# Using

# Mach3Turn

## A user's guide to installation, configuration and operation

# Using Mach3Turn



## or
## The nurture, care and feeding of the Mach3 controlled CNC Lathe or Borer

All queries, comments and suggestions welcomed via support@artofcnc.ca

Mach Developers Network (MachDN) is currently hosted at:
http://www.machsupport.com

© 2003/4/5/6 Art Fenerty and John Prentice

**Front cover: A vertical borer circa 1914**

**This manual is for Mach3Turn Release 1.84**

# Contents

# Contents

# Contents

# Contents

# 1.    Preface

Any machine tool is potentially dangerous. Computer controlled machines are potentially more dangerous than manual ones because, for example, a computer is quite prepared to rotate an 8" unbalanced cast iron four-jaw chuck at 3000 rpm!

This manual tries to give you guidance on safety precautions and techniques but because we do not know the details of your machine or local conditions we can accept no responsibility for the performance of any machine or any damage or injury caused by its use. It is your responsibility to ensure that you understand the implications of what you design and build and to comply with any legislation and codes of practice applicable to your country or state.

**If you are in any doubt you must seek guidance from a professionally qualified expert rather than risk injury to yourself or to others.**

This document is intended to give enough details about how the Mach3Turn software interacts with your machine tool, how it is configured for different axis drive methods and about the input languages and formats supported for programming to enable you to implement a powerful CNC turning system. Typical machine tools that can be controlled are lathes and vertical boring machines.

A companion online document (a wiki) *Customising Mach3* at www.machsupport.com explains in detail how to alter screen layouts, to design your own screens and Wizards and to interface to special hardware devices.

You are strongly advised to join one of the two online discussion groups for Mach3.

One is hosted by Yahoo! and contains postings on detailed technical; issues and bug reports for development versions of the Mach software.

The other is a "forum" style group with threads of a more general nature about the applications on Mach3 to CNC work.

Links to both are on the home page of www.machsupport.com.

You should, however, be aware that, while the online groups have many engineers with a vast range of experience as participants, they does not constitute a substitute for a contract with machine tool manufacturer's support network. If your application requires this level of support then you should buy the system from a local distributor or an OEM with a distributor network. In that way you will get the benefits of Mach3 with the possibility of on-site support.

Certain portions of text in this manual are printed "greyed out". They generally describe features found in machine controllers but which are not presently implemented in Mach3. The description of a greyed out feature here is not to be taken as a commitment to implement it at any given time in the future.

Thanks are due to numerous people including the original team who worked at National Institute for Standards and Testing (NIST) on the EMC project, Steve Blackmore for pioneering use of Mach2 and Mach3, contribution of  CAM post-processors and drafting and review of documentation, and to all the users of Mach2 and Mach3 without whose experience, materials and constructive comments this manual could not have been written. Credits are given for individual utilities and features as these are described in the body of the manual.

ArtSoft Corporation is dedicated to continual improvement of its products, so suggestions for enhancements, corrections and clarifications will be gratefully received.

Art Fenerty and John Prentice assert their right to be identified as the authors of this work. The right to make copies of this manual is granted solely for the purpose of evaluating and/or using licensed or demonstration copies of Mach3. It is not permitted, under this right, for third parties to charge for copies of this manual nor to distribute it except in its

entirety. OEM licensing of the software includes access to the source material to allow detailed customisation.

Every effort has been made to make this manual as complete and as accurate as possible but no warranty or fitness is implied. The information provided is on an "as is" basis. The authors and publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this manual, Use of the manual is covered by the license conditions to which you must agree when installing Mach3 software.

Windows XP and Windows 2000 are registered trademarks of Microsoft Corporation. If other trademarks are used in this manual but not acknowledged please notify ArtSoft Corporation so this can be remedied in subsequent editions.

# 2. Introducing CNC machining systems

## 2.1 Parts of a machining system

> This chapter will introduce you to terminology used in the rest of this manual and allow you to understand the purpose of the different components in a numerically controlled turning system.

The main parts of a system for numerically controlled turning system are shown in figure 1.1

The designer of a part generally uses a Computer Aided Design/Computer Aided Manufacturing (CAD/CAM) program or programs on a computer (1). The output of this
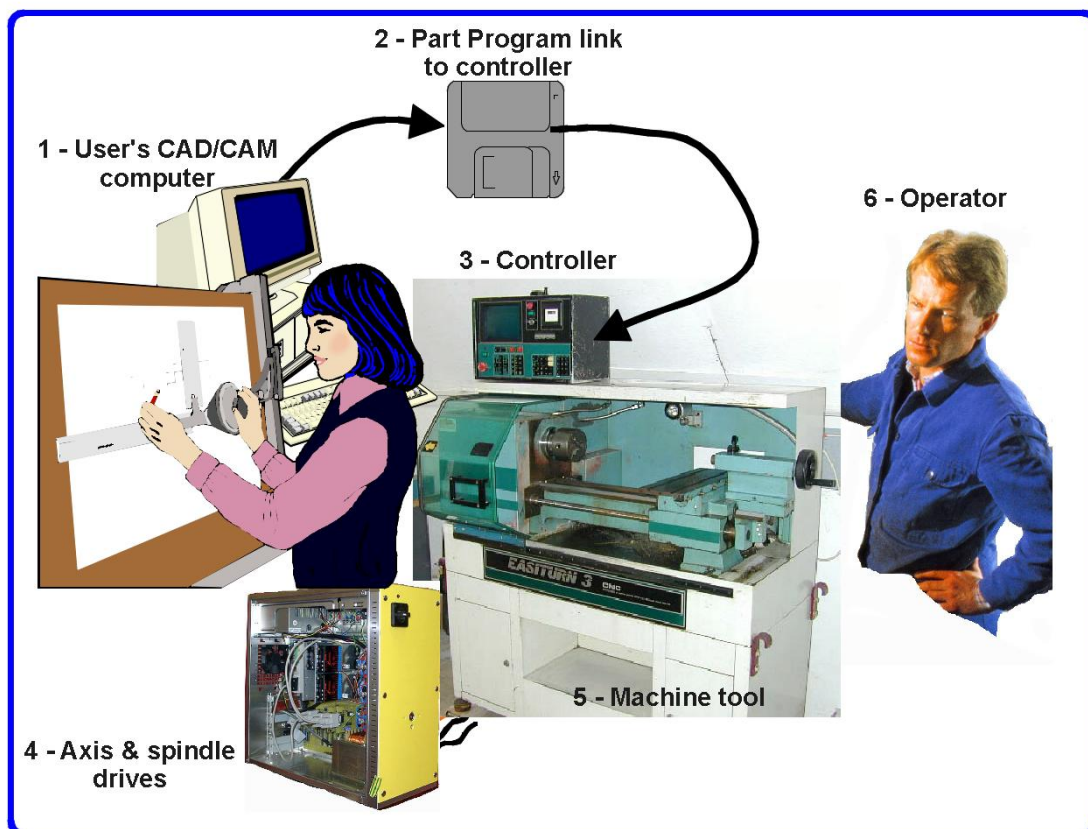


**Figure 1.1 - Typical NC machining system**

program, which is a part program and is often in "G-code" is transferred (by a network, USB "jump" or "key" drive or perhaps floppy disc) (2) to the Machine Controller (3). The Machine Controller is responsible for interpreting the part program to control the tool which will cut the workpiece. The axes of the Machine (5) are moved by screws which are powered by servo motors or stepper motors. The signals from the Machine Controller are amplified by the Drives (4) so that they are powerful enough and suitably timed to operate the motors.

Although a conventional lathe is illustrated, the machine can be a slant-bed lathe, vertical borer etc. We will generally use the term "lathe" in this manual to apply to any sort of turning machine. Separate documentation describes customizing Mach3 and using Mach3 for controlling a mill or similar machine.

Frequently the Machine Controller will control starting and stopping of the spindle motor and its speed, will turn coolant on and off and will check that a part program or Machine Operator (6) are not trying to move any axis beyond its limits.

The Machine Controller also has controls like buttons, a keyboard, potentiometer knobs, a manual pulse generator (MPG) wheel, or a joystick so that the Operator can control the machine manually and start and stop the running of the part program. The Machine Controller has a display so that the Operator knows what is happening.

Because the commands of a G-code program can request complicated co-ordinated movements of the machine axes, the Machine Controller has to be able to perform a lot of calculations in "real-time". Historically this made it an expensive piece of equipment.

## 2.2    How Mach3 fits in

Mach3 is a software package which runs on a PC and turns it into a very powerful and economical Machine Controller to replace (3) in figure 1.1. It also includes features (called Mach Wizards) which allows you to program simple parts by entering data conversationally, thus avoiding the use of separate CAD/CAM software.

To run Mach3 you need Windows XP (or Windows 2000) ideally running on a 1GHz processor with a 1024 x 768 pixel resolution screen. A desktop machine will give much better performance than most laptops and be considerably cheaper. You can, of course use this computer for any other functions in the workshop (such as (1) in figure 1.1 - running a CAD/CAM package) but not when it is controlling your machine tool.

Mach3 communicates with the machine tool and its drives via one (or optionally two) parallel (printer) ports and, if desired, a serial (COM) port using the ModBus protocol.

The hardware drivers for your machine's axis motors must be designed accept step pulses and a direction signal. Virtually all stepper motor drivers work like this, as do modern DC and AC servo systems with digital encoders. **Beware** if you are converting an old NC machine whose servos may use resolvers to measure position of the axes as you will have to provide a complete new drive for each axis.

# 3. An overview of Mach3 Machine Controller software

> You are still reading this so evidently you think Mach3 might be an asset in your workshop! The best thing to do now is to download a free demonstration version of the software and try it out on your computer. You do not need a machine tool to be connected up, indeed for the present it is better not to have one.
>
> If you have bought a complete system from a reseller then some or all of these installation steps may have be done for you already.

## 3.1 Installation

Mach3 is distributed by ArtSoft Corp. via the Internet. You download the package as one self installing file (which, in the present release, is about 9 megabytes). This will run for an unlimited period as a demonstration version with a few limitations on the speed, the size of job that can be undertaken and the specialist features (e.g. threading) supported. When you purchase a licence this will "unlock" the demonstration version you have already installed and configured. Full details of pricing and options are on the ArtSoft Corporation website www.artofcnc.ca

When you install Mach3 you will get the software for Turning and Milling. In fact the code for these two functions is common although the user interfaces look very different, being customised to suit the requirements and workflow of each type of machine. When you gain experience with Mach3Turn you may find it instructive to see how Mach3's features are exploited in Mach3Mill and consider if they could be used on your lathe.

### 3.1.1 Downloading

Download the package from www.artofcnc.ca using the right mouse button and *Save Target as...* to put the self-installing file in any convenient working directory (perhaps Windows\Temp). You should be logged in to Windows as an Administrator.

When the file has downloaded it can be immediately run by using the *Open* button on the download dialog or this dialog can be closed for later installation. When you want to do the installation you merely run the downloaded file. For example you could run Windows Explorer (right click *Start* button), and double-click on the downloaded file in the working directory.

### 3.1.2 Installing

You do not need a machine tool connected yet. If you are just starting it would be better not to have one connected. Note where the cable or cables from the machine tool are plugged into your PC. Switch off the PC, the machine tool and its drives and unplug the 25 pin connector(s) from the back of the PC. Now switch the PC back on.

When you run the downloaded file you will
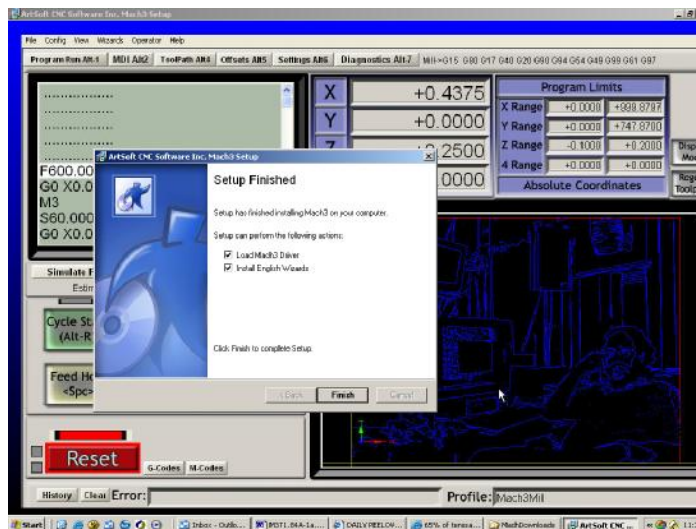


**Figure 3.1 – Driver options**

be guided through the usual installation steps for a Windows program such as accepting the license conditions and selecting the folder for Mach3.

The background image during installation is the standard Mach3Mill screen – do not worry as Mach3Turn is also being installed.

On the Setup Finished dialog you should ensure that *Load Mach3 Driver* and *Install English Wizards* are checked and then click *Finish.* You will now be told to reboot before running any Mach3 software.

### 3.1.3    The vital re-boot

This reboot is **vital**. If you do not do it then you will get into great difficulties which can only be overcome by using the Windows Control Panel to uninstall the driver manually. **So please reboot now.**

If you are interested in knowing why the reboot is required then read on, otherwise skip to the next section.

Although Mach3 will appear to be a single program when you are using it, it actually consists of two parts: a driver which is installed as part of Windows like a printer or network driver and a graphical user interface (GUI).

The driver is the most important and ingenious part. Mach3 must be able to send very accurately timed signals to control the axes of the machine tool. Windows likes to be in charge and runs normal user programs when it has nothing better to do itself. So Mach3 cannot be a "normal user program"; it must be at the lowest level inside Windows (that is it handles interrupts). Furthermore, to do this at the high speeds possibly required (each axis can be given attention 45,000 times per second), the driver needs to tune its own code. Windows does not approve of this (it's a trick that viruses play) so it has to be asked to give special permission. This process requires the reboot. So if you have not done the re-boot then Windows will give the Blue Screen of Death and the driver will be corrupt. The only way out of this will be to manually remove the driver.

Having given these dire warnings, it is only fair to say that the reboot is only required when the driver is first installed. If you update your system with a newer version then the reboot is not vital. The install sequence does however still ask you to do it. Windows XP boots reasonably quickly that it is not much hardship to do it every time.

### 3.1.4    Convenient desktop icons

So you **have** rebooted! The installation wizard will have created desktop icons for the main programs. Double-clicking the Mach3Turn shortcut is the usual way to run the program.

Mach3.exe is the actual GUI  code. If you run it, it will ask which Profile you wish to use. Mach3Mill, Mach3Turn etc. are just shortcuts which run this with a Profile defined by a "/p" argument in the shortcut target You will usually employ these to start the required system.

It is now worthwhile to setup some icons for desktop shortcuts to other Mach3 programs. Use Windows Explorer (right-click *Start*) and by right-clicking on the DriverTest.exe file. Drag this shortcut onto your desktop. Other programs such as a screen designer and a manipulator for screenset files are available as a separate download.
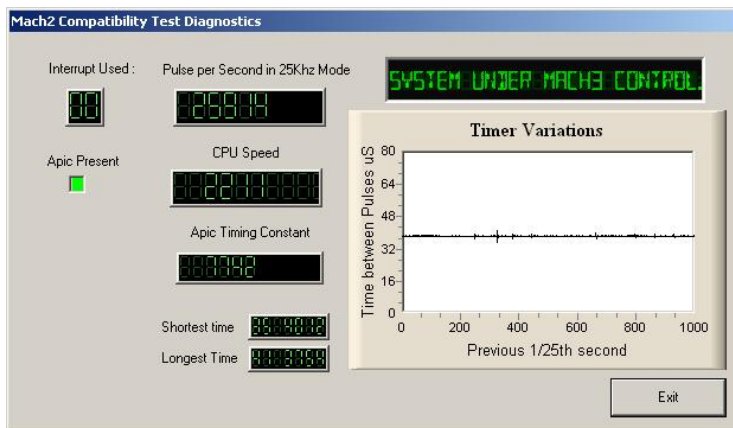


**Figure 3.2 – The driver test program display**

### 3.1.5    *Testing the installation*

It is now highly recommended to test the system. As mentioned above, Mach3 is not a simple program. It takes great liberties with Windows in order to perform its job; this means it will not work on all systems due to many factors. For example, the QuickTime system monitor (qtask.exe) running in the background can kill it and there will be other programs which you probably are not even aware are on your system that can do the same. Windows can and does start many processes in the background; some appear as icons in the system tray (bottom right of screen) and others do not show themselves in any way. Other possible sources of  erratic operation are local area network connections which may be configured to automatically speed detect. You should configure these to the actual speed 10 Mbps or 100 Mbps of your network. Finally a machine that has been surfing the Internet may have gained one or more of a host of "robot" type programs which spy on what you are doing and send data over the 'net to their originators. This traffic can interfere with Mach3 and is not something you want anyway. Use a search engine for terms like "Spybot" to locate software to tidy up your machine.

Because of these factors, it is important, though not mandatory, that you test your system when you suspect something is wrong or you just want to check that an install went well.

Double click the DriverTest icon that you set up. Its screen shot is in figure 3.2.

You can ignore all the boxes with the exception of the Pulse Frequency. It should be fairly steady around 25,000 Hz but yours may vary, even quite wildly. This is because Mach3 uses the Windows clock to calibrate its pulse timer and, over a short time scale, the Windows clock can be affected by other processes loading the computer. So you may actually be using an "unreliable" clock (the Windows one) to check Mach3 and so get the false impression that Mach3's timer is unsteady.

Basically, if you see a similar screen to figure 3.2 with only small spikes on the Timer Variations graph and a steady Pulse Freqency, everything is working well so **close the DriverTest program and skip to the section** *Screens* below.

Windows "experts" might be interested to see a few other things. The white rectangular window is a type of timing analyzer. When it is running it displays a line with small variations indicated. These variations are the changes in timing from one interrupt cycle to another. There should be no lines longer than ¼ inch or so on an 17" screen on most systems. Even if there are variations its possible they are below the threshold necessary to create timing jitters so when your machine tool is connected you should perform a movement test to see if jogging and G0/G1 moves are smooth.

You may have one of two things happen to you when running the test which may indicate a problem.

1.  "Driver not found or installed, contact Art.", this means that the driver is not loaded into Windows for some reason. This can occur on XP systems which have a corruption of their driver database, reloading Windows is the cure in this case. Or, you may be running Win2000. Win2000 has a bug/"feature" which interferes with loading the driver. It may need to be loaded manually see the next section

2.  When the system says, taking over…3…2…1.. and then reboots, one of two things has occurred. Either you didn't reboot when asked (told you!!) or the driver is corrupted or unable to be used in your system. In this case follow the next section and remove the driver manually, then re-install. If the same thing happens, please notify ArtSoft using the e-mail link on [www.artofcnc.ca](www.artofcnc.ca)  and you will be given guidance.
    A few systems have motherboards which have hardware for the APIC timer but whose BIOS code does not use it. This will confuse Mach3 install. A batch file `SpecialDriver.bat` is available in the Mach3 installation folder. Find it with Windows Explorer and double-click it to run it. This will make the Mach3 driver use the older i8529 interrupt controller. You will need to repeat this process whenever you download an upgraded version of Mach3 as installing the new version will replace the special driver. The file `OriginalDriver.bat` reverses this change.

### 3.1.6   *DriverTest after a Mach3 crash*

Should you for any reason have a situation when running Mach3 where it crashes - this might be an intermittent hardware problem or a software bug – then you **must** run DriverTest.exe as soon as possible after Mach3 has failed. If you delay for two minutes then the Mach3 driver will cause Windows to fail with the usual "Blue Screen of Death". Running DriverTest resets the driver to a stable condition even if Mach3 disappears unexpectedly.

You may find, after a crash, that it fails to find the driver the first time it is run. In this case merely run it again as the first run should fix things up.

### 3.1.7   *Notes for manual driver installation and un-installation*

**You only need to read and do this section if you have not successfully run the DriverTest program.**

The driver (Mach3.sys) can be installed and uninstalled manually using the Windows control panel. The dialog boxes differ slightly between Windows 2000 and Windows XP but the steps are identical.

> Open the Control panel and double-click on the icon or line for *System*.
>
> Select *Hardware* and click *Add Hardware wizard*. (As mentioned before Mach3's driver works at the lowest level in Windows). Windows will look for any new actual hardware (and find none).
>
> Tell the wizard you have already installed it and then proceed to the next screen.
>
> You will be shown a list of hardware. Scroll to the bottom of this and select *Add a new hardware device* and move to the next screen.
>
> On the next screen you do not want Windows to search for the driver so select *Install the hardware that I manually select from a list (Advanced)*
>
> The list you are shown will include an entry for *Mach1 pulseing engine*. Select this and go to the next screen.
>
> Click *Have disc* and on the next screen point the file selector to your Mach3 directory (C:\Mach3 by default). Windows should find the file *Mach3.inf*. Select this file and click *Open*. Windows will install the driver.

The driver can be uninstalled rather more simply.

> Open the Control panel and double-click on the icon or line for System.
>
> Select *Hardware* and click *Device Manager*
>
> You will be shown a list of devices and their drivers. *Mach1 Pulsing Engine* has the driver *Mach2 Driver* under it. Use the + to expand the tree if necessary. Right-click on *Mach2 Driver* gives the option to uninstall it. This will remove the file Mach3.sys from the Windows folder. The copy in the Mach3 folder will still be there. Notice that the names here refer to Mach1 and Mach2 even though you are actually dealing with Mach3. This is quite in order.

There is one final point to note. Windows remembers all the information about the way you have configured Mach3 in a Profile file. This information is not deleted by un-installing the driver and deleting other Mach3 files so it will remain whenever you upgrade the system. However in the very unlikely event that you need a totally clean installation from scratch then you need to delete the .XML profile file or files.

## 3.2   Screens

You are now ready to try out a "dry run" Mach3. It will be much easier to show you how to set up your actual machine tool when you have experimented with Mach3 like this. You can "pretend" to machine and learn a lot even if you haven't got a CNC machine tool yet. If you have got one, then do make sure it is not connected to the PC.

Mach3 is designed so that it is very easy to customize its screens to suit the way you work. This means that the screens you see may not look exactly like those in Appendix 1. If there

are major differences then your system supplier should have given you a revised set of screenshots to match your system.

Double-click the Mach3Turn icon on your desktop to run the program. You should see the Welcome screen similar to that in Appendix 1.

Notice the red Reset button. It will have a flashing Red LEDs (a simulation of a light emitting diode) beside it. If you click the button then the red LEDs go out. Mach3 is ready for action!

If you cannot stop the red Reset LEDs flashing then you have probably got an old XML (profile) file on your system or something plugged into your parallel port(s).

For now it will be easiest to work "offline". In the Operator menu, click the Control Offline entry so it becomes "checked". You will now be able to clear the Reset LEDs.

### 3.2.1  Types of object on screens

On the Welcome screen, use the mouse to click the button to switch to the Manual screen. You will see that it is made up of the following types of object:

Buttons (e.g. Spindle, Home X, etc.)

DROs or Digital Readouts. Anything with a number displayed will be a DRO. The main ones are, of course the current positions of the X, Z axes and, for most machines, the spindle speed.

LEDs (in various sizes and shapes)

Toolpath display (blank rectangle on your screen at the moment)

MDI (Manual Data Input) line (blue and below Home X, Home Z, etc. buttons.

There is one further important type of control that is not on the Program Run screen:

G-code display window (with its own scroll bars)

The buttons and the MDI line are your command inputs to Mach3.

DROs can be displays by Mach3 or can be used as inputs by you. The background colour changes when you are inputting.

The G-code window and Toolpath displays are for information from Mach3 to you. You can, however, manipulate both of them (e.g. scrolling the G-code window, zooming, rotating and panning the Toolpath display)

### 3.2.2  Using buttons and shortcuts

On the standard screens many buttons have a keyboard hotkey. Pressing the shortcut key when the screen is displayed is the same as clicking the button with the mouse. On the manual screen the key "Q" is a hotkey for the *Quit* button. Both return you to the Welcome screen. On the Welcome screen, you might like to try "M", "A", "T" and "D" as shortcuts to the Manual, Auto, Tooltable and Diags screens. Although letters are given in uppercase (for ease of reading) you do **not** use the shift key when using the shortcuts.

In a workshop it is convenient to minimise the times when you need to use a mouse. If you are building your own machine you can provide physical switches on a control panel to control Mach3 by use of a keyboard emulator board (e.g. Ultimarc IPAC) or via the ModBus interface. Full details of these features are given in the *Customising Mach3* wiki.

If a button does not appear on the current screen then its keyboard shortcut is not active.

There are certain special keyboard shortcuts which are global across all screens. Chapter 5 shows how these are set up.

### 3.2.3  Data entry to DRO

You can enter new data into any DRO by clicking in it with the mouse, clicking its hotkey (where set) or by using the global hotkey to select DROs and moving to the one that you want with the arrow keys)

On the Manual screen, try entering a feedrate like 45.3.  You **must** press the *Enter*  key to accept the new value or the *Esc* key to revert to the previous one. Clicking the mouse in another DRO is like *Esc*.

*Backspace* and *Delete* are not used when inputting to DROs.

**Caution:** It is not always sensible to put your own data into a DRO. For example the display of actual spindle speed is computed by Mach3. Any value you enter will be overwritten. You can put values into the axis DROs but you should not do it until you have read Chapter 7 in detail. This is **not** a way of moving the tool!

## 3.3    Jogging

You can move the tool relative to any place on your work manually by using various types of Jogging.

Jogging is available on any screen that has the Jog On/Off button on it (figure 3.3). This button is a toggle. One click turns jogging on, another turns it off. A LED indicated when jogging is turned on.

**Figure 3.3**

For now the easiest way to jog the system is using the Up/Down and Left/Right arrow keys on your keyboard. Try these out – with Jog switched on! – and see the X and Z DROs change value. When your machine is connected its tool will be moving. If nothing happens or it is very slow type the value 25 into the *Slow Jog %* DRO (see figure 3.5). Do not forget the *Enter* key to accept the new value.

The arrow keys are set by default to give you jogging on the main axes. You can configure these keys (see Chapter 5) to suit your own preferences – for example for a machine with  it tool behind the centreline.

Beside the *Jog On/Off* button is the *Jog Mode* button (figure 3.4). Associated with this is the *Jog Inc* LED.

**Figure 3.4**

If the *Jog Inc* LED is turned on, by toggling the *Jog Mode* button, then each press on the jog key will move the axis by a fixed amount which you can setup.

In figure 3.5 you will see how to configure continuous jog speed and the size of the step taken with incremental jogging.

In Continuous mode the chosen axis will jog for as long as you hold the key down. The speed of jogging is set by the *Slow Jog %* DRO. You can enter any value from 0.1% to 100% to get whatever speed you want. The - and + buttons beside this DRO will alter its value in 5% steps (or 0.1% steps when below 5%).

If you depress the *Shift* key then the jogging will occur at the full speed of the axis whatever the *Slow Jog Percentage* setting. This allows you to quickly jog to near your destination and then, by releasing *Shift*,  position accurately.

**Figure 3.5 – Jog and other controls**

In *Inc* mode, each press of a jog key will move the axis by the distance indicated below the Jog Cycle label. You can set this to whatever value you like by typing into the DRO or use the – and + buttons to cycle through a set of preset increments. Movement will be at the current Feedrate. If you have the feedrate defined as a feed per revolution then you must specify a spindle speed in the S DRO before you get movement.

If you have problems with incremental jogging then you can leave it for now. You will find it very useful when making small controlled movements of the tool when setting up a job on the machine.

One or two rotary encoders can be interfaced (via the parallel port input pins) to Mach3 and can be used as a Manual Pulse Generator (MPG).

There are actually many other jogging options in Mach3. The two described here are generally suitable for turning. If you have MPGs  on your lathe you will need access to the full range of jogging options. These are displayed by pressing the *Tab* key. The screen that flies out is shown in figure 3.6. A second  *Tab* will dismiss the fly-out.
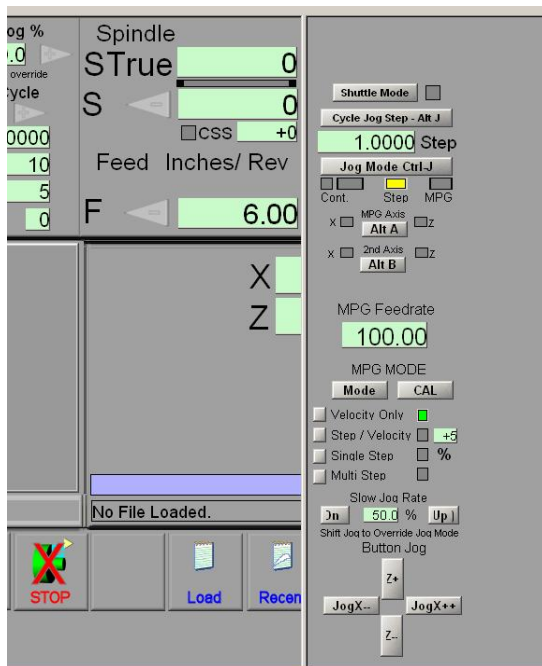


**Figure 3.6 – The Jogging fly-out screen**

## 3.4    Manual Data Input (MDI)

Go to the Manual screen.

This has a single line for data entry. You can click in it to select it or use press *Enter* which will automatically select it.

You can type any valid line that could appear in a part program and it will be executed when you press *Enter*. You can discard the line by pressing *Esc*. The *Backspace* key can be used for correcting mistakes in your typing.



**Figure 3.7 – The empty MDI line**

If you know some G-code commands then you could try them out. If not then try:

```
G00 X1.6 Z-2.3
```

Which will move the tool to coordinates X = 1.6 units and Z = -2.3 units. (it is G zero not G letter O). You will see the axis DROs move to the new coordinates.



**Figure 3.8 – Command typed in MDI line**

Try several different commands (or G00 to different places). If you use the up or down arrow keys while in the MDI line you will see that Mach3 scrolls you back and forwards through the history of commands you have used. This makes it easy to repeat a command without having to re-type it. When you select the MDI line you will have noticed a flyout box giving you a preview of this remembered text.

An MDI line (or block as a line of G-code is sometimes called) can have several commands on it and they will be executed in the "sensible" order as defined in Chapter 10 - not necessarily from left to right. For example setting a feed speed by something like F2.5 will take effect before any feed speed movements even if the F2.5 appears in the middle or even at the end of the line (block). If in doubt about the order that will be used then type several separate MDI commands in one by one.

## 3.5    Wizards – CAM without dedicated CAM software



**Figure 3.9 – Sample list of installed Wizards**

Mach3 allows the use of add-on screens which allow the automation of quite complex tasks by prompting the user to provide the relevant information. In this sense they are rather like the so-called Wizards in much Windows software that guide you through the information required for a task. The classic Windows Wizard will handle tasks line importing a file to a database or spreadsheet. In Mach3Turn, examples of Wizards include reducing the diameter of a length of bar, turning a taper, putting a radius on a bar end and threading. When you have gained experience with the system and understand G-code programs you can write your own Wizards.

For now it is easy to try one out. In the Wizards menu, choose Pick Wizard… A table of the Wizards installed on your system will be displayed (figure 3.9). As an example double-click on the line for Lathe Turning Wizards, which is in the standard Mach3 release, and/or single-click and then click *Run*.
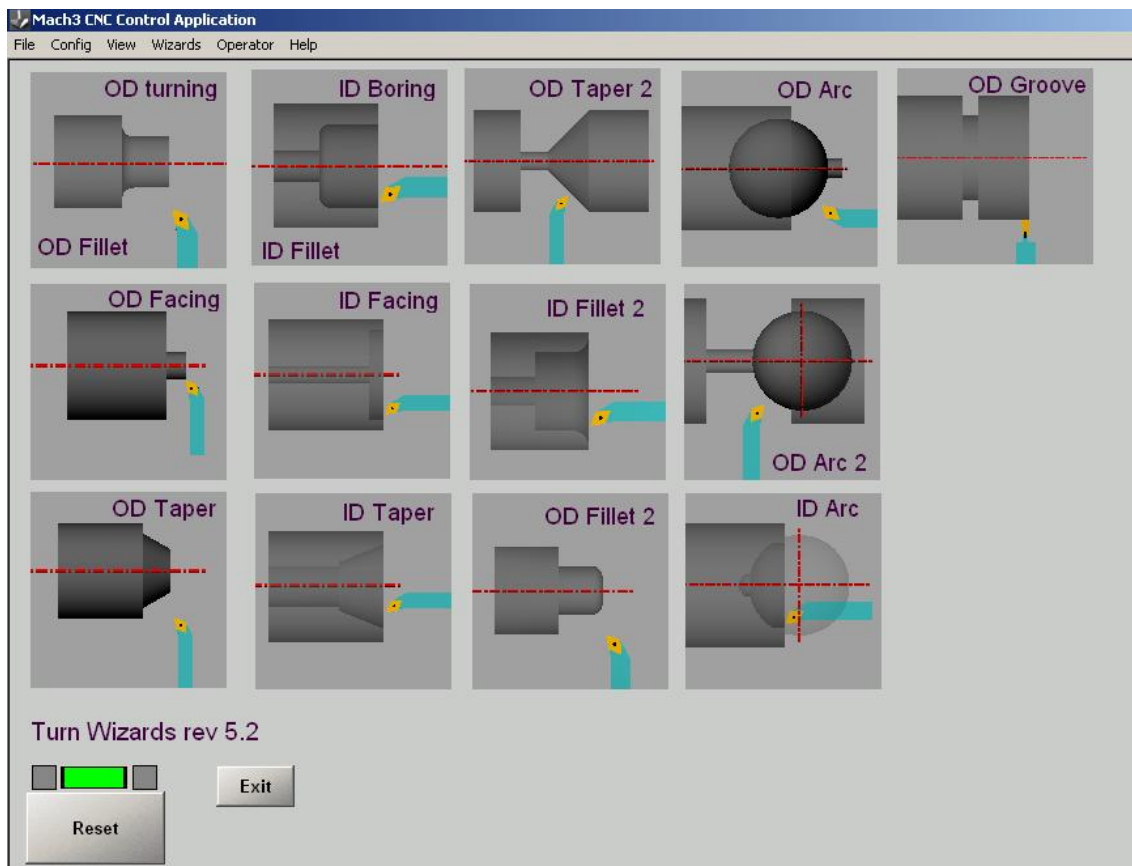


**Figure 3.10 – Menu of turning Wizards**

The Mach3 screen currently displayed will be replaced by the one shown in figure 3.10. Click on the OD Taper one as it is simple but quite interesting. You will now be shown a screen with some default values for the shape to be turned. Enter suitable values. The example in figure 3.11 is in metric units for radial programming and the feed is mm per minute. Remember that you must press *Enter* for each value typed into a DRO. Just clicking into another DRO discards the typed value.
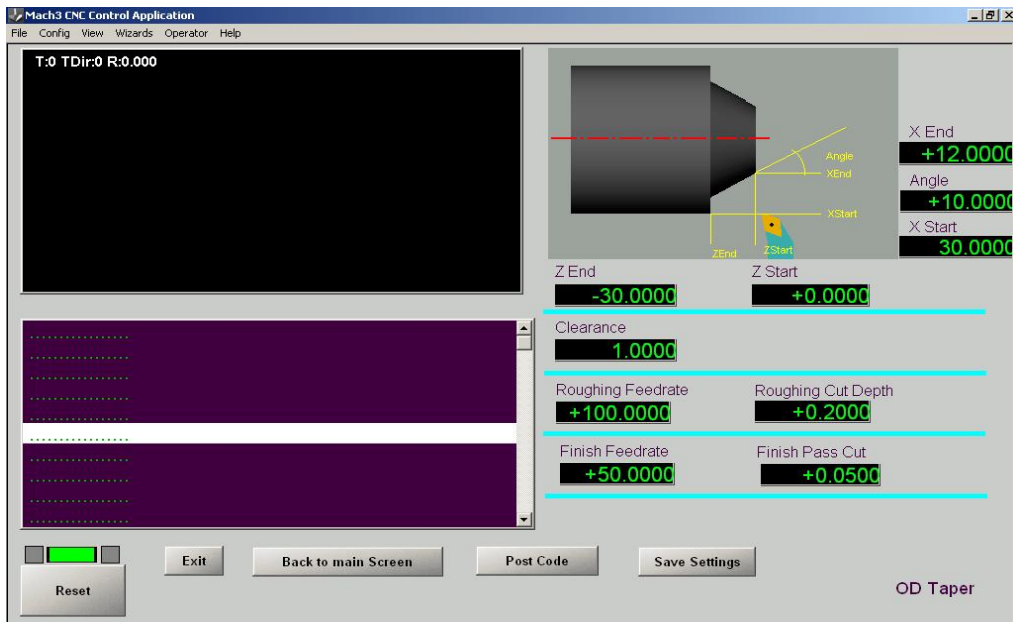
**Figure 3.11 – Parameters of taper entered**

When you are satisfied with the taper, click the *Post Code* button. This writes a G-code part program and loads it into Mach3. See figure 3.12. The toolpath display shows the cuts that will be made. You can revise your parameters to take smaller cuts or whatever and re-post the code.
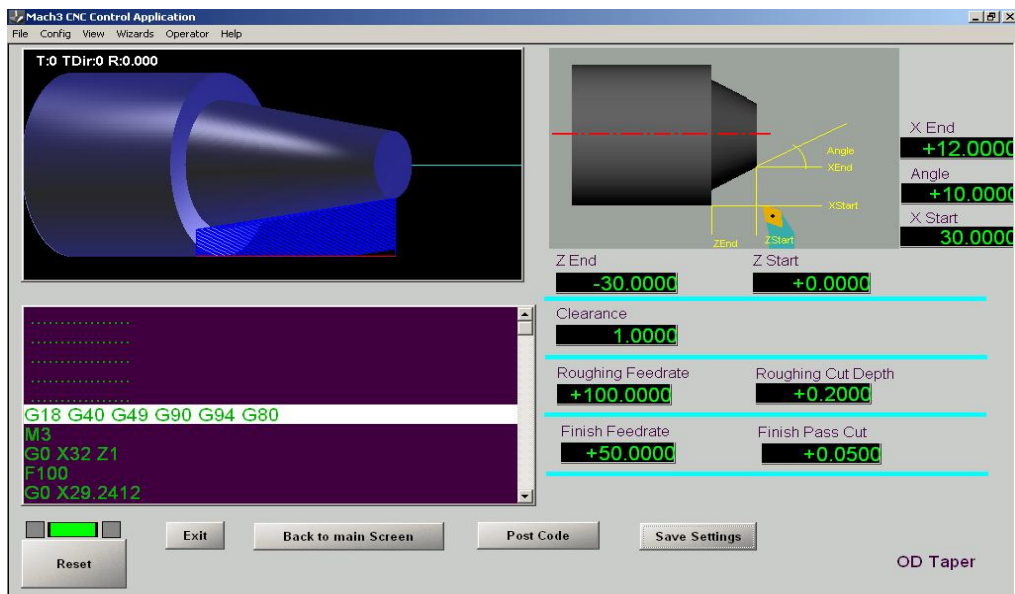


**Figure 3.12 – G-Code for taper posted**

If you wish, you can click *Save Settings* so the next time you run the Wizard the initial data will be what is currently defined.

When you click *Exit* you will be returned to the Mach3 screen that you came from. If this is not the Auto screen then use *Quit Mode* to return to the Welcome screen and click on *Auto*.

There are two screens used in Auto operation. This first one is the Auto Preparation screen. See figure 3.13. On this screen you can jog the tool, change the spindle speed and define the current position of the tool in the X or Z axis to be zero. Later when you have the lathe connected you would probably insert the bar to be turned and use jogging to face its end and then use the *Zero* button to set the tool Z position as zero. If you like you can pretend to do this now or alternatively click the *Cycle* button to move onto the Auto Cycle screen (figure 3.14).

The Auto Cycle screen allows you to run the part program produced by the Wizard.

The path that the tool will take is shown on the toolpath display. Different types of move (e.g. rapid moves, feedrate lines and feedrate arcs) are shown in different colors. This process will be often be quicker than reading the description here.
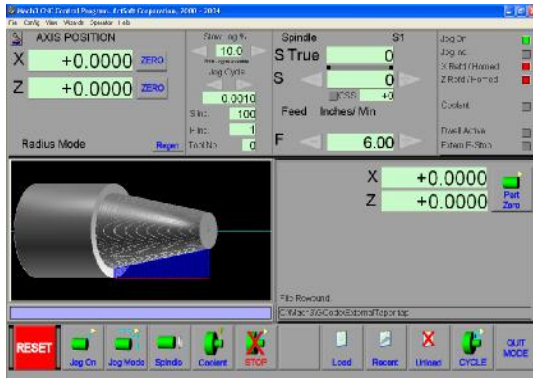


**Figure 3.13 – Auto Prep Screen**
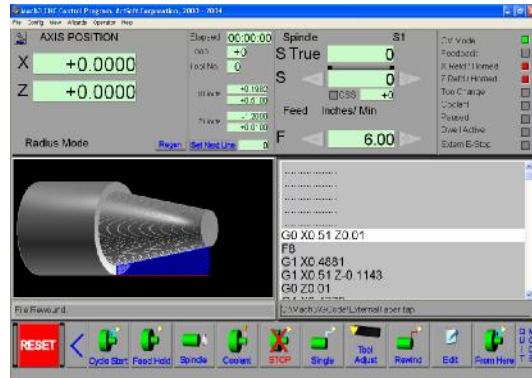


**Figure 3.14 – Auto Cycle Screen**

Now use Cycle Start to run the program. The DROs will show the movements that the tool will make on the actual machine and the line on the toolpath display is highlighted as the tool moves.

This is illustrated in figure 3.15. You can see that the toolpath has been zoomed to show extra detail and the next line of G-code top be executed is indicated by the highlight in the G-code window. When the program is running you can pause it with the *Feed Hold* button and make it run only one line at a time by selecting *Single* mode. All these functions are fully explained later.
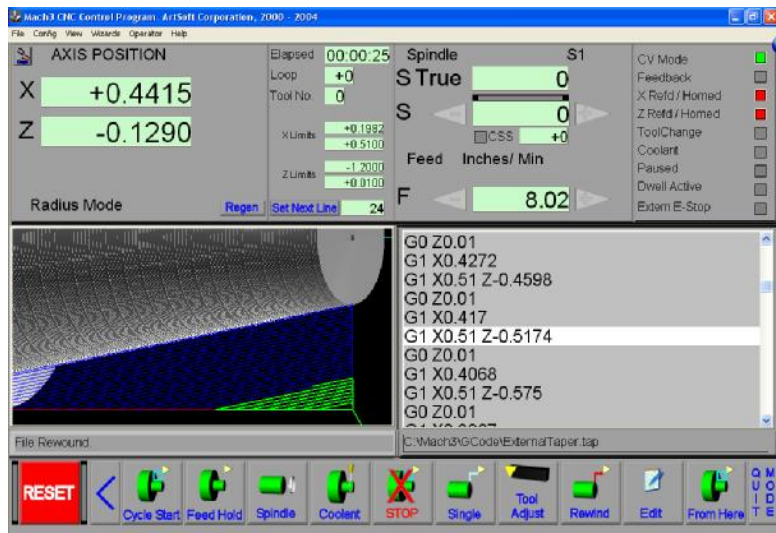


**Figure 3.15 – Wizard generated program part-way through its run**

## 3.6 Running a G-code program

Now it is time when you might like to try to input and edit a part program yourself. You will normally be able to edit programs without leaving Mach3 but, as we have not yet configured it to know which editor to use, it is easiest to set up the program outside Mach3.

Use Windows Notepad to enter the following lines into a text file and save it in a convenient folder (My Documents perhaps) as `TurnDown.tap`.

When doing the *Save As* for your file, you need to choose *Type of Save* as *All Files* to stop Notepad adding a .TXT your filename.

```
G20 G40 G49 G80 G90 G94(Imperial and standard modes)
G00 X0.3 Z0.05
F4
S1000 M0
G01 X0.28
G01 Z-1.2 (first cut)
G00 X0.3
G00 Z0.05
G01 X0.26
G01 Z-1.2 (second cut)
G00 X0.3
```

```
G00 Z0.05
M30 (rewind)
```

Again all the "0" are zeros in this. Don't forget to press the *Enter* key after the M30 line.

Now go to the Auto Preparation screen (using Quit mode and going from the Welcome screen if necessary). Use the Load button and the dialog that it displays to locate and select your file – figure 3.16
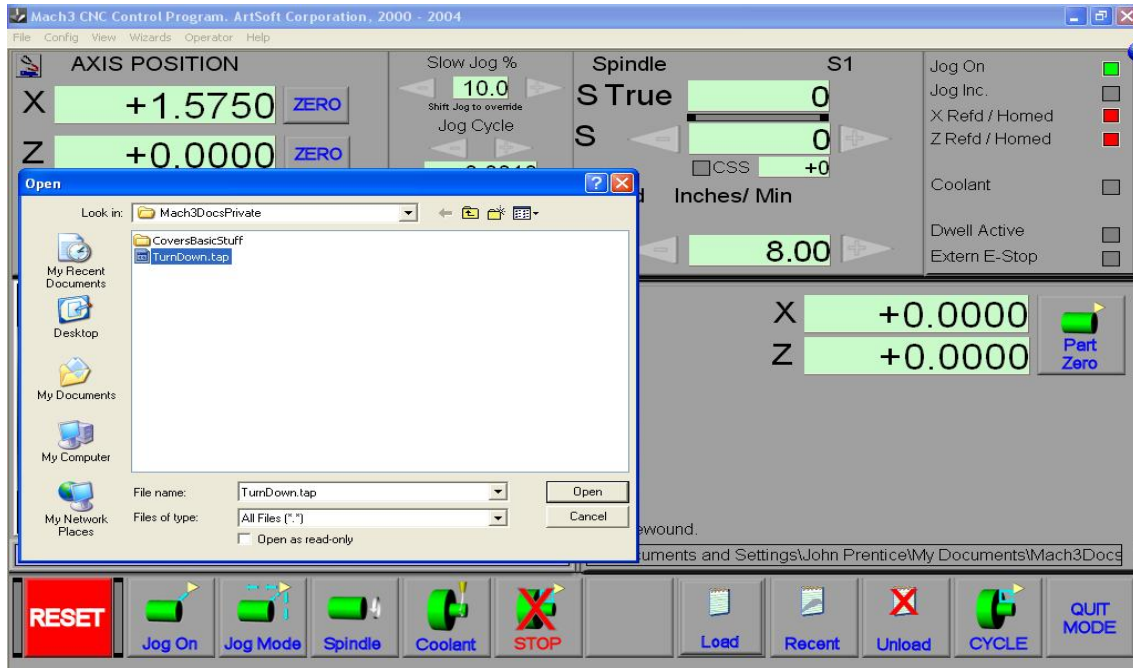


**Figure 3.16 – Loading G-code entered by hand**

The toolpath will show the simple cuts to be made - figure 3.17. You can now use the Auto Cycle screen to run your program

**Note:** You should always run your programs from a harddrive not a floppy drive or USB "key". Mach3 needs highspeed access to the file, which it maps into memory. The program file must not be read-only either.
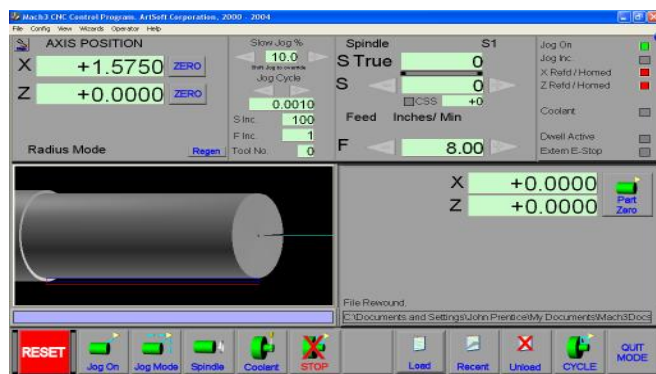


**Figure 3.17 – The TurnDown program loaded**

# 4.  Hardware issues and connecting the machine tool

> This chapter tells you about the hardware aspects of connections. Chapter 5 gives details of configuring Mach3 to use the connected items.
>
> If you have bought a machine that is already equipped to be run by Mach3 then you will probably not need to read this chapter (except out of general interest). Your supplier will have given you some documentation on how to connect the parts of your system together.
>
> Read this chapter to discover what Mach3 expects it is going to control and how you can connect up standard components like stepper motor drivers and micro-switches. We will assume that you can understand simple schematic circuit diagrams; if not, then now is the time to get some help.
>
> On the first reading you might not want to bother with sections after 4.6.

## 4.1  Safety - emphasised

Any machine tool is potentially dangerous. This manual tries to give you guidance on safety precautions and techniques but because we do not know the details of your machine or local conditions we can accept no responsibility for the performance of any machine or any damage or injury caused by its use. It is your responsibility to ensure that you understand the implications of what you design and build and to comply with any legislation and codes of practice applicable to your country or state.

**If you are in any doubt you must seek guidance from a professionally qualified expert rather than risk injury to yourself or to others.**

## 4.2  What Mach3 can control

Mach3 is a very flexible program designed to control a wide range of lathes and boring machines (and although not described here, milling type machines). The key characteristics of such machines controlled by Mach3 are:

Some user controls. An emergency stop (EStop) button **must** be provided on every machine

Two axes which are at right angles to each other (referred to as X and Z)

A tool which moves relative to a rotating workpiece. The origin of the axes is fixed in relation to the workpiece.

And optionally:

> Some switches to say when the tool is in the "Home" position
>
> Some switches to define the limits of permitted relative movement of the tool
>
> Controls for the speed and/or direction of the spindle..
>
> A mechanical means of changing the tool that is in use.
>
> A switch or switches which interlock the guards on the machine

Most connections between your machine and the PC running Mach3 will be made through the parallel (printer) port(s) of the computer. A simple machine will only need one port; a complex one will need two. Connections for signals which are not time critical can also be made through a serial interface using the ModBus protocol (such as the Homann ModIO board or a Programmable Logic Controller) or by a "keyboard emulator" which generates pseudo key presses in response to input signals from buttons

Mach3 can switch the spindle on rotating in either direction, and switch it off. It can also control the rate at which it rotates (rpm) and monitor its angular position for operations like cutting threads.

Mach3 can turn the two types of coolant on and off.

Mach3 will monitor the EStop circuits and can take note of the operation of the reference switches, the guard interlock and limit switches

Mach3 will store the properties of up to 256 different tools. If, however, your machine has an automatic tool changer or magazine then you will have to control it yourself.

## 4.3    The EStop control

Every machine tool must have one or more Emergency Stop (EStop) buttons; usually with a big red mushroom head. They must be fitted so that you can easily reach one from wherever you might be when you are operating the machine.

Each EStop button should stop all activity in the machine as quickly as is safely possible; the spindle should stop rotating and the axes should stop moving. This should happen **without** relying on generalpurpose software so we are usually talking about relays and contactors. The circuit should tell Mach3 what you have done and there is a special, mandatory input on the parallel port for this. It will generally not be good enough to turn off the AC power for an EStop because the energy stored in DC smoothing capacitors can allow motors to run on for some considerable time.

The machine should not be able to run again until a "reset" button has been pressed. If the EStop button locks, when pushed, then the machine should not immediately start when you release it by turning its head.

It will not generally be possible to continue machining a part after an EStop but you and the machine will at least be safe.

## 4.4    The PC parallel port

### 4.4.1   The parallel port and its history

When IBM designed the original PC (160k floppy disc drive, 64kbytes of RAM!) they provided an interface for connecting printers using a 25 conductor cable. This is the foundation of the parallel port we have on most PCs today. As it is a very simple way of transferring



**Figure 4.1 - Parallel port female connector (seen from back of PC)**

data it has been used for many things other than connecting printers. You can use it to transfer files between PCs, attach copy protection "dongles", connect peripherals like scanners and Zip drives and of course control machine tools.

USB is taking over many of these functions and this conveniently leaves the parallel port free for Mach3. Note, however, that a USB to printer convertor cable cannot be used with Mach3 to drive a machine tool.

The parallel port connector on the PC is a 25 way female "D" connector. Its sockets seen from the back of the PC are shown in figure 4.1. The arrows give the direction of information flow relative to the PC. Thus, for example, pin 15 is an input to the PC.

### 4.4.2    Logic signals

On first reading, you may wish to skip to the next heading and return here if you have to get involved with the nitty-gritty of interface circuits. It will probably be useful to read it with the documentation for your axis drive electronics.

All the signals output by Mach3, and input to it, are binary digital (i.e. zeros and ones) These signals are voltages supplied by the output pins or supplied to the input pins of the parallel port. These voltages are measured relative to the computer's 0 volt (ground) line (which is connected to pins 18 to 25 of the port connector).

The first widely used family (74xx series) of integrated circuits used TTL (transistor-transistor logic). In TTL circuits, any voltage between 0 and 0.8 volts is called "lo" and any voltage between 2.4 and 5 volts is called "hi". Connecting a negative voltage or anything above 5 volts to a TTL input will produce smoke.[1] The parallel port was originally built using TTL and, to this day, these voltages define its "lo" and "hi" signals. Notice that in the worst case there is only 1.6 volts difference between them.

It is, of course, arbitrary whether we say that a "lo" represents a logic one or a logic zero. However, as is explained below, "lo" = one is actually better in most practical interface circuits. This is often termed an "Active Low" signal.

For an output signal to do anything, some current will have to flow in the circuit connected to it. When it is "hi" current will flow **out** of the computer. When it is "lo" current will flow **into** the computer. The more current you have flowing in, the harder it is to keep the voltage near zero, so the nearer to the permitted maximum value of 0.8 volts "lo" will become. Similarly, current flowing out of a "hi" will make the voltage be lower and nearer to the 2.4 volts minimum limit. So with **too** much current the difference between "lo" and "hi" will be even less than 1.6 volts and things will become unreliable. Finally, it's worth noting you are allowed roughly 20 times more current flowing into a "lo" than you are allowed flowing out of a "hi".

So this means that it is best to assign logic 1 to be a "lo" signal. Fairly obviously this is called **active lo** logic. The main practical disadvantage of it is that the device connected to the parallel port has to have a 5 volt supply to it. This is sometimes taken from the PC game port socket , a USB socket or from a power supply in the device that is connected.

Turning to input signals, the computer will need to be supplied with some current (less than 40 microamps) for "hi" inputs and will supply some (less than 0.4 milliamps) for "lo" inputs.

Because modern computer motherboards combine many functions, including the parallel port, into one chip we have experienced systems where the voltages only just obey the "hi" and "lo" rules. You might find that a machine tool that ran on an old system becomes temperamental when you upgrade the computer. Pins 2 to 9 are likely to have similar properties (they are the data pins when printing). Pin 1 is also vital in printing but the other output pins are little used and may be less powerful in a "carefully optimised" design. A good isolating breakout board (see next section) will protect you from these electrical compatibility problems.

### 4.4.3    Electrical noise and expensive smoke

**Even if you skipped the previous section you had better read this one!**

You will see that pins 18 to 25 are connected to the 0 volt side of the computer's power supply. All signals inside and outside the PC are relative to this. If you connect many long wires to it, especially if they run near wires carrying high currents to motors, then these wires will have currents flowing in then that create voltages which are like noise and can cause errors. You might can even crash the computer.

The axis and perhaps spindle drives, which you will connect to Mach3 through your parallel port, are likely to work at between 30 and 240 volts and they will be able to supply currents

---

[1] Some people think that integrated circuits work in some way by using smoke. Certainly no one has ever seen one work after the smoke has escaped!

of many amps. Properly connected they will do no harm to the computer **but** an accidental short circuit could easily destroy the entire computer mother-board and even the CD-ROM and hard drives as well.
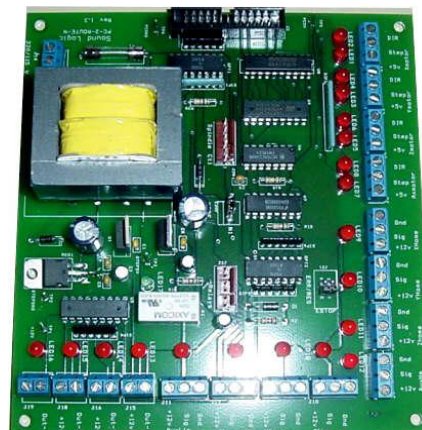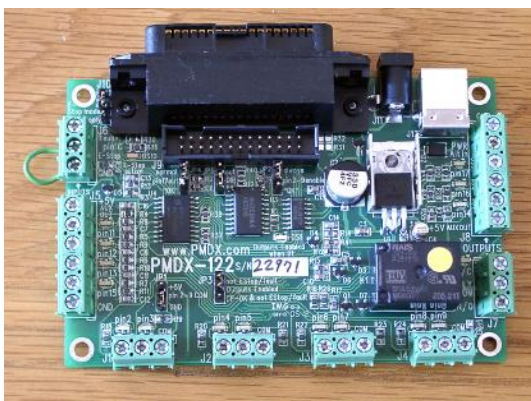


**Figure 4.2 – Two examples of commercially available breakout boards**

For these **two** reasons you are very strongly advised to buy a device called an "isolating breakout board". This will provide you with terminals that are easy to connect to, a separate 0 volt (common) for the drives, home switches etc. and will avoid exceeding the permitted current in and out of the port. This breakout board, your drive electronics and power supply should be neatly installed in a metal case to minimise the risk of interference to your neighbours' radio and television signals. If you build a "rat's nest" then you are inviting short circuits and tragedy. Figure 4.2 shows two commercial breakout boards.

Note that although the "D" connector on the PC is always a female, different breakout boards have different gender connectors so you need to use the cable that is supplied with the board or, if you are buying your own, check if you need a male-to-male or male-to-female cable.

*Here ends the sermon!*

## 4.5    Axis drive options

### 4.5.1    Steppers and Servos

There are two possible types of motive power for axis drives:

- Stepper motor
- Servo motor (either AC or DC)

Either of these types of motor can then drive the axes through leadscrews (plain- or ball-nut). The maximum speed of the motor will determine any gearing required between the motor and machine.

Properties of a bipolar stepper motor drive are:

1. Low cost
2. Simple 4-wire connection to motor
3. Low maintenance
4. Motor speed limited to about 1000 rpm and torque limited to about 3000 ounce inches. (21 Nm). Getting the maximum speed depends on running the motor or the drive electronics at their maximum permitted voltage. Getting the maximum torque depends on running the motor at its maximum permitted current (amps)
5. For practical purposes on a machine tool, steppers need to be driven by a chopped micro-stepping controller to ensure smooth operation at any speed with reasonable efficiency.

6. Steppers generally only provides open loop control which means it is possible to lose steps under high loading and this may not immediately be obvious to the machine user. This disadvantage can be overcome by special additional hardware.

On the other hand a servo motor drive is:

1. Relatively expensive (especially if it has a brushless motor)

2. Needs wiring for both the motor and encoder

3. Maintenance of brushes is required on DC motors

4. Motor speed in excess of 4000 rpm and a practically unlimited torque (if your budget can stand it!) but will need reduction gearing to the screws

5. Provides closed loop control so drive position is always known to be correct (or a fault condition will be raised)

**Figure 4.3 - Small DC servo motor with encoder (left) and gearbox**

In practice stepper motor drives will give satisfactory performance up to something like a 6" centre height lathe unless you want exceptional accuracy and speed of operation.

It is worth giving two warnings here:

Firstly, if you have a servo systems from a old machines it is probably not digital; in particular it will not be controlled by a series of step pulses and a direction signal. To use an old motor with Mach3 you will need to discard the resolver (which gave the position) and fit a quadrature encoder and you will have to replace **all** the electronics.

Secondly beware of secondhand stepper motors unless you can get manufacturer's data for them. They might be designed for 5-phase operation, may not work well with a modern chopped micro-stepping controller and might have a much lower rated torque than the same size of modern motor.. Unless you can test them, you may find that they have been accidentally demagnetised and so be useless. Unless you are really confident of your skills and experience, then the axis drives should be current products bought from suppliers who will support them. If you buy **right** then you will only need to buy **once**.

### 4.5.2  *Doing Axis drive calculations*

A full set of kinematic calculations for the X and Z axis drives would be very complicated and anyway you probably do not have all the necessary data (e.g. what is the maximum cutting force you want to use). Some calculation is, however, necessary for success and the basics will help you configure suitable hardware.

*If you are reading the manual for an overview then you might like to skip this section.*

Fuller details of the calculations are given in chapter 5.

**Example 1 - LATHE CROSS SLIDE (imperial units)**

We start with checking the minimum possible move distance. This is an absolute limit to the accuracy of work that can be done on the machine. We will then check rapid speeds and torque.

As an example suppose you are designing a 6" lathe cross-slide (X axis) drive. You are going to use a screw with a 0.1" pitch single start thread and a ball nut. You want to aim for a minimum move of 0.0001" (two tenths of a thou on the diameter). This is $^1/_{1000}$ of a revolution of the motor shaft if it is coupled directly to the screw.

### (a) Slide with stepper motor

The minimum step with a stepper motor depends on how it is controlled. There are usually 200 full steps per revolution. You need to use micro-stepping for smooth running over the full range of feed speeds and many controllers will allow you to have 10 micro-steps per full step. This system would give $^1/_{2000}$ of a revolution as the minimum step which is fine.

Next look at the possible rapid feed speed. Assume, conservatively, that the maximum motor speed is 500 rpm. This would give a rapid of 50 inches/minute or about 5 seconds for the full slide travel. This would be satisfactory.

At this speed the micro-stepping motor drive electronics need 16,666 (500 * 200 * 10 / 60) pulses per second. On a 1 GHz PC, Mach3 can comfortably generate 35,000 pulses per second simultaneously on each of the axes. So there are no problems here.

You now have to choose the torque that the machine will require. One way to measure this is to set up the machine for the heaviest cut you think you will ever make and, with a long lever (say 12") on the slide handwheel, turn it at the end with a spring balance (of set of spring kitchen scales). The torque for the cut (in ounce-inches) is the balance reading (in ounces) x 12. The other way is to use a motor size and specification that you know works on someone else's machine with the same type of slide and screw!

As the rapid feed speed was reasonable you could consider slowing it down by 2:1 gearing (perhaps by a toothed belt drive) which would nearly double the available torque on the screw and is an easier way of connecting the motor than by using a shaft coupling.

### (b) Slide with servo motor

Again we look at the size of one step. A servo motor has an encoder to tell its drive electronics where it is. This consists of a slotted disc and will generate four "quadrature" pulses for each slot in the disc. Thus a disc with 300 slots generates 300 cycles per revolution (CPR) This is fairly low for commercial encoders. The encoder electronics will output 1200 quadrature counts per revolution (QCPR) of the motor shaft.

The drive electronics for the servo will usually turn the motor by one quadrature count per input step pulse. Some high specification servo electronics can multiply and/or divide the step pulses by a constant (e.g. one step pulse moves by 5 quadrature pulses or even 36/17 pulses). This is often called **electronic gearing**.

As the maximum speed of a servo motor is around 4000 rpm we will certainly need a speed reduction on the mechanical drive. 5:1 would seem sensible. This gives a movement of 0.0000167" per step which is much better than that required (0.0001")

What maximum rapid speed will we get? With 35,000 step pulses per second we get 5.83 revolutions [35000/(1200 * 5)] of the leadscrew per second. This is OK at about 8 seconds for 4" travel of the slide. Notice, however, that the speed is limited by the pulse rate from Mach3 **not** the motor speed. This is actually only about 1750 rpm in the example. The limitation would be even worse if the encoder gave more pulses per revolution. It will often be necessary to use servo electronics with electronic gearing to overcome this limitation if you have high count encoders.

Finally one would check on available torque. On a servo motor less safety margin is required than with a stepper motor because the servo cannot suffer from "lost steps". If the torque required by the machine is too high then the motor may overheat or the drive electronics raise an over-current fault but you will not loose accuracy.

### Example 2 – LATHE LONG AXIS (metric units)

For the Z axis you will need a travel of 600 mm to 1000 mm depending on the length of the lathe bed..

We might choose a minimum step of 10 microns (0.01 mm) with a screw of pitch 5mm

Suppose we want to be able to cut a 2 mm pitch thread at 700 rpm spindle speed. This requires 2 ÷ 5 * 700 = 280 rpm at the Z axis screw. This would be OK with a stepper driving directly or even with a 2:1 reduction.

The torque calculation on this machine is more difficult than with the cross slide as, with the mass of the saddle to be moved, inertia, during acceleration and deceleration, is probably more important than the cutting forces. The experience of others or experiments will be the best guide. If you join the Mach3 support groups at www.machsupport.com then you will have access to the experience of hundreds of other users.

### 4.5.3   *How the Step and Dir signals work*

Mach3 puts one pulse (logic 1) on the Step output pin for each step that the axis is required to make. The Dir output will have been set before the step pulse appears to say what direction of move is required.

The logic waveform will be like that shown in figure 4.4. The gap between the pulses will be smaller the higher the speed of the steps.
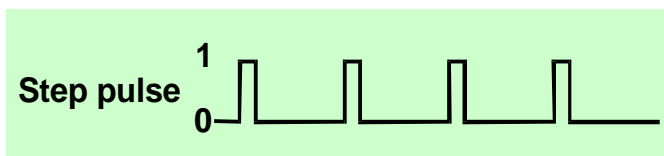


**Figure 4.4 - Step pulse waveform**

Stepper drive electronics usually use the Active Lo configuration for Step and Dir signals. Mach3 should be setup so these outputs are Active Lo. If this is not done then the Step signal still goes up and down but the drive thinks that the gaps between the pulses are the pulses and vice-versa and this often causes very rough or unreliable running of the motor. The "inverted" pulses are shown in figure 4.5.
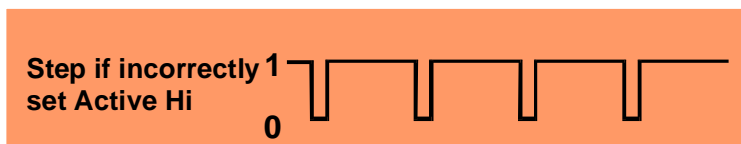


**Figure 4.5 - Wrongly configured output alters step waveform**

## 4.6     Limit and Home switches

### 4.6.1   *Strategies*

**Limit switches** are used to prevent the X or Z axis moving too far and so causing damage to the structure of the machine. Limit switches are less useful on a lathe than on a mill because there are more things to crash into – especially with the Z axis. For example, a limit that prevents a conventional turning tool hitting the chuck will not save a carelessly jogged boring bar. Similarly it is easy to hit the tailstock with both X-minus and Z-plus



**Figure 4.6 – Limit/Home microswitch mounted on the saddle is tripped by a plate (below). Other limit switch is above for X-minus movement**

movements. A switch is, however, always going to be useful at the X-plus limit.

If you have fast and/or powerful axis drives then you will probably want to install limit switches to protect the ballscrews and nuts at both ends of their travel.

An axis may also have a **Home switch**. Mach3 can be commanded to move one (or both) axes to the home position. The benefit of this is that the software has no way of knowing

where the tool is when the system is switched on. If you do not provide a home switch then you will have to jog the axes by eye to a reference position.

A home switch for the X axis is exceptionally useful. The reason for this is that, for a given tool, the X coordinate of the centre-line of the spindle and tailstock should always be zero. On a mill all the zeros depend on where you put the work on the table and how thick it is but the lathe centreline is always X = 0.

The home switch for an axis can be at any position and you define the coordinates of this location to Mach3. Thus the home switches do not have to be at **Machine Zero**.

As you will have seen, each axis **could** need three switches (i.e. limit switches at the two ends of travel and a home switch). So a basic lathe would require six parallel port inputs for them. This is not much good as a parallel port only has 5 inputs! This problem can be solved in three ways:

- The limit switches are connected to external logic (perhaps in the drive electronics) and this logic switches off the drives when the limit is reached. The separate reference switches are connected inputs to Mach3

- One pin can share all the inputs for an axis and Mach3 is responsible for controlling both limits and detecting home

- The switches can be interfaced by ModBus or a keyboard emulator.

The first method is best and mandatory for a very large, expensive or fast machine where you cannot trust the software and its configuration to prevent mechanical damage. Switches connected to the drive electronics can be intelligent and only allow motion away from a switch when the limit
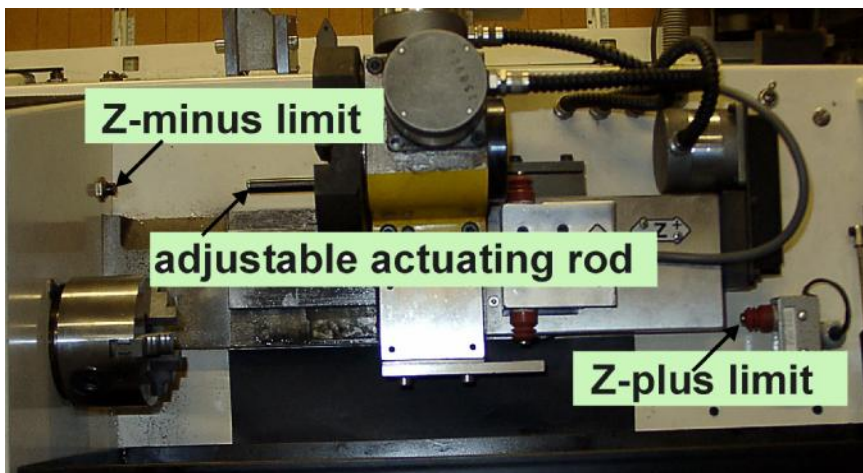


**Figure 4.7 – The lathe in fig 4.6 showing its Z limit switches**

is hit. This is safer than disabling the limits so a user can jog the machine off its limits but it does rely on having a sophisticated drive.

With the second method you only need two inputs to Mach3 and only two switches are required per axis as one limit and the home can share a switch. This is probably the best solution for the average small to medium lathe.

In the third solution you are constrained because serial interface has a much slower response time that the parallel port but it is satisfactory for limit switches on a machine without highspeed feeds. For details of the keyboard emulator and ModBus architecture see *Mach3 Customisation* wiki.

### 4.6.2 The switches

There are several choices you need to make when selecting switches:

(a) If you are going to have two switches
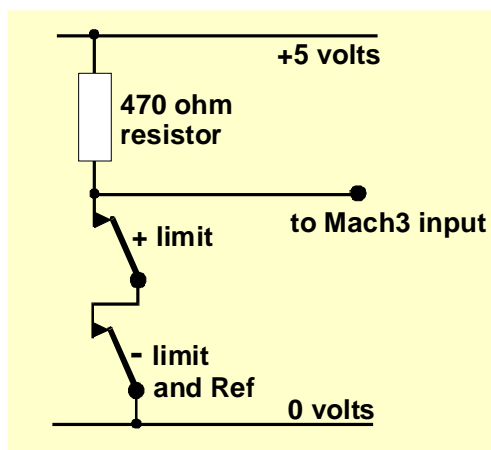


**Figure 4.8 - Two NC mechanical switches give logic OR**

sharing an input then they need to be connected so the signal is a logic "1" if **either** switch is operated (i.e. the logical OR function). This is easy with mechanical switches. If they have normally closed contacts and are wired in series as shown in figure 4.8, then they will give an Active Hi signal if either switch is operated. Note that for reliable operation you need to "pull up" the input to the parallel port. As mechanical switches can carry a significant current a value of 470R is shown which gives a current of about 10 milliamps. As the wiring to the switches might be quite long and liable to pickup of noise make sure that you have a good connection to the 0 volt side of your input (the frame of your machine tool will not be satisfactory) and consider using shielded cable with the shield connected to the main ground terminal of your controller.

If you use electronic switches like a slotted detector with a LED and photo-transistor, then you will need some sort of an OR-gate (which could be a "wired-or" if an Active Lo input is driven by open collector transistors).

(b) Optical switches, if out of the way of coolant, should be OK on a metalworking machine but are liable to malfunction with wood dust.

Don't use magnetic switches (reed switches or Hall effect devices) as the swarf will "fuzz-up" the magnet.

(c) The repeatability of the operating point, particularly with mechanical switches, is very dependent on the quality of the switch and the rigidity of its mounting and actuating lever. The setup in Figure 4.6 shows a high quality switch with a very rigid actuator. The repeatability is very important for a switch to be used for home as it will determine the accuracy parts that are turned.

(d) Overtravel is the movement of the switch that occurs after it has operated. With a limit switch it can be caused by the inertia of the drive. Choose a switch that will not be damaged by ¼" or more of overtravel.

### 4.6.3    *Where to mount the switches*

The choice of mounting position for switches is often a compromise between keeping them away from swarf and dust and having to use flexible rather than fixed wiring.

For example figure 4.6 shows switches mounted well away from the cutting tool position. The Z switches in figure 4.7 have fixed wiring and are out of the way of any coolant..

You might find it convenient to have only one moving cable with the wires in it for both axes (e.g. put both sets of switches on the saddle). Do not be tempted to share a multi-way cable between motor and switch wiring. You may want to run two separate cables together and this will not cause trouble if both a shielded (with braid or foil) and the shields are grounded to one common point at the electronic drives.

You might find it helpful to look at commercial machines and pictures of examples on the discussion forums for more ideas and techniques for switches.

### 4.6.4    *How Mach3 uses shared switches*

This section refers to the configuration for small machines where Mach3 rather than external EStop logic is controlled by the switches (option 2 above).

For a full understanding of this you will also have to read the section in chapter 5 on configuring Mach3, but the basic principle is easy. You connect the two limit switches to one input. You define, to Mach3, a direction as the direction to travel to move when looking for a home switch. The limit switch at that end of the axis is also used as the home switch.

In normal use when Mach3 is moving an axis and sees its limit input become active, it will stop running (like an EStop) and display that a limit switch has been tripped. You will be unable to move the axes unless:

1)  *Auto limit override* is switched on (by a toggle button on the Diagnostics screen). In this case you can click Reset and jog off the limit switch. You should then reference the machine

2) You click *Override limits* button on the Diagnostics screen. A red flashing LED warns you of the temporary override. This will again allow you Reset and to jog off the switch and will then turn itself and the flashing LED off. Again you should reference the machine. An input signal can also be defined to override the limit switches so you can wire a switch on your control panel to it.

Note that, although Mach3 uses limited jogging speed when limits are overridden, you will not be prevented from jogging further **onto** the switch and maybe crashing the axis into a mechanical stop. **Take great care**.

### 4.6.5 Referencing in action

When you request referencing (by a button or G-code) the axis (or axes) which have home switches defined will travel (at a selectable low speed) in the defined direction until the home switch operates. The axis will then move back in the other direction so as to be off the switch. During referencing the limits do not apply.

When you have referenced an axis, then zero (or some other value which is set up in the Config>Homing/Limits dialog) can be loaded into the axis DRO as the axis absolute machine coordinate. If you use a value of zero, then the home switch position is also the machine zero position of the axis. If the reference goes in the negative direction of an axis (usual for X and Y) the you might get referencing to load something like -0.25" into the DRO. This means that the home is quarter of an inch clear of the limit. This wastes a bit of the axis travel but if you overshoot, when jogging near home, then you will not accidentally trip the limits. See also Software Limits as another way of solving this problem.

If you happen to be already on a switch in a setup where both limits and home share an input and ask Mach3 to reference then it can travel in the opposite direction to usual (because it knows that you are already on the home switch) and will stop when you get off the switch. This is fine when you have a separate home switch or are on the limit at the **reference end of the axis**. If, however, you are on the other Limit switch (and Mach3 cannot know this as they are shared) then the axis moves for ever away from the actual home point until it crashes. So the advice is: **always jog carefully off the limit switches, then reference.**

### 4.6.6 Other Home and Limit options and hints

**Separate high accuracy home switch**

The X axis on a high precision machine might have a separate home switch to achieve the required accuracy. Diameters are usually the tightest toleranced dimensions on a part.

**Limit switches of multiple axes connected together**

Because Mach3 does not take any notice of which limit of which axis has tripped, then **all** the limits can be ORed together and fed into one limit input. Each axis can then have its own reference switch connected to the reference input. A two axis machine still only needs three inputs.

**Home switches of multiple axes connected together**

If you are **really** short of inputs to Mach3 then you can OR the home switches together and define all home inputs to be that signal. In this case you can only reference one axis at once – so you need to remove *REF All* buttons from your screens – and your home switches must all be at the same end of travel on their respective axes (e.g. maximum positive position).

## 4.7 Spindle control

There are three different ways in which Mach3 can control your "spindle" or you can ignore all of these and control it manually.

1. Relay/contactor control of motor On (Clockwise or Counterclockwise) and motor Off

2. Motor controlled by Step and Direction pulses (e.g. spindle motor is a servo)

3.  Motor controlled by a pulse width modulated signal

## 1. On/Off motor control

This method requires you to control your spindle speed by hand. This is a major disadvantage in a lathe where you may want a range of speeds depending on the diameter of the portion of your job that you are turning. If possible we advise you to interface Mach3 so it can control the spindle speed.

The M03 command and a screen button will request that the spindle starts in a clockwise direction. M04 will request that the spindle starts in an counterclockwise direction. M05 requests that the spindle stops. M03 and M04 can be configured to activate external output signals which can be associated with output pins on the parallel port(s). You then wire these outputs (probably via relays) to control the motor contactors for your machine.

Although this sounds straightforward, in practice **you need to be very careful**. Unless you really need to run the spindle "backwards" it would be better to treat M03 and M04 as the same or to allow M04 to activate a signal which you do not connect to anything.

The reason for this is that it is possible, in an error situation, for the clockwise and counterclockwise signals to be active together. This may cause the contactors to short the mains supply. Special mechanically interlocked reversing contactors can be obtained and if you are going to allow your spindle to run counterclockwise then you need to use one. Another difficulty is that in the "G-code" definition it says that it is legal to issue an M04 when the spindle is running clockwise under an M03 (and vice-versa). If your spindle drive is an AC motor, just changing the direction when running at full speed is going to impose very large forces on the mechanical drive of the machine and will probably blow the AC fuse or trip a circuit breaker. Mach3 does have a software facility to impose fixed delays in the output of these signals but you need to take care, especially with a powerful spindle drive.

See also the note about the limited number of Relay Activation Signals in the section on Coolant.

## 2. Step and Direction motor control

If your spindle motor is driven by an invertor with a Step and Direction input or by a servomotor with a step and direction drive (like the axis drives) then you can configure two output signals to control its speed and direction of rotation. Mach3 will take account of a variable step pulley drive or gearbox between the motor and the spindle. For full details see Motor Tuning in chapter 5

## 3. PWM motor control

As an alternative to Step and Direction control, Mach3 can output a pulse width modulated signal whose duty cycle is the percentage of full speed that you require. You could, for example, convert the duty cycle of the signal to a voltage ( PWM signal on for 0% of time gives 0 volts 50% gives 5 volts and 100% gives 10 volts) and use this to control an induction motor with a variable frequency inverter drive. Alternatively the PWM signal could be used to trigger a triac in a simple DC speed controller.



**Figure 4.9 – A 20% pulse width modulated signal**
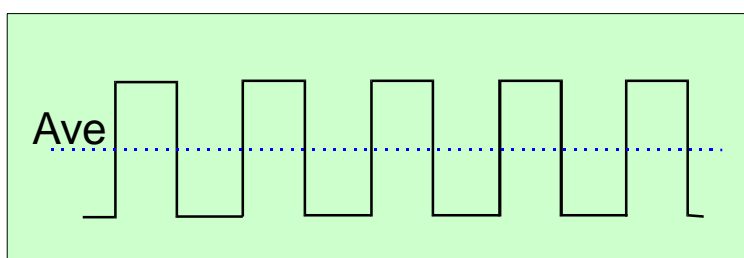


**Figure 4.10 – A 50% pulse width modulated signal**

Figures 4.9 and 4.10 show the pulse width at approximately 20% of

the cycle and 50% of the cycle.

In order for the PWM spindle speed signal to be turned into direct current the pulse signal it must transformed. In essence a circuit is used to find the average of the pulse width modulated signal. You need to take care with the electronics as the control inputs of most motor controllers are **not isolated from the mains**. You are strongly advised to use a commercial interface board (e.g. the Homann Digispeed or Campbell Spindle Motor Controller) to ensure safety.

The PWM signal is output on the spindle Step pin. You may need to take special precautions to switch off the motor at low speeds using the Motor Clockwise/Counterclockwise outputs.

**Note:** Many users have found that variable speed spindle drives are often a serious source of electrical noise which can cause problems with the machine axis drives, limit switch sensing etc. We strongly recommend you to use an optically isolated breakout board and take care to shield cables and run the power cables a few inches away from the control cables.

## 4.8    Spindle index pulse

When threading, it is essential for Mach3 to know the exact angular position of the spindle. This can be sensed in one of two ways which are similar to each other.

Figure 4.11 shows a spindle with a slotted disc on it. The slots are detected by a sensor consisting of an infra-red LED and a photo-transistor mounted so that the disc turns between them. They is on the aluminium block shown below the disc. Mach3 has an input for one or more pulses generated each revolution of the spindle.

It is also possible to use a reflective opto-sensor "looking at" white stripes on a blackened pulley or gear. This requires more adjustment when setting it up to get reliable results but may be easier to arrange.
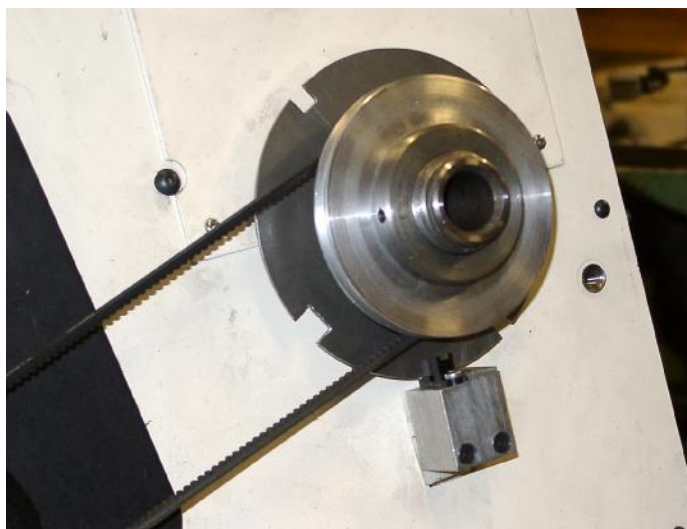


**Figure 4.11 – Spindle position sensor**

The disc shown in figure 4.11 has four slots in it. The one at the eight-o'clock position is wider than the others by at least 50%. The other slots are approximately evenly spaced around the disc. Precision is not required here as Mach3 calibrates itself and even discovers the number of slots (or stripes for a reflective sensor) that are in use.

However, spindles with a reasonable amount of inertia work fine with just one slot. Our advice is initially to use one slot or mark unless you already have a disc with more.

For reliable running Mach3 needs a pulse of at least 200 microseconds from the slots. You can calculate your slot angle or width as follows:

Suppose the maximum speed of your spindle is *N* (rpm).

The slot angle required *alpha* (degrees) is given by

$$alpha = 0.0012 \times N$$

If the diameter of the disc is *D* (inches or mm) then the slot width *W* (in same units) will be approximately

$$W = 0.0088 \times alpha \times D$$

So for example if maximum spindle speed is 3000 RPM and disc diameter is 110 mm then:

Alpha = 0.0012 x 3000 = $3.6^o$

Slot width = 0.0088 x 3.6 x 110 = 3.48 mm

Big slot needs to be 50% wider = 3.48 x 1.5 = 5.22 mm

The example in figure 4.11 has three slots at about 4 mm and one about 8 mm wide.

You will see that you cannot use a disc with many slots or holes in it or use a conventional encoder. The pulse would be too fast. The original index sensing arrangements can usually be simplified when retrofitting an old lathe . These often had multiple slots for speed sensing together with a single index slot.

As well as for timing threading operations, Mach3 also uses the index pulse to display the actual speed of the spindle and can implement a feed-per-revolution, rather than feed-per-minute, based on it.

## 4.9 Coolant

Output signals can be used to control valves or pumps for flood and mist coolant. These are activated by screen buttons and/or M07, M08, M09.

## 4.10 Manual Pulse Generators

It is often convenient to control a CNC lathe by hand. Mach3 has powerful jogging controls but the most intuitive manual control can be obtained using handwheels connected to encoders (MPGs). Figure 4.12 shows a prototype control panel that is so equipped. Traditional MPGs include detents to give a click action as they are turned. For manual turning some operators prefer the feel of the machine if these are disabled and the control has uniform friction.
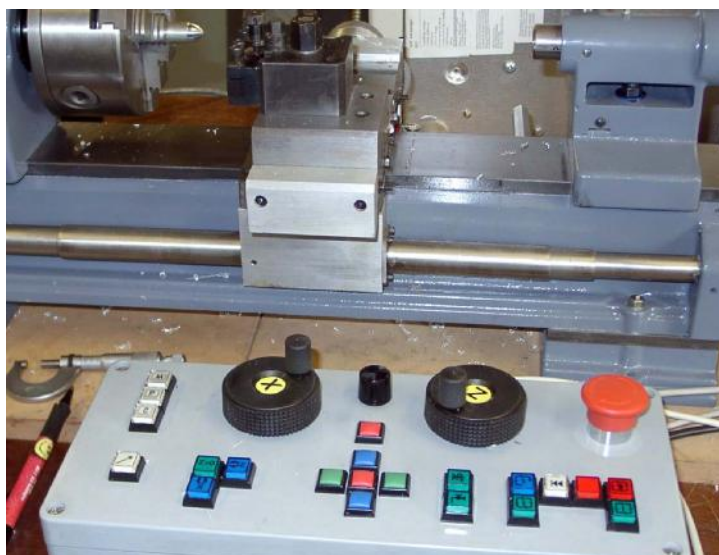


**Figure 4.12 – Bench lathe with control panel including two MPGs**

## 4.11 Charge pump - a pulse monitor

Mach3 will output a constant pulse train whose frequency is approximately 12.5 kHZ on one or both of the parallel ports whenever it is running correctly. This signal will not be there if the Mach3 has not been loaded, is in EStop mode or if the pulse train generator fails in some way. This function is often implemented in commercial breakout boards. You can use this signal to charge a capacitor through a diode pump (hence the name) whose output, showing Mach3's health, enables your axis and spindle drives etc.

## 4.12 Other functions

Mach3 has sixteen OEM Trigger input signals which you can assign for your own use. For example they can be tested in user written macros.

The first three of these inputs can be emulated by keystrokes from the PC keyboard port instead of being physically wired to a parallel port. Full details of the architecture of Input Emulation are given in *Mach3 Customisation* wiki. The setup dialog is defined in section 5.

Input #1 can be used to inhibit running of the part program. It might be connected to the guards on your machine.

The six Relay Activation outputs have already been mentioned under Spindle and Coolant. Any spares can be used by you and controlled in user written macros.

**And a final thought -** before you get carried away with implementing too many of the features in this chapter, remember that you do not have an unlimited number of inputs/outputs. Even with two parallel ports there are only ten inputs for supporting all functions and, although a ModBus device will help giving more inputs and outputs, these cannot be used for all functions. Full details are given in chapter 5.

# 5. Configuring Mach3 for your machine and drives

> If you have bought a machine tool with a computer running Mach3 then you
> will probably not need to read this chapter (except out of general interest).
> Your supplier will probably have installed the Mach3 software and set it up
> and/or will have given you detailed instructions on what to do.
>
> You are recommended to ensure that you have a paper copy of how Mach3 is
> configured should you ever need to re-install the software from scratch.
> Mach3 stores this information in an XML file which you can view.

## 5.1 A configuration strategy

This chapter contains a lot of very fine detail. You should, however, find that the
configuration process is straightforward if you take it step-by-step, testing as you go. A
good strategy is to skim through the chapter and then work with it on your computer and
machine tool. We will assume that you have already installed Mach3 for the dry running
described in chapter 3.

Virtually all the work you will do in this chapter is based on dialog boxes reached from the
Config(ure) menu. These are identified by, for example, Config>Logic which means that
you choose the Logic entry from the Config menu.

## 5.2 Defining the native setup units

The first thing to decide is whether you wish to
define the properties of your machine in Metric
(millimetres) or Inch units. You will be able to
run part programs in either units whichever you
choose. The maths for configuration will be
slightly easier if you choose the same system as
ballscrew is made in. So a screw with 0.2" lead (5
tpi) is easier to configure in inches than in
millimetres. Similarly a 2mm lead screw will be
easier in millimetres. The multiplication and/or
division by 25.4 is not difficult but is just
something else to think about.

There is, on the other hand, a slight advantage in
having the setup units be the units in which you
usually work. This advantage is that you can lock
the DROs to display in this system whatever the part program is doing (i.e. switching units
by G20 and G21).

**Figure 5.1 - Native Units dialog**

So the choice is yours. Use Config>Setup Units to choose MMs or Inches (see figure 5.1).
Once you have made a choice you must not change it without going back over all the
following steps or total confusion will reign!

## 5.3 Initial configuration of engine and ports

The first dialog to use is Config>Ports and Pins. This dialog has many tabs but the initial
one is as shown in figure 5.2.

If you are only going to use one parallel port and it is the one on your computer's
motherboard then the default address of Port 1 of 0x378 (i.e. Hexadecimal 378) is almost
certainly correct.

If you are using one or more PCI add-on cards then you will need to discover the address
each responds to as there seem to be no standards!  Run the Windows Control Panel from
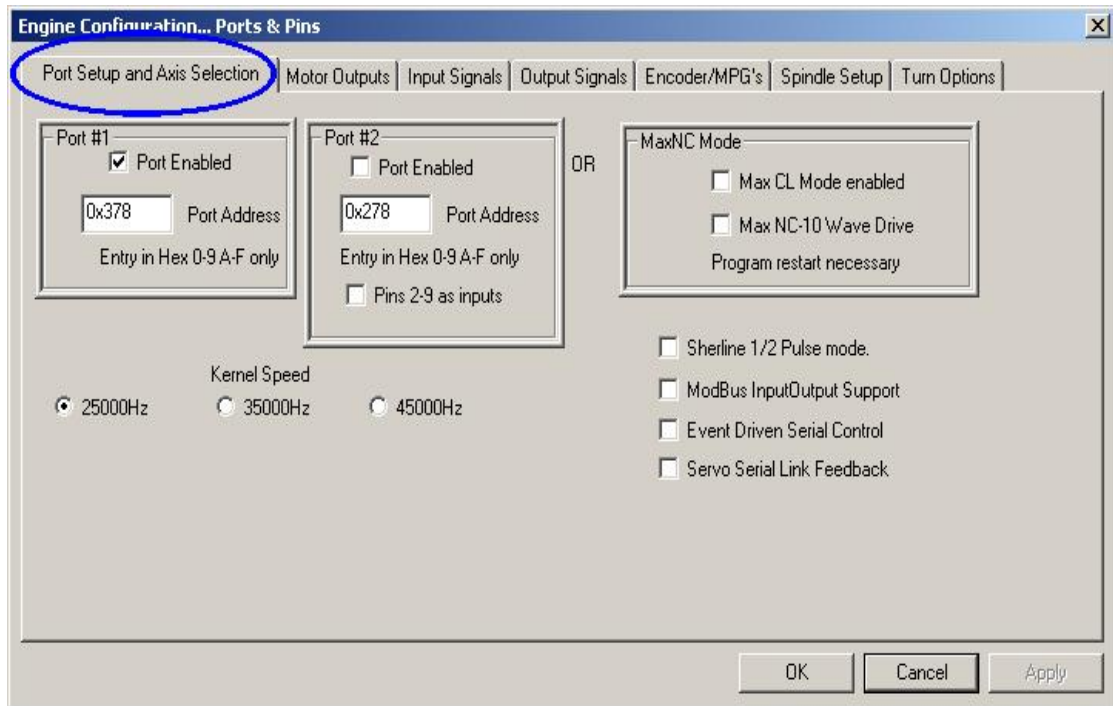
**Figure 5.2 - Ports and Axis selection tab**

the Windows *Start* button. Double click on *System* and choose the *Hardware* tab. Click the *Device Manager* button. Expand the tree for the item "Ports (COM & LPT)".

Double click the first LPT or ECP port. Its properties will be displayed in a new window. (see figure 5.3) Choose the Resources tab. The first number in the first IO range line is the address to use. Note the value down and close the Properties dialog.

**Note:** that installing or removing any PCI card can change the address of a PCI parallel port card even if you have not touched it.

If you are going to use a second port repeat the above paragraph for it.

Close the Device Manager, System Properties and Control Panel windows.

Enter your first port's address (do not provide 0x prefix to say it is Hexadecimal as Mach3 assumes this)

If necessary check Enabled for port 2 and enter its address. If you require a lot of input signals on your machine the you can check *Pins 2-9 as Inputs* and get 13 inputs on port 2.

You should now choose the Kernel speed. If you are using the demonstration version (i.e. have not yet purchased and installed your license) the you can only use 25000 Hz. The kernel speed affects the maximum pulse rate that
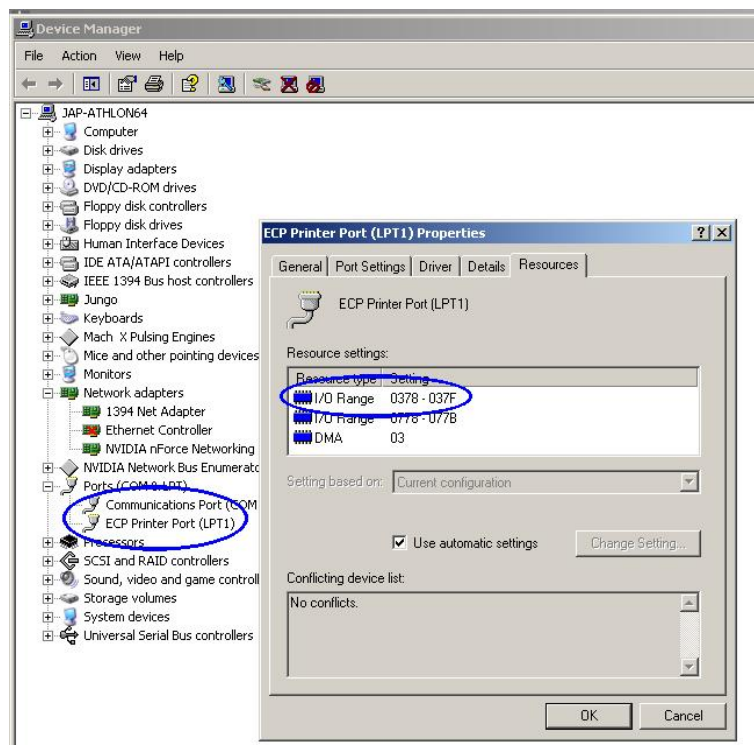


**Figure 5.3 – Finding PCI port address**

Mach3 can output. Unless you are using servo drives on your axes or spindle and they have high resolution encoders (more than 200 CPR) you can safely leave the speed at 25000 Hz. This will maximise the CPU time available for other functions like updating the toolpth display.

**Warning:** If you change the kernel speed then you **must** retune all you axes and spindle or you risk motors stalling and/or the computer locking up.

Unless you have special hardware leave the Sherline ½ Pulse Mode, ModBus Input/Output, Event Driver Serial control, Servo Serial Link Feedback and MAX NC mode boxes unchecked.

Now click the *Apply* button to save these values. This is most important. **Mach3 will not remember values when you change from tab to tab or close the Port & Pins dialog unless you *Apply*.**

## 5.4 Defining input and output signals that you will use

Now that you have established the basic configuration it is time to define which input and output signals you will be going to use and which parallel port and pin will be used for each. The documentation for your breakout board may give guidance on what outputs to use if it has been designed for use with Mach3 or the board may be supplied with a skeleton Profile (.XML) file with these connections already defined.

### 5.4.1 Axis and Spindle output signals to be used

First view the *Motor Outputs* tab. This will look like figure 5.4.

Define where the drives for your X and Z axes are connected and click to get a check-mark



| Signal | Enabled | Step Pin# | Dir Pin# | Dir LowActive | Step Low Ac... | Step Port | Dir Port |
|--------|---------|-----------|----------|---------------|----------------|-----------|----------|
| X Axis | ✓ | 2 | 3 | ✓ | ✓ | 1 | 1 |
| Y Axis | ✗ | 0 | 0 | ✗ | ✗ | 0 | 0 |
| Z Axis | ✓ | 6 | 7 | ✗ | ✓ | 1 | 1 |
| A Axis | ✗ | 0 | 0 | ✗ | ✗ | 0 | 0 |
| B Axis | ✗ | 0 | 0 | ✗ | ✗ | 0 | 0 |
| C Axis | ✗ | 0 | 0 | ✗ | ✗ | 0 | 0 |
| Spindle | ✓ | 1 | 12 | ✗ | ✗ | 0 | 0 |
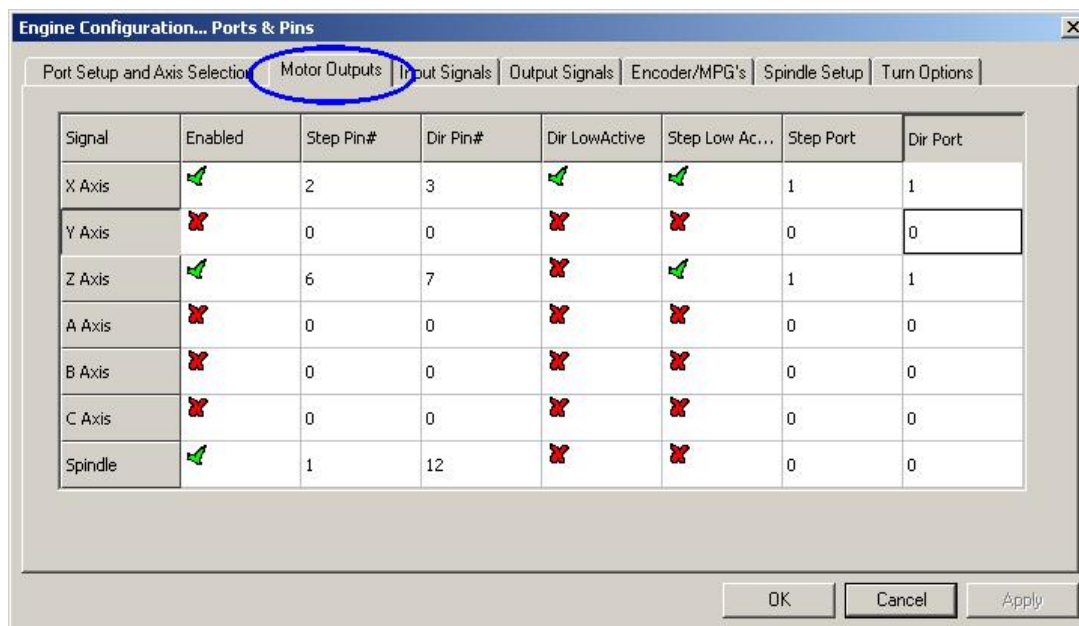
**Figure 5.4 – Defining the connections for axes and the controlled spindle**

Enable these axes. If your interface hardware (e.g. Gecko 201 stepper driver) requires an active-lo signal ensure that these are checked for the Step and Dir(ection) signals.

You are unlikely to have Y, A, B or C axes when turning so leave these disabled (with a red cross).

If your spindle speed will be controlled by hand then you have finished this tab. **Click the *Apply* button to save the data on this tab.**

If your spindle speed will be controlled by Mach3 then you need to Enable the spindle and allocated a Step pin/port for it if it uses pulse width modulated control with relays to control its direction or to allocate Step and Direction pins/ports if it has full control. You should also define if these signals are active-lo. When done, **click the *Apply* button to save the data on this tab.**

### 5.4.2    Input signals to be used

Now select the *Input Signals* tab. This will look like figure 5.5.

We assume that you have chosen one of the limit and home switch strategies from chapter 4.6.
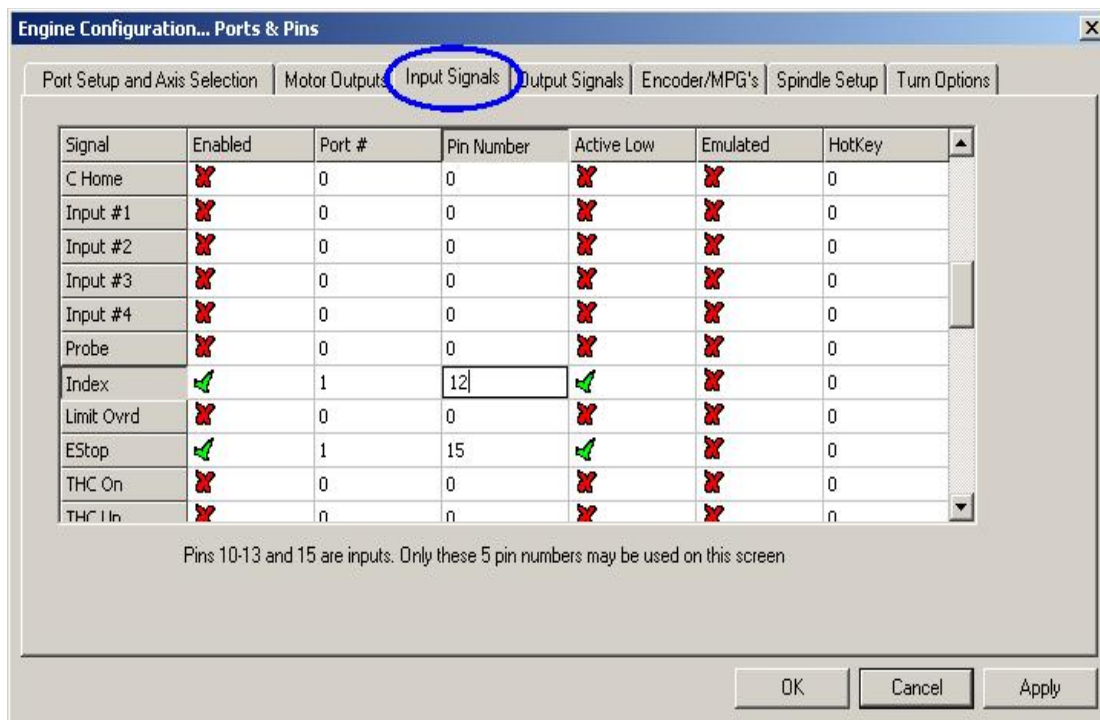


**Figure 5.5 – Input signals**

If you have used strategy one and the limit switches are connected together and trigger an EStop or disable the axis drives via the drive electronics then you do not check any of the Limit inputs.

With strategy two you will probably have home switches on the X and Z axes. Enable the *Home* switches boxes for these axes and define the Port/Pin to which each is connected. If you are combining limits and the home switch then you should enable the *Limit --*, the *Limit ++* and *Home* for each axis and allocate the same pin to *Home*, *Limit—* and *Limit++*.

As with axis drives you will leave Y, A, B and C switch inputs disabled. Notice the scroll bar to access the rest of the table which is not visible in figure 5.5.

The *Input #1* is special in that it can be used to inhibit running a part program when safety guards are not in place. The other three (and #1 if not used for the guard interlock) are available for your own use and can be tested in the code of macros. The *Input #4* can be used to connect an external pushbutton switch to implement the Single Step function. You may wish to configure them later.

Enable and define *Index Pulse* if you have a spindle sensor with just one slot or mark.

Enable and define *Limits Override* if you are letting Mach3 control your limit switches and you have an external button which you will press when you need to jog off a limit. If you have no switch then you can use a screen button to achieve the same function.

Enable and define *EStop* to indicate to Mach3 that the user has demanded an emergency stop.

Enable and define *OEM Trigger* inputs if you want electrical signals to be able to call OEM button functions without a screen button needing to be provided.

Enable and define *Timing* if you have a spindle sensor with **more than one** slot or mark.

Probe, THCOn, THCUp and THCDown do not apply to Mach3Turn.

If you have one parallel port then you have 5 available inputs; with two ports there are 10 (or with pins 2 to 9 defined as inputs, 13). It is very common to find that you are short of

input signals especially if you are also going to have some inputs MPGs. You may have to compromise by not having things like a physical Limit Override switch to save signals!

You can also consider using a ModBus device for some input signals.

**Click the *Apply* button to save the data on this tab.**

### 5.4.3  *Emulated input signals*

If you check the Emulated column for an input then the Port/Pin number and active-lo state for that signal will be ignored but the entry in the Hotkey column will be interpreted. When a key-down message is received with code that matches a Hotkey value then that signal is considered to be active. When a key-up message is received then it is inactive.

The key-up and key-down signals usually come from a keyboard emulator (like the Ultimarc IPAC or Hagstrom) which is triggered by switches connected to its inputs. This allows more switches to be sensed than spare pins on your parallel ports but there may be significant time delays before the switch change is seen and indeed a key-up or key-down message can get lost by Windows.

Emulated signals cannot be used for *Index* or *Timing* and should not be used for *EStop*.

### 5.4.4  *Output Signals*

Use the Output signals tab to define the outputs you require. See figure 5.6.
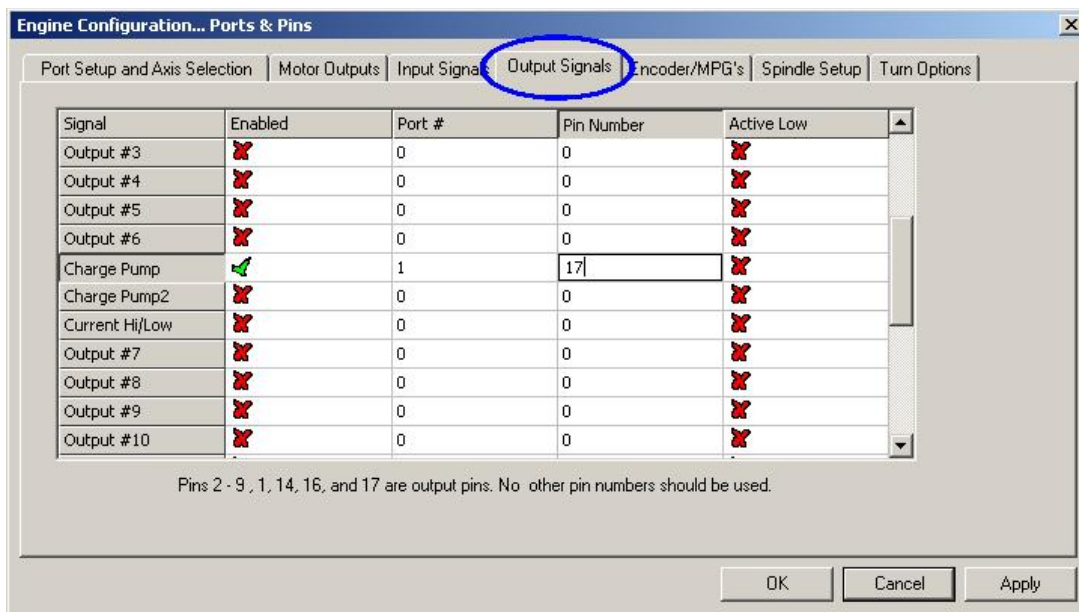


**Figure 5.6 – Output signals**

You will probably only want to use one *Enable* output (as all the axis drives can be connected to it). Indeed if you are using the charge pump/pulse monitor feature then your breakout board may enable your axis drives from its output.

The *Output#* signals are for use to control a stop/start spindle (clockwise and optionally counterclockwise), the Flood and Mist coolant pumps or valves and for control by your own customized Mach3 buttons or macros.

The *Charge* Pump line should be enabled and defined if your breakout board accept this pulse input to continually confirm correct operation of Mach3. *Charge Pump2* is used if you have a second breakout board connected to the second port  or want to verify the operation of the second port itself.

**Click the *Apply* button to save the data on this tab.**

## 5.4.5  Defining MPG and encoder inputs
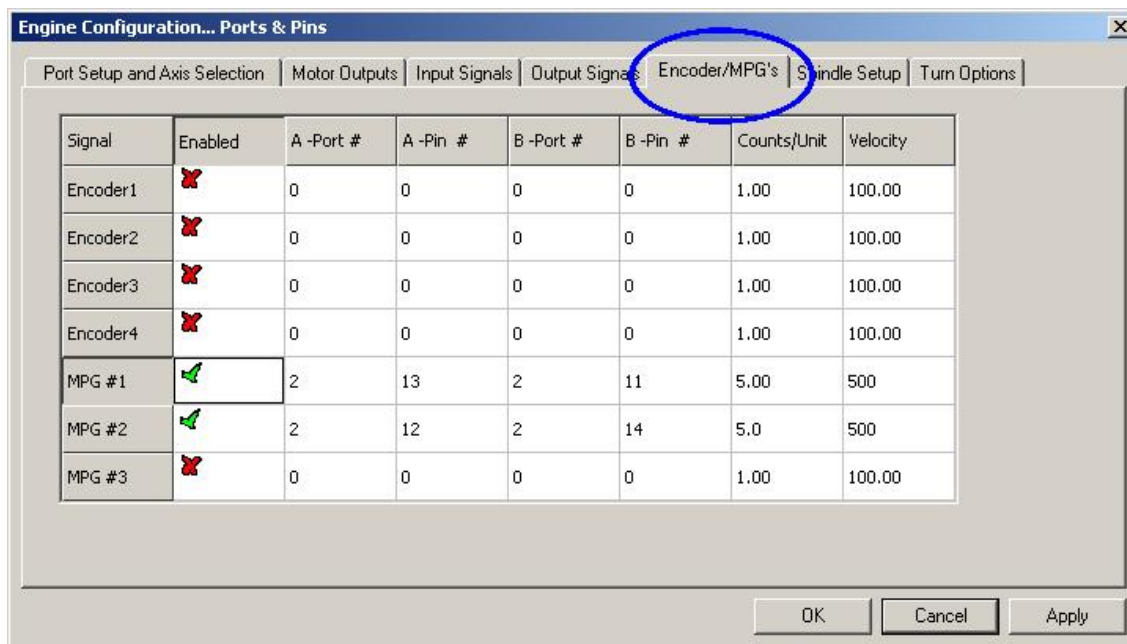


**Figure 5.5 – Encoder inputs**

The Encoder/MPGs tab is used to define the connections and the resolution of linear encoders or Manual Pulse Generators (MPGs) used for jogging the axes. It is covered here for completeness of the description of Config>Ports & Pins.

This dialog does not need an active-lo column as, if the encoders count the wrong way it is merely necessary to swap the pins allocated for A and B inputs.

### 5.4.5.1  Encoders

The *Counts per unit* value should be set to correspond to the resolution of the encoder. Thus a linear scale with rulings at 20 microns produces a count every 5 microns (remember the quadrature signal), that is 200 counts per unit (millimetre). If you have Native units set as inches the it would be 200 x 25.4 = 5080 counts per unit (inch). The *Velocity* value is not used.

### 5.4.5.2  MPGs

The *Counts per unit* value is used to define the number of quadrature counts that need to be generated for Mach3 to see movement of the MPG. For a 100 CPR encoder, a figure of 4 is suitable. For higher resolutions you should increase this figure to get the mechanical sensitivity you want. We find 100 works well with 1024 CPR encoders.

The *Velocity* value determines the scaling of pulses sent to the axis being controlled by the MPG. The lower the value given in *Velocity* the faster the axis will move. Its value is best set by experiment to give a comfortable speed when spinning the MPG as fast as is comfortable.

## 5.4.6  Configuring the spindle

The next tab on Config>Ports & Pins is *Spindle Setup*. This is used to define the way in which your spindle and coolant is to be controlled. You may opt to allow Mach3 to do nothing with it, to turn the spindle on and off or to have total control of its speed by using a Pulse Width Modulated (PWM) signal or a step and direction signal. The dialog is shown in figure 5.6.

### 5.4.6.1  Coolant control

Code M7 can turn Flood coolant on, M8 can turn Mist coolant on and M9 can turn all coolant off. The Flood Mist control section of the dialog defines which of the output signals
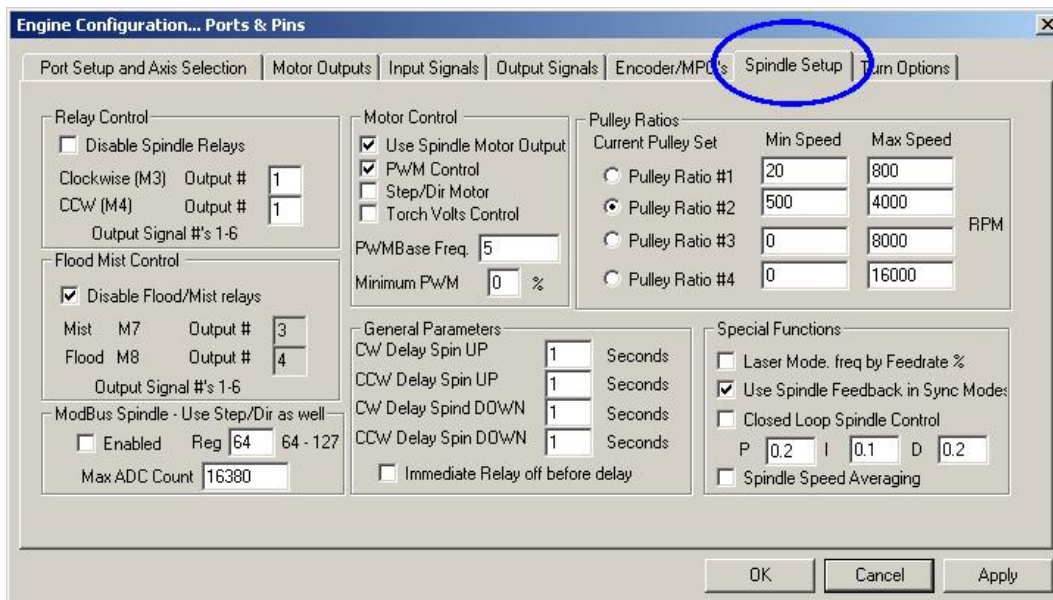
**Figure 5.6 – Spindle setup**

are to be used to implement these functions. The Port/Pins for the outputs have already been defined on the *Output Signals* tab.

If you do not want to use this function check *Disable Flood/Mist Relays*.

### 5.4.6.2 Spindle relay control

If the spindle speed is controlled by hand or by using a PWM signal then Mach3 can define its direction and when to start and stop it (in response to M03, M04 and M05) by using two outputs. The Port/Pins for the outputs have already been defined on the *Output Signals* tab.

If you control the spindle by Step and Direction then you do not need these controls. M03, M04 and M05 will control the pulse train generated automatically.

If you do not want to use this function check *Disable Spindle Relays*.

### 5.4.6.3 Motor Control

Check *Use Motor Control* if you want to use PWM or Step and Direction control of the spindle. When this is checked then you can choose between *PWM Control* and *Step/Dir Motor*.

**PWM Control**

A PWM signal is a digital signal, a "square" wave where the percentage of the time the signal is high specifies the percentage of the full speed of the motor at which it should run.

So, suppose you have a motor and PWM drive with maximum speed of 3000 rpm then figure 4.9 would run the motor at 3000 x 0.2 = 600 RPM. Similarly the signal in figure 4.10 would run it at 1500 RPM.

Mach3 has to make a trade off in how many different widths of pulse it can produce against how high a frequency the square wave can be. If the frequency is 5 Hz the Mach3 running with a 25000 Hz kernel speed can output 5000 different speeds. Moving to 10Hz reduces this to 2500 different speeds but this still amounts to a resolution of one or two RPM.

A low frequency of square wave increases the time that it will take for the motor drive to notice that a speed change has been requested. Between 5 and 10 Hz gives a good compromise. The chosen frequency is entered in the *PWMBase Freq* box.

Many drives and motors have a minimum speed. Typically because the cooling fan if very inefficient at low speeds whereas high torque and current might still be demanded. The *Minimum PWM %* box allows you to set the percentage of maximum speed at which Mach3 will stop outputting the PWM signal.

You should be aware that the PWM drive electronics may also have a minimum speed setting and that Mach3 pulley configuration (see section x.x) allows you to set minimum speeds. Typically you should aim to set the pulley limit slightly higher than the *Minimum PWM %* or hardware limit as this will clip the speed and/or give a sensible error message rather than just stopping it.

**Step and Direction motor**

This may be an variable speed drive controlled by step pulses or a full servo drive.

You can use the Mach3 pulley configuration (see section ??.??) to define a minimum speed if this is needed by the motor or its electronics.

Note that you do not **need** a servo spindle drive to implement any of the Mach3Turn functions. Indeed threading can be performed quite successfully even with manual control of spindle speed.

#### 5.4.6.4 General Parameters

These allow you to control the delay after starting or stopping the spindle before Mach3 will execute further commands (i.e. a Dwell). These delays can be used to allow time for acceleration before a cut is made and to provide some software protection from going directly from clockwise to counterclockwise. The dwell times are entered in seconds.

#### 5.4.6.5 Pulley ratios

Mach3 has control over the speed of your spindle motor. You program spindle speeds through the S word. The Mach3 pulley system allows you to define the relationship between these for four different pulley or gearbox settings. It is easier to understand how it works after tuning your spindle motor so it is described in section ??.??, below.

#### 5.4.6.6 Special function

*Laser mode* should always be unchecked.

*Use Spindle feedback in sync mode* should always be checked.

*Closed Loop Spindle Control*, when checked, implements a software servo loop which tries to match the actual spindle speed seen by the Index or Timing sensor with that demanded by the S word. The exact speed of the spindle is not likely to be important so you are not likely to need to use this feature in Mach3Turn.

If you do use it then the P, I and D variables should be set in the range 0 to 1. P controls the gain of the loop and an excessive value will make the speed oscillate, or hunt, around the requested value rather than settling on it. The D variable applies damping so stabilising these oscillations by using the derivative (rate of change) of the speed. The I variable takes a long term view of the difference between actual and requested speed and so increases the accuracy in the steady state. Tuning these values is assisted by using the dialog opened by Operator>Calibrate spindle.

*Spindle Speed Averaging,* when checked, causes Mach3 to average the time between index/timing pulses over several revolutions when it is deriving the actual spindle speed. You might find it useful with a very low inertia spindle drive or one where the control tends to give short-term variations of speed.

### 5.4.7 Turn Options tab

The final tab on Config>Ports & Pins is *Turn Options.* See figure 5.7.

#### 5.4.7.1 X Mode

The *X Mode* checkbox determines whether X coordinates are to be interpreted as the radius of the part being machined or its diameter.

You should decide which mode best suits your method of working. Do not be tempted to chop and change between modes. In particular you must use the mode which corresponds to the code output by the post processor of your CAM system.

Notice because of the way arcs are defined in G-code, using the wrong mode does not merely produce a part that is twice or half the expected size.

In diameter mode the X axis DRO displays diameter values and the X word in G-code is interpreted as a diameter. Offsets in the tool table, for fixtures and the positions of limit switches are not, however, multiplied by 2; they are actual distances.
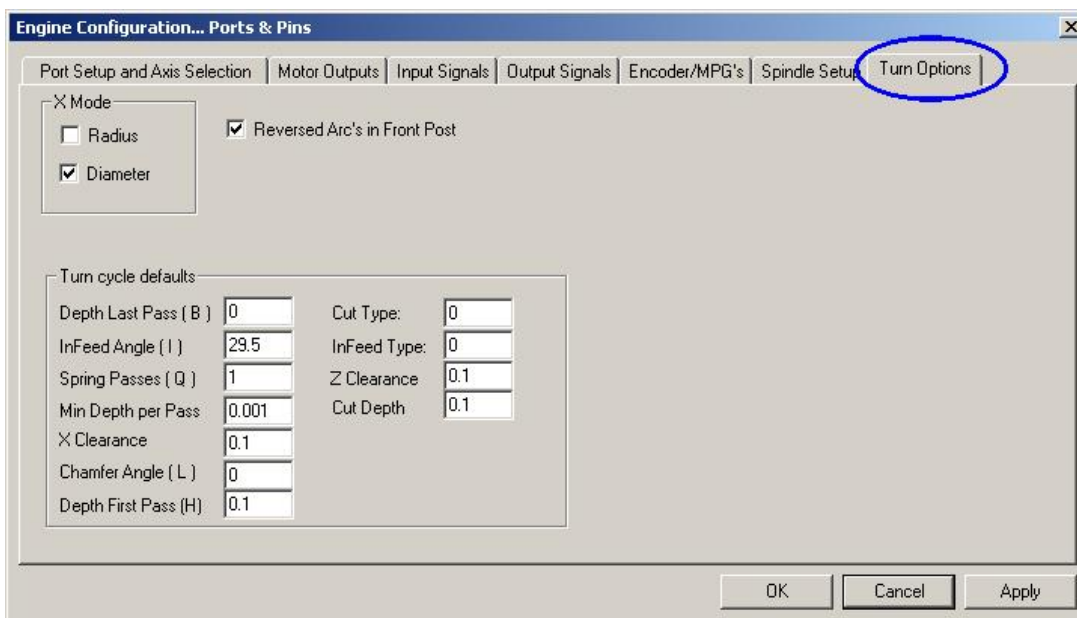


**Figure 5.7 – Turn Options Tab**

### 5.4.7.2   Threading defaults

Most users will not need to be concerned with these as the CAM post processor or Mach3 Wizard will generally provide the system with all the required information to define a thread. These default values are, however, the ones that will be used in a given word is omitted in a G76 (threading) canned cycle.

## 5.5   Testing

Your software is now configured sufficiently for you to do some simple tests with the hardware. If it is convenient to connect up the inputs from the limit and home switches then do so now.

Run Mach3Turn and display the Diagnostics screen (figure 5.8). This has a bank of LEDs displaying the logic level of the inputs and outputs. Ensure that the external Emergengy Stop signal is not active (Red *Emergency* LED not flashing) and press the red *Reset* button on the screen. Its LED should stop flashing.

Operate the home or the limit switches. You should see the appropriate LEDs glow yellow when their signal is active.

Next, if you have associated any Relay outputs with coolant then you can click in the MDI window and use G-code like M07, M09 etc. to turn the outputs on and off. The appropriate red output LEDs should light. The machine should also respond or you can monitor the voltages of the signals at the breakout board with a multimeter.

**Warning:** As you have not yet completely set up the spindle parameters for a PWM or Step and Direction drive it is unwise to turn it on yet.

These tests will let you see that your parallel port is correctly addressed and the inputs and outputs are appropriately connected.

If you have two ports and all the test signals are on one then you might consider a temporary modification to your configuration so that one of the home or limit switches is connected via the second port so that you can check its correct operation. Don't forget the *Apply* button when doing this sort of testing. If all is well then you should restore the proper configuration.
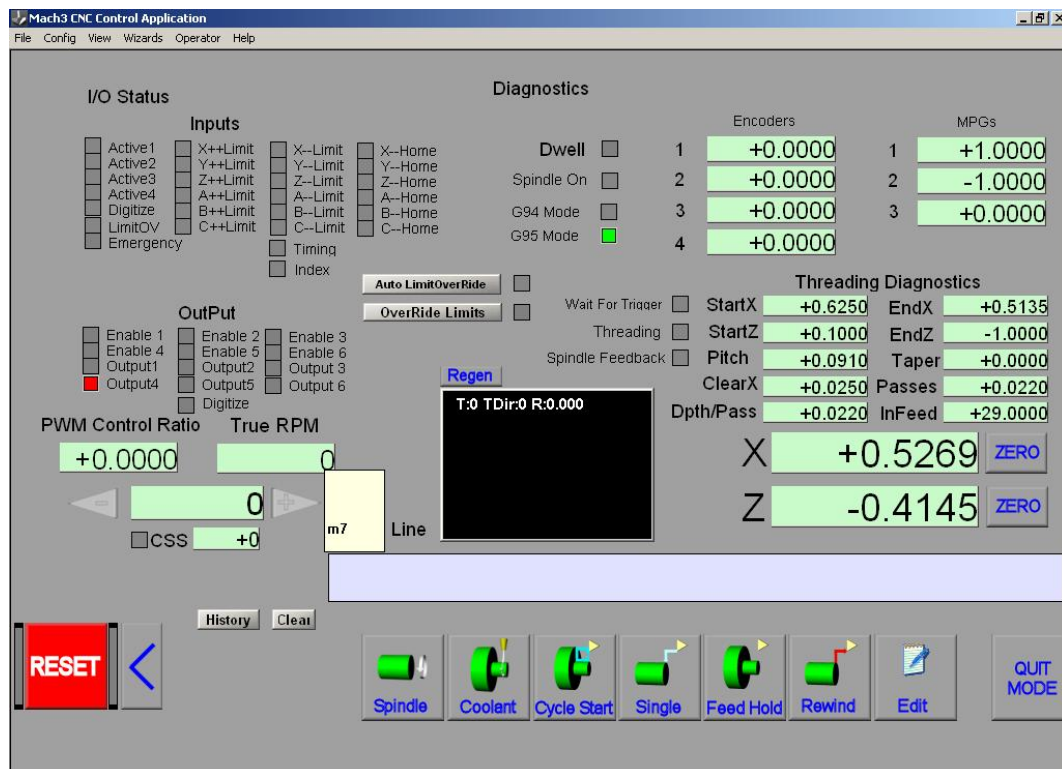


**Figure 5.8 – Testing Inputs and Outputs**

If you have problems you should sort them out now as this will be much easier than when you start trying to drive the axes. If you do not have a multimeter then you will have to buy or borrow a logic probe or a D25 adaptor (with actual LEDs) which let you monitor the state of its pins. In essence you need to discover if (a) the signals in and out of the computer are incorrect (i.e. Mach3 is not doing what you want or expect) or (b) the signals are not getting between the D25 connector and your machine tool (i.e. a wiring or configuration problem with the breakout board or machine). 15 minutes help from a friend can work wonders in this situation even if you only carefully explain to him/her what your problem is and how you have already looked for it! You will be amazed how often this sort of explanation suddenly stops with words like:

 "…… Oh! I see what the problem must be, it's ….."

## 5.6   Tuning motors

Well after all that detail it's now time to get things moving - literally! This section describes setting up your axis drives and, if its speed will be controlled by Mach3, the spindle drive.

The overall strategy for each axis is: (a) to calculate how many step pulses must be sent to the drive for each unit (inch or mm) of movement of the tool or table, (b) to establish the maximum speed for the motor and (c) to set the required acceleration/deceleration rate.

We advise you to deal with one axis at a time. It will be convenient to do the Z axis first as this is easiest to use for checking your calculations. You also might wish to try running the motor before it is mechanically connected to the machine tool.

So now connect up the power to one of your axis driver electronics and double check the wiring between the driver electronics and your breakout board/computer. You are about to mix high power and computing so it is better to be safe than smoky!

### 5.6.1 *Calculating the steps per unit*

The number of steps Mach3 must send for one unit of movement depends on the mechanical drive (e.g. pitch of ballscrew, gearing between the motor and the screw), the properties of the stepper motor or the encoder on the servo motor and the micro-stepping or electronic gearing in the drive electronics. We look at these three points in turn then bring them together.

#### 5.6.1.1 Calculating mechanical drive

You are going to calculate the number of revolutions of the motor shaft (***motor revs per unit***) to move the axis by one unit. This will probably be greater than one for inches and less than one for millimetres but this makes no difference to the calculation which is easiest done on a calculator anyway.

For a **screw and nut** you need the raw pitch of the screw (i.e. thread crest to crest distance) and the number of starts. Inch screws may be specified in threads per inch (tpi). The pitch is 1/tpi (e.g. the pitch of an 8 tpi single start screw is $1 \div 8 = 0.125"$)

If the screw is multiple start multiply the raw pitch by the number of starts to get the effective pitch. The ***effective screw pitch*** is therefore the distance the axis moves for one **revolution of the screw.**

Now you can calculate the ***screw revs per unit***

   *screw revs per unit = 1 ÷ effective screw pitch*

If the screw is directly driven from the motor then this is the motor revs per unit. If the motor has a gear, chain or belt drive to the screw with $N_m$ teeth on the motor gear and $N_s$ teeth on the screw gear then:

   *motor revs per unit = screw revs per unit* x $N_s \div N_m$

For example, suppose our 8 tpi screw is connected to the motor with a toothed belt with a 48 tooth pulley on the screw and an 16 tooth pulley on the motor then the motor shaft pitch would be $8$ x $48 \div 16 = 24$ (**Hint**: keep all the figures on your calculator at each stage of calculation to avoid rounding errors)

As a metric example, suppose a two start screw has 5 millimetres between thread crests (i.e. effective pitch is 10 millimetres) and it is connected to the motor with 24 tooth pulley on the motor shaft and a 48 tooth pulley on the screw. So the *screw revs per unit* = 0.1 and motor revs per unit would be $0.1$ x $48 \div 24 = 0.2$

#### 5.6.1.2 Calculating motor steps per revolution

The basic resolution of all modern stepper motors is 200 steps per revolution (i.e. $1.8^o$ per step). Note some older steppers are 150 steps per rev.

The basic resolution of a servo motor depends on the encoder on its shaft. The encoder resolution is usually quoted in CPR (cycles per revolution) Because the output is actually from two quadrature signals the effective resolution will be **four** time this value. You would expect a CPR in the range of about 125 to 2000 corresponding to 500 to 8000 steps per revolution.

#### 5.6.1.3 Calculating Mach3 steps per motor revolution

We very strongly recommend that you use micro-stepping drive electronics for stepper motors. If you do not do this and use a full- or half-step drive then you will need much larger motors and will suffer from resonances that limit performance at some speeds.

Some micro-stepping drives have a fixed number of micro-steps (typically 10) while others can be configured. In this case you will find 10 to be a good compromise value to choose. This means that Mach3 will need to send 2000 pulses per revolution for a stepper axis drive.

Some servo drives require one pulse per quadrature count from the motor encoder (thus giving 1200 steps per rev for a 300 CPR encoder. Others include electronic gearing where

you can multiply the
input steps by an
integer value and,
sometimes, the
divide the result by
another integer
value. The
multiplication of
input steps can be
very useful with
Mach3 as the speed
of small servo
motors with a high
resolution encoder
can be limited by the
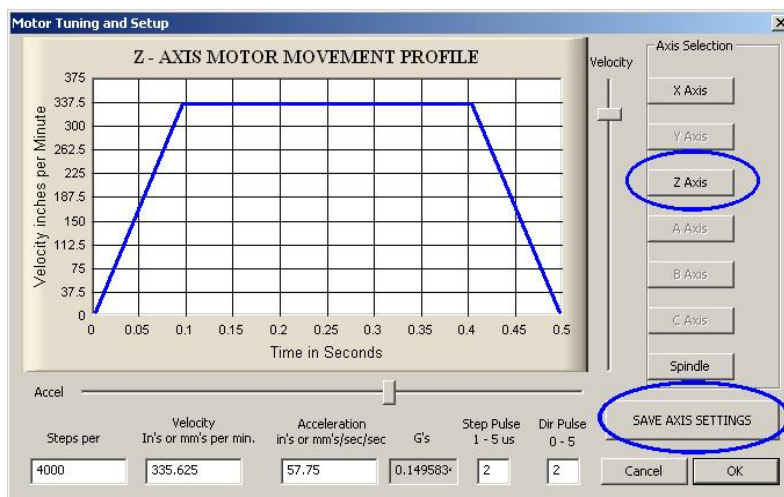maximum pulse rate
which Mach3 can
generate.


**Figure 5.9 – Motor Tuning dialog**

#### 5.6.1.4 Mach3 steps per unit

So now we can finally calculate:

*Mach3 steps per unit*  =  *Mach3 steps per rev* x *Motor revs per unit*

Figure 5.9 shows the dialog for Config>Motor Tuning. Click a button to select the axis
which you are configuring and enter the calculated value of Mach3 *Steps per unit* in the box
above the *Save* button.. This value does not have to be an integer so you can achieve as
much accuracy as you wish. **To avoid forgetting later click** *Save Axis Tuning* **now.**

### 5.6.2 Setting the maximum motor speed

Still using the Config>Motor Tuning dialog, as you move the Velocity slider you will see a
graph of velocity against time for a short  imaginary move. The axis accelerates, maybe has
a flat top running at full speed and then decelerates. Set the velocity slider to maximum for
now. Use the Acceleration slider to alter the rate of acceleration/deceleration (these are
always the same as each other)

As you use the sliders the values in the *Velocity* and *Accel* boxes are updated. *Velocity* is in
units per second. *Accel* is in units per second[2] The maximum velocity you can display will
be limited by the maximum pulse rate of Mach3. Suppose you have configured this to
35,000 Hz and 400 steps per unit then the maximum possible *Velocity*  is 35000/400 = 87.5.

This maximum is, however, not necessarily safe for your motor, drive mechanism or
machine; it is just Mach3 running "flat out". You can make the necessary calculations or do
some practical trials. Let's just try it out first.

#### 5.6.2.1 Practical trials of motor speed

You saved the axis after setting the Steps per unit. OK the dialog and make sure that
everything is powered up. Click the Reset button so its LED stops flashing.

Go back to Config>Motor Tuning and select your axis. Use the Velocity slider to have the
graph about 20% of maximum velocity. Press the cursor Up key on your keyboard. The axis
should move in the Plus direction. If it runs away then choose a lower velocity. If it crawls
then choose a higher velocity. The cursor Down key will make it run the other way (i.e. the
Minus direction).

If the direction is wrong then, Save the axis and **either** (a) change the Low Active setting
for the Dir pin of the axis in Config>Ports and Pins>Motor Outputs tab (and *Apply* it) or (b)
check the appropriate box in the Reversals column of Config>Homing/Limits for the axis
that you are using. You can also, of course, just switch off change the wiring. For a stepper
drive you need to reverse **one** pair of physical connections to the motor from the drive

electronics. For a brushed-motor servo drive you will need to reverse both the wires to the armature and the A and B inputs from the encoder.

If a stepper motor hums or screams then you have wired it incorrectly or are trying to drive it much too fast. The labelling of stepper wires (especially 8-wire motors) is sometimes very confusing. You will need to refer to the motor and driver electronics documentation.

If a servo motor runs away at full speed or flicks and indicates a fault on its driver then its armature (or encoder) connections need reversing (see your servo electronics documentation for more details). If you have any troubles here then you will be pleased if you followed the advice to buy current and properly supported products – "buy right, buy once"!

Most drives will work well with a 1 microsecond minimum pulse width and need no delay before the *Dir* pin is changed (*Direction PreChange*). If you have problems with the test moves (e.g. motor seems too noisy) first check that your step pulses are not inverted (by *Low active* being set incorrectly for Step on the Output Pins tab of Ports and Pins) then you might try increasing the pulse width to, say, 5 microseconds. The Step and Direction interface is very simple but, because it "sort of works" when configured badly, can be difficult to fault-find without being very systematic and/or looking at the pulses with an oscilloscope.

### 5.6.2.2 Motor maximum speed calculations

If you feel that you want to calculate the maximum motor speed then read this section.

There are many things which define the maximum speed of an axis:

- Maximum allowed speed of motor (perhaps 4000 rpm for servo or 1000 rpm for stepper)
- Maximum allowed speed of the ballscrew (depends on length, diameter, how its ends are supported
- Maximum speed of belt drive or reduction gearbox
- Maximum speed which drive electronics will support without signalling a fault
- Maximum speed to maintain lubrication of machine slides

The first two in this list are most likely to affect you. You will need to refer to the manufacturers' specifications, calculate the permitted speeds of screw and motor and relate these to units per second of axis movement. Set this maximum value in the *Velocity* box of Motor Tuning for the axis involved.

The online Yahoo! group and the Forum are a useful places to get advice from other Mach3 users, world-wide, on this sort of topic.

## 5.6.3 Deciding on acceleration

### 5.6.3.1 Inertia and forces

No motor is able to change the speed of a mechanism instantly. A torque is needed to give angular momentum to the rotating parts (including the motor itself) and torque converted to force by the mechanism (screw and nut etc.) has to accelerate the machine parts and the tool. Some of the force also goes to overcome friction and, of course, to make the tool cut.

Mach3 will accelerate (and decelerate) the motor at a given rate (i.e. a straight line speed/time curve) If the motor can provide more torque than is needed for the cutting, the friction and the inertia forces at the given acceleration rate, then all is well. If the torque is insufficient then it will either stall (if a stepper) or the servo position error will increase. If the servo error gets too great then the drive will probably signal a fault condition but even if it does not then the accuracy of the cutting will have suffered. This will be explained in more detail shortly.

### 5.6.3.2    Testing different acceleration values

Try starting and stopping your machine with different settings of the Acceleration slider in the Motor Tuning dialog. At low accelerations (a gentle slope on the graph) you will be able to hear the speed ramping up and down.

### 5.6.3.3    Why you want to avoid a big servo error

Most moves made in a part program are co-ordinated with two, or more, axes moving together. Thus in a move from X=0, Y=0 to X=2, Y=1, Mach3 will move the X axis at twice the speed of the Y axis. It not only co-ordinates the movements at constant speed but ensures that the speed required relationship applies during acceleration and deceleration but accelerating all motions at a speed determined by the "slowest" axis.

If you specify too high an acceleration for a given axis then Mach3 will assume it can use this value but as, in practice, the axis lags behind what is commanded (i.e. the servo error is big) then the path cut in the work will be inaccurate.

### 5.6.3.4    Choosing an acceleration value

It is quite possible, knowing all the masses of parts, moments of inertia of the motor and screws, friction forces and the torque available from the motor to calculate what acceleration can be achieved with a given error. Ballscrew and linear slide manufacturers' catalogues often include sample calculations.



**Figure 5.10 - Establishing a zero position**

However, unless you want the ultimate in performance from your machine, we recommend setting the value so that test starts and stops sound "comfortable". Sorry it's not very scientific but it seems to give good results!
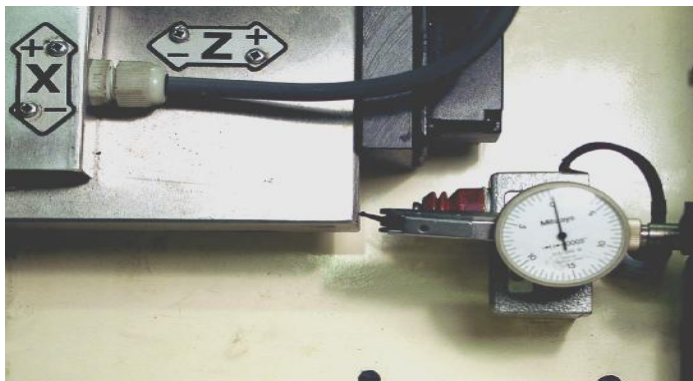
## 5.6.4    Saving and testing axis

**Finally don't forget to click *Save Axis* to save the acceleration rate before you move on.**

If you tested the axis drive motor off the machine then now is the time to install it.

Then you should test your acceleration and maximum speed values by executing some G00 moves using Manual Data Input. Mach3 can jog axes more smoothly than it can run them under program control because it does not have to worry about synchronising the movement of more than one axis when it is jogging. If you have rough sounding movement then you should slightly reduce the speed and/or acceleration settings.

You should now check your steps per unit calculations by using the MDI on the Diagnostics screen to make a defined G00 move. For a rough check you can use a steel rule. A more accurate test can be made with a Dial Test Indicator (DTI)/Clock and a slip gage block.

Suppose you are testing the Z axis and have a 3" gage block.

Use the MDI screen to select Inch units and absolute coordinates. (G20 G90) Set up the DTI away from the chuck touching a flat face on the saddle.  Jog the Z axis so the DTI probe just touches the saddle as in figure 5.10. Ensure you finish by a move in the plus Z direction to avoid problems with backlash.

Rotate the DTI bezel to zero the reading.

Now click in the Z axis DRO, type 0 (zero) and press *Enter*.

Move the table to Z = -3.5 by G00 Z-3.5 The gap should be about three and a half inches. If it is not then there is something badly wrong with your calculations of the Steps per Unit value. Check and correct this.

Insert the gage block and move to Z = -3.0 by G00 Z-3.0   This move is in the Z plus direction as was the jog so the effects of backlash in the mechanism will be eliminated. The reading on the DTI will give your positioning error. It should only be up to a thou or so. Figure 5.13 shows the gage in position.
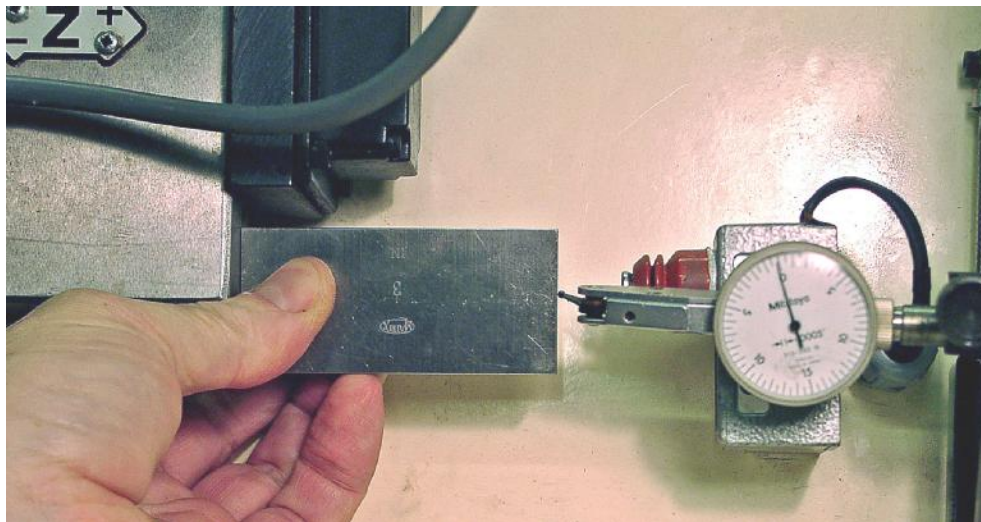


**Figure 5.13 - Gage block in position**

Remove the gage and G00 Z0 to check the zero value. Repeat the 3" test to get an set of, perhaps, 20 values and see how reproducible the positioning is. If you get big variations then there is something wrong mechanically. If you get consistent errors then you can fine tune the Steps per Unit value to achieve maximum accuracy.

Next you should check that the axis does not lose steps in repeated moves at speed. Remove the gage block. Use MDI to G00 Z0 and check the zero on the DTI.

Use the editor to input the following program:

```
F1000 (i.e. faster than possible but Mach3 will limit speed)
G20 G90 (Inch and Absolute)
M98 P1234 L50 (run subroutine 50 times)
M30 (rewind and stop)
O1234 (letter O)
G01 Z-3
G01 Z0 (do a feed rate move and move back)
M99 (return from subroutine)
```

Click *Cycle Start* to run it. Check that the motion sounds smooth.

When it finishes the DTI should of course read zero. If you have problems then you will need to fine tune the maximum velocity or acceleration of the axis.

### 5.6.5    Repeat configuration process on other axis

With the confidence you will have gained with the first axis you should be able to quickly repeat the process for the other axis.

### 5.6.6    Spindle motor setup

If the speed of your spindle motor is fixed or controlled by hand then you can ignore this section. If the motor is switched on and off, in either direction, by Mach3 then this will have been setup with the relay outputs.

If Mach3 is to control the spindle speed either by a servo drive that accepts Step and Direction pulses or by a Pulse Width Modulated (PWM) motor controller then this section tells you how to configure your system.

### 5.6.6.1 Configuring PWM and Step & Direction spindle motors.

**PWM:** We have already looked at configuring a PWM spindle drive. To complete the spindle setup all you will need to bear in mind is the maximum speed of the motor (i.e. its speed with a 100% on pulse stream).

**A Step & Direction** drive is configured on the same dialog as axis drives. First you need to know the maximum speed that you want your motor to run at. This will depend on its design and should be available on its rating plate. You can of course opt for a lower speed than the maker specifies if you wish.

Next you need to work out how many step pulses needed to be input to the drive electronics to make its shaft turn by one revolution. This will depend on the construction of the motor, any encoder on its shaft and the any electronic gearing in the drive electronics. You should be able to work it out from the documentation that is supplied with the motor and its drive. This number is the value to enter into *Steps per Unit* in the dialog. If this value is bigger than 500 then you may have problems getting Mach3 to run the motor at full speed. If you have electronic gearing then use it to reduce the *Steps per Unit* for full speed.

Now, rather than setting the velocity by a smooth feel as was described for axes, you must set *Vel* to the maximum speed you have chosen. The unit are revolutions per **second** so if you want 3000 rpm you need to enter (3000 ÷ 60) = 50.0 in the *Vel* box.

If you have a large number of *Steps per Unit* you may find that Mach3 will not allow the *Vel* you require. You can get round this by using the electronic gearing of your drive to reduce the number of steps per unit required or possibly by setting the kernel frequency to 35000 or even 45000 Hz. *Note:* if you change the kernel speed then you will need to retune the axes.

Finally you should set the acceleration. This is best done by ear to give a smooth start to the spindle when running with the highest speed pulleys. You can type a value into the acceleration box and this can give much lower accelerations that are obtained by the slider. If you want you can allow acceleration to take around 10 seconds!

**Note:** Some small lathes have chucks that are screwed onto the spindle nose. Do not allow these to decelerate too quickly or the chuck may come unscrewed.

### 5.6.6.2 Motor speed, spindle speed and pulleys

So now you can get Mach3 to control the speed of the **motor**. When you are machining what you and the part program (the S word) want to set is the speed of the **spindle**. The motor and spindle speed are, of course, related by the pulleys or gears connecting them. We will use the term "pulley" to cover both sorts of drive in this manual.

The settings for pulleys, which we ignored earlier, are on the *Spindle Setup* tab of Config>Ports and Pins.

If you do not have motor speed control then choose Pulley 4 with a high maximum speed like 10,0000 rpm and 0 as minimum speed. This will prevent Mach3 complaining if you run a program with a S word asking for say 6000 rpm.

There are two steps involved using pulleys. When the system is configured (i.e. what you are doing now) you



**Figure 5.14 – A 4-speed pulley drive on low speed**

define up to 4 available pulley combinations. These are , of course, defined by the physical sizes of the pulleys or the ratios in a geared head. Then when a part program is being run
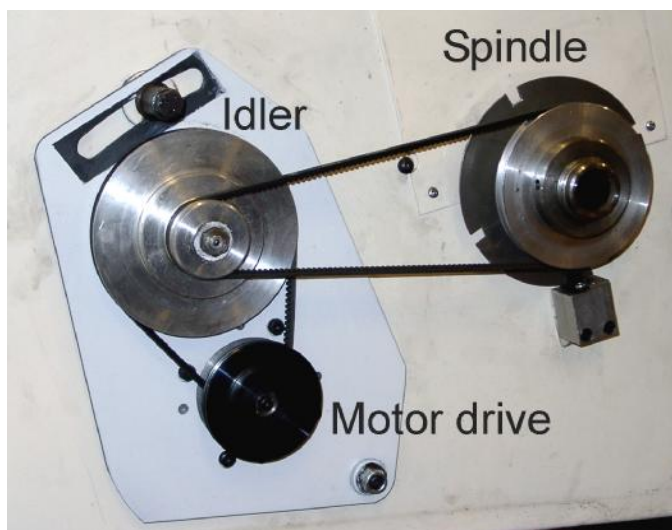
the operator must define, perhaps by switches or entering a number into a DRO which pulley (1 to 4) is in use.

The machine's pulley ratios are set on the Config>Ports and Pins *Spindle Setup* tab dialog (figure 5.6). Under *Pulley Ratios* the minimum and maximum speed of the four pulley sets are defined together with the default one to be used. The maximum speed is the speed at which the **spindle** will rotate when the motor is at full speed. Full speed is achieved by 100% pulse width in PWM or the set *Vel* value on Motor Tuning "spindle Axis" for Step & Direction.



**Figure 5.15 – Pulleys on high speed setting**

As an example, suppose the configuration that we will call "Pulley 1" is a step-down of 5:1 from motor to spindle and the maximum speed of the motor is 3600 rpm. That means that the maximum speed of Pulley 1 will be set to 720 rpm (3600 ÷ 5). Now suppose that Pulley 4 is a step-up of 4:1. With the same motor speed its maximum speed would be set to 14,400 rpm (3600 x 4). The other pulleys (if available) would presumably be intermediate ratios. The pulleys do not need to be defined in increasing speeds but the numbers should relate in some logical way to the controls on the machine tool.

The minimum spindle speed is to avoid trying to run a motor below its minimum rating. Set these (if required) by using the same ratios on the minimum motor speed.

Mach3 uses the pulley ratio information as follows:

- When the part program executes an S word or a value is entered into the set speed DRO then the value is compared with the maximum speed for the currently selected pulley. If the requested speed is greater than the maximum then an error occurs.

- When the part program executes an S word or a value is entered into the set speed DRO then the value is compared with the minimum speed for the currently selected pulley. If the requested speed is less than the minimum then an error occurs.

- Otherwise the percentage of the maximum for the pulley that has been requested and this is used to set the PWM width or the Step pulses are generated to produce that percentage of the maximum motor speed as set in Motor Tuning for the "Spindle Axis".

As an example suppose the max spindle speed for Pulley #1 is 1000 rpm. S1100 would be an error. S600 would give a pulse width of 60%. If the maximum Step and Direction speed is 3600 rpm then the motor would be "stepped" at 2160 rpm (3600 x 0.6).

If you have a drive which "faults out" on steps that are too fast then you should set your minimum speeds slightly higher than the speed at which your drive cuts out so that Mach3 knows that there is a problem rather than trying to cut with a stationary spindle.

### 5.6.6.3  Testing the spindle drive

If you have a tachometer or stroboscope then you can measure the spindle speed of your machine. If not you will have to rely on your index or timing sensor and experience.

On the Manual screen, choose a pulley that will allow 900 rpm.  Set the belt or gearbox on the machine to the corresponding position. Then type 900 into the S DRO. Click the Spindle button to start it. Measure or estimate the speed. If it is wrong you will have to revisit your calculations and setup.

You might also check the speeds on all the other pulleys in the same way but with suitable set speeds.

## 5.7    Other configuration

There are many other configuration options in Mach3Turn. You can look at these in the Config menu. You should find that the default values allow you to actually use your machine quite satisfactorily.

When you wish to look at any of these you should refer to chapter 8, where they are described in detail.

# 6. Mach3 controls and running a part program

> This chapter is intended for reference to explain the screen controls provided by Mach3 for setting up and running a job on the machine. It is of relevance to machine operators and for part-programmers who are going to prove their programs on Mach3.

## 6.1 Introduction

This chapter covers a lot of detail. You may wish to skim section 6.2 and then look at the sections for inputting and editing part programs before returning to the details of all the screen controls.
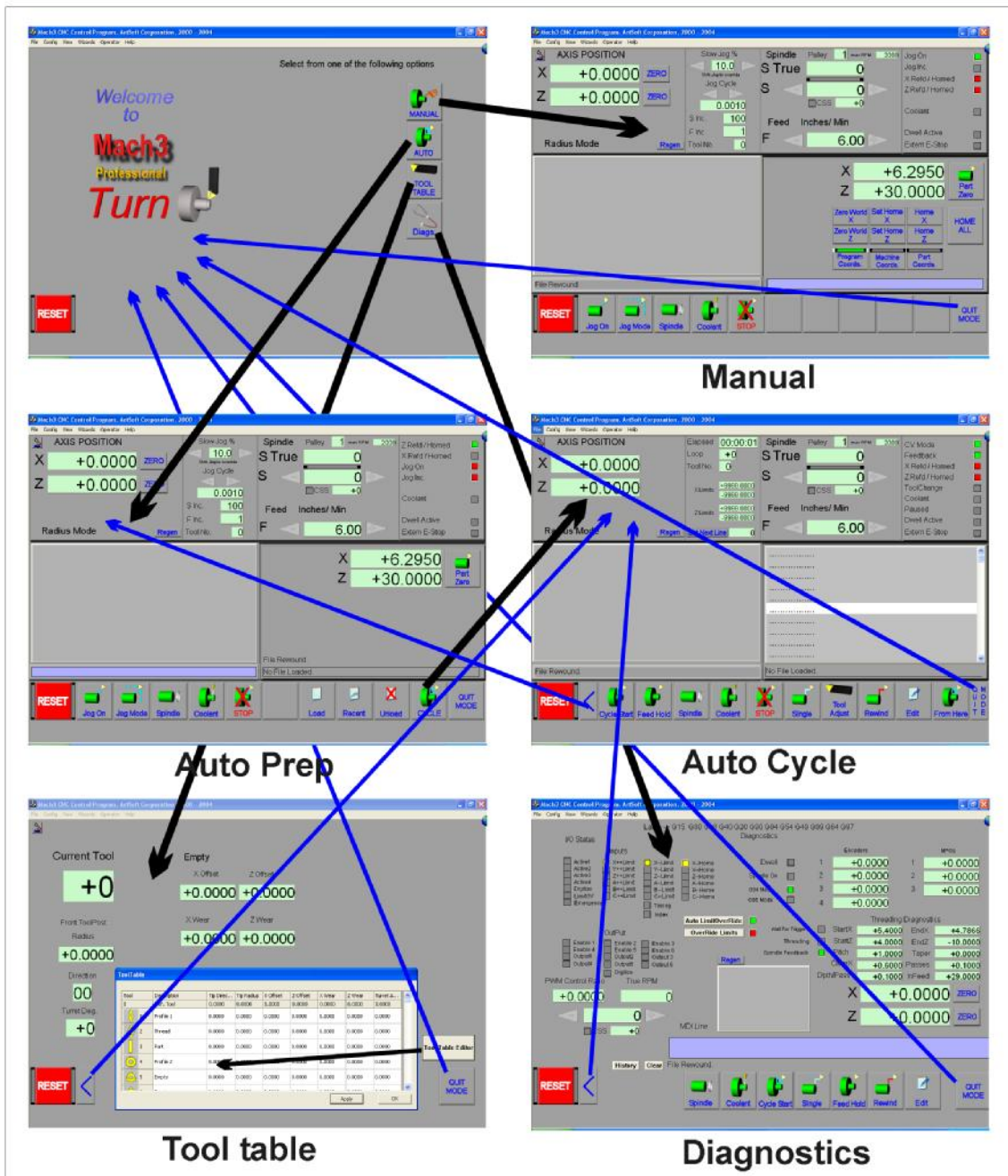


**Figure 6.1 – Navigation up and down the screens**

## 6.2 How the controls are explained in this chapter

Although at first sight you may feel daunted by the range of options and data displayed by Mach3, this is actually organised in a way to suit the sequence of operations you will perform when turning (often called the Workflow).

There are screens for Manual operation (*Manual*),. Preparing for Automatic operation (*Auto Prep*), running the part program in an Automatic cycle (*Auto Cycle*), setting up the Tool Table (*Tool Table*) and Diagnosing trouble (*Diags*).

Screens consist of a set of controls. A control is a button or can refer to the information displayed by DROs (digital read-outs), labels or LEDs (light emitting diodes).

The LEDs are grouped on an LED panel. Some controls appear in groups (families) on several screens. These are explained first and then the workflow for some typical jobs is given. Finally the use of the *Diags* screen is covered.

A Screen Designer is provided that allows controls to be removed or added from the screens of a set of screens. You can modify standard screens or design your own screens from scratch so that you can add any controls to a particular screen if your application requires this. For details see the *Mach3 Customization* wiki.

### 6.2.1 Screen switching  controls

These controls and the way they navigate around the set of screens is shown in figure 6.1.

The *Quit Mode* buttons take you back to the Welcome screen and the *Back* (<) buttons return you to a logical place to continue work from the current screen.

#### 6.2.1.1 Reset button

All screens have a Reset button. This is a toggle. Before Reset the LEDs will flash red. When the system has been Reset the LEDs will be dark, the charge pump pulse monitor (if enabled) will output pulses and any Enable outputs chosen will be active.

#### 6.2.1.2 Labels

There are several "intelligent labels" display things like the last "error" message, the current modes, the file name of the currently loaded part program (if any) and the Profile that is in use.

### 6.2.2 Axis position family

This family is concerned with the current position of the tool (or more precisely, the controlled point). See figure 6.2.
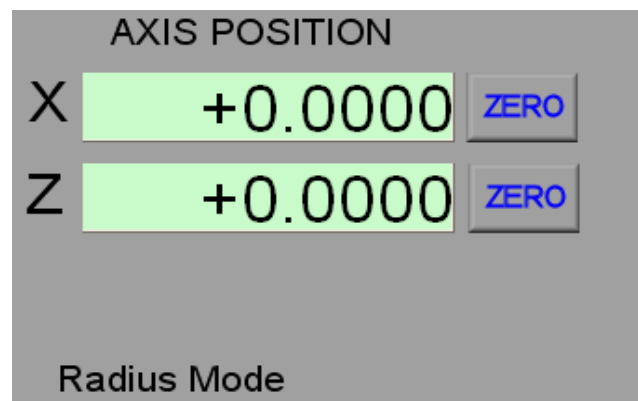
The axes have the following controls:

#### 6.2.2.1 Coordinate value DRO

These are displayed in the current units (as set by G20/G21) unless locked to the setup units on the Config>Logic dialog. The value is the coordinate of the controlled point in the displayed coordinate system. This will generally be the coordinate system of the current Work Offset (usually 1 - i.e. G54) together with any G52 or G92 offsets applied. It can however be switched to display Absolute Machine Coordinates.



**Figure 6.2 – Axis position family**

The *Zero* button allows you to make the relevant coordinate of the controlled point be 0.0. This will update the relevant current Work Offset. You can also type a new value into an Axis DRO. This will modify the current Work Offset to make the controlled point in the

current coordinate system be the value you have set. The Zero button is just a handy way of "typing" 0.0 into the DRO.

### 6.2.2.2 X axis mode

A smart label defines if the machine is set to Radius mode or Diameter mode. This mode determines how the value in X-axis DROs should be interpreted and what position X-words define when used in part programs.

## *6.2.3   Motion control family*

The Motion control family allows you to set up and monitor the speed of the spindle and of feed-rate motion of the tool. The controls are shown in figure 6.3

### 6.2.3.1 Spindle

The *S* DRO displays the currently requested spindle speed. It is set by a part program or MDI using an S word, by typing a value into the DRO and pressing *Enter* or by clicking the plus or minus arrows.

The LED above the *S* DRO glows when the spindle is running.



**Figure 6.3 – Motion control family**

The family displays the pulley number and its maximum speed. You can define the pulley currently in use by typing a number (between 1 and 4) into the *Pulley* DRO and pressing *Enter*.

You cannot run slower than the minimum speed or exceed the maximum speed for the current pulley. Such a request will generate an error message and clip the speed to the relevant limit.

The *S True* DRO displays the actual speed of the spindle as sensed by the Timing or Index input.

A part program (or the MDI line) can program the spindle speed to aim to give a constant surface speed (CSS) for cutting. This is particularly useful in operation facing large diameter stock. CSS mode is entered using the G96 with an S word giving the surface speed. In this mode the spindle will run faster and faster as the X coordinate of the controlled point reduces until it reaches its maximum speed.

The *S* DRO and *CSS* DRO reflect what is happening and the LED indicates that the CSS mode is active.

Revolution per minute mode is restored by G97 with an S word giving the speed in RPM.

### 6.2.3.2 Feedrate

When no part program is running, the *F* DRO defines the current requested feedrate. It is set by the F-word in a part program or entered in the MDI line or by typing into the DRO. When a program is running it will display the actual feedrate being used taking account of the coordinated motion of the two axes and the maximum speeds that are possible.

A smart label gives the units of the feedrate allowing for whether the machine is in metric or imperial units.

The feedrate is the speed of movement of the tooltip based on combining the coordinated X and Z motions. If this would require either axis to go faster than its maximum speed then the actual feedrate will be suitably clipped.

In feed-per-rev (G95 mode) the actual speed of tool movement will of course depend on the speed of the spindle as fedback by the Index/Timing sensor. If the spindle speed is nearly or
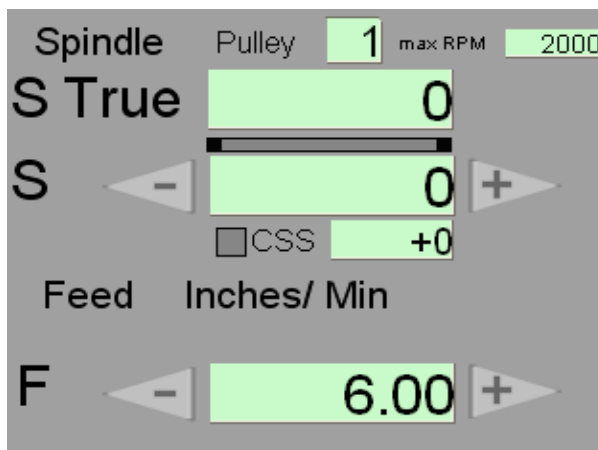
actually zero then Mach3 will give an error message and tell you it has sets its own default value.

The *F* DRO can be incremented and decremented by a defined amount using the plus and minus arrows beside it. The increment is set in the Increments control family.

**Warning:** Do take care when using the MDI to change from feed-per-minute to feed-per-rev modes as "sensible" feeds are numerically very different in the two modes. A feed of F = 6 inches per minute would be a very slow movement but F = 6 inches per rev at 700 rpm could be dangerous. Care is also needed with the increment/decrement buttons as a suitable increment for feed-per-minute will not be sensible in feed-per-rev.

**Figure 6.4 – LEDs panel on Manual**

## 6.2.4 *Jogging, Jog speed and Increments control family*

### 6.2.4.1 Jogging

The *Jog On* button is a toggle which enables and disables jogging of the machine. Jogging is inhibited on any screen (for example Auto Cycle) which does not have the *Jog On* button. There is a LED on the LEDs panel to show when it is enabled.

**Figure 6.5**

For now we will assume that you will jog your machine with the up and down cursor arrows for the X axis and the left and right cursor arrows for the Z axis.

Associated with *Jog On* is the *Jog Mode* button. Assuming you only use keyboard jogging there are two modes *Continuous* and *Incremental* (or *Step*) jogging.

In *Continuous* the axis moves all the time that the jog key is held down. The speed of movement is set by the *Slow Jog%* DRO in the Increments control family (see figure 6.6). If the *Shift* key is depressed with the jog key then the *Slow Jog%* value is ignored and the axis will travel at its fast traverse rate (as set when you tuned the motor).

In *Incremental (Step)*, which is indicated by the *Inc* light on the LEDs panel, the axis will move one increment or step for each key press. The speed of movement is set by the current feedrate (*F* DRO) and feedrate mode (G94/G95). The size of the step

**Figure 6.6 – Increments family**

that will be used is shown in the DRO under the *Jog Cycle* label in the increments family (i.e. 0.0010 in figure 6.6). You can type whatever value of increment you like into this DRO. Mach3 has a list of 10 preset increments. This list is cycled and copied into the DRO using the plus and minus arrows under *Jog Cycle*. You can create your own list using the table in the Config>State dialog.
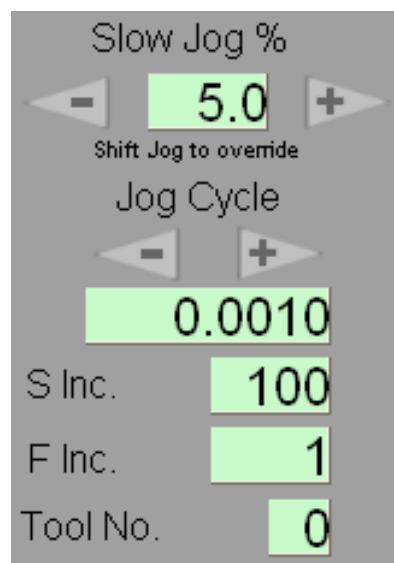
**Note:** If your jogging keys suddenly do not work it is probably because you have the MDI line open (see below).

Suitably equipped machines can also be jogged using MPG (manual pulse generator) handwheels.

Provision of two MPGs will allows the lathe to be used for manual turning and will greatly improve the speed in setting up prototype jobs on the machine. Details of setting up MPG jogging is give in chapter 8.

### 6.2.4.2 Other increments etc.

The *S Inc.* DRO is the value by which the *S* DRO is changed when its plus and minus buttons are used.

The *F Inc.* DRO is the value by which the *F* DRO is changed when its plus and minus buttons are used.

The *Tool No.* DRO shows the number of the tool currently in use.

## 6.2.5  Machine Setup control family

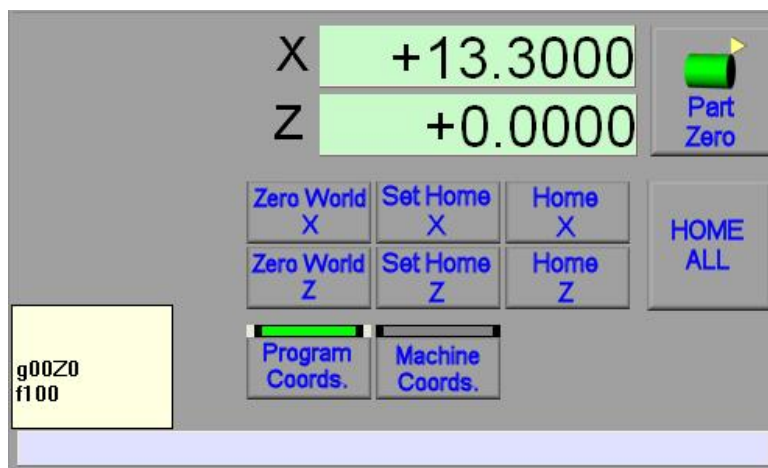This family allows you to setup your machine ready for working. See figure 6.6.



**Figure 6.6 – Machine setup family**

## 6.2.6  Coordinate system display

Normally the axis DROs display the coordinates of the controlled point allowing for the tool offsets and any work offset set by entering a new value into an axis DRO or the Fixture table. It is this coordinate system in which moves by G00, G01 G02/03 etc. are normally made. In Diameter mode, the X value will be the diameter of the component that would be turned at the given setting. This display mode is chosen by the *Program Coords.* Button and indicated by the attached LED.

The *Machine Coords.* Button will display the absolute machine coordinate system. This is the position of the axes without any offsets applied. You will probably not need to use this unless you are troubleshooting problems with your offset tables. Machine coordinates are always distances (i.e. the X value is a radius not a diameter whatever mode the system is set to)

## 6.2.7  MDI line

In figure 6.6, the MDI line is shown active. It is a pale color and the cream pop-up shows some recently executed MDI commands.

You make it active by clicking in the line or typing *Enter* on a screen with an MDI line.

You can type any line that would appear in a part program and execute it by terminating it with *Enter*. The line remains open until it is closed by clicking on another control or pressing *Esc*. You can select from the recently executed commands by using the up and down cursor keys. The *backspace* and *Del* keys can be used with the left and right arrows to correct typing mistakes.

**Notes:** The MDI line is not automatically closed by Enter. If the jogging keys do not work it is probably because the line is open.

Moves made with tool nose radius compensation turned on will not be as expected, as this feature needs to "look-ahead" in a part-program for its operation.

### 6.2.8   *Referencing/Homing*

Before you can do useful work you have to make sure that Mach3 "knows" when the tool actually is with respect to the work. You will have to give this information whenever the system is switched on and when the work is moved in the chuck. This process is called Referencing or Homing.

For the present we assume that you only have one tool (Tool 1). Chapter 7 covers using several different tools.

#### 6.2.8.1   Axis with Home switches

If your lathe is configured with home switches then the buttons *Set Home* will cause the relevant axis to move slowly in the direction of the Home switch. The speed and direction are setup using Config>Homing/Limits. For details see chapter 8. When the switch is hit then the axis will move a short distance in the reverse direction until it is clear of the switch.

The current absolute position of the axis will then be set to the location of the switch that is defined in Config>Homing/Limits (often 0.0 for the Z axis and the size of work that would be turned by Tool 1 for the X axis).

#### 6.2.8.2   Axis without Home switches

With no Home switch defined for an axis, Mach3 will assume that you have jogged the axis to its home position before pressing *Set Home*.

The current absolute position of the axis will then be set to the location that is defined in Config>Homing/Limits (often 0.0 for both axes).

#### 6.2.8.3   Referenced state

The LED panel displays the axis that have been referenced (i.e. is in a known actual position after homing)

#### 6.2.8.4   Moving to Home position

The *Home X* or *Home Z* button moves the corresponding axis to its home position providing it has been already referenced.

The *Home All* button does a *Home X* followed by a *Home Z*

#### 6.2.8.5   Machine coordinates

The *Zero World*  buttons set the current position of the corresponding axis to 0.0. No motion takes place. Zero World does not alter the referenced status of the axis so should be used with care as it will almost certainly alter the action of the *Home* and *Home All* functions.

#### 6.2.8.6   Part zero

This function is described in the context of setting up tools in chapter 7.

## 6.3   Using Wizards

### 6.3.1   *What is a Wizard*

A Mach3 Wizard is a screen (or set of screens) which helps automate a machining task. It shows you a series of labelled DROs. There will usually
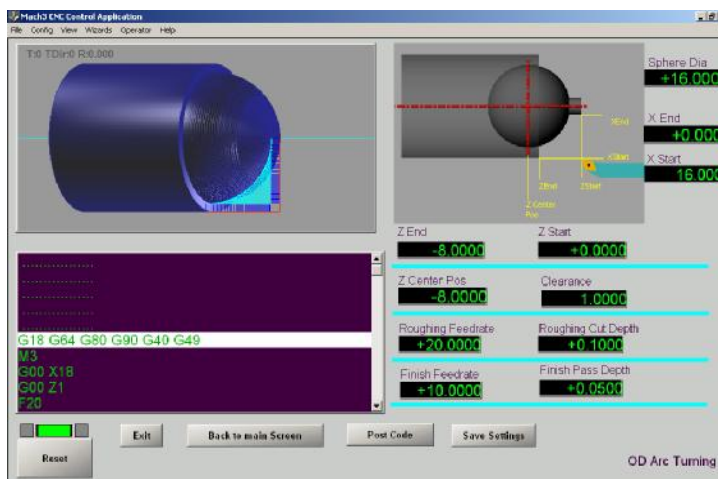


**Figure 6.22 – Data, code and toolpath for a Wizard**

be a graphic image to illustrate the data to put into the DROs. Figure 6.22 illustrates the external sphere Wizard in use.

There are two types on Wizard.

The most common takes the data provided and uses this to write a G-code program to cut the component specified. Most Wizards will display this code in a G-code window and will show the related toolpath display.

The other sort of Wizard actually does the machining rather than presenting you with the part program to run yourself. This sort of Wizard is suitable for simple operations.

Several Wizards are supplied with Mach3Turn but Wizards are also available from third party suppliers and as contributions from other users linked on the MachSupport site. If you have experience in writing G-code then you can implement your own Wizards. Details are given in the *Customizing Mach3* wiki.

### 6.3.2   *Wizard controls*

Although Wizards from different authors will have different screen layouts they should al have the following controls:

*Post Code*: This takes the data from the DROs and writes the G-code program to do the cuts. This will be replaced by an **Execute** button for a Wizard which does its own machining.

*Save Settings*: This will remember the current values of the DROs for the next time you run the Wizard

*Exit*: Returns to the main Mach3 screens so that you can run a generated part program

Wizards will generally collect data on the tool, the feedrates and the cut depths that you require. They will, however, generally assume that you will set a spindle speed, start the spindle and turn on any coolant required before running the G-code.

### 6.3.3   *Re-using Wizard generated code*

When you have posted the code and exited to Mach3 then the generated program will be in a temporary file. You can use the editor to alter this and of course to save it in a file in one of your own folders.

The editor can also be used to combine the code from several Wizards to produce a part program for something like the shaft-end illustrated in  figure 7.1. in this way you can produce quite complicated components without having to use a CAD/CAM system.

## 6.4    Loading and running a G-code part program

### 6.4.1   *Introduction*

A part program written by hand, by a Wizard or by a CAD/CAM system is run using the *Auto* screens.

### 6.4.2   *Auto Prep*

The Auto Preparation screen (figure 6.25) allows you to load the G-code file, to jog and setup the correct program coordinate origin



**Figure 6.25 – The Auto Preparation screen**

(see Chapter 7), to define a feedrate and spindle speed (in case the program does not set these).

The *Load* button displays an open file dialog to allow you to select and load a part program file. When the file is chosen, Mach3 will load and analyse the code. This will generate a toolpath for it, which will be displayed. The loaded program code will be displayed in the G-code list window. You can scroll through this moving the highlighted current line using the scroll bar.

The *Recent* button displays a list of recently loaded files. See figure 6.26

The *Unload* button closes the part program file.

After Preparation you move to the Auto Cycle screen using the *Cycle* button.
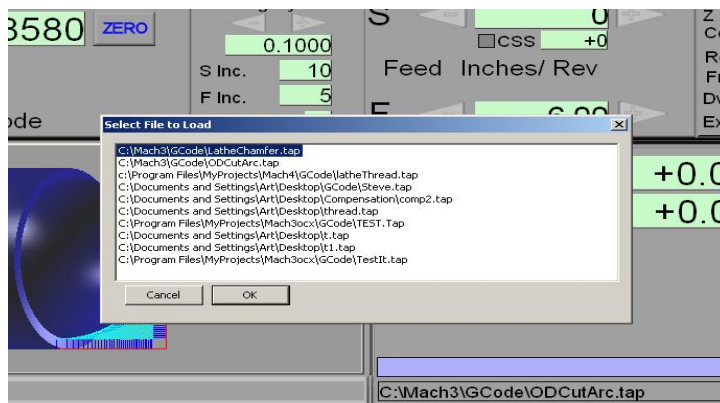
**Figure 6.26 – Recent file list**

### 6.4.3 Auto Cycle

This screen is used to run and, if necessary amend, a loaded part program. See figure 6.27

The function of its buttons are defined as follows:

*Cycle Start*: This will run the part program from the current block (line) or continue after a feed hold or an M00 or M01 command.

*Feed Hold*: Will stop the part program running as soon as possible but in a totally controlled way so that it can be restarted. Spindle and Coolant will stay on if they were

**Figure 6.27 – Auto Run screen**

on at the time of the Feed Hold. During Feed Hold you can perform whatever operation, such a jogging, updating Wear offsets in the tool table, altering spindle speed etc. that is required to overcome a problem with the machining. Cycle Start will resume the run of the program.

*Stop*: Will stop the run of the part program as quickly as possible. Resumption of the run will not generally be possible and the axes may have lost or gained steps due to the rapid stop so the lathe needs to be re-referenced.

*Single*: This button is a toggle. When it is On, Cycle Start will only execute one block and then pause. This can be useful for debugging a part program.

*Tool Adjust*: Allows you to edit the offsets of the current tool or to edit the tool table.

*Rewind*: Rewinds the part program file to the first block.

*Edit*: Runs the external editor program (whose name is defined in Config>Logic)

### 6.4.4 Editing a part program

Provided you have defined a program to be used as the G-code editor (in Config>Logic), you can edit the code by clicking the *Edit* button. Your nominated editor will open in a new window with the code loaded into it.

When you have finished editing you should save the file and exit the editor. This is probably most easily done by using the close box and replying Yes to the "Do you want to save the changes?" dialog.

While editing, Mach3 is suspended. If you click in its window it will appear to be



**Figure 6.28 – Editing a part program (with Notepad program)**

locked up. You can easily recover by returning to the editor and closing it.

After editing the revised code will again be analysed and used to regenerate the toolpath and extrema. You can regenerate the toolpath at any time using the *Regenerate* button.

### 6.4.5 Inputting a hand-written program

If you want to write a program "from scratch" then it is easiest do so by running the editor outside Mach3 and saving the file.

It is good practice for a part program to make no assumptions about the state of the machine when it starts. It should therefore include G17/G18/G19, G20/G21, G40, G49, G61/G62, G90/G91, G93/G94.

You need to decide whether the program starts with an S word or if you need to set the spindle speed by hand or by entering a value in the *S* DRO.

You will need to ensure that a suitable feedrate is set before any G01/G02/G03 commands are executed. This may be done by an F word or entering data into the *F* DRO.

Next you may need to select a Tool and apply its offsets with G43 (or G44).

Finally, unless the program has been proved to be valid you should attempt a dry run, cutting "air" to see that nothing terrible happens.

### 6.4.6 Running your program

You should monitor the first run of any program with great care. You may find that you need to override the feed rate or, perhaps, spindle speed to minimise chattering or to optimise production. When you want to make changes you should either do this on the "fly" or use the *Feed Hold* button, make your changes and then click *Cycle Start*.

## 6.5 The principles of CNC threading

This section explains how Mach3 implement threading using a single point tool.

We assume that you will use disposable carbide tooling for your threading because it avoids the difficulty of accurate grinding of the angles when sharpening a high-speed tool. If you can do this there is, of course, no reason why Mach3Turn will not work with HSS tooling.

*Note:* that actually cutting a thread is not available on a Demo (unlicensed) version of the program.

### 6.5.1 Threading in general

A thread is in principle merely a cut made with a feedrate per revolution of the stock equal to the thread pitch. In practice one needs to make several passes to get the required thread depth and each must, of course, start at the "same angular position" on the stock.

For threading Mach3 makes use of the pulse from the spindle index sensor to know when to start cutting (i.e. rather like a thread indicator on a manual lathe tells the operator when to close the half-nuts). By timing the index pulses, Mach3 can also work out the feed per minute that corresponds to the pitch of the thread.

Although the principles are simple, the calculations of where to position Z to start each pass of the cutting are tedious, so most users will use Mach3 Wizard or a CAD/CAM system to generate the required G-code.

### 6.5.2 Speeds and cut depths

Because there is no manual intervention, it is usual for a CNC lathe to cut a thread with a higher spindle speed and more passes than one might use by hand. Unless your Z axis drive is very slow or you are cutting very coarse threads you should expect to run the spindle between 400 and 1000 RPM depending on the diameter of the work.

Each pass of the thread should consist of a few pitches of cutting in the air (i.e. a Z clearance) so the Z axis can accelerate without pitch errors in the actual thread.

### 6.5.3 Infeed

With manual threading it is usual to set over the topslide so that the feed is down the thread flank. This controls wear on the tool and optimises finish.

With CNC threading there are various infeed strategies. Full details are given in the tool insert manufacturer's literature which can be consulted via the Internet. An infeed of slightly less than half the thread angle (e.g. 29 degrees for 60 degree thread-forms) is suitable for most work.

The depth of each pass is usually arranged to **decrease** as the width of the cut **increases**. In other words a constant chip volume is cut in each pass. This optimises the life of the tool and the quality of the cut. CAD/CAM packages and the threading Wizards will do this for you.

### 6.5.4 Threading with a Wizard

Maach3Turn is supplied with a simple threading Wizard which can be used to cut a wide range of threads without having to do difficult calculations. The main Wizard screen is shown in figure 6.29.

The values in the DROs may be different on your system. When you have set up values you can make them your default by clicking the *Save Settings* button. Remember that when you want to change a DRO you click it, type in the new value and then press the *Enter* key. If you just click the mouse in another DRO then the value you have typed but not *Enter*ed will be lost.

#### 6.5.4.1 Thread properties

The example shown uses numbers for a metric thread. You can just as well cut imperial threads. The current G20 (inch) or G21 (metric) setting of your system will be used by the Wizard. You can use the Wizard whether you are running Demo Radius or Diameter mode but the graphics on the screen depict a Diameter mode setup.

**X Start & X End**: These values specify the tool position for the crest and the root of the thread. The exact values to use depend on the type of threading insert that you are going to use. A "full form" insert will only produce one pitch of thread (for example a different

insert is required for an ISO 1.0 mm thread and an ISO 1.5 mm thread). It does, however produce the correct root radius and will cut the crest. A generic 60° insert can cut any pitch of thread with that angle (e.g. metric, UNC etc.) It will however produce a rather small radius root (and of course has a weaker tip because of this) and you are responsible for providing the correct diameter for the crest. You will need to refer to a set of thread tables or measure an existing thread to set these values – especially if you want a close fitting thread.
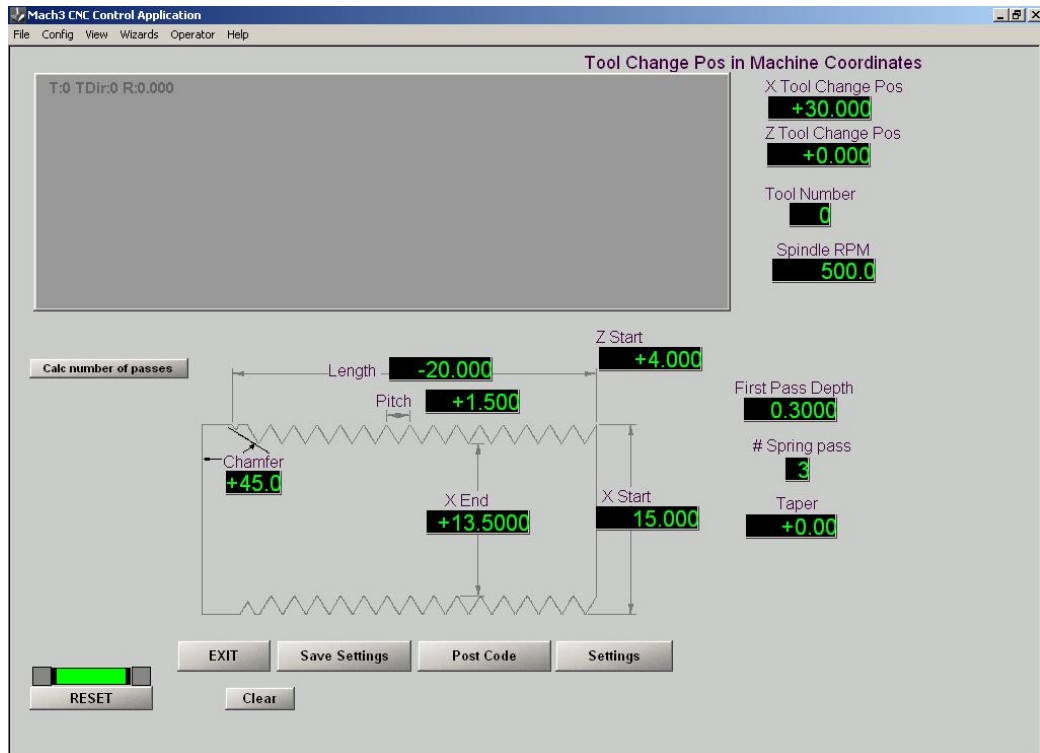


**Figure 6.29 – Default threading parameters**

**Z Start & Length:** It is assumed that you will be cutting a right hand thread towards the chuck (i.e. the tool moves in a negative-Z direction). It is easiest to have Z = 0 as the end of the stock but this is not vital. Set Z Start so it is two or three pitches clear of the end of the stock. This allows the Z motion to stabilize by cutting the first bit of thread in air. The length is the total length required including the air cut.

**Chamfer:** Mach3 will pull the tool out at the end of the thread so you do not need to machine a relief groove for it to run into. An angle of 45° will be suitable for most work. If you want to pull out with a very small angle then you will need a high performance drive on the X axis.

**Pitch:** Is the distance between two successive crests of a thread. In imperial threads this will be the reciprocal of the threads per inch (e.g. an 8 tpi thread has a pitch of 0.125")

**Taper:** You can cut a tapered thread, such as is used on steel steam pipe fittings. The default value of 0.0 is used for conventional thread used as fixings.

**Spindle RPM:** Set the required RPM here. The Wizard will warn you if the chosen speed is too fast for the pitch you are going to cut and the performance of your Z axis drive. Too low a cutting speed will compromise the finish from your cutting insert.

**First pass depth:** This is the depth of cut for the first threading pass. You will need to experiment to get a feel for suitable values. The Wizard will use an equal volume strategy for successive passes and, as the width of chip in the first pass is small, you will be able to take quite a deep cut. Too small a value will lead to an excessive number of passes.

**# Spring Passes:** This DRO defines the number of passes to be made at the *X End* diameter. These make fine cuts taking out any spring in the machine structure and the work. If you are threading a material that work hardens badly then you may wish to set this to 0 to avoid the tool rubbing.

**Tool change position:** It is easiest to mount the threading tool and set it up as tool 0 before you run the Wizard. If you set any other tool number in the *Tool Number* DRO then the Wizard will program a tool change at the start of its code. Even if Mach3 is configured to ignore tool changes, this will involve moving to the machine coordinates set in *X Tool Change Pos* and *Y Tool Change Pos*. Unless your machine is referenced with X and Z as some sensible location and these values are set in relation to it then this tool change move can cause you difficulties.

#### 6.5.4.2 Posting the code

When you have defined the properties of the thread then you can generate ("post") the code of a part-program to cut it. Click the *Post Code* button.
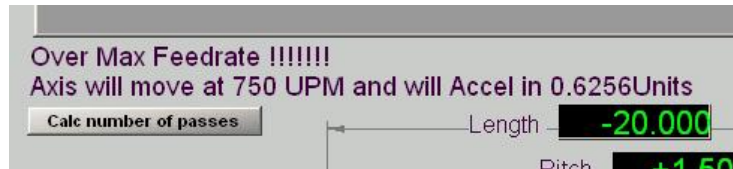


**Figure 6.30 – Spindle speed too high**

If the spindle speed is too fast for your Z axis then you will get the error shown in figure 6.30.

Provided the axis can handle your values you will get a toolpath display of the thread cuts and a report on the number of passes to be used. See figure 6.31. If there are too many passes the you should increase the *First Pass Depth* and click *Post Code* again. If there are too few passes then you risk breaking the tool or stalling the spindle so you should reduce the *First Pass Depth*.
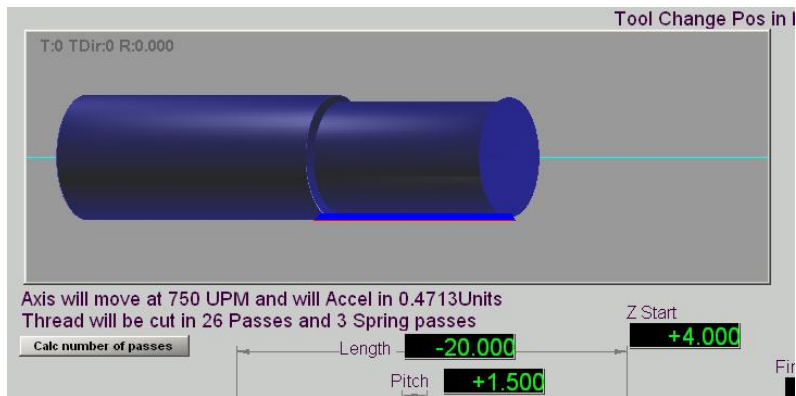


**Figure 6.31 – Preview of thread toolpath**

When you are satisfied with the parameters click Exit to return to Mach3Turns main screens. Go to the Auto Prep screen.

#### 6.5.4.3 Setting up tool and cutting thread

On the Auto Prep screen you will see your threading toolpath. If you have not already done so, load the stock and set the offsets so the tool is cutting the correct diameter and Z = 0 is in the correct place on the stock (e.g. the free end).

Now go to the Auto Cycle screen (figure 6.32). You will see the toolpath and the G-code. Click *Cycle Start* and your thread will be cut.
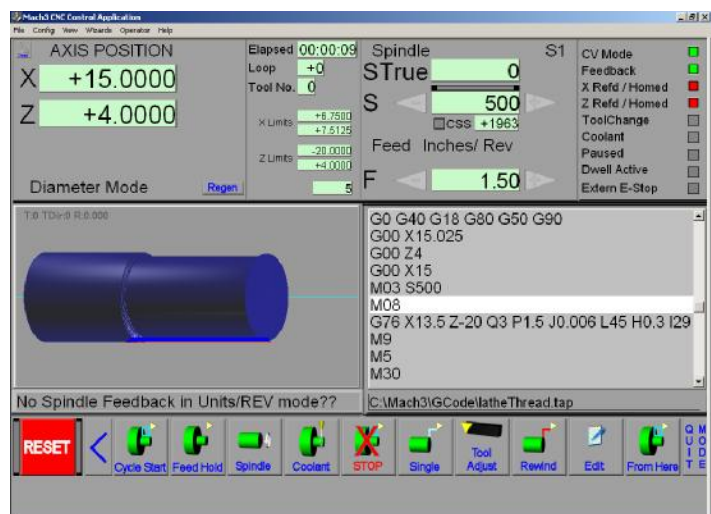


**Figure 6.32 – Ready to cut thread**

### 6.5.4.4 Fine tuning

The threading Wizard uses some additional values for fine tune the thread it produces. You can alter these by clicking the Settings button on the threading Wizard main screen (figure 6.29). This displays the Settings screen (figure 6.33)

**X Clearance and Z Clearance:** These values are the clearance values used when returning
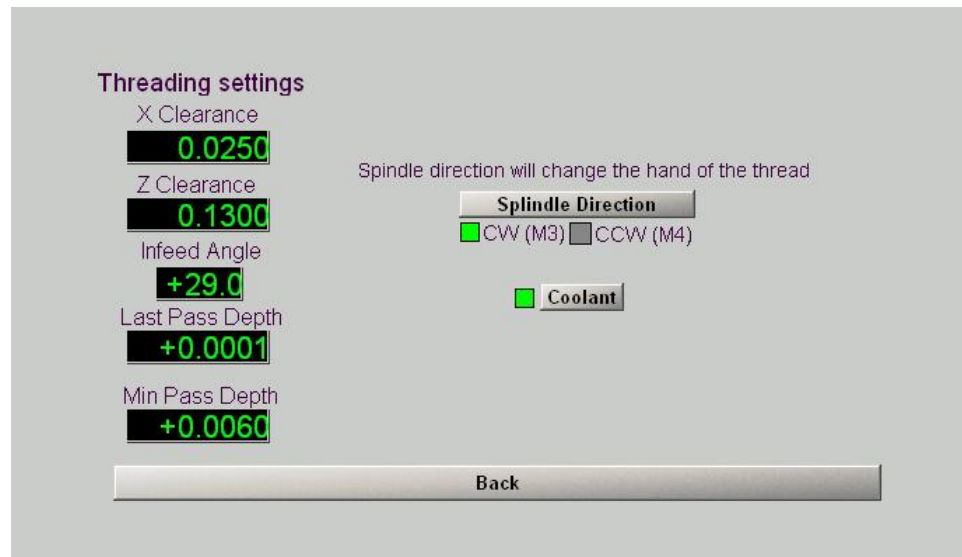


**Figure 6.33 – Threading Settings**

from each pass.

**Infeed angle:** This is the angle of the path taken by the tool infeeding each pass. It should be slightly smaller than the flank angle of the thread (e.g. $29^o$ is suitable for a $60^o$ thread and $26.5^o$ could be used for a Whitworth form thread)

**Last Pass Depth:** This constrains the last pass to be a given depth irrespective of the constant volume depth calculation.

**Min Pass Depth:** This DRO overrides the constant volume calculation. It can be used to prevent a tool rubbing in difficult materials. However if set too big it will reduce the number of passes that are ever used and as the cut width gets large near the root of the thread may cause chattering on a springy setup or tool tip breakage. It is suggested that the *Min* and *Last* pass depths are set small or even zero unless you have problems with the tool rubbing when making small cuts.

**Spindle Direction and Coolant:** These buttons let you choose the hand of your thread (you will have to invert tool in its holder, of course) and optionally turn on the coolant before threading.

### 6.5.4.5 Troubleshooting

**Values seem totally wrong**

When you upgrade Mach3 the saved DRO values are replaced by the default set provided by the Wizard writer. These are intended for imperial (G20) operation so will be unreasonable for metric machines. Use the Settings button, re-define your chosen defaults on the main Wizard screen and click *Save Settings*

**Adjusting First Pass does not let me get sensible number of passes**

You probably have an unsuitably large *Min Pass Depth* or perhaps *Last Pass Depth* in Settings. This is particularly likely to happen if you are changing from metric to imperial operation.

**Wizard will not post code**

Check that it is not failing as spindle speed is too high for the thread pitch chosen. Z axis tuning might be wrong. You could have a silly value in the *Pitch* or *Spindle RPM* DROs.

# 7. Setting up a job and tool tables

This chapter explains how Mach3 works out where exactly you mean when you ask the tool to move to a given position. It describes the idea of a coordinate system, defines the Machine Coordinate System and shows how you can specify the position of the cutting point of each Tool, the position of the workpiece in the chuck.

You may find it heavy going on the first read. We suggest that you try out the techniques using your own machine tool. It is not easy to do this just "desk" running Mach3 as you need to see where an actual tool is and you will need to understand simple G-code commands like G00 and G01.

Mach3 can be used without a detailed understanding of this chapter but you will find that using its concepts makes setting up jobs on your machine is very much quicker and more reliable.

## 7.1 The definition of a part

Figure 7.1 shows a partially dimensioned drawing of the end of a shaft with a shoulder and diameter for mounting a bearing, a thread for retaining it and a smaller diameter for a driving pulley. We will use metric units for this section of the manual as the numbers are
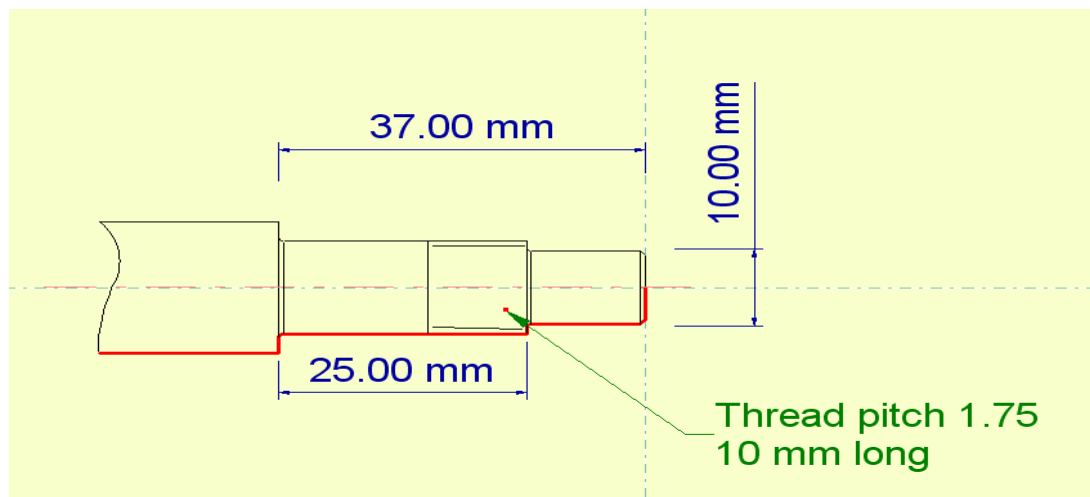


**Figure 7.1 – A shaft end dimensioned for manual turning**

convenient.

Figure 7.2 shows the finished product with the nut in position.

A G-code program needs to tell the lathe how to move the tool or tools to machine the part. The issues with the two axes are somewhat different.

### 7.1.1 X axis – diameter/radius

The position of the tool in the X-direction sets the diameter of the various parts of the shaft end. X = 0.0 is, of course, an unambiguous position; the cutting point of the tool is exactly on the centreline of the machine. X = 10.0 can, however, have one of two meanings. It can be



**Figure 7.2 – The machined product**

to turn a part with a diameter of 10 mm or a radius of 10 mm (i.e. diameter of 20). You choose how you want Mach3 to interpret X values when you set it up (Config>Ports & Pins – Turn Options tab). Your choice depends on a few things.

- A given part program will have been produced for one of Radius or Diameter mode; this will depend on the post-processor used in the CAM system. If you are going to use existing part programs then you must use the mode for which they were post-processed.

- If you have done a lot of manual turning then you might be used to working with radial values and wish to continue doing so. In CAD/CAM, the coordinates of the profile to be turned will be distances from the centreline (i.e. radii). If so choose Radius mode

- On the other hand, when you measure a piece of work with a micrometer or callipers you will get the diameter. It is often handy that this is the value displayed on the X-axis DRO. If so choose Diameter mode.

So the choice is yours, neither is right or wrong. Once you have chosen, stick to the one mode. In this chapter we will assume the use of Diameter mode and explain the differences in setup if you opt for Radius.

### 7.1.2 *Z axis zero position*

As with Radius and Diameter mode, if you have an existing part program then this will determine what position on the part is Z = 0 and you will need to set up the stock in the machine so the part can be turned without waste. If you are producing your own part programs, either by CAD/CAM or Mach3 Wizards, then you can choose where Z = 0 will be on the part. The following gives some guidance.

If the machining is to be on a portion of the stock, such as the shaft end shown, then it is often convenient that the finished end as Z = 0. All machining will be done with negative-Z values. The stock must, of course, project far enough from the chuck or collet so that the entire length to be machined is clear of the jaws.

If machining is to produce a component like a pin or a screw then it is often convenient to use the position where the finished component will be parted-off (e.g. the head of the pin or screw) as Z = 0. In this case machining is done with positive-Z values and enough stock must project from the jaws to provide the finished length of the component.

In boring and internal threading operations the tool may be used inside the chuck body so Z = 0 could even be inside the spindle.

Once again there is no universal right or wrong method. Here we will initially assume that you choose Z = 0 at the tailstock end of the component and describe the differences in workflow for other methods. If you are a CNC mill user then you will recognise this approach as similar to using the top face of the un-machined stock as Z = 0. Using the parting-off position is analogous to the less common method of using the surface of a mill table as Z = 0.

## 7.2 The Controlled Point and tools

The *Controlled Point* is the X and Z coordinates of a point relative to the origin (X=0 Z=0) of the component. It is the value generally displayed by the axis DROs and the part program (G-code) moves specify where it should be (except with a G53 in the block). It is, of course, where the effective cutting point of the tool should be. There are two slightly subtle points in this statement.



**Figure 7.3 – An idealised tool tip**

#### 7.2.1.1 Effective cutting point

First let's look at what is meant by the
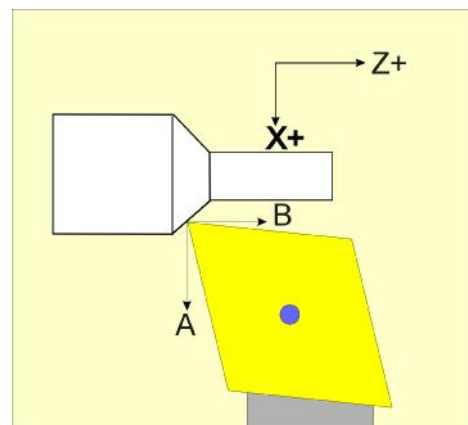
"effective cutting point".

Figure 7.3 shows a tool with a cutting tip that is quite sharp (i.e. has zero nose radius). "A"
shows the position it
cuts when facing and
"B" shows where it
cuts when turning. If it
cuts a chamfer then
this will be cut where
"A" and "B" meet so
the cuts will all be
correct. Such a tool
would, however, be
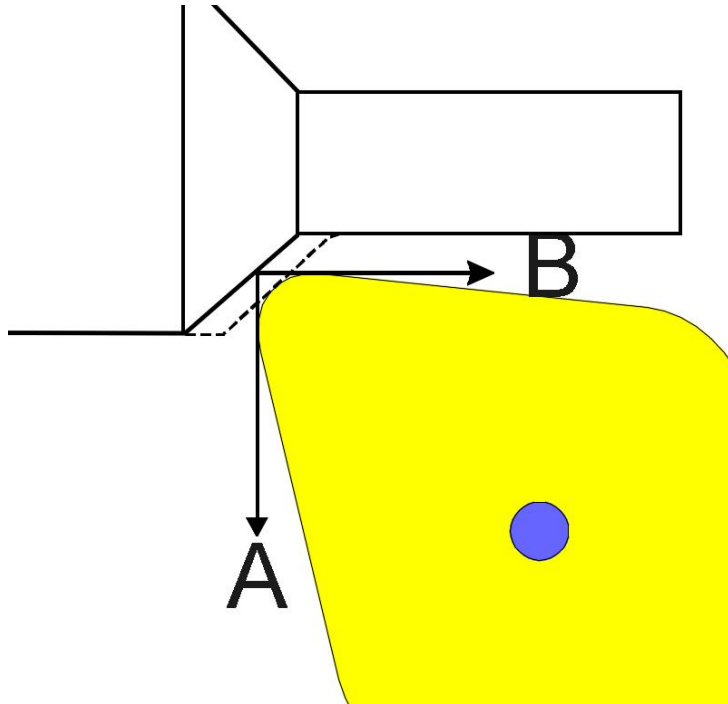much too weak to use
as its point would soon
snap off.

Figure 7.4 shows a real
tool with a rather
exaggerated nose
radius. "A" again
shows where it cuts
when facing and "B"
where it cuts when
turning. So the
intersection of the lines
is the effective cutting



**Figure 7.4 – Cutting with a tool with an exaggerated nose-radius**

point. You will, however, see that when it is used the chamfer does not get cut in the
expected place. CAD/CAM software and Mach3 provide ways of correcting for this but it is
why we cannot talk about a single *actual* cutting point of the tool; if the nose is radiused
then the cutting point depends on the angle at which the cut is being made.

### 7.2.1.2 The Program coordinates and Machine coordinates

We have seen that the controlled point is always the X and Z coordinates relative to the
origin of the part being machined.
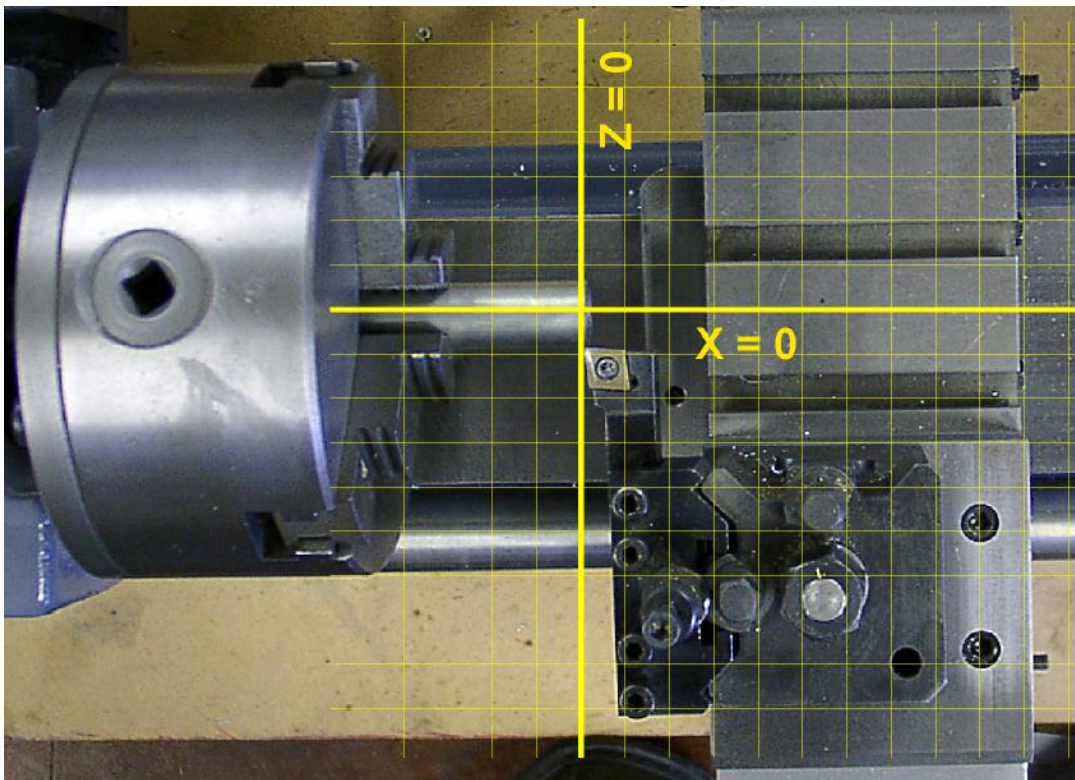


**Figure 7.5 – The Program coordinate system**

Figure 7.5 shows a piece of stock (actually 16 mm in diameter) with the tip on a tool positioned on the outside diameter at the faced-off end. Superimposed on the picture is a grid showing the centreline (X = 0) and Z = 0 as the faced end.

When you switch the computer and machine tool on Mach3 does not have any way of knowing where the cross-slide is positioned so does not know where X = 0 should be. Similarly, at switch on, it does not know where the saddle is located and whenever a piece of stock is put in the chuck Mach3 does not know how far it projects and cannot therefore know where Z = 0 should be.

There are different ways of telling Mach3 where the machine slides actually are and how this coordinate system (called the Machine Coordinate system) relates to the Program Coordinate system. These are all explained below. Which you choose to use will depend on personal preference, the type of work you are doing and whether your lathe has a Home switch on its X axis.

### 7.2.1.3 Using different tools

Very few turning jobs will be completed using just one tool. You must be able to change tools in such a way that a given tool is always located in the same position. This might be done by an automatic tool changer (for example a rotating turret) or by one of the proprietary quick change tool systems. A set of tools for use with the Dickson system is shown in figure 7.6.
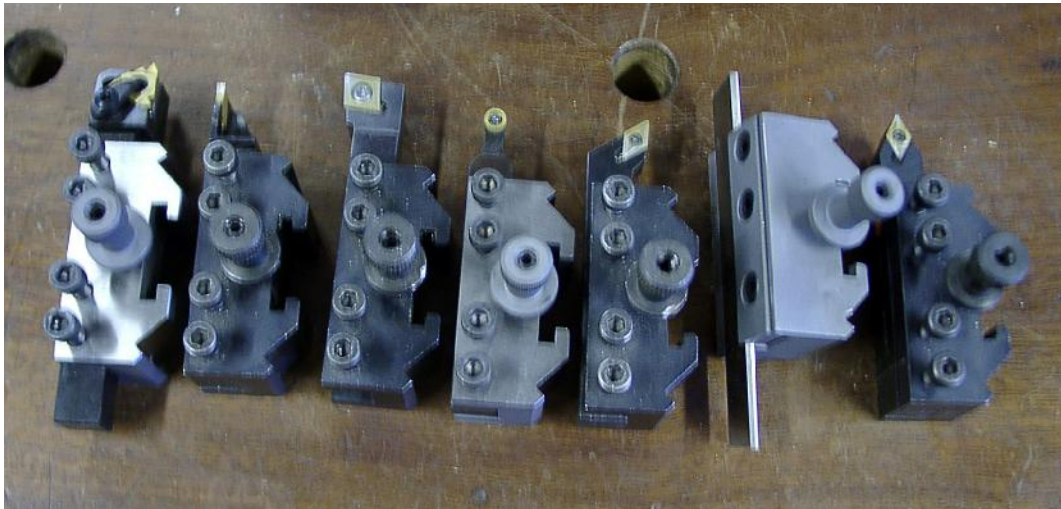


**Figure 7.6 – Some Dickson system toolholders**

Notice that, although the position of any given tool will be in the same position relative to the cross-slide each time it is used, the Effective Cutting Points of different tools will generally not be in the same position as each other; indeed, for left and right handed tools, they often **cannot** be mounted to cut in the same Z position. This is illustrated in figure 7.7. The saddle and cross-slide were not moved between the four photographs; only the tool holders were exchanged.

Mach3 has a Tool Table which stores information about up to 254 different tools to allow for these difference so the same Program Coordinate system can be used whichever tool you are using.

The values in the tool table are saved on disc so are remembered when the machine tool and computer are switched off. The tool table entries only need to be amended when a tool is changed or, for accurate work, to correct for wear of the tool tip.

### 7.2.1.4 Summary

To summarise this section:

- The Controlled Point is the coordinates to which most G-codes in a part program moves the machine and the Program Coordinate system values displayed on the axis DROs.

- Mach3 needs to discover where the saddle and cross-slide are set when it and the lathe are turned on. It keeps track of their subsequent motion using the Machine Coordinate system.

- Tools have an Effective Cutting Point when cutting parallel to the X or Z axes but the tool nose radius affects chamfered and radius cuts.

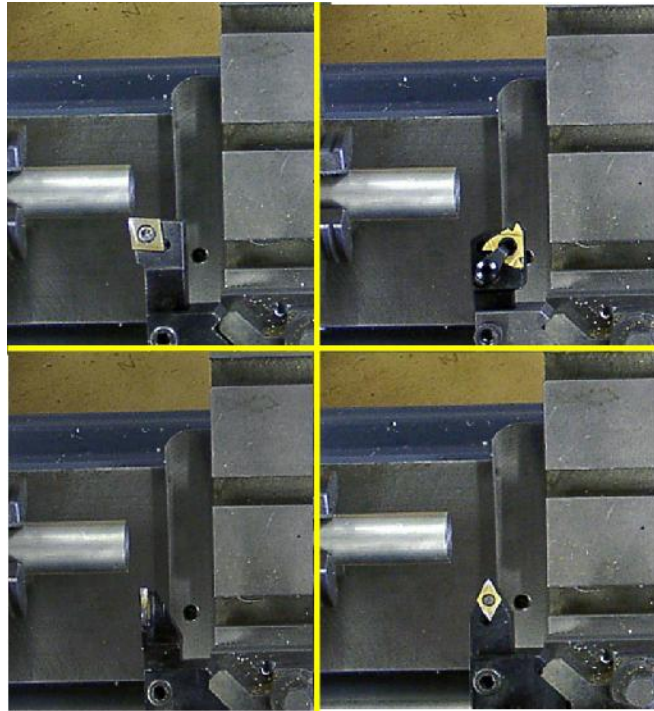- The Tool Table records the relative positions on the machine for each different tool to be used.



**Figure 7.7 – The tool tip position with different tools**

## 7.3 Referencing the machine

Referencing is the process of telling Mach3 where the saddle and cross-slide of the lathe actually are. You will generally do it whenever to switch on the lathe and start running Mach3 or after a problem such as an axis stalling through a problem with a part program or operator error like jogging into the tailstock. Mach3 (unlike some CNC systems) does not insist that you reference before doing anything else but is good practice to do so.

We will assume that the current tool is tool number 1 and this tool is in the tool holder or in the active position of the toolchanger. The significance of this will become clear when you have setup some tool table entries.



**Figure 7.8 – Controlling referencing and coordinate systems**

We first explain referencing with a machine without home switches and then explain the differences for an axis which does have a home switch.

Referencing is done on the Manual screen. Figure 7.8 shows the relevant buttons.

### 7.3.1 Referencing

#### 7.3.1.1 Setting Home if there is no home switch on an axis

Jog the X and Z axes to the location you want to be "home". Then click the *Set Home X* button followed by the *Set Home Z* button. This will define the physical position where you were when you *Set Home* as 0.0 in the Machine Coordinate system. (More precisely it is set to the value in the *Home Off*. column of the Config>Homing/Limits dialog which is 0 by default).

It is usually best that Z = 0.0 is near where the tailstock will normally be clamped to the bed and X = 0.0 be set with the cross slide as far from the centreline on the machine as possible. This means that in use machine Z and X coordinates will be negative numbers.

#### 7.3.1.2 Setting Home if there is a home switch on an axis

Unless you have a quick change tool holder which allows you to insert tools in a repeatable position or a tool changer then it is not worth using home switches; you will have to set the position of each tool as it is put into the machine by the touching process.

If an axis has a home switch defined, then the action of the *Set Home* button is different to that described above. As mentioned earlier under configuration, a single physical switch can be used by Mach3 as both a limit and a home switch. If you have a limit switch on the plus direction of the X axis (which is the direction where it is easiest to crash into a mechanical stop) then it is worthwhile using it as a home switch. A home switch can also be used on the Z axis but is much less useful as the projection of the stock from the chuck is a variable from job to job.

*Set Home* on an axis configured **with** a home switch will move the axis at a low speed in the plus direction until the switch is operated. The axis will then reverses direction until the switch is no longer operated and stop. The axis position in the machine coordinate system is then set to the value in the *Home Off.* column of Config>Homing/Limits (0.0 by default)

You need to think in which order to Home the axes. Generally it will be better to home X first the Z as this minimises the chance of the tool crashing into any stock in the machine. You can of course jog away from the stock before homing to avoid problems.

The advantage of using a good quality, and therefore repeatable, home switch on the X axis is that its position only needs to be setup once when the machine is configured.

### 7.3.2 *Relating the Program coordinates to Machine Coordinates*

Mach3 now knows that the toolpost is at a known (or in the case of no home switch an approximately known) position and has recorded the Machine Coordinates of this as 0.0.

The next thing to do is to define to Mach3 the X-distance between your tool's effective cutting point and the centreline of the machine. This is also called the X Offset between the Machine Coordinate system and the Program Coordinate system. It is also sometimes referred to as a Work or, in milling, a Fixture offset.

There are two ways of doing this; you can touch the tool tip on the circumference of a piece of known diameter stock or you can take a test cut and measure the diameter of the machined stock.

This procedure is usually done on the Auto Prep screen.

You will only have to do this once after referencing. The setting will stay valid until you reference again or, of course, switch off the machine. At present we are only thinking about one tool. Setting this Program (Work) offset will actually also allow s to use any tools without further setting up once we have put information about them in the Tool Table. Details of this are given later.

#### 7.3.2.1 Setting X by touching

This is a quick method, whose accuracy depends on your skill in judging when the tool is touching the stock and how much run-out there is with the chucked stock. Use the Trial Machining method for maximum accuracy.

You can use this method in setting a tool to a partially machined component.

The aim is to jog the axis so that the tool is just touching the circumference of a piece of stock with a known diameter. In practice you will not be able to see the gap as you jog X towards the stock, so you need a feeler gage of some sort.

Cigarette paper is traditional but the plastic foil-like packets from foodstuffs are also useful. Measure the thickness of this. Put the "gage" against the stock and very slowly jog the tool towards it until you feel the paper/plastic beginning to be trapped.

A better method, which does not risk damage to the tooltip by jogging carelessly, is to use a short length of small diameter ground bar or the shank of a small twist drill as a gage. About 3mm ($^1/_8$") diameter is convenient. Accurately measure the diameter of the gage. By eye, jog the X axis so that the gap between tooltip and stock is slightly less than the diameter of the gage. Now, as you very slowly jog away from the stock, roll the gage along the tapered gap formed by the relief on the tool edge. When it slips through the gap at the cutting point is exactly the diameter of the gage.



Figure 7.9 shows a situation where the gap is still slightly too small.

The beauty of the technique is that jogging too far does not crash the tool into the stock.

**Figure 7.9 – Touching with a cylindrical slip gage**

Having moved the axis to the "touch" you can calculate the correct offset for the tool. Let the actual stock diameter be *D* and the gage diameter or thickness be *g*

Calculate the position of X *(Px)*:

**Mach3Turn in Radius mode:**

$$Px = g + (D / 2)$$

**Mach3Turn in Diameter mode:**

$$Px = (2 \times g) + D$$

Now click the X axis DRO and use the keyboard to type in the value of *Px* you have calculated. Press *Enter*.

You can make some allowance for the stock not being held concentrically in the chuck by taking three or four values of Px at different angular positions and then averaging. It is, however, almost certainly quicker to use the trial machining method if you want this degree of accuracy.

### 7.3.2.2 Setting X by trial machining

Chuck a short length of round stock (a diameter of 12 to 16 mm or ½" is fine). Using slow jogging and/or MDI take a fine turning cut off the diameter. You only need to turn a length of 6 mm ($^1/_4$") Jog the Z axis, taking great care not to move X, so you can measure the turned diameter. Call this *d*.

Type this value into the X axis DRO and click *Enter*.

## 7.4    Chucking stock and setting Z Program Coordinate

The final stage before we can run a part-program to make a part is to set the Z offset between the Machine Coordinates and the Program Coordinates so that Z = 0 is at the appropriate place on the stock. Remember that where Z = 0 will depend on how the part program has been written. You need to check this if you are unsure.

### 7.4.1    *Z = 0 at tailstock end of part*

This is probably the easier situation. Look at the design of the part or the G-code in the part program and decide the most negative Z value that will be used. The limits of movement of both axes are displayed on the Auto Cycle screen (see figure 7.10). The stock must protrude from the chuck by at least this amount. Add 6mm (¼") for safety and jog the Z axis so it is this
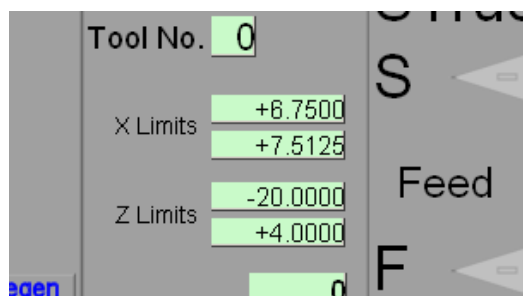


**Figure 7.10 – Program Extents**

distance from the chuck jaws with X near to zero.

Insert the stock in the chuck so that it is touching the side cutting edge of the tool and tighten the chuck.

If the end of the stock is already clean and does not need facing then click the *Zero* button for the Z axis.

If you do need to face the end of the stock then jog X out to larger than the stock diameter and jog Z so that the tool will take a small facing cut. Face the end, repeating the cut if it does not clean up. Then click the *Zero* button for the Z axis.

### 7.4.2   Z = 0 at parting-off point on part

In this situation you again look at the design or G-code but looking for the largest positive Z value used. This will correspond to the program Z coordinate of tailstock end of the finished part. Jog the tool so its Z position is at least this distance from the chuck jaws with a additional safety margin of 12 mm (½" ) or so – you need more than the previous case to allow for the groove of the parting-off cut.

If the end of the stock is already clean and does not need facing then type the calculated largest Z value into the Z axis DRO and press *Enter*.

If you do need to face the end of the stock then jog X out to larger than the stock diameter and jog Z so that the tool will take a small facing cut. Face the end, repeating the cut if it does not clean up. Then type the calculated largest Z value into the Z axis DRO and press *Enter*.

### 7.4.3   Repetition work

If you are making a series of identical parts then you can save making errors when entering the Z coordinate of the end of the part by putting the values in the Z DRO by the *Part Zero* button on the Auto Prep screen. Clicking this button is like typing the values in the Z DROs so you would do this to setup each part after clamping it in the chuck or facing it off.

**Note** that you need to arrange that the X axis is already at the position given in its part zero DRO when you use this button or your part diameters will be all wrong.

## 7.5   Using more than one tool

### 7.5.1   Introduction

We have seen that you can write a part program without worrying about the machine that will run it because Mach3 will convert Program Coordinate values into Machine Coordinate values using the information given when the machine is setup and the stock loaded into the chuck. We have also seen that there can be a problem with using different tools because their Effective Cutting Points will be in different places.

This section shows how to work with many tools. We will assume that you have replaceable carbide inserts for your tools because these can be exchanged without altering the nominal cutting position of the tool. If you use HSS tools and sharpen them by grinding, then you will need to do work resetting the tools offsets after each sharpening. We  will assume that you have a conventional "front" tool-post. Mach3 will function with a rear-post (e.g the machine shown in figure 4.7) but the detailed setup will depend on whether the tools are set to cut on the upper or lower faces when one is used.

### 7.5.2   Selecting a tool

It will generally be the responsibility of the part program to tell Mach3 which tool is to be used for any given set of G-code blocks. A Wizard or CAD/CAM system will select a tool and set the spindle speed, depth of cut, feedrate etc. to suit the operation being performed be it roughing, finishing, threading etc.

The process of selecting a tool, using G-code, is simpler in Mach3Turn than in Mach3Mill. This is because the control of a mill is designed to handle the situation where dozens of tools might be used. This is not the case on a lathe.

The tool changing process occurs when a part program uses the T word. This nominates which tool is going to be used and adjusts the Program Coordinates to allow for the position of the effective cutting point. To see how the Program Coordinates are offset we must look at the Tool Table.
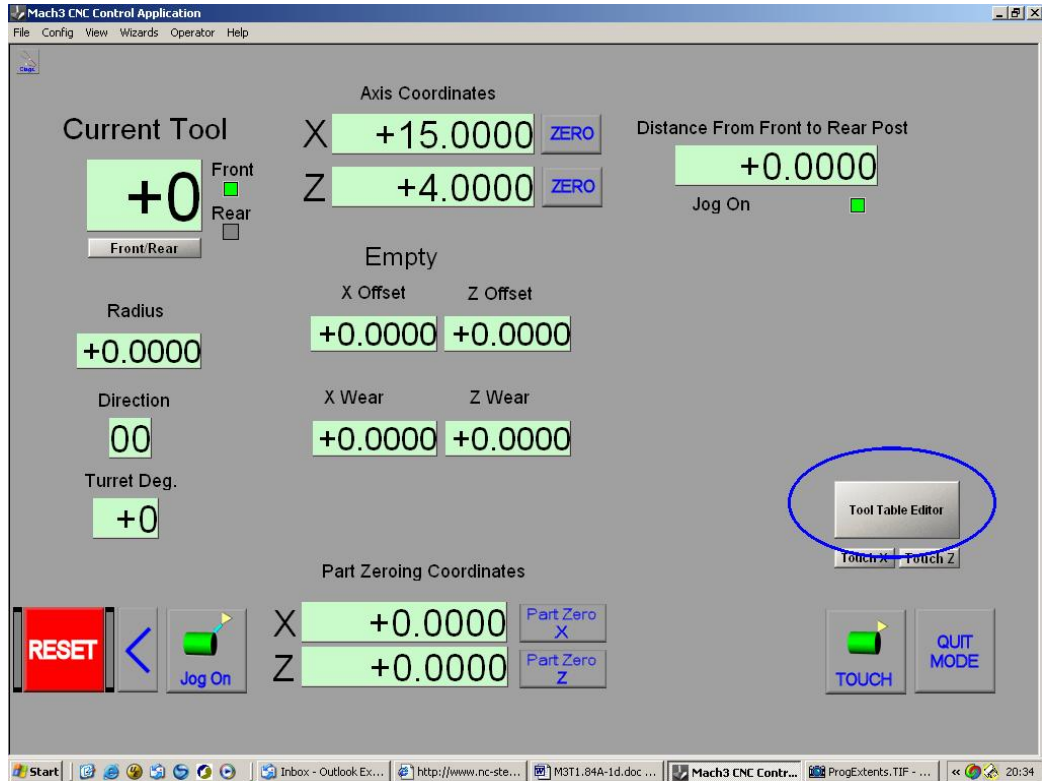


**Figure 7.11 – Tools Screen**

### 7.5.3    Tool table



**Figure 7.12 – The tool table editor**

The whole tool table is displayed by going via the Tool Table button on the Welcome screen and clicking the *Tool Table Editor* button (see figure 7.11).

An example of a Tool Table is shown in figure 7.12

Tool 0 is a dummy entry. Ignore it as most CAD/CAM systems will not allow you to select it in a part program. By scrolling the window you can access all the entries up to tool 99. You are advised to use the lower numbers as some CAD/CAM systems limit the highest tool number you can use.

### 7.5.3.1 Format of tool table

The meaning of the columns is as follows:

**Tool:** The table entry number to be used in the T word. The graphic can be used as a reminder of the shape of the tool; it is not used by Mach3. Choose it by cycling through the choices by clicking the cell in the table

**Description:** A free text description of the tool function. Displayed on screen by Mach3.

**Tip Direction:** This is a code defining the direction in which the Effective Cutting Point points which is used in tool tip radius compensation. See chapter 9 for details.

**Tip Radius:** This is used to compensate during angled cutting. See chapter 9 for details.

**X offset, Z offset, X wear, Z wear:** These are distances defining the position of the Effective Cutting Point of the tool.

**Turret angle:** not currently used by Mach3 but can be used by custom toolchanger macros.

There are several ways of describing the relative positions of tool tips (i.e. X offset, Z offset, X wear etc.) in a tool table. We will describe the Master Tool method. In this, one tool, usually Tool #1 is the Master Tool. It is the one used to set up the stock in the chuck. This means that its X and Z offsets and Wear offsets **will always** be zero. The offsets of all other tools are the differences in position of their cutting points from the tip of Tool #1.

You do not have to use the tool table in this way and, indeed, if you have a tool pre-setter for your tools you can set the offsets in the table relative to any datum position used by the pre-setter. If you understand the Master Tool principle you should have no difficulty in using the tool table in other ways.

### 7.5.3.2 Choosing which actual tool will be master

The master tool must be able to cut when facing and turning. Ideally you should put a tool which has light use in as the Master Tool in position 1 of the table. A finishing tool would therefore be a good choice. If you have a limited choice of tools then you can use your general turning tool. Make sure the toolholder is clearly labelled "Master – Tool #1".

### 7.5.3.3 Defining a facing/turning tool's entry in the tool table

The T word has the following form

    Txxyy

In this xx selects the tool to be used (e.g. to command a tool changer) and yy defines which entry in the tool table defines the offsets to be used.

For example T0101, selects tool 1 with offsets in table entry 1 (conventionally 0.0 for the Master Tool) and T0127 selects tool 1 but with the offsets in table entry 27

To set up a new tool's entry work on the Tools Screen (figure 7.11). M make the current tool be your master tool by MDI T0101. You can also type 0101 (or indeed 101) into the T DRO. If you do not have an automatic tool changer put the master tool into the toolpost.

If you have not already done so, reference the axes and set up the Program Coordinate X offset as shown in 7.3.2.2. and the Z coordinate as shown in 7.41. The Touching method for setting X is not generally accurate enough for setting up an entire tool table.

### 7.5.3.4 Setting up entry

Now choose the tool whose entry is to be setup. Make sure its centre height is exactly right. Decide which tool table number it will be and clearly label it. Suppose it is Tool 2. Enter 0202 into the Tool # DRO

Jog to take a fine turning cut off the circumference of the stock. Measure the diameter (and convert to the radius if you are in Radius mode) of the turned stock and type it into the *X Part Zeroing Coordinate* DRO. Do not change the X axis DRO.
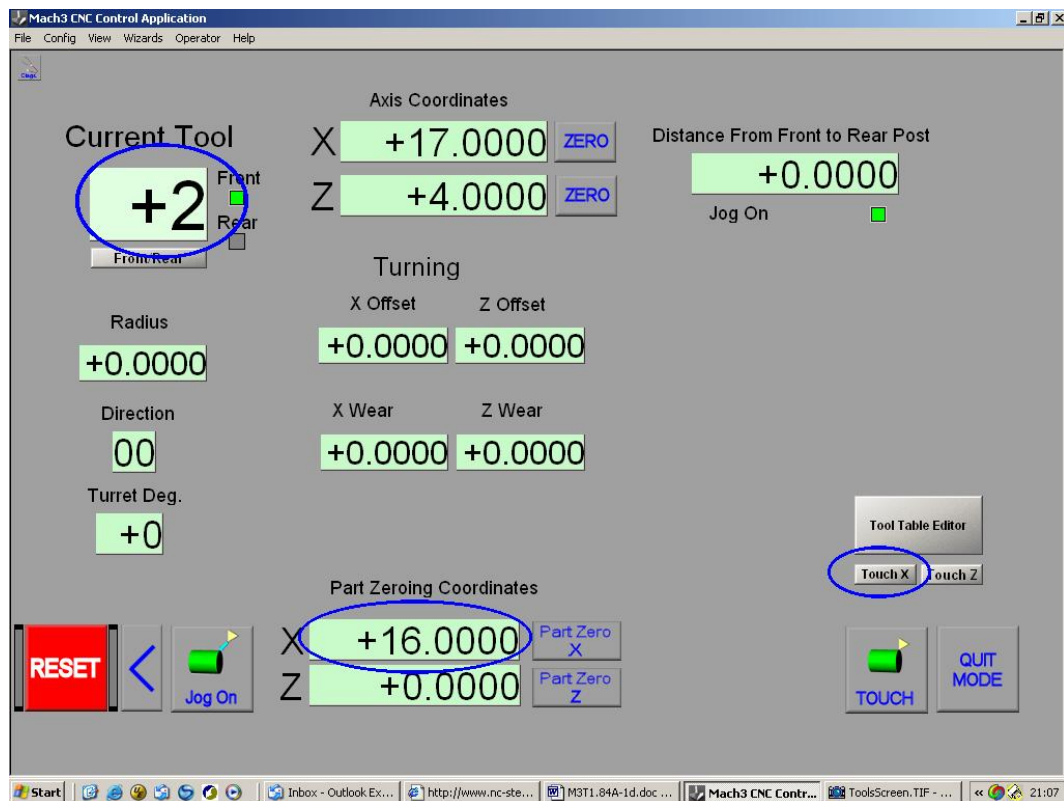


**Figure 7.14 – Setting X offset of tool #2**

Then click the Touch X button. These steps are indicated by rings in figure 7.14.

Now touch the side cutting edge of the tool on the faced end of the stock (see figure 7.15). For tools which do not make facing cuts (such as a turning or profiling tool see the method below).

Enter the gage thickness into the *Z Part Zeroing Coordinates DRO* and click *Touch Z*

Now click *Tool Table Editor*. Choose an icon and a name for the tool and a descriptive name for the tool. Ensure that *X Wear* and *Z Wear* are zero. Click *Apply* and *OK* to close the table.

You can now repeat this "Setting Up Entry" section for any other tools.
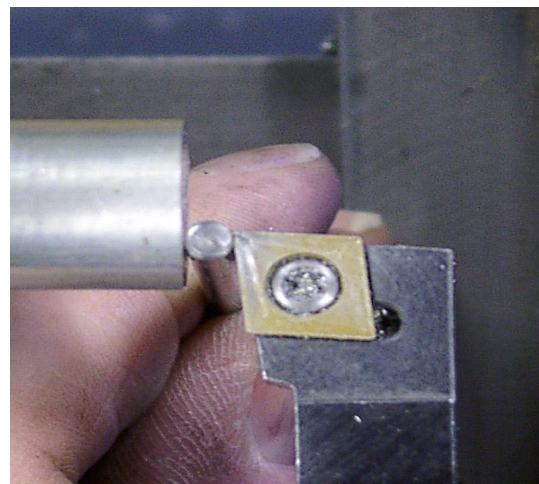


**Figure 7.15 – Touching Z with cylindrical gage**

The effect of this is that the offsets of a tool from the master tool are positive for a tool which cuts nearer to the centreline and nearer to the chuck.

### 7.5.3.5 Setting tool table for "special" tools

The techniques for special tools are illustrated in figure 7.14.

For tools which will not "face", for example, a threading tool, you need to judge the Z position where the tip cuts by eye. *Z Part Zeroing Coordinates DRO* should be set to 0.0 and *Touch Z* clicked. Fortunately the exact Z position of cuts with this sort of tool is often not too important.

For a parting or grooving tool, you will want to define the tailstock edge as the effective cutting point. Place a small flat piece of steel against the faced end and jog Z until the tool touches it. *Z Part Zeroing Coordinates DRO* should be set to 0.0 and *Touch Z* clicked.

For a boring bar, use a small flat piece of steel on the circumference and set the *X Offset* in the way given above having done a trial turning.



**Figure 7.16 – Setting special tools**

#### 7.5.3.6  Wear offsets

So far we have not put any values in the *X Wear* and *Z Wear* columns of the tool table. Whenever the offsets are used by Mach3, the *Z Wear* is subtracted from the *Z Offset* before it is used as the actual Z offset from the master tool. The same applies to X.

The intention is to allow for the fact that the main offsets are set up with a brand new tip in the tool but as it wears the size of the turned component will tend to increase. If you find that a part is oversize with a given worn tool calculate an *X Wear* offset as follows:

Measure the diametral amount that the component is oversize. Divide this by 2 in order to get the radial error.. Enter this error into the *X Wear* column of the tool that took the cut.

Z measurements are often less important than diameters but, if needed, you can use the *Z Wear* value in the same way. Store the distance by which a turned shoulder is above length in the *Z Wear*.. Notice you do not divide the error by two in this case.

The wear offsets are positive values if the worn tool cuts further from the centreline and further away from the chuck.

When you eventually index or replace the worn tip you should be able to get an accurate cut by resetting the *Wear* values to zero.

## 7.6  Rear toolposts

Most conventional lathes turn with a tool in a toolpost mounted on the cross-slide that is in front of the work as seen by the operator. The spindle runs in a clockwise direction when viewed looking towards the tailstock and the tool cuts on its top surface which can easily be seen.

This front toolpost arrangement will thus be common on lathes which have been retrofitted for CNC turning.

Custom CNC lathes are usually of the "slant bed" design where the tools are above and behind the work as seen by the operator. Figure 4.7 shows such an arrangement. This is termed a "rear" toolpost. The spindle rotation is often still clockwise so the tool cuts "underneath". This gives very good swarf clearance down the slant of the bed to the bottom of the machine and, often, onto a conveyor. Sometimes (and the Boxford in figure 4.7 is an example of this) cutting is done on the top of the tool with a counter-clockwise spindle. This has poorer swarf clearance but is easier for setting up tools.

A conventional or slant-bed machines can have toolposts in both the front **and** rear positions. These are generally carried on one slide and so move together.

Similar issues apply to vertical boring mills which often have two turrets on their carriage although the terms front and rear are less obvious. In this case the toolposts often can move independently of each other.

The setup information in this manual generally assumes a front toolpost machine.

The main issues you need to consider for other machines are:

- Jogging is usually arranged so that clockwise motion of the MPG and an upward jog key moves the X axis away from the operator.

- The X DRO will read a more and more positive value as the tool moves away from the centreline of the machine. Negative values are when the tool is beyond the centerline.

- Operating a slantbed lathe with a clockwise cutting spindle is like operating a conventional lathe lying on the floor underneath it! This means that ones interpretation of clockwise and counterclockwise (G02 and G03) arcs will be "wrong".

- A slantbed lathe that cuts with a counterclockwise spindle will cut right hand threads towards the tailstock. This has implications for the hand of threading tool required and the need for a relief groove at the start of the thread.

**Note:** The current version of Mach3Turn will support all these arrangements where there is a single "cross slide". You are however strongly advised not to switch dynamically between from and rear toolposts without re-referencing the machine and redefining the Progream Coordinates by touching.

The Customizing Mach3 wiki gives details of setup processes and requirements.

## 7.7 Behind the scenes

If you find that using the above explanations as recipes works for you then you might like to skip this section. Some people however would like to know what is actually happening within Mach3; if so read on.

### 7.7.1 Machine Coordinate system

As described above Mach3 has a Machine Coordinate System.

For a lathe with home switches, when the machine is referenced, this is fixed in relation to these switches with the axis zeros (or other fixed values in Config>Homing/Limits) being the saddle/cross-slide positioned at the switches.

For a lathe with no switches, then the Machine Coordinate system origin or (if Persistent DROs is checked in Config>Logic the coordinates when Mach3 was closed down) will correspond to where the saddle/cross-slide is when Mach3 is loaded.

### 7.7.2 Work (Fixture) offsets

Although Mach3Turn, in common with Mach3Mill, has 254 sets of "fixture offsets". These are used by Mach3Turn in a special way to implement front and rear toolposts so you will only exceptionally use the G54, G55 etc. commands.

The G54 offset system stores the relative position of the tool in the front toolpost and the G55 offset deals with the rear toolpost. This attempts to allow the setup of lathes with two toolposts to be optimised. The relationship of tools in them will not alter so the X axis only needs to be set up once after referencing. Whenever you switch to a tool defined to be in the rear toolpost the G55 offset will be updated to be related to the current G54 offset and the declared distance between the toolposts.

**In the current revision of Mach3Turn you are advised not to dynamically switch between toolposts.**

There is an option to reload the G54 offsets (usually to zero) from table entry G59.253 (this is set in Config>Logic) or you can use the *Zero World* buttons on the Manual screen to set both Machine Coordinates to zero at the current position of the saddle/cross-slide and to set the G54 offsets to zero.

The *Zero* buttons by the axis DROs are exactly equivalent to typing 0.0 into the DRO.

### 7.7.3 Program Coordinates

A point in the Program Coordinate system will be converted into machine coordinates as follows:

- In Diameter mode the Program coordinates are divided by 2 to make them radial distances.

- The entry in the tool table, selected by the T word is used and the Z Wear subtracted from the Z offset and X Wear subtracted from the X offset giving the actual tool offset values. These tool offset values are then added to the Program Coordinates. (**Note** that when displayed in DRO these values are shown as Diameter values in Diameter mode)

- The G52 local offsets are applied. **Note** that the value in G52X$vv$ is given in the current mode but and offset display DRO is always a radial distance.

- The G54/G55 offsets values (depending on whether the tool is in the front or rear post) are then applied to give values in the Machine Coordinate system. Note that these values are set in the current mode but displayed in DROs as a radial distance.

### 7.7.4 Other Radius and Diameter mode issues

When Machine Coordinates are displayed the X coordinate is always a radial distance but when moves are made in the Machine Coordinate system (G53 in the block) the value of the X word is treated as Diameter or Radius depending on the current mode..

Similarly, the offsets stored in the tool table and fixture table and those defined in Config>Homing/Limits for the Home switch position and the soft limits are all radial values although the relevant DROs display them in the current mode.

We strongly advise you **not** to change between Radius mode and Diameter mode (on Config>Ports & Pins Turn Options) without totally reconfiguring Mach3.

# 8. Advanced configuration options

> This chapter describes additional configuration options that you might find helpful when you have become familiar with the basic operation of your lathe.
>
> The *Customizing Mach3* wiki gives details of the techniques you can use to implement your own control panel, screen layouts and to write Wizards

## 8.1 Configure homing

### 8.1.1.1 Referencing speeds and direction

The Config>Homing/Limits dialog allows you to define what happens when a move to home or a reference operation (G28, G28.1 or pressing a *Set Home* screen button) is performed on axes with a Home switch defined. Figure 8.1 shows the dialog. The referencing speed is used to avoid crashing into the stop of an axis at full speed when looking for the reference switch. When you are referencing, Mach3 has no idea of the
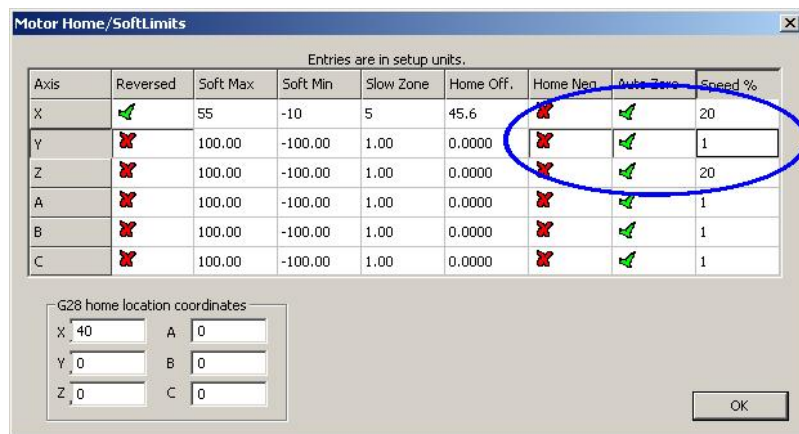


**Figure 8.1 – Homing direction speed and action**

position of an axis. The direction it moves in depends on the *Home Neg* check boxes. If the relevant box is checked then the axis will move in the minus direction until the Home input becomes active.

If the Home input is already active then the action will depend of the checked/unchecked state of *Home Sw Safety* on the Config>State dialog. If it is checked then nothing will happen. If it is unchecked then it will move in the plus direction.

Similarly if the *Home Neg* box is unchecked then the axis moves in the plus direction until the input is active and optinally in the minus direction if it is already active.

### 8.1.1.2 Position of home switches

If the *Auto Zero* checkbox is checked then the axis DROs will be set to the Reference/Home Switch location value defined in the *Home Off.* column (rather than actual Zero). This feature is less useful on a lathe than on a Mill.

It is, of course, necessary to have separate limit and reference switches if the reference switch is not at the end of an axis.

### 8.1.1.3 G28 positions

The machine coordinates in setup units to which the machine should co when a G28 is performed are defined for axes X to C on this dialog. For more details see Chapter 10.

## 8.1.2 Configure Backlash

Mach3 will attempt to compensate for backlash in axis drive mechanisms by attempting to approach each required coordinate from the same direction. While this is useful in

applications like drilling or boring, it cannot overcome problems with the machine in continuous cutting.

The Config>Backlash dialog allows you to give an estimate of the distance which the axis must back up by to ensure the backlash is taken up when the final "forward" movement is made. The speed at which this movement is to be made is also specified. See figure 8.2

**Note:** (a) These settings are only used when backlash compensation is enabled by the check box.

(b) Backlash compensation is a "last resort" when the mechanical design of your machine cannot be improved. It will prevent the effective opersation of yje machine in Constant Velocity mode and may give rise to unanticipated stalling in stepper system working anywhere near their performance limits!
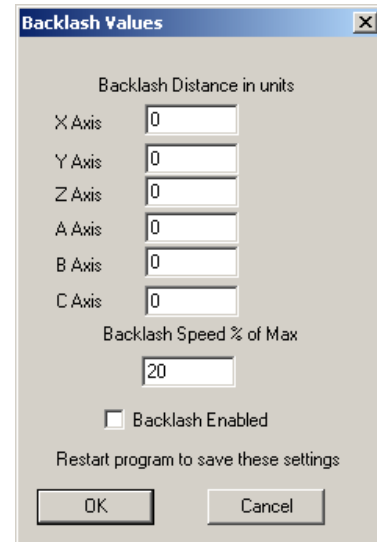


**Figure 8.2 – Config Backlash**

### 8.1.3 Configure Soft Limits

As discussed earlier most implementations of limit switches involve some compromises and hitting them accidentally will require intervention by the operator and may require the system to be reset and re-referenced. Soft limits can provide a protection against this sort of inconvenient accident.



**Figure 8.3 - Soft Limits configuration**

The software will refuse to allow the axes to move outside the declared range of the soft limits of the X and Z axes. These can be set in the range -999999 to + 999999 units for each axis. When jogging motion gets near to the limit then its speed will be reduced when inside an *Approach Safety* zone. These values are defined on the Config>Home/SoftLimits dialog. See figure 8.3.

The *Slow Zone* values are the distance from the limits at which the approach speed is automatically reduced. If this value is set too big you will reduce the effective working area of the machine. If they are set too small then you risk hitting the hardware limits.

The defined limits only apply when switched on using the *Software Limits* toggle.

If a part program attempts to move beyond a soft limit then it will raise an error.

### 8.1.4 Configure Initial State

Config>State opens a dialog which allows you to define the modes which are active when Mach3 is loaded (i.e. the initial state of the system). It is shown in figure 8.4.

**Motion mode:** *Constant velocity* sets G64, *Exact Stop* sets G61. For details of these option see Constant Velocity and Exact Stop in chapter 10. It is unusual to operate a lathe in *Exact Stop* mode.

**Distance mode:** *Absolute* sets G 90, *Inc* sets G91

**Active plane:** Should always be X-Z. This sets G18

**Figure 8.4 - Initial State configuration**

**I/J Mode:** In addition you can set the interpretation to be placed on I & J in arc moves. This is provided for compatibility with different CAM post-processor and to emulate other machine controllers. In *Inc IJ* mode I and J (the center point) are interpreted as relative to the starting point of a center format arc. This is compatible with NIST EMC. In *Absolute IJ* mode I and J are the coordinates of the center in the current coordinate system (i.e. after application of work, tool and G52/G92 offsets). If circles always fail to display or to cut properly (especially obvious by them being too big if they are far from the origin) then the IJ mode is not compatible with you part program.

**An error in this setting is the most frequent cause of questions from users when trying to cut circles.**

**Initialisation String:** is a set of valid G-codes to set the desired initial state of Mach3 when it is started. These are applied after the values set in the radio buttons above so may override them. Use the radio buttons wherever possible to avoid confusion. If *Use Init on ALL "Resets"* is checked then these codes will be applied however Mach2 is reset – e.g. after an EStop condition.

**Other check boxes:**

*Persistent Jog Mode*, if checked, will remember the Jog Mode you have chosen between runs of Mach3Turn.

*No FRO on Queue*, if checked, will mean that changes to the feed rate override DRO only take effect when the queue of moves waiting to be sent to the lathe is empty. Checking this will reduce the responsiveness of the override but avoid problems increasing the feed rate above 100% on stepper systems working near their speed and acceleration limits.

*Auto Screen Enlarge*, if checked, will cause Mach3 to enlarge any screen, and all the objects on it, if it has fewer pixels than the current PC screen mode so ensuring that it fills the entire screen area.

*Persistent Offsets*, if checked, will save the work and tool offsets in the permanent tables you have selected between runs of Mach2Mill. See also *Optional Offset Save*.

*Optional Offset Save*, if checked, will prompt to check, when you close Mach3, that you want to actually do any save requested in *Persistent Offsets*.

*Copy G54 from G59.253 on startup*, if checked, will re-initiaise the G54 offset (i.e. work offset 1) values from the work offset 253 values when Mach3 is started. Check this if you want to start up G54 to always be a fixed coordinate system (e.g. the machine coordinate

system) even if a previous user might have altered it and saved a non-standard set of values. This is generally most convenient with Mach3Turn

*Home Sw Safety,* if checked, inhibits movement if a Home switch is active when referencing is requested. It should only be used if all Home switches are positioned at one end of the axis travel

*Shortest Rot.,* if checked, optimises the positioning of all rotary axes. When using this option, the A, B or C DRO will not necessarily show the requested angle, for example if the A axis is at 370 and you request G00A0.0, the A will rotate back to 360. While the DRO will show the total of degrees moved since the job started, internally, the axis will only consider what the shortest path is to the requested angle treated modulo 360 and will go there. Being at 730 and performing a G00A1.0 will, for example, only move back 9 degrees and the DRO will read 721.

*Debug this run,* is for system testing only. It is not persistent.

*Use WatchDogs* Should normally be enabled. The Watchdogs will thrip an EStop condition if Mach3 does not seem to be running properly. They can be disabled, with a reduction in safety, if you find that they are spurioudle triggered on a slow machine, for example when running Wizards.

*Use MacroPump.* Should be disabled except on a machine with special controls which are controlled by the special MacroPump.m1s macro.

*Enhanced Pulsing*, if checked, will ensure the greatest accuracy of timing pulses (and hence smoothness of stepper drives) at the expense of additional central processor time. You should only select this if you are using a 1.2 GHz processor or faster.

*Charge Pump on in EStop* gives a continuous output on the charge pump pin. If you use the charge pump to enable/disable axis drives then you should leave this unchecked.

*Z is 2.5D on Output #6* does not apply to Mach3Turn

**Jog Increments in Cycle Mode**: The *Cycle Jog Step* button will load the values in the list into the *Step* DRO in turn. This is often more convenient than typing into the *Step* DRO. Code the special value 999 in an entry to switch to Cont Jog Mode.

## 8.1.5   *Configure other Logic items*

Miscellaneous features of Mach3 are configured in the Config>Logic dialog (see figure 8.5).

**G20/G21 Control**:  If Lock DROs to set up units is checked then even though G20 and G21 will alter the way X, Y, Z etc. words are interpreted (inch or millimetre) the DROs will always display in the Setup Unit system.

**Tool change:** An M6 tool change request can be ignored or used to call the M6 macros (q.v.). If Auto Tool Changer is checked then the M6Start/M6End macros will be called but Cycle Start does not need to be pressed at any stage.

**Persistent DROs:** if checked, then the axis DROs will have the same values on startup as when Mach2 is closed down. Note that the positions of the physical axes are unlikely to be preserved if the machine tool is powered down, especially with micro-stepper drives.

**Disable Gouge/Concavity checks**: if unchecked, then, during cutter compensation (G41 and G42), Mach3 will check if the tool diameter is too large to cut "inside corners" without gouging the work. Check the box to disable the warning.

**Debounce interval/Index Debounce:** Is the number of Mach3 pulses that a switch must be stable for its signal to be considered valid. So for a system running at 35,000 Hz , 100 would give about a 3 millisecond debounce  ($100 \div 35000 = .0029$ secs). The Index pulse and the other inputs have independent settings.

**M01 Control**: If checked then the default setting of the "Optional Stop" switch is zero. A button can be added to the  screens to control this dynamically if desired, see *Customizing wikl.*
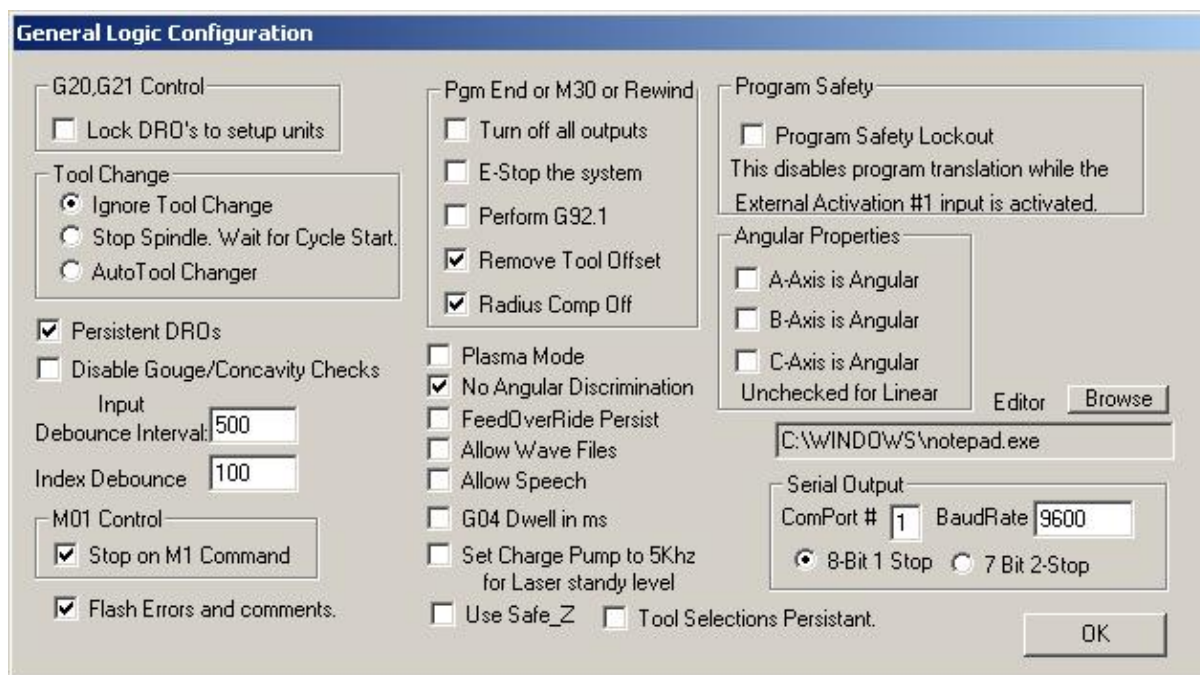
**Figure 8.5 — Logic Configuration dialog**

**Pgm end or M30 or Rewind**: defines action(s) to take place at end or a rewind of your part program. Check the required functions. *Caution:* Before checking the items to remove offsets and to perform G92.1 you should be absolutely clear on how these features work or you may find that the current position has coordinates very different from what you expect at the end of a program.

**Angular properties:** Generally not applicable to Mach3Turn. An axis defined as angular is measured in degrees (that is to say G20/G21 do not alter the interpretation of A, B, C words)

**Program safety:** When checked, enables Input #1 as a safety cover interlock.

**Editor:** The filename of the executable of the editor to be called by the G-code edit button. The Browse button allows a suitable file (e.g. C:\windows\notepad.exe) to be found.

**Serial output:** Defines the COM port number to be used for the serial output channel and the baud rate at which it should output. This port can be written to from VB script in a macro and can be used to control special functions of a machine (e.g. LCD display, tool-changers, axis clamps, swarf conveyor etc.) It is not the baudrate applicable to ModBus devices.

**Other checkboxes:**

*Plasma Mode*, Not applicable to Mach3Turn

*No Angular Discrimination*: This is also only relevant to constant velocity working. When unchecked Mach3 treats changes of direction whose angle is greater than the value set in the *CV Angular Limit* DRO as exact stop (even if CV mode is set) to avoid excessive rounding of sharp corners. Full details of Constant Velocity mode are given in chapter 10.

*FeedOveride Persists*, if checked, then the selected feed override will be retained at the end of a part program run.

*Allow wave files*: Allows a macro to play a wave file for audible warning

*Allow speech*: Allows the speech synthesiser for initialisation and Help system messages

*G04 Dwell param in Milliseconds***:** If you check this then the command G4 5000 will give a Dwell in running of 5 seconds. If the control is unchecked it gives a dwell of 1 hour 23 minutes 20 seconds!

*Set Charge Pump to 5KHz for Laser Standby Level*: Not applicable to Mach3Turn

*Use Safe_Z*: Not applicable to Mach3Turn.

*Tool Selections Persistent*, if checked, retains the last selected tool in the T DRO when Mach3Turn is reloaded.

## 8.2    How the Profile information is stored

When the Mach3.exe program is run it will prompt you for the Profile file to use. This will generally be in the Mach3 folder and will have the extension .XML. You can view and print the contents of Profile files with Internet Explorer (as XML is a mark-up language used on web pages)

Shortcuts are set up by the system installer to run Mach3.exe with default Profiles for a Mill and for Turning (i.e. Mach3Mill and Mach3Turn). You can if you wish create your own shortcuts each with a different Profile so one computer can control a variety of machine tools.

This is very useful if you have more than one machine and they require different values for the motor tuning, or have different limit and home switch arrangements.

You can either run Mach3.exe and choose from the list of available profiles or you can set up extra shortcuts that specify the profile to use.

In a shortcut, the profile to load is given in the "/p" argument in the Target of the shortcut properties. As an example you should inspect the Properties of the Mach3Mill shortcut. This can be done, for example, by right clicking the shortcut and choosing Properties from the menu.

An .XML file for a profile **can** be edited by an external editor but you are **very strongly** advised not to do this unless you are fully conversant with the meaning of each entry in the files and your editor will not insert unwanted invisible characters as some users have encountered very strange effects with mis-formatted files. Notice that some tags (e.g. the screen layout) are only created when a built-in default value is overridden using Mach3 menus. **It is much safer to use Mach3's configuration menus to update the XML profiles.**

**Note:** If you create your own profile(s) you must take care that the appropriate macro files are copied into the C:\Mach3\Macros\\*profilename* folder.

# 9. Tool tip radius compensation

Tool tip radius compensation is a feature of Mach3 which currently under development. Most CAD/CAM programs can be told the profile of your tool and will output part programs which cut the part profile which you have drawn by allowing for the tool tip shape. Because the CAD/CAM software has a better overall view of the shapes being cut it will generally do a better job than Mach3 can if you have a complicated profile.

## 9.1 Introduction to compensation

As we have seen Mach3 controls the movement of the Controlled Point. Figure 9.1 shows a "sharp pointed" tool cutting a diameter, a taper and facing.

Now if we have a radiused tip (say 3 mm radius like a 6 mm button tool) centred where the point was it will cut the diameter too small as shown at A in figure 9.2. This can be corrected by adding 3 mm to the X coordinate given by the program (this is of course 6 mm in Diameter mode). The facing cut would also be wrong by 3 mm, so 3 mm needs adding to the Z coordinate too. This can of course be done by putting 3.0 in the *X Offset* and *Z Offset* columns of the tool table for the current tool.
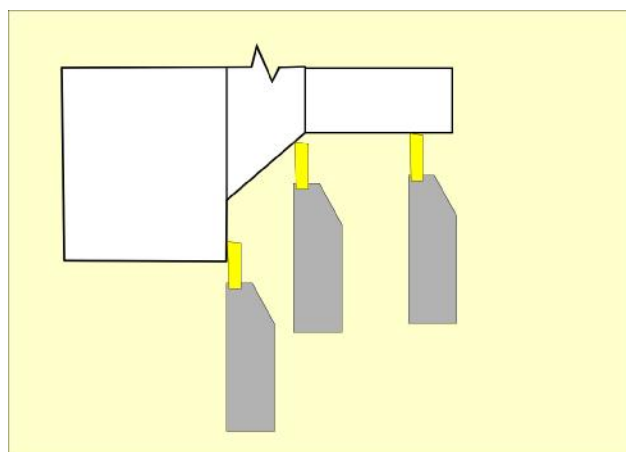
**Figure 9.1 – Turning with a "sharp" tool**

This will mean that the tool cuts as in positions B and D in figure 9.2. There is however a problem, at position C, on the chamfer. The centre of the tool is 4.24 mm away from the correct line of cut (i.e. the hypotenuse of a right-angled triangle with sides 3 mm). It should of course only be 3 mm away.
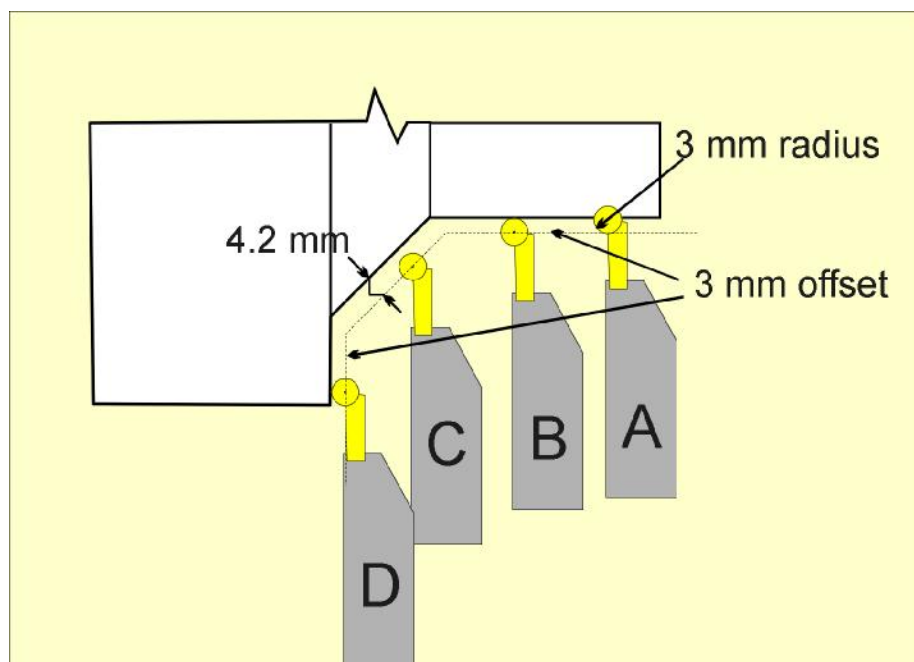
**Figure 9.2 – The offset toolpath for a radiused tool tip**

If Mach3 knows the angle of a chamfer then it is quite easy to calculate the sides of the triangle to get the hypotenuse 3 mm away from the work. Figure 9.3 shows a 60° chamfer and some more intricate profiling. The tool at position A is compensated by an offset of 2.6 mm in Z (3 x sin(60)) and 1.5 mm in X (3 x cos (60)) giving a hypotenuse of 3 mm.

There are, however some difficulties when the tool is near a corner. What Mach3 needs to know is the angle of the tangential contact line. This is not hard where two straight lines meet but is non-trivial when near curves like positions B and C. These curves may actually be made up of moves in many blocks of G-code and Mach3 can only know where the tool should touch the profile by looking ahead.
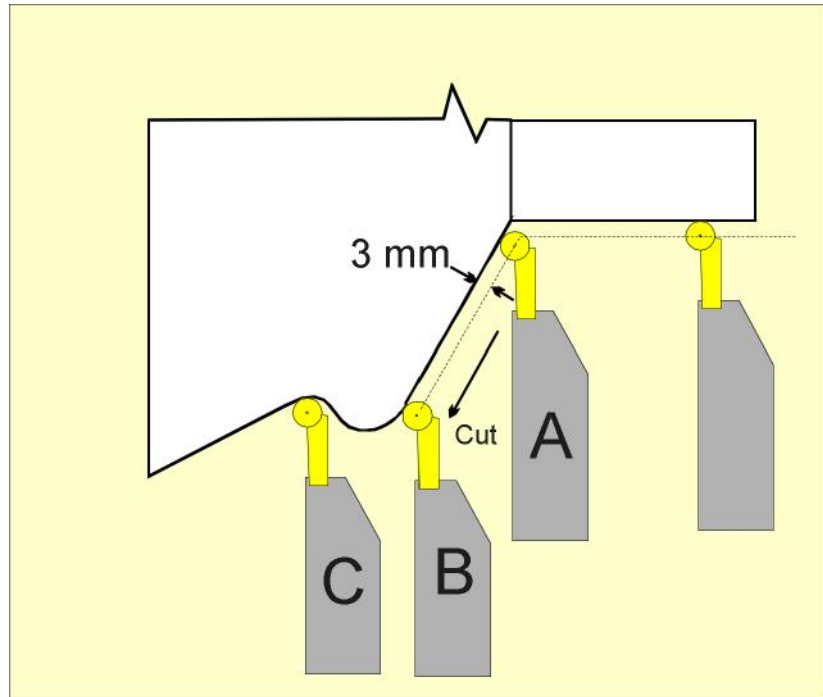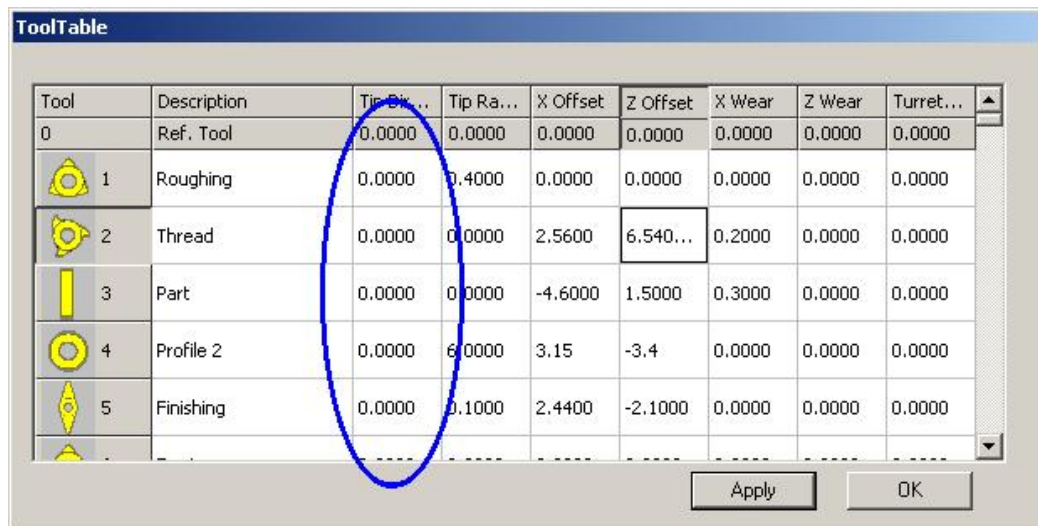


**Figure 9.3 – Tools contacting at different angles**

It is generally easiest to allow your CAD/CAM software to take account of this. This software has an overall view of the shapes. Mach3 only sees a local view.

If you are hand coding or using a Wizard it is useful for Mach3 to do the best with what it does know.

## 9.2    How compensation is specified

When you set up the X and Z offsets for a tool you are defining the effective cutting point.



**Figure 9.4 – Tool direction entry in table**

In order to do tool tip radius correction Mach3 needs to know the radius involved and where the centre of this radius is. The centre can, of course, easily be worked by using the radius and effective cutting point provided Mach3 knows the direction in which the tool was pointing when the offsets were setup. This direction is given in the Tip Direction column of the tool table – see figure 9.4.

In the example (figure 9.3) we would offset to the left when cutting from tailstock to chuck with a tool in the front toolpost as indicated by the arrow.

Figure 9.5 shows the numbering of tools in the front toolpost and figure 9.6 the comparable values for a rear tool post.
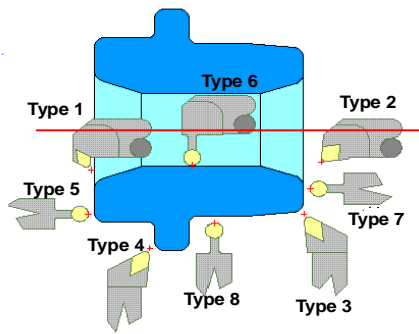


**Figure 9.5 – Front tool post directions**    **Figure 9.6 – Rear tool post directions**
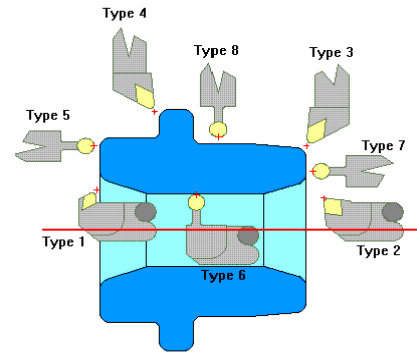
Compensation offsetting is switched on by G41 or G42. G41 offsets to the left and G42 offsets to the right. G40 switches tip radius compensation off.

G41 and G42 use the tool nose radius and direction of the current tool unless a D word is specified. If D is specified then its value is the index number in the tool table which specifies the radius and direction.

## 9.3    Potential difficulties

Obviously the tool will move into its new position in the mover after you switch compensation on with G41 or G42. You should make sure that it is not near the stock when you turn compensation on. A value of twice the tip radius in the Z direction and four times on the diameter is always safe.

Sometimes when you change the direction of a cut at a corner you will need to change from right to left hand compensation or vice-versa. You must be sure that the tool will not crash into the stock when this change is taking place.

# 10. Mach3 G- and M-code language reference

> This section defines the language (G-codes etc.) that are understood and interpreted by Mach3Turn.
>
> The description is oriented towards turning (e.g. concentrates on two axes) but Mach3Turn and Mach3Mill are a common body of code so many Mill functions can in fact be used on turning machines. Ask for advice from ArtSoft Corporation on particular applications.
>
> Information on unused axes is sometimes included in this chapter for completeness.

## 10.1 Some definitions

### 10.1.1 Linear Axes

The X and Z axes form a standard right-handed (although the direction of Y is generally unimportant) coordinate system of orthogonal linear axes. Positions are expressed using coordinates in these axes. A spindle and optionally a tailstock are mounted parallel to the Z axis. The centreline is generally made to be at X = 0.0.

### 10.1.2 Scaling input

It is possible to set up scaling factors for each axis. These will be applied to the values of X, Z, I, K and R words whenever these are entered. This allows the size of features machined to be altered and mirror images to be created - by use of negative scale factors.

The scaling is the first thing done with the values and things like feed rate are always based on the scaled values.

The offsets stored in tool and fixture tables are not scaled before use. Scaling may, of course, have been applied at the time the values were entered (say using G10).

### 10.1.3 Controlled Point

The controlled point is the notional point whose position and rate of motion are controlled. Commands like G00 and G01 specify by their X and Z words the new position of the controlled point and the Axis DROs display it.

### 10.1.4 Co-ordinated Linear Motion

To drive a tool along a specified path, a machining system must often co-ordinate the motion of several axes. We use the term "co-ordinated linear motion" to describe the situation in which, nominally, each axis moves at constant speed and all axes move from their starting positions to their end positions at the same time. This produces motion in a straight line, hence the word "linear" in the term. In actual motions, it is often not possible to maintain constant speed because acceleration or deceleration is required at the beginning and/or end of the motion. It is feasible, however, to control the axes so that, at all times, each axis has completed the same fraction of its required motion as the other axes. This moves the tool along the same path, and we also call this kind of motion co-ordinated linear motion.

Co-ordinated linear motion can be performed either at the prevailing feed rate, or at rapid traverse rate. If physical limits on axis speed make the desired rate unobtainable, all axes are slowed to maintain the desired path.

### 10.1.5          Feed Rate

The rate at which the controlled point or the axes move is nominally a steady rate which may be set by the user. In the Interpreter, the interpretation of the feed rate is as follows unless inverse time feed rate (G93) mode is being used:

For motion involving one or more of the linear axes (X, Z), the feed rate means length units per minute along the programmed linear XZ path.

### 10.1.6          Arc Motion

The pair of the linear axes ( XZ) can be controlled to move in a circular arc in the plane of that pair of axes.

The feed rate during arc motion is as described in Feed Rate above.

### 10.1.7          Coolant

Flood coolant and mist coolant may each be turned on independently. They are turned off together.

### 10.1.8          Dwell

A machining system may be commanded to dwell (i.e., keep all axes unmoving) for a specific amount of time. The most common use of dwell is to break and clear chips or for a spindle to get up to speed. The units in which you specify Dwell are either seconds or Milliseconds depending on the setting on Configure>Logic

### 10.1.9          Units

Units used for distances along the X and Z axes may be measured in millimetres or inches. Units for all other quantities involved in machine control cannot be changed. Different quantities use different specific units. Spindle speed is measured in revolutions per minute.. Feed rates are expressed in current length units per minute, in current length units per revolution of the spindle or in degrees per minute, as described above.

**Warning:** We advise you to check very carefully the system's response to changing units while tool and fixture offsets are loaded into the tables, while these offsets are active and/or while a part program is executing

### 10.1.10          Current Position

The controlled point is always at some location called the "current position" and Mach3 always knows where that is. The numbers representing the current position are adjusted in the absence of any axis motion if any of several events take place:

Length units are changed (but see Warning above)

Tool length offset is changed

Coordinate system offsets are changed.

### 10.1.11          Selected Plane

There is always a "selected plane", which for turning is the XZ-plane of the machining system.

### 10.1.12          Tool Table

Zero or one tool is assigned to each slot in the tool table.

### 10.1.13          Tool Change

Mach3 allows you to implement a procedure for implementing automatic tool changes using macros or to change the tools by hand when required.

### 10.1.14        *Path Control Modes*

The machining system may be put into any one of two path control modes: (1) exact stop mode, (2) constant velocity mode. In exact stop mode, the machine stops briefly at the end of each programmed move. In constant velocity mode, sharp corners of the path may be rounded slightly so that the feed rate may be kept up. These modes are to allow the user to control the compromise involved in turning corners because a real machine has a finite acceleration due to the inertia of its mechanism.

*Exact stop* does what it says. The machine will come to rest at each change of direction and the tool will therefore precisely follow the commanded path.

*Constant velocity* will overlap acceleration in the new direction with deceleration in the current one in order to keep the commanded feedrate. This implies a rounding of any corner but faster and smoother cutting. The lower the acceleration of the machine axes, the greater will be the radius of the rounded corner.

It is also possible to define an limiting angle so that changes in direction of more than this angle will always be treated as Exact Stop even though Constant Velocity is selected. This allows gentle corners to be smoother but avoids excessive rounding of sharp corners even on machines with low acceleration on one or more axes. This feature is not likely to be required on a lathe.

## 10.2   Interpreter interaction with controls

The standard screens do not have controls for all these functions but they can be implemented on custom screens. See *Mach3 Customization wiki*.

### 10.2.1        *Feed and Speed Override controls*

Mach3 commands which enable (M48) or disable (M49) the feed and speed override switches. It is useful to be able to override these switches for some machining operations e.g. threading. The idea is that optimal settings have been included in the program, and the operator should not change them.

### 10.2.2        *Block Delete control*

If the block delete control is ON, lines of code which start with a slash (the block delete character) are not executed. If the switch is off, such lines are executed.

### 10.2.3        *Optional Program Stop control*

The optional program stop control (see Config>Logic) works as follows. If this control is ON and an input line contains an M1 code, program execution is stopped at the end on the commands on that line until the *Cycle Start* button is pushed.

## 10.3   Tool File

Mach3 maintains a tool file for each of the 253 tools which can be used.

Each data line of the file contains the data for one tool. This allows the definition of the tool tip shape, offsets and radius.

## 10.4   The language of part programs

### 10.4.1        *Overview*

The language is based on lines of code. Each line (also called a "block") may include commands to the machining system to do several different things. Lines of code may be collected in a file to make a program.

A typical line of code consists of an optional line number at the beginning followed by one or more "words." A word consists of a letter followed by a number (or something that evaluates to a number). A word may either give a command or provide an argument to a

command. For example, `G1 X3` is a valid line of code with two words. "G01" is a command meaning "move in a straight line at the programmed feed rate," and "X3" provides an argument value (the value of X should be 3 at the end of the move). Most commands start with either G or M (for General and Miscellaneous). The words for these commands are called "G codes" and "M codes."

The language has two commands (M2 or M30), either of which ends a program. A program may end before the end of a file. Lines of a file that occur after the end of a program are not to be executed in the normal flow so will generally be parts of subroutines.

### 10.4.2 Parameters

A Mach3 machining system maintains an array of 10,320 numerical parameters. The forst 1000 parameters will be undefined when Mach3 is loaded but they can be used as working variables in a part-program.

### 10.4.3 Coordinate Systems

The machining system has an absolute coordinate system and 254 work offset (fixture) systems.

You can set the offsets of tools by `G10 L1 P~ X~ Z~.` The P word defines the tool number to be set.

You can set the offsets of the fixture systems using `G10 L2 P~ X~ Y~ Z~ A~ B~ C~` The P word defines the fixture to be set. The X, Y, Z etc words are the coordinates for the origin of for the axes in terms of the absolute coordinate system. Note however that the work offsets G54 and G55 are used buy the system to implement the distance between front and rear toolposts so should not be set by the user.

You can select one of the first seven work offsets by using G54, G55, G56, G57, G58, G59. Any of the 255 work offsets can be selected by `G59 P~` (e.g. `G59 P23` would select fixture 23). The absolute coordinate system can be selected by `G59 P0`.

You can offset the current coordinate system using G52, G92 or G92.3. This offset will then applied on top of work offset coordinate systems. This offset may be cancelled with G92.1 or G92.2.

You can make straight moves in the absolute machine coordinate system by using G53 with either G00 or G01.

## 10.5 Format of a Line

A permissible line of input code consists of the following, in order, with the restriction that there is a maximum (currently 256) to the number of characters allowed on a line.

> an optional block delete character, which is a slash "/" .
>
> an optional line number.
>
> any number of words, parameter settings, and comments.
>
> an end of line marker (carriage return or line feed or both).

Any input not explicitly allowed is illegal and will cause the Interpreter to signal an error or to ignore the line.

Spaces and tabs are allowed anywhere on a line of code and do not change the meaning of the line, except inside comments. This makes some strange-looking input legal. For example, the line `g0x +0. 12 34z 7` is equivalent to `g0 x+0.1234 z7`

Blank lines are allowed in the input. They will be ignored.

Input is case insensitive, except in comments, i.e., any letter outside a comment may be in upper or lower case without changing the meaning of a line.

### 10.5.1 Line Number

A line number is the letter N followed by an integer (with no sign) between 0 and 99999 written with no more than five digits (000009 is not valid, for example). Line numbers may be repeated or used out of order, although normal practice is to avoid such usage. A line number is not required to be used (and this omission is common) but it must be in the proper place if it is used.

### 10.5.2 Subroutine labels

A subroutine label is the letter O followed by an integer (with no sign) between 0 and 99999 written with no more than five digits (000009 is not permitted, for example). Subroutine labels may be used in any order but must be unique in a program although violation of this rule may not be flagged as an error. Nothing else except a comment should appear on the same line after a subroutine label.

### 10.5.3 Word

A word is a letter other than N or O followed by a real value.

Words may begin with any of the letters shown in figure 10.2. The table includes N and O for completeness, even though, as defined above, line numbers are not words. Several letters (I, J, K, L, P, R) may have different meanings in different contexts.

A real value is some collection of characters that can be processed to come up with a number. A real value may be an explicit number (such as 341 or -0.8807), a parameter value, an expression, or a unary operation value. Definitions of these follow immediately. Processing characters to come up with a number is called "evaluating". An explicit number evaluates to itself.

#### 10.5.3.1 Number

The following rules are used for (explicit) numbers. In these rules a digit is a single character between 0 and 9.

- A number consists of (1) an optional plus or minus sign, followed by (2) zero to many digits, followed, possibly, by (3) one decimal point, followed by (4) zero to many digits - provided that there is at least one digit somewhere in the number.

- There are two kinds of numbers: integers and decimals. An integer does not have a decimal point in it; a decimal does.

- Numbers may have any number of digits, subject to the limitation on line length. Only about seventeen significant figures will be retained, however (enough for all known applications).

- A non-zero number with no sign as the first character is assumed to be positive.

Notice that initial (before the decimal point and the first non-zero digit) and trailing (after the decimal point and the last non-zero digit) zeros are allowed but not required. A number written with initial or trailing zeros will have the same value when it is read as if the extra zeros were not there.

Numbers used for specific purposes by Mach3 are often restricted to some finite set of values or some to some range of values. In many uses, decimal numbers must be close to integers; this includes the values of indexes (for parameters and carousel slot numbers, for example), M codes, and G codes multiplied by ten. A decimal number which is supposed be close to an integer is considered close enough if it is within 0.0001 of an integer.

#### 10.5.3.2 Parameter Value

A parameter value is the hash character # followed by a real value. The real value must evaluate to an integer between 1 and 10320. The integer is a parameter number, and the value of the parameter value is whatever number is stored in the numbered parameter.

The # character takes precedence over other operations, so that, for example, #1+2 means the number found by adding 2 to the value of parameter 1, not the value found in parameter 3. Of course, #[1+2] does mean the value found in parameter 3. The # character may be repeated; for example ##2 means the value of the parameter whose index is the (integer) value of parameter 2.

### 10.5.3.3 Expressions and Binary Operations

An expression is a set of characters starting with a left bracket [ and ending with a balancing right bracket ]. In between the brackets are numbers, parameter values, mathematical operations, and other expressions. An expression may be evaluated to produce a number. The expressions on a line are evaluated when the line is read, before anything on the line is executed. An example of an expression is:

```
[1+acos[0]-[#3**[4.0/2]]]
```

Binary operations appear only inside expressions. Nine binary operations are defined. There

| Letter | Meaning |
|--------|---------|
| A | A-axis of machine |
| B | B-axis of machine |
| C | C-axis of machine |
| D | tool radius compensation number |
| F | feedrate |
| G | general function (see Table 5) |
| H | tool length offset index |
| I | X-axis offset for arcs<br>X offset in G87 canned cycle |
| J | Y-axis offset for arcs<br>Y offset in G87 canned cycle |
| K | Z-axis offset for arcs<br>Z offset in G87 canned cycle |
| L | number of repetitions in canned cycles/subroutines<br>key used with G10 |
| M | miscellaneous function (see Table 7) |
| N | line number |
| O | Subroutine label number |
| P | dwell time in canned cycles<br>dwell time with G4<br>key used with G10 |
| Q | feed increment in G83 canned cycle<br>repetitions of subroutine call |
| R | arc radius<br>canned cycle retract level |
| S | spindle speed |
| T | tool selection |
| U | Synonymous with A |
| V | Synonymous with B |
| W | Synonymous with C |
| X | X-axis of machine |
| Y | Y-axis of machine (not relevant in Turn) |
| Z | Z-axis of machine |

**Figure 10.2 - Word initial letters**

are four basic mathematical operations: addition (+), subtraction (-), multiplication (*), and division (/). There are three logical operations: non-exclusive or (OR), exclusive or (XOR), and logical and (AND). The eighth operation is the modulus operation (MOD). The ninth operation is the "power" operation (**) of raising the number on the left of the operation to the power on the right.

The binary operations are divided into three groups. The first group is: power. The second group is: multiplication, division, and modulus. The third group is: addition, subtraction, logical non-exclusive or, logical exclusive or, and logical and. If operations are strung together (for example in the expression [2.0/3*1.5-5.5/11.0]), operations in the first group are to be performed before operations in the second group and operations in the second group before operations in the third group. If an expression contains more than one operation from the same group (such as the first / and * in the example), the operation on the left is performed first. Thus, the example is equivalent to: [((2.0/3)*1.5)-(5.5/11.0)] which simplifies to [1.0-0.5] which is 0.5.

The logical operations and modulus are to be performed on any real numbers, not just on integers. The number zero is equivalent to logical false, and any non-zero number is equivalent to logical true.

### 10.5.3.4 Unary Operation Value

A unary operation value is either "ATAN" followed by one expression divided by another expression (for example ATAN[2]/[1+3]) or any other unary operation name followed by an expression (for example SIN[90]). The unary operations are: ABS (absolute value), ACOS (arc cosine), ASIN (arc sine), ATAN (arc tangent), COS (cosine), EXP (e raised to the given power), FIX (round down), FUP (round up), LN (natural logarithm), ROUND (round to the nearest whole number), SIN (sine), SQRT (square root), and TAN (tangent). Arguments to unary operations which take angle measures (COS, SIN, and TAN) are in degrees. Values returned by unary operations which return angle measures (ACOS, ASIN, and ATAN) are also in degrees.

The FIX operation rounds towards the left (less positive or more negative) on a number line, so that FIX[2.8]=2 and FIX[-2.8]=-3, for example. The FUP operation rounds towards the right (more positive or less negative) on a number line; FUP[2.8]=3 and FUP[-2.8]=-2, for example.

## 10.5.4    *Parameter Setting*

A parameter setting is the following four items one after the other:

- a pound character #
- a real value which evaluates to an integer between 1 and 10320,
- an equal sign = , and
- a real value. For example "#3 = 15" is a parameter setting meaning "set parameter 3 to 15."

A parameter setting does not take effect until after all parameter values on the same line have been found. For example, if parameter 3 has been previously set to 15 and the line #3=6 G1 x#3 is interpreted, a straight move to a point where x equals 15 will occur and the value of parameter 3 will be 6.

## 10.5.5    *Comments and Messages*

A line that starts with the percent character, %, is treated as a comment and not interpreted in any way. Printable characters and white space inside parentheses is a comment. A left parenthesis always starts a comment. The comment ends at the first right parenthesis found thereafter. Once a left parenthesis is placed on a line, a matching right parenthesis must appear before the end of the line. Comments may not be nested; it is an error if a left parenthesis is found after the start of a comment and before the end of the comment. Here is an example of a line containing a comment: G80 M5 (stop motion)

Comments do not cause the machining system to do anything.

A comment contains a message if `MSG,` appears after the left parenthesis and before any other printing characters. Variants of `MSG,` which include white space and lower case characters are allowed. Note the comma which is required. The rest of the characters before the right parenthesis are considered to be a message to the operator. Messages are displayed on screen in the "Error" intelligent label.

### 10.5.6 Item Repeats

A line may have any number of G words, but two G words from the same modal group may not appear on the same line.

A line may have zero to four M words. Two M words from the same modal group may not appear on the same line.

For all other legal letters, a line may have only one word beginning with that letter.

If a parameter setting of the same parameter is repeated on a line, `#3=15 #3=6`, for example, only the last setting will take effect. It is silly, but not illegal, to set the same parameter twice on the same line.

If more than one comment appears on a line, only the last one will be used; each of the other comments will be read and its format will be checked, but it will be ignored thereafter. It is expected that putting more than one comment on a line will be very rare.

### 10.5.7 Item order

The three types of item whose order may vary on a line (as given at the beginning of this section) are word, parameter setting, and comment. Imagine that these three types of item are divided into three groups by type.

The first group (the words) may be reordered in any way without changing the meaning of the line.

If the second group (the parameter settings) is reordered, there will be no change in the meaning of the line unless the same parameter is set more than once. In this case, only the last setting of the parameter will take effect. For example, after the line `#3=15 #3=6` has been interpreted, the value of parameter 3 will be 6. If the order is reversed to `#3=6 #3=15` and the line is interpreted, the value of parameter 3 will be 15.

If the third group (the comments) contains more than one comment and is reordered, only the last comment will be used.

If each group is kept in order or reordered without changing the meaning of the line, then the three groups may be interleaved in any way without changing the meaning of the line. For example, the line `g40 g1 #3=15 (so there!) #4=-7.0` has five items and means exactly the same thing in any of the 120 possible orders - such as `#4=-7.0 g1 #3=15 g40 (so there!)` - for the five items.

### 10.5.8 Commands and Machine Modes

Mach3 has many commands which cause a machining system to change from one mode to another, and the mode stays active until some other command changes it implicitly or explicitly. Such commands are called "modal". For example, if coolant is turned on, it stays on until it is explicitly turned off. The G codes for motion are also modal. If a G01 (straight move) command is given on one line, for example, it will be executed again on the next line if one or more axis words is available on the line, unless an explicit command is given on that next line using the axis words or cancelling motion.

"Non-modal" codes have effect only on the lines on which they occur. For example, G04 (dwell) is non-modal.

## 10.6    Modal Groups

**The modal Groups for G codes are**
- group 1 = {G00, G01, G02, G03, G38.2, G80, G81, G82, G84, G85, G86, G87, G88, G89} motion
- group 2 = {G17, G18, G19} plane selection
- group 3 = {G90, G91} distance mode
- group 5 = {G93, G94} feed rate mode
- group 6 = {G20, G21} units
- group 7 = {G40, G41, G42} cutter radius compensation
- group 8 = {G43, G49} tool length offset
- group 10 = {G98, G99} return mode in canned cycles
- group 12 = {G54, G55, G56, G57, G58, G59, G59.xxx} coordinate system selection
- group 13 = {G61, G61.1, G64} path control mode

**The modal groups for M codes are:**
- group 4 = {M00, M01, M02, M30} stopping
- group 6 = {M06} tool change
- group 7 = {M03, M04, M05} spindle turning
- group 8 = {M07, M08, M09} coolant (special case: M07 and M08 may be active at the same time)
- group 9 = {M48, M49} enable/disable feed and speed override controls

**In addition to the above modal groups, there is a group for non-modal G codes:**
- group 0 = {G04, G10, G28, G30, G53, G92, G92.1, G92.2, G92.3}

**Figure 10.3 - Modal groups**

Modal commands are arranged in sets called "modal groups", and only one member of a modal group may be in force at any given time. In general, a modal group contains commands for which it is logically impossible for two members to be in effect at the same time - like measure in inches vs. measure in millimetres. A machining system may be in many modes at the same time, with one mode from each modal group being in effect. The modal groups are shown in figure 10.3.

For several modal groups, when a machining system is ready to accept commands, one member of the group must be in effect. There are default settings for these modal groups. When the machining system is turned on or otherwise re-initialized, the default values are automatically in effect.

Group 1, the first group on the table, is a group of G codes for motion. One of these is always in effect. That one is called the current motion mode.

It is an error to put a G-code from group 1 and a G-code from group 0 on the same line if both of them use axis words. If an axis word-using G-code from group 1 is implicitly in effect on a line (by having been activated on an earlier line), and a group 0 G-code that uses axis words appears on the line, the activity of the group 1 G-code is suspended for that line. The axis word-using G-codes from group 0 are G10, G28, G30, and G92.

## 10.7    G Codes

G codes of the Mach3 input language are shown in figure 10.4 and are the described in detail.

The descriptions contain command prototypes, set in `courier` type.

In the command prototypes, the tilde (~) stand for a real value. As described earlier, a real value may be (1) an explicit number, 4.4, for example, (2) an expression, [2+2.4], for example, (3) a parameter value, #88, for example, or (4) a unary function value, acos[0], for example.

| Summary of G-codes | |
|---|---|
| G00 | Rapid positioning |
| G01 | Linear interpolation |
| G02 | Clockwise circular/helical interpolation |
| G03 | Counterclockwise circular/Helical interpolation |
| G04 | Dwell |
| G10 | Coordinate system origin setting |
| G15/G16 | Polar Coordinate moves in G0 and G1 |
| G17 | XY Plane select |
| G18 | XZ plane select |
| G19 | YZ plane select |
| G20/G21 | Inch/Millimetre unit |
| G28 | Return home |
| G28.1 | Reference axes |
| G30 | Return home |
| G32 | Threading |
| G40 | Cancel nose radius compensation |
| G41/G42 | Start nose radius compensation left/right |
| G50 | Reset all scale factors to 1.0 |
| G51 | Set axis data input scale factors |
| G52 | Temporary coordinate system offsets |
| G53 | Move in absolute machine coordinate system |
| G54 | Use fixture offset 1 |
| G55 | Use fixture offset 2 |
| G56 | Use fixture offset 3 |
| G57 | Use fixture offset 4 |
| G58 | Use fixture offset 5 |
| G59 | Use fixture offset 6 / use general fixture number |
| G61/G64 | Exact stop/Constant Velocity mode |
| G76 | Canned cycle – screw cutting |
| G77 | Canned cycle |
| G78 | |
| G80 | Cancel motion mode (including canned cycles) |
| G81 | Canned cycle - drilling |
| G82 | Canned cycle - drilling with dwell |
| G83 | Canned cycle - peck drilling |
| G90 | Absolute distance mode |
| G91 | Incremental distance mode |
| G92 | Offset coordinates and set parameters |
| G92.x | Cancel G92 etc. |
| G94 | Feed per minute mode |
| G95 | Feed per rev mode |
| G98 | Initial level return after canned cycles |
| G99 | R-point level return after canned cycles |

**Figure 10.4 - Table of G codes**

In most cases, if axis words (any or all of X~, Z~) are given, they specify a destination point. Axis numbers relate to the currently active coordinate system, unless explicitly described as being in the absolute coordinate system. Where axis words are optional, any omitted axes will have their current value. Any items in the command prototypes not explicitly described as optional are required. It is an error if a required item is omitted.

In the prototypes, the values following letters are often given as explicit numbers. Unless stated otherwise, the explicit numbers can be real values. For example, G10 L2 could equally well be written G[2*5] L[1+1]. If the value of parameter 100 were 2, G10 L#100 would also mean the same. Using real values which are not explicit numbers as just shown in the examples is rarely useful.

If `L~` is written in a prototype the "~" will often be referred to as the "L number". Similarly the "~" in `H~` may be called the "H number", and so on for any other letter.

If a scale factor is applied to any axis then it will be applied to the value of the corresponding X, Z word and to the relevant I, K or R words when they are used.

### 10.7.1 *Rapid Linear Motion – G00*

(a) For rapid linear motion, program `G00 X~ Z~`, where the axis words are optional, except that at least one must be used. The G00 is optional if the current motion mode is G00. This will produce co-ordinated linear motion to the destination point at the current traverse rate (or slower if the machine will not go that fast). It is expected that cutting will not take place when a G00 command is executing.

It is an error if:

> all axis words are omitted.

If cutter radius compensation is active, the motion will differ from the above; see Cutter Compensation. If G53 is programmed on the same line, the motion will also differ; see Absolute Coordinates.

### 10.7.2 *Linear Motion at Feed Rate – G01*

(a) For linear motion at feed rate (for cutting or not), program `G1 X~ Z~` where the axis words are optional, except that at least one must be used. The G01 is optional if the current motion mode is G01. This will produce co-ordinated linear motion to the destination point at the current feed rate (or slower if the machine will not go that fast).

It is an error if:

- all axis words are omitted.

If cutter radius compensation is active, the motion will differ from the above; see Cutter Compensation. If G53 is programmed on the same line, the motion will also differ; see Absolute Coordinates.

### 10.7.3 *Arc at Feed Rate – G02 and G03*

A circular arc is specified using either G02 (clockwise arc) or G03 (counterclockwise arc). The axis of the circle or helix must be parallel to the Y-axis of the machine coordinate system. The axis (or, equivalently, the plane perpendicular to the axis) is selected with G18 (Y-axis, XZ-plane).

If cutter radius compensation is active, the motion will differ from the above; see Cutter Compensation.

Two formats are allowed for specifying an arc. We will call these the center format and the radius format. In both formats the G02 or G03 is optional if it is the current motion mode.

#### 10.7.3.1 Radius Format Arc

In the radius format, the coordinates of the end point of the arc in the selected plane are specified along with the radius of the arc. Program `G2 X~ Z R~` (or use G03 instead of G02). R is the radius. The axis words are all optional except that at least one must be used. The R number is the radius. A positive radius indicates that the arc turns through 180 degrees or less, while a negative radius indicates a turn of 180 degrees to 359.999 degrees.

It is an error if:

- both of the axis words for the axes of the selected plane are omitted,
- the end point of the arc is the same as the current point.

It is not good practice to program radius format arcs that are nearly full circles or are semicircles (or nearly semicircles) because a small change in the location of the end point will produce a much larger change in the location of the center of the circle (and, hence, the middle of the arc). The magnification effect is large enough that rounding error in a number

can produce out-of-tolerance cuts. Nearly full circles are outrageously bad, semicircles (and nearly so) are only very bad. Other size arcs (in the range tiny to 165 degrees or 195 to 345 degrees) are OK.

### 10.7.3.2    Center Format Arc

In the center format, the coordinates of the end point of the arc in the selected plane are specified along with the offsets of the center of the arc from the current location. In this format, it is OK if the end point of the arc is the same as the current point. It is an error if:

- when the arc is projected on the selected plane, the distance from the current point to the center differs from the distance from the end point to the center by more than 0.0002 inch (if inches are being used) or 0.002 millimetre (if millimetres are being used).

The center is specified using the I and K words. There are two ways of interpreting them. The usual way is that I and K are the center relative to the current point at the start of the arc. This is sometimes called *Incremental IJ mode.* The second way is that I and K specify the center as actual coordinates in the current system. This is rather misleadingly called *Absolute IJ mode.* The IJ mode is set using the Configure>State… menu when Mach3 is set up. The choice of modes are to provide compatibility with commercial controllers. You will probably find Incremental to be best. In Absolute it will, of course usually be necessary to use both I and K words unless by chance the arc's centre is at the origin.

*Note:* **If arcs come out at silly sizes on the toolpath display or the lathe then your part program is probably not compatible with the IJ Mode setting in Mach3.**

Program G02 X~ Z~ I~ K~ (or use G03 instead of G02). The axis words are all optional except that at least one of X and Z must be used. I and K are the offsets from the current location or coordinates - depending on IJ mode (X and Z directions, respectively) of the center of the circle. I and K are optional except that at least one of the two must be used. It is an error if:

- X and Z are both omitted,
- I and K are both omitted.

## 10.7.4        *Dwell – G04*

For a dwell, program G04 P~ . This will keep the axes unmoving for the period of time in seconds or milliseconds specified by the P number. The time unit to be used is set up on the Configure>Logic dialog. For example, with units set to Seconds, G4 P0.5 will dwell for half a second. It is an error if:

- the P number is negative.

## 10.7.5        *Set Coordinate System Data Tool and work offset tables - G10*

See details of tool and work offsets for further information on coordinate systems

To set the offset values of a tool, program
G10 L1 P~ X~ Z~ A~, where the P number must evaluate to an integer in the range 0 to 255 - the tool number - Offsets of the tool specified by the P number are reset to the given. The A number will reset the tool tip radius. Only those values for which an axis word is included on the line will be reset.

To set the coordinate values for the origin of a fixture coordinate system, program
G10 L2 P~ X~ Z~, where the P number must evaluate to an integer in the range 1 to 255 - the fixture number - (Values 1 to 6 corresponding to G54 to G59) and all axis words are optional. The coordinates of the origin of the coordinate system specified by the P number are reset to the coordinate values given (in terms of the absolute coordinate system). Only those coordinates for which an axis word is included on the line will be reset.

Note that the X coordinate is used in the G54 and G55 systems to handle front and rear tool post relative positions.

It is an error if:

- the P number does not evaluate to an integer in the range 0 to 255.

If origin offsets (made by G92 or G92.3) were in effect before G10 is used, they will continue to be in effect afterwards.

The coordinate system whose origin is set by a G10 command may be active or inactive at the time the G10 is executed.

The values set will not be persistent unless the tool or fixture tables are saved using the buttons on Tables screen.

Example: `G10 L2 P1 x3.5 Z17.2` sets the origin of the first coordinate system (the one selected by G54) to a point where X is 3.5 and Z is 17.2 (in absolute coordinates).

### 10.7.6        Plane Selection - G17, G18, and G19

Program G18 to select the XZ-plane which is the only relevant plane for Turn.

### 10.7.7        Length Units - G20 and G21

Program G20 to use inches for length units. Program G21 to use millimetres.

It is usually a good idea to program either G20 or G21 near the beginning of a program before any motion occurs, and not to use either one anywhere else in the program. It is the responsibility of the user to be sure all numbers are appropriate for use with the current length units.

### 10.7.8        Return to Home - G28 and G30

A home position is defined by Config>Softlimits & Homing. The values are in terms of the absolute coordinate system, but are in setup length units.

To return to home position by way of the programmed position, program
`G28 X~ Z~` (or use G30). All axis words are optional. The path is made by a traverse move from the current position to the programmed position, followed by a traverse move to the home position. If no axis words are programmed, the intermediate point is the current point, so only one move is made.

### 10.7.9        Reference axes - G28.1

Program `G28.1 X~ Z~` to reference the given axes. The axes will move at the current feed rate towards the home switch(es), as defined by the Configuration. When the absolute machine coordinate reaches the value given by an axis word then the feed rate is set to that defined by Config>Homing/Limits. Provided the current absolute position is approximately correct, then this will give a soft stop onto the reference switch(es).

### 10.7.10        Threading - G32

To cut a single pass of a thread program `G32 X~ Z~ F~`

Before the command the X and Z position of the controlled point should be set to give Z clearance at the start of the thread (two or three pitches should suffice with a reasonably fast Z axis). The F word specifies the feed per revolution (i.e. the pitch of the thread to be cut).

The Z word gives the finishing position of the thread and the X word (if present) gives the X value at the end of the thread.. If X is omitted then it remains unchanged through the cut (i.e. a parallel thread is cut).

For example (Diameter and G21 modes): Suppose a pieces of stock 12 mm in diameter is chucked with its faced end as Z=0.0 The preparatory moves could be:

```
G00 Z4
G00 X11.9 (position at cut depth with Z clearance)
```

A 1.5 mm pitch thread 0.05 mm deep 25 mm along the bar will be cut by

```
G32 Z-25 F1.5
```

And the tool is the retracted to a clear X position by

```
G00 X14
```

And returned to the starting clearance.

Mach3 will wait until it knows the spindle speed and position (using the Index signal) and the accelerate the Z axis in the air so that it is up to the specified "feed" when it cuts. If a change in X is specified then this will be executed as motion coordinated with Z giving a tapered thread (air and cut) from the starting position.

The next pass would require a calculation of the new Z start position, if an angled in-feed is required. The X movement needs to be calculated to give the desired chip size. Typically the cut depth will be reduced as the width of the chip rises as the thread is formed.

Multiple start threads are produced by adding an offset (e.g. hapf, one third etc. of a pitch) to the starting Z positions.

For imperial threads which are usually specified as "threads per inch" the feed is the reciprocal of the TPI e.g. 16 TPI is a pitch of 1 / 16 = 0.0625 inch.

G32 is ideal for the code output by a CAD/CAM post-processor, as the writer has total control of the thread. It is, however, very inconvenient for hand coding on account of the amount of calculation required to set X and Z correctly. The G76 canned threading cycle is provided to overcome this difficulty.

### 10.7.11 Nose Radius Compensation - G40, G41, and G42

To turn cutter radius compensation off, program G40. It is OK to turn compensation off when it is already off.

To turn nose radius compensation on left (i.e., the cutter stays to the left of the programmed path when the tool radius is positive), program G41 D~ To turn cutter radius compensation on right (i.e., the cutter stays to the right of the programmed path when the tool radius is positive), program G42 D~ The D word is optional; if there is no D word, the radius of the tool currently in the spindle will be used. If used, the D number should normally be the slot number of the tool in the spindle, although this is not required. It is OK for the D number to be zero; a radius value of zero will be used.

It is an error if:

- the D number is not an integer, is negative or is larger than the number of carousel slots,
- cutter radius compensation is commanded to turn on when it is already on.

The behavior of the machining system when cutter radius compensation is ON is described in the chapter 8. Notice the importance of programming adequate clearance at the start and end of moves and at changes of direction.

### 10.7.12 Scale factors G50 and G51

To define a scale factor which will be applied to an X, Z, A, B, C, I & J word before it is used program G51 X~ Z~ A~ B~ C~ where the X, Z etc. words are the scale factors for the given axes. These values are, of course, never themselves scaled.

It is not permitted to use unequal scale factors to produce elliptical arcs with G2 or G3.

To reset the scale factors of all axes to 1.0 program G50

### 10.7.13 Temporary Coordinate system offset – G52

To offset the current point by a given positive or negative distance (without motion), program G52 X~ Z , where the axis words contain the offsets you want to provide. The axis words are optional, except that at least one must be used. If an axis word is not used for a given axis, the coordinate on that axis of the current point is not changed. It is an error if:

- all axis words are omitted.

G52 and G92 use common internal mechanisms in Mach3 and may not be used together.

When G52 is executed, the origin of the currently active coordinate system moves by the values given.

The effect of G52 is cancelled by programming G52 X0 Z0 etc.

Here is an example. Suppose the current point is at X = 4 in the currently specified coordinate system, then G52 X7 sets the X-axis offset to 7, and so causes the X-coordinate of the current point to be -3.

The axis offsets are always used when motion is specified in absolute distance mode using any of the fixture coordinate systems. Thus all fixture coordinate systems are affected by G52.

### 10.7.14        *Move in Absolute Coordinates - G53*

For linear motion to a point expressed in absolute coordinates, program G01 G53 X~ Z~ (or similarly with G00 instead of G01), where all the axis words are optional, except that at least one must be used. The G00 or G01 is optional if it is in the current motion mode. G53 is not modal and must be programmed on each line on which it is intended to be active. X will always be a radial value irrespective of the mode (Radius or Diameter) for which Mach3 is configured. This will produce co-ordinated linear motion to the programmed point. If G01 is active, the speed of motion is the current feed rate (or slower if the machine will not go that fast). If G00 is active, the speed of motion is the current traverse rate (or slower if the machine will not go that fast).

It is an error if:

- G53 is used without G0 or G1 being active,
- G53 is used while cutter radius compensation is on.

Note: that a G53 move with an un-referenced machine is likely to cause a crash as the machine coordinate system will contain undefined values. See chapter 7 for an overview of coordinate systems.

### 10.7.15        *Select Work Offset Coordinate System - G54 to G59 & G59 P~*

To select work offset #1, program G54, and similarly for the first six offsets. The system-number-G-code pairs are: (1-G54), (2-G55), (3-G56), (4-G57), (5-G58), (6-G59)

To access any of the 254 work offsets (1 - 254) program G59 P~ where the P word gives the required offset number. Thus G59 P5 is identical in effect to G58.

It is an error if:

- one of these G-codes is used while cutter radius compensation is on.

Fixture/Work offsets are not often explicitly used when turning. The X values of G54 and G55 are dynamically updated by Mach3 when touching or entering values into the axis DROs.

### 10.7.16        *Set Path Control Mode - G61, and G64*

Program G61 to put the machine into exact stop mode, or G64 for constant velocity mode. It is OK to program for the mode that is already active. These modes are described in detail above.

### 10.7.17        *Canned Cycle – High Speed Peck Drill G73*

The G73 cycle is intended for deep drilling or milling with chip breaking. See also G83. The retracts in this cycle break the chip but do not totally retract the drill from the hole. It is suitable for tools with long flutes which will clear the broken chips from the hole. This cycle takes a Q number which represents a "delta" increment along the Z-axis. Program

```
G73 X~ Z~ A~ B~ C~ R~ L~ Q~
```

Preliminary motion, as described in G81 to 89 canned cycles.

Move the Z-axis only at the current feed rate downward by delta or to the Z position, whichever is less deep.

Rapid back out by the distance defined in the *G73 Pullback* DRO on the Settings screen.

Rapid back down to the current hole bottom, backed off a bit.

Repeat steps 1, 2, and 3 until the Z position is reached at step 1.

Retract the Z-axis at traverse rate to clear Z.

It is an error if:

> the Q number is negative or zero.

## 10.7.18          *Canned Cycle – Threading G76*

Program G76 X~ Z~ Q~ P~ H~ I~ R~ K~ L~ C~ B~ T~ J~ to cut a complete thread.

> X - XEnd
> Z - ZEnd
> Q - Spring Passes. (optional)
> P - Pitch
> H - Depth of first pass
> I - Infeed angle
> R - XStart (optional)
> K - Z Start (optional)
> L - Chamfer (optional)
> C - X Clearance
> B - Depth Last Pass (optional)
> T - Taper (optional)
> J - Minimum depth per pass (optional)

The controlled point should give a suitable clearance for air cutting on the Z axis and have X equal to the crest diameter (radius). This is not necessary if R (the starting X value) and/or K (the starting Z value) are specified.

The length and depth of the thread are given by X and Z (in relation to the start values). If the "depth" is negative then the thread is taken as internal.

The C word gives the X clearance for the rapid return motion. It is always specified as positive and Mach3 handles the reversed direction for an internal thread

The P word gives the pitch of the thread (use 1/TPI for imperial threads)

The H word determines not only the first pass but, because the G76 ensures that subsequent passes cut the same volume of material per revolution determine the cuts for the entire thread and hence the number of passes.

The B word determines the depth of the last pass. If this is set too large then it can interact in an unexpected way with the constant volume calculations and the number of actual passes used.

The Q word gives the number of spring passes (i.e. cuts at the X end position).

The I word gives the infeed angle in degrees. This is Zero for radial infeed and typically 29 or 29.5 degrees for a unified or ISO metric thread or 27 degrees for 55 degree angle forms like BSW or BSF.

The values of words which are omitted are taken from the values set on the Turn Options tab of Config>Ports & Pins.

**Note:** This cycle is actually implemented by the macro M1076.M1S which can be inspected for implementation details. Note that one instance of this macro (usually in C:\Mach3\Macros) is used for all profiles.

### 10.7.19          *Canned cycle – Turning G77*

**Note:** This cycle is implemented by the macro M1077.M1S which can be inspected for implementation details. There is a separate instance of this macro for the profile that you are using

### 10.7.20          *Canned cycle – Facing G78*

**Note:** This cycle is implemented by the macro M1078.M1S which can be inspected for implementation details. There is a separate instance of this macro for the profile that you are using

### 10.7.21          *Cancel Modal Motion - G80*

Program G80 to ensure no axis motion will occur. It is an error if:

- Axis words are programmed when G80 is active, unless a modal group 0 G code is programmed which uses axis words.

### 10.7.22          *Canned Cycles - G81 to G89*

The canned cycles G81 through G89 have been implemented as described in this section. Two examples are given with the description of G81 below.

All canned cycles are performed with respect to the currently selected plane. In Turn XY should be selected.

All canned cycles use X, R, and Z numbers in the NC code. These numbers are used to determine X, R, and Z positions. The R (usually meaning retract) position is along the axis perpendicular to the currently selected plane (Z-axis for XY-plane). Some canned cycles use additional arguments.

Notice that non-zero X values are only valid with a twist drill if the toolholder is live (i.e. turns the drill). If the spindle turns the work then a boring bar is being used.

For canned cycles, we will call a number "sticky" if, when the same cycle is used on several lines of code in a row, the number must be used the first time, but is optional on the rest of the lines. Sticky numbers keep their value on the rest of the lines if they are not explicitly programmed to be different. The R number is always sticky.

In incremental distance mode: when the XY-plane is selected, X, and R numbers are treated as increments to the current position and Z as an increment from the Z-axis position before the move involving Z takes place. In absolute distance mode, the X, R, and Z numbers are absolute positions in the current coordinate system.

The L number is optional and represents the number of repeats. L=0 is not allowed. If the repeat feature is used, it is normally used in incremental distance mode, so that the same sequence of motions is repeated in several equally spaced places along a straight line. In absolute distance mode, L > 1 means "do the same cycle in the same place several times," Omitting the L word is equivalent to specifying L=1. The L number is not sticky.

When L>1 in incremental mode with the XY-plane selected, the X position is determined by adding the given X number either to the current X position (on the first go-around) or to the X position at the end of the previous go-around (on the repetitions). The R and Z positions do not change during the repeats.

The height of the retract move at the end of each repeat (called "clear Z" in the descriptions below) is determined by the setting of the retract mode: either to the original Z position (if that is above the R position and the retract mode is G98), or otherwise to the R position.

It is an error if:

- X, and Z words are both missing during a canned cycle,
- a P number is required and a negative P number is used,
- an L number is used that does not evaluate to a positive integer,
- rotational axis motion is used during a canned cycle,

- inverse time feed rate is active during a canned cycle,
- cutter radius compensation is active during a canned cycle.

The Z number is sticky, and it is an error if:

- the Z number is missing and the same canned cycle was not already active,
- the R number is less than the Z number.

### 10.7.22.1  Preliminary and In-Between Motion

At the very beginning of the execution of any of the canned cycles, with the XY-plane selected, if the current Z position is below the R position, the Z-axis is traversed to the R position. This happens only once, regardless of the value of L.

In addition, at the beginning of the first cycle and each repeat, the following move is made:

- a straight traverse parallel to the XY-plane to the given X-position at the same time as a straight traverse of the Z-axis only to the R position, if it is not already at the R position.

Beware of the dual axis nature of the move if the starting or finishing Z lies "within" the part as a crash of X could occur.

### 10.7.22.2  G81 Cycle

The G81 cycle is intended for drilling. Program G81 X~ Z~ R~ L~

Preliminary motion, as described above.

Move the Z-axis only at the current feed rate to the Z position.

Retract the Z-axis at traverse rate to clear Z.

**Example 1**. Suppose the current position is (1, 3) and the following line of NC code is interpreted.

```
G90 G81 G98 X4 Z1.5 R2.8
```

This calls for absolute distance mode (G90), old "Z" retract mode (G98) and calls for the G81 drilling cycle to be performed once. The X number and X position are 4. The Z number and Z position are 1.5. The R number and clear Z are 2.8. The following moves take place.

- a traverse parallel to the XY-plane to (X=4, Z =3)
- a traverse parallel to the Z-axis to (X=4, Z=2.8)
- a feed parallel to the Z-axis to (X=4, Z=1.5)
- a traverse parallel to the Z-axis to (X=4, Z=3)

**Example 2**. Suppose the current position is (X=1, Z=3) and the following line of NC code is interpreted.

```
G91 G81 G98 X4 Z-0.6 R1.8 L3
```

This calls for incremental distance mode (G91), old "Z" retract mode and calls for the G81 drilling cycle to be repeated three times. The X number is 4, the Z number is -0.6 and the R number is 1.8. The initial X position is 5 (=1+4), the clear Z position is 4.8 (=1.8+3), and the Z position is 4.2 (=4.8-0.6). Old Z is 3.0

The first move is a traverse along the Z-axis to (X=1, Z=4.8), since old Z < clear Z.

The first repeat consists of 3 moves.

- a traverse parallel to the XY-plane to (X=5, Z=4.8)
- a feed parallel to the Z-axis to (X=5, Z=4.2)
- a traverse parallel to the Z-axis to (X=5, Z=4.8)
- The second repeat consists of 3 moves. The X position is reset to 9 (=5+4).
- a traverse parallel to the XY-plane to (X=9, Z=4.8)

- a feed parallel to the Z-axis to (X=9, Z=4.2)

a traverse parallel to the Z-axis to (X=9, Z=4.8)

The third repeat consists of 3 moves. The X position is reset to 13 (=9+4).

- a traverse parallel to the XY-plane to (X=13, Z=4.8)
- a feed parallel to the Z-axis to (X=13, Z=4.2)
- a traverse parallel to the Z-axis to (X=13, Z=4.8)

### 10.7.22.3  G82 Cycle

The G82 cycle is intended for drilling. Program

        G82 X~ Z~ R~ L~ P~

Preliminary motion, as described above.

Move the Z-axis only at the current feed rate to the Z position.

Dwell for the P number of seconds.

Retract the Z-axis at traverse rate to clear Z.

### 10.7.22.4  G83  and G83.1 Cycles

The G83 cycle (often called peck drilling) and G83.1 (rapid peck drilling) is intended for deep drilling or milling with chip breaking. See also G73. The retracts in this cycle clear the hole of chips and cut off any long stringers (which are common when drilling in aluminum). This cycle takes a Q number which distance used by each peck. Program

        G83 X~ Z~ R~ L~ Q~

Preliminary motion, as described above.

Move the Z-axis only at the current feed rate downward by delta or to the Z position, whichever is less deep.

For G83 rapid back out to the clear Z. For G83.1 rapid out by the peck distance (Q)

Rapid back down to the current hole bottom, backed off a bit.

Repeat steps 1, 2, and 3 until the Z position is reached at step 1.

Retract the Z-axis at traverse rate to clear Z.

It is an error if:

- the Q number is negative or zero.

**Note:** This cycle is implemented by the macro M1083.M1S which can be inspected for implementation details. There is a separate instance of this macro for the profile that you are using.

### 10.7.22.5  G85 Cycle

The G85 cycle is intended for boring or reaming, but could be used for drilling or milling. Program G85 X~ Z~ R~ L~

Preliminary motion, as described above.

Move the Z-axis only at the current feed rate to the Z position.

Retract the Z-axis at the current feed rate to clear Z.

### 10.7.22.6  G86 Cycle

The G86 cycle is intended for boring. This cycle uses a P number for the number of seconds to dwell. Program G86 X~ Z~ R~ L~ P~

Preliminary motion, as described above.

Move the Z-axis only at the current feed rate to the Z position.

Dwell for the P number of seconds.

Stop the spindle turning.

Retract the Z-axis at traverse rate to clear Z.

Restart the spindle in the direction it was going.

The spindle must be turning before this cycle is used. It is an error if:

- the spindle is not turning before this cycle is executed.

### 10.7.22.7  G88 Cycle

The G88 cycle is intended for boring. This cycle uses a P word, where P specifies the number of seconds to dwell. Program G88 X~ Z~ R~ L~ P~

Preliminary motion, as described above.

Move the Z-axis only at the current feed rate to the Z position.

Dwell for the P number of seconds.

Stop the spindle turning.

Stop the program so the operator can retract the spindle manually.

Restart the spindle in the direction it was going.

### 10.7.22.8  G89 Cycle

The G89 cycle is intended for boring. This cycle uses a P number, where P specifies the number of seconds to dwell. program G89 X~ Z~ R~ L~ P~

Preliminary motion, as described above.

Move the Z-axis only at the current feed rate to the Z position.

Dwell for the P number of seconds.

Retract the Z-axis at the current feed rate to clear Z.

## 10.7.23          *Set Distance Mode - G90 and G91*

Interpretation of Mach3 code can be in one of two distance modes: absolute or incremental.

To go into absolute distance mode, program G90. In absolute distance mode, axis numbers (X, Z) usually represent positions in terms of the currently active coordinate system. Any exceptions to that rule are described explicitly in this section describing G-codes.

To go into incremental distance mode, program G91. In incremental distance mode, axis numbers (X, Z) usually represent increments from the current values of the numbers.

I and K numbers always represent increments or absolute distances depending on IJ Mode setting

## 10.7.24          *G92 Offsets - G92, G92.1, G92.2, G92.3*

You are strongly advised **not** to use this legacy feature on any axis where there is another offset applied. G52 is much easier to use.

To make the current point have the coordinates you want (without motion), program G92 X~ Z~ , where the axis words contain the axis numbers you want. The axis words are optional, except that at least one must be used. If an axis word is not used for a given axis, the coordinate on that axis of the current point is not changed. It is an error if:

- all axis words are omitted.

G52 and G92 use common internal mechanisms in Mach3 and may not be used together.

When G92 is executed, the origin of the currently active coordinate system moves. To do this, origin offsets are calculated so that the coordinates of the current point with respect to the moved origin are as specified on the line containing the G92. In addition, parameters 5211 to 5216 are set to the X, Z, A, B, and C-axis offsets. The offset for an axis is the amount the origin must be moved so that the coordinate of the controlled point on the axis has the specified value.

Here is an example. Suppose the current point is at X=4 in the currently specified coordinate system and the current X-axis offset is zero, then G92 X7 sets the X-axis offset to -3, sets parameter 5211 to -3, and causes the X-coordinate of the current point to be 7.

The axis offsets are always used when motion is specified in absolute distance mode using any of the fixture coordinate systems  Thus all fixture coordinate systems are affected by G92.

Being in incremental distance mode has no effect on the action of G92.

Non-zero offsets may be already be in effect when the G92 is called. They are in effect discarded before the new value is applied. Mathematically the new value of each offset is A+B, where A is what the offset would be if the old offset were zero, and B is the old offset. For example, after the previous example, the X-value of the current point is 7. If G92 X9 is then programmed, the new X-axis offset is -5, which is calculated by [[7-9] + -3]. Put another way the G92 X9 produces the same offset whatever G92 offset was already in place.

To reset axis offsets to zero, program G92.1 or G92.2 G92.1 sets parameters 5211 to 5216 to zero, whereas G92.2 leaves their current values alone.

To set the axis offset values to the values given in parameters 5211 to 5216, program G92.3

You can set axis offsets in one program and use the same offsets in another program. Program G92 in the first program. This will set parameters 5211 to 5216. Do not use G92.1 in the remainder of the first program. The parameter values will be saved when the first program exits and restored when the second one starts up. Use G92.3 near the beginning of the second program. That will restore the offsets saved in the first program.

### 10.7.25 *Set Feed Rate Mode - G94 and G95*

Two feed rate modes are recognized: units per minute and units per revolution of spindle. Program G94 to start the units per minute mode. Program G95 to start the units per rev mode.

In units per minute feed rate mode, an F word on the line is interpreted to mean the controlled point should move at a certain number of inches per minute, millimetres per minute, or degrees per minute, depending upon what length units are being used and which axis or axes are moving.

In units per rev feed rate mode, an F word on the line is interpreted to mean the controlled point should move at a certain number of inches per spindle revolution, millimetres per spindle revolution, or degrees per spindle revolution, depending upon what length units are being used and which axis or axes are moving.

### 10.7.26 *Set Canned Cycle Return Level - G98 and G99*

When the spindle retracts during canned cycles, there is a choice of how far it retracts:

1. retract perpendicular to the selected plane to the position indicated by the R word, or

2. retract perpendicular to the selected plane to the position that axis was in just before the canned cycle started (unless that position is lower than the position indicated by the R word, in which case use the R word position).

To use option (1), program G99 To use option (2), program G98 Remember that the R word has different meanings in absolute distance mode and incremental distance mode.

## 10.8  Built-in M Codes

M codes interpreted directly by Mach2 are shown in figure 10.7.

### 10.8.1 *Program Stopping and Ending – M00, M01, M02, M30*

To stop a running program temporarily (regardless of the setting of the optional stop switch), program `M00`.

To stop a running program temporarily (but only if the optional stop switch is on), program `M1`.

It is OK to program M00 and M01 in MDI mode, but the effect will probably not be noticeable, because normal behavior in MDI mode is to stop after each line of input, anyway.

If a program is stopped by an M00, M01, pressing the cycle start button will restart the program at the following line.

To end a program, program `M02` or `M30`. M02 leaves the next line to be executed as the M02 line. M30 "rewinds" the G-code file. These commands can have the following effects depending on the options chosen on the Configure>Logic dialog:

Axis offsets are set to zero (like G92.2) and origin offsets are set to the default (like G54).

Selected plane is set to XZ (like G18).

Distance mode is set to absolute (like G90).

Feed rate mode is set to Units per minute mode (like G94).

Feed and speed overrides are set to ON (like M48).

Cutter compensation is turned off (like G40).

The spindle is stopped (like M05).

The current motion mode is set to G01 (like G01).

Coolant is turned off (like M09).

No more lines of code in the file will be executed after the M02 or M30 command is executed. Pressing cycle start will resume the program (M02) or start the program back at the beginning of the file (M30).

### 10.8.2 *Spindle Control – M03, M04, M05*

To start the spindle turning clockwise at the currently programmed speed, program `M03`.

| M-code | Meaning |
|---|---|
| M00 | Program stop |
| M01 | Optional program stop |
| M02 | Program end |
| M03/04 | Rotate spindle clockwise/counterclckwise |
| M05 | Stop spindle rotation |
| M06 | Tool change (by two macros) |
| M07 | Mist coolant on |
| M08 | Flood coolant on |
| M09 | All coolant off |
| M30 | Program end and Rewind |
| M47 | Repeat program from first line |
| M48 | Enable speed and feed override |
| M49 | Disable speed and feed override |
| M98 | Call subroutine |
| M99 | Return from subroutine/repeat |

**Figure 10.7 - Built in M-codes**

To start the spindle turning counterclockwise (CCW) at the currently programmed speed, program M04. Some lathes have chucks mounted on a screwed spindle nose. Use CCW rotation with the greatest care if it is available as the chuck will tend to come unscrewed.

For a PWM or Step/Dir spindle the speed is programmed by the S word. For an on/off spindle control it will be set by the gearing/pulleys on the machine.

To stop the spindle from turning, program M05.

It is OK to use M03 or M04 if the spindle speed is set to zero. If this is done (or if the speed override switch is enabled and set to zero), the spindle will not start turning. If, later, the spindle speed is set above zero (or the override switch is turned up), the spindle will start turning. It is permitted to use M03 or M04 when the spindle is already turning or to use M05 when the spindle is already stopped but see the discussion on safety interlocks in configuration for the implications of a sequence which would reverse an already running spindle.

### 10.8.3        Tool change – M06

M06 is not needed as tool changing is actually performed when the T word is used.

### 10.8.4        Coolant Control – M07, M08, M09

To turn flood coolant on, program M07.

To turn mist coolant on, program M08.

To turn all coolant off, program M09.

It is always OK to use any of these commands, regardless of what coolant is on or off.

### 10.8.5        Re-run from first line - M47

On encountering an M47 the part program will continue running from its first line. It is an error if:

M47 is executed in a subroutine

The run can be stopped by the Pause or Stop buttons

See also the use of M99 outside a subroutine to achieve the same effect.

### 10.8.6        Override Control - M48 and M49

To enable the speed and feed override, program M48. To disable both overrides, program M49. It is OK to enable or disable the switches when they are already enabled or disabled.

### 10.8.7        Call subroutine - M98

This has two formats:

(a) To call a subroutine program within the current part program file code M98 P~ L~ or M98 ~P ~Q The program must contain an O line with the number given by the P word of the Call . This O line is a sort of "label" which indicates the start of the subroutine. The O line man not have a line number (N word) on it. It, and the following code, will normally be written with other subroutines and follow either an M2, M30 or M99 so it is not reached directly by the flow of the program.

(b) To call a subroutine which is in a separate file code M98(*filename*)L~

for example M98 (test.tap)

For both formats:

The L word (or optionally the Q word) gives the number of times that the subroutine is to be called before continuing with the line following the M98. If the L (Q) word is omitted then its value defaults to 1.

By using parameters values or incremental moves a repeated subroutine can make several roughing cuts around a complex path or cut several identical objects from one piece of material.

Subroutine calls may be nested. That is to say a subroutine may contain a M98 call to another subroutine. As no conditional branching is permitted it is not meaningful for subroutines to call themselves recursively.

### 10.8.8 Return from subroutine

To return from a subroutine program M99 Execution will continue after the M98 which called the subroutine.

If M99 is written in the main program, i.e. not in a subroutine, then the program will start execution from the first line again. See also M47 to achieve the same effect.

## 10.9 Macro M-codes

### 10.9.1 Macro overview

If any M-code is used which is not in the above list of built-in codes then Mach2 will attempt to find a file named "M*xx*.M1S" in the Macros folder. If it finds the file then it will execute the VB script program it finds within it.

The Operator>Macros menu item displays a dialog which allows you to see the currently installed macros, to Load, Edit and Save or Save As the text. The dialog also has a Help button which will display the VB functions which can be called to control Mach2. For example you can interrogate the position of axes, move axes, interrogate input signals and control output signals.

New macros can be written using an external editor program like Notepad and saved in the Macros folder or you can load an existing macro within Mach2, totally rewrite it and save it with a different file name.

## 10.10 Other Input Codes

### 10.10.1 Set Feed Rate - F

To set the feed rate, program F~

Depending on the setting of the Feed Mode toggle the rate may be in units-per-minute or units-per-rev of the spindle.

The units are those defined by the G20/G21 (G70/G71) mode.

Depending on the setting in Config>Ports & Pins-Spindle a revolution of the spindle may be defined as a pulse appearing on the Index input or be derived from the speed requested by the S word or *Set Spindle speed* DRO.

The feed rate may sometimes be overidden as described in M48 and M49 above.

### 10.10.2 Set Spindle Speed - S

To set the speed in revolutions per minute (rpm) of the spindle, program S~ The spindle will turn at that speed when it has been programmed to start turning. It is OK to program an S word whether the spindle is turning or not. If the speed override switch is enabled and not set at 100%, the speed will be different from what is programmed. It is OK to program S0; the spindle will not turn if that is done. It is an error if:

the S number is negative.

### 10.10.3 Select Tool – T

To select a tool, program T~ where the T number is entry number in the tool table for the tool to be used and the entry number for the offsets to be applied.

For example `T0202` (or equivalently `T202`) selects tool 2 with its own offsets and `T0207` selects tool 2 wth the offsets for tool 7.

`T02` is treated as equivalent to `T0202` and similarly for any other value less than or equal to 99.

Provided tool change requests are not to be ignored (as defined in Configure>Logic), Mach3 will call a macro (q.v) M6Start when the command is encountered. It will then, optionally, wait for Cycle Start to be pressed, execute the macro M6End and continue running the part program. If an automatic tool changer is in use then the Cycle Start is not needed and M6End will not be called.

You can provide Visual Basic code in the macros to operate your own mechanical tool changer and to move the axes to a convenient location to tool changing if you wish. Full details can be found in *Customizing Mach3 wiki.*

If tool change requests are set to be ignored (in Configure>Logic) then the macros are not called..

It is OK to program T0; no tool will be selected. This is useful if you want the spindle to be empty after a tool change. It is an error if:

- a negative T number is used, or a T number larger than 9999 is used.

## 10.11 Error Handling

This section describes error handling in Mach3.

Mach3 attempts to report syntax errors in part programs. If, however, a command does not work as expected or does not do anything check that you have typed it correctly. Common mistakes are GO, instead of G0 i.e. letter O instead of zero) and too many decimal points in numbers. Mach3 does not check for axis overtravel (unless software limits are in use) or excessively high feeds or speeds. Nor does it does not detect situations where a legal command does something unfortunate, such as machining the chuck.

## 10.12 Order of Execution

The order of execution of items on a line is critical to safe and effective machine operation. Items are executed in the order shown in figure 10.9 ( on next page) if they occur on the same line. If you are in any doubt code commands on separate lines there is no cost in terms of part-program performance.

| Order | Item |
|---|---|
| 1 | Comment (including message) |
| 2 | Set feed rate mode (G93, G94, G95) |
| 3 | Set feed rate (F) |
| 4 | Set spindle speed (S) |
| 5 | Select tool |
| 6 | Tool change (M6) and Execute M-code macros |
| 7 | Spindle On/Off (M3, M4, M5) |
| 8 | Coolant On/Off (M7, M8, M9) |
| 9 | Enable/disable overrides (M48, M49) |
| 10 | Dwell (G4) |
| 11 | Set active plane (G17, G18, G18) |
| 12 | Set length units (G20, G21) |
| 13 | Cutter radius compensation On/Off (G40, G41, G42) |
| 14 | Tool table offset On/Off (G43, G49) |
| 15 | Fixture table select (G54 - G58 & G59 P~) |
| 16 | Set path control mode (G61, G61.1, G64) |
| 17 | Set distance mode (G90, G91) |
| 18 | Set canned cycle return level mode (G98, G99) |
| 19 | Home, or change coordinate system data (G10), or set offsets (G92, G94) |
| 20 | Perform motion (G0 to G3, G12, G13, G80 to G89 as modified by G53 |
| 21 | Stop or repeat (M0, M1, M2, M30, M47, M99) |
| | |

**Table 10.9 - Order of execution on a line**

# 11. Appendix 1 - Mach3Turn screenshot pullout
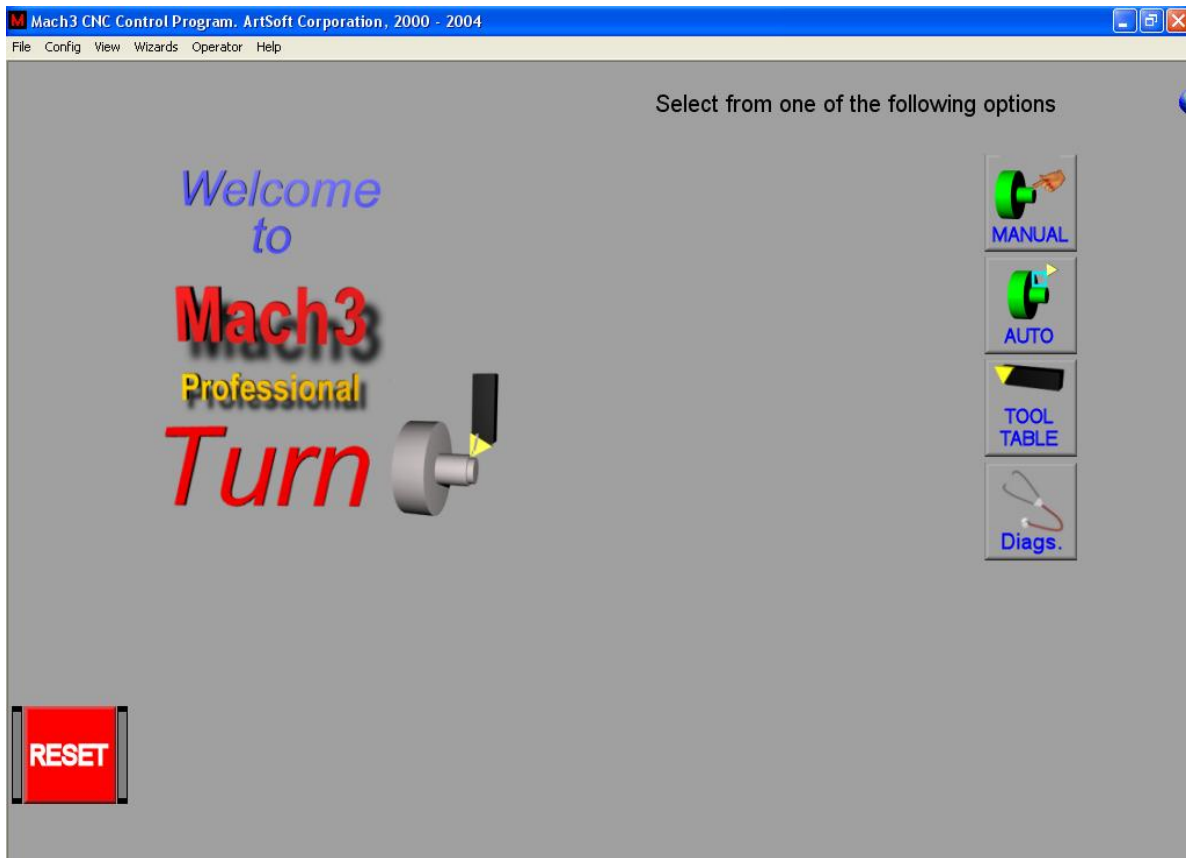


**Welcome screen**
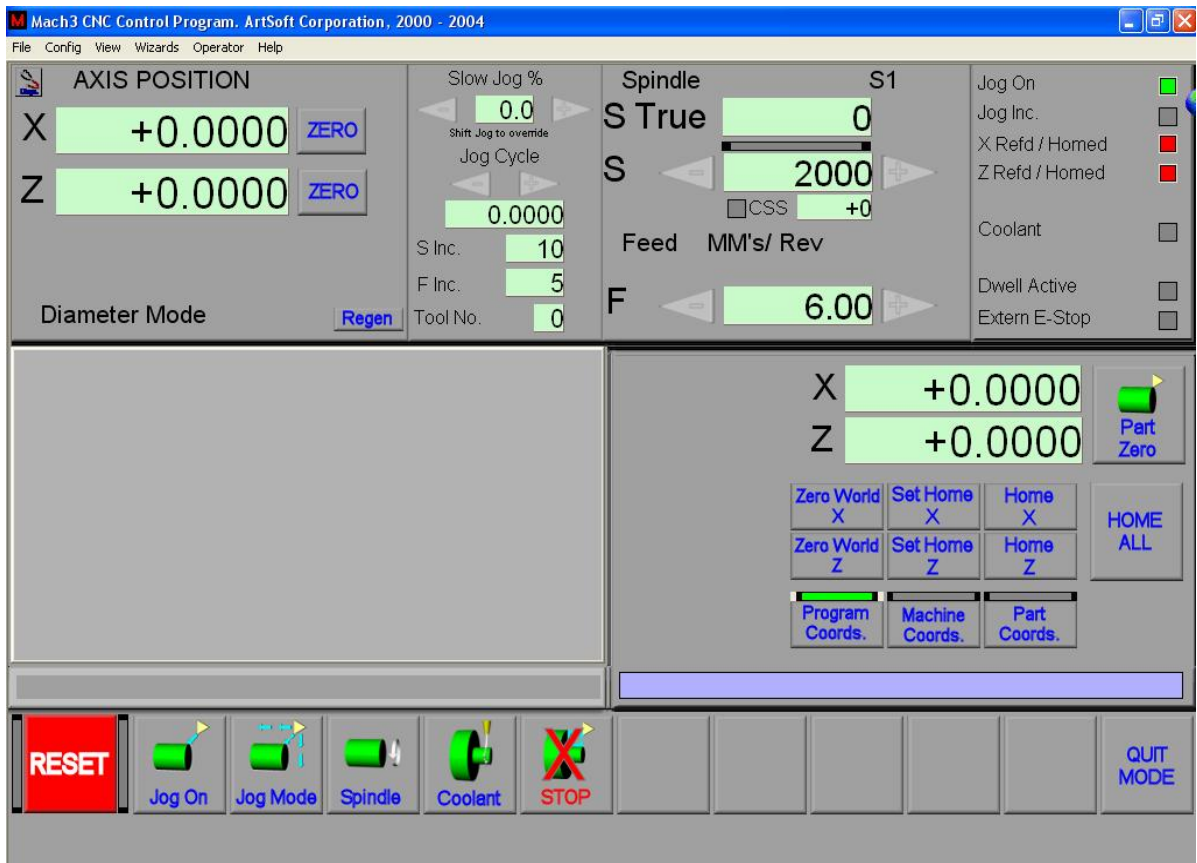


**Manual screen**

# Sample schematic diagrams
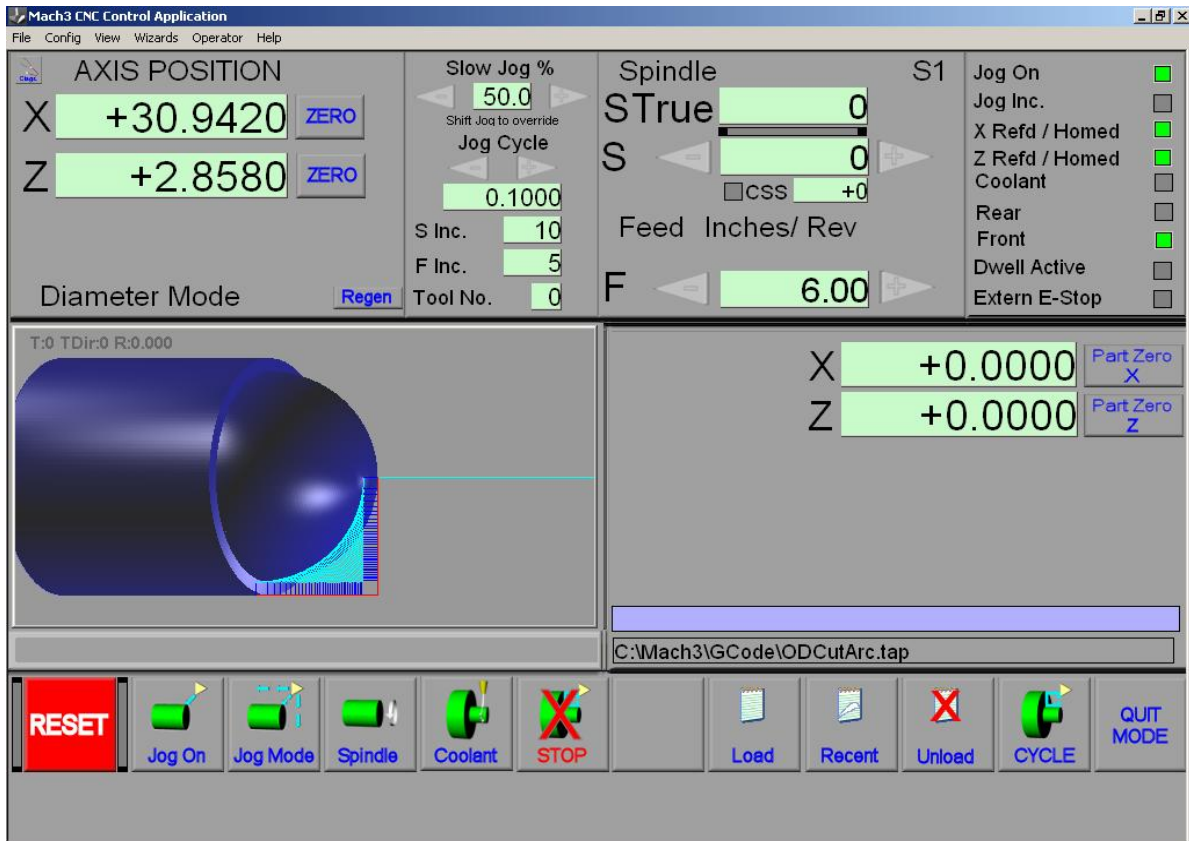


**Auto Preparation screen**
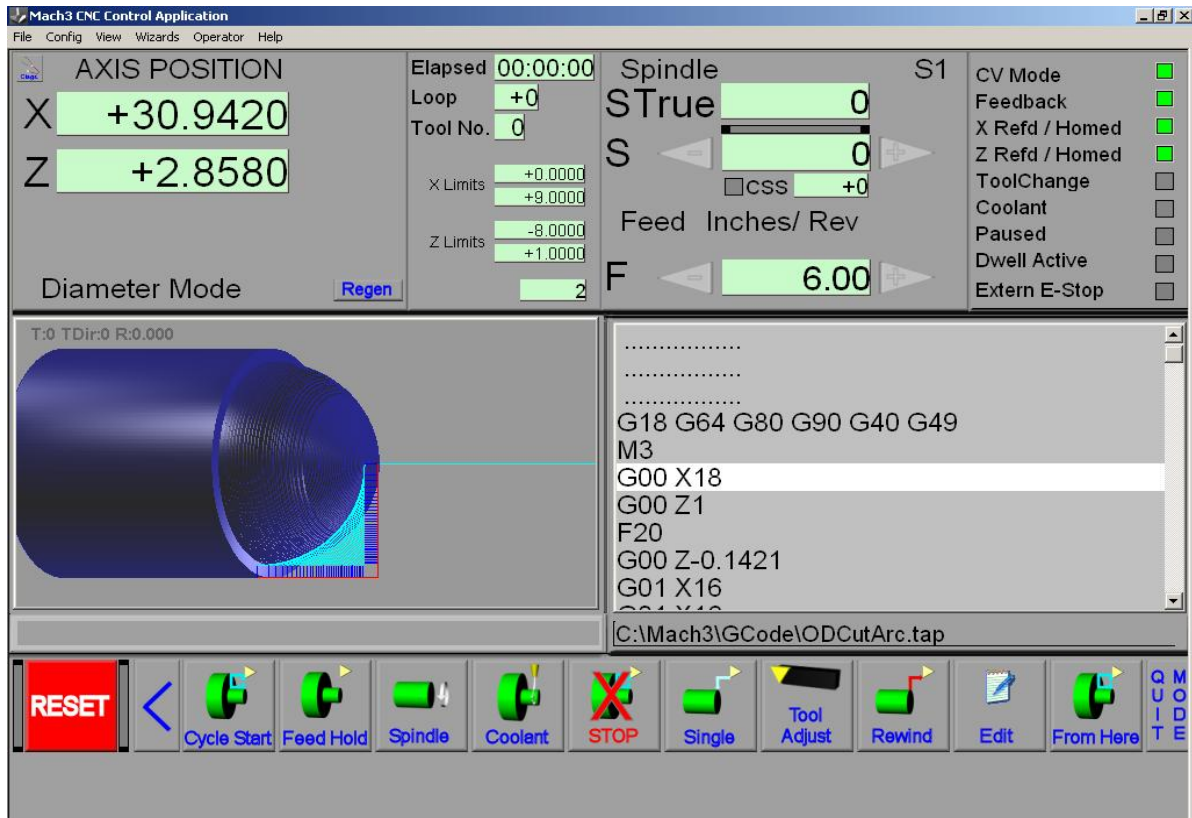


**Auto Cycle screen**

# Sample schematic diagrams
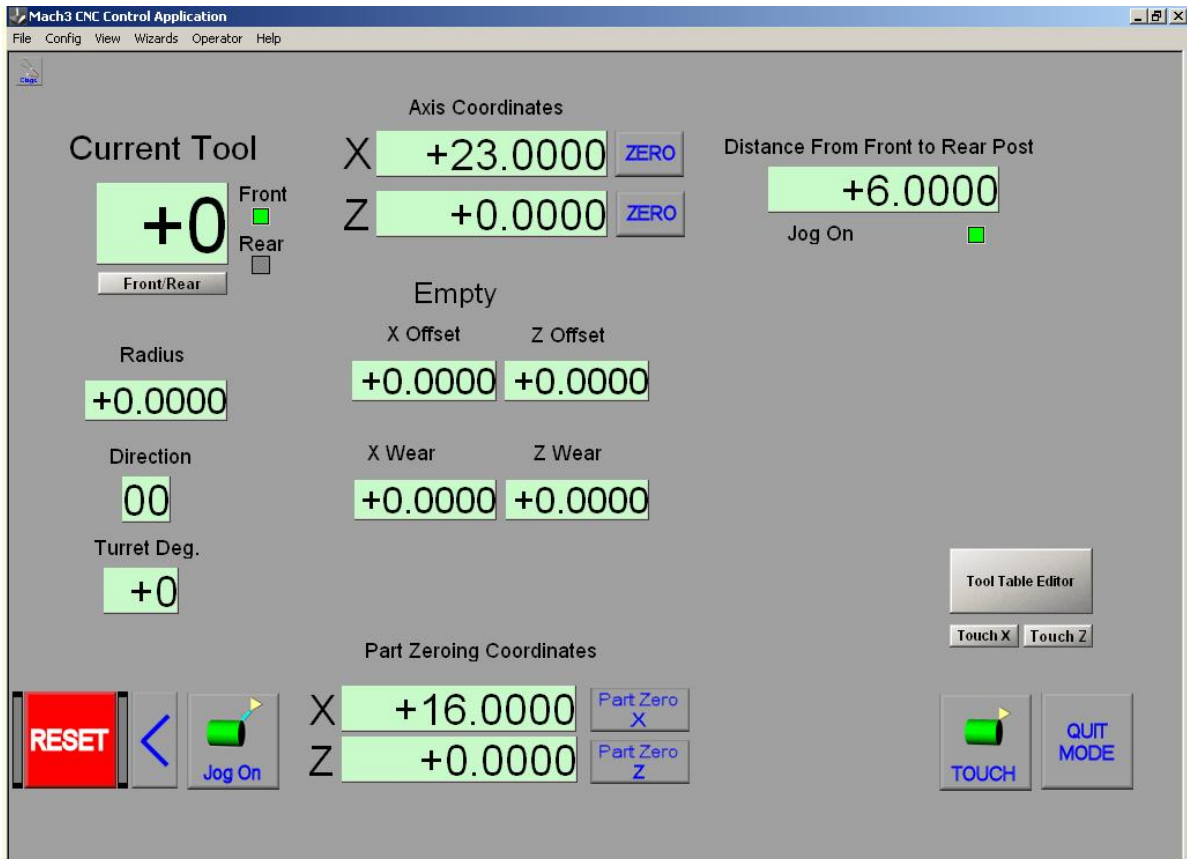


**Tool table screen**
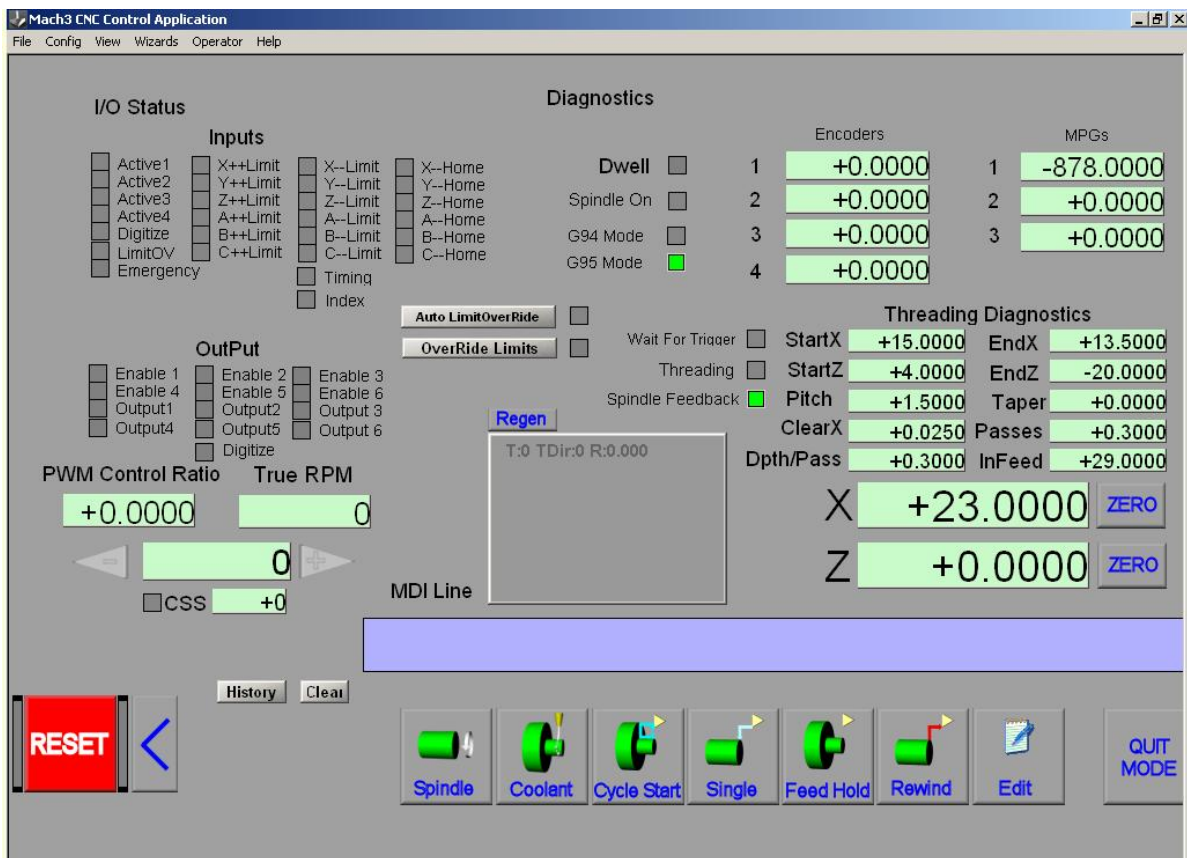


**Diagnostics screen**

# Appendix 2 - Sample schematic diagrams

## 12.1   EStop and limits using relays
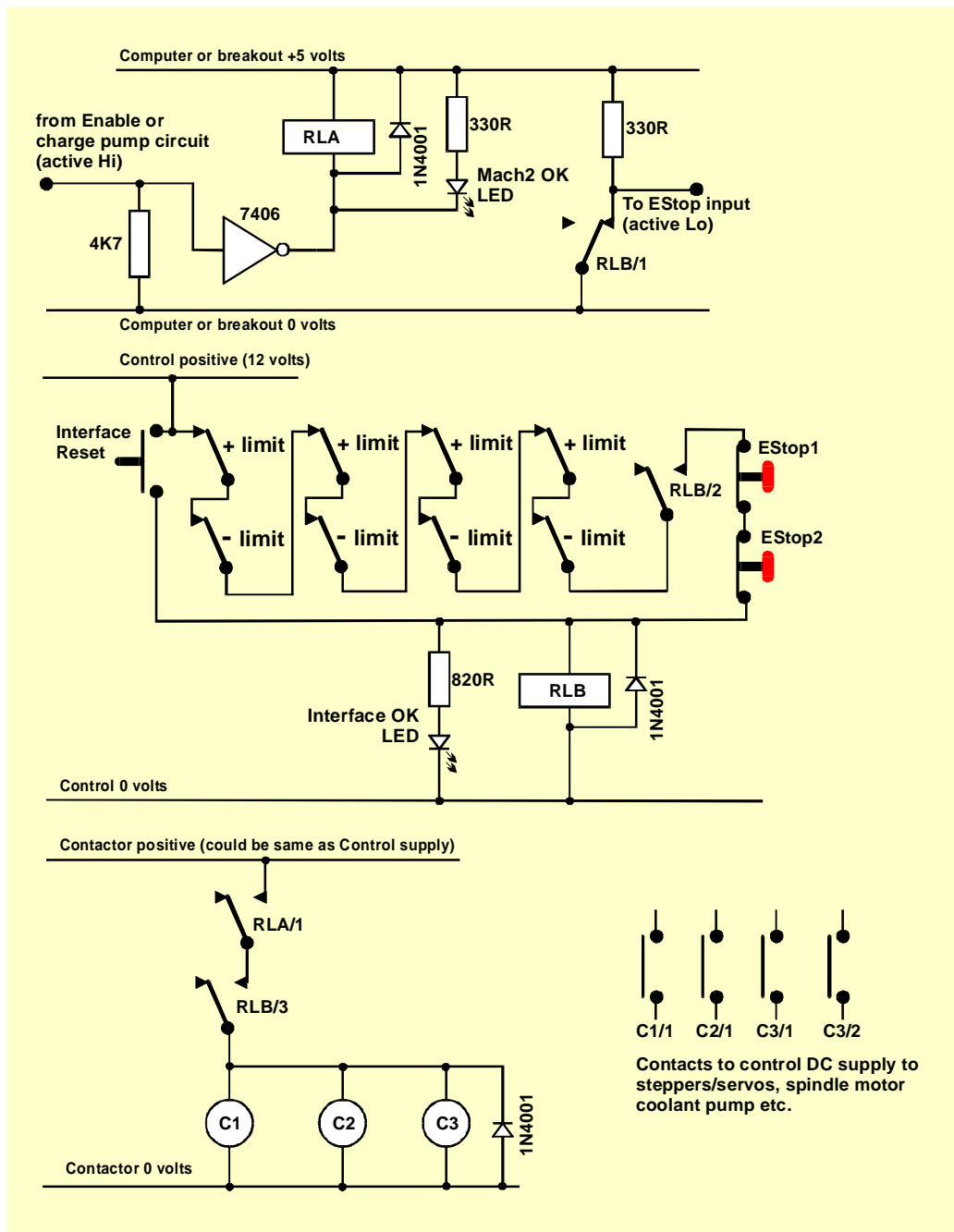


**Figure 12.1 - Sample EStop and Limit switch connections**

**Notes:**

1.  This circuit is only illustrative of one possible solution to externally connected limit switches. If you require reference switches then these should be separate and connected to Mach3 inputs.

2.  Relay contacts are shown in the de-energised position. Limit switches and push buttons are not operated.

3.  Holding Interface Reset pressed will allow the Mach3 Reset button to be pressed and the axes to be jogged off the limit switches. The Interface Reset will then latch.

4. Relay A needs one NO contact. It must have a 5 volt coil that is at least 150 ohms (i.e. not require more than 33 milliamps to operate). Omron G6H-2100-5 is suitable with contacts rated at 1 amp 30 volts DC

5. Relay B needs 1 NC and 2 NO contacts. It can have any convenient coil voltage to suit an available supply. The common of this should, ideally, not be the PC 0 volt rail to avoid the long wiring of the limit and EStop switches inducing noise. The Omron MY4 series is suitable having four contacts rated at 5 amps 220 volts AC.

6. The LEDs are optional but useful as an indication of what is happening. The current limiting resistor for the Interface OK LED needs to be 1.8 kilohms if a 24 volt supply is used.

7. If the coil voltages are suitable then the contactors can use the "Control" positive and common supply.

8. The arrangement of contactors (Coils shown as C1, C2, C3) depends on your drive power supply arrangements and the wiring of the motors in the machine tool. You should aim to switch the DC supply to the steppers and/or servos after the smoothing capacitor to ensure a prompt stop. You may wish to rewire the spindle and coolant motors so that the control contactor does not trip the no-volt release circuitry (i.e. you may wish to switch the motor leads **after** the main machine contactors. Do not share contacts on a given contactor between AC mains and the stepper/servo DC supply on account of the greatly increased risk of a short circuit between these supplies. **Seek advice if you are unsure, especially before working with 230/415 volt 3-phase circuits.**

9. The catching diodes across the relay and contactor coils are needed to absorb the back emf when switching the current off in the coils. Contactors may come with suitable coil suppression circuits built in.

# 13.  Appendix 3 - Record of configuration used

You should keep a paper record of your Mach3 setup!

A complete Mach3 configuration includes a lot of detailed information. You will not wish to repeat the process step by step when you update your computer.

Mach3 profiles are .XML files and you will probably keep them in the Mach3 folder. Use Windows Explorer to find the profile you wish to copy and drag it to another folder **while holding down the Control key**. You can of course use any other file copying technique if you prefer.

If you double-click the file name then your web browser (probably Internet Explorer) will open the .xml file and display it
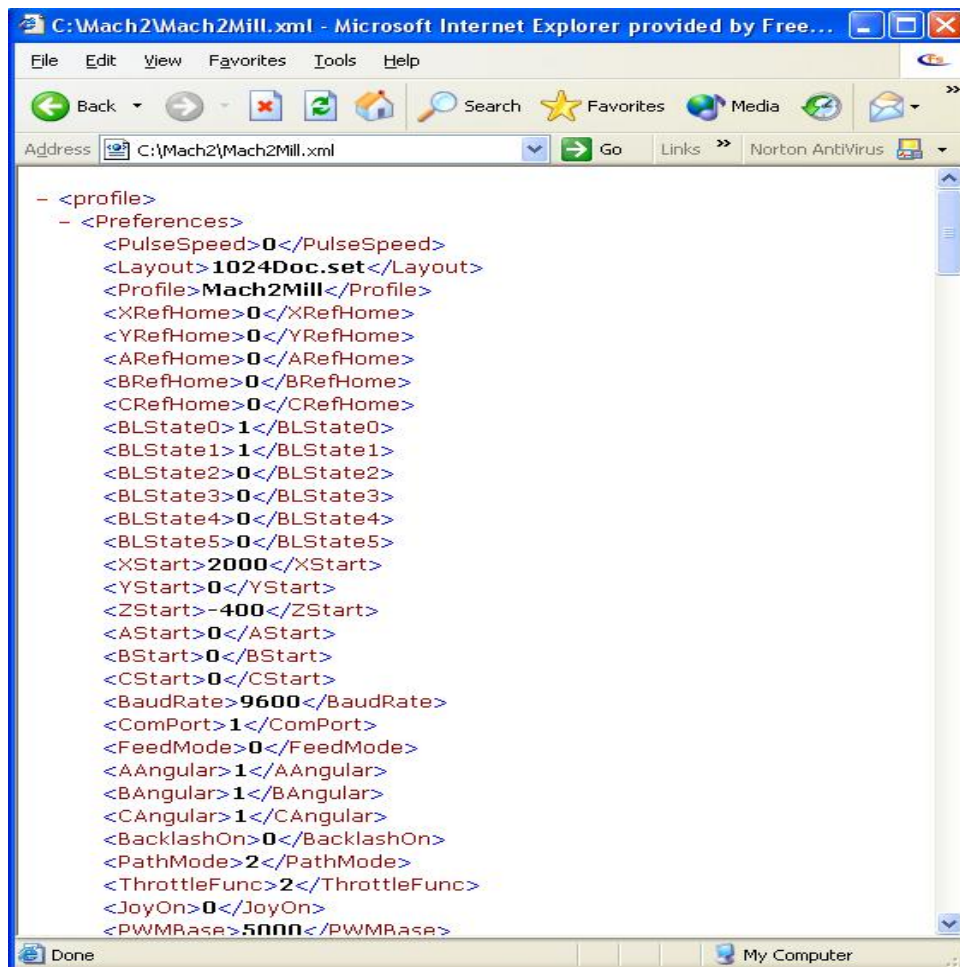


**Figure 13.1 – Internet Explorer displaying Profile**

The XML file can be edited by a text editor such as Notepad but this is **strongly** not recommended,

The profile file can be useful information to attach to an e-mail when asking for support from ArtSoft Corp

# 14. Revision history

| | | |
|---|---|---|
| Rev 1.84A-2 | 2 February 2006 | Re-issue to correct missing images in PDF |
| Rev 1.84A-1 | 31 January 2006 | Documentation for first release. Includes changes to describe new tool changing strategy, tool table setup by touching and tool nose compensation details |
| Rev Beta 7.57A-1 | 26 April 2005 | Initial preliminary release |
| Rev 7-xxA-1 | 10 February 2005 | Drafting version |

# 15. Index

> **Hint:** Where there is a choice, most index entries are made using the name of a thing (e.g. Axis drive) rather than an action (e.g. Tuning) so you will get better results thinking about the part on which you want information. Thus looking for "Axis drives - tuning" will give better results than looking for "Tuning - axis drives". For important information both entries will probably appear.

> If you have difficulty because you tried to look something up and the index entry was missing, please take a moment to e-mail support@artofcnc.ca with a note of (a) the words you were looking up and (b) where in the manual you found the information you wanted - assuming you did!

## T

## U

## W

## Z