Machine and Robot in Harmony

RPS: Robot setup and configuration

Trio Motion Technology

Trio Robotics Series

First Edition • 2020

Revision: 1.4

Trio Programming Guides are designed to aid learning of the TrioBASIC language through description and examples. Each one will cover a particular topic and discuss which commands and parameters in the TrioBASIC are required to complete the task.

A general understanding of TrioBASIC is required and it is recommended to attend an introduction to TrioBASIC training course. The programming guides are not a replacement for the TrioBASIC help files which can be found in *Motion* Perfect as well as the manual which cover each command and parameter in more detail and should be referenced when required.

Any examples given in the programming guide will work and have been tested on an isolated controller. If you choose to use these examples on a machine please take care that it will not cause damage or injury and that they are correctly included in the project changing parameters and values where required.

All goods supplied by Trio are subject to Trio's standard terms and conditions of sale.

The material in this manual is subject to change without notice. Despite every effort, in a document of this scope errors and omissions may occur. Therefore, Trio cannot be held responsible for any malfunctions or loss of data as a result.

UK | USA | CHINA | INDIA

www.triomotion.com

SAFETY WARNING

During the installation or use of a control system, users of Trio products must ensure there is no possibility of injury to any person, or damage to machinery.

Control systems, especially during installation, can malfunction or behave unexpectedly.

Bearing this in mind, users must ensure that even in the event of a malfunction or unexpected behaviour the safety of an operator or programmer is never compromised.

This document uses the following icons for your reference:

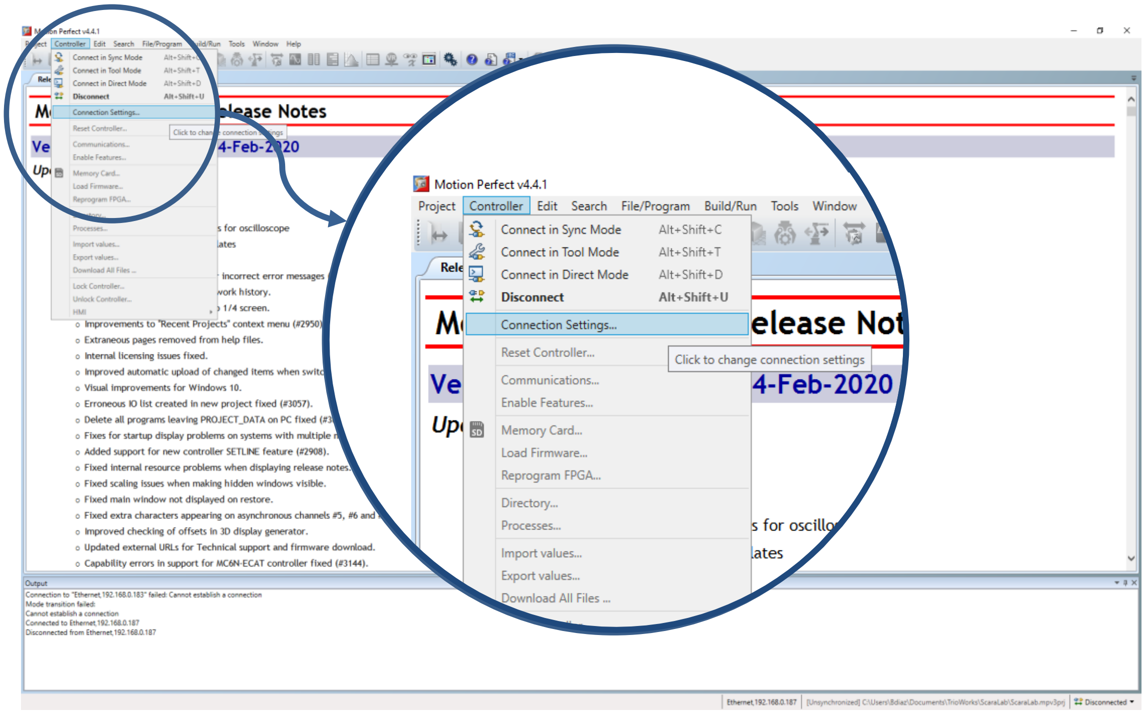
💣	📄	✪	➡
Information that relates to safety issues and critical software information.	Information to highlight key features or methods.	Useful tips and techniques.	Example programs.

Contents

1 Creating a project

To create a new project in Motion Perfect, follow these steps:

1) Launch Motion Perfect and connect to controller:



- s for oscilloscope
- lates
- incorrect error messages
- work history.
- 1/4 screen.
- Improvements to "Recent Projects" context menu (#2950)
- Extraneous pages removed from help files.
- Internal licensing issues fixed.
- Improved automatic upload of changed items when switc
- Visual improvements for Windows 10.
- Erroneous IO list created in new project fixed (#3057).
- Delete all programs leaving PROJECT_DATA on PC fixed (#3
- Fixes for startup display problems on systems with multiple r
- Added support for new controller SETLINE feature (#2908).
- Fixed internal resource problems when displaying release notes.
- Fixed scaling issues when making hidden windows visible.
- Fixed main window not displayed on restore.
- Fixed extra characters appearing on asynchronous channels #5, #6 and
- Improved checking of offsets in 3D display generator.
- Updated external URLs for Technical support and firmware download.
- Capability errors in support for MC6N-ECAT controller fixed (#3144).

```
Connection to "Ethernet,192.168.0.183" failed: Cannot establish a connection
Mode transition failed:
Cannot establish a connection
Connected to Ethernet,192.168.0.187
Disconnected from Ethernet,192.168.0.187
```

Figure 1. Connect to Controller

There are various options to choose from:

Sync Mode – A multichannel connection is made to a controller and a local project on the PC is opened. The contents of the controller and the project are synchronised so that the local copy of all programs matches those on the controller. All of Motion Perfect's tools are available and programs can be edited. The synchronisation process can involve deleting programs or copying them from the controller to the PC or vice versa.

Tool Mode – A multichannel connection is made to a controller allowing the monitoring tools within Motion Perfect to be used. This mode allows the user to see a list of the programs on the controller (so that they can be started and stopped) but does not allow editing any of the programs.

Direct Mode – A direct connection is made to a controller allowing a Terminal tool to be used for a direct interaction with the command line on the controller.

2) Connect to a controller in '*Sync Mode*' by entering the IP address or connect to a simulator (virtual environment).

It is important to distinguish virtual and real connections. The virtual environment is accessed through 127.0.0.1 address.

If Motion Perfect cannot connect to a real controller, change computer’s IPV4 IP allocation address. This can be done by following these steps:

a) Launch Control Panel >> Network and internet >> Network and Sharing Center.
b) Click on the Ethernet connection under Active Networks panel.
c) Open properties >> Internet Protocol Version 4 (TCP/IPv4).
d) Select the ‘*Use the following IP address*’ button and enter the 192.168.0.1 with subnet 255.255.255.0 and the gateway empty.
e) Click on ‘*OK*’ and close all open control panel windows.
f) Try to connect to the controller again.

2 Creating a Robot

2.1 Create ROBOT_DEFINITIONS program

In order to have RPS running, a robot has to be defined in a special program called ROBOT_DEFINITIONS. This program can be done through a graphical interface or a classic text editor.

To create a ROBOT_DEFINITIONS program, do right click on "Programs" in the Controller tree and add a new program by clicking on "New…":

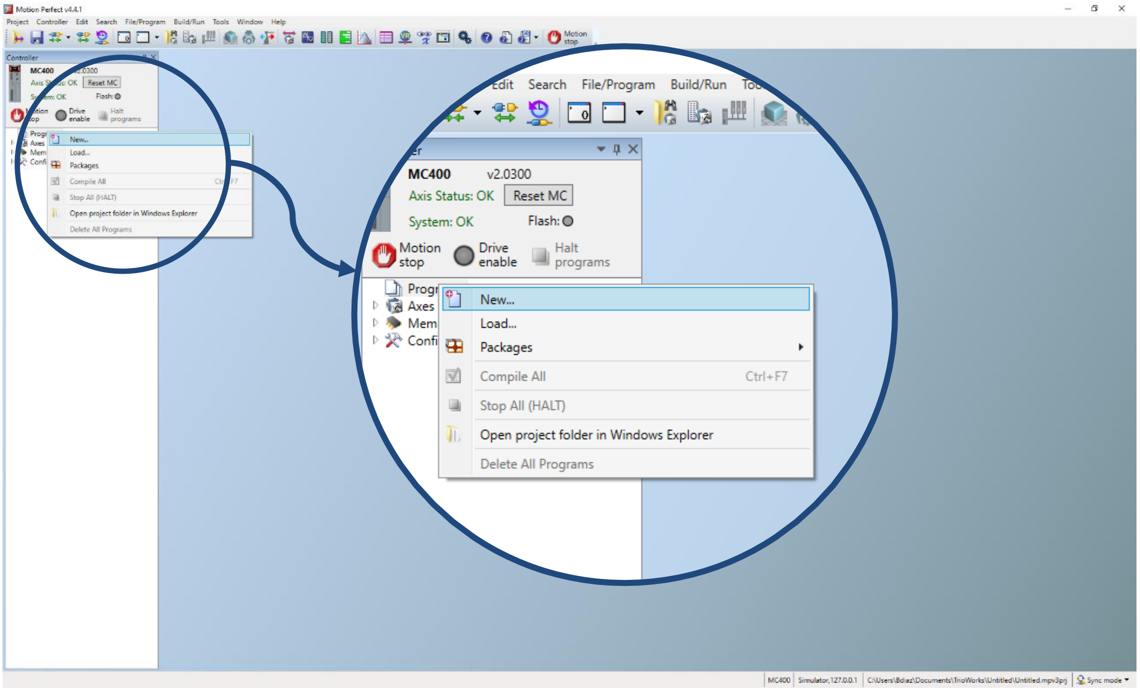
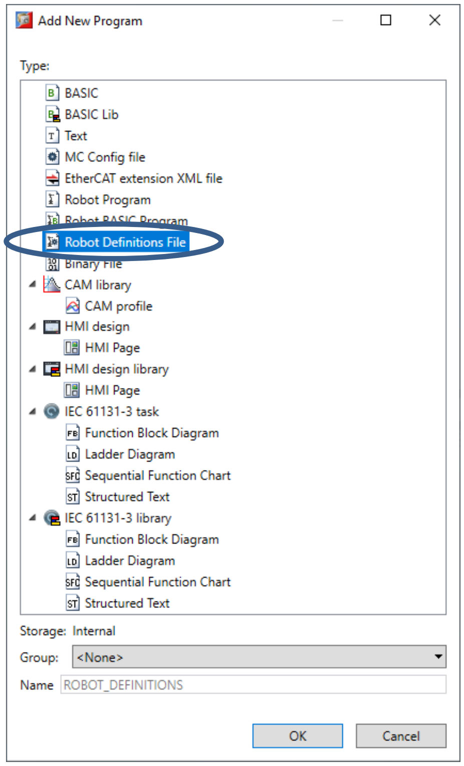


Figure 2. Create ROBOT_DEFINITIONS Program

A new window will prompt.

Select ROBOT_DEFINITIONS file and click '*OK*'.



Storage: Internal

Group: <None>

Name ROBOT_DEFINITIONS

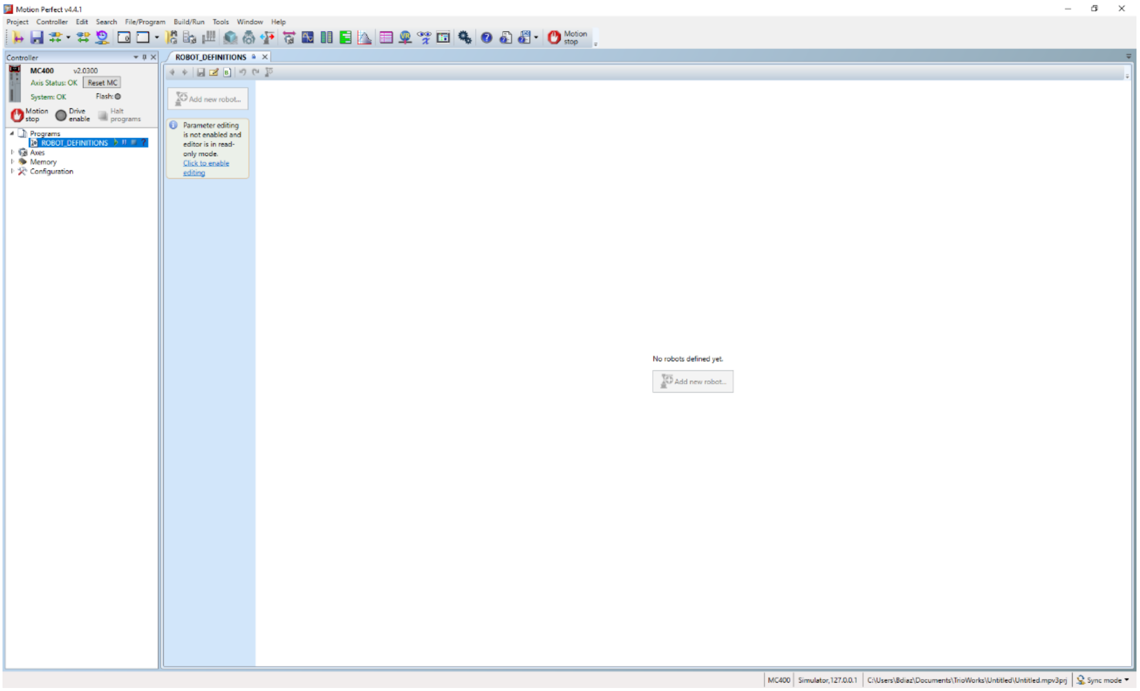
Figure 3. Select ROBOT_DEFINITIONS File

The ROBOT_DEFINITONS program is now created.

2.2 Editing the program

To edit ROBOT_DEFINITIONS, users can either directly write/change the program (unguided editor) or use the Robot Setup Wizard to setup a robot with a graphical interface.

By default, the program will open in guided mode with the following aspect:



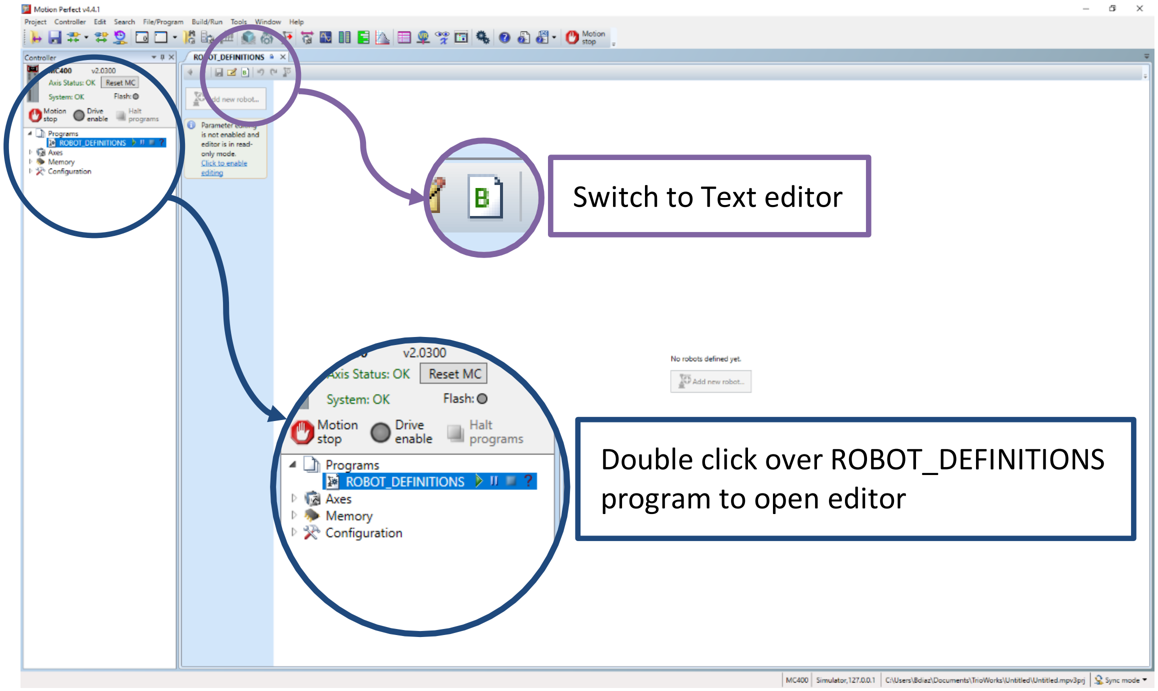
Parameter editing is not enabled and editor is in read-only mode. Click to enable editing

No robots defined yet.

Figure 4. Guided editor / Robot Setup Wizard

If ROBOT_DEFINITIONS is being closed, it could be reopened through double click over ROBOT_DEFINITIONS program in controller tree.

It is also possible to switch between guided editor and text editor through the following icon:



Parameter editing is not enabled and editor is in read-only mode. Click to enable editing

No robots defined yet.

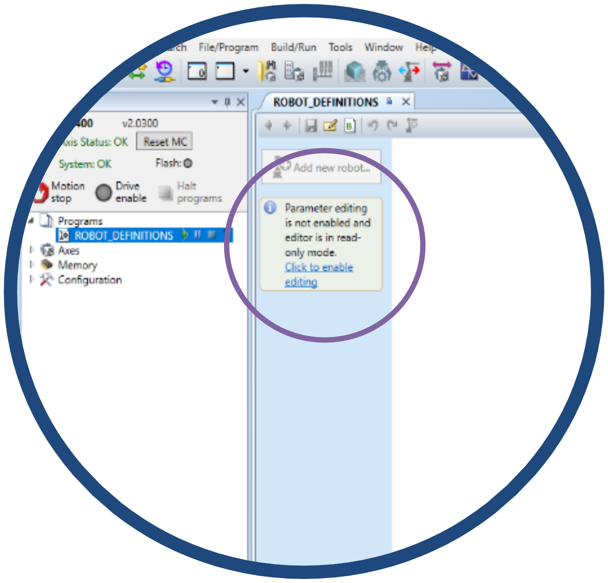
Switch to Text editor

Double click over ROBOT_DEFINITIONS program to open editor

Figure 5. Open guided editor / switch to text editor

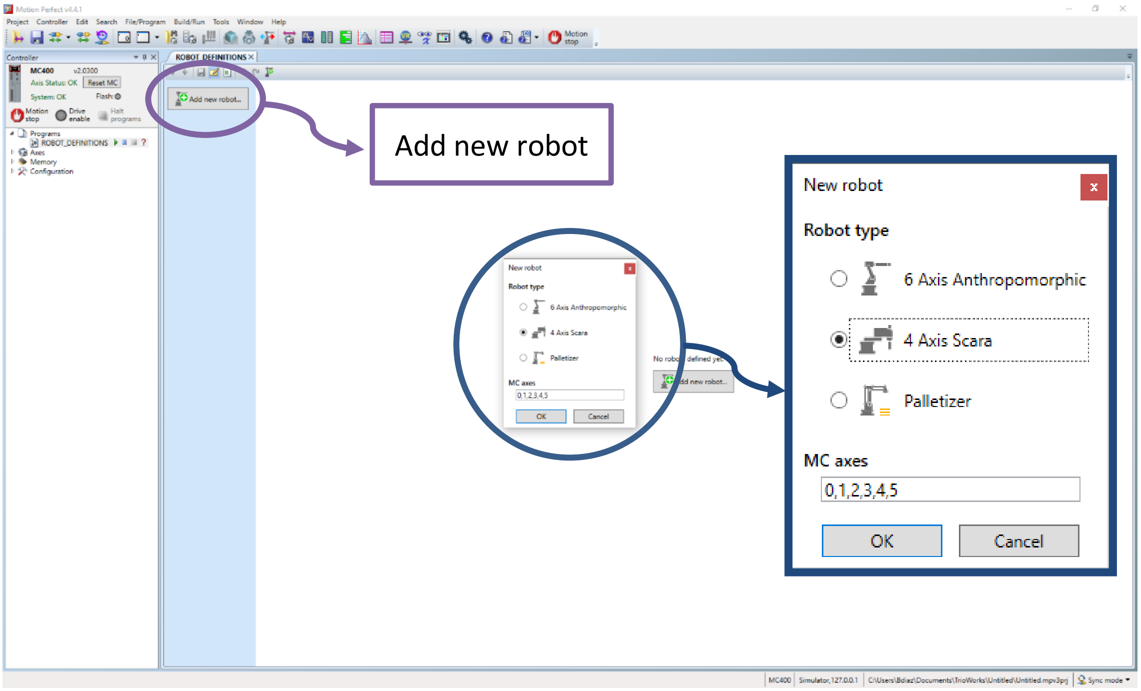
2.3 Setting up a Robot using Robot Setup Wizard

Click on ‘*Click to enable editting*’. The ‘*Add new robot*’ button will be enabled.



Parameter editing is not enabled and editor is in read-only mode.
Click to enable editing

Click on “*Add new robot“* button and select a type of robot to add.



New robot

Robot type

○ 6 Axis Anthropomorphic

● 4 Axis Scara

○ Palletizer

MC axes

0,1,2,3,4,5

OK | Cancel

Figure 6. Add a new robot

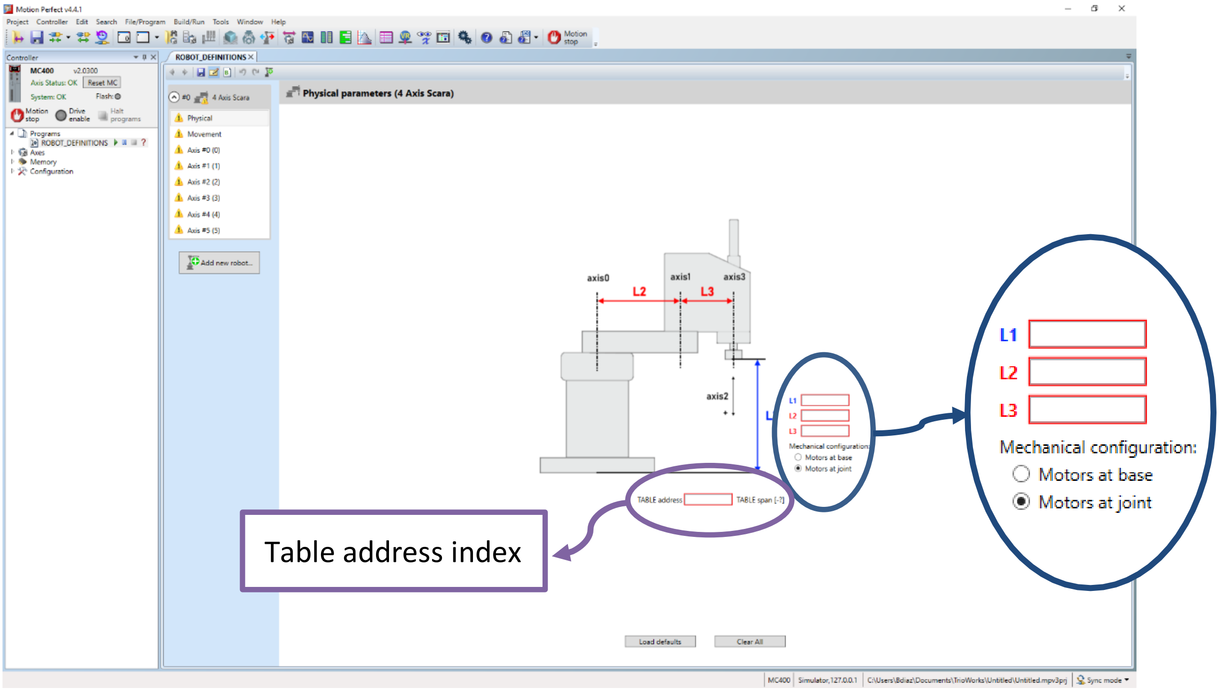
The ‘*MC axes*’ textbox allows users to select the axis numbers they want the robot to be on.

Once a robot is selected, click on ‘*OK*’.

The Physical parameters setup will appear after press “*OK*”.

Note that this may vary depending on the type of robot that is selected. This example is for a SCARA Robot.

2.3.1 Physical setup



L1 ▭
L2 ▭
L3 ▭

Mechanical configuration:

○ Motors at base

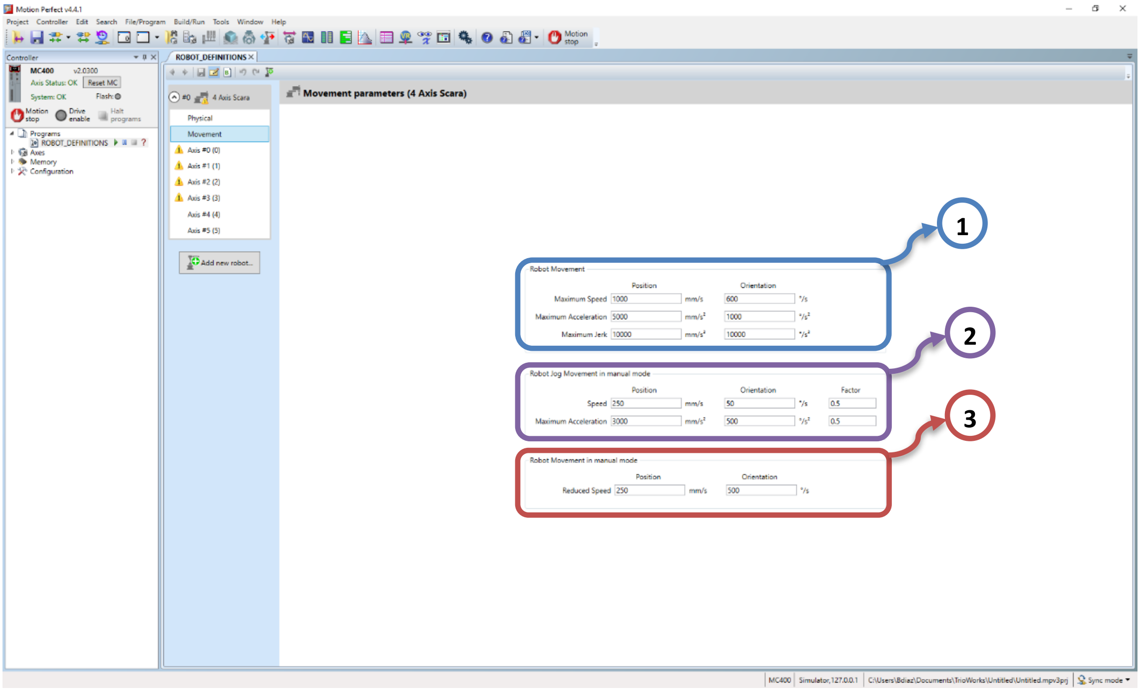
● Motors at joint

TABLE address ▭ TABLE span [-?]

Figure 7. Physical parameters – Robot Setup Wizard

Enter the required link lengths. All the necesary data required to use a robot is stored in TABLE memory on the controller. The TABLE memory is a dedicated global part of the volatile memory that can be use for general purpose. RPS uses part of the TABLE memory to store robot data. The ‘*Table address*’ parameter is the start address to store robot data. One hundred table memory indices are required by the wizard starting form the ‘*Table address*’. These indices will be locked by RPS and cannot be written to by any program.

2.3.2 Movement Setup



Robot Movement

	Position		Orientation	
Maximum Speed	1000	mm/s	600	°/s
Maximum Acceleration	5000	mm/s^2	1000	$°/s^2$
Maximum Jerk	10000	mm/s^3	10000	$°/s^3$

Robot Jog Movement in manual mode

	Position		Orientation		Factor
Speed	250	mm/s	50	°/s	0.5
Maximum Acceleration	3000	mm/s^2	500	$°/s^2$	0.5

Robot Movement in manual mode

	Position		Orientation	
Reduced Speed	250	mm/s	500	°/s

Figure 8. Movement parameters – Robot Setup Wizard

Once all the paramters in the page are setup, click on the next page, '*Movement*'.

The Movement parameters are related to the robot's World/ Linear movements.

The '*Robot Movement*' 1 panel requires the maximum speed, acceleration and Jerk of the end effector for linear movements. '*Position*' refers to the cartesian X, Y and Z movements. The units are in millimetres. '*Orientation*' refers to the Euler U, V and W movements. The units are in degrees.

The '*Robot jog movements in manual mode*' 2 panel refers to the jogging parameters of the end effector. The '*Factor*' refers to a percentage of the entered maximum jogging speed and acceleration. Speed and Acceleration factors are used while jogging to set different speeds. This value is always between 0 and 1.

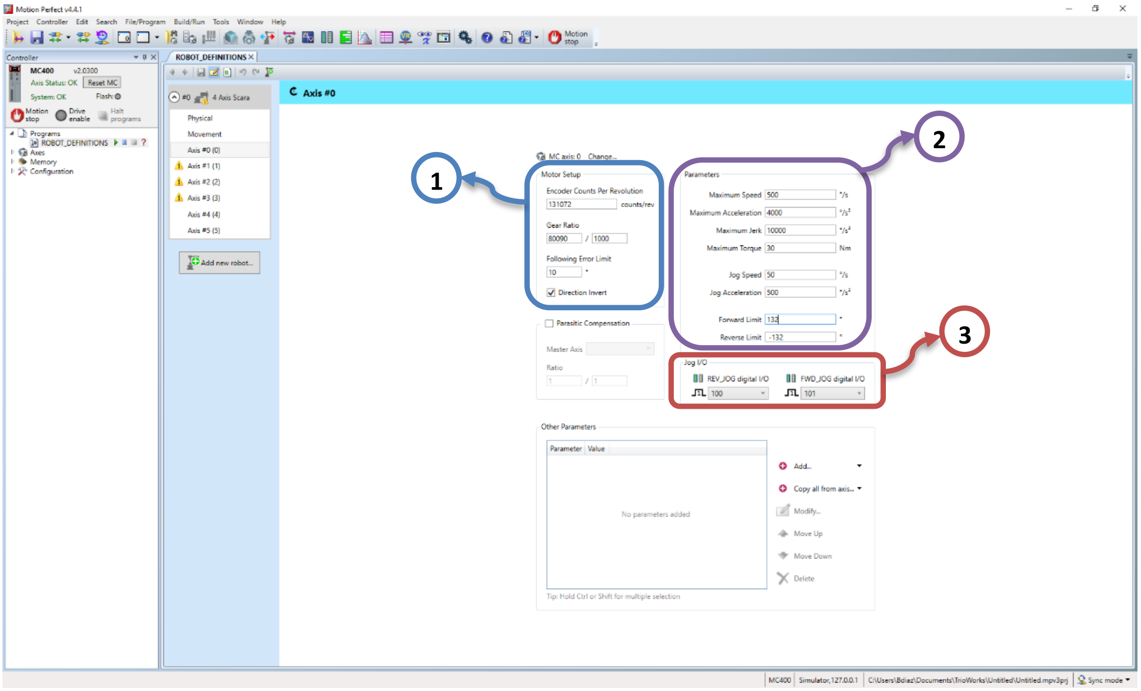
The '*Robot movement in manual mode*' 3 panel refers to the speed in manual mode. The robot will not exceed this speed when it is being operated in manual mode. Any move speed programmed higher than this will be capped to the manual speed.

For example, MOVEL GTA(0) S:=900. In this example the move speed is requested to be 900mm/s but this move will be executed at 250mm/s while in manual mode.

Note: Manual mode can be activated by calling LOOKAHEAD_FACTOR_MODE(1). LOOKAHEAD_FACTOR_MODE(0) will disable manual mode.

2.3.3 Axis Setup

Once all the movement parameters are set, click on the next page, Axis(0):



MC axis: 0 Change...

Motor Setup

Encoder Counts Per Revolution: 131072 counts/rev

Gear Ratio: 80090 / 1000

Following Error Limit: 10 °

☑ Direction Invert

☐ Parasitic Compensation

Master Axis:

Ratio: 1 / 1

Parameters

Maximum Speed	500	°/s
Maximum Acceleration	4000	°/s²
Maximum Jerk	10000	°/s³
Maximum Torque	30	Nm
Jog Speed	50	°/s
Jog Acceleration	500	°/s²
Forward Limit	132	°
Reverse Limit	-132	°

Jog I/O

REV_JOG digital I/O: 100

FWD_JOG digital I/O: 101

Other Parameters

Parameter	Value

No parameters added

Tip: Hold Ctrl or Shift for multiple selection

Figure 9. Axis parameters – Robot Setup Wizard

The ↻ icon indicates that the axis is a rotary axis. The »» icon indicates that the axis is a prismatic/linear axis. ***All rotary axes units are in degrees and all prismatic axes units are in millimetres.***

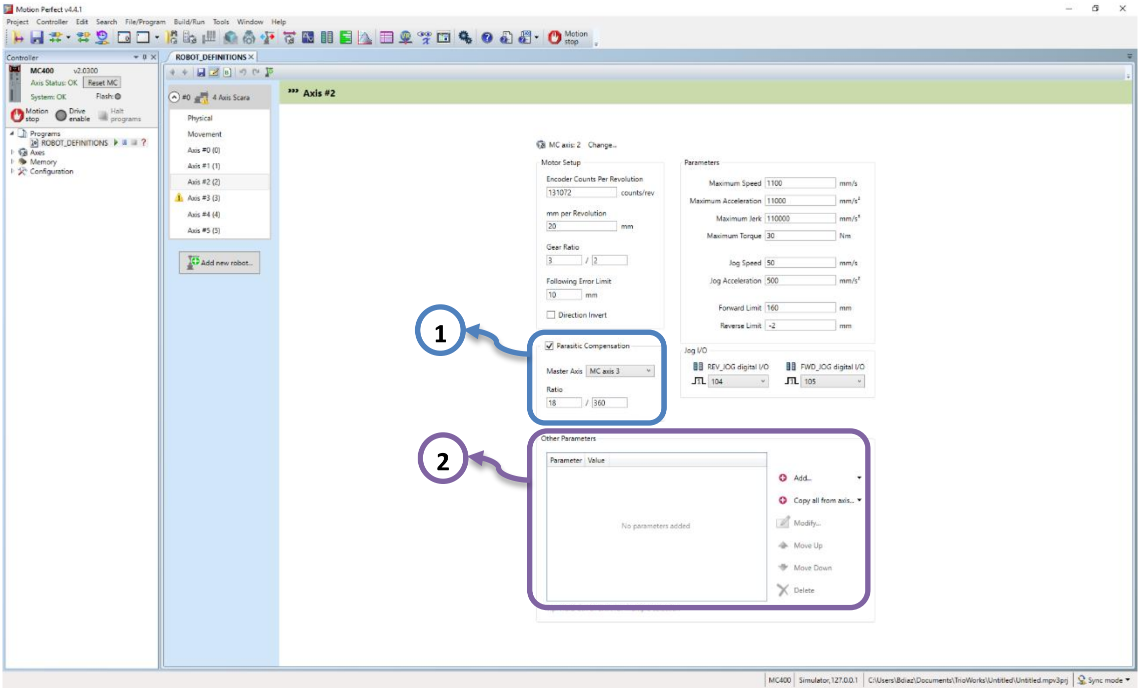
The '*Motor setup*' ① panel requires the user to enter all the necesary data to setup the AXIS_UNITS for each axis. To do so, the **Encoder edges per revolution**, **gear ratio** and the **following error limit** is required. This information is readily available in a robot's mechanical manual. If the direction of rotation of an axis does not match the direction of rotation mentioned in the physical setup, click on '*Direction Invert*' to change the direction of rotation of that axis.

The '*Parameters*' ② panel requires the user to set all the motion parameters for an axis. This includes the maximum possible **Speed** (in degrees/sec), **Acceleration**, **Jerk** and **Torque**. This information is readily available in the datasheet of the motor and driver. Similar to the movement tab, is it required to set the jog parameters. It is also necesary to set the forward and the reverse software limit for each axis.

The Maximum Toruque is not used in the controller. This feature is under development. All users will be notified when this value is used.

The maximum value parameters are used in MOVE instructions in *Automatic Mode*. Jog parameters are used while jogging the axis.

Through '*Jog IO*' 3 panel it is possible to select the jog inputs. *These inputs have to be virtual so please, make sure the correct ones are selected*. This panel contains also a button ⎍ to invert the selected inputs. The forward and inverse jog inputs are active low by default. *It is mandatory to have active high logic selected to change the default behaviour.*



MC400 v2.0300
Axis Status: OK Reset MC
System: OK Flash: ●

MC axis: 2 Change...

Motor Setup

Encoder Counts Per Revolution: 131072 counts/rev

mm per Revolution: 20 mm

Gear Ratio: 3 / 2

Following Error Limit: 10 mm

□ Direction Invert

☑ Parasitic Compensation

Master Axis: MC axis 3

Ratio: 18 / 360

Parameters		
Maximum Speed	1100	mm/s
Maximum Acceleration	11000	mm/s²
Maximum Jerk	110000	mm/s³
Maximum Torque	30	Nm
Jog Speed	50	mm/s
Jog Acceleration	500	mm/s²
Forward Limit	160	mm
Reverse Limit	-2	mm

Jog I/O

REV_JOG digital I/O: 104

FWD_JOG digital I/O: 105

Other Parameters

Parameter	Value

No parameters added

MC400 | Simulator,127.0.0.1 | C:\Users\Bdiaz\Documents\TrioWorks\Untitled\Untitled.mpv3prj

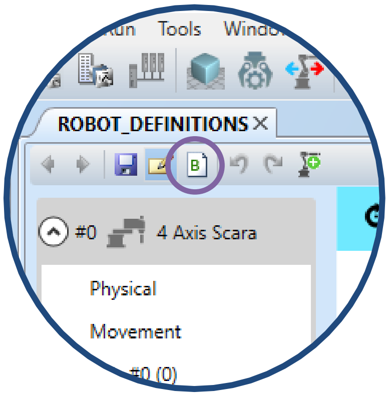
Figure 10. Parasitic compensation – Robot Setup Wizard

If there is a parasitic motion between two axes, it can be compensated.

Click on the '*parasitic compensation*' 1 option to activate its setup. Enter the master/driving axis and the ratio. For example, Axis 3 is mechanically linked to axis 2. When axis 3 rotates 1 turn, axis 2 moves 20mm. Using the above information, the MC Master axis is 3 and the ratio is 20/1.

The '*Other Parameters*' 2 panel allows users to add additional axis parameters to the setup file.

Once all the robot axes are setup, click on B to view the generated ROBOT_DEFINITIONS file.



After setting a robot, ROBOT_DEFINITIONS file has to be executed so the controller can allocate the seleced axes as a robot and the RPS system would be up and running.

Once RPS is up and running, the rest of the remaining tools of the Robotics Tool Box will be enabled and Robot programs can be created. Please, refer to Robotics Tool Box documentation or RPS to learn how to create Robot applications.

3 ROBOT_DEFINITIONS file

The definitions of all the parameters in the file are listed below (the values correspond to this particular example and may be different):

TABLE_PROTECT : This function turns on/off protection for a selected table range.
KINEMATIC_GROUP(0,-1) : Deletes a kinematic group
TABLE(0,160,325,275,(2^17)*(5000/1000)/(2*PI),(2^17)*(5/1)/(2*PI))
TABLE(5,(2^17)*(1500/1000)/20,(2^17)*(600/10)/(2*PI),1000,1000,0,0)

: Store robot link lengths and encoder edges per radian

WORLD_POS_REDUCED_SPEED : Maximum allowed linear speed in manual mode
WORLD_POS_MAX_SPEED : Maximum allowed linear Speed
WORLD_POS_MAX_ACC : Maximum allowed linear Acceleration
WORLD_POS_MAX_JERK : Maximum allowed linear Jerk
WORLD_ORI_REDUCED_SPEED : Maximum allowed orientation speed in manual mode
WORLD_ORI_MAX_SPEED : Maximum allowed orientation Speed
WORLD_ORI_MAX_ACC : Maximum allowed orientation Acceleration
WORLD_ORI_MAX_JERK : Maximum allowed orientation Jerk
WORLD_POS_JOG_SPEED : Maximum allowed linear jog speed
WORLD_POS_JOG_ACC : Maximum allowed linear Jog acceleration
WORLD_ORI_JOG_SPEED : Maximum allowed orientation jog speed
WORLD_ORI_JOG_ACC : Maximum allowed orientation jog acceleration
JOG_SPEED_FACTOR : Factor to control jogging speed from 0 to maximum.
JOG_ACC_FACTOR : Factor to control jogging acceleration from 0 to maximum.
'Start Axis 0 configuration
BASE(0)
AXIS_UNITS : Encoder edges per degree for selected axis
AXIS_MAX_SPEED : Maximum allowed axis speed
AXIS_MAX_ACC : Maximum allowed axis acceleration
AXIS_MAX_JERK : Maximum allowed axis Jerk
AXIS_MAX_JOG_SPEED : Maximum allowed jog speed
AXIS_MAX_JOG_ACC : Maximum allowed jog acceleration
AXIS_MAX_TORQUE : Maximum allowed Torque *– Not used*
AXIS_FS_LIMIT : Forward software limit
AXIS_RS_LIMIT : Reverse software limit
AXIS_MODE.16 : invert axis direction
FE_LIMIT : Following Error limit
DRIVE_FE_LIMIT : Drive Following Error
VP_MODE : Velocity Profile Mode
INVERT_IN(111,ON)
OP(111,OFF)
FWD_JOG : Forward jog input
INVERT_IN(110,ON)

```
OP(110,OFF)
REV_JOG                    : Reverse jog input
```