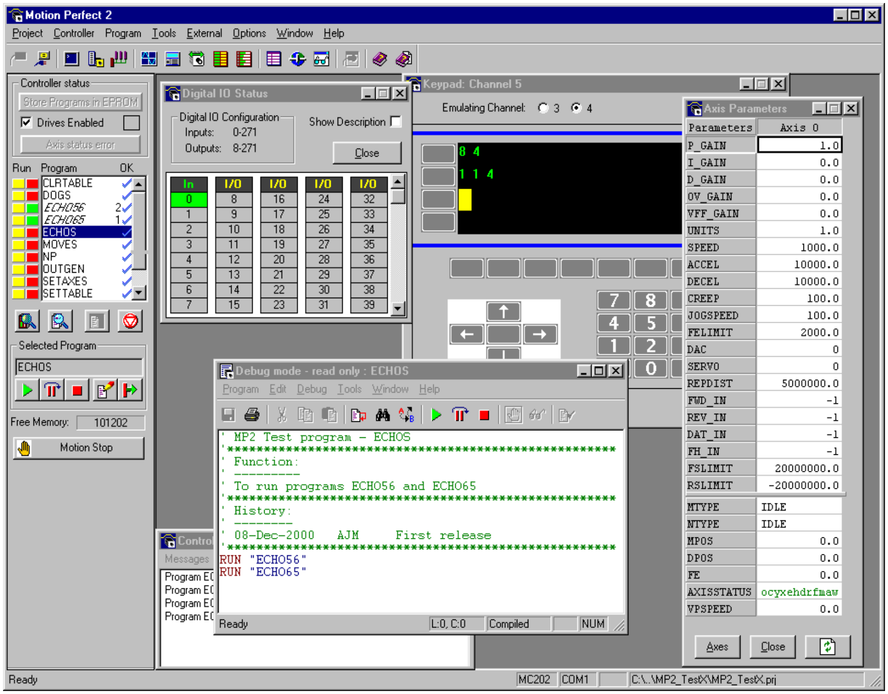
CHAPTER

10

SUPPORT SOFTWARE

Motion Perfect 2

Motion Perfect 2 is an application for the PC, designed to be used in conjunction with the *Motion Coordinator* range of multi tasking motion controllers.



☑ Drives Enabled

Run	Program	OK
🟨🟥	CLRTABLE	✓
🟨🟥	DOGS	✓
🟨🟩	*ECHO56*	2 ✓
🟨🟩	*ECHO65*	1 ✓
🟨🟥	ECHOS	✓
🟨🟥	MOVES	✓
🟨🟥	NP	✓
🟨🟥	OUTGEN	✓
🟨🟥	SETAXES	✓
🟨🟥	SETTABLE	✓

Selected Program: ECHOS

Free Memory: 101202

Digital IO Configuration
Inputs: 0-271
Outputs: 8-271

Show Description □

In	I/O	I/O	I/O	I/O
0	8	16	24	32
1	9	17	25	33
2	10	18	26	34
3	11	19	27	35
4	12	20	28	36
5	13	21	29	37
6	14	22	30	38
7	15	23	31	39

Emulating Channel: ○ 3 ● 4

```
8 4
1 1 4
```

Parameters	Axis 0
P_GAIN	1.0
I_GAIN	0.0
D_GAIN	0.0
OV_GAIN	0.0
VFF_GAIN	0.0
UNITS	1.0
SPEED	1000.0
ACCEL	10000.0
DECEL	10000.0
CREEP	100.0
JOGSPEED	100.0
FELIMIT	2000.0
DAC	0
SERVO	0
REPDIST	5000000.0
FWD_IN	-1
REV_IN	-1
DAT_IN	-1
FH_IN	-1
FSLIMIT	20000000.0
RSLIMIT	-20000000.0
MTYPE	IDLE
NTYPE	IDLE
MPOS	0.0
DPOS	0.0
FE	0.0
AXISSTATUS	ocyxehdrfmaw
VPSPEED	0.0

```
' MP2 Test program - ECHOS
'*****************************************************
' Function:
' ---------
' To run programs ECHO56 and ECHO65
'*****************************************************
' History:
' --------
' 08-Dec-2000   AJM     First release
'*****************************************************
RUN "ECHO56"
RUN "ECHO65"
```

Program EC
Program EC
Program EC
Program EC

Ready | L:0, C:0 | Compiled | NUM

Ready | MC202 | COM1 | C:\..\MP2_TestX\MP2_TestX.prj

Motion Perfect provides the user with an easy to use Windows based interface for controller configuration, rapid application development, and run-time diagnostics of processes running on the *Motion Coordinator*.

System Requirements:

The following equipment is required to use *Motion* Perfect 2.

PC

	Minimum Specification	Recommended
CPU	Pentium class processor, operating at 75MHz	Pentium class processor, operating at 200MHz
RAM	16 Mb	32Mb RAM (64Mb for Windows 2000)
Hard disk space	10 Mb	10 Mb
	For Serial Operation: Windows 98 or Windows NT v4 **For USB/ETHERNET Communications:** Windows 98, Windows ME, or Windows 2000 **PCI Bus Communications:** Windows 2000/XP	Windows 98 SE, Windows ME, Windows 2000 or Windows XP.
Display	800 x 600 256 colours	1024 x 768 16-bit colour
Communications	Single RS232 Serial Port/ PCI slot for PCI 208	RS232 serial port, USB port
Other	Mouse or similar pointing device	

Motion Coordinator

- *Motion Coordinator* controller or compatible controllers.

  Compatible controllers include:
  MC2, MC202, MC204, MC402e, Euro 205, MC206, MC216, MC224 and PCI 208

In order to use the Packet Communications mode, system software version 1.49 or higher is required.

Note: *You should always try to use the most recent version of* ***Motion Perfect****. Updates are available from your local distributor or you can download the latest version from the Trio Web site:* **WWW.TRIOMOTION.COM**

New Features for *Motion* Perfect 1 Users

If you have previously used the first version of *Motion* Perfect you will find version 2 very familiar. We have maintained a core functionality and working environment as close to the original as possible whilst adding a number of new and enhanced features.

New in Version 2

- Full Windows 32 bit Application
- Designed for 32 Bit Windows (Windows 95,98, ME, and Windows 2000)
- Full Support for Latest Trio Products
- Supports controller locking for project security
- Supports Feature Enable Codes for the Euro 205 and MC206
- New Editor Features:
  Syntax highlighting
  Error highlighting
  Offline program editing
  Windows restored after connection
- Links to External Applications
- Integrated support for CAD2Motion, DocMaker
- Enhanced Communications Features
  Protected RS-232 communications mode
  USB support for high speed communications
  Motion Coordinator simulation
- New/Enhanced Tools
- TABLE Editor
- Variable Editor
- Load / Save TABLE files
- Resizable Oscilloscope

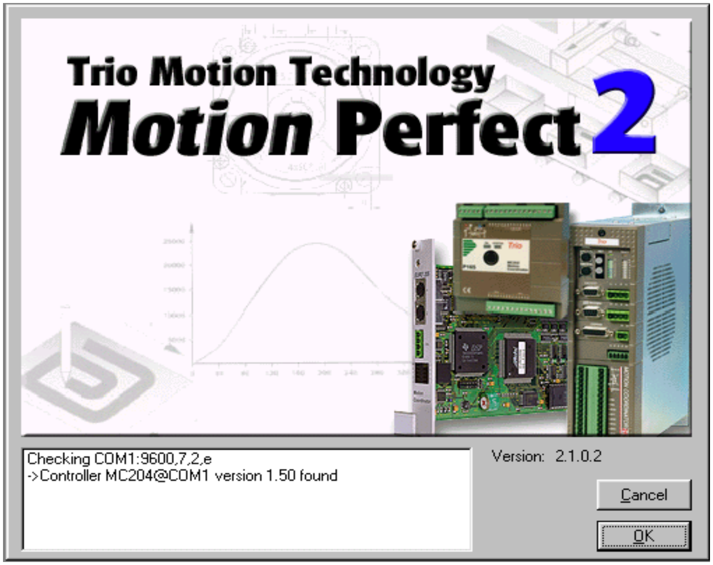
Connecting *Motion* Perfect to a controller

Motion Perfect can be connected to the *Motion Coordinator* using a serial connection, USB, Ethernet, or via the PCI bus if using the PCI 208. A suitable serial cable can be supplied by Trio Motion Technology. *Motion* Perfect may use either of the standard serial ports on a PC, COM1 or COM2, but it will connect quicker to your system if COM1 is used as this is the first port it tries. The communication method(s) can be selected under the menu Options->Communications

If you wish to edit a project but do not have a controller connected to your PC, it is possible to edit off-line by connecting to a 'virtual' controller running on your PC. See the information on the MCSimulation at the end of this chapter.

Running *Motion* Perfect 2 for the First time

Turn your PC on and enter Windows. Make sure the *Motion Coordinator* is turned on and then launch *Motion* Perfect. During initialisation you will see a splash screen such as the one below.



```
Checking COM1:9600,7,2,e
->Controller MC204@COM1 version 1.50 found
```

Figure 1: Splash screen

The splash screen features a small messages window (bottom left) which is used to display the status of the connection process. In this example *Motion* Perfect is connected to an MC204 controller via serial port COM1.

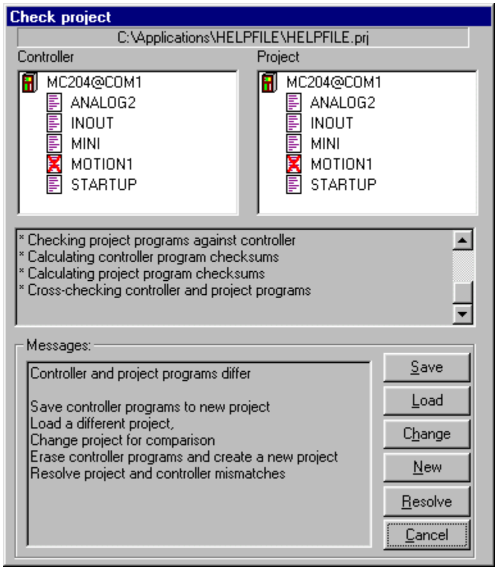
Motion Perfect 2 Projects

One of the keys to using *Motion* Perfect is to understand its concept of a "Project". The project facilitates the application design and development process, by providing a disk based copy of the multiple controller programs, parameters and data which may be used for a single motion application. Once the user has defined a project, *Motion* Perfect works behind the scenes automatically maintaining consistency between the programs on the controller and the files on the PC. When creating or editing programs on the controller they are automatically duplicated on the PC which means you do not have to worry about loading or saving programs and you can be confident that next time you connect to the controller you will have the correct information on your PC .

The Project Check Window

Whenever you connect to the controller, *Motion* Perfect will perform a **project check** to compare the programs on the controller with those defined in the current project on the pc. During the project check a window similar to the one below will be displayed. If the projects match then you will see a "project checked ok" message and an OK button to continue. If however there is any inconsistency between the controller and the PC, the display will feature a number of addition options, shown below.

You can force *Motion* Perfect to perform a project check at any time with the "Check Project" option from the project menu. (Ctrl+Alt+P)



* Checking project programs against controller
* Calculating controller program checksums
* Calculating project program checksums
* Cross-checking controller and project programs

Controller and project programs differ

Save controller programs to new project
Load a different project,
Change project for comparison
Erase controller programs and create a new project
Resolve project and controller mismatches

Project Check Options

`Save` **Save the controller contents to disk.**

If you have never connected with this controller before, and therefore do not have the project on your PC, or if there in an inconsistency in the project check and you are sure that the project on the controller is the correct version, then select SAVE to copy the programs on the controller to disk.

Note: This will of course overwrite any programs already in the PC copy of the project. If you are unsure which is the correct versiom, you should save the project with a new name to avoid overwriting any existing project programs on the PC.

`Load` **Load the PC files onto the controller**

If you are uploading a complete project from the PC to the controller, or the project check fails and you are sure the version on your PC is correct, then you should use this option to upload the entire project from the PC to the controller.

Note: The entire contents of the programs on the controller will be erased. If you are unsure, SAVE the controller contents first!

`Change` **Change the project on the PC to compare with..**

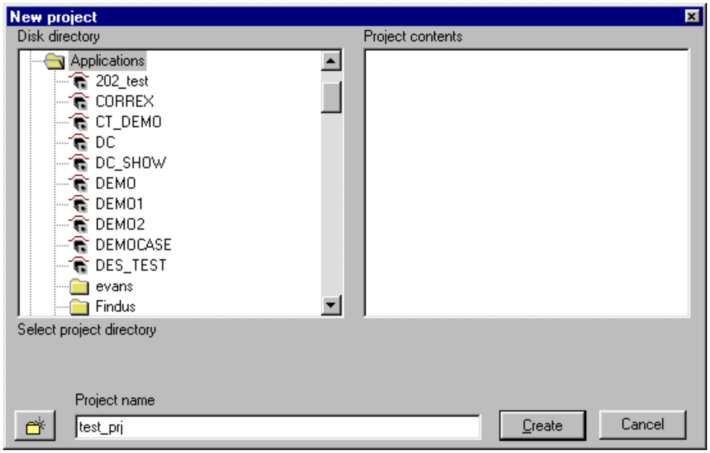
If you have been working on more than one project, the project on the controller may not match the 'last project' remembered by *Motion* Perfect. If this is the case you can use this option to select another project on the PC. Once you select an alternative, *Motion* Perfect will perform a fresh project check and the above process will be repeated.

New

Create a new project

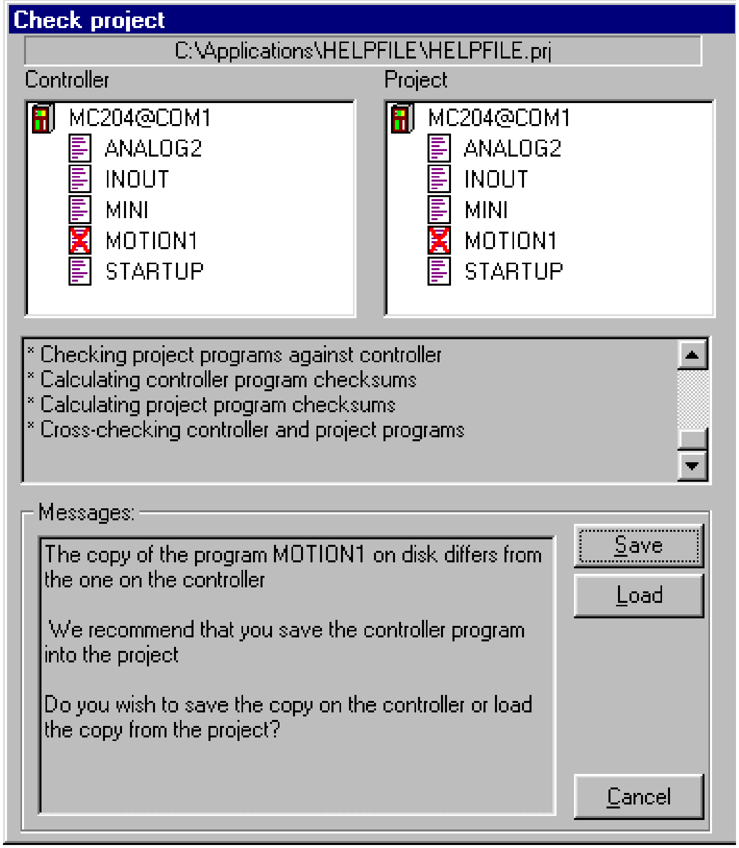
The controller contents will be erased and a new project created on the PC. You will be prompted to select a directory and project name.

When you create a new project, *Motion* Perfect will make a new directory with the project name, and within that directory a project file with the same name (the .PRJ extension is added to the filename).



Project name: test_prj

`Resolve` This option should be used when you have the correct project selected, but one or more of the files differ between the controller and PC version, or do not exist in one of the copies.



C:\Applications\HELPFILE\HELPFILE.prj

Controller	Project
MC204@COM1	MC204@COM1
ANALOG2	ANALOG2
INOUT	INOUT
MINI	MINI
☒ MOTION1	☒ MOTION1
STARTUP	STARTUP

```
* Checking project programs against controller
* Calculating controller program checksums
* Calculating project program checksums
* Cross-checking controller and project programs
```

The copy of the program MOTION1 on disk differs from the one on the controller

We recommend that you save the controller program into the project

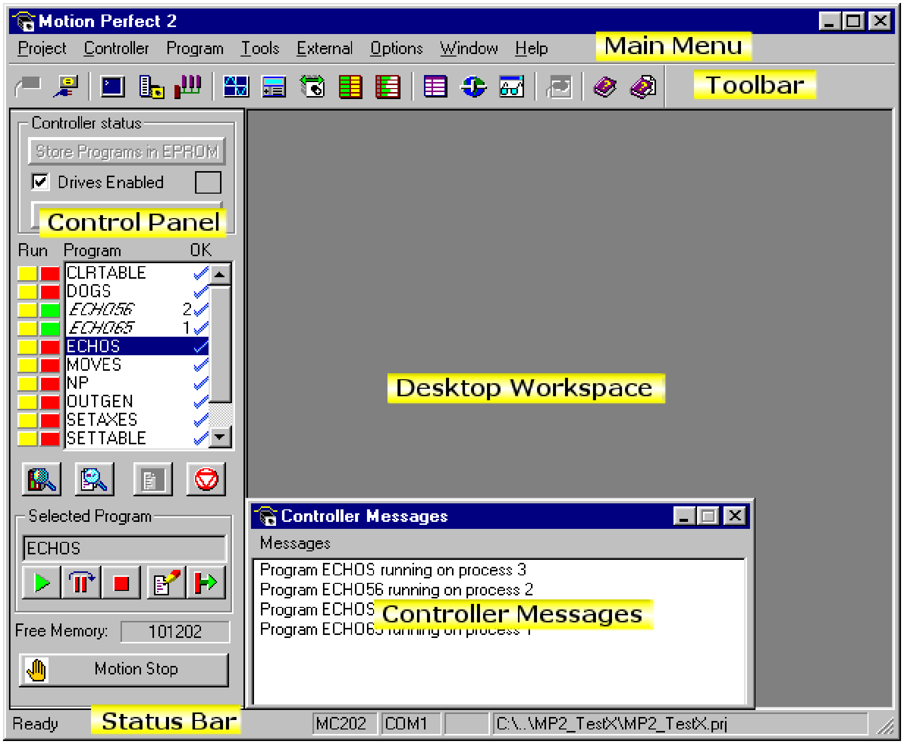
Do you wish to save the copy on the controller or load the copy from the project?

You will need to use your judgment to decide whether the disk or controller version is correct. Typically, if you are recovering the project after a comms failure or PC crash then the version on the controller should be saved. If you have modified the disk based copy of the program then you will need to load this version onto the controller.

`Cancel` Cancels the connection process and starts *Motion* Perfect in disconnected mode.

Once the project has been checked and is consistent then a backup copy of the PC project will be created.

The *Motion* Perfect Desktop



Run	Program	OK
🟨🟥	CLRTABLE	✓
🟨🟥	DOGS	✓
🟨🟩	*ECHO56*	2 ✓
🟨🟩	*ECHO65*	1 ✓
🟨🟥	ECHOS	✓
🟨🟥	MOVES	✓
🟨🟥	NP	✓
🟨🟥	OUTGEN	✓
🟨🟥	SETAXES	✓
🟨🟥	SETTABLE	✓

```
Program ECHOS running on process 3
Program ECHO56 running on process 2
Program ECHOS
Program ECHO65 running on process 1
```

Main Menu Standard Windows menu to access all features of the *Motion* Perfect application.

Toolbar Shortcut buttons to access the *Motion* Perfect tools

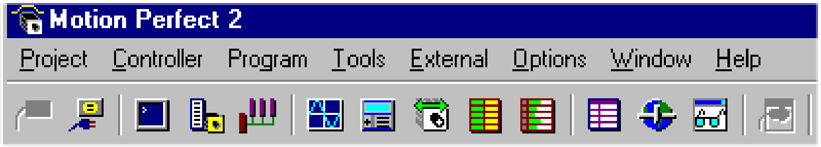
Control Panel Displays the current controller contents and provides controls for interrogating the controller status, running / editing programs

Desktop Workspace This area is used to display the user windows and tools

Controller Messages Status and error messages reported by the controller

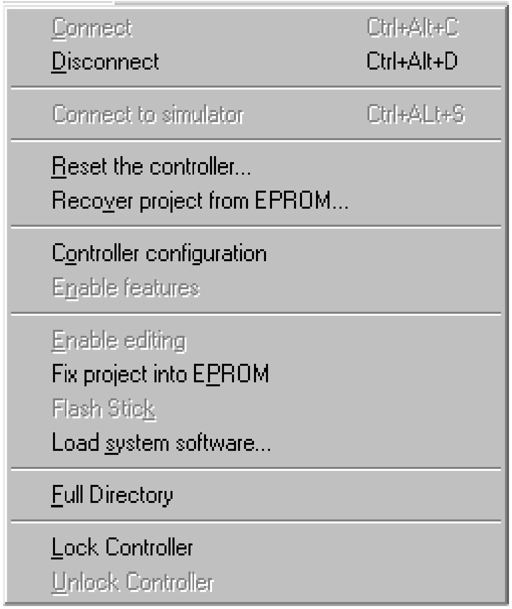
Status Bar Information about the current project and controller connection.

Main Menu


Motion Perfect 2
Project Controller Program Tools External Options Window Help

`Project`	Options for Creating, Loading & Saving *Motion* Perfect Projects, Loading/Saving program files and Table data
`Controller`	Options relating to the controller hardware, including connecting/disconnecting and checking configuration information.
`Program`	Program specific options, including creating, editing and running controller tasks.
`Tools`	Access to the main *Motion* Perfect tools. These options are also available from the Toolbar
`Options`	Configure the *Motion* Perfect Environment. Includes options to setup the communications ports and to customise the editor display.
`Window`	Control the appearance of the *Motion* Perfect desktop.
`Help`	Access the help files and version information.

Controller Menu



The controller menu contains the following items:

Connect Connect using serial communications to the controller and start the project manager. This is only available if *Motion* Perfect is currently disconnected from the controller.

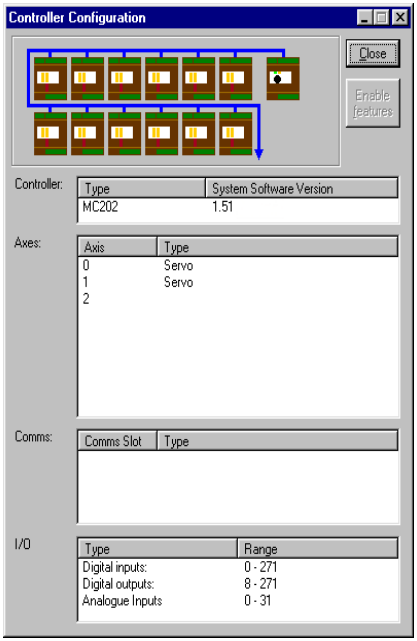
Disconnect Disconnect the serial communications, and stop using the project tools. Only available if *Motion* Perfect 2 is currently connected to the controller.

Reset Controller Perform a software-reset (EX) on the controller. This will cause *Motion* Perfect 2 to disconnect from the controller

Recover Project from EPROM Reset the controller and restore the programs which were previously stored in the EPROM

Controller Configuration

This screen interrogates the hardware and displays the configuration information reported back by the controller.



Controller:

Type	System Software Version
MC202	1.51

Axes:

Axis	Type
0	Servo
1	Servo
2	

Comms:

Comms Slot	Type

I/O

Type	Range
Digital inputs:	0 - 271
Digital outputs:	8 - 271
Analogue Inputs	0 - 31

Looking at the example screen shown here from top to bottom:

Controller: We are connected to a *Motion Coordinator* MC202

Software Version: The controller is running version 1.51 of the system software.

Axis Types A list of the types of all available axes.

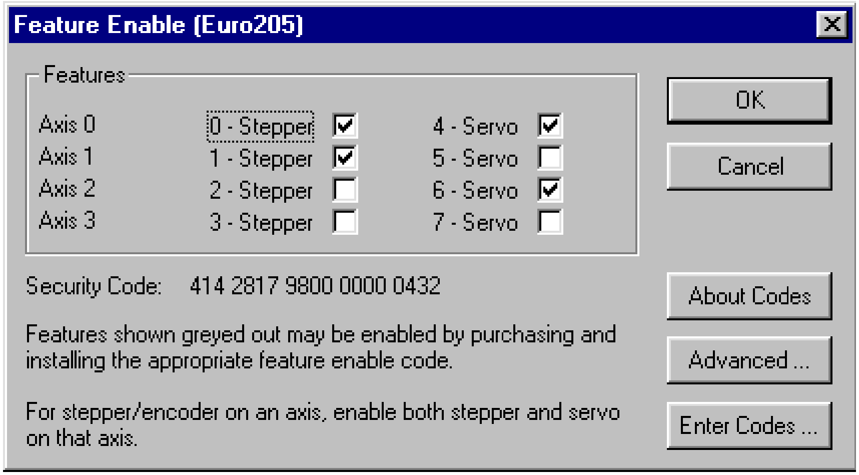
Comms Boards If the controller is fitted with any of the extended / communications daughter boards, that capability will be indicated here.

I/O The channel range available for each type of I/O. Remember that on many *Motion Coordinator*s the channels are shared, i.e. if Output 15 is available, then it implies that Input 15 is also available and shares the same connector.

Feature Enable

Certain *Motion Coordinators*, such as the EURO205 and MC206, have the ability to unlock additional axes by entering a "Feature Enable Code".

When you access the Feature Enable dialog, you will be presented with a display similar to the following:



Feature Enable (Euro205)

Features

Axis 0	0 - Stepper	☑	4 - Servo	☑
Axis 1	1 - Stepper	☑	5 - Servo	□
Axis 2	2 - Stepper	□	6 - Servo	☑
Axis 3	3 - Stepper	□	7 - Servo	□

Security Code: 414 2817 9800 0000 0432

Features shown greyed out may be enabled by purchasing and installing the appropriate feature enable code.

For stepper/encoder on an axis, enable both stepper and servo on that axis.

This display illustrates the feature which are currently available. If the codes for additional features have been purchased, the relevant boxes will be available for checking.

Enabling Additional Features

To enable a feature you must enter a Feature Enable Code, which is unique to each controller and feature. To obtain a Feature Enable Code, you will need to specify the feature required and the security code for the specific controller to be updated. The order for the required codes should be FAXed to Trio or an authorised Trio distributor.

Security Code

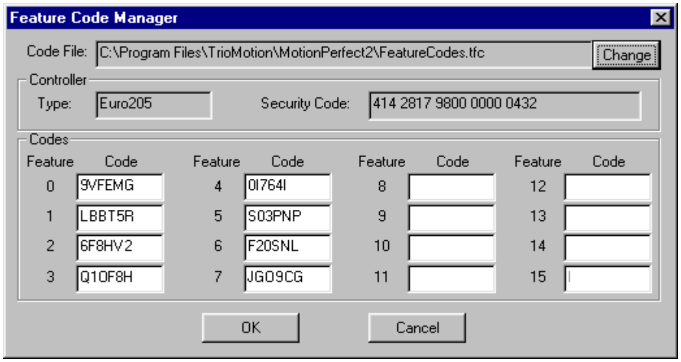
Controllers with features which can be enabled each have a unique security code number which is implanted when the unit is manufactured. This security code number is displayed on the above screen (as highlighted right).

Security Code: 070 7717 9800 0000 0281

Features shown greyed out may be enabled by p
installing the appropriate feature enable code.

Once you have the required codes, select the Enter Codes ... button.

A dialog similar to the following example will appear.



Code File: C:\Program Files\TrioMotion\MotionPerfect2\FeatureCodes.tfc

Controller

Type: Euro205

Security Code: 414 2817 9800 0000 0432

Codes

Feature	Code	Feature	Code	Feature	Code	Feature	Code
0	9VFEMG	4	0I764I	8		12	
1	LBBT5R	5	S03PNP	9		13	
2	6F8HV2	6	F20SNL	10		14	
3	Q10F8H	7	JG09CG	11		15	

Each feature requested has a feature number. Enter the relevant code for each feature number, being careful to enter the characters in upper case. Take care to check that 0 (zero) is not confused with O and 1 (one) is not confused with the letter I.

Feature Code File

Motion Perfect stores all of the Feature Enable Codes of which it is aware in a file called "`FeatureCodes.TFC`". By default this file is located in the same directory as the *Motion* Perfect 2 executable file.

`Enable Editing` Restore the power-up state of a controller currently starting from EPROM to run from RAM and allow editing.

`Fix Project into EPROM` Store the programs in RAM into the controllers flash-eprom memory. The startup state for each program will not be changed.

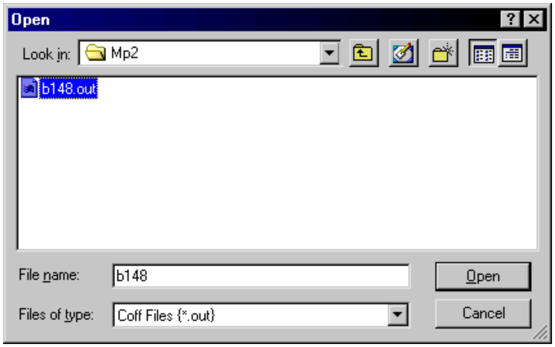
`Full Directory` Display a complete listing of all files on the controller, details of memory used and the run status of each program.

Loading New System Software

*Motion Coordinator*s feature a flash eprom for storage of both user programs and the system software. From *Motion* Perfect 2 it is possible to upgrade the software to a newer version using a system file supplied by Trio.

NOTE: ***We do not advise that you load a new version of the system software unless you are specifically advised to do so by your distributor or by Trio.***

When you select the 'Load System Software' option from the controller menu, you will first be presented with a warning dialog to ensure you have saved your project and are sure you wish to continue. When you press OK you will be presented with the standard Windows file selector to choose the file you wish to load.



File name: b148

Files of type: Coff Files (*.out)

Each *Motion Coordinator* controller has its own system file, identified by the first letter of the file name.

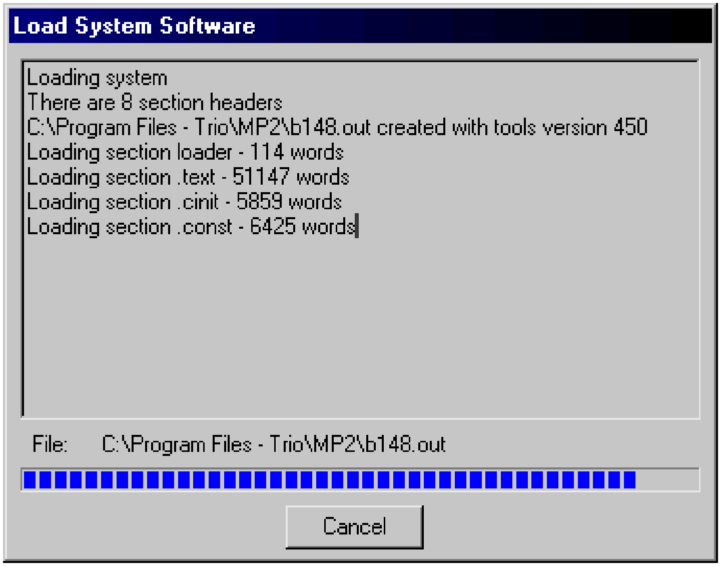
System Software File Prefix Codes:

Filename	Controller Type
`Annn.OUT`	MC2
`Bnnn.OUT`	MC204
`Cnnn.OUT`	MC216
`Dnnn.OUT`	MC402
`Ennn.OUT`	Euro205
`Fnnn.OUT`	MC202 / Motion Module
`Gnnn.OUT`	MCW202
`Hnnn.OUT`	MC402e
`Jnnn.OUT`	MC206

You must ensure that you load only software designed for your specific controller, other versions will not work

When you have chosen the appropriate file you will be prompted once again to check that you wish to continue. Press OK to start the download process.

Downloading will take several minutes, depending on the speed of your PC. During the download, you should see the progress of each section updated as follows:-



```
Loading system
There are 8 section headers
C:\Program Files - Trio\MP2\b148.out created with tools version 450
Loading section loader - 114 words
Loading section .text - 51147 words
Loading section .cinit - 5859 words
Loading section .const - 6425 words
```

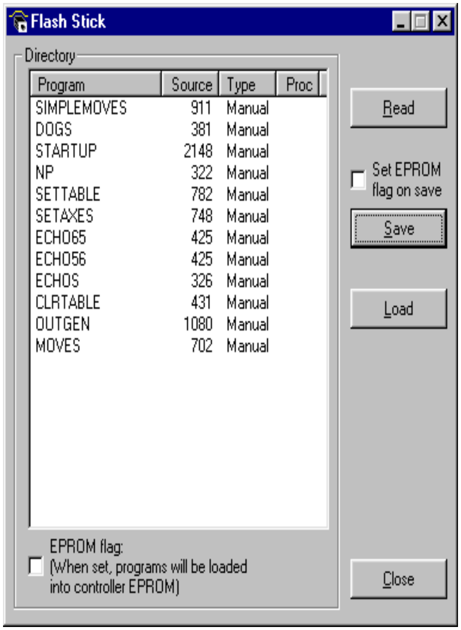
File: C:\Program Files - Trio\MP2\b148.out

When the download is complete, a checksum is performed to ensure that the download process was successful. If it saw you will be presented with a confirmation screen and asked if you wish to store the software into Eprom.

When you press Yes, the controller will take a few moments to fix the project into the eprom and you can then continue as normal.

At this point you can check the controller configuration to confirm the new software version.

Flashstick support



Program	Source	Type	Proc
SIMPLEMOVES	911	Manual	
DOGS	381	Manual	
STARTUP	2148	Manual	
NP	322	Manual	
SETTABLE	782	Manual	
SETAXES	748	Manual	
ECHO65	425	Manual	
ECHO56	425	Manual	
ECHOS	326	Manual	
CLRTABLE	431	Manual	
OUTGEN	1080	Manual	
MOVES	702	Manual	

□ EPROM flag:
(When set, programs will be loaded
into controller EPROM)

□ Set EPROM flag on save

This only applies to controllers which are fitted with a flashstick socket (eg MC206).

When a controller with flashstick support is powered on with a flashstick inserted, then the controller will automatically load the programs from the flashstick into the controller RAM.

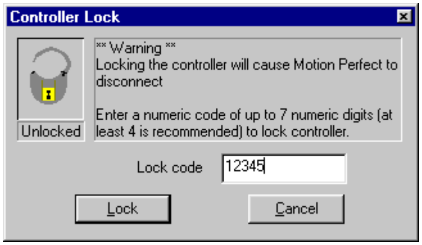
The **Read** button will read the directory of the flashstick and display it. As the directory is automatically read when the tool window is created, this button only needs to be used when the flashstick is changed.

The **Save** button will save the programs on the controller to the flashstick eassing any programs already stored on the flashstick. If the **set EPROM fag on save** box is checked then a flag is set on the flashstick which makes the controller store the programs on the flashstick in controller EPROM as well as in controller RAM at power-up.

The **Load** button will load the programs from the flashstick onto the controller. This is done by resetting the controller.

Lock / Unlock

Lock Controller


Controller Lock

** Warning **
Locking the controller will cause Motion Perfect to disconnect

Enter a numeric code of up to 7 numeric digits (at least 4 is recommended) to lock controller.

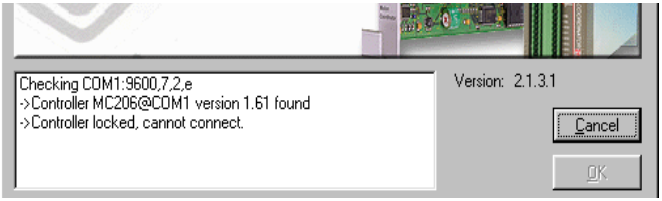
Lock code 12345

Locking the controller will prevent any unauthorised user from viewing or modifying the programs in memory.

You simply need to enter a numeric code (up to 7 digits). This value will be encoded by the system and used to lock the directory structure. The lock code is held in encrypted form in the flash memory of the *Motion Coordinator*.

This can also be achieved by issuing the LOCK(lock_code) command from the controller's command line.

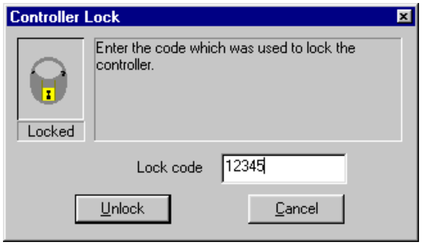
Once the *Motion Coordinator* is locked it is not possible to list, edit or save any of the controller programs. You cannot connect to the controller with *Motion* Perfect 2, although the terminal screen and unlock dialog will still be available.


```
Checking COM1:9600,7,2,e
->Controller MC206@COM1 version 1.61 found
->Controller locked, cannot connect.
```

Version: 2.1.3.1

WARNING: ***If you forget the lock code there is no way to unlock the controller. You will need to return it to Trio or a distributor to have the lock removed.***

Unlock Controller


Controller Lock

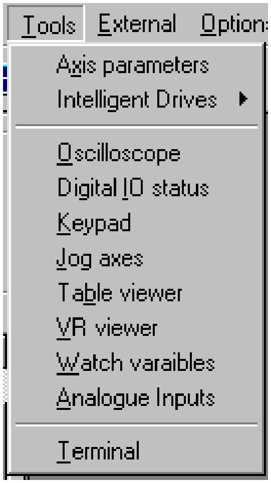
Enter the code which was used to lock the controller.

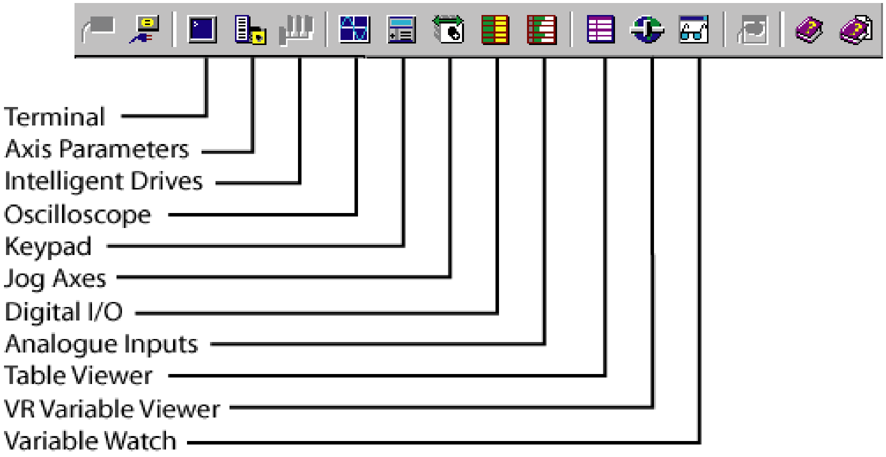
Lock code 12345

In order to unlock the controller you need to enter the same numeric code which was used to lock it. Once the unlock code is entered it will be possible to gain full access to the programs in memory.

Motion Perfect Tools

The *Motion* Perfect tools can be accessed from either the Tools Menu or the Toolbar buttons



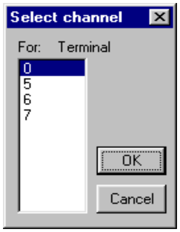


Terminal

The terminal window provides a direct connection to the *Motion Coordinator*. Most of the functions that must be performed during the installation, programming and commissioning of a system with a *Motion Coordinator* have been automated by the options available in the *Motion* Perfect menu options. However, if direct intervention is required the terminal window may be used.

Selecting a communications channel

If *Motion* Perfect is connected to the controller, you will be prompted to select from any of the active user communications channels:



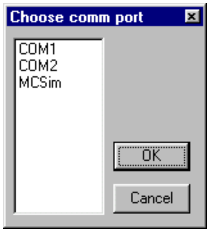
For this example we will select channel 0 which is used for the *Motion Coordinator* Command Line interface.

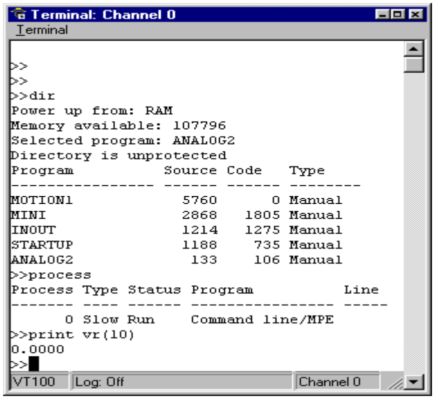
Using the terminal without a *Motion* Perfect Connection

If *Motion* Perfect is not connected to the controller, then the terminal window may be used to talk to the controller in a “No *Motion* Perfect connection”. In this mode the terminal window has complete control over the serial link.

Once connected to the controller, the terminal window opens a communications channel to the controller. The user must select the required channel.

If this channel is already in use, either by another terminal window or a keypad emulation then access will be denied.





```
>>
>>
>>dir
Power up from: RAM
Memory available: 107796
Selected program: ANALOG2
Directory is unprotected
Program          Source Code   Type
---------------- ------ ------ --------
MOTION1            5760      0 Manual
MINI               2868   1805 Manual
INOUT              1214   1275 Manual
STARTUP            1188    735 Manual
ANALOG2             133    106 Manual
>>process
Process Type Status Program          Line
------- ---- ------ ---------------- -----
      0 Slow Run    Command line/MPE
>>print vr(10)
0.0000
>>
```

Axis Parameters

The Axis Parameters window enables you to monitor and change the motion parameters for any axis on the controller.

The window is made up of a number of cells, separated into two banks, bank 1 at the top and bank 2 at the bottom:

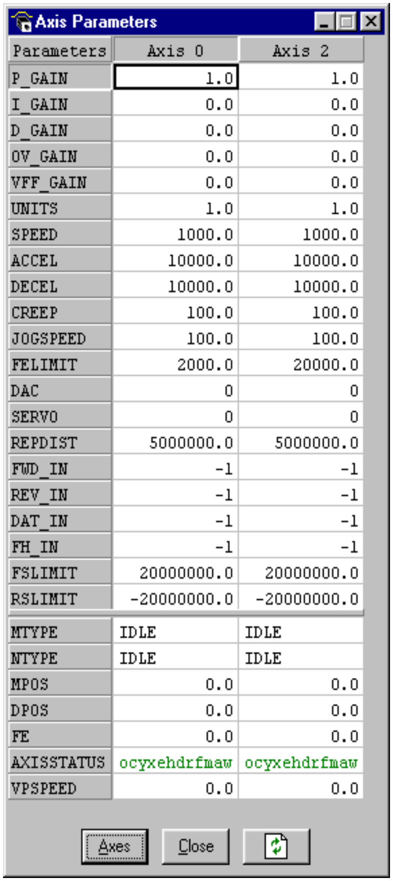
Bank 1. contains the values of parameters that may be changed by the user.

Bank 2. contains the values of parameters that cannot be changed by the user, as these values are set by the system software of the *Motion Coordinator* as it processes the Trio BASIC motion commands and monitors the status of the external inputs.

The black dividing bar that separates the two banks may be repositioned using the mouse to redistribute the space occupied by the different banks, for example to allow the user to shrink the window and view other windows whilst still watching the bank 2 information.

When there are more parameters in a bank that can be shown in the window a scroll bar will appear beside that bank so that the user can scroll up and down the parameter list to see the required values.

The user can select different parameters using the cursor keys or using the mouse. Multiple items may be selected by pressing the shift key and then using the cursor keys or the clicking the mouse to select a different cell, or by pressing the left mouse button in the start cell and the moving the mouse to select the last cell in the selection. Functions may be implemented in the future that work on a selection of multiples cells.



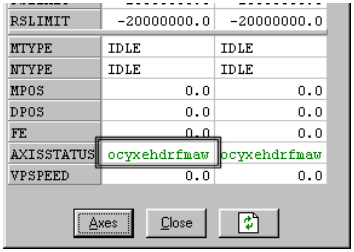
Parameters	Axis 0	Axis 2
P_GAIN	1.0	1.0
I_GAIN	0.0	0.0
D_GAIN	0.0	0.0
OV_GAIN	0.0	0.0
VFF_GAIN	0.0	0.0
UNITS	1.0	1.0
SPEED	1000.0	1000.0
ACCEL	10000.0	10000.0
DECEL	10000.0	10000.0
CREEP	100.0	100.0
JOGSPEED	100.0	100.0
FELIMIT	2000.0	20000.0
DAC	0	0
SERVO	0	0
REPDIST	5000000.0	5000000.0
FWD_IN	-1	-1
REV_IN	-1	-1
DAT_IN	-1	-1
FH_IN	-1	-1
FSLIMIT	20000000.0	20000000.0
RSLIMIT	-20000000.0	-20000000.0
MTYPE	IDLE	IDLE
NTYPE	IDLE	IDLE
MPOS	0.0	0.0
DPOS	0.0	0.0
FE	0.0	0.0
AXISSTATUS	ocyxehdrfmaw	ocyxehdrfmaw
VPSPEED	0.0	0.0

When the user changes the **UNITS** parameter for any axis, all the data for this axis is re-read as many of the parameters, such as the **SPEED**, **ACCEL**, **MPOS**, etc., are adjusted by this factor to be shown in user units.

In the *Motion* Perfect parameter screen the **AXISSTATUS** parameter is displayed as a series of characters, **ocyxehdrfmaw**.

These characters represent **AXISSTATUS** bits in order, as follows:-

char	status bit
w	Warning FE Range
a	Drive Comms Error
m	Remote Drive Error
f	Forward Limit
r	Reverse Limit
d	Datum Input
h	Feed Hold Input
e	Following Error
x	Forward Soft Limit
y	Reverse Soft Limit
c	Cancelling Move
o	Encoder Overcurrent



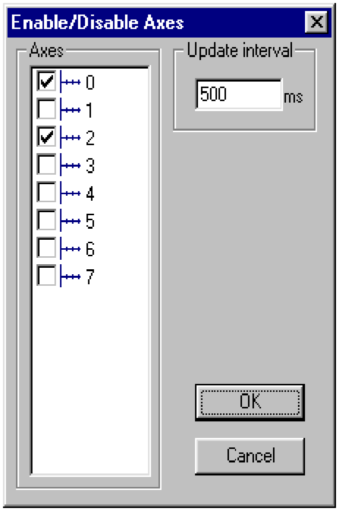
RSLIMIT	-20000000.0	-20000000.0
MTYPE	IDLE	IDLE
NTYPE	IDLE	IDLE
MPOS	0.0	0.0
DPOS	0.0	0.0
FE	0.0	0.0
AXISSTATUS	ocyxehdrfmaw	ocyxehdrfmaw
VPSPEED	0.0	0.0

Parameter Screen Options

Select Axes

This shows a dialog that allows the user to select the axes for which the data will be displayed.

The axes set by the last Create Startup, Jog Axes window or Axes Parameters window will be displayed by default.



Enable/Disable Axes

Axes
☑ 0
□ 1
☑ 2
□ 3
□ 4
□ 5
□ 6
□ 7

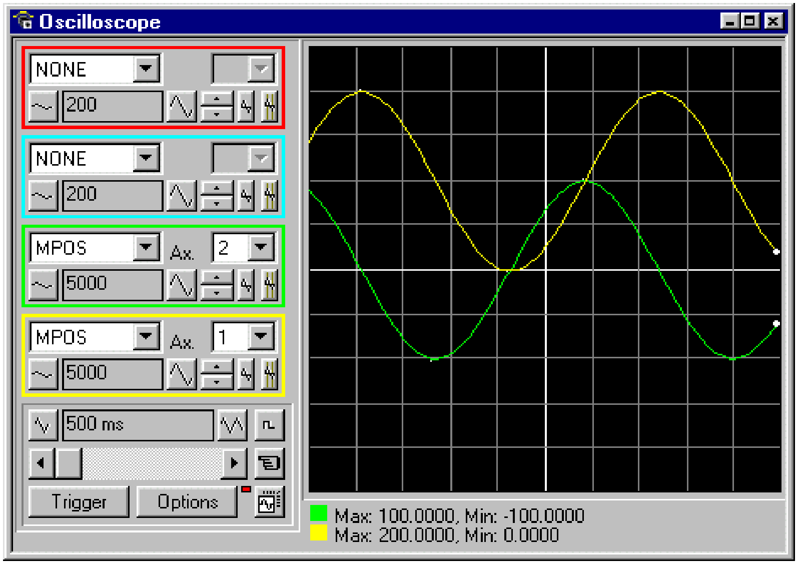
Update interval: 500 ms

Refresh Display

In order to minimise the load placed upon the controller communications, the parameters in the bank 1 section are only read when the screen is first displayed or the parameter is edited by the user. It is possible that if a parameter is changed in a user program then value displayed may be incorrect. The refresh button will force *Motion* Perfect to read the whole selection again.

Note *If there is any possibility that a program has changed any of the parameters then you should ensure that your refresh the display before making changes.*

Oscilloscope



NONE 200

NONE 200

MPOS Ax. 2 5000

MPOS Ax. 1 5000

500 ms

🟩 Max: 100.0000, Min: -100.0000
🟨 Max: 200.0000, Min: 0.0000

The software oscilloscope can be used to trace axis and motion parameters, aiding program development and machine commissioning.

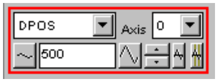
There are four channels, each capable of recording at up to 1000 samples/sec, with manual cycling or program linked triggering.

The controller records the data at the selected frequency, and then uploads the information to the oscilloscope to be displayed. If a larger time base value is used, the data is retrieved in sections, and the trace is seen to be plotted in sections across the display. Exactly when the controller starts to record the required data depends upon whether it is in manual or program trigger mode. In program mode, it starts to record data when it encounters a `TRIGGER` instruction in a program running on the controller. However, in manual mode it starts recording data immediately.

Controls

The oscilloscope controls are organised into four channel specific control blocks, followed by the oscilloscope general controls.

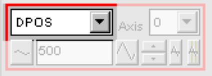
Oscilloscope Channel Controls



Each oscilloscope channel has the following channel specific controls organised in each of four 'channel control blocks' surrounded by a coloured border which indicates the colour of this channels trace on the display.

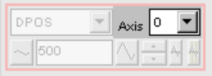
There are parameter list box / axis list box / vertical scale up-down buttons/ vertical offset scrollbar/ vertical offset reset button and cursor bars on-off button controls per scope channel.

Parameter



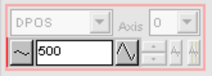
The parameters which the oscilloscope can record and display are selected using the pull-down list box in the upper left hand corner of each channel control block. Depending upon the parameter chosen, the next label switches between 'axis' or 'ch' (channel). This leads to the second pull-down list box which enables the user to select the required axis for a motion parameter, or channel for a digital input/output or analog input parameter. It is also possible to plot the points held in the controller table directly, by selecting the 'TABLE' parameter, followed by the number of a channel whose first/last points have been configured using the advanced options dialog. If the channel is not required then 'NONE' should be selected in the parameter list box.

Axis / Channel Number



A pull-down list box which enables the user to select the required axis for a motion parameter, or channel for a digital input/output or analog input parameter. The list box label switches between being blank if the oscilloscope channel is not in use, 'axis' if an axis parameter has been selected, or 'ch' if a channel parameter has been selected.

Vertical Scaling



The vertical scale (units per grid division on the display) are selected per channel, and these can be configured in either automatic or manual mode.

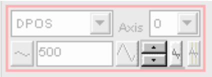
In automatic mode the oscilloscope calculates the most appropriate scale when it has finished running, prior to displaying the trace. Hence if the oscilloscope is running with continuous triggering, it will initially be unable to select a suitable vertical scale. It must be halted and re-started, or used in the manual scaling mode.

In manual mode the user selects the scale per grid division.

The vertical scale is changed by pressing the up/down scale buttons either side of the current scale text box (left hand side button decreases the scale, and the right hand side button increases the scale value.) To return to the automatic scaling mode, continue pressing the left hand side button (decreasing the scale value) until the word 'AUTO' appears in the current scale text box.

Channel Trace Vertical Offset

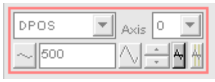
The vertical offset buttons are used to move a trace vertically on the display. This control is of particular use when two or more traces are identical, in which case they overlay each other and only the uppermost trace will be seen on the display.



The offset value remains for a channel until the vertical offset reset button is pressed, or the scrollbar is used to return the trace to its original position.

Vertical Offset Reset

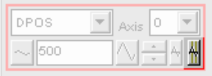
The vertical offset value applied using the vertical offset scroll bars can be cleared by pressing this vertical offset reset button.



The button latches on/off. When the button is latched ON then the vertical offset will automatically rescale each time the oscilloscope display is redrawn.

Cursor Bars

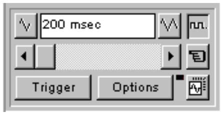
After the oscilloscope has finished running, and has displayed a trace, cursor bars can be enabled. These are displayed as two vertical bars, of the same colour as the channel trace, and initially located at the maximum and minimum trace location points. The values these represent are shown below the oscilloscope display, and again the text is of the same colour as the channel values represented.



The cursor bars are enabled/disabled by pressing the cursor button which toggles alternately displaying and removing the cursor bars. The bars can then be moved by positioning the mouse cursor over the required bar, holding down the left mouse button, and dragging the bar to the required position. The respective maximum or minimum value shown below the display is updated as the bar is dragged along with the value of the trace at the current bar position.

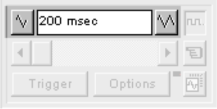
When the cursor bars are disabled, the maximum and minimum points are indicated by a single white pixel on the trace.

Oscilloscope General Controls



The oscilloscope general controls appear at the bottom left of the oscilloscope window. From here you can control such aspect as the time base, triggering modes and memory used for the captured data.

Time Base

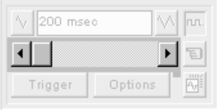


The required time base is selected using the up/down scale buttons either side of the current time base scale text box (left side button decreases the scale, and the right side button increases the scale value.) The value selected is the time per grid division on the display.

If the time base is greater than a predefined value, then the data is retrieved from the controller in sections (as opposed to retrieving a compete trace of data at one time.) These sections of data are plotted on the display as they are received, and the last point plotted is seen as a white spot.

After the oscilloscope has finished running and a trace has been displayed, the time base scale may be changed to view the trace with respect to different horizontal time scales. If the time base scale is reduced, a section of the trace can be viewed in greater detail, with access provided to the complete trace by moving the horizontal scrollbar.

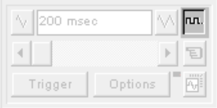
Horizontal scrollbar



Once the oscilloscope has finished running and displayed the trace of the recorded data, if the time base is changed to a faster value, only part of the trace is displayed. The remainder can be viewed by moving the thumb box on the horizontal scrollbar.

Additionally, if the oscilloscope is configured to record both motion parameters and plot table data, then the number of points plotted across the display can be determined by the motion parameter. If there are additional table points not visible, these can be brought into view by scrolling the table trace using the horizontal scrollbar. The motion parameter trace does not move.

One Shot / Continuous Trigger Mode



Button Raised = One Shot Trigger Mode:

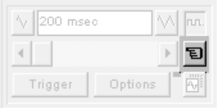
In one-shot mode, the oscilloscope runs until it has been triggered and one set of data recorded by the controller, retrieved and displayed.

Button Pressed = Continuous Trigger Mode:

In continuous mode the oscilloscope continues running and retrieving data from the controller each time it is re-triggered and new data is recorded. The oscilloscope continues to run until the trigger button is pressed for a second time.

Manual/Program Trigger Mode

The manual/program trigger mode button toggles between these two modes. When pressed, the oscilloscope is set to trigger in the program mode, and two program listings can be seen on the button. When raised, the oscilloscope is set to the manual trigger mode, and a pointing hand can be seen on the button.



Button Raised = Manual Trigger Mode:

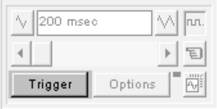
In manual mode, the controller is triggered, and starts to record data immediately the oscilloscope trigger button is pressed.

Button Depressed = Program Trigger Mode:

In program mode the oscilloscope starts running when the trigger button is pressed, but the controller does not start to record data until a TRIGGER instruction is executed by a program running on the controller. After the trigger instruction is executed by the program, and the controller has recorded the required data. The required data is retrieved by the oscilloscope and displayed.

The oscilloscope stops running if in one-shot mode, or it waits for the next trigger on the controller if in continuous mode

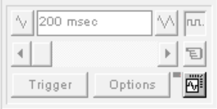
Trigger Button



When the trigger button is pressed the oscilloscope is enabled. If it is manual mode the controller immediately commences recording data. If it is in program mode then it waits until it encounters a trigger command in a running program.

After the trigger button has been pressed, the text on the button changes to ‘Halt’ whilst the oscilloscope is running. If the oscilloscope is in the one-shot mode, then after the data has been recorded and plotted on the display, the trigger button text returns to ‘Trigger’, indicating that the operation has been completed.The oscilloscope can be halted at any time when it is running, and the trigger button is displaying the ‘Halt’ text, by pressing this button.

Reset Oscilloscope Configuration



The current scope configuration (the state of all the controls) is saved when the scope window is closed, and retrieved when the scope window is next opened. This removes the need to re-set each individual control every time the scope window is opened.

The configuration reset button (located at the bottom right hand side of the scope control panel) can be pressed to reset the scope configuration, clearing all controls to their default values.

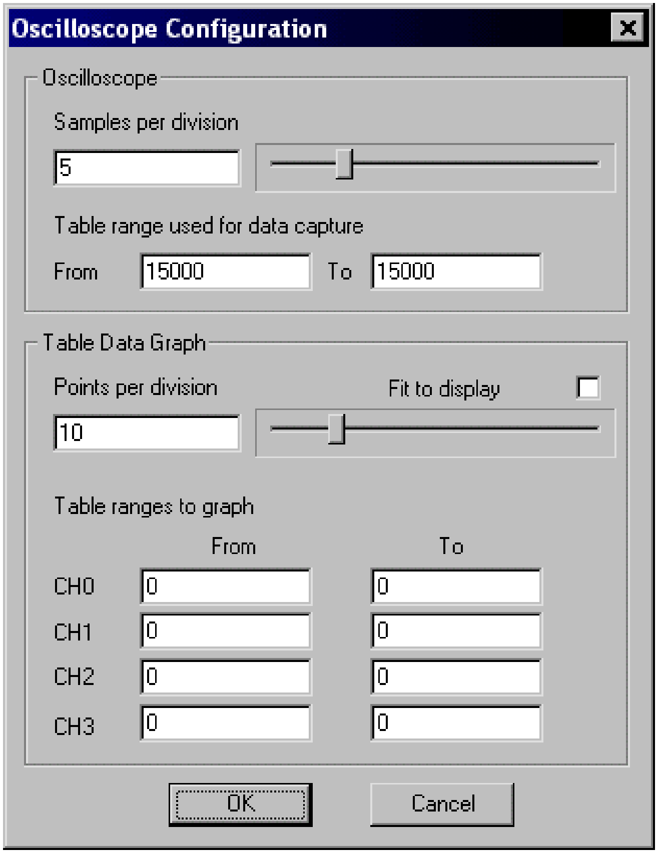
Status Indicator

The status indicator is located in between the options and configuration reset buttons. This lamp changes colour according to the current status of the scope, as follows:

Red	oscilloscope stopped.
Black	polling controller waiting for it to complete recording the required data.
Yellow	retrieving data from the controller.

Advanced Oscilloscope Configuration Options

When the options button is pressed the advanced oscilloscope configuration settings dialog is displayed, as shown below. Click the mouse button over the various controls to reveal further information.



Samples per division: 5

Table range used for data capture

From 15000 To 15000

Points per division: 10

Fit to display □

Table ranges to graph

	From	To
CH0	0	0
CH1	0	0
CH2	0	0
CH3	0	0

Samples per division

The oscilloscope defaults to recording five points per horizontal (time base) grid division. This value can be adjusted using the adjacent scrollbar.

To achieve the fastest possible sample rate it is necessary to reduce the number of samples per grid division to 1, and increase the time base scale to its fastest value (1 servo period per grid division).

It should be noted that the trace might not be plotted completely to the right hand side of the display, depending upon the time base scale and number of samples per grid division.

Oscilloscope Table Values

The controller records the required parameter data values in the controller as table data prior to uploading these values to the scope. By default, the lowest oscilloscope table value used is zero. However, if this conflicts with programs running on the controller which might also require this section of the table, then the lower table value can be reset.

The lower table value is adjusted by setting focus to this text box and typing in the new value. The upper oscilloscope table value is subsequently automatically updated (this value cannot be changed by the user), based on the number of channels in use and the number of samples per grid division. If an attempt is made to enter a lower table value which causes the upper table value to exceed the maximum permitted value on the controller, then the original value is used by the oscilloscope.

Table Data Graph

It is possible to plot controller table values directly, in which case the table limit text boxes enable the user to enter up to four sets of first/last table indices.

Parameter Checks

If analog inputs are being recorded, then the fastest oscilloscope resolution (sample rate) is the number of analog channels in msec (ie 2 analog inputs infers the fastest sample rate is 2msec). The resolution is calculated by dividing the time base scale value by the number of samples per grid division.

It is not possible to enter table channel vales in excess of the controllers maximum TABLE size, nor to enter a lower oscilloscope table value. Increasing the samples per grid division to a value which causes the upper oscilloscope table value to exceed the controller maximum table value is also not permitted.

If the number of samples per grid division is increased, and subsequently the time base scale is set to a faster value which causes an unobtainable resolution, the oscilloscope automatically resets the number of samples per grid division.

General Oscilloscope Information

Displaying Controller Table Points -

If the oscilloscope is configured for both table and motion parameters, then the number of points plotted across the display is determined by the time base (and samples per division). If the number of points to be plotted for the table parameter is greater than the number of points for the motion parameter, the additional table points are not displayed, but can be viewed by scrolling the table trace using the horizontal scrollbar. The motion parameter trace does not move.

Data Upload from the controller to the oscilloscope -

If the overall time base is greater than a predefined value, then the data is retrieved from the controller in blocks, hence the display can be seen to be updated in sections. The last point plotted in the current section is seen as a white spot.

If the oscilloscope is configured to record both motion parameters, and also to plot table data, then the table data is read back in one complete block, and then the motion parameters are read either continuously or in blocks (depending upon the time base).

Even if the oscilloscope is in continuous mode, the table data is not re-read, only the motion parameters are continuously read back from the controller.

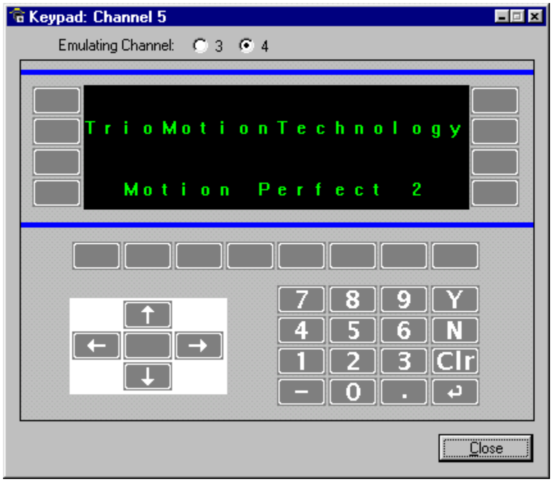
Enabling/Disabling of oscilloscope controls -

Whilst the oscilloscope is running all the oscilloscope controls except the trigger button are disabled. Hence, if it is necessary to change the time base or vertical scale, the oscilloscope must be halted and re-started.

Display accuracy -

The controller records the parameter values at the required sample rate in the table, and then passes the information to the oscilloscope. Hence the trace displayed is accurate with respect to the selected time base. However, there is a delay between when the data is recorded by the controller and when it is displayed on the oscilloscope due to the time taken to upload the data via the serial link.

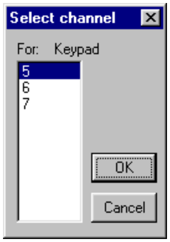
Keypad Emulation



Keypad: Channel 5

Emulating Channel: ○ 3 ● 4

```
Trio Motion Technology
  Motion  Perfect  2
```



Select channel

For: Keypad

5
6
7

The keypad requires one of the user communications channels, and so you will be prompted for the channel to use.

If the specified channel is already in use, either by another keypad or a terminal window, the window will not open. Once a channel has been reserved then the keypad will be shown.

In the Trio BASIC program the channel definition for the commands that are associated with the Keypad must be changed from 3 (or 4) to the channel that corresponds with the channel selected for the emulation. We recommend that the channel assignment be made through a variable, so when time comes to run the program on the real machine, only one program change will be required.

example:

```
kpd=5
PRINT #kpd, "Press any key.."
```

Emulating Channel

The normal operation of the keypad emulation returns the characters as if they were read from channel #3 with the **DEFKEY** translation. Alternatively, the *Motion Coordinator* can read the characters returned directly from the Keypad using channel 4. If the emulate #4 codes is selected then the keypad emulation will return the raw characters.

Note: It is only possible to emulate the default **DEFKEY** table.

Key Functions

menu keys This is a keypad menu key. Normally it is associated with a message on the display. This button can only be pressed by clicking the mouse over it.

function keys 1-8 This is the keypad function key 1. Normally it has an associated user label. This button can be pressed by clicking the mouse over it or using the ‘1’ - '8' keys in the QWERTY area of the PC keyboard.

number keys This is a keypad number key. It can be pressed by clinking the mouse over it or using the corresponding number in the numerical keypad of your PC keyboard.

Y/N keys This is the keypad ‘Y’ and 'N' keys. This is usually used to respond YES or NO to some question on the display. It can be pressed by clicking the mouse over it or using the ‘Y’/'N' keys in the QWERTY area of the PC keyboard.

CLR key This is the keypad ‘CLR’ key. This is usually used to perform some form of CANCEL operation. It can be pressed by clicking the mouse over it or using the ‘ESC’ in the QWERTY area of the PC keyboard.

Return key This is the keypad Return key. This is usually used to perform some form of ACCEPT operation. It can be pressed by clicking the mouse over it or using the ‘Enter’ in the QWERTY area or numerical keypad of the PC keyboard.

- key This is the keypad ‘-’ key. This is usually used for entering negative numbers. It can be pressed by clicking the mouse over it or using the ‘-’ in the QWERTY area or numerical keypad of the PC keyboard.

. key This is the keypad ‘.’ key. This is usually used for entering fractional numbers. It can be pressed by clicking the mouse over it or using the ‘.’ in the QWERTY area or numerical keypad of the PC keyboard.

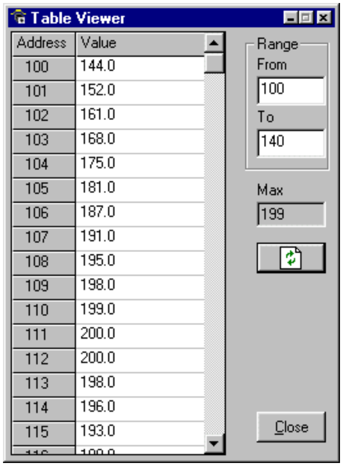
arrow keys This is the keypad up arrow key. This is usually used to select between options on the display. It can be pressed by clicking the mouse over it or using the appropriate arrow key of the PC keyboard.

centre button This is the keypad centre key.
It can only be pressed by clicking the mouse over it.

Table / VR Editor

The Table and VR Editor tools are very similar. You are presented with a spreadsheet style interface to view and modify a range of values in memory.

To modify a value, click on the existing value with the mouse and type in the new value and press return. The change will be immediate and can be made whilst programs are running.



Address	Value
100	144.0
101	152.0
102	161.0
103	168.0
104	175.0
105	181.0
106	187.0
107	191.0
108	195.0
109	198.0
110	199.0
111	200.0
112	200.0
113	198.0
114	196.0
115	193.0

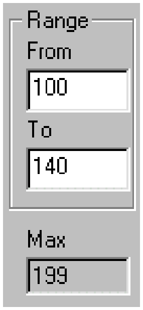
Range
From 100
To 140
Max 199

Options

Range

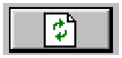
In both tools you have the option to set the start and end of the range to view. In the Table view tool the max value displays the highest value you can read (this is the system parameter TSIZE).

If the range of values is larger than the dialog box can display, then the list will have a scrollbar to enable all the values to be seen.



Range
From 100
To 140
Max 199

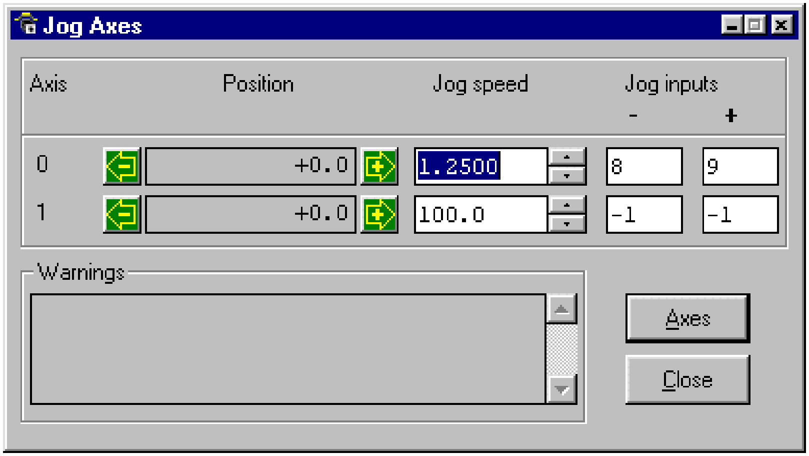
Refresh Button



This screen does not update automatically, so if a Table or VR is changed by the program you will not see the new value until you refresh the display.

Jog Axes

This window allows the user to move the axes on the *Motion Coordinator*.



Axis	Position	Jog speed	Jog inputs -	Jog inputs +
0	+0.0	1.2500	8	9
1	+0.0	100.0	-1	-1

This window takes advantage of the bi-directional I/O channels on the *Motion Coordinator* to set the jog inputs. The forward, reverse and fast jog inputs are identified by writing to the corresponding axis parameters and are expected to be connected to NC switches. This means that when the input is on (+24V applied) then the corresponding jog function is DISABLED and when the input is off (0V) then the jog function is ENABLED.

The jog functions implemented here disable the fast jog function, which means that the speed at which the jog will be performed is set by the `JOGSPEED` axis parameter. What is more this window limits the jog speed to the range 0..demand speed, where the demand speed is given by the `SPEED` axis parameter.

Before allowing a jog to be initiated, the jog window checks that all the data set in the jog window and on the *Motion Coordinator* is valid for a jog to be performed.

Jog Reverse

This button will initiate a reverse jog. In order to do this, the following check sequence is performed:

- If this is a SERVO or RESOLVER axis and the servo is off then set the warning message
- If this axis has a daughter board and the WatchDog is off then set the warning message
- If the jog speed is 0 the set the warning message
- If the acceleration rate on this axis is 0 then set the warning message
- If the deceleration rate on this axis is 0 then set the warning message
- If the reverse jog input is out of range then set the warning message
- If there is already a move being performed on this axis that is not a jog move then set the warning message

If there were no warnings set, then the message “Reverse jog set on axis?” is set in the warnings window, the **FAST_JOG** input is invalidated for this axis, the **CREEP** is set to the value given in the jog speed control and finally the **JOG_REV** output is turned off, thus enabling the reverse jog function.

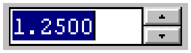
Jog Forward

This button will initiate a forward jog. In order to do this, the following check sequence is performed:

- If this is a SERVO or RESOLVER axis and the servo is off then set the warning message
- If this axis has a daughter board and the WatchDog is off then set the warning message
- If the jog speed is 0 the set the warning message
- If the acceleration rate on this axis is 0 then set the warning message
- If the deceleration rate on this axis is 0 then set the warning message
- If the reverse jog input is out of range then set the warning message
- If there is already a move being performed on this axis that is not a jog move then set the warning message

If there were no warnings set, then the message “Forward jog set on axis?” is set in the warnings window, the **FAST_JOG** input is invalidated for this axis, the **CREEP** is set to the value given in the jog speed control, and finally the **JOG_FWD** output is turned off, thus enabling the forward jog function.

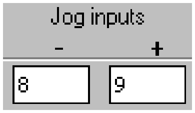
Jog Speed



1.2500

This is the speed at which the jog will be performed. This window limits this value to the range from zero to the demand speed for this axis, where the demand speed is given by the **SPEED** axis parameter. This value can be changed by writing directly to this control or using the jog speed control. The scroll bar changes the jog speed up or down in increments of 1 unit per second

Jog Inputs



Jog inputs	
-	+
8	9

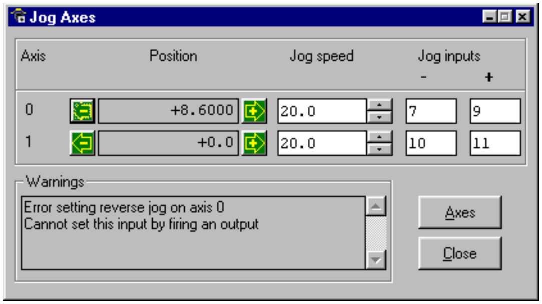
These are the inputs which will be associated with the forward / reverse jog functions.

They must be in the range 8 to the total number of inputs in the system as the input channels 0 to 7 are not bi-directional and so the state of the input cannot be set by the corresponding output.

The input is expected to be ON for the jog function to be disabled and OFF for the reverse jog to be enabled. In order to respect this, when this is set to a valid input number, the corresponding output is set ON and then the corresponding **REV_JOG** axis parameter is set.

Warnings

This shows the status of the last jog request. For example, the screen below shows axis 0 with IO channel 7 selected. This is an Input-only channel and therefore cannot be used in the jog screen.



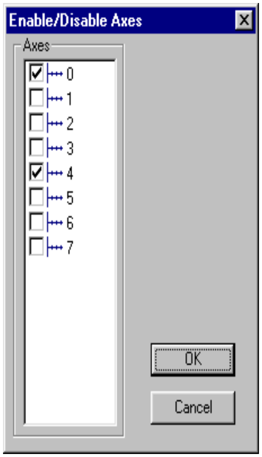
Jog Axes

Axis	Position	Jog speed	Jog inputs -	Jog inputs +
0	+8.6000	20.0	7	9
1	+0.0	20.0	10	11

Warnings

Error setting reverse jog on axis 0
Cannot set this input by firing an output

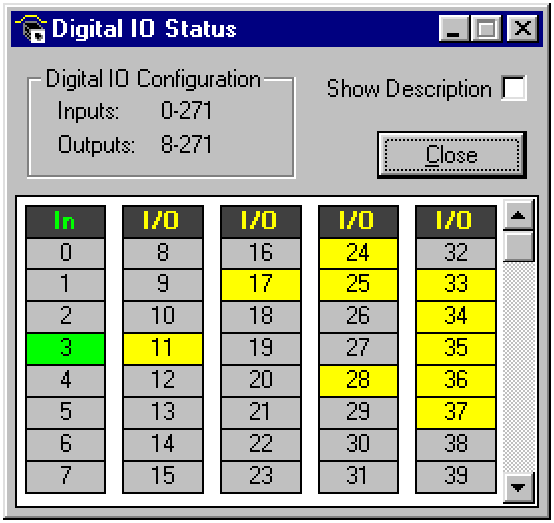
Axes



This displays an axis selector box which enables the user to select the axis to include in the jog axes display. By default, the physical axes fitted to the controller will be displayed.

Digital IO Status

This window allows the user to view the status of all the IO channels and toggle the status of the output channels. It also optionally allows the user to enter a description for each I/O line.



Digital IO Configuration
Inputs: 0-271
Outputs: 8-271

Show Description ☐

In	I/O	I/O	I/O	I/O
0	8	16	24	32
1	9	17	25	33
2	10	18	26	34
3	11	19	27	35
4	12	20	28	36
5	13	21	29	37
6	14	22	30	38
7	15	23	31	39

Digital Inputs

This shows the total number of input channels on the *Motion Coordinator*.

Digital Outputs

This shows the total number of output channels on the *Motion Coordinator*.

IO Mimic

Input Bank 0-7

This first bank of 8 LEDs shows the status of the dedicated input channels. If an LED is green then the corresponding input is ON. If an LED is white then the corresponding input is OFF.

Input/Output Banks

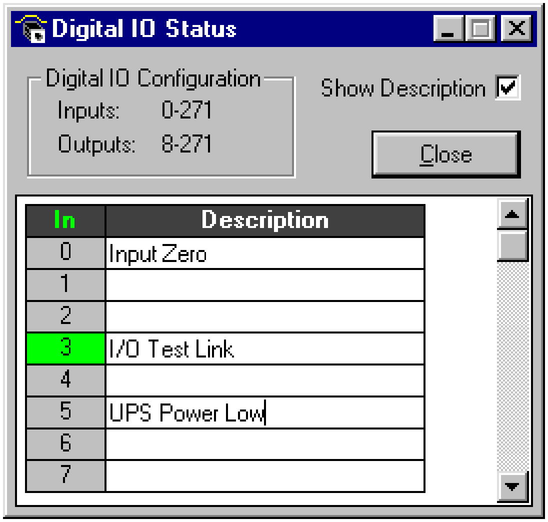
These banks of LEDs show the status the bi-directional IO channels. If an LED is yellow then the corresponding input is ON. If an LED is white then the corresponding input is OFF. Under normal conditions the input status mimics the output status, except:

5) If this input is connected to an external 24V then it may be on without the corresponding output being on.

6) If the output chip detects an overcurrent situation, then the output chip will shut down and so the outputs will not be driven, even though they may be turned on.

If the LED is clicked with the mouse the status corresponding output channel is toggled, i.e. if the LED is white then the output will be turned on, if the LED is yellow then the output channel will be turned off.

Checking the **Show Description** check box will toggle between dsecriptions on, and descriptions off. Descriptions are stored in the project file



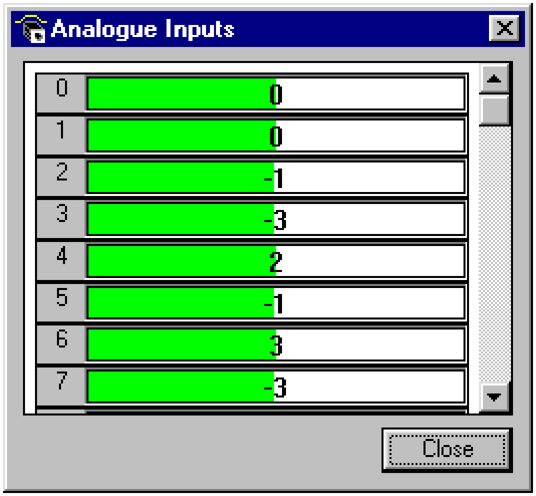
Digital IO Configuration
Inputs: 0-271
Outputs: 8-271

Show Description ☑

In	Description
0	Input Zero
1	
2	
3	I/O Test Link
4	
5	UPS Power Low
6	
7	

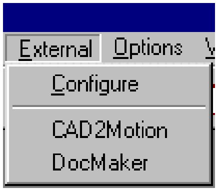
Analogue Input Viewer

The analogue input viewer is only available if the system has analogue inputs. It displays the input values of all analogue inputs in the system using a bar-graph with numeric display. All inputs have the range -2048 to 2047.



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

Linking to External Tools

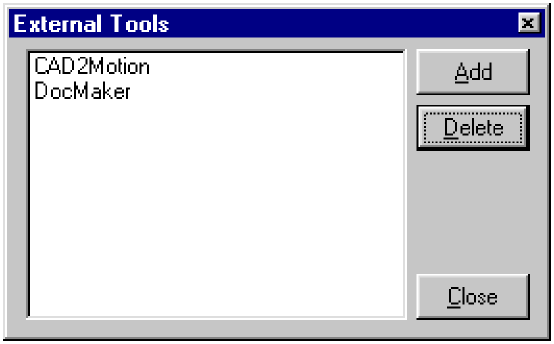


The EXTERNAL menu in *Motion* Perfect allows you to run other programs directly from the main *Motion* Perfect menu. In our example shown here, the menu has been configured to launch two other Trio applications, CAD2Motion and DocMaker. Further information on these applications is given at the end of this chapter.

Note: ***Cad2Motion and DocMaker are available to download from the Trio Website at*** `www.triomotion.com.`

Configuring Items on the External menu

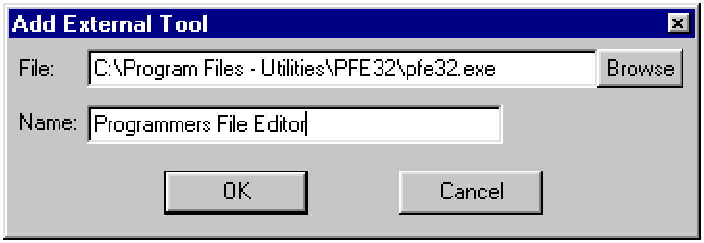
Clicking on the Configure item will bring up a list of all installed applications and from here we can add or delete items from this list.



CAD2Motion
DocMaker

Adding a new programs to the menu

Clicking on the Add button will open the following dialog:

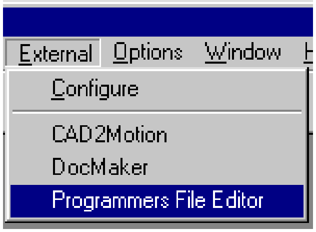


File: C:\Program Files - Utilities\PFE32\pfe32.exe

Name: Programmers File Editor

You can either directly enter the path and program file name in the "File" box, or use the "Browse" option to open up a standard windows file selector box which you can use to locate the file on your computer.

Once you have selected the file, it will automatically appear in the External menu every time you run *Motion* Perfect 2.

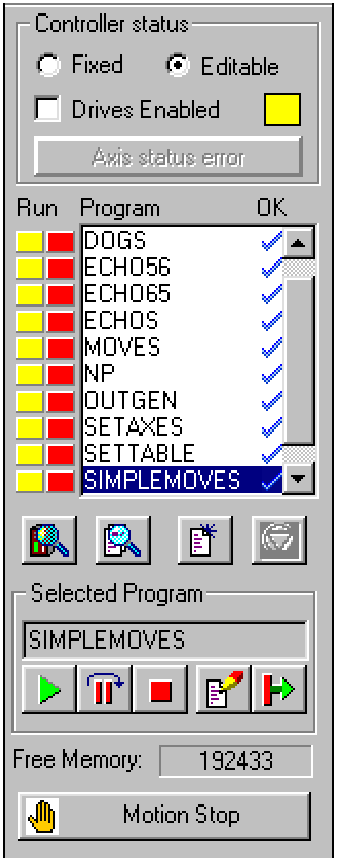


Removing program items from the menu

To delete a program from the External menu, you simply need to click on the program name in the list and press the Delete button.

Note: ***This simply removes the program from the menu. it does NOT affect the original program on disk!***

Control Panel



Controller status

○ Fixed ● Editable

□ Drives Enabled 🟨

Run	Program	OK
🟨 🟥	DOGS	✓
🟨 🟥	ECHO56	✓
🟨 🟥	ECHO65	✓
🟨 🟥	ECHOS	✓
🟨 🟥	MOVES	✓
🟨 🟥	NP	✓
🟨 🟥	OUTGEN	✓
🟨 🟥	SETAXES	✓
🟨 🟥	SETTABLE	✓
🟨 🟥	SIMPLEMOVES	✓

Selected Program: SIMPLEMOVES

Free Memory: 192433

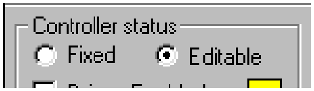
The control panel appears on the left hand side of the main *Motion* Perfect window.

It provides direct links to many of the frequently used operations within *Motion* Perfect, in particular the file and directory functions.

Please Note: Certain Control Panel Features behave differently on controller without a battery backup. The differences are described later in this section.

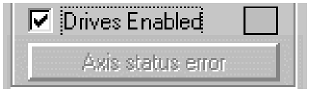
Control Panel Features

Fixed/Editable radio buttons

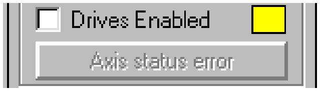

Controller status
○ Fixed ● Editable

When the project is “fixed”, the programs are copied to the Flash EPROM on the *Motion Coordinator*, the *Motion Coordinator* is set to run from EPROM and the programs cannot be modified by *Motion* Perfect. Usually this is done when the machine programs are completed. The Flash EPROM provides a reliable permanent storage for the programs.

Drives radio button


☑ Drives Enabled □

Drives Enabled

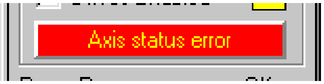

□ Drives Enabled 🟨

Drives Disabled

This radio button toggles the state of the enable (watchdog) relay on the controller, going between drives disabled (watchdog off) and drives enabled (watchdog on).

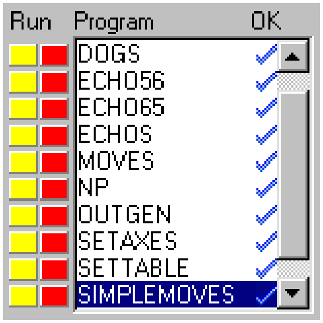
The LED mimic next to this control shows the status of the error LED on the *Motion Coordinator*. If it is yellow then the drives are disabled, if it is grey the drives are enabled and if it is flashing then there has been a motion error on at least one axis of the controller.

Axis Status Error



This will normally be greyed out unless a motion error occurs on the controller.
When an error does occur you can use this button to clear the error condition.

Program directory



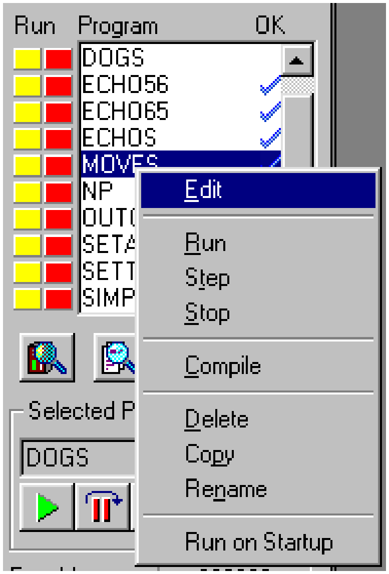
This is a scrollable list of the programs on the controller. The list shows the program name followed by two optional indicators. The first is a number which specifies which process that program is running on. If it is not running then this space is blank. The second indicator shows the status of the program. If it has a tick then the program has been compiled successfully and is ready to be run. If it has a cross then there was an error during the compilation of the program and it cannot be run.

If it is blank then it has not been compiled.

If a program name is clicked then it will become the selected program if there are no programs running. If there are programs running the select will be ignored.

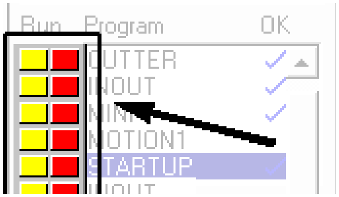
If a program name is double clicked then it will be opened for editing assuming that there are no programs running. If there are programs running then the edit will be ignored.

The names of programs which are currently running are displayed in italics.



Right clicking on an entry in the program list causes a pop-up menu to appear allowing easy access to operations commonly performed on programs. The right click operation highlights the entry in the program list under the cursor whilst the pop-up menu is visible. It reverts to the program which is currently selected on the controller when the menu is closed.

Run buttons

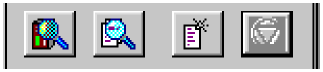


The run buttons provide short cut keys for running, stopping and single stepping programs. They can be in one of three states, red, green or yellow.

RED	Click on the red button to start the corresponding program running. The button will turn Green. *
GREEN	Click on the green button to stop the corresponding program. The button will turn red. *
YELLOW	Click on the yellow button to single step through the program.

* If the program goes into trace mode, through the use of a trace button, the selected program step button, the debug option of the program menu, the debug button on the tool bar or a TRON/STEPLINE command in a program or the terminal window, then the red/green run button will turn yellow. If the button is clicked when it is yellow then the program will be stepped one line.

General Options



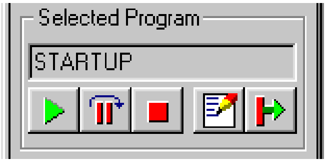
Show Controller Configuration

Show a full directory of all programs in memory

Create a New Program. Same as the Program menu item.

HALT - Stop all programs which are running

Selected Program



The text box displays the currently selected program and the buttons below, the operations which can be performed on that program.

From left to right they are:
Run, Step, Stop, Edit and Power Up Mode

Free Memory

Shows the total free memory available on the controller

Motion Stop

Cancel moves on all axes and disable the watchdog relay.

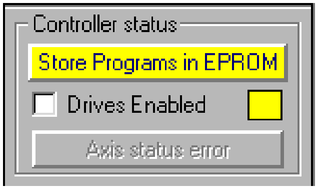
Note *MOTION STOP is a software function. It is not a substitute for a hardware E_Stop circuit and should not be used as an emergency stop.*

Control Panel Variations for Motion Coordinators without battery backup

On those controllers without a battery backup such as the MC202, the fixed / editable radio buttons are replace by a single button labeled "Store Programs into Eprom".

As the controller does not feature a battery, it is essential that you store your programs into the eprom to avoid loss of data.

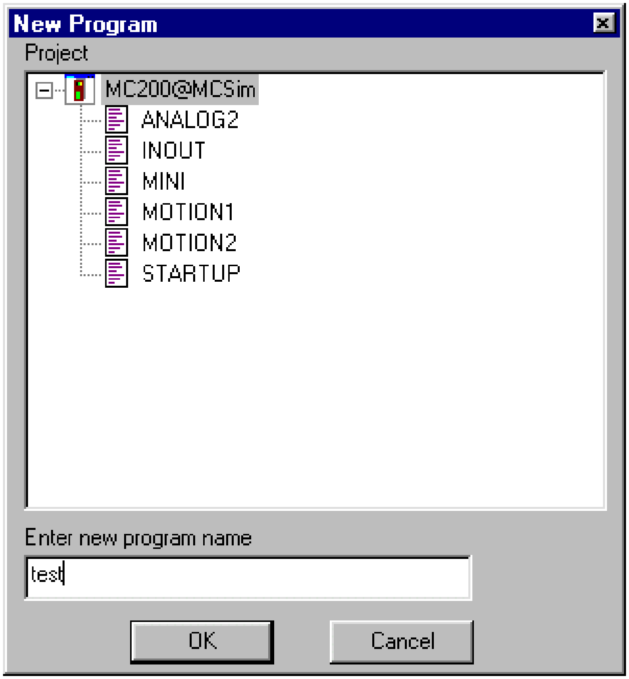
If the programs in memory have been edited, the button will be highlighted to remind you to fix into eprom before exiting from the program



Creating and Running a program

In order to create a new program on the controller, you must first have an active project. If you have already connected to the controller then you can use the default project which was created at this time.

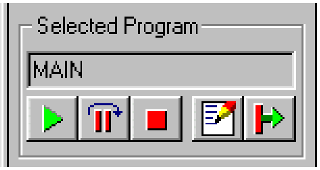
You will be presented with a program selector dialog and prompted to enter a name for your program file. It is a good idea to make this name representative of the task performed by the program, for example “mmi”, “motion”, “logic” or something similar. In the following example, we will add a program called “test” to the current project.



Enter new program name
test

Once you have created a new program it will be added to both the controller and the *Motion* Perfect project file. You can now edit the file in *Motion* Perfect Editor.

The *Motion* Perfect Editor



Program Section of the Control Panel

You can start the Editor from the main Program Menu, the Edit button in the program section of the control panel or by right clicking in an entry in the control panel program list and selecting **Edit** from the pop-up menu.

If you launch the editor from the control panel it will start immediately. From the program menu you will first be prompted with a program selector dialog to confirm the file you wish to edit.

The *Motion* Perfect Editor is designed to operate in a similar manner to any simple text editor found on a PC. Standard operations such a block editing functions, text search and replace and printing are all supported and conform to the standard Windows shortcut keys. In addition it provides TrioBASIC syntax highlighting, program formatting and program debugging facilities.

Editor Options

Options for the editor are controlled by the **Editor Options** dialog. This can be opened by selecting **Options/Editor** from *Motion* Perfect's main menu. The dialog allows to user to change the fonts used for screen display and printing, the colours used for syntax and line highlighting and the spacings used when automatically formatting a program.

Screen Font

This is the font used by *Motion* Perfect to display text in the editor window on the screen. The font is restricted to fixed pitch fonts only.

Printer Font

This is the font used by *Motion* Perfect to print program listings. The font is restricted to fixed pitch fonts only.

Colours

Colours can be specified for the following:

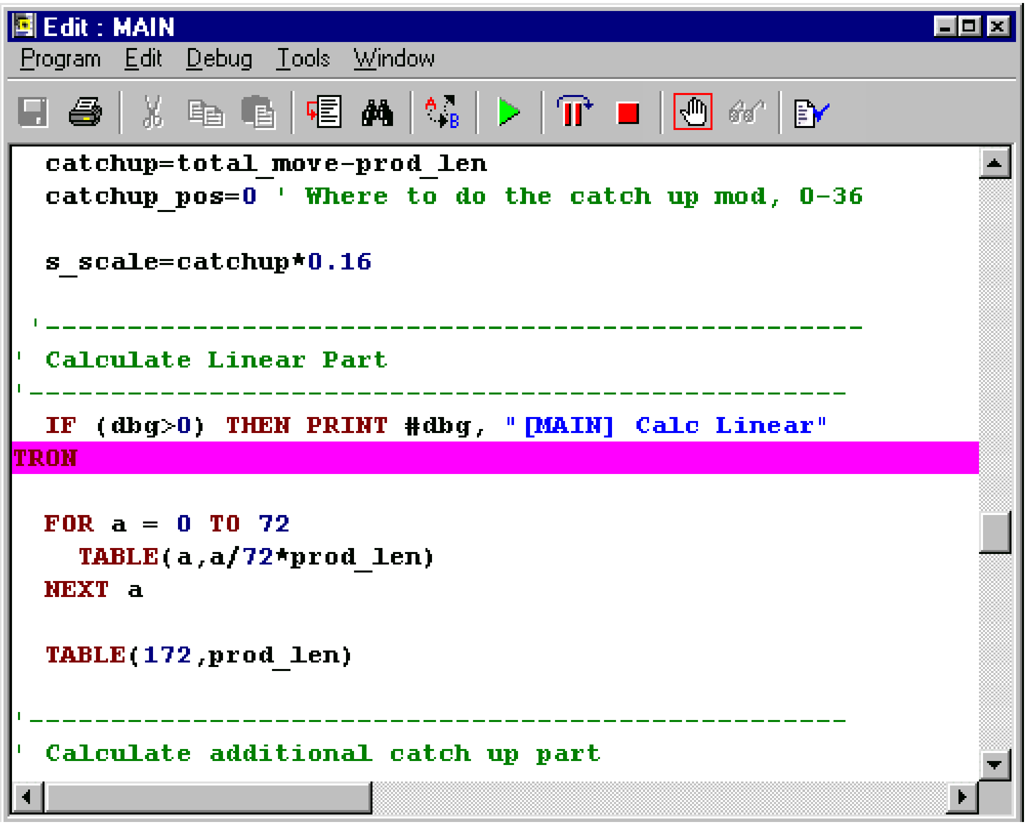
- Normal Text - Text which is not highlighted using systax highlighting.
- Screen Background.
- Current Line (background) - the current line during debugging.
- Break Line (background) - a line containing a break (TRON) command.
- Current Line Break (background) - a line containing a break command which is also the current line.
- Error Line (background) - The first line containing a compilation error.
- BASIC key word - A key word in the TrioBASIC language, usually a command or some type of system variable.

- Comment Text
- Constant Text - text making up a constant value (number).
- Strings
- Label Defenition - where a program label is defined.
- Label Reference - where a jump or branch (GOTO, GOSUB etc.) in program execution is required. The jump or branch changes the execution point to the place where the label is defined.

Format

The format options affect text entry and the automatic reformatting preformed by *Motion* Perfect.
For automatic reformatting the code start column and the tab width are specifiable. As label definitions always start in column 0, the code start column can be used to indent all lines containing code thus making label definitions clearer.

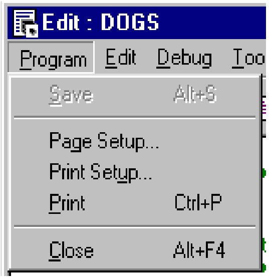
The when **Auto tab on enter** check box is checked pressing the "enter" key will automatically indent the next (new) line to the same position as the current one.



```
  catchup=total_move-prod_len
  catchup_pos=0 ' Where to do the catch up mod, 0-36

  s_scale=catchup*0.16

 '---------------------------------------------------
' Calculate Linear Part
'---------------------------------------------------
  IF (dbg>0) THEN PRINT #dbg, "[MAIN] Calc Linear"
TRON

  FOR a = 0 TO 72
    TABLE(a,a/72*prod_len)
  NEXT a

  TABLE(172,prod_len)

'---------------------------------------------------
' Calculate additional catch up part
```

Editor Menus

Program



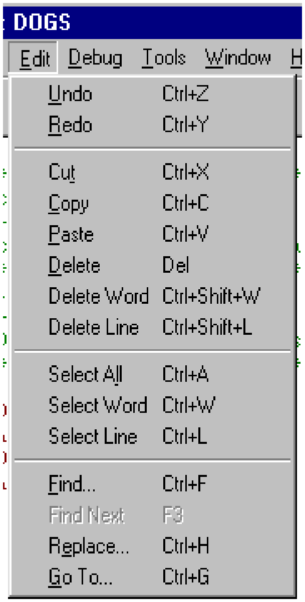
Save

Normally the program is only saved to disk when the editor is closed or a program is run, however if you have modified the program the Save Button will be available and will force *Motion* Perfect to save the file immediately.

Printing

Use **Page Setup** to set the page mrrgins, **Print Setup** to configure you printer settings and the **Print** option to send the program to the printer.

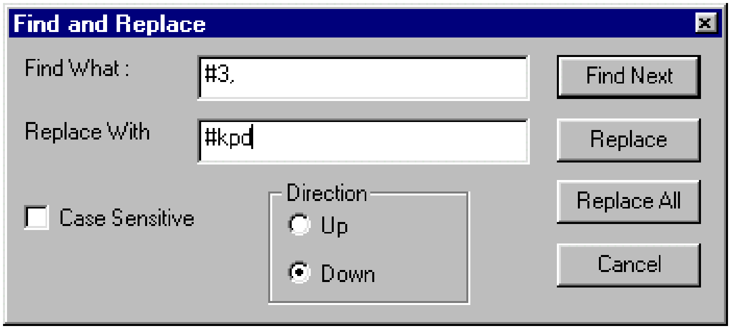
Edit



The edit menu functions are similar to many other text editors and provide the standard block cut/ copy/paste operations as well as a simple text find/ replace, and various select and delete functions.

Find/Replace

The options for the find & replace dialogs are very similar and feature many of the same options



Find What : #3,

Replace With #kpd

☐ Case Sensitive

Direction
○ Up
● Down

You should enter the text to search for in the "Find What" box, and if using find and replace, the text to replace it with in the "Replace With" box.

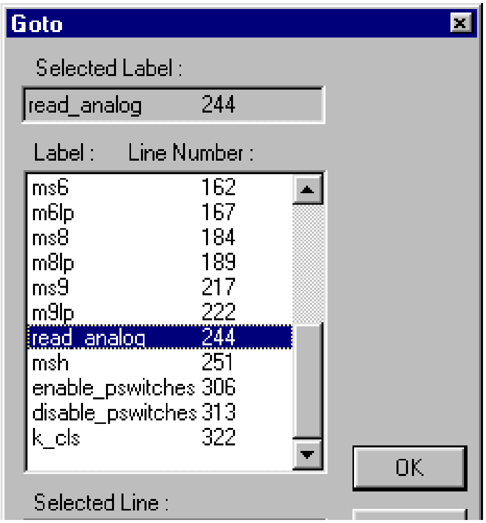
Normally the "Case Sensitive" search option is not selected, You should only use this option if you have an exact pattern to match, generally the default option is best.

GOTO

The Goto option will bring up the following dialog:

A list of labels defined in the program is displayed. You can either select a label from here, or enter a line number directly in the "Selected Line" field.

Press OK and the cursor will jump directly to the beginning of the selected line.



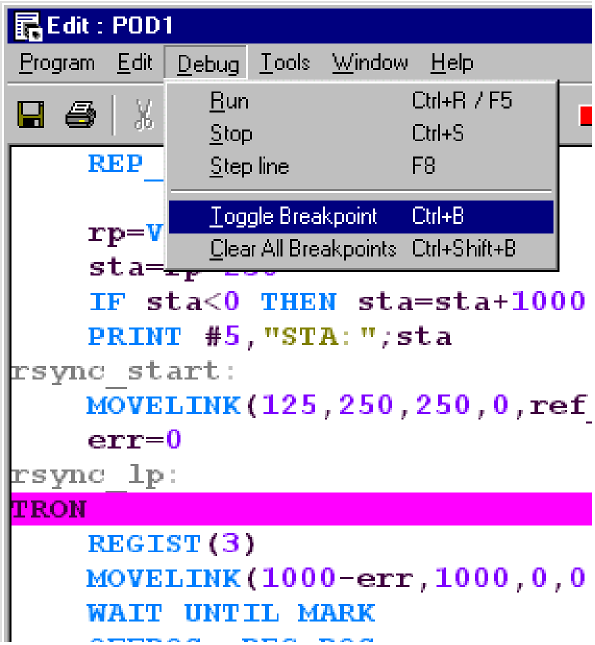
Selected Label : read_analog 244

Label :	Line Number :
ms6	162
m6lp	167
ms8	184
m8lp	189
ms9	217
m9lp	222
read_analog	244
msh	251
enable_pswitches	306
disable_pswitches	313
k_cls	322

Selected Line :

Program Debugger

The *Motion* Perfect debugger allows you to run a program directly from the editor window in a special 'trace mode, executing one line at a time (known as stepping) whilst viewing the line in the window. It is also possible to set breakpoints in the program, and run it at normal speed until it reaches the breakpoint where it will stop, and this line of code will be highlighted in the debug window.



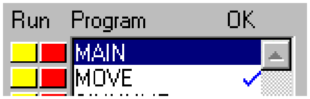
```
    REP_
    rp=V
    sta=
    IF sta<0 THEN sta=sta+1000
    PRINT #5,"STA:";sta
rsync_start:
    MOVELINK(125,250,250,0,ref_
    err=0
rsync_lp:
TRON
    REGIST(3)
    MOVELINK(1000-err,1000,0,0
    WAIT UNTIL MARK
```

When programs are running on the controller, any open editor windows will automatically switch to Debug Mode and will become read-only. Hence, breakpoints are set in the edit window, and the code viewed in the same window in debug mode when the program is running.

Stepping Through a program

To commence stepping a program:

Use the mouse to press the yellow button alongside the required program name in the list box on the control panel



if the required program is currently selected, press the 'Step' button on the control panel() or use the menu item 'Debug-Step line'

The currently executing line of code is indicated in the debug window by highlighting it with a green background, and a breakpoint is highlighted with a red background.

To continue stepping the program, repeatedly press the yellow button alongside the program name in the list box on the control panel, or press the 'Step' button or the 'F8' functional key if the program required is currently selected on the *Motion Coordinator*.

Breakpoints

Breakpoints are special place markers in the code which allow a particular section (or sections) of the program to be identified when debugging the code. If a breakpoint is inserted, the program will pause at that point and return control to *Motion* Perfect where the controller may be interrogated or the program run in step mode as described above.

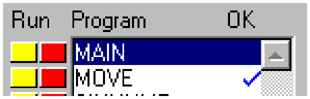
To insert a breakpoint, first position the text cursor on the line at which you want the break to occur, then use either Ctrl-B or the menu item to insert the breakpoint.

The Trio BASIC instruction TRON is used to mark a breakpoint and TROFF to terminate a 'traced' block.

Note: ***It is not possible to add or remove breakpoints whilst any programs are running.***

Running to a breakpoint

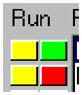
A program can be run to the next break point by:



- using the mouse to press the red button alongside the program name in the list box on the control panel.
- if it is the currently selected program on the *Motion Coordinator*, you can pressing the 'Run' button (▶) on the control panel/editor tool bar, or by using the keyboard <F5> function key.
- by selecting the 'Debug->Run' menu option

Stopping a Program

If it is necessary to stop the program running before it reaches the breakpoint then:



- press the green button alongside the program name (running on the required process) in the list box on the control panel.
- press the stop button (■)on the control panel if the program is currently selected (this will stop all running copies of the program)
- use the 'Debug-Stop' menu option.

Alternatively all programs can be stopped by pressing the 'Halt' button on the control panel, or selecting the 'Program' 'Halt all programs' menu option, or using the <Ctrl><F> key combination.

Switching a running program into trace mode

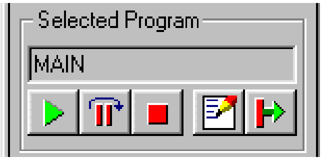
A running program can enter trace (stepping) mode by pressing the yellow button alongside the required program name in the list box on the control panel, or the 'Step' button if the required program is currently selected on the *Motion Coordinator*.

Running Programs

You can start/stop programs running in one of four ways:

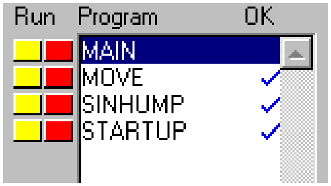
From the control panel

If the program is currently selected (highlighted in the control panel), you can press the green start arrow in the "selected program" box.

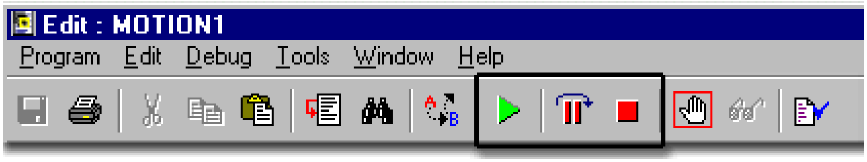


From the program list

Pressing the red button to the left of the program name in the list will start it running, the button will change to green and it will then function as a stop button for the same task.



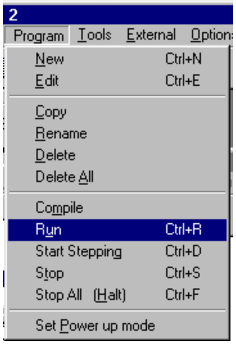
From the editor toolbar



If you have an editor or debug window open for the program you can use either the Debug menu or toolbar buttons to start the program running

From the Program Menu

The program menu provides us with a slightly different option when running the program as we are presented with a program selector box which includes an option to choose which task we want to run the program on

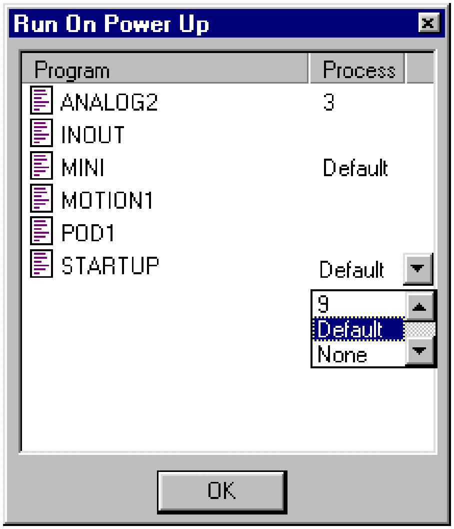


Making programs run automatically

Set Powerup Mode

It is possible to make the programs on the controller run automatically when the system first starts up. From the **Program** Menu, select “**Set Powerup Mode**” to open the following dialog.

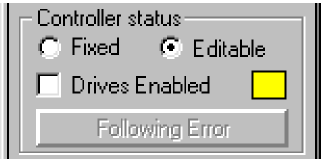
Click on the program you want to auto run and a small drop-down list will appear to the right of the window. If you are happy to let the controller allocate which task to run on then you should choose “default” as the process number, otherwise you can specify the task explicitly in the box.


Run On Power Up

Program	Process
ANALOG2	3
INOUT	
MINI	Default
MOTION1	
POD1	
STARTUP	Default

Storing Programs in the Flash EPROM

This is accomplished by selecting the “Fixed” option in the controller status section of the control panel, or the “Fix Program Into EPROM” option from the controller menu.



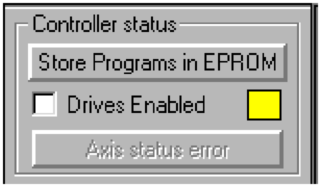
When the controller is fixed into eprom, the programs actually still run from RAM. The information stored is copied into RAM when the controller is first started, therefore if the controller has been switched off for an extended period, or there is any corruption of the RAM, it will be refreshed with a correct copy of the programs.

When the controller is set to fixed you will not be able to edit any programs. In order to make changes you must select “Editable” from the control panel or “Enable Editing” from the controller menu.

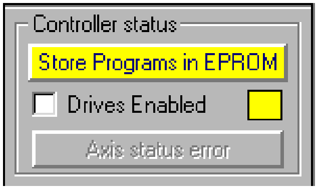
Variations for controllers without battery backup

On those controllers without battery backup, such as the MC202, it is essential that you store your programs into the eprom to avoid loss of data.

The control panel the fixed / editable radio buttons are replaced by a single button labelled "Store Programs into Eprom".

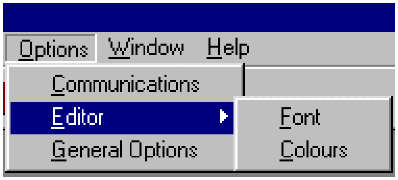


If the programs in memory have been edited, the button will be highlighted to remind you to fix into eprom before exiting the program.



Configuring The *Motion* Perfect 2 Desktop

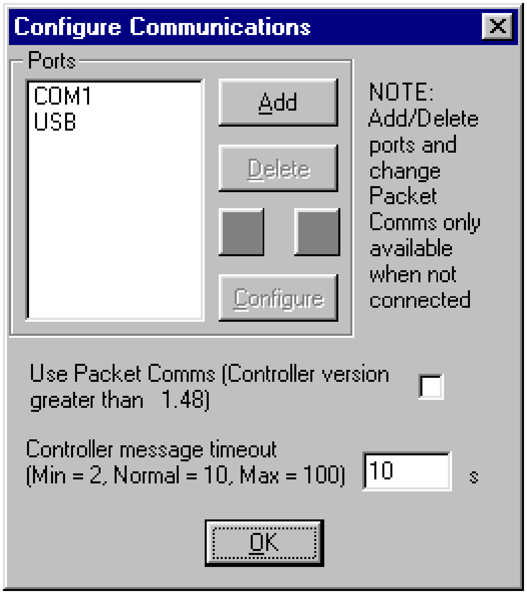
There are a number of ways in which you can configure *Motion* Perfect 2 to suit your requirements.   The Options menu provides a number of choices:-



Communications

Set up the default communications device for *Motion* Perfect 2 to use.

Motion Perfect 2 needs a single serial connection to the controller in order to operate. This can be an RS-232 serial connection, or a USB connection if your controller and PC have USB ports. It will normally recognise the ports installed in your PC and will display these in the Configure Communications window.


Configure Communications

Ports

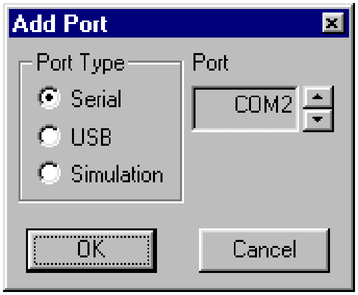
COM1
USB

NOTE: Add/Delete ports and change Packet Comms only available when not connected

Use Packet Comms (Controller version greater than 1.48) □

Controller message timeout (Min = 2, Normal = 10, Max = 100) 10 s

If the port you wish to use is not shown, you need to select the Add Port option which will select the following dialog.



Add Port

Port Type

● Serial

○ USB

○ Simulation

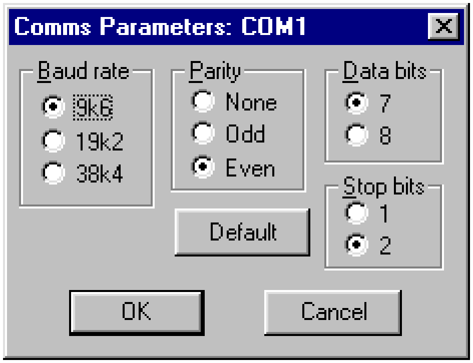
Port: COM2

Most *Motion* Perfect 2 users will connect via a standard RS-232 serial (COM) port. Other options include a connection via a high speed Universal Serial Bus (USB) port or a controller simulation for offline editing.

Choose the port options you require and click on the "OK" button to install the new port.

Changing Comms Port Parameters

The default settings for serial communications is:- 9600 baud, 7 data bits, 2 stop bits, even parity. If you wish to change these values you can do so with the configure button in the Configure Communications dialog.



Comms Parameters: COM1

Baud rate

● 9k6

○ 19k2

○ 38k4

Parity

○ None

○ Odd

● Even

Data bits

● 7

○ 8

Stop bits

○ 1

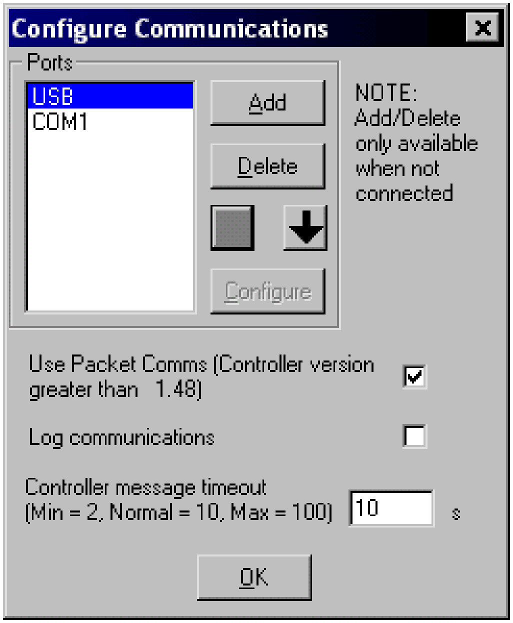
● 2

Note: *If you change the port setting to anything other than the default, you may encounter problems when the controller is reset because it will revert to the default values.*

In order to avoid this your controller will need to set the comms parameters within an auto-running program. See the SETCOM instruction in the Trio BASIC reference for further information.

Setting the Default Port

You can change the order in which the ports are scanned by the program by selecting the port in the list and using the up/down arrows to move the items as required.



NOTE:
Add/Delete
only available
when not
connected

Use Packet Comms (Controller version greater than 1.48) ☑

Log communications □

Controller message timeout (Min = 2, Normal = 10, Max = 100) 10 s

Packet Based Communications

Motion Perfect 2 features an enhanced communications protocol which uses a packet based structure with error checking to significantly improve the reliability of the serial port communications. Packet communications can be used over any type of communications link but as some of these (USB, ethernet) use packetised communications as part of their own protocol, adding an extra level of packetisation may make communications less efficient without giving any reliability improvement.

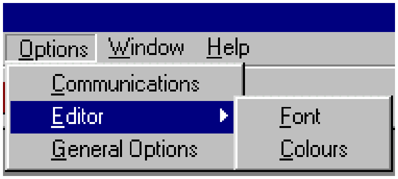
In order to use the Packet Comms mode you must have system software 1.49 or higher.

Controller message timeout

Certain controller operations may cause the controller to stop communicating for a few seconds. If you find that your PC seems to disconnect often, you can change *Motion* Perfect 2's default timeout value to allow the program to wait for a longer time before disconnecting.

Editor Options

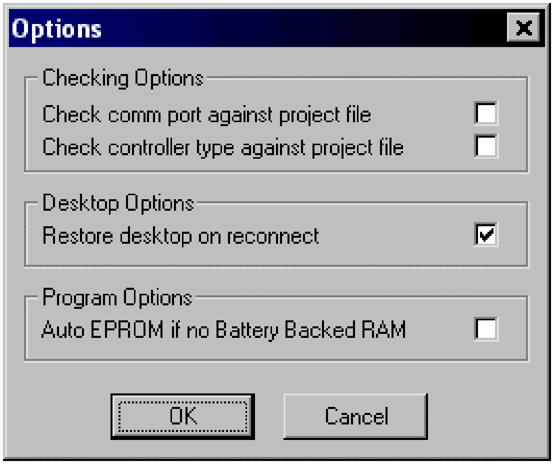
The Editor sub-menu allow you to modify the appearance of the *Motion* Perfect 2 editor to suit your own personal taste. You can change both the default font used and the colours used by the syntax highlighting feature.



General Options

This dialog allows the user to change a number of options relating to how *Motion* Perfect 2 starts up and handles projects.

When you select General Options you will be presented with the following screen.


Options

Checking Options

Check comm port against project file □

Check controller type against project file □

Desktop Options

Restore desktop on reconnect ☑

Program Options

Auto EPROM if no Battery Backed RAM □

The check-boxes enable the following features:

Check comm port against project file

Checks the comm port being used against the one in the project file when a Check Project operation is performed.

Check controller type against project file

Checks the type of the connected controllerg used against the one in the project file when a Check Project operation is performed.

Restore desktop on reconnect

If this option is selected, the program will attempt to automatically save the desktop layout when disconnecting from the controller.

When you reconnect, *Motion* Perfect 2 will automatically restore the last desktop layout saved.

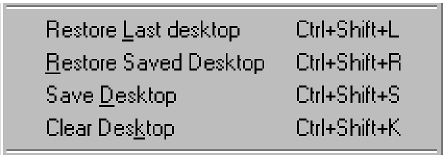
Auto EPROM if no Battery Backed RAM

If the controller does not feature battery-backed RAM memory and this option is selected, the program will attempt to automatically save the controller memory to EPROM before disconnecting from the controller.

Saving the Desktop Layout

When you have a number of windows open, you can save the layout so that it can be quickly restored later. Alternatively the desktop can be set to restore automatically on each re-connection by ticking the checkbox under the menu: Options/General options.

From the Window menu...


Restore Last desktop Ctrl+Shift+L
Restore Saved Desktop Ctrl+Shift+R
Save Desktop Ctrl+Shift+S
Clear Desktop Ctrl+Shift+K

Restore Last Desktop

Restores the last desktop which was automatically saved by *Motion* Perfect 2 when it disconnected from the controller

Restore Saved Desktop

Restore the last desktop saved using the Save Desktop option.

Save Desktop

Saves the current desktop layout to a file on disk

Clear Desktop

Closes all open tool windows.

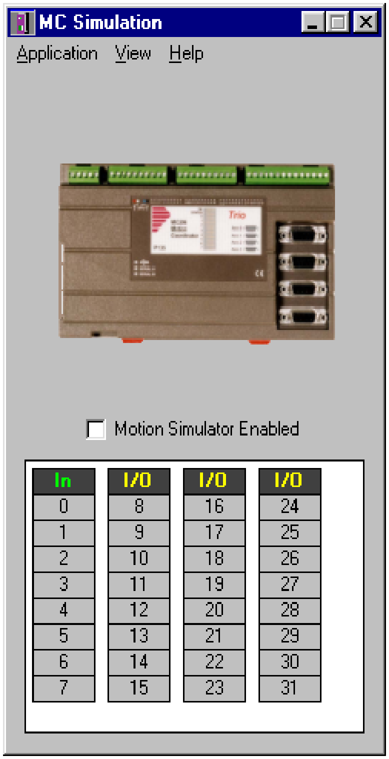
Running *Motion* Perfect 2 Without a Controller

Normally you will run *Motion* Perfect 2 on-line, that is connected to a controller. In fact *Motion* Perfect 2 is designed to operate in this manner and has little functionality without the connection.

In order that you can view or edit your project programs without a controller connected there is a special application to simulate the controller operation and to allow *Motion* Perfect 2 to operate in many ways as if a real controller were connected.

MC Simulation

MC Simulation (MCSim) is a very simple program designed to run alongside *Motion* Perfect 2 in the background. There are no options or configurations to worry about, you just have to run the program and connect as usual.

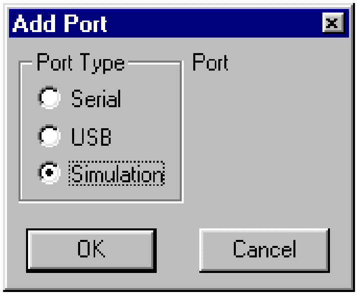


☐ Motion Simulator Enabled

In	I/O	I/O	I/O
0	8	16	24
1	9	17	25
2	10	18	26
3	11	19	27
4	12	20	28
5	13	21	29
6	14	22	30
7	15	23	31

Starting MC Simulatiion from *Motion* Perfect 2

MC Simulatoion is automatically started (if it is not already running) when *Motion* Perfect tries to connect to it. To connect to MCSimuolation either use the **Connect to Simulator** tool button or set up an **MCSim** port in the connection list.



Add Port

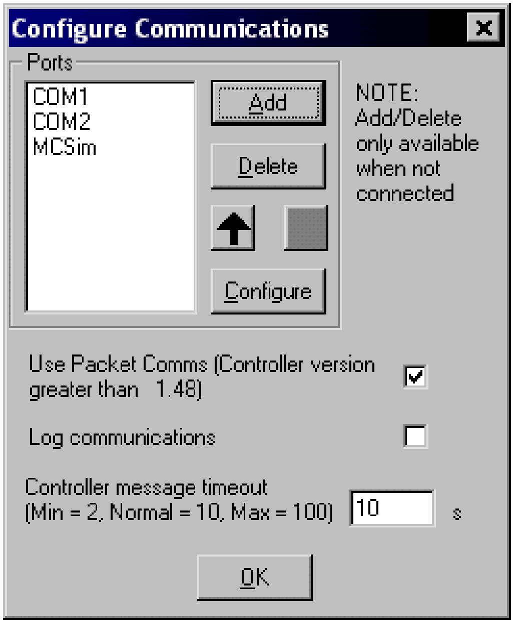
Port Type

○ Serial

○ USB

● Simulation

Use the Add Port option to select a new port and choose "Simulation" as the Port Type.



Configure Communications

Ports

COM1
COM2
MCSim

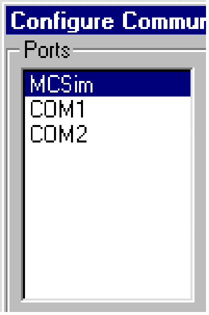
NOTE: Add/Delete only available when not connected

Use Packet Comms (Controller version greater than 1.48) ☑

Log communications □

Controller message timeout (Min = 2, Normal = 10, Max = 100) 10 s

The new device will normally appear at the end of the list. Use the "move up" button to make it the default option.



Configure Commun

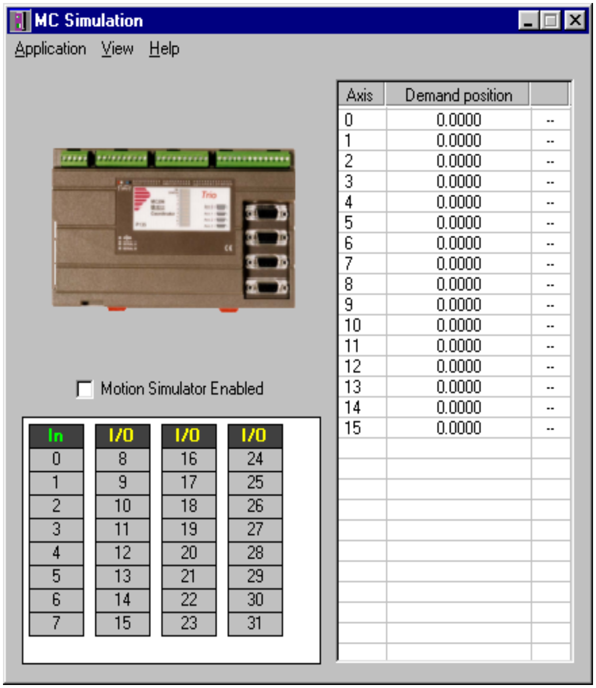
Ports

MCSim
COM1
COM2

Limitations of MC Simulation

The MCSimulation program does not yet cover all the functionality present in a real controller. It does allow connection to *Motion* Perfect for program editing and the running of programs in the simulated environment. There are some unsupported TrioBASIC commands (mainly those related to communications busses such as CAN).

The motion engine built into the simulation is still under development although it will handle all move types except linked moves. There is an axis demand position display which can be used to monitor the axes when moves are taking place. This can be toggled on and off by selecting **View/Axes** from the MCSimulation main menu. The motion engine can be enabled/disabled by checking/unchecking the the **Motion Simulator Enabled** check box.



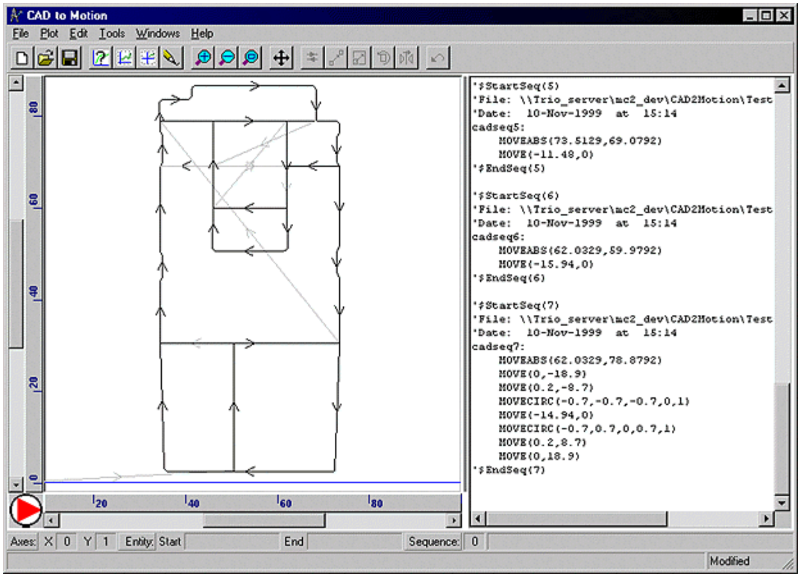
☐ Motion Simulator Enabled

In	I/O	I/O	I/O
0	8	16	24
1	9	17	25
2	10	18	26
3	11	19	27
4	12	20	28
5	13	21	29
6	14	22	30
7	15	23	31

Axis	Demand position	
0	0.0000	--
1	0.0000	--
2	0.0000	--
3	0.0000	--
4	0.0000	--
5	0.0000	--
6	0.0000	--
7	0.0000	--
8	0.0000	--
9	0.0000	--
10	0.0000	--
11	0.0000	--
12	0.0000	--
13	0.0000	--
14	0.0000	--
15	0.0000	--

CAD2Motion

CAD2Motion is a program designed to allow users to translate CAD generated two dimensional motion paths into Trio BASIC programs.



```
'$StartSeq(5)
'File: \\Trio_server\mc2_dev\CAD2Motion\Test
'Date:  10-Nov-1999  at  15:14
cadseq5:
    MOVEABS(73.5129,69.0792)
    MOVE(-11.48,0)
'$EndSeq(5)

'$StartSeq(6)
'File: \\Trio_server\mc2_dev\CAD2Motion\Test
'Date:  10-Nov-1999  at  15:14
cadseq6:
    MOVEABS(62.0329,59.9792)
    MOVE(-15.94,0)
'$EndSeq(6)

'$StartSeq(7)
'File: \\Trio_server\mc2_dev\CAD2Motion\Test
'Date:  10-Nov-1999  at  15:14
cadseq7:
    MOVEABS(62.0329,78.8792)
    MOVE(0,-18.9)
    MOVE(0.2,-8.7)
    MOVECIRC(-0.7,-0.7,-0.7,0,1)
    MOVE(-14.94,0)
    MOVECIRC(-0.7,0.7,0,0.7,1)
    MOVE(0.2,8.7)
    MOVE(0,18.9)
'$EndSeq(7)
```

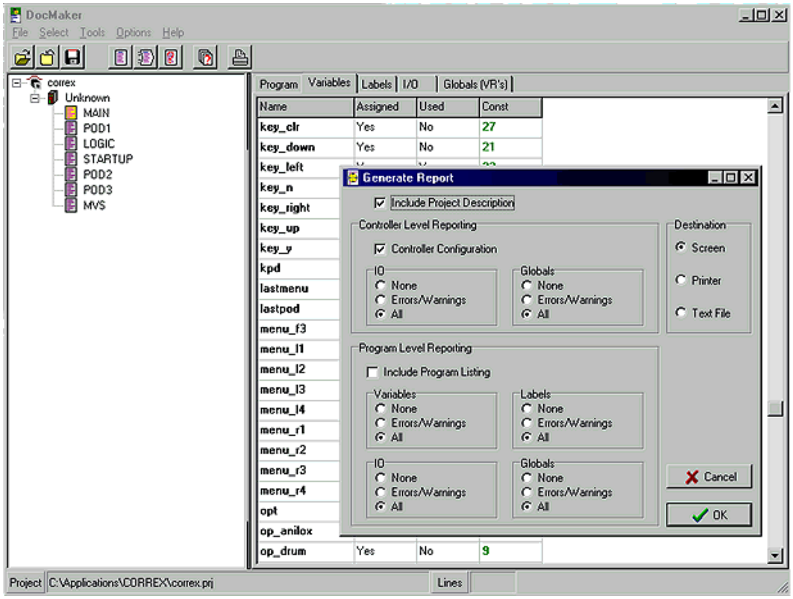
The program allows the user to create motion paths in a CAD package such as AutoCAD and convert them into code executable by a *Motion Coordinator*. Typically the path information will be drawn on a single layer in the CAD package and exported as a DXF file. The DXF file (layer with motion path only) is read into CAD2Motion to create a program to follow the motion path.

The motion path can be manipulated and edited before being saved as a Trio BASIC program file which can be loaded on to a *Motion Coordinator*.

- Can read Industry standard DXF files and TrioBASIC files.
- Outputs files as TrioBASIC programs.
- Graphical display of motion path with full pan and zoom facilities.
- Built in program text editor.
- Program list and graphical display linked together to show correlation between program code and moves within the motion path.
- Built in tools to mirror, scale, shift, reverse and rotate motion paths.
- Full undo on all tool and editor operations.
- Will handle multiple motion paths (sequences) in the same program.

DocMaker

DocMaker is a Windows application designed to assist in documenting a Trio BASIC project created with *Motion* Perfect.



Name	Assigned	Used	Const
key_clr	Yes	No	27
key_down	Yes	No	21
key_left			
key_n			
key_right			
key_up			
key_y			
kpd			
lastmenu			
lastpod			
menu_f3			
menu_l1			
menu_l2			
menu_l3			
menu_l4			
menu_r1			
menu_r2			
menu_r3			
menu_r4			
opt			
op_anilox			
op_drum	Yes	No	9

☑ Include Project Description

Controller Level Reporting

☑ Controller Configuration

IO: ○ None ○ Errors/Warnings ● All

Globals: ○ None ○ Errors/Warnings ● All

Destination: ● Screen ○ Printer ○ Text File

Program Level Reporting

☐ Include Program Listing

Variables: ○ None ○ Errors/Warnings ● All

Labels: ○ None ○ Errors/Warnings ● All

IO: ○ None ○ Errors/Warnings ● All

Globals: ○ None ○ Errors/Warnings ● All

DocMaker analyses the content of the program files in the project. It can be used to print program listings and to report on the programs (variables, labels, I/O and VR's) and on overall I/O and VR usage. There is also a checking routine which does a quick check on the whole project and flags up possible errors

DocMaker Benefits

- Automatic Analysis of MotionPerfect Project Files
- Highlight potential errors due to labels or variables
- Generates fully cross-referenced reports
- Reformat programs with auto-indenting

Docmaker Hardware Requirements

- IBM PC or Compatible running Microsoft Windows 95 or higher
- Works with all current Motion Coordinators: MC202, MC204, MC216, MC2 and Euro205

Project Autoloader

Trio Project Autoloader is a stand alone program to load projects created using Motion Perfect 2 onto a Trio Motion Coordinator.

The program is small enough to fit onto a 1.44MByte floppy disk and is intended for easy loading of projects onto controllers without the need to run Motion Perfect and so allows OEM manufacturers to update customers equipment easily.

Operation of the program is controlled using a script file which gives a series of commands to be processed, in order, by the program.

Using the Autoloader

General

The autoloader is primerally intended to be used from a floppy disk to update controllers already installed in equipment to allow OEM manufacturers to update customers equipment easily. It can also be used from a hard disk or CD-ROM.

Script File

The commands to be executed are held in a script file AutoLoader.tas which must be in the LoaderFiles directory.

Project

The project to be loaded using LOADPROJECT is in the form of a normal Motion Perfect 2 project. This consists of a directory containing a project definition file and TrioBASIC program files. The directory must have the same name as the project definition file less the extension.

i.e. project definition file TestProj.prj, directory TestProj

The project directory must be in the LoaderFiles directory.

Tables

Any tables to be loaded must be in the form of *.lst files produced by Motion Perfect 2.

Normally these table files will be in the LoaderFiles directory.

Extra Programs

Programs which need to be loaded using LOADPROGRAM because they are not in the project being loaded (or if no project is being loaded)

Normally these program files will be in the LoaderFiles directory.

Files

The autoloader is designed to work with the following file structure (fixed names are shown in bold type).

Base Directory	AutoLoader.exe		
	LoaderFiles	AutoLoader.tas	
		Table1.lst	
		ExtProg1.bas	
		Project	Project.prj
			Prog1.bas
			Prog2.bas

Where:

Base Directory is normally the root directory on a floppy disk (A:\), but can be any directory.

Project is the Motion Perfect 2 project directory for the project to be loaded using the LOADPROJECT command, Project.prj being the project file and Proj?.bas are the program files in the project.

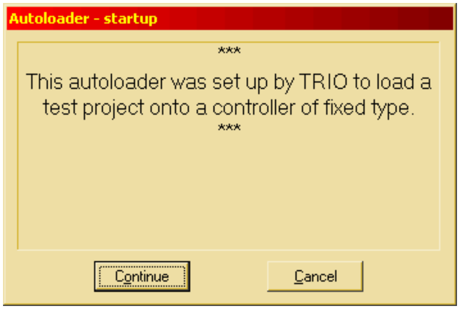
Table?.lst are the table files to be loaded using the LOADTABLE command.

ExtProg?.bas are the extra programs to be loaded using the LOADPROGRAM command.

Running the program

The program can be started in the same way as any other Windows program.

Start Dialog

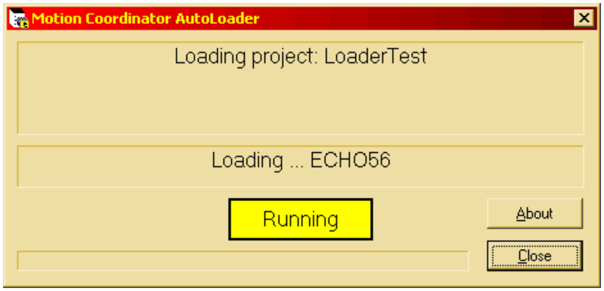


Autoloader - startup

***
This autoloader was set up by TRIO to load a test project onto a controller of fixed type.
***

The start dialog displays a message specified in the script and has continue and cancel buttons so that the user can exit from the program without running the script.

Main Window



Motion Coordinator AutoLoader

Loading project: LoaderTest

Loading ... ECHO56

Running

The program main window consists of two message windows; one to display the current command and the other to display the name of the program or file currently being loaded. There is a button to show the current status (Starting, running, pass or fail) and a progress bar to show the progress during file and table loading.

The close button closes the dialog. If it is pressed while a script is being processed then script processing will be terminated at the end of the current operation.

Script Commands

The following commands are available for use in script files

```
COMMLINK
CHECKTYPE
CHECKVERSION
CHECKUNLOCKED
LOADPROGRAM
LOADPROJECT
EPROM
SETRUNFROMEPROM
LOADTABLE
Comment
```

All commands return a result of OK or Fail. An OK result allows script execution to continue, a Fail result will make script execution terminate at that point.

COMMLINK

Purpose: To set the communications port and parameters.

Syntax: COMMLINK <PortSpec>

Where <PortSpec> is a string specifying a communications port and the connection parameters.

For a serial port this string is similar to COM1:9600,7,e,2 to specify the port, speed, number of data bits, parity and number of stop bits. 9600,7,e,2 are the default parameters for a controller.

Examples: COMMLINK COM2:9600,7,e,2

CHECKTYPE

Purpose: To check the controller type.

Syntax:

```
CHECKTYPE <Controller List>
```

Where <Controller List> is a comma separated list of one or more valid controller ID numbers.

i.e. 206,216

Examples:

```
CHECKTYPE 206
CHECKTYPE 202,216,206
```

Controller ID Numbers

Each type of controller returns a different ID number in response to the TrioBASIC command ?CONTROL[0] . The table below gives the ID number for current controllers.

Controller	ID Number
MC202	202
MC204	204
Euro205	205
MC206	206
MC216	216

CHECKVERSION

Purpose: To check the version of the controller system code.

Syntax:

```
CHECKVERSION <Operator><Version>
CHECKVERSION <LowVersion>-<HighVersion>
```

Examples:

```
CHECVERSION >1.49
CHECKVERSION >= 1.51
CHECKVERSION 1.42-1.50
```

CHECKUNLOCKED

Purpose: To check that the controller is not locked.

Syntax:

```
CHECKUNLOCKED
```

LOADPROJECT

Purpose: To load a project from disk onto the controller.

Syntax: `LOADPROJECT <ProjectName>`
Where <ProjectName> is the path of the project directory. If the project directory is in the LoaderFiles directory then it is just the name of the of the project directory.

Examples: `LOADPROJECT TestProj`

LOADPROGRAM

Purpose: To load a program which is not part of a project from disk onto the controller.

Syntax: `LOADPROGRAM <ProgramFile>`

Where <ProgramFile> is the path of the program file. If the program file is in the LoaderFiles directory then it is just the name of the of the program file.

Note: ***This command should always be used after the LOADPROJECT command*.**

Examples: `LOADPROGRAM TestProg.bas`

LOADTABLE

Purpose: `To load a table onto the controller.`

Syntax: `LOADTABLE <TableFile>`
Where <TableFile> is the path of the table file. If the table file is in the LoaderFiles directory then this is just the file name of the table file.

This command should always be used after the LOADPROJECT command.

Examples: `LOADTABLE Tbl.lst`

EPROM

Purpose: To store the project currently in controller RAM into EPROM

Syntax: `EPROM`

SETRUNFROMEPROM

Purpose: To set the controller to use the programs stored in its EPROM. (It actually copies the programs from EPROM into RAM at startup).

Syntax: `SETRUNFROMEPROM <State>`

Where <State> is 1 for copy from EPROM and 0 is use programs currently in RAM.

A single @ character can be used to specify state in the project file.

Examples:

```
SETRUNFROMEPROM 1
SETRUNFROMEPROM @
```

Note: This command only applies to controllers which have battery backed RAM (controllers with no battery backed RAM will always copy programs from EPROM).

Script File

The autoloader program uses a script file AutoLoader.tas as a source of commands. These commands are executed in order until all commands have been processed or an error has occurred.

If any command fails the exececution terminates without completing the scripted command sequence.

Sample Script

```
' Test Script
' **************
' Startup Message
# ***
# This autoloader was set up by TRIO to load a test project
# onto a controller of fixed type.
# ***
COMMLINK COM1:9600,7,e,2
CHECKTYPE 206
CHECKVERSION > 1.45
CHECKUNLOCKED
LOADPROJECT LoaderTest
LOADTABLE tbl_1.lst
CHECKPROJECT LoaderTest
LOADPROGRAM flashop.bas
LOADPROGRAM clrtable.bas
LOADPROGRAM settable.bas
EPROM
SETRUNFROMEPROM @
```

For this script to work correctly the LoaderFiles directory must contain a project directory LoaderTest, a table file tbl_1.lst and three program files: flashop.bas, clrtable.bas and settable.bas.

Creating an AutoLoader floppy disk

To create an AutoLoader floppy disk follow the steps below:

- Take a blank floppy disk and copy the AutoLoader.exe program file into the root directory.
- Create a sub-directory called LoaderFiles in the root directory.
- If you are going to load a project then copy the project directory of the project to be loaded into the LoaderFiles directory created above, so that it is a sub-directory of LoaderFiles.
- If you are going to load any tables then copy all the table files into the LoaderFiles directory.
- If you are going to load any extra programs then copy the program files into the LoaderFiles directory.
- Using an ASCII text editor (i.e. Windows Notepad) create and edit the script file (AutoLoader.tas) in the LoaderFiles directory, using the appropriate commands for the operations to be carried out.