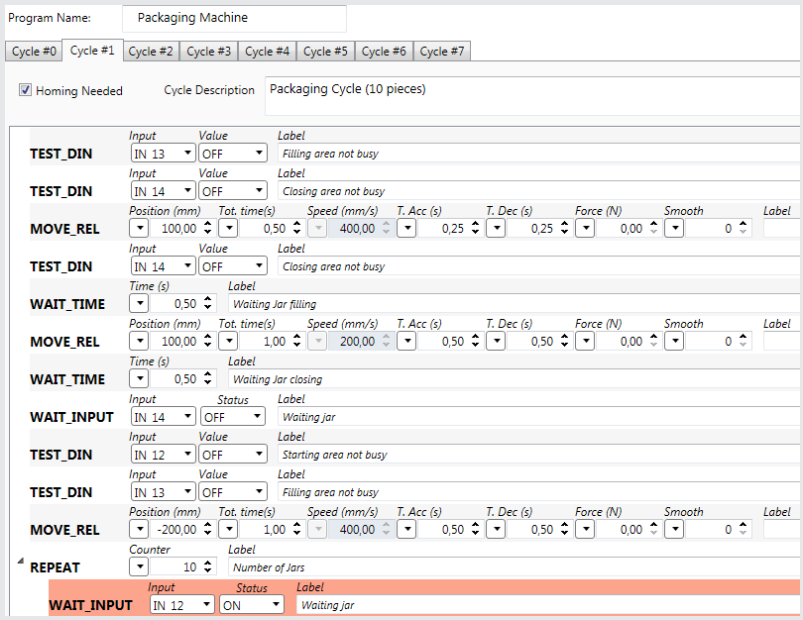


Now the system plans the execution 10 repetitions of this cycle.
First insert the repetition command through the usual procedure: click last record entered with the right mouse button and select the menu item: InsertStep->Enqueue.

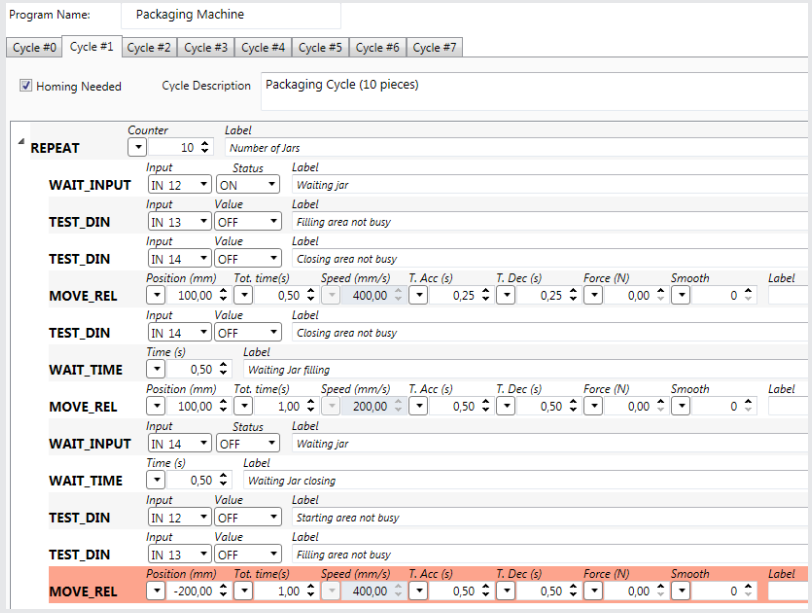


EN

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.



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.

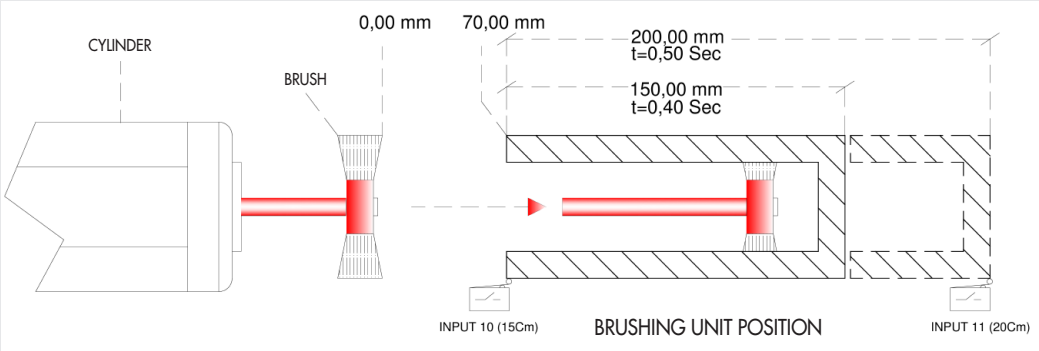


The final result is illustrated in the figure.

EN

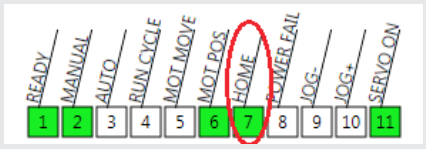
EXAMPLE 2: Brushing unit

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***.motion must provide two sets of instructions to be adapted to the selected input.

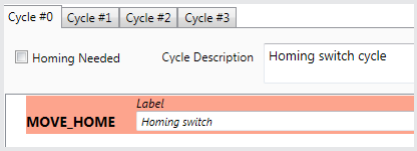


Two cycles are created: one for the zeroing, one for actual operation. Please refer to the Setup->Cycle section for further details on the execution sequence of the various cycles.

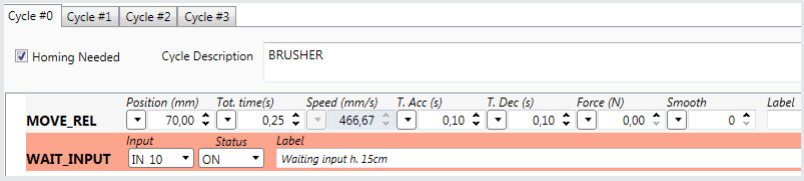
For **Cycle #0**, the comment "Zeroing cycle" is inserted as a description.
The cycle consists of a single MOVE_HOME instruction, whose comment is: Execute Zeroing.
In this type of cycle, it is very important not to check the Request Homing box. The "HOME flag" status activates only on completion of zeroing.



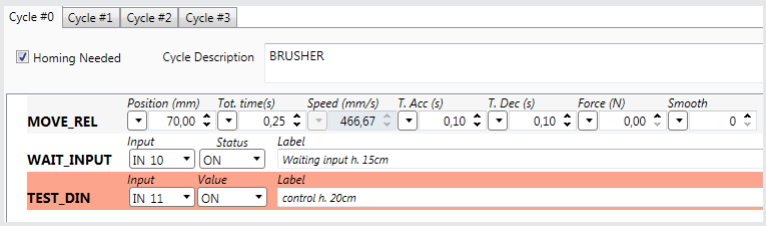
This flag is a mandatory condition for executing the cycle when the Request Homing box has been checked.



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 = 0.25 s, Acc T. = Dec T. = 0.10 s.
Comment: Arrival at Brushing Unit area.



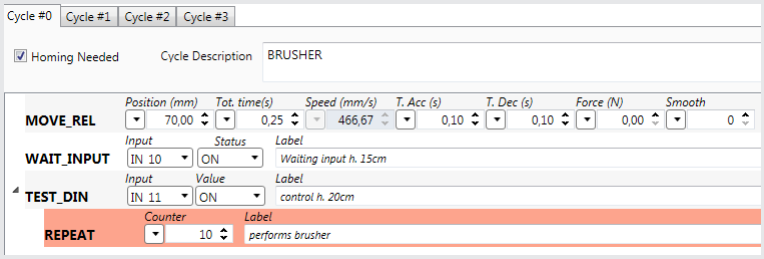
When the digital input IN 10 activates, it means that the photoelectric cell has detected the presence of an object.
Step WAIT_INPUT is used precisely for this operation. A test is performed on digital input IN 11 via the TEST_DIN step to select which of the two possible objects are to be brushed.



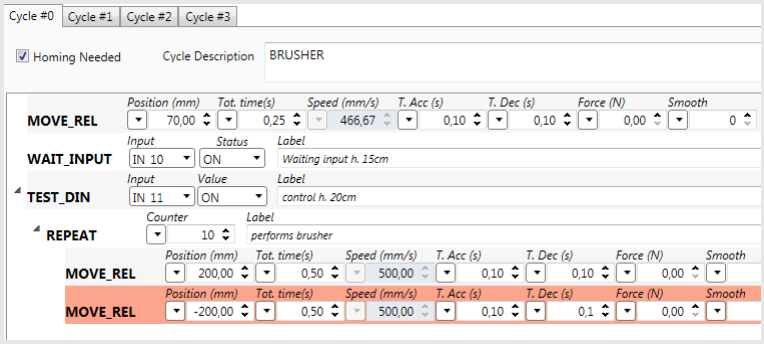
EN

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.

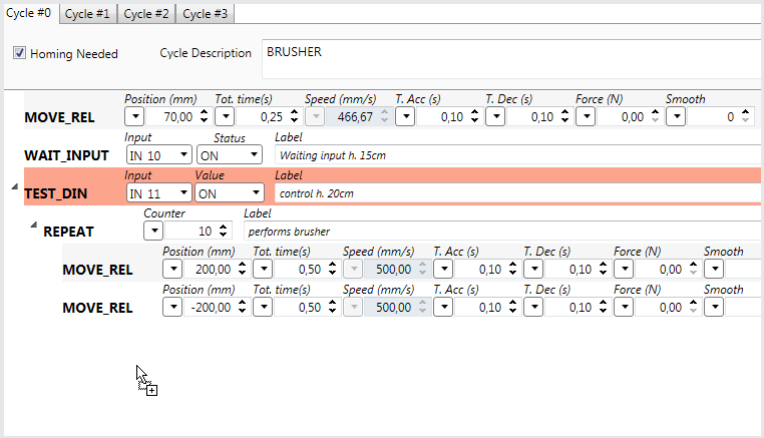
Considering the number of required repetitions, 10 is entered in the Counter field, with the comment: Execute Brushing Unit.



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.

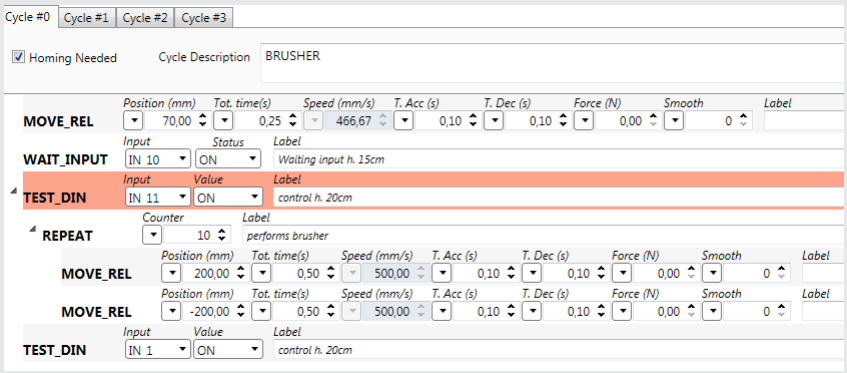


If the test result on IN 11 is negative, the brushing unit requests a dozen passes to remove any burrs on the object of a height greater than or equal to 150 mm and less than 200 mm. Considering that the instruction sequence is the one just enabled, you can enter the second part of the cycle with the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

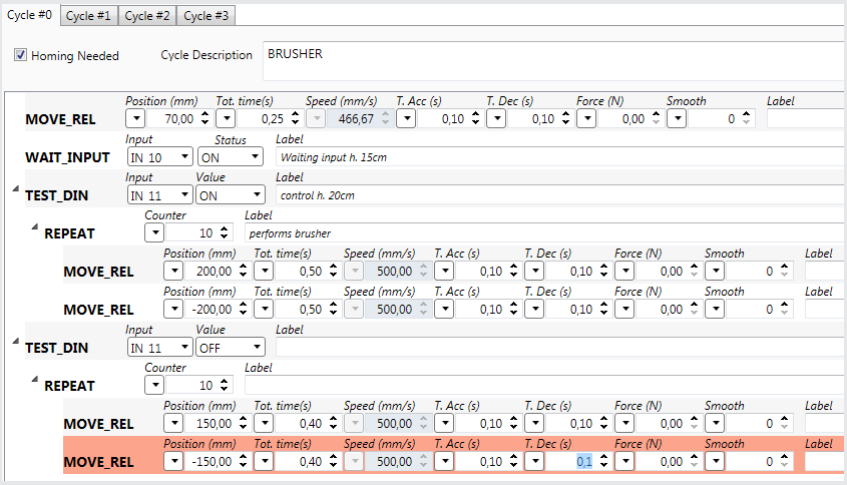


When the "+" icon appears, you can release the mouse button and all the steps will be copied and enqueued.

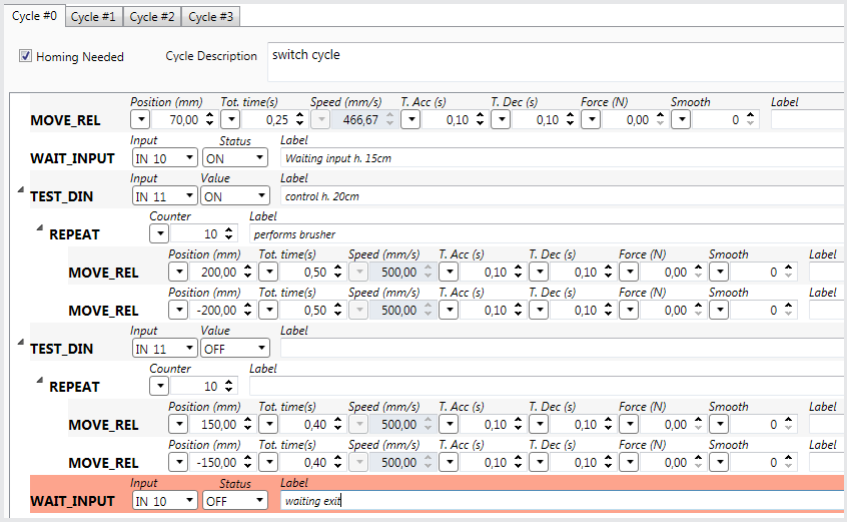
EN



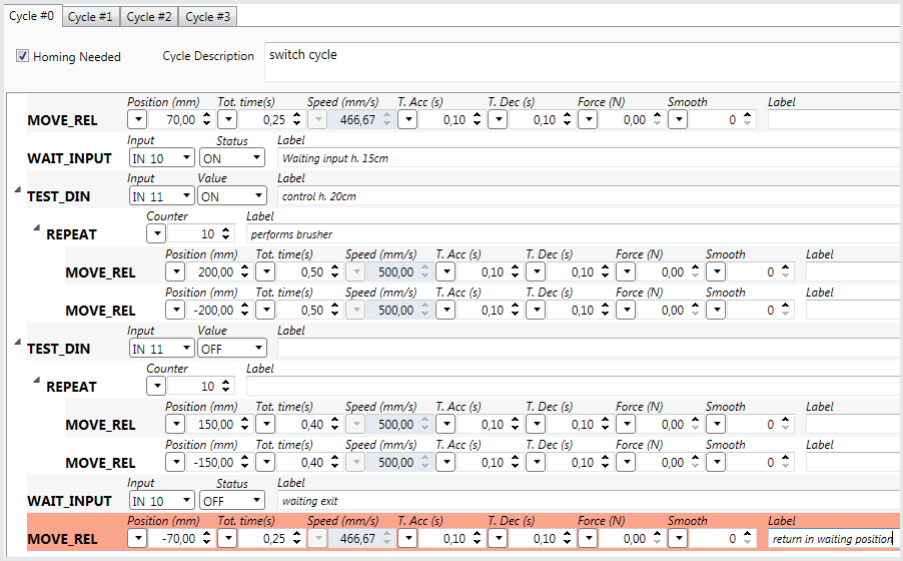
At this point, you only need to modify the just added TEST_DIN instruction.
In the MOVE_REL instructions, simply change the parameters related to the stroke to be covered (Position = 150.00 mm) and the speed at which the movement must be performed. To change the speed parameter, being a position movement,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



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.



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.

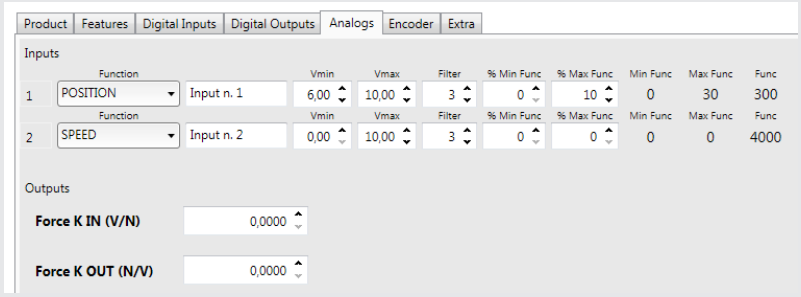


The end result is illustrated in the figure.

EN

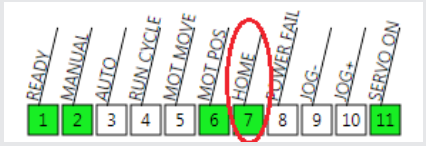
EXAMPLE 3: Humidity control

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.

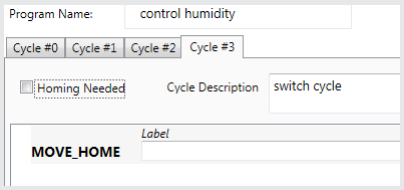


When a signal falling within the set range (6.00/10.00 V) is measured, this is converted in real time into an analogue command of the actuator position between 0.00 mm and 300.00 mm (maximum stroke of the actuator defined in the Setup parameters).
Two types of cycle are provided in the PROGRAM dialogue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

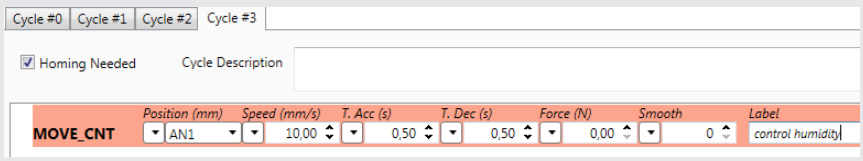
For **Cycle #0** cthe comment "zeroing cycle" is inserted as a description.
The cycle consists of a single MOVE_HOME instruction, whose comment is: Execute Zeroing.
In this type of cycle, it is very important not to check the Request Homing box. The HOME flag status comes on only upon completion of zeroing.



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



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 = 0.50s and Comment: Humidity Control.



The MOVE_CNT is a locking command and can be terminated by a STOP_GLOBAL digital input.

EN

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

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