Mach3 CNC Controller
Software Installation and Configuration

Version 3



Copyright © 2003, 2004, 2005, 2006, 2008 ArtSoft USA. All Rights Reserved.

The following are registered trademarks of Microsoft Corporation: Microsoft, Windows. Any other trademarks used in this manual are the property of the respective trademark holder.

Please send queries, comments, complaints, corrections, and suggested improvements for this manual to: support@machsupport.com.

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

Printed November 6, 2008

Contents

Chapter 1 Introduction to CNC Systems

Before You Begin.... 1-1

1.1 Introduction 1-2
1.2 Components of a CNC Machining System 1-2
1.3 How Mach3 Fits In 1-4
1.4 What Mach3 Can Do 1-4

Chapter 2 Installing the Mach3 Software

2.1 Installation 2-1
2.1.1 Downloading 2-1
2.1.2 Installing 2-1
2.1.2.1 If a machine tool is connected, disconnect it now. 2-1
2.1.2.2 Run the Mach3 software installation package. 2-2
2.1.2.3 If You're Using Windows Vista 2-4
2.1.3 The Vital Reboot 2-4
2.2 Testing The Installation 2-4
2.2.1 If You Are Using the Default Parallel Port Driver 2-5
2.3 Mach3 Profiles 2-7
2.3.1 Creating a Profile 2-7
2.4 Installation Problems 2-9
2.4.1 Running DriverTest After a Mach3 Crash 2-9
2.4.2 Manual Driver Installation and Uninstallation 2-9

Chapter 3 Introducing the Mach3 Screens and Commands

3.1 Screens 3-1
3.1.1 Types of Objects on Screens 3-2
3.1.2 Using Buttons and Shortcuts 3-3
3.1.3 Data Entry to DROs 3-3
3.2 Jogging 3-3
3.3 Manual Data Input (MDI) and Teaching 3-5
3.3.1 Teaching 3-5

Chapter 4 Hardware Requirements and Connecting the Machine Tool

4.1 The PC Parallel Port and its History 4-1
4.2 Logic Signals 4-2
4.2.1 Isolating Breakout Boards 4-3
4.3 The EStop Control 4-4
4.4 Axis Drive Options 4-5
4.4.1 Steppers and Servos 4-5
4.4.2 Determining Axis Drive Requirements 4-6
4.4.2.1 Example 1 - Mill Table Cross Slide 4-6
4.4.2.2 Example 2 - Router Gantry Drive 4-8
4.4.3 How the Step and Dir Signals Work 4-8
4.5 Limit and Home Switches 4-9
4.5.1 The Switches 4-10
4.5.2 Where to Mount the Switches 4-13
4.5.3 How Mach3 Uses Shared Switches 4-14
4.5.4 Home Referencing in Action 4-14
4.5.5 Other Home and Limit Options and Hints 4-15
4.5.5.1 Home Switch not Near Limit Switch 4-15
4.5.5.2 Separate High Accuracy Home Switch 4-15
4.5.5.3 Limit Switches of Multiple Axes Connected Together 4-15
4.5.5.4 Home Switches of Multiple Axes Connected Together 4-15
4.5.5.5 Slaving 4-15
4.5.6 Summary Wiring Diagram 4-17
4.6 Spindle Control 4-18
4.6.1 On/Off Motor Control 4-18
4.6.2 Step and Direction Motor Control 4-18
4.6.3 PWM Motor Control 4-18
4.7 Coolant 4-20
4.8 Knife Direction Control 4-20
4.9 Digitize Probe 4-20
4.10 Linear (Glass Scale) Encoders 4-20
4.11 Spindle Index Pulse 4-22
4.12 Charge Pump — a Pulse Monitor 4-22
4.13 Other Functions 4-23
4.14 Sample Schematic of EStop and Limits Using Relays 4-24

Chapter 5 Configuring Mach3 for Your Machine and Drives

5.1 A Configuration Strategy 5-1
5.2 Initial Configuration 5-1
5.2.1 Defining Addresses of Port(s) to Use 5-2
5.2.2 Choosing Kernel Speed 5-3
5.2.3 Defining Special Features 5-3
5.3 Defining Input and Output Signals To Use 5-4

5.3.1 Axis and Spindle Output Signals to Be Used 5-4
5.3.2 Input Signals To Be Used 5-5
5.3.3 Emulated Input Signals 5-7
5.3.4 Output Signals 5-8
5.3.5 Defining Encoder and Manual Pulse Generator (MPG) Inputs 5-9
5.3.5.1 Setting Up Encoders 5-10
5.3.5.2 Setting Up MPGs 5-10
5.3.6 Configuring the Spindle 5-10
5.3.6.1 Coolant Control 5-11
5.3.6.2 Spindle Relay Control 5-11
5.3.6.3 Motor Control 5-11
5.3.6.4 Modbus Spindle Control 5-12
5.3.6.5 General Parameters 5-12
5.3.6.6 Pulley Ratios 5-12
5.3.6.7 Special Function 5-13
5.3.7 Mill Options Tab 5-13
5.3.8 Initial Testing 5-14
5.4 Defining the Setup Units 5-14
5.5 Tuning Motors 5-15
5.5.1 Calculating the Steps Per Unit 5-15
5.5.1.1 Calculating Mechanical Drive 5-16
5.5.1.2 Calculating Motor Steps Per Revolution 5-18
5.5.1.3 Calculating Mach3 Steps Per Motor Revolution 5-18
5.5.1.4 Mach3 Steps Per Unit 5-18
5.5.2 Setting the Maximum Motor Speed 5-19
5.5.2.1 Practical Trials of Motor Speed 5-19
5.5.2.2 Motor Maximum Speed Calculations 5-20
5.5.2.3 Automatic Setting of Steps per Unit 5-21
5.5.3 Choosing an Acceleration Value 5-22
5.5.3.1 Inertia and Forces 5-22
5.5.3.2 Testing Different Acceleration Values 5-23
5.5.3.3 Why You Want To Avoid a Big Servo Error 5-23
5.5.3.4 Choosing an Acceleration Value 5-23
5.5.4 Saving and Testing Axis 5-23
5.5.4.1 Repeat Configuration of Other Axes 5-25
5.5.5 Spindle Motor Speed Control Setup 5-26
5.5.5.1 Motor Speed, Spindle Speed, and Pulleys 5-26
5.5.5.2 Pulse Width Modulated Spindle Controller 5-28
5.5.5.3 Step and Direction Spindle Controller 5-29
5.5.5.4 Testing the Spindle Drive 5-29
5.6 Other Configuration 5-29
5.6.1 Configure Homing and Soft Limits 5-29
5.6.1.1 Referencing Speeds and Direction 5-30
5.6.1.2 Position of Home Switches 5-30
5.6.1.3 Configure Soft Limits 5-30
5.6.1.4 G28 Home Location 5-31
5.6.2 Configure System Hotkeys 5-31
5.6.3 Configure Backlash 5-32
5.6.4 Configure Slaving 5-33

5.6.5 Configure Toolpath 5-33
5.6.6 General Configuration 5-35
5.6.6.1 General Logic Configuration, Column 1 5-35
5.6.6.2 General Logic Configuration, Column 2 5-36
5.6.6.3 General Logic Configuration, Column 3 5-36
5.6.6.4 General Logic Configuration, Column 4 5-38
5.7 How The Profile Information Is Stored 5-39
5.8 Documenting Your Setup 5-40

Appendix A Torch Height Controls in Mach3

Figures

1-1 Main Parts of a CNC System 1-3
2-1 Select Program Components Screen 2-2
2-2 Create a Custom Profile Screen 2-3
2-3 Create Mill Profile 2-3
2-4 The Running DriverTest Program 2-6
2-5 Profile Selection Window 2-7
2-6 Create Profile Window 2-8
2-7 Shortcut to "MyMill" 2-8
3-1 Select Control Device Screen 3-1
3-2 Screen Selection Tabs 3-2
3-3 Control Buttons Flyout 3-4
3-4 MDI Example 3-5
3-5 Teaching a Rectangle 3-6
3-6 Taught Program Running 3-7
4-1 Parallel Port Female Connector (seen from back of PC) 4-2
4-2 Three Examples of Commercially Available Breakout Boards 4-4
4-3 Example of Servo Motor with Encoder 4-6
4-4 Step Pulse Form (Active Lo) 4-8
4-5 Inverted Pulse Form (Active Hi) 4-9
4-6 Example of a Mechanical Limit Switch 4-9
4-7 Two Normally Closed Switches Give a Logical OR 4-11
4-8 Optical Switch on Table with Vane on Bed of Machine 4-11
4-9 Two Switches Operated By Frame with Overtravel Prevented by Mechanical Stops 4-12
4-10 Mill with Tool at X=0, Y=0 Position (Note the Dog Is on Limit Switch) 4-13
4-11 Ramps Operating One Switch 4-14
4-12 Example Wiring Diagram 4-17
4-13 A 50% Pulse Width Modulated Signal 4-19
4-14 A 20% Pulse Width Modulated Signal 4-19
4-15 Glass Scale Encoder 4-21
4-16 Quadrature Signals 4-21
4-17 Encoder DROs 4-22
5-1 Selecting the Ports and Pins Dialog from the Config Menu 5-2
5-2 Port Setup and Axis Selection Tab on the Ports and Pins Dialog 5-2
5-3 Motor Outputs Tab on the Ports and Pins Dialog 5-4
5-4 Input Signals Tab on the Ports and Pins Dialog 5-5

5-5 Output Signals Tab on the Ports and Pins Dialog 5-8
5-6 Encoder/MPG Tab on the Ports and Pins Dialog 5-10
5-7 Spindle Setup Tab of the Ports and Pins Dialog 5-11
5-8 Mill Options Tab of the Ports and Pins Dialog 5-13
5-9 Select Native Units Dialog 5-15
5-10 Machine Drivetrain Components 5-16
5-11 Motor Tuning Dialog 5-19
5-12 Settings Alt6 Tab 5-21
5-13 Automatic Setting of Steps per Unit 5-21
5-14 Window to Enter Nominal Move Distance 5-22
5-15 Window to Enter Actual Move Distance 5-22
5-16 Selecting MDI 5-23
5-17 Manually Entering G20 G90 Commands 5-24
5-18 Establishing a Zero Position 5-24
5-19 Gage Block in Position 5-25
5-20 Step Pulleys 5-26
5-21 Config>Spindle Pulleys... Dialog 5-27
5-22 Config>Homing/Limits Dialog 5-30
5-23 Configure System Hotkeys Dialog 5-31
5-24 Configure Backlash Dialog 5-32
5-25 Configure Slave Axis Dialog 5-33
5-26 Configure Toolpath Dialog 5-34
5-27 General Configuration Dialog 5-35

Tables

5-1 List of Possible Input Signals 5-6
5-2 List of Possible Output Signals 5-9
5-3 Pulley Configuration for Bridgeport Step Pulley J Head 5-28
5-4 Defaults for Jog Hotkeys 5-32
5-5 Document Your Motor Outputs Settings Here 5-40
5-6 Document Your Input Signal Settings Here 5-41
A-1 Torch Height Controls in Mach3 ..... A-1

This is a blank left-hand page for two-sided printing.

Chapter 1 Introduction to CNC Systems

This chapter introduces you to terminology used in the rest of this manual and explains the purpose of the different components in a computer numerically controlled (CNC) system.



Before You Begin....

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, to plunge a panel-fielding router cutter deep into a piece of oak, or to mill away the clamps holding your work to the table.

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, be sure to seek guidance from a professionally qualified expert rather than risk injury to yourself or to others.

1.1 Introduction

This document tells you how to install and configure the Mach3Mill software to control a milling machine or similar machine tool. It also describes the hardware components required and how to interface them to your computer. Typical machine tools that can be controlled are mills, routers, and plasma cutting tables. A separate document, *Using Mach3Mill*, explains how to use Mach3Mill after you have installed and configured it.

YOU WILL NEED TO READ THIS DOCUMENTATION! Mach3 is a complex piece of software. You will not be successful if you simply start trying to "get it to work." While that approach may be appropriate for some software, it is not appropriate for Mach3. Save yourself a lot of aggravation by taking the time to work through the chapters in this manual, taking the installation and configuration step by step.

An online wiki format document, *Mach Customization Wiki* (link at www.machsupport.com/MachCustomizeWiki/index.php?title=Main_Page), explains in detail how to alter screen layouts, to design your own screens and Wizards, and to interface to special hardware devices.

ArtSoft USA strongly advises you to join one or both of the online discussion fora for Mach3. Links to join are at www.machsupport.com. While these forums have many engineers and machinists with a vast range of experience as participants, they do not constitute a substitute for a machine tool manufacturer's support network. If your application requires that level of support, then you should buy your CNC 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.

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.

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.

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

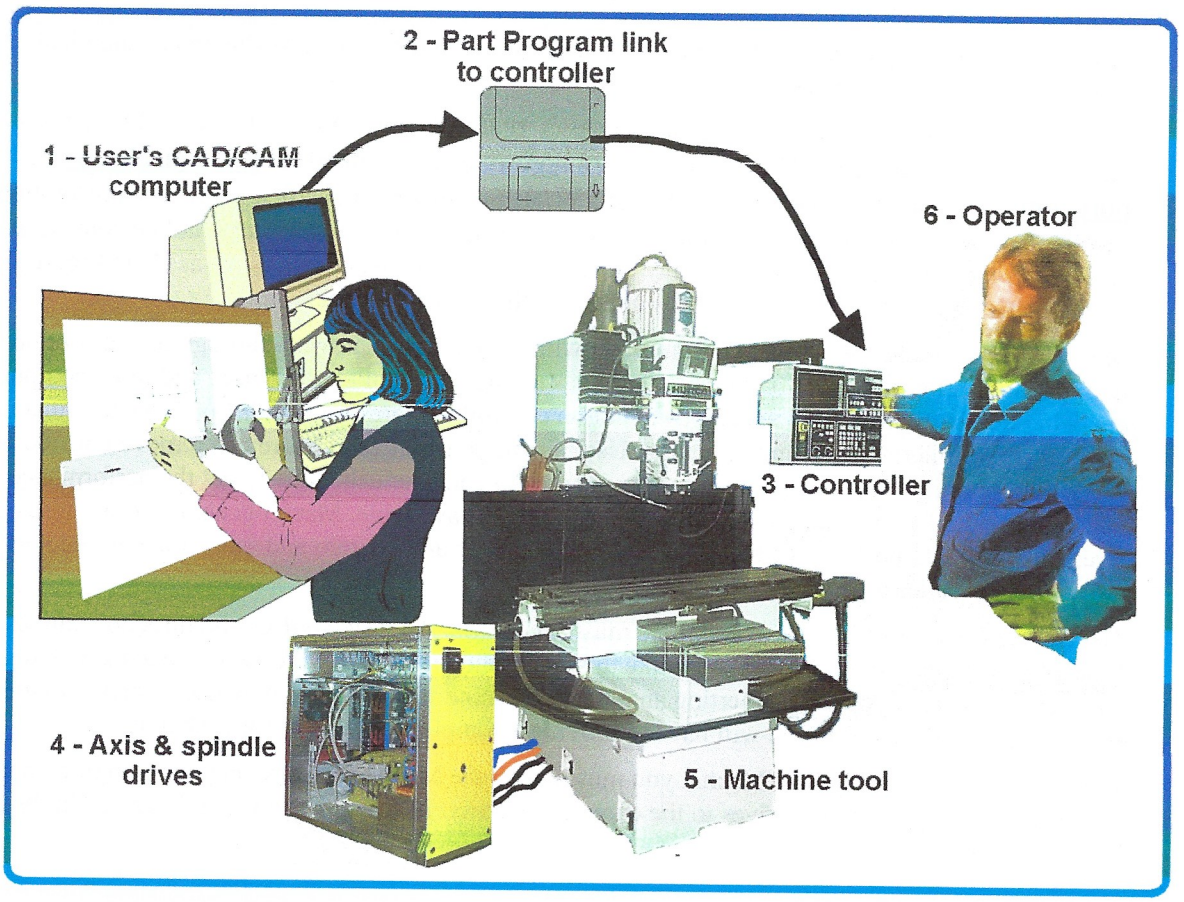
1.2 Components of a CNC Machining System

The main components of a CNC system are shown in Figure 1-1. These are:

1. A Computer Aided Design/Computer Aided Manufacturing (CAD/CAM) program. The part designer uses the CAD/CAM program to generate an output file called a *part program*. The part program, often written in "GCode," describes the machine steps required to make the desired part. You can also create a GCode program manually.
2. A file transfer medium such as a USB flash drive, floppy disk, or network link, transfers the output of the CAD/CAM program to a Machine Controller.
3. A Machine Controller. The Machine Controller reads and interprets the part program to control the tool which will cut the workpiece. Mach3, running on a PC, performs the Machine Controller function and send signals to the Drives.

4. The Drives. The signals from the Machine Controller are amplified by the Drives so they are powerful enough and suitably timed to operate the motors driving the machine tool axes.
5. The machine tool. The axes of the machine are moved by screws, racks or belts which are powered by servo motors or stepper motors.

Figure 1-1: Main Parts of a CNC System



Although a milling machine is illustrated, the machine could be a router or a plasma or laser cutter.

If the interfaces exist, in addition to controlling tool position the Machine Controller can start and stop the spindle motor, control its speed, turn coolant on and off, and check that a part program's commands or the Operator are not trying to move any machine axis beyond its limits.

The Machine Controller can also have controls such as pushbuttons, 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 GCode program can request complicated coordinated movements of the machine axes, the Machine Controller has to be able to perform a lot of calculations in "real-time" (e.g. cutting a helix requires a lot of trigonometric calculation). Historically this made it an expensive piece of equipment.

1.3 How Mach3 Fits In

Mach3 is a software package that runs on a PC and turns it into a very powerful and economical Machine Controller to replace (3) in Figure 1-1.

To run Mach3, you need a PC running the Windows 2000, Windows XP, or Windows 32-bit Vista operating system. (Windows Vista may require a registry patch, available at www.machsupport.com.) ArtSoft USA recommends at least 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 use this computer for any other functions in the workshop (such as (1) in Figure 1-1 - running a CAD/CAM package) when it is not controlling your machine.

Mach3 and its parallel port driver communicate with the machine hardware through one (or optionally two) parallel (printer) ports. If your computer does not have a parallel port (more and more computers are being built without one), you can buy a motion controller board from a third-party vendor that uses a USB port or Ethernet for communication. Use of a motion controller board can remove considerable processing load from the computer, so you may want to consider using one to get the performance advantage even if your computer does have a parallel port available. Mach3 generates step pulses and direction signals to perform the steps defined by a GCode part program and sends them to the port(s) or motion controller board.

The drivers for your machine's axis motors must accept March3's step pulses and direction signals. 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.

To set up a CNC system to use Mach3, you must install the Mach3 software on your computer, and properly connect your motor drives to the computer's ports. These operations are described in the following chapters.

1.4 What Mach3 Can Do

Mach3 is a very flexible program designed to control machines such as milling machines, lathes, plasma cutters, and routers. Features of these machines that are used by Mach3 include:

- Some user controls. An emergency stop (EStop) button **must** be provided on every machine.
- Two or three axes of motion, which are usually at right angles to each other (referred to as X, Y, and Z).
- A tool which moves relative to a workpiece. The origin of the reference axes is fixed in relation to the workpiece. The relative movement can be by (1) the tool moving (e.g. the quill of a milling spindle moves the tool in the Z direction, or a lathe tool mounted on a cross-slide and a saddle moves the tool in the X and Z directions) or by (2) the table and workpiece moving (e.g. on a knee type mill the table moves in the X, Y, and Z directions while the tool remains fixed in the spindle).

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.
- A controlled "spindle." The spindle can rotate the tool (mill) or the workpiece (turning).
- Up to three additional axes. These can be either Rotary (i.e. their movement is measured in degrees) or Linear. One of the additional linear axes can be slaved to the X or Y or Z axis. The two

will move together at all times in response to a part program's moves and to your jogging, but they will each be referenced separately. (see Section 5.6.4 for more details).

- A switch or switches which interlock the guards on the machine.
- Controls for the way coolant is delivered (Flood and/or Mist).
- A probe in the tool holder that allows digitizing of an existing part.
- Encoders, such as linear glass scales, which can display the position of parts of the machine.
- Special functions.

Most connections between your machine and the PC running Mach3 are made through the parallel (printer) port(s) of the computer. A simple machine will need only one port; a complex one will need two.

Connections for control of special functions like an LCD display, a tool-changer, axis clamps, or a swarf conveyor can also be made through a ModBus device (e.g. a PLC or Homann Designs ModIO controller).

Buttons can be interfaced by a "keyboard emulator" which generates pseudo-key presses in response to input signals.

Mach3 will control up to six axes simultaneously, coordinating their movement with linear interpolation or perform circular interpolation on two axes (out of X, Y or Z) while simultaneously linearly interpolating the other four with the angle being swept by the circular interpolation. The tool can thus move in a tapering helical path if required! The feed rate during these moves is maintained at the value requested by your part program, subject to limitations of the acceleration and maximum speed of the axes. You can move the axes by hand with various jogging controls.

If the mechanism of your machine is like a robot arm or a hexapod, then Mach3 **will not** be able to control it because of the kinematic calculations that would be needed to relate the "tool" position in X, Y and Z coordinates to the length and rotation of the machine arms.

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 switch and can take note of the operation of the reference switches, the guard interlock, and the limit switches

Mach3 can store the properties of up to 256 different tools. If, however, your machine has an automatic tool changer or magazine, you will have to control it yourself. Mach3 provides program macro capability, but you must do the programming.

This is a blank left-hand page for two-sided printing.

Chapter 2 Installing the Mach3 Software

If you haven't done so already, download the Mach3 software from www.machsupport.com and try it out on your computer. You do not need a machine tool to be connected. Indeed, for the present it is better not to have one.

If you have bought a complete CNC system from a reseller, then some or all of the installation steps described in this chapter may have already been done for you.

2.1 Installation

Mach3 is distributed by ArtSoft USA over the Internet. You download the package as one self installing file (which, in the present release, is about 25 megabytes). When installed, it will run for an unlimited period as a demonstration version. As demonstration software, it places a few limitations on the speed, the size of job that can be undertaken, and the specialist features supported. When you purchase a licence, you can "unlock" the demonstration version you have already installed and configured to give full software functionality. Details of pricing and options are available at www.machsupport.com.

2.1.1 Downloading

Download the installation package from www.machsupport.com. Use the right mouse button and *Save Target as*... to put the self-installing file on the Desktop or in a convenient folder. You should be logged in to Windows as an Administrator.

When the file has downloaded, it can be run immediately by using the *Open* button on the download dialog window, or the download dialog window can be closed and the installation done later. When you want to do the installation, run the file you downloaded and saved. For example, if you saved the installation file on the Desktop, just double-click it. If you saved the file in a folder, run Windows Explorer (right click *Start* button), and double-click on the name of the downloaded file in the folder.

2.1.2 Installing

This section will guide you through the installation of the Mach3 software. If you already have a version of Mach3 installed on the computer, you can install the new version on top of it. You do not need to remove the old version first.

2.1.2.1 If a machine tool is connected, disconnect it now.

You do not need a machine tool connected to install the software. In fact, if you are just starting it is probably better not to have one connected. If a machine tool is connected to your computer, note where the cable or cables from the machine tool are plugged into your PC. Turn 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.

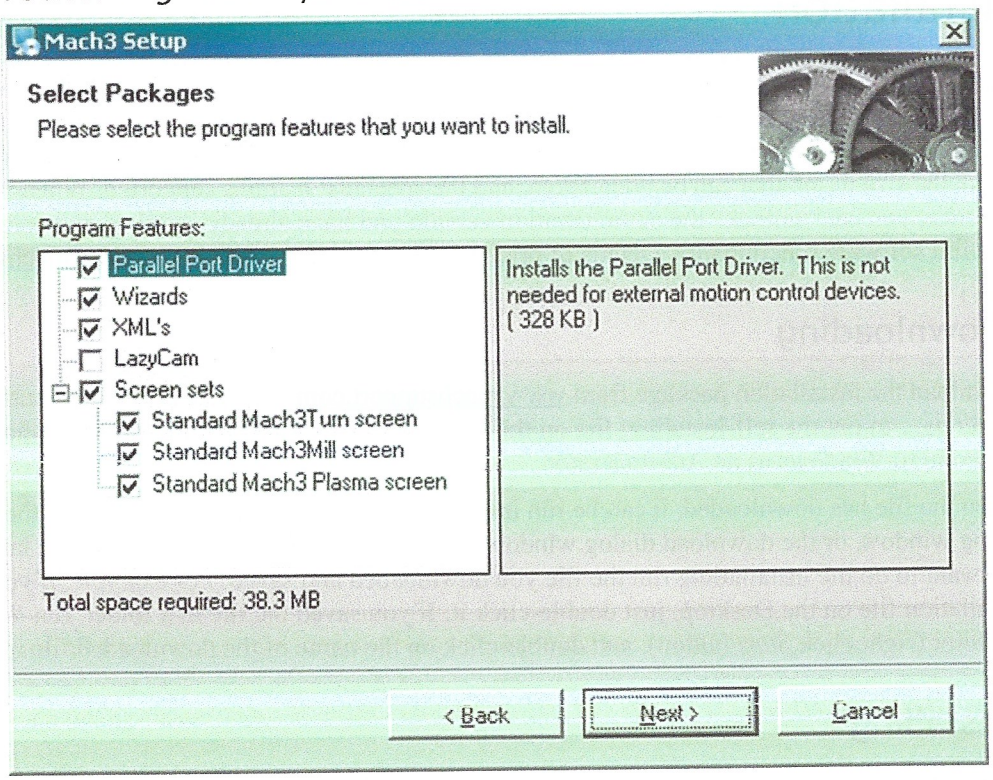
2.1.2.2 Run the Mach3 software installation package.

When you run the downloaded file, you will be guided through the usual installation steps for a Windows program such as accepting the license conditions and selecting the folder for Mach3. ArtSoft USA recommends that you allow Mach3 to use its default installation folder `C:\Mach3`.

The background image during installation is the standard Mach3Mill screen – do not worry if you are planning to control a lathe, as Mach3Turn is also being installed.

You will be asked if you want to install various program components, as shown in Figure 2-1:

Figure 2-1: Select Program Components Screen



You will need the parallel port driver if you are using the computer's parallel port(s) to interface with the machine tool. If you are using a motion controller board from a third-party vendor that uses USB or Ethernet, you should uncheck the parallel port driver.

The Wizards are a set of macros that let you quickly create GCode to do common tasks such a bolt circles, pockets, etc. You will almost certainly find these useful. Installing the Wizards also installs the Mach3 Addons for Mill, although they require a separate license for activation.

The XMLs are the files that hold Mach3's configuration information. There are three default `.XML` files: `Mach3Mill.xml`, `Mach3Turn.xml`, and `Mach3Plasma.xml`. These give you a known starting point for creating your own custom profiles. ArtSoft USA **STRONGLY** recommends you create custom profiles of your own instead of modifying the default profile(s). If you have previously modified one or more of the default profiles, however, and do not want to overwrite your configuration information, you should uncheck the XMLs box.

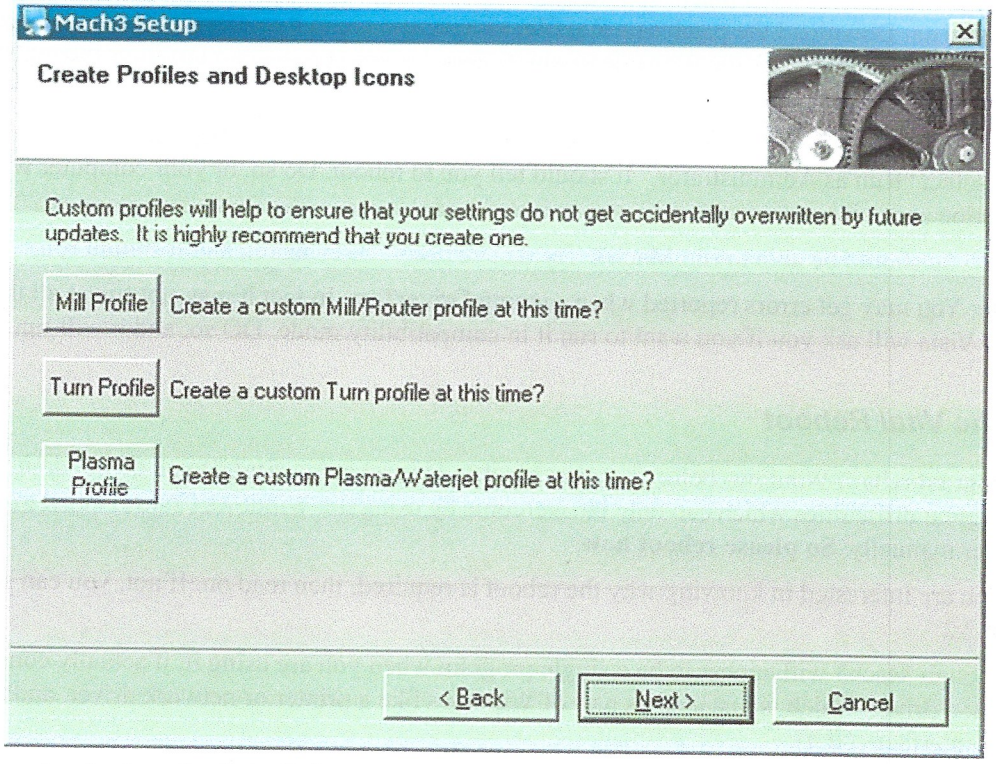
LazyCam is a beta-release free importer included with Mach3. Its purpose is to import standard dxf, cmx, and other file types to allow those that do not use CAM programs to more easily generate GCode to be run under Mach3. You do not need it to run the Mach3 software.

The Screen Sets define Mach3's screen appearance. Unless you have created your own custom screens, you will want these.

When you have selected the components you want, click the *Next* button.

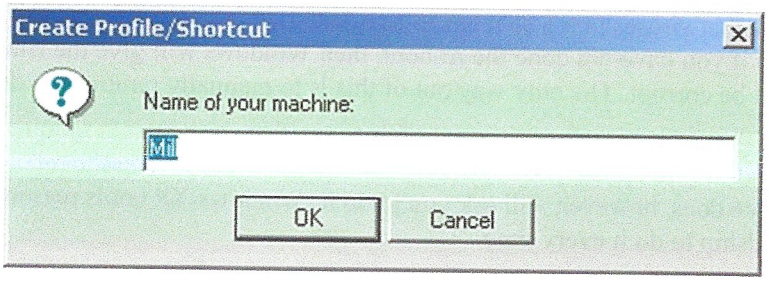
The installation procedure will ask if you want to create a custom profile, as shown in Figure 2-2:

Figure 2-2: Create a Custom Profile Screen



As previously described, ArtSoft USA strongly recommends that you create custom profiles instead of modifying the default profiles. This screen lets you clone one or more of the default profiles, assigning your own names to the cloned profiles. For example, if you click the Mill Profile button, the screen shown in Figure 2-3 appears.

Figure 2-3: Create Mill Profile



Enter the name you want to assign to the profile (possibly "MyMill") and click the *OK* button. If you want, you can create several different profiles. When you have created your custom profile(s), click the *Next* button.

2.1.2.3 If You're Using Windows Vista

Vista may require a registry patch for the Mach3 parallel port driver to run. (If you are using a third-party motion controller board that communicates with USB or Ethernet instead of the parallel port driver, you do not need the patch.) First, do the normal Mach3 installation, then install the patch. The patch is available at www.machsupport.com, together with any updated information about using Windows Vista. Download the patch as a Zip file, save it, and unzip it to extract the file `memoryoverride.reg`. Double-click the filename to run it. `memoryoverride.reg` modifies the registry to allow Mach3's driver to run.

Now, go to the `C:\Mach3` folder, (or wherever you installed Mach3). Right click `drivertest.exe` and select "Run as Administrator." It should tell you to reboot. Do so, or your computer will crash. No question about it.

Now you should be able to run Mach3. Try `drivertest.exe` again, and it should run.

Note: You may get errors reported when running DriverTest. In fact it may not run at all the first time, then Vista will ask you if you want to run it in compatibility mode. DO so, and it will run.

2.1.3 *The Vital Reboot*

You **must** reboot Windows before running Mach3. This reboot is **vital**. If you do not do it, 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. If not, you can skip to Section 2.2.

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. It runs normal user programs when it has nothing better to do itself. Because Mach3's operation is so time-critical, it 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 may be given attention up to 100,000 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 is to manually remove the driver.

Having given these dire warnings, it is only fair to say that the reboot is required only 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, so it is not much hardship to do it every time.

2.2 Testing The Installation

So you **have** rebooted! (If you haven't, go back and read Section 2.1.3.)

ArtSoft USA recommends you now test the installed system. As mentioned above, Mach3 is not a simple program. It takes great liberties with Windows 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 Mach3, and there may be other programs you are not even aware of 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), while others do not show themselves in any way. Other possible causes 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 spyware scanner such as Spybot, available from www.safer-networking.org, to locate and delete undesirable software on 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.

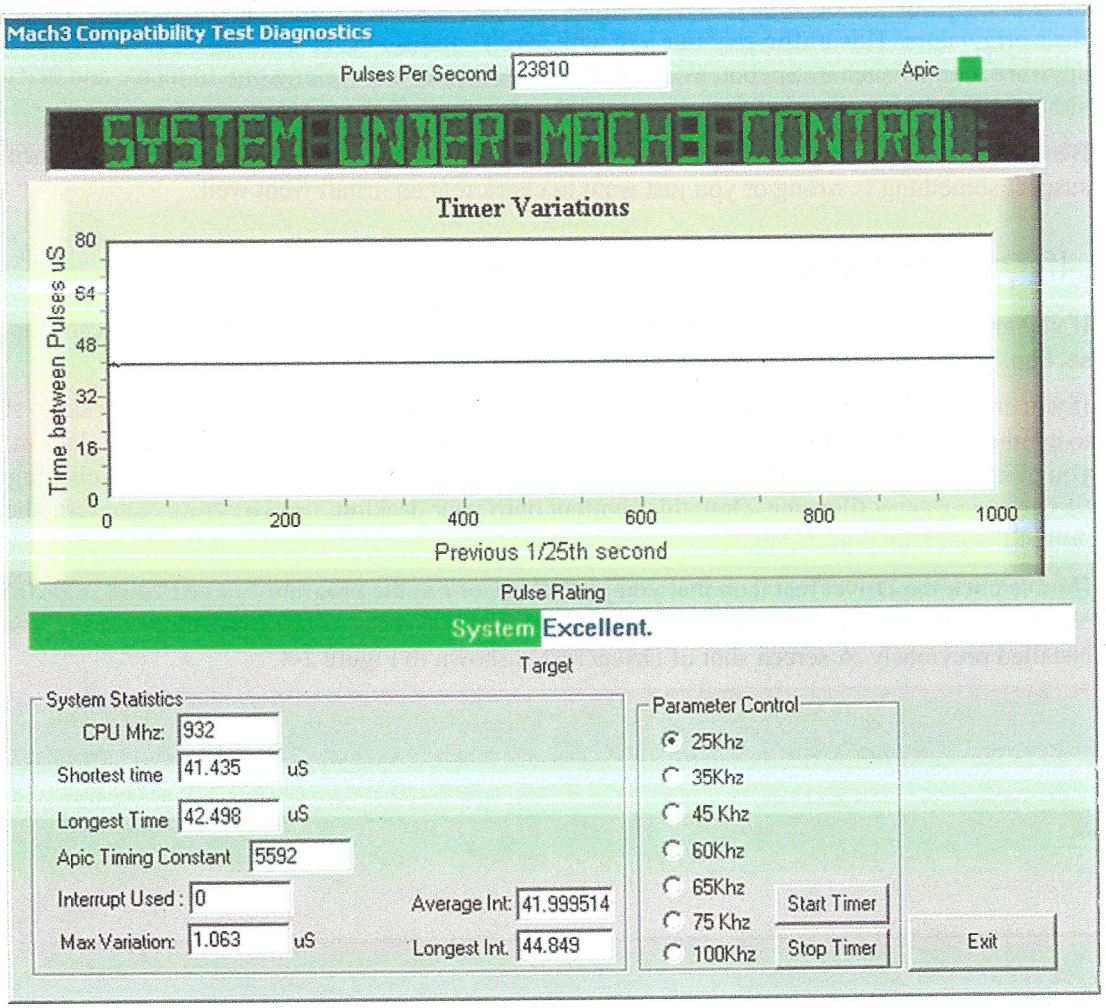
2.2.1 If You Are Using the Default Parallel Port Driver

If you are using a third-party motion controller in place of the parallel port driver, you can skip this section.

If you are using the Mach3 parallel port driver, it is worthwhile to set up an icon for a desktop shortcut to another Mach3 program. Use Windows Explorer (right-click *Start*), navigate to the folder in which you placed the Mach3 installation, and create a shortcut to `DriverTest.exe` by right-clicking on the `DriverTest.exe` filename. Drag this shortcut onto your desktop. `DriverTest.exe` tests the operation of the parallel port driver.

Double click the DriverTest icon that you just set up, or run the program `DriverTest.exe` from the Mach3 installation folder. Running `DriverTest.exe` will install the parallel port driver if it was not installed previously. A screen shot of DriverTest is shown in Figure 2-4.

Figure 2-4: The Running DriverTest Program



You can ignore all the boxes with the exception of the Pulses Per Second. It should be fairly steady around your chosen kernel pulse frequency (25,000 Hz, 35,000 Hz, etc.). Your pulse rate may vary, however, 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.

DriverTest evaluates the pulse stream and displays a Pulse Rating below the Timer Variations graph. In Figure 2-4, the pulse rating is Excellent. If your system has more variation, the pulse rating may be good, fair, or poor. If you see a screen similar to Figure 2-4, with only small spikes on the Timer Variations graph, a steady number of pulses per second, and a good or excellent rating, everything is working well. Close the DriverTest program and proceed to Section 2.3 describing Mach3 Profiles, below. If you have problems with the installation, refer to Section 2.4, *Installation Problems*.

2.3 Mach3 Profiles

Profile files (`.XML` files, stored in the `\Mach3` installation folder) define the operating appearance and characteristics of Mach3, allowing Mach3 to be configured for use with different types of machines: lathes, milling machines, plasma cutters, routers, etc. All your configuration setup choices are saved in the profile file that you select.

The installation wizard creates desktop icons for Mach3Mill, Mach3Turn, Plasma, and Mach3 Loader. Mach3Mill, Mach3Turn, and Plasma are shortcuts that run Mach3 with a preconfigured profile for a particular type of machine. The profile to use is identified by a "`/p`" argument in the shortcut target. (To see this command line, right-click on one of the desktop shortcuts and select Properties from the pop-up menu. See also Figure 2-7.)

While you can use one these shortcuts to start a preconfigured system, ArtSoft USA recommends that you do not. Instead, you should create your own profile(s). Doing so has two important benefits:

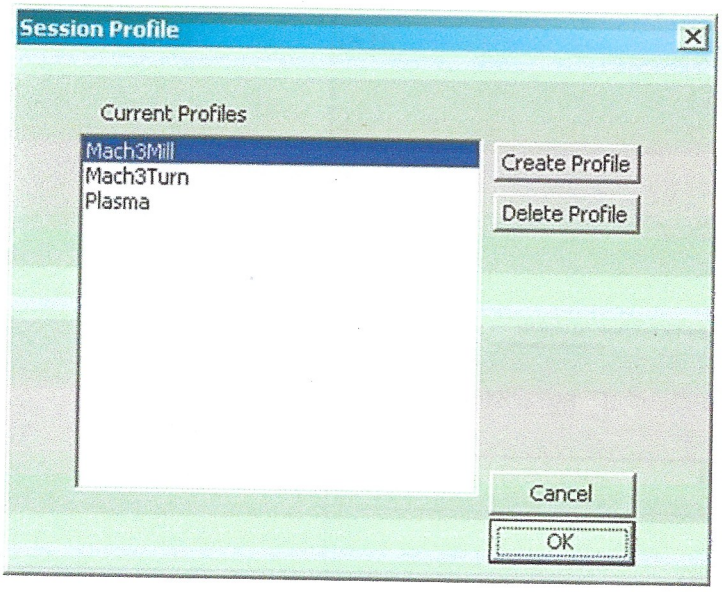
- The supplied profiles (`Mach3Mill.XML`, `Mach3Turn.XML`, and `Plasma.XML`) will not be modified by your configuration setup. They will always be a known starting point for creating additional profiles, and a recovery point if your own profile(s) becomes corrupted.
- Your profile(s) will not be overwritten and lost if you install an updated version of Mach3. During an update, the default profiles (`Mach3Mill.XML`, etc.) will be overwritten with a new version. If the old version of `Mach3Mill.XML` contained all your laboriously-entered configuration information, you will not be happy!

The Mach3 Loader shortcut has no preset profile choice. It runs Mach3 with a startup menu asking you to choose which profile to use. It also provides a way for you to create your own custom profiles.

2.3.1 Creating a Profile

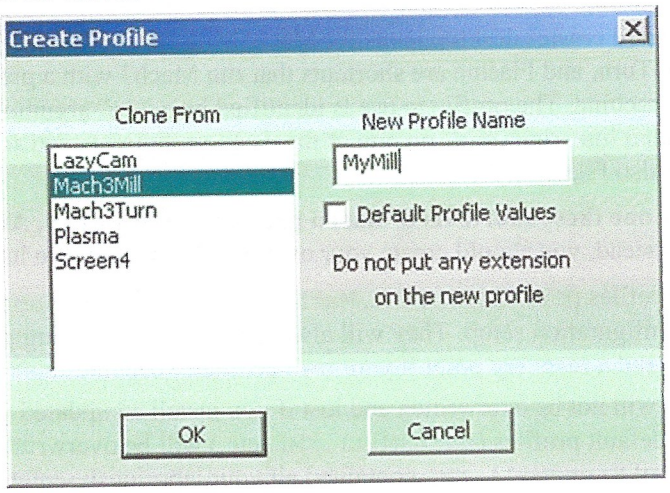
Run Mach3Loader using the preconfigured shortcut. The window shown in Figure 2-5 will appear.

Figure 2-5: Profile Selection Window



Click the *Create Profile* button. The window shown in Figure 2-6 will appear.

Figure 2-6: Create Profile Window

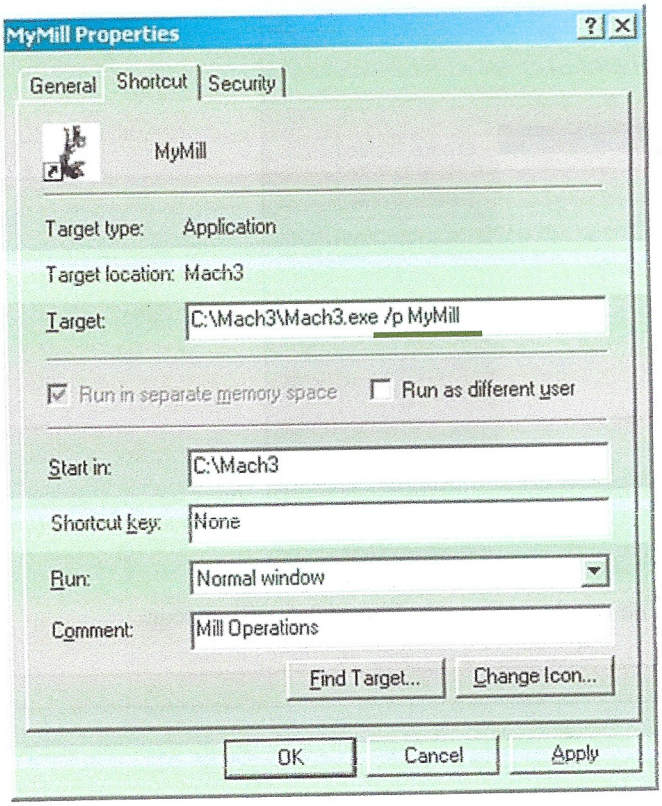


In the left-hand list, click on the profile you want to clone (in this example, it is Mach3Mill). Type the name you want to assign to the new profile in the *New Profile Name* box. Do not check the *Default Profile Values* box. (Selecting *Default Profile Values* produces a minimal profile.)

Click the *OK* button.

You can run Mach3 with your new profile by running Mach3Loader, selecting your profile name in the list, and clicking the *OK* button. For convenience, you may want to create a shortcut to Mach3 with your profile name in the command line, as shown in Figure 2-7.

Figure 2-7: Shortcut to "MyMill"



2.4 Installation Problems

Two things may occur when running the test that indicate a problem:

1. The display reads "Driver not found or installed, contact Art." This display will appear if the driver did not install into Windows. This can occur on XP systems that have a corruption of their driver database. The fix is to reinstall Windows. Or, you may be running Win2000. Win2000 has a bug/feature that can interfere with loading the driver. The driver may need to be loaded manually. See Section 2.4.2.
2. If the display reads "Taking over...3...2...1.." but then reboots, one of two things has occurred. Either you didn't reboot when asked during the installation of Mach3 (Told you! See Section 2.1.3), or the driver is corrupted or unable to be used in your system. In this case, follow the instruction in Section 2.4.2 and remove the driver manually, then re-install Mach3. If the same thing happens, please notify ArtSoft USA using the e-mail link on www.machsupport.com and you will be given guidance.

A few systems have motherboards that have hardware for the APIC timer but whose BIOS code does not use it. This will confuse the 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.

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.

2.4.1 Running 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.

2.4.2 Manual Driver Installation and Uninstallation

You need to read and do this section only 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.

1. Open the Windows Control Panel and double-click on the icon or line for *System*.
2. Select *Hardware* and click *Add Hardware wizard*. Windows will look for any new actual hardware (and find none).
3. Tell the wizard you have already installed it and then proceed to the next screen.

4. 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.
5. 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)*
6. The list you are shown will include an entry for *Mach1/2 pulsing engine*. Select this and go to the next screen.
7. Click *Have disc* and on the next screen point the file selector to your Mach3 folder (`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.

1. Open the Control panel and double-click on the icon or line for System.
2. Select *Hardware* and click *Device Manager*
3. You will be shown a list of devices and their drivers. *Mach1 Pulsing Engine* has the driver *Mach3 Driver* under it. Click on the + symbol to expand the tree if necessary. Right-click on *Mach3 Driver*. This will display a short menu that includes the option to uninstall it. Click *Uninstall*. This will remove the file `Mach3.sys` from the Windows folder. The copy in the Mach3 folder will still be there.

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 may need to delete the `.XML` profile file or files.

Chapter 3 Introducing the Mach3 Screens and Commands

You are now ready to try a "dry run" of Mach3. It will be easier to understand how to set up your actual machine tool after you have experimented a bit with the software. You can "pretend" to machine and learn a lot even if you do not have a CNC machine tool yet. If you do have one, then make sure it is not yet connected to the PC.

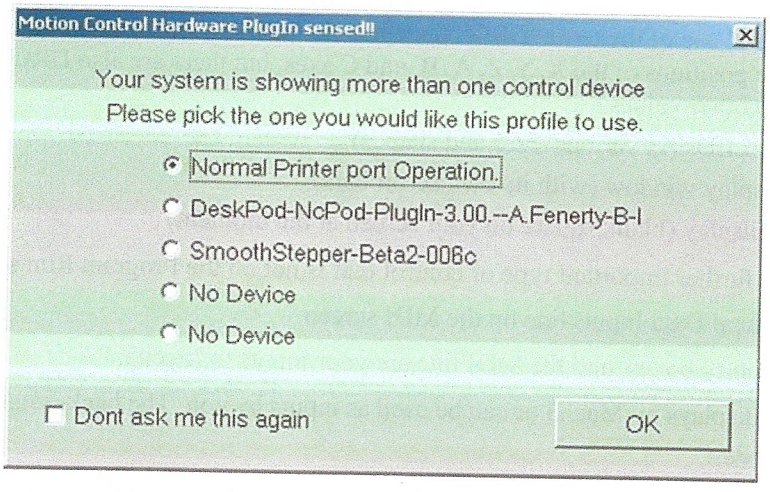
Mach3 is designed so that it is very easy to customize its screens to suit the way you work. This means that the screens you see may not look exactly like those in this manual if you bought a preconfigured system from a vendor. If there are major differences, then your system supplier should have given you a revised set of screenshots to match your system.

3.1 Screens

If you created your own custom profile as shown in Figure 2-2 and Figure 2-3, Mach3 will have created a shortcut icon on the desktop with the name of your custom profile. Double-click the appropriate icon to launch the program using that profile. You can also double-click the Mach3 Loader icon to run the program, select from a list the name of the profile to use, then click the *OK* button. (You did create your own profile as described in Section 2.3, didn't you? If not, go back and read that section.)

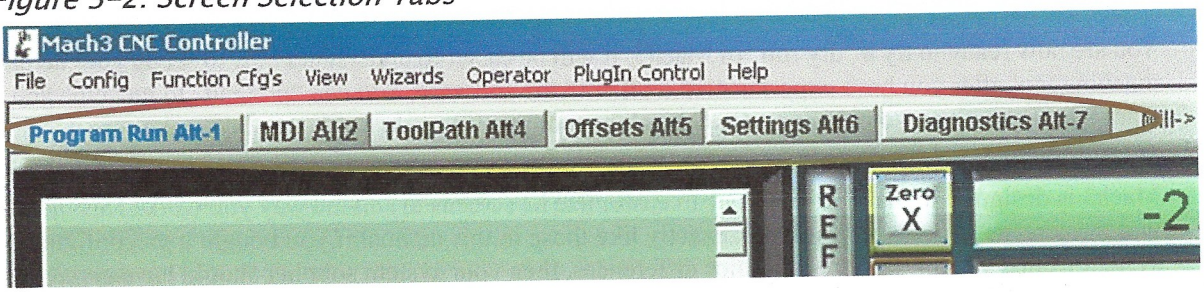
If you have installed more than one driver or third-party motion control plugin, you may see a screen similar to Figure 3-1. (The content of the screen you see will depend on what you have installed.) Select what you want to use by clicking the appropriate button, then click the *OK* button.

Figure 3-1: Select Control Device Screen



You should see the Mill Program Run screen. The other major screens, identified on tabs are MDI (manual data input), Tool Path, Offsets, Settings, and Diagnostics, as shown in Figure 3-2. Be sure the Program Run screen is selected; its name will be displayed in blue.

Figure 3-2: Screen Selection Tabs



Notice the red *Reset* button. It will have a flashing Red/Green LED (simulation of a light emitting diode) above it and some yellow LEDs lit. If you click the button, then the yellow LEDs will go out and the flashing LED turns to solid green. Mach3 is ready for action!

If you cannot reset, then the problem is probably something plugged into your parallel port or ports (a "dongle" perhaps), or the PC has previously had Mach3 installed on it with an unusual allocation of port pins to the Emergency Stop (EStop signal). By clicking on the *Offline* button you should be able to reset the system. **Most of the tests and demonstrations in this chapter will not work unless Mach3 is reset out of EStop mode.**

3.1.1 Types of Objects on Screens

You will see that the Program Run screen is made up of the following types of objects:

- Buttons (for example, Reset, Stop Alt-S, etc.)
- DROs, or Digital Readouts. **Anything with a number displayed will be a DRO**. This may be a more general use of the term "DRO" than you are accustomed to. The main DROs are of course the current positions of the X, Y, Z, A, B, and C axes, but there are also DROs for feed rate, spindle speed, and other values.
- Simulated LEDs (in various sizes and shapes)
- GCode display window (with its own scroll bars)
- Toolpath display (blank square on your screen at the moment)

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

- MDI (Manual Data Input) line on the MDI screen

Buttons, data entry boxes, and the MDI line are your inputs to Mach3.

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

The GCode window and Toolpath displays provide information from Mach3 to you. You can, however, manipulate both of them (for example, scroll the GCode window and zoom, rotate, and pan the Toolpath display).

3.1.2 Using Buttons and Shortcuts

Most standard screen buttons have a keyboard hotkey, which may be a single key or a key combination. This is often shown as part of the name on the button itself or in a label near it. For example, the shortcut to go to the MDI screen is *Alt-2*. Pressing the named key or key combination when the screen is displayed is the same as clicking the button with the mouse. You might like to try using the mouse and keyboard shortcuts to turn on and off jogging, to turn on Flood coolant, and to switch to the MDI screen. Notice that letters are sometimes combined with the *Control* or *Alt* keys. Although letters are shown as uppercase (for ease of reading), **do not** use the shift key when typing the shortcuts.

In a workshop, it is often convenient to minimize the times when you need to use a mouse. Physical switches on a control panel can be used to control Mach3 by use of a keyboard emulator board (for example, the Ultimarc IPAC). This connects in series with your keyboard and will send Mach3 "pretend" key presses to activate buttons using the keyboard shortcuts.

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 those are set up.

3.1.3 Data Entry to DROs

You can enter new data into any DRO by clicking in it with the mouse, by 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).

Try entering a feedrate like 45.6 on the Program Run screen. Click in the feedrate box and type the numbers. You **must** press the *Enter* key to accept the new value, or press the *Esc* key to revert to the previous one. *Backspace* and *Delete* are not active when inputting to DROs.

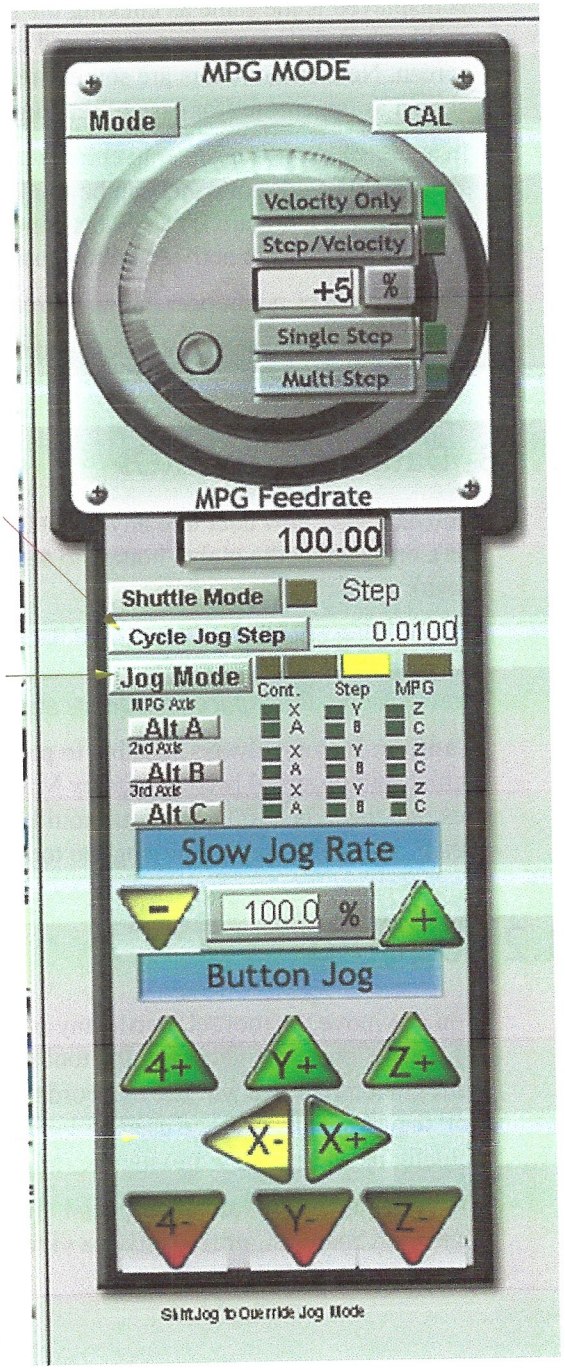
Caution: It is not always sensible to put your own data into a DRO. For example, the display of your actual spindle speed is computed by Mach3. Any value you enter will be overwritten. You can put values into the axis DROs, but you should not do it until you have read the *Using Mach3Mill* manual in detail. It is **not** a way of moving the tool!

3.2 Jogging

You can move the tool relative to any place on your work manually by using various types of jogging. Of course, on some machines, the tool itself will move and on others it will be the machine table or slides that move. We will use the words "move the tool" here for simplicity. Whether the tool moves or not, the frame of reference used will assume tool movement. That is, "move the tool to the left" may actually be achieved by moving the table to the right.

The jogging controls are on a special "fly-out" screen. This is shown and hidden by using the *Tab* key on the keyboard. Figure 3-3 gives a view of the flyout.

Figure 3-3: Control Buttons Flyout



You can also use the keyboard for jogging. The arrow keys are set by default to give you jogging on the X and Y axes and Pg Up/PgDn jogs the Z axis. You can re-configure these keys to suit your own preferences. You can use the jogging keys on any screen with the *Jog ON/OFF* button on it.

In Figure 3-3, you will see that the Step LED is shown lit. The *Jog Mode* button toggles between *Continuous, Step,* and *MPG* modes,

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 Percentage* DRO. You can enter any value from 0.1% to 100% to get whatever speed you want. The *Up* and *Down* screen buttons beside this DRO will alter its value in 5% steps. If you depress the *Shift* key, then the jogging will occur at 100% speed whatever the override setting. This allows you to quickly jog to near your destination.

In Step mode, each press of a jog key will move the axis by the distance indicated in the *Step* DRO. You can cycle through a list of predefined Step sizes with the *Cycle Jog Step* button. Movement will be at the current Feedrate.

Rotary encoders can be interfaced (via the parallel port input pins) to Mach3 as Manual Pulse Generators (MPGs). It is used to perform jogging by turning its knob when in *MPG* mode. The buttons marked *Alt A*, *Alt B,* and *Alt C* cycle through the available axes for each of three MPGs, and the LEDs define which axis is currently selected for jogging.

The another option for jogging is a joystick connected to the PC games port or USB. Mach3 will work with any Windows-compatible "analog joystick" (so you could even control your X axis by a Ferrari steering wheel!). The appropriate Windows driver will be needed for the joystick device. The 'stick is enabled by the *Joystick* button and, for safety, must be in the central position when it is enabled.

If you have an actual joystick and it has a throttle control, then this can be configured either to control the jog override speed or the control the feed rate override (see Chapter 5 again). Such a joystick is a cheap way of providing very flexible manual control of your machine tool. In addition, you can use multiple joysticks (strictly Axes on Human Interface Devices) by installing manufacturer's profiler software or, even better, the KeyGrabber utility supplied with Mach.

Now would be a good time to try all the jogging options on your system. Don't forget that there are keyboard shortcuts for the buttons, so why not identify them and try them. You should soon find a way of working that feels comfortable.

3.3 Manual Data Input (MDI) and Teaching

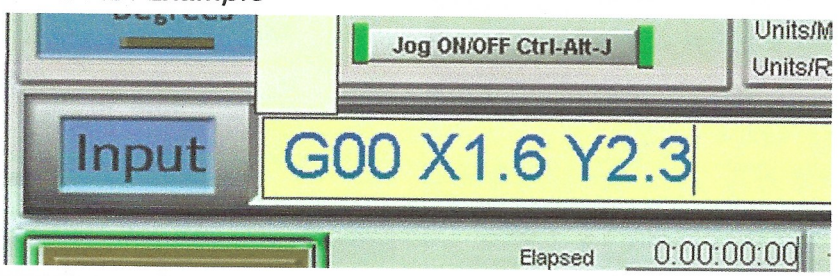
Use the mouse or keyboard shortcut to display the MDI (Manual Data Input) screen.

This has a single line for data entry. You can click in it to select it or 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.

If you know some GCode commands, you can try them out. Or you can try:

```
G00 X1.6 Y2.3
```

Figure 3-4: MDI Example



That command will move the tool to coordinates X = 1.6 units and Y = 2.3 units. (It is G zero not G letter O.) You will see the axis DROs move to the new coordinates.

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

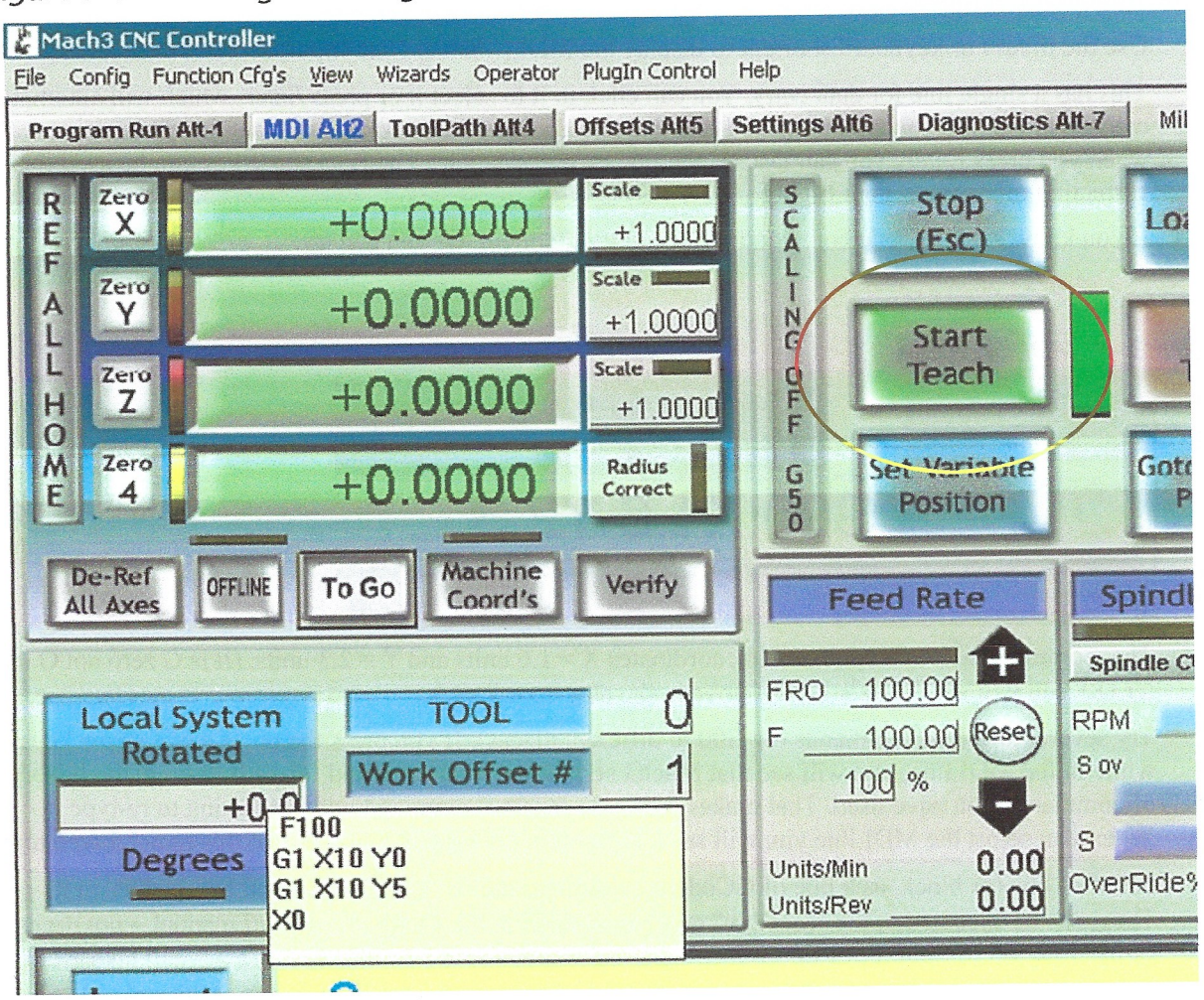
An MDI line (or block, as a line of GCode is sometimes called) can have several commands on it. They will be executed in the "sensible" order as defined in the *Using Mach3Mill* manual – not necessarily from left to right. For example setting a feed speed using a command like F2.5 will take effect before 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, enter each MDI command on a separate line.

3.3.1 Teaching

Mach3 can remember a sequence of lines that you enter using MDI and write them to a file. This file can then be run again and again as a GCode program.

On the MDI screen, click the *Start Teach* button. The LED next to it will light to remind you that you are teaching. Type in a series of MDI lines. Mach3 will execute each command as you press *Enter* after each line. Mach3 will store the sequence of commands you enter in a named Teach file.

Figure 3-5: Teaching a Rectangle

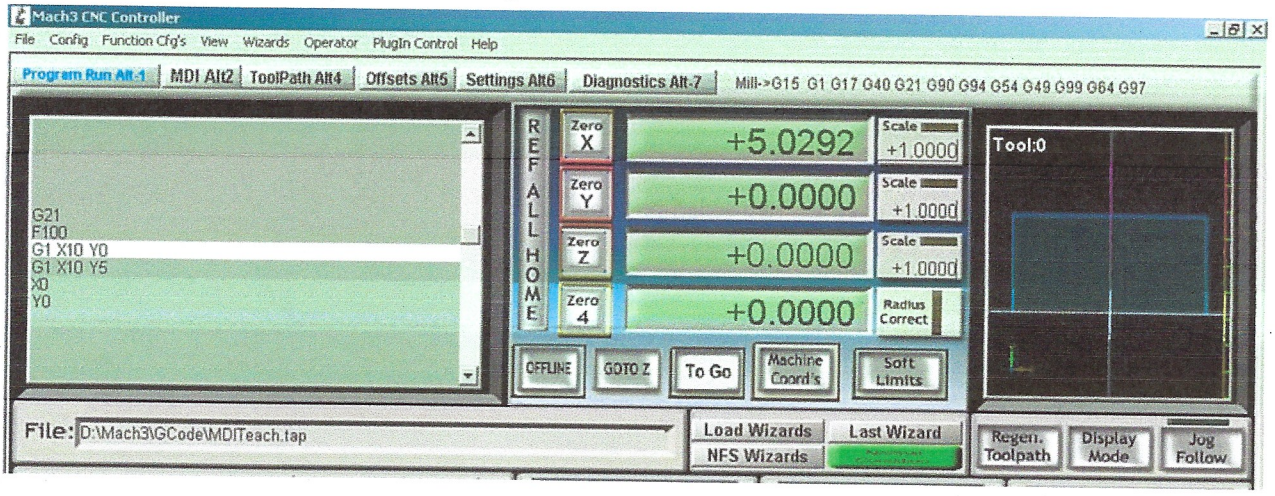


You can type your own code, or try:

```
g21
f100
g1 x10 y0
g1 x10 y5
x0
y0
```

All the 0 are zeros, not capital Os. When you have finished, click *Stop Teach*. Next, click *Load/Edit* and go to the Program Run screen. You will see the lines you typed displayed in the GCode window (Figure 3-6). If you click Cycle Start, then Mach3 will execute your program.

Figure 3-6: Taught Program Running



If you use the editor, you will be able to correct any mistakes and save the program in a file of your own choosing.

This is a blank left-hand page for two-sided printing.

Chapter 4 Hardware Requirements and Connecting the Machine Tool

This chapter tells you about the hardware aspects of connecting a machine tool to your PC. Chapter 5 gives details of configuring Mach3 to use the connected hardware.

If you 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 should have given you some documentation explaining how to connect the parts of your system together.

Read this chapter to learn what Mach3 expects it is going to control and how you can connect standard components like stepper motor drivers and micro-switches to construct a CNC system. The descriptions 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 may not want to concern yourself with sections following Section 4.5, *Limit and Home Switches*, and keep in mind the information in Section 1.4, *What Mach3 Can Do*, as you plan you own configuration.

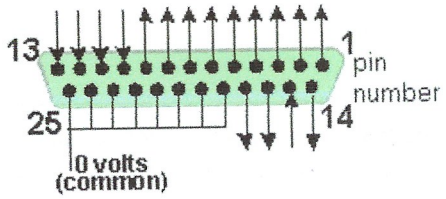
4.1 The PC Parallel Port and its History

Mach3 interfaces to the motor drivers, limit switches, and other hardware through your PC's parallel port(s). This section describes the characteristics of the parallel port

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 data, it has been used for many things other than connecting printers. You can transfer files between PCs, attach copy protection "dongles," connect peripherals such as scanners and Zip drives, and of course control machine tools using it. The USB interface is taking over many of these functions, and this conveniently leaves the parallel port free for Mach3.

The parallel port connector on the PC is a 25-pin female "D" connector. The connector, as seen from the back of the PC, is shown in figure 4.1. The arrows give the direction of information flow relative to the PC. Thus, for example, pin 15, the second pin from the right on the bottom row, is an input to the PC.

Figure 4-1: Parallel Port Female Connector (seen from back of PC)



Note: Convertors which plug into a USB port and have a 25 pin connector will **not** drive a machine tool using Mach3, even though they are perfectly suitable for the simpler task of connecting a printer.

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 line, which is connected to pins 18 to 25 of the port connector.

The first successful 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. 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. As is explained below, however, letting "lo" = one is usually more desirable in most practical interface circuits.

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 limit 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 lower 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".

The net result is 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 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 seen 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 "optimized" design. A good isolating breakout board (see Section 4.2.1, *Isolating Breakout Boards*) will protect you from these electrical compatibility problems.

4.2.1 Isolating Breakout Boards

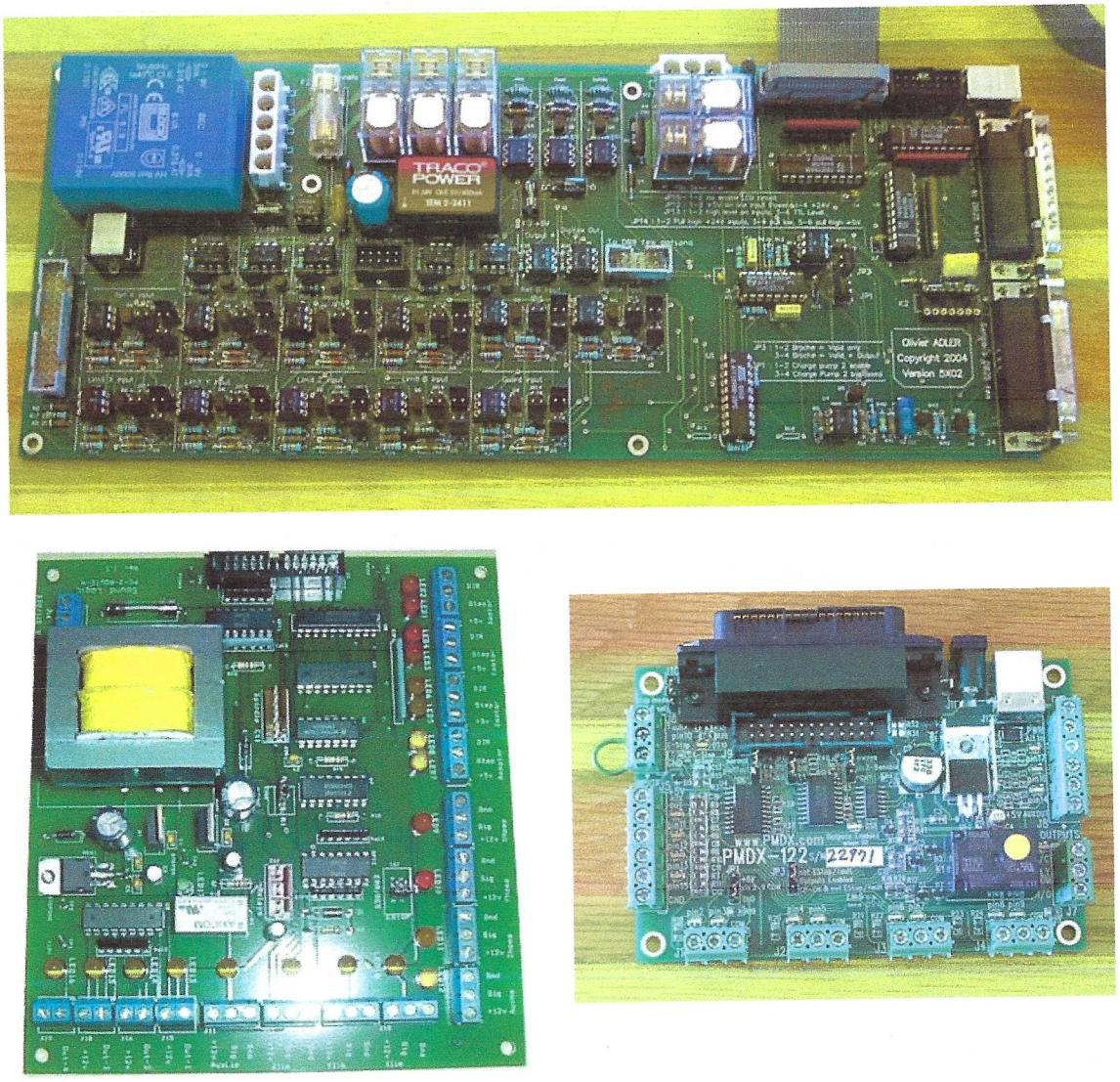
Even if you skipped the previous section you had better read this one!

You will see that pins 18 to 25 of the parallel port 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 induced currents flowing in them that can create voltage "noise" and can cause errors. The interference might 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 of many amps. Properly connected they will do no harm to the computer but an accidental short circuit could easily destroy the entire computer motherboard and even the CD and hard drives as well.

For these two reasons ArtSoft USA strongly advises you to use an interface board called an "isolating breakout board." This will provide 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 minimize the risk of interference to your neighbors' radio and television signals. If you build a "rat's nest," then you are inviting short circuits and tragedy. Figure 4-2 shows three representative commercial breakout boards.

Figure 4-2: Three Examples of Commercially Available Breakout Boards



(Boards not necessarily to same scale)

4.3 The EStop Control

Every machine tool should have one or more Emergency Stop (EStop) buttons, usually with a big red mushroom head. They should be fitted so that you can easily reach one from wherever you might be when you are operating the machine. This is particularly vital on a CNC 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 software – so we are talking about relays and contactors. The circuit should tell Mach3 what you have done, and there is a special, mandatory input for this. It will generally not be good enough to just turn off the AC power for an EStop event, 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 start when you release it by turning its head.

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

4.4 Axis Drive Options

4.4.1 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 drive the axes through leadscrews (plain- or ball-nut), belts, chains, or rack and pinion. The mechanical drive method will determine the speed and torque required from the motors and hence any gearing required between the motor and machine.

Properties of a bipolar stepper motor drive include:

- Low cost.
- Simple 4-wire connection to motor.
- Low maintenance.
- 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).
- 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.
- 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. In practice, stepper motor drives give satisfactory performance with conventional machine tools up to a Bridgeport turret mill or a 12" swing (6" center height) lathe unless you want exceptional accuracy and speed of operation.

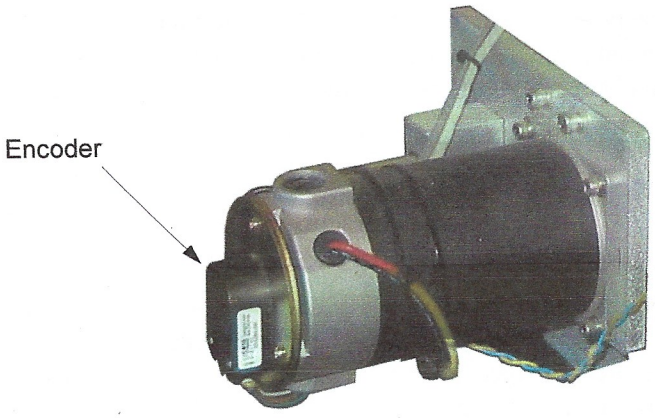
On the other hand, a servo motor drive:

- Is relatively expensive (especially if it has an AC motor).
- Needs wiring for both the motor and encoder.
- Requires maintenance of brushes on DC motors.
- Allows motor speed of 4000 rpm or more, and a practically unlimited torque (if your budget can stand it!)
- Provides closed loop control so drive position is always known to be correct (or a fault condition will be raised if an error occurs).

It is worth giving two warnings here. First, servo systems on old machines are probably not digital. That is, they are not controlled by a series of step pulses and a direction signal as required by Mach3. To use an old motor with Mach3, you will need to discard the resolver (which gave the position) and install a quadrature encoder, and you will have to replace **all** the electronics. Second, beware of second-hand 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 demagnetized and so be useless. Unless you are really confident of your skills and experience, buying used drive motors is likely to be false economy and a source of unending trouble. Any amount of money saved, and more, may well be spent in a futile and frustrating effort to patch together a working system. ArtSoft USA recommends that the axis drives be current products, bought from suppliers who will support them. If you buy **right** then you will only need to buy **once**, and you will be up and running sooner.

Figure 4-3: Example of Servo Motor with Encoder



4.4.2 Determining Axis Drive Requirements

A full set of calculations for the axis drive requirements would be very complicated, and you probably do not have all the necessary data (for example, what is the maximum cutting force you want to use). Some calculation is, however, necessary for success. As a minimum, you need to determine how many motor steps it will take to move an axis a specified distance. The answer depends on the machine hardware configuration (for example, the pitch of the leadscrew) and the motor characteristics (how many steps it makes per revolution).

4.4.2.1 Example 1 - Mill Table Cross Slide

Start by defining the minimum possible move distance desired. This is an absolute limit to the accuracy of work done on the machine. Then check rapid speeds and torque.

For example, suppose you are designing a mill cross-slide (Y axis) drive, and the cross slide has a total travel of 12". You are going to use a screw with a 0.1" pitch single start thread and a ball nut. You want the machine to be able to do a minimum move of 0.0001". One full turn of the 0.1" pitch screw will move 0.1", so a movement of 0.0001" is 1/1000 of that. This will require 1/1000 of a revolution of the motor shaft if it is coupled directly to the screw.

Using a Stepper Motor

The minimum step with a stepper motor depends on how it is controlled. Many commonly available stepper motors have 200 full steps per revolution, but controllers also permit "micro-stepping." Micro-stepping helps give smooth running over the full range of feed speeds, and many controllers will allow you to have 10 micro-steps per full step. A 200-step motor with 10 micro-steps per full step would allow 1/2000 of a revolution as the minimum step. In our example above, two micro-steps would give the desired 0.0001" minimum move. This must be viewed with some caution, however. As the number

of microsteps per step increases, the torque falls off rapidly. Depending on the load being imposed on the motor, there may not be enough torque to actually move the motor by a single microstep. It may be necessary to make several microsteps before there is sufficient torque. In brief, you should not count on microstepping to achieve fine accuracy. The primary benefits of microstepping are reduced mechanical noise, gentler actuation, and reduced resonances problems.

Next, look at the possible rapid feed speed. Assume, conservatively, that the maximum motor speed is 500 rpm. With our example 0.1" pitch leadscrew, 500 rpm would give a rapid feed of 50 inches/minute, or just under 15 seconds for the full 12" slide travel. This would be satisfactory, although not spectacular.

At that speed, the micro-stepping motor drive electronics would need 16,667 pulses per second. (500 rpm * 200 steps per rev * 10 microsteps per step / 60 seconds per minute)
On a 1 GHz PC, Mach3 can generate 35,000 pulses per second simultaneously on each of the six possible axes. So there are no problems here.

You now have to determine the torque that the machine will require, which will determine the size of motor required. 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 (or set of spring kitchen scales). The torque for the cut (in ounce-inches) is the balance reading (in ounces) x 12. An easier and perhaps more reliable way, if the information can be found, is to use a motor size and specification that you know works on a similar machine with the same type of slide and screw. Since a stepper motor, if overloaded, can have "lost steps" and resulting inaccuracy, be sure to use a motor of adequate size with plenty of torque for the application.

You can also increase available torque by gearing. If your rapid feed speed calculation results in a reasonable value, 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. That could permit the use of smaller (cheaper) motors, at the expense of some performance.

Using a 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 which will generate four "quadrature" pulses for each slot in the disc. Thus a disc with 300 slots, for example, generates 300 cycles per revolution (CPR). This is fairly low for commercial encoders. The encoder electronics would output 1200 quadrature counts per revolution (QCPR) of the motor shaft. With our 0.1" pitch leadscrew, one step would give a minimum movement of 0.000083", which is better than the 0.0001" desired.

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 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 will also decrease the movement per step, and in our example would give a movement of 0.000083" / 5, or 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 9 seconds for 5" travel of the slide. Notice, however, that the speed is limited by the pulse rate from Mach3, **not** by the motor speed. This is 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, check on available torque. A servo motor requires less safety margin than a stepper motor because the servo cannot suffer from "lost steps." If the torque required by the machine is too high, however, the motor may overheat or the drive electronics raise an over-current fault.

4.4.2.2 Example 2 - Router Gantry Drive

A gantry router might need a travel of at least 60" on the gantry axis. A ballscrew for that length would be expensive and difficult to protect from dust. Many designers would opt for a chain and sprocket drive.

We might choose a minimum step of 0.0005". A drive chain sprocket of 20 teeth with $^1/_4$" pitch chain gives 5" gantry movement per revolution of the sprocket. A stepper motor (ten micro-steps) gives 2000 steps per revolution so a 5:1 reduction (belt or gear box) is needed between the motor and sprocket shaft to make one step equal to 0.0005" of travel [0.0005" = 5"/(2000 x 5)], and because of the 5:1 gear reduction one revolution of the stepper motor will result in 1" of travel.

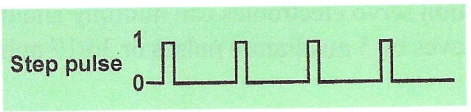
With this design, if we get 500 rpm from the stepper, travel will be 500 inches per minute, or 8.33 inches per second. The rapid feed of 60" would, neglecting acceleration and deceleration time, take a reasonable 7.2 seconds. [60" / 8.33 = 7.2]

The torque calculation on this machine is more difficult than with the cross slide because, with the mass of the gantry 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 ArtSoft USA Mach1Mach2CNC user group on Yahoo!, or join the General Discussion forum on www.machsupport.com, you will have access to the experience of hundreds of other users.

4.4.3 How the Step and Dir Signals Work

Mach3 puts out one pulse (logic 1) on the Step output for each step that the axis is to make. The Dir output will have been set before the step pulse appears.

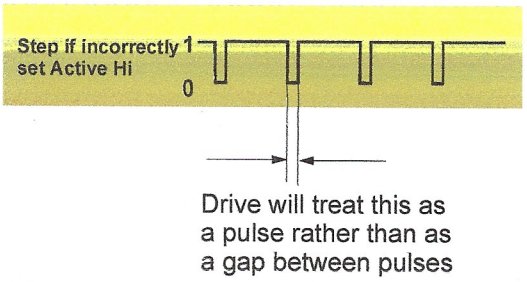
Figure 4-4: Step Pulse Form (Active Lo)



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.

Drive electronics usually use the Active Lo configuration for Step and Dir signals. Mach3 should be set up so these outputs are Active Lo. If this is not done, the Step signal will still go up and down but the drive will interpret the gaps between the pulses as the pulses, and vice-versa. This often causes very rough or unreliable running of the motor. The "inverted" pulses are shown in Figure 4-5.

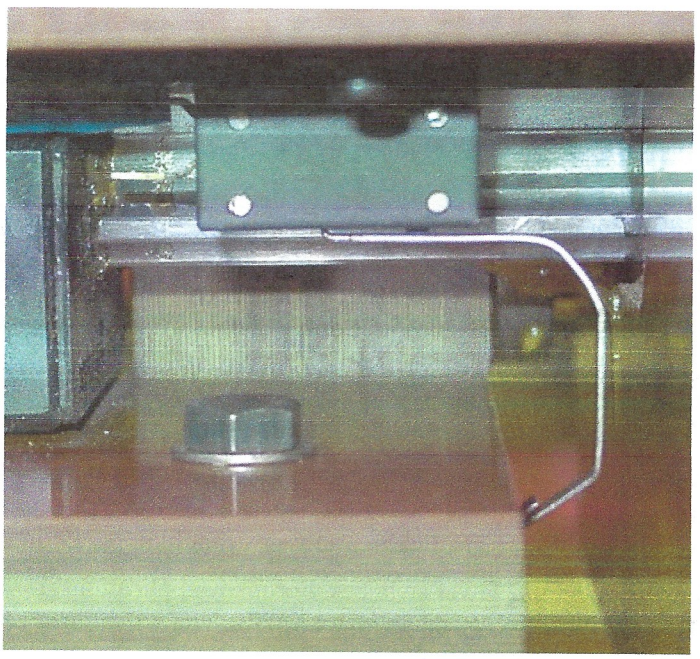
Figure 4-5: Inverted Pulse Form (Active Hi)



4.5 Limit and Home Switches

Limit switches are used to prevent any linear axis from moving too far and causing damage to the structure of the machine. You can run a machine without them, but the slightest mistake in setting up or programming can cause a lot of expensive damage. Figure 4-6 shows a Limit switch on a milling machine.

Figure 4-6: Example of a Mechanical Limit Switch



A microswitch mounted on the table is tripped when its lever hits the bed of the machine. This setup would not be rigid enough for good position repeatability.

An axis may also have a **Home switch**. Mach3 can be commanded to move one (or all) axes to the Home position. This needs to be done whenever the system is switched on so that it knows where the axes are currently positioned. If you do not provide Home switches, then you will have to jog the axes by eye to a reference position. The Home switch for an axis can be at any coordinate position, and you define this location. Thus, the Home switches do not have to be at Machine Zero.

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 shown in Figure 4-6 would be very imprecise. Repeatability is very important for a switch used for the Home function.

Each axis could therefore need three switches (i.e. two Limit switches at the two ends of travel and a Home switch). Even a basic mill would require nine parallel port inputs for them. This is not practical, as a parallel port has only five inputs! The problem of insufficient inputs can be solved in three ways:

1. Connect the limit switches to external logic (perhaps in the drive electronics), and use this logic to switch off the drives when a limit is reached rather than interfacing them to Mach3. Separate reference switches would still be connected inputs to Mach3.
2. Use one pin to share all the inputs for an axis, and make Mach3 responsible for controlling both limits and detecting Home. For example, if Mach3 was instructed to "move to Home" on a milling machine, it could move the X axis to the left (table to the right) until a switch was triggered. In context, that would be interpreted as "Home." If that same switch was triggered while machining, however, it would be interpreted as "exceeded Limit."
3. Interface the switches by a keyboard emulator.

The first method is best and mandatory for a very large, expensive, or fast machine where you cannot trust 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 is hit. This is safer than disabling the limits so a user can jog the machine off the limits, but it does rely on having a sophisticated drive.

The second method is suitable for smaller machines with direct operator control. With this setup, it is possible to use only three inputs to Mach3 for a three-axis mill (four for a gantry type machine - see Slaving). Since you (or the operator) are present and can take appropriate action based on the situation, only two switches are required per axis. Both Limit switches can be tied together, and the Home position can share functionality with one of the Limit switches. All the two switches need to do is signal Stop! The operator can then figure out what to do for recovery. Refer to Section 4.5.3, *How Mach3 Uses Shared Switches* for more discussion of shared switches.

The keyboard emulator has a much slower response time than the parallel port, but that solution is satisfactory for limit switches on a machine without high speed feeds. For details of the architecture see *Mach3 Customization Wiki*. There is a link to it at www.machsupport.com.

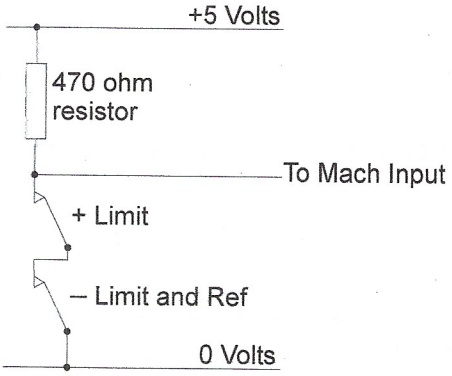
There is also a fourth option: use fewer switches. For many applications, for example, Home switches are of little use. If the machine works on a variety of jobs, you will almost certainly have to locate a different "part zero" for each job that will have no relationship to any "Home" position that might be defined. Unless they are meaningful for your particular application, there is no point in installing them.

4.5.1 The Switches

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

If you are going to have two switches 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 4.7, then they will give an Active Hi signal if either switch is operated. Refer to the schematic in Figure 4-7.

Figure 4-7: Two Normally Closed Switches Give a Logical OR



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 470 ohms is shown which gives a current of about 10 milliamps. As the wiring to the switches may be quite long and liable to pickup of electrical 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).

Optical switches, if out of the way of coolant, should be OK on a metalworking machine, but are liable to malfunction with wood dust. Figure 4-8 shows an example of an installed optical switch.

ArtSoft USA recommends against magnetic switches (reed switches or Hall effect devices) on a machine that may cut ferrous metal, because the swarf will "fuzz-up" the magnet.

Figure 4-8: Optical Switch on Table with Vane on Bed of Machine

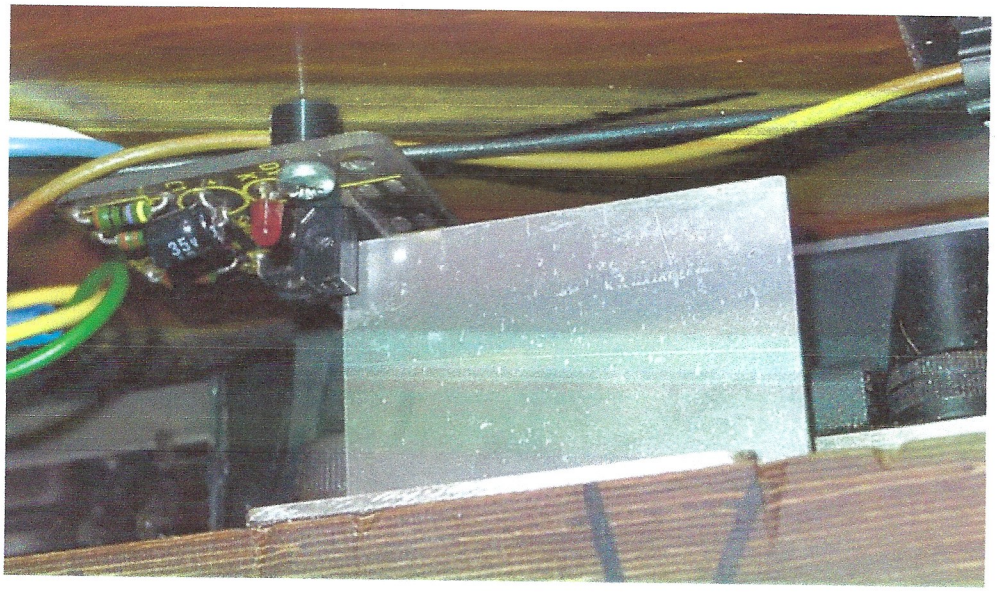
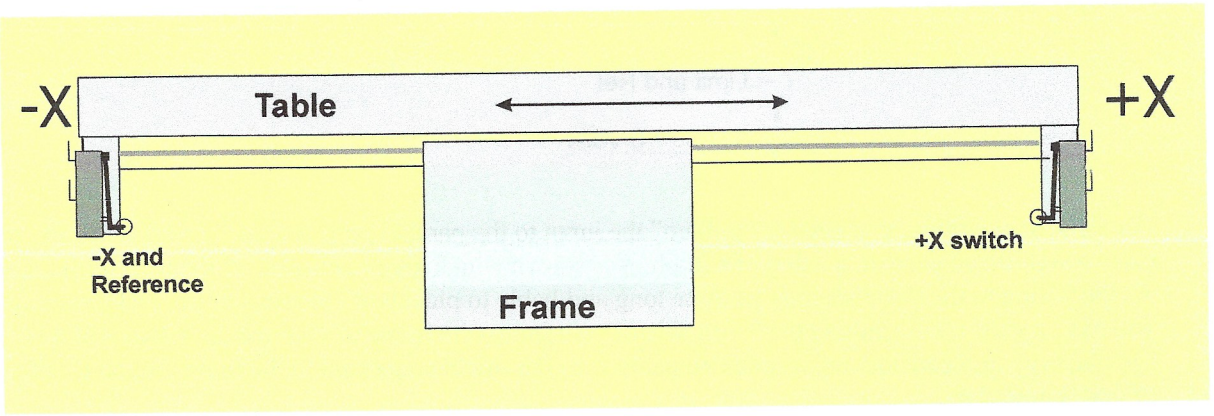
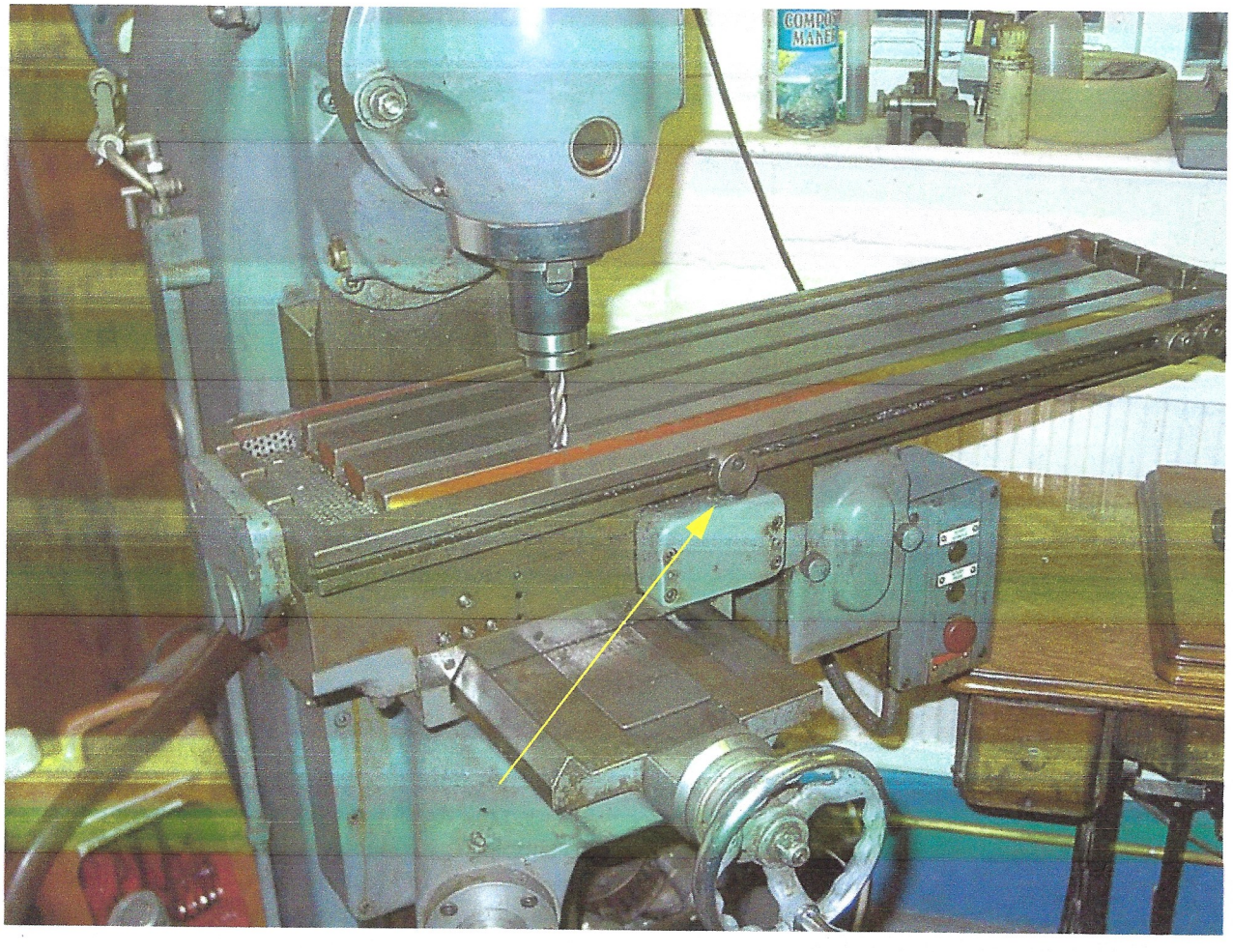


Figure 4-9: Two Switches Operated By Frame with Overtravel Prevented by Mechanical Stops



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. On an optical switch like Figure 4-8, there will be no difficulties provided the vane is long enough. A microswitch can be given arbitrary overtravel by operating a roller on it by a ramp (see Figure 4-11). The slope of the ramp does, however, reduce the repeatability of operation of the switch. It is often possible to use one switch for both limits by providing two ramps or vanes, as shown in Figure 4-11.

Figure 4-10: Mill with Tool at X=0, Y=0 Position (Note the Dog Is on Limit Switch)



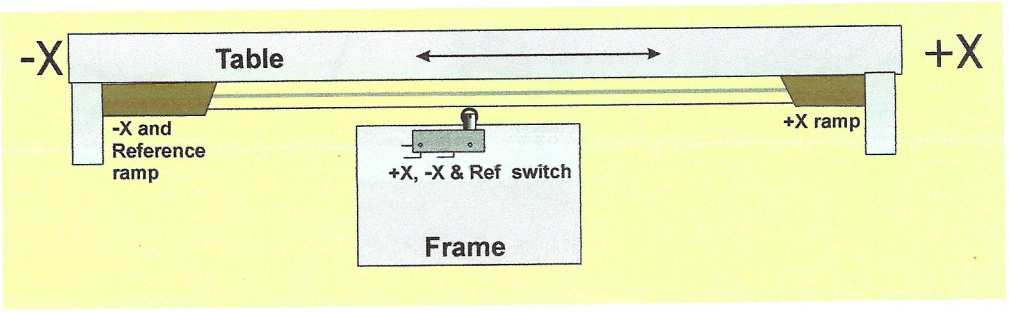
4.5.2 Where to Mount the Switches

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, the switches in Figure 4-6 and Figure 4-8 are both mounted under the table even though they need a flexible cable, as they are much better protected there.

You may find it convenient to have one moving cable with the wires in it for two or more axes (e.g. the X and Y axes of a gantry router could have switches on the gantry itself and a very short cable loop for the Z axis could then join the other two). Do not be tempted to share a multi-way cable between motor and switch wiring. You may want to run two separate cables together. This will not cause trouble if both are shielded (with braid or foil), and the shields are grounded to one common point at the electronic drives.

Figure 4-11: Ramps Operating One Switch



You may find it helpful to look at commercial machines and pictures of examples on the ArtSoft USA Yahoo! group for more ideas and techniques for switches. www.machsupport.com has a link to the Yahoo! group.

4.5.3 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.

For a full understanding of this you will also have to read the section in Chapter 5 on configuring Mach3, but the basic principle of sharing switches is simple. You connect the two limit switches to one input (or have one switch and two vanes or ramps). You define, to Mach3, a direction as the direction to travel to move when looking for a Home reference switch. The limit switch (vane or ramp) at that end of the axis is also 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 axis unless:

- Auto Limit override is switched on (by a toggle button on the Settings screen). In this case, you can click Reset and jog off the Limit switch. You should then re-reference the machine position.
- You click Override Limits button. 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. You should then re-reference the machine position. An input can also be defined to override the Limit switches.

Note that although Mach3 uses limited jogging speed, choice of direction is entirely up to you. Nothing will prevent you, in either case, from jogging farther onto the switch and possibly crashing the axis into a mechanical stop. Take great care.

4.5.4 Home Referencing in Action

When you request referencing (by button or GCode) the axis (or axes) that 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 in the *Home Off(set)* column of the **Config>Homing/Limits** dialog, can be loaded into the axis DRO as its absolute machine coordinate. If you use 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), then you

might get referencing to load something like -0.5" into the DRO. This means that the Home is half an inch clear of the limit. This wastes a bit of the axis travel but if you overshoot, when jogging to Home, you will not accidentally trip the limits. See also Software Limits as another way of solving this problem.

If you ask Mach3 to reference **before** you jog off the switch, then it will travel in the opposite direction from its "reference" direction (because the machine already on the Home switch) and stop when the machine gets 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, if the switches are shared), then the axis will move away from the actual Home point until it crashes. So the advice is **always jog carefully off the Limit switches, then reference.** If you are concerned about this problem, it is possible to configure Mach3 so it will not automatically jog off the home switch.

4.5.5 Other Home and Limit Options and Hints

4.5.5.1 Home Switch not Near Limit Switch

It is sometimes not convenient to have the Home switch at a limit of travel. Consider a large moving column floor mill or a big planer-mill. The Z travel on the column might be 8 feet and could be quite slow without affecting the overall cutting performance of the machine. If, however, the Home position is the top of the column, then referencing might involve nearly 16 feet of slow Z travel. If the reference position was chosen half way up the column, then this time can be halved. Such a machine would have a separate Home switch for the Z axis (thus requiring another input on the parallel port, but still only four inputs in a three axis machine) and would use the ability of Mach3 to set any value for an axis DRO, after referencing, to make machine-Z zero to be the top of the column.

4.5.5.2 Separate High Accuracy Home Switch

The X and Y axes on a high precision machine might have a separate Home switch to achieve the required accuracy.

4.5.5.3 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 Home reference switch connected to the reference input. A three axis machine still only needs four inputs.

4.5.5.4 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 reference only a single axis at a time — so you should remove REF All buttons from your screens – and your Home switches must all be at the end of travel on their respective axes.

4.5.5.5 Slaving

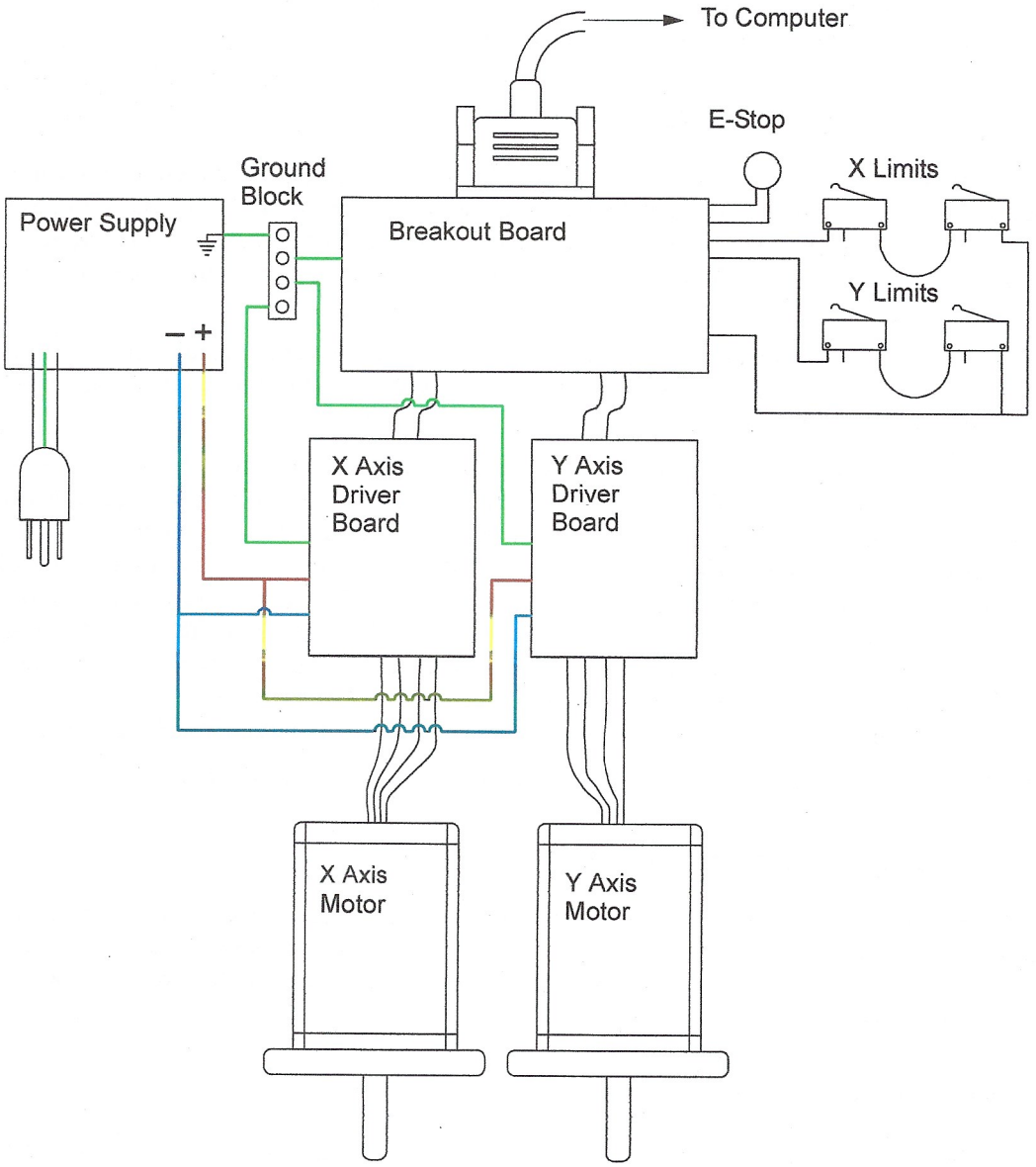
On a gantry type mill or router where the two "legs" of the gantry are driven by separate motors, then each motor should be driven by its own axis. For example, if the gantry moves in the Y direction, then axis A should be defined as a linear (i.e. non-rotational) axis and A should be slaved to Y – see Chapter 5 on Configuring Mach3 for details. Both axes should have Limit and Home switches. In normal use, both Y and A will be sent exactly the same step and direction commands by Mach3. When a Ref-

erence operation is performed, then the axes will run together until the final part of referencing, which is moving just off the Home switches. Here they will move so that each stops the same distance off its own switch. Referencing will therefore correct any racking (i.e. out of squareness) of the gantry, which might have occurred when the machine was switched off or because of lost steps.

4.5.6 Summary Wiring Diagram

The wiring diagram in Figure 4-12 gives one example of how the various pieces of a CNC system might go together. The diagram is not necessarily complete, but it shows the main components. Note the relationship of the various parts: the power supply, motor driver boards, a breakout board, the stepper motors, and limit switches. Your configuration may not necessarily duplicate what is shown here, but the diagram should give you an idea of what you are trying to accomplish. The breakout board will probably have its own separate power supply. All grounds, including any cable shields, should tie to a single point. Consult the documentation from the manufacturers for specific details.

Figure 4-12: Example Wiring Diagram



4.6 Spindle Control

There are three different ways in which Mach3 can control the machine 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.

4.6.1 On/Off Motor Control

M3 and a screen button will request that the spindle starts in a clockwise direction. M4 will request that the spindle starts in a counterclockwise direction. M5 requests that the spindle stops. M3 and M4 can be configured to activate external output signals, which can be associated with output pins on the parallel ports. 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 in reverse, it would be better to treat M3 and M4 as the same or to allow M4 to activate a signal that you do not connect to anything.

Clearly 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 main power 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 the "GCode" definition says that it is valid to issue an M4 when the spindle is running clockwise under an M3 (and vice-versa). If your spindle drive is an AC motor, instantly 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. For safety, you need to introduce time delays on the operation of the contactors or use a modern inverter drive which allows you to change direction with a running motor.

See also the note about the limited number of Relay Activation Signals in Section 4.7, *Coolant*.

4.6.2 Step and Direction Motor Control

If your spindle motor is a servo motor 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.

4.6.3 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, 25% gives 2.5 volts, 50% gives 5 volts, up to 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 4-13: A 50% Pulse Width Modulated Signal

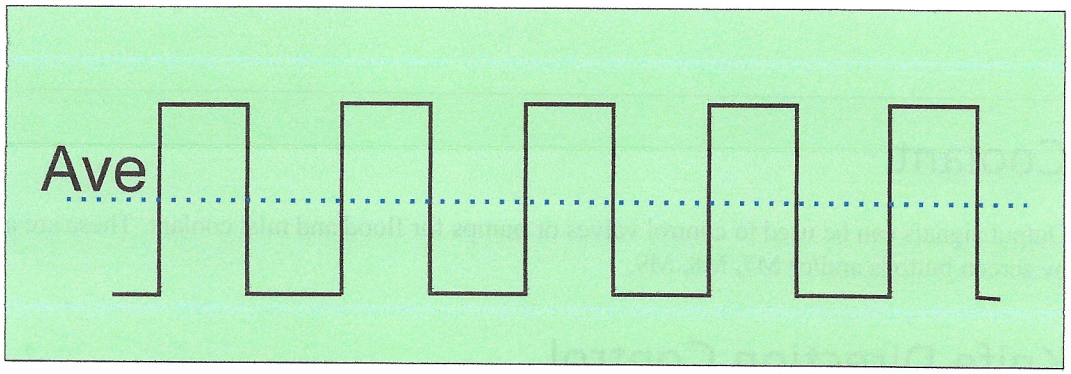


Figure 4-14: A 20% Pulse Width Modulated Signal

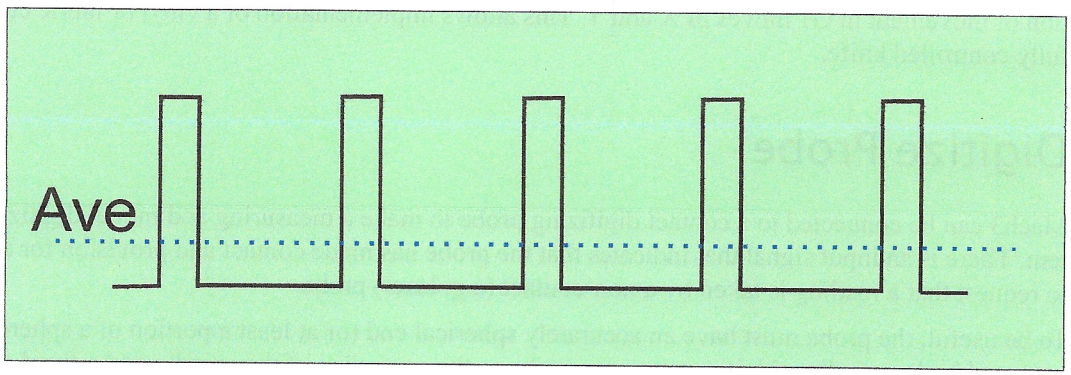


Figure 4-13 and Figure 4-14 show the pulse width at approximately 50% of the cycle and 20% of the cycle.

To turn the PWM spindle speed signal into a proportional DC voltage, the pulse signal must be transformed. In essence, a circuit must be used to find the average of the pulse width modulated signal. The circuit can be a simple capacitor and resistor, or it can be much more complex depending (a) on how linear you want the relationship between the width and the final output voltage, and (b) on the speed of response you need to the changing pulse width.

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

Refer to the manufacturer's documentation that should come with your controller. Further details may be found by using "PWM converter" or "PWM Digispeed" as a search term to Google or your favorite search engine.

Note: Many users have found that PWM and other 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. If you use such a spindle drive, ArtSoft USA strongly recommends that 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. You need to take care with the electronics, as the inputs of many cheap PWM speed controllers are not isolated from the main power supply.

4.7 Coolant

Output signals can be used to control valves or pumps for flood and mist coolant. These are activated by screen buttons and/or M7, M8, M9.

4.8 Knife Direction Control

Rotary axis A can be configured so it rotates to ensure that a tool like a knife is tangential to the direction of movement in G1 moves of X and Y. This allows implementation of a vinyl or fabric cutter with fully controlled knife.

4.9 Digitize Probe

Mach3 can be connected to a contact digitizing probe to make a measuring and model digitizing system. There is an input signal that indicates that the probe has made contact and provision for an output to request that a reading is taken by a non-contact (e.g. laser) probe.

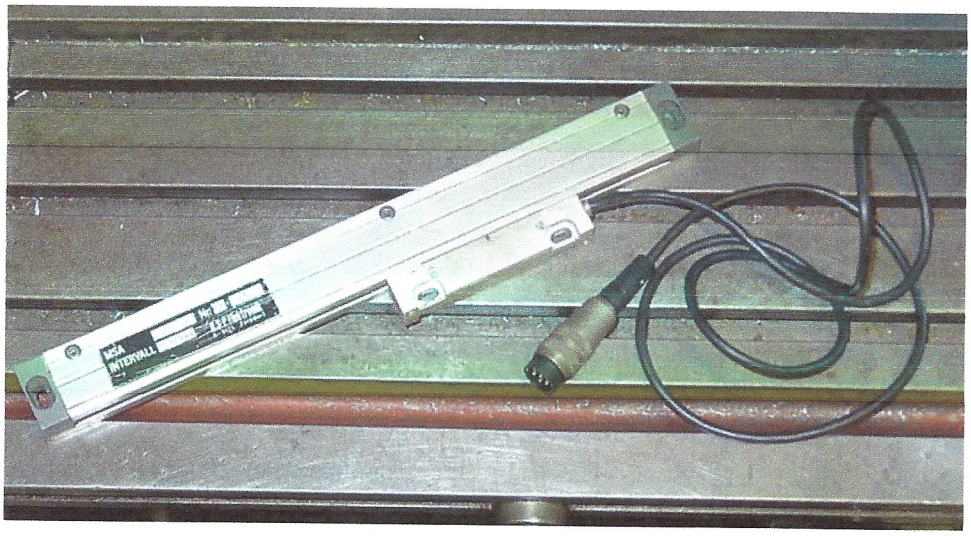
To be useful, the probe must have an accurately spherical end (or at least a portion of a sphere) mounted in the spindle with its center accurately on the centerline of the spindle and a fixed distance from a fixed point in the Z direction (e.g. the spindle nose). To be capable of probing non-metallic materials (and many models for digitization will be made in foam, MDF or plastic), the probe requires to make (or break) a switch with a minute deflection of its tip in any (XY or Z) direction). If the probe is to be used with an automatic toolchanger, then it also needs to be "cordless." These requirements are a major challenge for the designer of a probe to be built in a home workshop, and commercial probes are not cheap.

A development feature is implemented to allow the use of a laser probe.

4.10 Linear (Glass Scale) Encoders

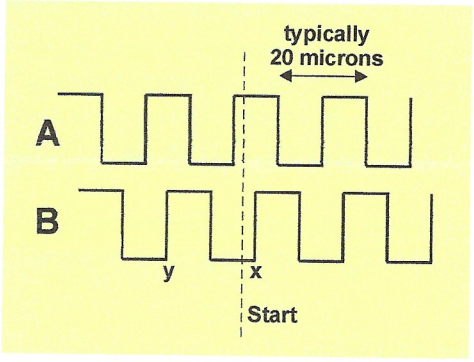
Mach3 has four pairs of inputs to each of which an encoder with quadrature outputs can be connected. Typically these might be "glass scale" encoders. Figure 4-15 shows an example. Mach3 will display the position of each of these encoders on a dedicated DRO, shown in Figure 4-17. These values can be loaded from and saved to the main axis DROs.

Figure 4-15: Glass Scale Encoder



Inside the case of the encoder is a glass (or sometimes plastic) strip ruled with lines, often 10 microns wide, separated by the same sized clear space. A light shining on a phototransistor through the ruling generates a signal of the form shown as A in Figure 4-16. One complete cycle corresponds to a movement of 20 microns.

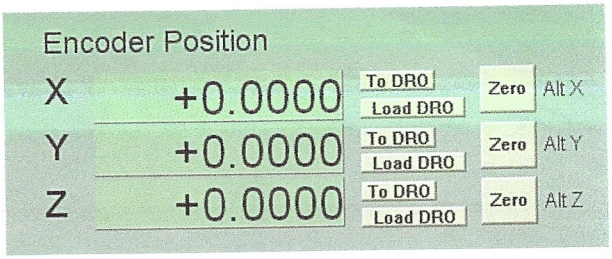
Figure 4-16: Quadrature Signals



Another light and phototransistor located 5 microns (in this example) away from the first one would generate signal B a quarter of a cycle out from A (hence the name *quadrature*)

Notice that a signal, either A or B, changes every 5 microns of movement so the resolution of the scale is 5 microns. We can tell which way it is moving by the sequence of changes. For example, if B goes from Lo to Hi when A is Hi (point *x*), then we are moving to the right of the marked Start, whereas if B goes from Hi to Lo when A is Hi (point *y*) then motion is to the left, away from Start.

Figure 4-17: Encoder DROs



Mach3 expects logic signals. Some glass scales (for example, certain Heidenhain models) give an analog sine wave. This allows clever electronics to interpolate to a higher resolution than 5 microns. If you want to use these encoders, then you need to square off the waveform with an operational amplifier/ comparator. TTL output encoders will connect directly to the input pins of the parallel port but, as noise will give false counts, they are better interfaced through what is known as a Schmitt trigger chip. The scales require a DC supply (often 5 volts) for the lights and any driver chips in them.

Notes:

- You can not easily use a linear scale as the feedback encoder for a servo drive, as the slightest backlash or springiness in the mechanical drive will make the servo unstable.
- It is not easy to connect the rotary encoders on the servo motor to the encoder DROs, although this would be attractive for manual operation of the axes with position readout. The problem is that the 0 volt (common) inside the servo drive used for the motor encoders is almost certainly not the same 0 volt as your PC or breakout board. Connecting them together will cause problems - don't be tempted to do it!
- The main benefit of using linear encoders on linear axes is that their measurements do not depend on the accuracy or backlash of the drive screw, belt, chain etc.

4.11 Spindle Index Pulse

Mach3 has an input for one or more pulses generated each revolution of the spindle. Mach3Mill can use this to display the actual speed of the spindle. In Mach3Turn, it can be used to co-ordinate the movement of the tool and work when cutting threads and for orientating the tool for the back boring canned cycle. It can be used to control feed on a per-rev rather than per-minute basis.

4.12 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. 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. This function is often implemented in commercial breakout boards.

4.13 Other Functions

Mach3 accepts fifteen OEM Trigger input signals which you can assign for your own use. For example, they can be used to simulate clicking a button or to call user-written macros.

In addition there are four user inputs which can be interrogated in user macros.

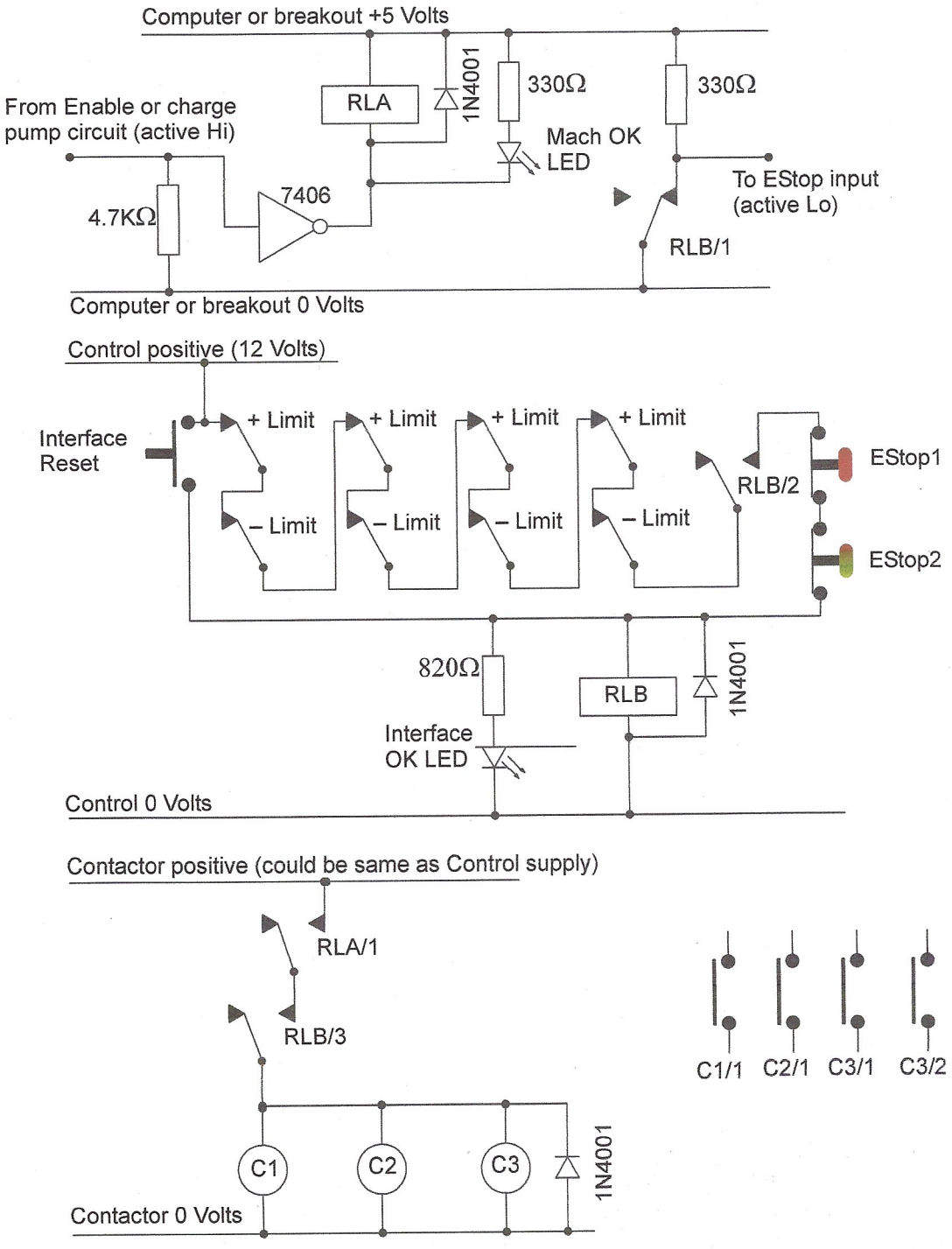
Input #1 can be used to inhibit running of the part program. It might be connected to the guards on your machine.

Full details of the architecture of Input Emulation are given in the *Mach3 Customization* wiki. The setup dialog is defined in Section 5.

The Relay Activation outputs not used for the Spindle and Coolant 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 keyboard emulator can give more inputs, these cannot be used for all functions. You may have to use a ModBus device to significantly expand custom input/output.

4.14 Sample Schematic of EStop and Limits Using Relays



Circuit Notes:

This circuit represents only one possible configuration for externally-connected limit switches. If you require reference switches, then those should be separate and connected to Mach3 inputs.

Relay contacts are shown in their de-energized position. Limit switches and pushbuttons are in their non-operated position.

Holding the *Interface Reset* button will allow the Mach3 Reset button to be pressed and the axes to be jogged off their limit switches. The *Interface Reset* button will then latch.

Relay A (RLA) needs to have one normally open (NO) contact. It must have a 5-volt coil that is at least 150 Ohms so it requires no more than 33 milliamps. Omron part number G6H-2100-5 is suitable and has contacts rated at 1 amp, 30 volts DC.

Relay B (RLB) needs to have one NC and two NO contacts. It can have any convenient coil voltage to suit an available power supply. The Common of this should, ideally, not be the PC's 0 volt rail to avoid long lengths of wiring for the limit and EStop switches, which can induce noise. The Omron MY4 series should be suitable, with four contacts rated a 5 amps, 220 volts AC.

The LEDs are optional but useful to indicate what is happening. The current limiting resistor for the Interface OK LED will need to be 1.8K ohms if a 24-volt supply is used.

If the coil voltages are suitable, then the contactors can use the "Control" positive and common supply.

The arrangement of contactors (coils shown as C1, C2, C3) depends on your drive power supply arrangements and the wiring of the motors for the machine tool. You should aim to switch the DC supply to the steppers or servos after the smoothing capacitor to ensure a prompt stop. You may want to rewire the spindle and coolant motors so that the control contactor does not trip the no-volt release circuitry (that is, you may want to switch the motor leads **after** the main machine contactors). Do not share contacts on a given contactor between the AC supply and the stepper/servo DC supply because of the greatly increased risk of a short circuit between those supplies. **Seek advice if you are unsure**, especially before working with 220/440 3-phase circuits.

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.

This is a blank left-hand page for two-sided printing.

Chapter 5 Configuring Mach3 for Your Machine and Drives

If you have bought a machine tool already interfaced to 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.

ArtSoft USA recommends that you keep a printed copy of how your version of Mach3 is configured. The information will be extremely helpful if you ever need to re-install the software from scratch. Mach3 stores this information in an `.XML` file in the `\Mach3` folder that you can view and print.[1] If you created a custom profile for your configuration, as recommended in Section 2.3, *Mach3 Profiles*, there will be an `.XML` file ("`MyMill.xml`," or whatever you named your profile) in the `\Mach3` folder containing that profile information. Although you could edit the `.XML` file, ArtSoft USA **strongly** recommends that you do not. Change the Mach3 configuration only through the program dialogs.

5.1 A Configuration Strategy

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

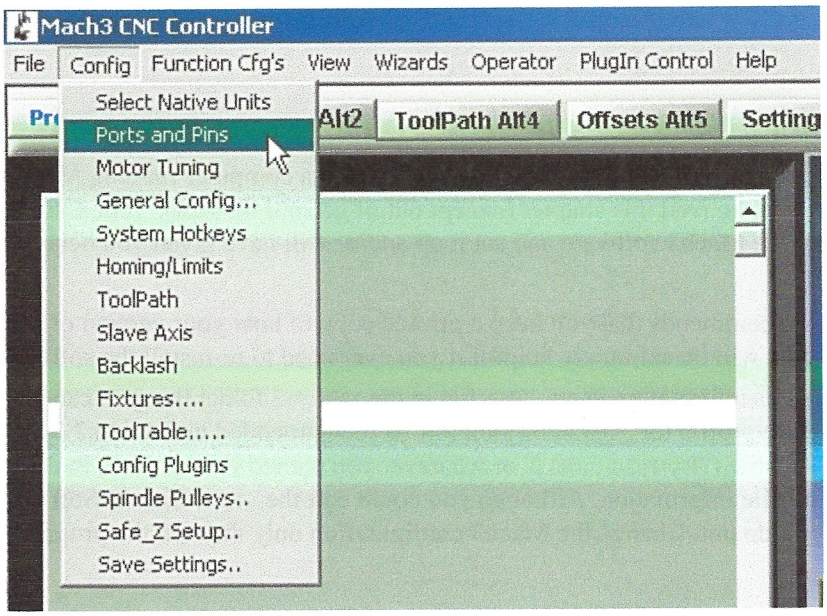
Virtually all the work you will do in this chapter is based on dialog boxes reached from the **Config**(ure) menu. These are identified in this manual by, for example, **Config>Ports and Pins,** which means that you choose the **Ports and Pins** entry from the **Config** menu.

5.2 Initial Configuration

Begin with the **Config>Ports and Pins** dialog. Figure 5-1 shows this selection on the **Config** menu. The **Ports and Pins** dialog has many tabs, but the initial one is as shown in Figure 5-2.

1. The default configuration files are `Mach3Mill.xml`, `Mach3Turn.xml`, and `Plasma.xml`.

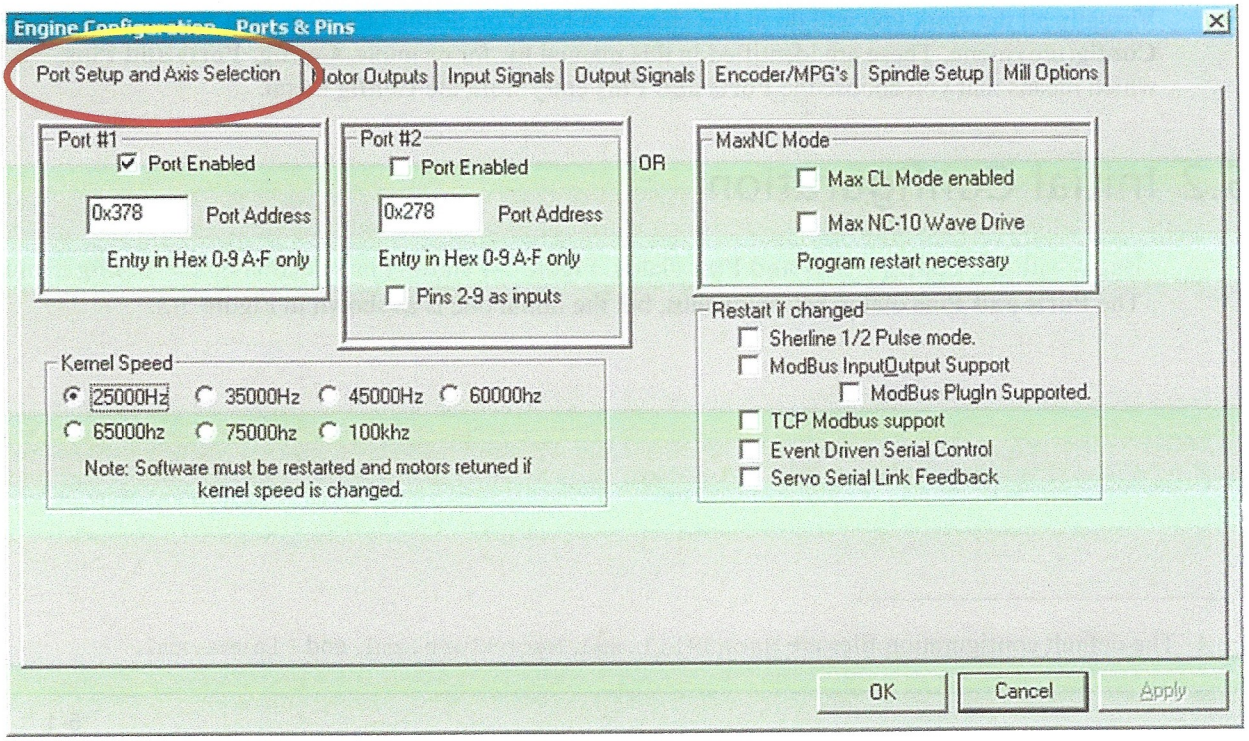
Figure 5-1: Selecting the Ports and Pins Dialog from the Config Menu



5.2.1 Defining Addresses of Port(s) to Use

Select the **Port Setup and Axis Selection** tab on the **Ports and Pins** dialog, as shown in Figure 5-2.

Figure 5-2: Port Setup and Axis Selection Tab on the Ports and Pins Dialog



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

If you are using one or more PCI add-on parallel port cards, then you will need to determine the address to which each responds. There are no standards! Open the Windows Control Panel from 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. Choose the **Resources** tab. The first number in the first IO range line is the address to use. Note the value and close the **Properties** dialog.

Note: 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 steps for it.

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

Enter the address that you just looked up as the first port's address on the **Port Setup and Axis Selection** dialog. Do not provide the 0x prefix to say it is Hexadecimal, as Mach3 assumes this. If necessary, check *Enabled* for port 2 and enter its address in the same way.

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 click *Apply*.**

5.2.2 Choosing Kernel Speed

The Mach3 driver can run at frequencies from 25,000 Hz (pulses per second) up to 100,000 Hz, depending on the speed of your processor and other loads placed on it when running Mach3.

The frequency you need depends on the maximum pulse rate you need to drive any axis at its top speed. 25,000 Hz will probably be suitable for stepper motor systems. With a 10 micro-step driver such as a Gecko 201, you will get around 750 RPM from a standard 1.8 degree stepper motor with a 25,000 Hz pulse rate. Higher pulse rates are needed to achieve desired motor RPM for servo drives that have high resolution shaft encoders on the motor. Further details are given in the section on motor tuning and in Section 4.4.2, *Determining Axis Drive Requirements*.

Computers with a 1 GHz clock speed will almost certainly be able to run at 35,000 Hz, so you can choose this if you need a high step rate (for example, if you have very fine pitch lead screws).

The demonstration version of Mach3 will run at 25,000 Hz **only**. In addition, if Mach3 is forcibly closed, then on re-start it will automatically revert to 25,000 Hz operation. The actual frequency of the running system is displayed on the standard **Diagnostics** screen.

Click the box next to the desired kernel speed.

Don't forget to click the *Apply* button before proceeding.

5.2.3 Defining Special Features

The **Port Setup and Axis Selection** dialog includes check boxes for a variety of special configurations. They should be self-explanatory if you have the relevant hardware in your system. If not, then leave then unchecked.

Don't forget to click the *Apply* button before proceeding.

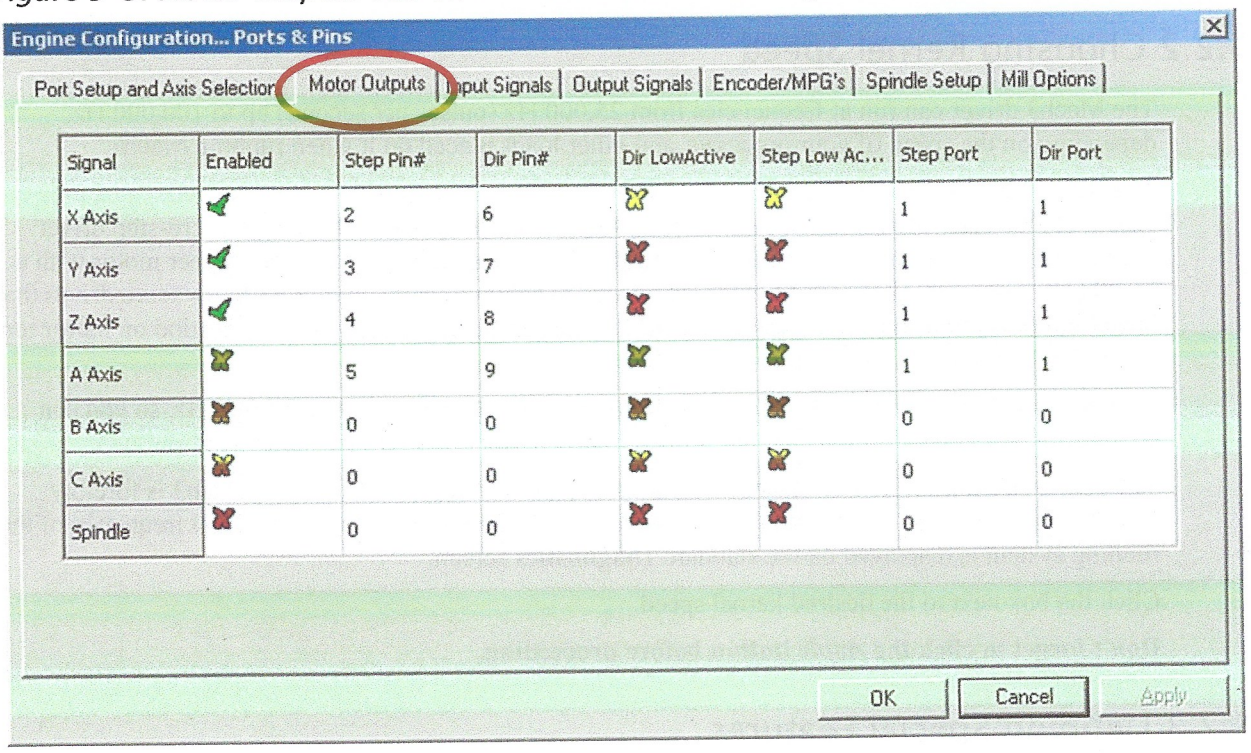
5.3 Defining Input and Output Signals To Use

Now that you have established the basic configuration, it is time to define which input and output signals you will use and which parallel port and pin will be used for each signal. 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.

5.3.1 Axis and Spindle Output Signals to Be Used

View the **Motor Outputs** tab of the **Ports and Pins** dialog. This will look similar to Figure 5-3.

Figure 5-3: Motor Outputs Tab on the Ports and Pins Dialog



Define where the drives for your X, Y and Z axes are connected, and click in the **Enabled** column to get a check-mark to Enable these axes. If any axis is enabled that shouldn't be, click in the **Enabled** column to change the green check to a red X.

If you need to edit any boxes in the **Step Pin #**, **Dir Pin #**, **Step Port**, or **Dir Port** columns, double-click the appropriate boxes and edit the entries.

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

If you have rotary or slaved axes, then you should enable and configure them.

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.

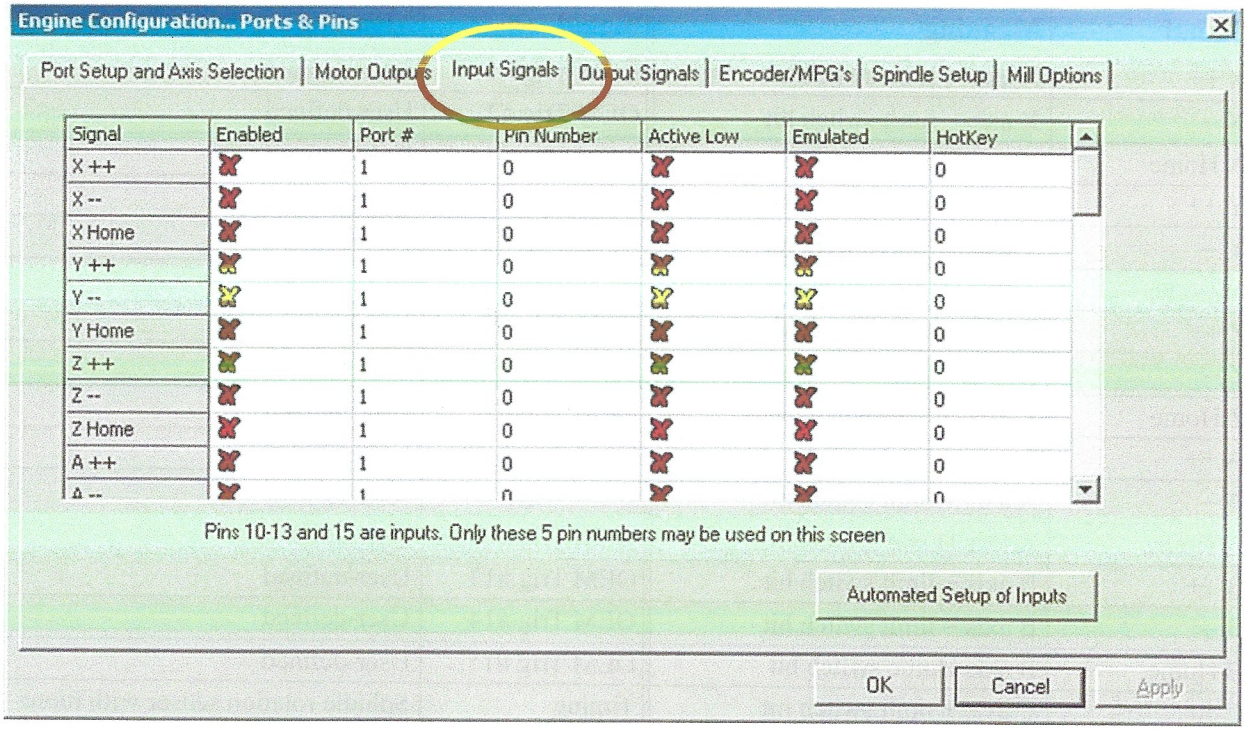
Enable the spindle if your spindle speed will be controlled by Mach3. Allocate a Step pin/port for it if using pulse width modulated control with relays to control its direction, or allocate Step and Direction

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.**

5.3.2 Input Signals To Be Used

Now select the **Input Signals** tab. This will look like Figure 5-4.

Figure 5-4: Input Signals Tab on the Ports and Pins Dialog



Notice the scroll bar at the right side of the window. Table 5-1 gives the complete list of input signals.

Table 5-1: List of Possible Input Signals

Signal	Meaning	Signal	Meaning
X ++	X axis + limit switch hit	OEM Trig #1	User-defined
X --	X axis – limit switch hit	OEM Trig #2	User-defined
X Home	X axis Home switch hit	OEM Trig #3	User-defined
Y ++	Y axis + limit switch hit	OEM Trig #4	User-defined
Y --	Y axis – limit switch hit	OEM Trig #5	User-defined
Y Home	Z axis Home switch hit	OEM Trig #6	User-defined
Z ++	Z axis + limit switch hit	OEM Trig #7	User-defined
Z --	Z axis – limit switch hit	OEM Trig #8	User-defined
Z Home	Z axis Home switch hit	OEM Trig #9	User-defined
A ++	A axis + limit switch hit	OEM Trig #10	User-defined
A --	A axis – limit switch hit	OEM Trig #11	User-defined
A Home	A axis Home switch hit	OEM Trig #12	User-defined
B ++	B axis + limit switch hit	OEM Trig #13	User-defined
B --	B axis – limit switch hit	OEM Trig #14	User-defined
B Home	B axis Home switch hit	OEM Trig #15	User-defined
C ++	C axis + limit switch hit	Timing	Spindle rotation sensor with more than one slot or mark
C --	C axis – limit switch hit	Jog X ++	Move X in + direction
C Home	C axis Home switch hit	Jog X --	Move X in – direction
Input #1	Safety guards not in place, or user-defined	Jog Y ++	Move Y in + direction
Input #2	User-defined	Jog Y --	Move Y in – direction
Input #3	User-defined	Jog Z ++	Move Z in + direction
Input #4	Single step, or user-defined	Jog Z --	Move Z in – direction
Probe	Digitizing probe enable	Jog A ++	Move A in + direction
Index	Spindle rotation sensor with one slot or mark	Jog A --	Move A in – direction
Limit Ovrd	Limit override enable		
EStop	Emergency Stop button pressed		
THC On	Plasma torch control		
THC Up	Plasma torch control		
THC Down	Plasma torch control		

This discussion assumes you have chosen one of the home/limit switch configurations described in Section 4.5, *Limit and Home Switches*.

- If you implemented configuration one from Section 4.5, with the limit switches connected together to trigger an EStop or disable the axis drives through the drive electronics, then you do not need to select any of the Limit inputs (X++, X--, etc.).

- If you implemented configuration two from Section 4.5, you will probably have Home switches on the X, Y, 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 as one signal, then you should enable Limit --, Limit ++, and Home for each axis and allocate the same pin to Home, Limit --, and Limit ++.

If present, also configure axes A, B, and C.

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. 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.

Enable Probe for digitizing.

Enable THC On, THC Up, and THC Down for control of a Plasma torch.

Enable and define Jog inputs if you have separate buttons for Jog control. (The keyboard arrow keys provide Jog control in any case.)

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 also want to have some inputs for glass scales or other encoders. You may have to compromise by not having things like a physical Limit Override switch to save signals.

You can also consider using a Keyboard Emulator for some input signals. Response time will be slower than with a parallel port. Refer to Section 5.3.3.

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

5.3.3 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.

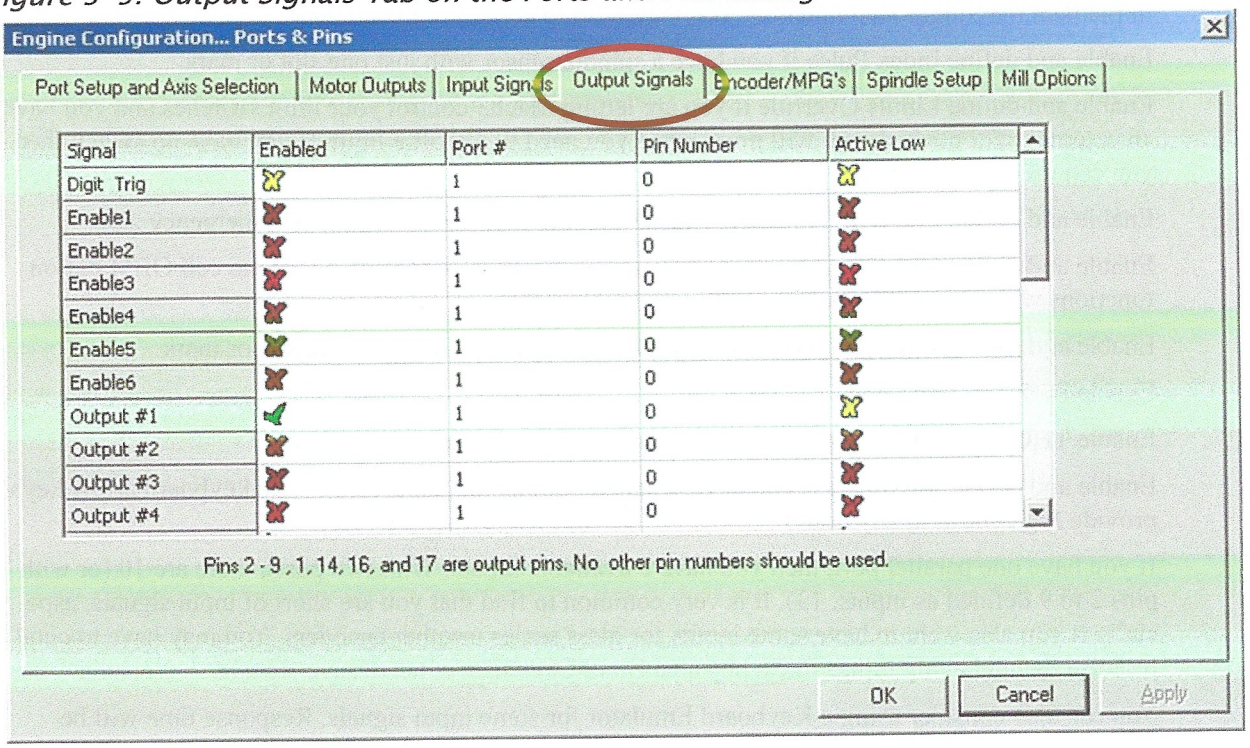
The key-up and key-down signals usually come from a keyboard emulator (like the Ultimarc IPAC or Hagstrom) that is triggered by switches connected to its inputs. This allows more switches to be sensed than spare pins on your parallel ports, but there may be significant time delays before the switch change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

5.3.4 Output Signals

Use the **Output Signals** tab to define the outputs you require. See Figure 5-5.

Figure 5-5: Output Signals Tab on the Ports and Pins Dialog



Notice the scroll bar on the right side of the window. Table 5-2 gives a complete list of output signals.

Table 5-2: List of Possible Output Signals

Signal	Meaning	Signal	Meaning
Digit Trig		Current Hi/Lo	
Enable1		Output #7	
Enable2		Output #8	
Enable3		Output #9	
Enable4		Output #10	
Enable5		Output #11	
Enable6		Output #12	
Output #1		Output #13	
Output #2		Output #14	
Output #3		Output #15	
Output #4		Output #16	
Output #5		Output #17	
Output #6		Output #18	
Charge Pump		Output #19	
Charge Pump2		Output #20	

You will probably want to use only one *Enable* output (as all the axis drives can be connected to it). If you are using the charge pump/pulse monitor feature, then you can 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.

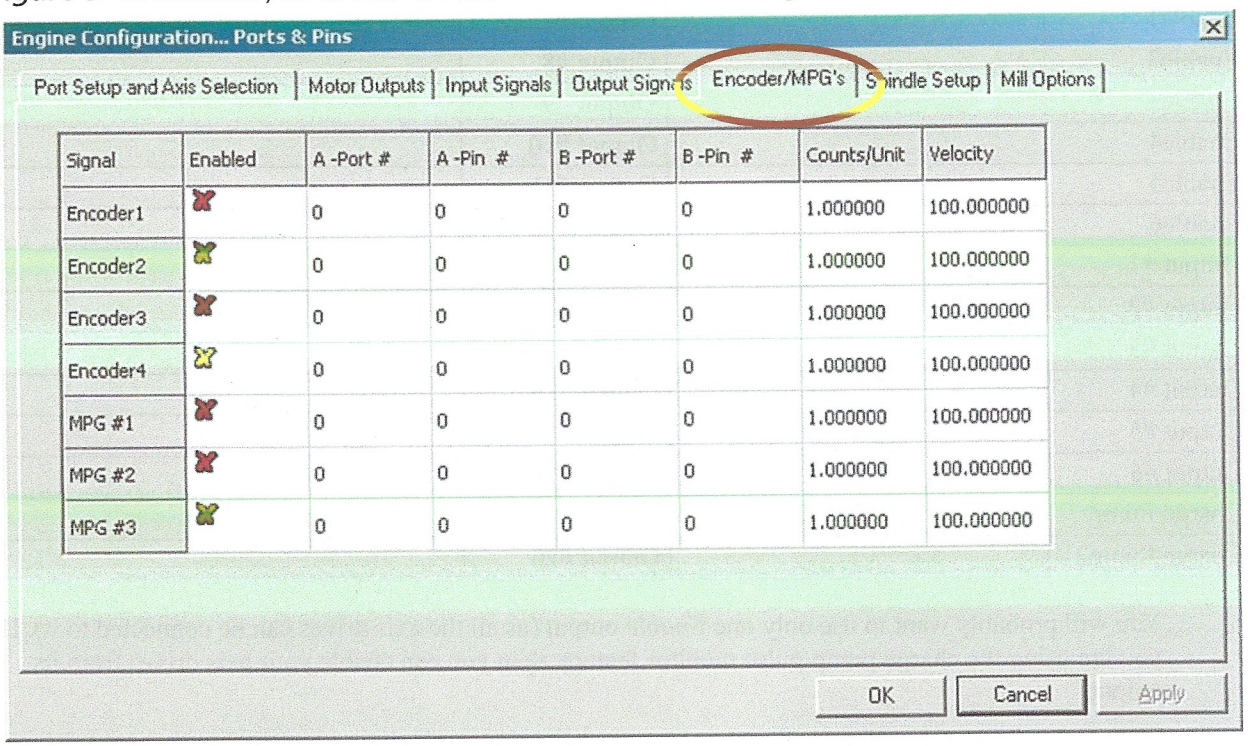
The *Charge Pump* line should be enabled and defined if your breakout board accepts this pulse input to continually verify correct operation of Mach3. *Charge Pump2* is used if you have a second breakout board connected to the second port or want to verify the operation of the second port itself.

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

5.3.5 Defining Encoder and Manual Pulse Generator (MPG) Inputs

Use the **Encoder/MPGs** tab to define the connections and the resolution of linear encoders or Manual Pulse Generators (MPGs) used for jogging the axes. See Figure 5-6.

Figure 5-6: Encoder/MPG Tab on the Ports and Pins Dialog



This dialog does not need an active-lo selection. If the encoders count the wrong way, just swap the pins allocated for the A and B inputs.

5.3.5.1 Setting Up Encoders

The *Counts per unit* value should be set to correspond to the resolution of the encoder and your selection of Native units. For example, a linear scale with rulings at 20 microns produces a count every 5 microns (remember the quadrature signal), or 200 counts per unit (millimeter). If you have Native units set to inches, then it would be 200 x 25.4 = 5080 counts per unit (inch) because there are 25.4 millimeters per inch. The *Velocity* value is not used.

5.3.5.2 Setting Up MPGs

The *Counts per unit* value defines the number of quadrature counts that need to be generated for Mach3 to sense movement of the MPG. For a 100 counts-per-revolution (CPR) encoder, a value of 2 should be suitable. For higher resolutions, you will probably want to increase this value to get the mechanical sensitivity you want. A value of 100 works well with 1024 CPR encoders.

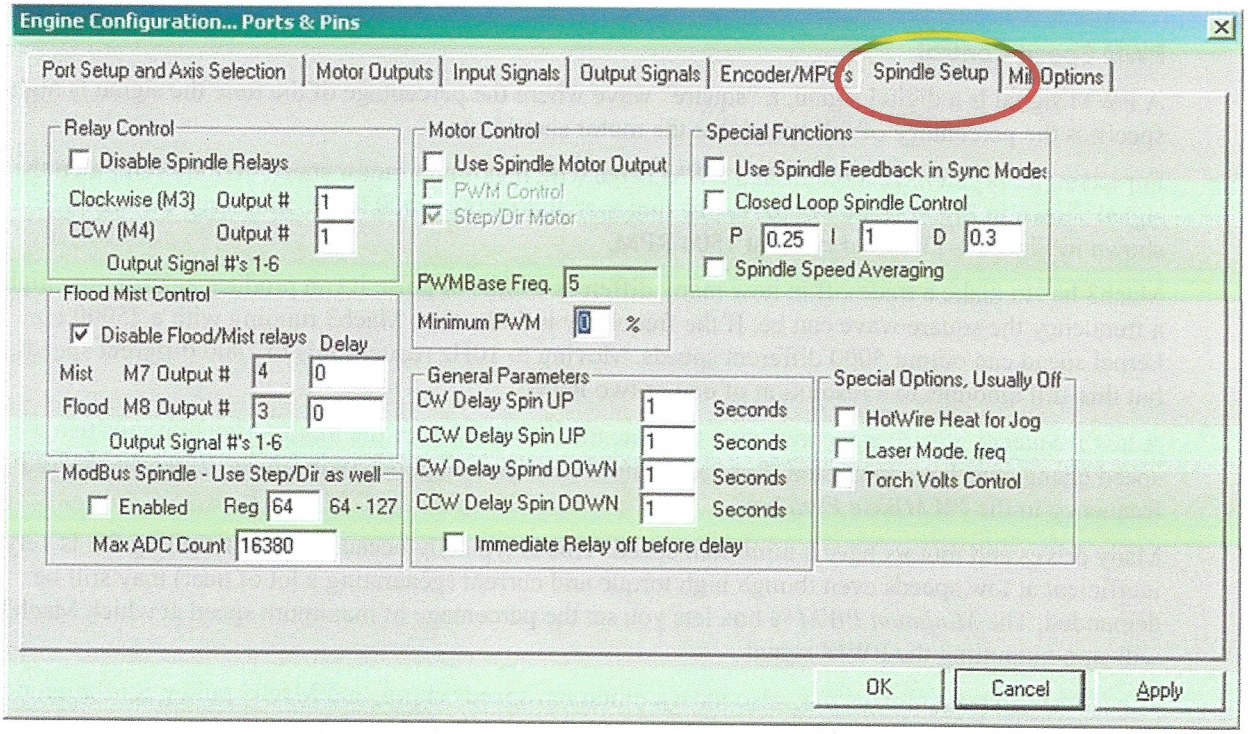
The *Velocity* value determines the scaling of pulses sent to the axis being controlled by the MPG. The lower the value given in *Velocity,* the faster the axis will move. Its value is best set by experiment to give a reasonable axis movement speed when spinning the MPG as fast as is comfortable.

5.3.6 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 only 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 5-7. Refer to Section 5.5.5, *Spindle Motor Speed Control Setup*, for a description.

Figure 5-7: Spindle Setup Tab of the Ports and Pins Dialog



5.3.6.1 Coolant Control

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

5.3.6.2 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 M3, M4 and M5) by using two outputs that you specify here. The Port/Pins for those 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. M3, M4, and M5 will control the pulse train generated automatically.

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

5.3.6.3 Motor Control

Check *Use Spindle Motor Output* 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*. Checking (or clearing) this box automatically checks (or clears) the Spindle checkbox on the Motor Outputs tab.

PWM Speed 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 full speed that the motor should run.

For example, if you have a motor and PWM drive with maximum motor speed of 3000 rpm, then the signal shown in Figure 4-14 would run the motor at 3000 x 0.2 = 600 RPM. Similarly, the signal shown in Figure 4-13 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, then 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 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. Enter the desired frequency in the *PWMBase Freq* box.

Many drives and motors have a minimum speed. This is typically because a motor cooling fan is very inefficient at low speeds even though high torque and current (generating a lot of heat) may still be demanded. The *Minimum PWM %* box lets you set the percentage of maximum speed at which Mach3 will stop outputting the PWM signal.

The PWM drive electronics may also have a minimum speed setting, and Mach3 pulley configuration (see Section 5.5.5.1) 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 can be a variable-speed drive controlled by step pulses, or a full servo drive.

You can use the Mach3 pulley configuration (see Section 5.5.5.1) to define a minimum speed if this is needed by the motor or its electronics.

5.3.6.4 Modbus Spindle Control

This block allows the setup of an analog port on a Modbus device (e.g. a Homann ModIO) to control spindle speed. For details see the documentation of your ModBus device.

5.3.6.5 General Parameters

These let you specify a delay after starting or stopping the spindle before Mach3 will execute further commands (that is, a Dwell time). 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 spindle rotation. Enter the dwell times in seconds.

Immediate Relay off before delay, if checked, will switch the spindle relay off as soon as an M5 command is executed. If unchecked, the spindle relay will stay on until the spin-down delay period has elapsed.

5.3.6.6 Pulley Ratios

Mach3 has control over the speed of your spindle motor. You program spindle speeds through the S word. The Mach3 pulley system lets you define the relationship between these for 15 different pulley or gearbox settings. It is easier to understand how this works after going through the process of tuning the spindle motor, so pulley ratio setup is described in Section 5.5.5.1.

5.3.6.7 Special Function

Laser mode should always be unchecked except for controlling the power of a cutting laser by the feed rate.

Use Spindle feedback in sync mode should be unchecked.

Closed Loop Spindle Control, when checked, implements a software servo loop that 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 critical, so you are not likely to need to use this feature.

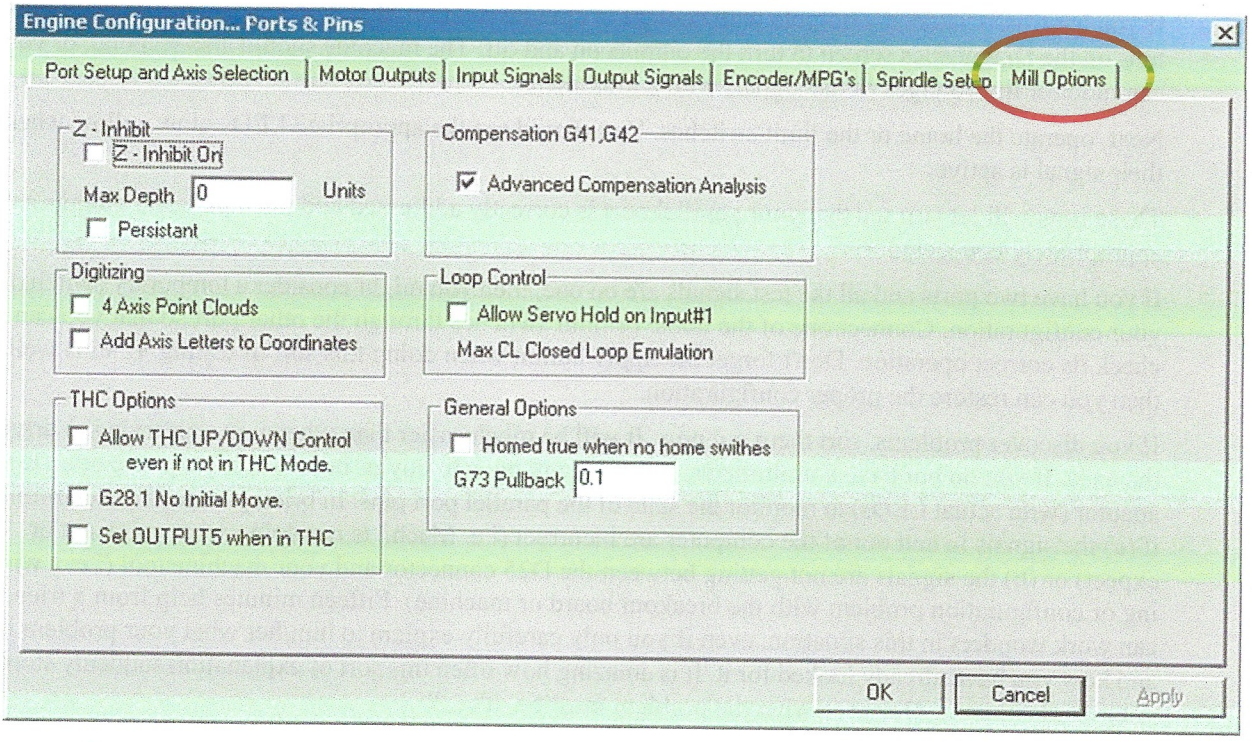
If you do use it, then the P, I and D variables should be set in the range 0 to 1. P controls the gain of the loop and an excessive value will make the speed oscillate, or hunt, around the requested value rather than settling on it. The D variable applies damping to stabilize these oscillations by using 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 **Function Cfg's>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 may find it useful with a very low inertia spindle drive, or one where the control tends to give short-term variations of speed.

5.3.7 Mill Options Tab

The final tab on **Config>Ports & Pins** is **Mill Options**. See Figure 5-8.

Figure 5-8: Mill Options Tab of the Ports and Pins Dialog



Z-inhibit. The *Z-inhibit On* checkbox enables this function. *Max Depth* gives the lowest Z value to which the axis will move. The Persistent checkbox remembers the state (which can be changed by a screen toggle) from run to run of Mach3.

Digitizing: The *4 Axis Point Clouds* checkbox enables recording of the state of the A axis as well as X, Y, and Z. The Add Axis Letters to Coordinates prefixes the data with the axis name in the point cloud file.

THC Options: Additional options for plasma torch control.

Compensation G41,G42: The Advanced Compensation Analysis checkbox turns on a more thorough look-ahead analysis that will reduce the risk of gouging when compensating for cutter diameter (using G41 and G42) on complex shapes.

Homed true when no Home switches: This option will make the system appear to be Home referenced (i.e. LEDs green) at all times. It should be used only if no Home switches are defined under **Ports & Pins>Input Signals** tab.

5.3.8 Initial Testing

The software is now configured sufficiently for you to do some simple tests with the hardware. If it is convenient to connect the inputs from the manual switches such as *Home,* then do so now.

Run Mach3Mill and display the Diagnostics screen. This has a bank of LEDs displaying the logic level of the inputs and outputs. Ensure that the external Emergency Stop signal is not active (Red *Emergency* LED not flashing), and click the red *Reset* button on the screen. Its LED should stop flashing.

If you have associated any outputs with coolant or spindle rotation, then you can use the relevant buttons on the Diagnostics screen to turn the outputs on and off. The machine should also respond, or you can monitor the voltages of the signals with a multimeter.

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

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

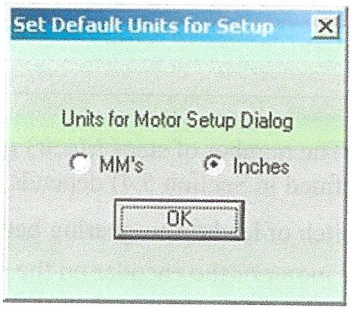
If you have two ports and all the test signals are on one, then you might consider a temporary switch of your configuration. Connect one of the home or limit switches through the other port so that you can check its correct operation. Don't forget the *Apply* button when doing this sort of testing. If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

If you discover problems, sort them out now. It 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) to monitor the state of the parallel port pins. In brief, you need to determine 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). Fifteen 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. It is amazing how often this sort of explanation suddenly stops with words like "... Oh! I see what the problem must be, it's ..."

5.4 Defining the Setup Units

With the basic functions working, it's time to configure the axis drives. The first thing to decide is whether you want to define their properties in Metric (millimeters) or Inch units. This is done using the **Config>Select Native Units** dialog. This is shown in Figure 5-9.

Figure 5-9: Select Native Units Dialog



The computations for axis configuration will be slightly easier if you choose the same system as your drive train (for example, the ballscrew pitch) was made in. A screw with 0.2" lead (5 tpi) is easier to configure in inches than in millimeters. Similarly, a 2mm lead screw will be easier to configure in millimeters. Otherwise, you will need to multiply or divide by 25.4 to convert from one units system to the other when doing the axis configurations. The multiplication or division by 25.4 is not difficult, but it is just something else to think about. You will be able to run part programs using either units, whichever option you choose here.

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

So the choice is yours. Use **Config>Select Native Units** to choose Millimeters or Inches (see Figure 5-9). Once you have made a choice, you cannot change it without going back over all the following steps or total confusion will reign! A message box reminds you of this when you use **Config>Select Native Units**.

5.5 Tuning Motors

After all the preliminaries, you can finally start setting up the motors. This section describes setting up your axis drives and, if its speed will be controlled by Mach3, the spindle drive.

Setup for each axis requires three steps:

1. Calculate how many step pulses must be sent to the drive for each unit (inch or mm) of movement of the tool or table. The choice you made in Section 5.4 determines which units will be used.
2. Establish the maximum speed for the motor.
3. Set the required acceleration/deceleration rate.

ArtSoft USA advises you to deal with one axis at a time. You may want to try running each motor before it is mechanically connected to the machine tool.

Connect the power to 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!

5.5.1 Calculating the Steps Per Unit

Mach3 can automatically perform a test move on an axis and calculate the steps per unit, as described in Section 5.5.2.3, *Automatic Setting of Steps per Unit*, but that is probably best left for fine tuning. This section gives the overall theory and tells you how to calculate the valule.

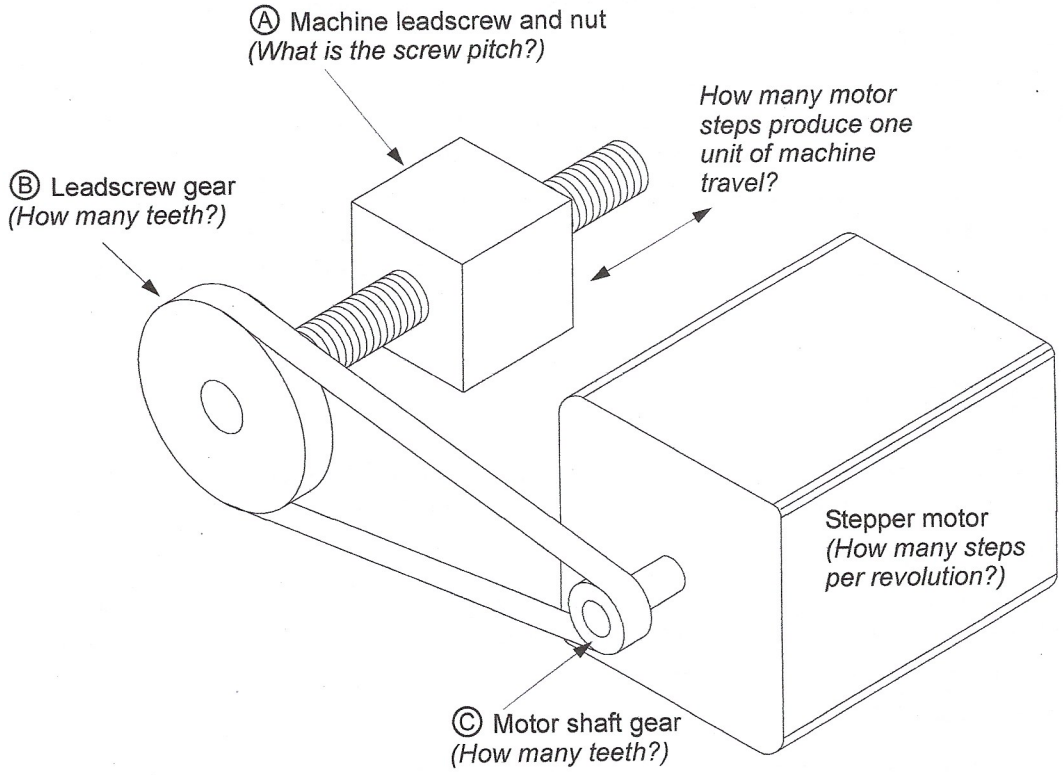
Mach3 rotates a motor by steps. The number of steps Mach3 must send to a motor to cause one "unit" of movement (inch or mm, as defined in Section 5.4) depends on:

1. The mechanical drive (e.g. pitch of leadscrew, gearing between the motor and the screw).
2. The properties of the stepper motor or the encoder on the servo motor.
3. The micro-stepping or electronic gearing in the drive electronics.

We will look at these three points in turn, then bring them together.

Figure 5-10 shows components of a typical drivetrain.

Figure 5-10: Machine Drivetrain Components



5.5.1.1 Calculating Mechanical Drive

You need to calculate the number of revolutions of the motor shaft (***motor revs per unit***) required to move the axis by one unit (one inch or one millimeter). This number will probably be greater than one for inches and less than one for millimeters, but that makes no difference to the calculation.

For a **screw and nut** Ⓐ, you need to know the pitch of the screw (that is, the thread crest-to-crest distance) and the number of starts. Inch screws are usually specified in threads per inch (tpi). The pitch is

```
P = 1 ÷ tpi
```

For example, the pitch of an 8 tpi single-start screw is `1 ÷ 8 = 0.125"`. The pitch of a 16 tpi single-start screw is

```
P = 1 ÷ 16 = 0.0625"
```

Metric screws are usually specified in terms of pitch, so no conversion calculation is necessary. You might have, for example, a 2 mm pitch screw.

If the screw is a multiple start thread, 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.** For example, a 16 tpi two-start thread has an effective screw pitch of:

`0.0625" x 2 = 0.125"`

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

`(screw revs per unit) = 1 ÷ (effective screw pitch)`

For example, for an 8 tpi single-start screw:

`(screw revs per unit) = 1 ÷ 0.125 = 8 revs/inch`

For a 2 mm pitch screw:

`(screw revs per unit) = 1 ÷ 2 = 0.5 revs/mm`

If the screw is directly driven from the motor (1:1 drive ratio), then this is also 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 single-start screw is connected to the motor with a toothed belt, with a 48 tooth pulley Ⓑ on the screw and a 16 tooth pulley Ⓒ on the motor. Then the ***motor revs per unit*** would be:

8 x 48 ÷ 16 = 24 motor revs produce 1 inch of machine travel.

(**Hint**: keep all the figures on your calculator at each stage of calculation to minimize rounding errors.)

As a metric example, suppose a two-start screw has 5 millimeters between thread crests (i.e. effective pitch is 10 millimeters) 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 ÷ 24 = 0.2

The calculation is similar for a **rack and pinion** or **toothed belt** or **chain** drive.

Find the pitch of the belt teeth or chain links. Belts are available in metric and imperial pitches with 5 or 8 millimeters common metric pitches and 0.375" ($^3/_8$") common for inch belts and for chain. For a rack, find its tooth pitch. This is best done by measuring the total distance spanning 50 or even 100 gaps between teeth. Note that, because standard gears are made to a diametral pitch, the length will not be a rational number as it includes the constant π (pi = 3.14159...).

If you know the diametral pitch *dp* of the rack, you can calculate the tooth pitch *tp* as tp = π / dp.

If the number of teeth on the pinion/sprocket/pulley on the primary shaft which drives the rack/belt/chain is N_s then:

shaft revs per unit = 1 ÷ (*tooth pitch* x N_s)

For example, with a $^3/_8$" chain and a 13 tooth sprocket on the motor shaft, then the *motor revs per unit* = 1 ÷ (0.375 x 13) = 0.2051282. In passing, we observe that this is quite "high geared" and the motor might need an additional reduction gearbox to meet the torque requirements. In this case, you multiply the motor revs per unit by the reduction ratio of the gearbox.

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

For example, a 10:1 box would give 2.051282 revs per inch.

For **rotary axes** (e.g. rotary tables or dividing heads), the unit is the **degree**. You need to calculate the number of motor revs required to produce 1 degree of axis rotation, based on the worm gear ratio of the rotary table or dividing head. This is often 90:1 for rotary tables and 40:1 for dividing heads (but check yours!) With a direct motor drive to a 90:1 worm, one motor rev would rotate the spindle 4 degrees, so *Motor revs per unit* would be 0.25. A reduction of 2:1 from motor to worm would give 0.5 revs per unit.

5.5.1.2 Calculating Motor Steps Per Revolution

The basic resolution of all modern stepper motors is 200 steps per revolution (that is, 1.8° per step). Some older steppers are 180 steps per rev, but you are not likely to meet them if you are buying supported new or nearly new equipment. Check yours though, to be sure what you have.

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 two quadrature signals, the effective resolution will be **four** time this value. You will typically see a CPR in the range of about 125 to 2000, corresponding to 500 to 8000 steps per revolution.

5.5.1.3 Calculating Mach3 Steps Per Motor Revolution

ArtSoft USA very strongly recommends that you use micro-stepping drive electronics for stepper motors. If you do not, 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. Micro-stepping will give smoother operation.

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. A value of 10 micro-steps means that Mach3 will need to send 2000 pulses per motor shaft revolution for a stepper axis drive (assuming a stepper motor with a basic resolution of 200 steps/rev).

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, then 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.

5.5.1.4 Mach3 Steps Per Unit

Now that we know the required motor revs per unit of travel, we can finally calculate:

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

Figure 5-11 shows the dialog for **Config>Motor Tuning**. Click a button at the right of the dialog to select the axis that you are configuring, and enter the calculated value of *Mach3 steps per unit* in the box labeled *Steps per*. This value does not have to be an integer, so you can achieve as much accuracy as you wish. Whatever the number, ***it is a specific, calculated number determined by the drive configuration***. It is **NOT** a "tuneable" quantity. If you don't get the correct machine travel when testing, you have made an error in your calculation. Be sure to calculate and set a value for each axis you are using, as they may not be the same.

To avoid forgetting later, click *Save Axis Settings* now.

Figure 5-11: Motor Tuning Dialog



5.5.2 Setting the Maximum Motor Speed

Now set the maximum motor speed. The **Config>Motor Tuning** dialog gives a graphical display showing of velocity against time for a short imaginary move using the currently selected parameters. The axis accelerates, perhaps runs at full speed, then decelerates. You can click and drag the *Accel* and *Velocity* sliders to see how changes affect the performance. Set the velocity 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 minute. *Accel* is in units per second^2. The acceleration value is also given in Gs to give you a subjective impression of the forces that will be applied to a massive table or workpiece.

The maximum velocity you can display is limited by the maximum pulse rate of Mach3. For example, if you have configured this to 25,000 Hz and 2000 steps per unit then, the maximum possible *Velocity* is 750 units per minute.

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. Section 5.5.2.1 tells how to do trials. Section 5.5.2.2 tells how to do the calculations.

5.5.2.1 Practical Trials of Motor Speed

You saved the axis after setting the *Steps per unit*. If you didn't, do so now. Click **OK** to close the dialog, and make sure that everything is powered up. Click the **Reset** button so its LED glows continuously.

Go back to **Config>Motor Tuning** and select the axis to test. Use the *Velocity* slider to set the graph to about 20% of maximum velocity. Press and hold the cursor **Up** key on your keyboard. The machine axis should move in the Plus direction. If it moves faster than seems desirable, then choose a lower velocity. If it moves too slowly, then choose a higher velocity. The cursor **Down** key will make the axis move the other way (i.e. the Minus direction).

If the axis moves in the wrong direction (Minus when it should move Plus), there are three ways to fix the problem. Click the *Save* button to save the axis settings, then do **one** of the following:

- Change the *Low Active* setting for the *Dir* pin of the axis in **Config>Ports and Pins>Motor Outputs** tab (and *Apply* it).
- Check the *Reversed* box in **Config>Homing/Limits** for the axis that you are using.
- Switch off the power, then reverse **one** pair of physical connections to the motor from the drive electronics.

If a stepper motor hums or screams, then you have wired it incorrectly or are trying to drive it much too fast. Reduce the acceleration and velocity. We have seen small systems that had a maximum velocity of only 4 or 5 inches per minute. If that doesn't help, check the wiring. 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 are having any troubles here, you will be pleased if you followed the advice to buy current and properly supported products – buy right, buy once!

Pulse width is another consideration. Most drives will work well with a 1 microsecond minimum pulse width. 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 **Ports and Pins>Motor Outputs** tab), then you might try increasing the pulse width to, say, 5 microseconds. The Step and Direction interface is very simple but, because it can still "sort of work" when configured badly, it can be difficult to fault-find without being very systematic and/or looking at the pulses with an oscilloscope.

5.5.2.2 Motor Maximum Speed Calculations

The trials approach described in Section 5.5.2.1 is likely to be the easiest way to set up motor speed, but if you want to calculate the maximum motor speed this section outlines what you need to consider.

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

1. Maximum allowed speed of motor (perhaps 4000 rpm for servo or 1000 rpm for stepper).
2. Maximum allowed speed of the ballscrew (depends on length, diameter, how its ends are supported).
3. Maximum speed of belt drive or reduction gearbox.
4. Maximum speed which drive electronics will support without signalling a fault.
5. 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 Mach1/Mach2 Yahoo! online forum is a useful place to get advice from other Mach3 users, worldwide, on this sort of topic. See www.machsupport.com for a link.

5.5.2.3 Automatic Setting of Steps per Unit

Section 5.5.1, *Calculating the Steps Per Unit*, explains how to calculate the steps per unit, but you may not be able to measure the gearing of your axis drive or know the exact pitch of a screw. Provided you can accurately measure the distance moved by an axis, however, perhaps using a dial test indicator and gage blocks, then Mach3 can calculate the steps per unit that should be configured. For best results, you should get the value approximately correct by calculation, even if you have to guess at some values, before performing the automatic setup.

Select the **Settings Alt6** tab in the main Mach3 CNC Controller window, as shown in Figure 5-12.

Figure 5-12: Settings Alt6 Tab

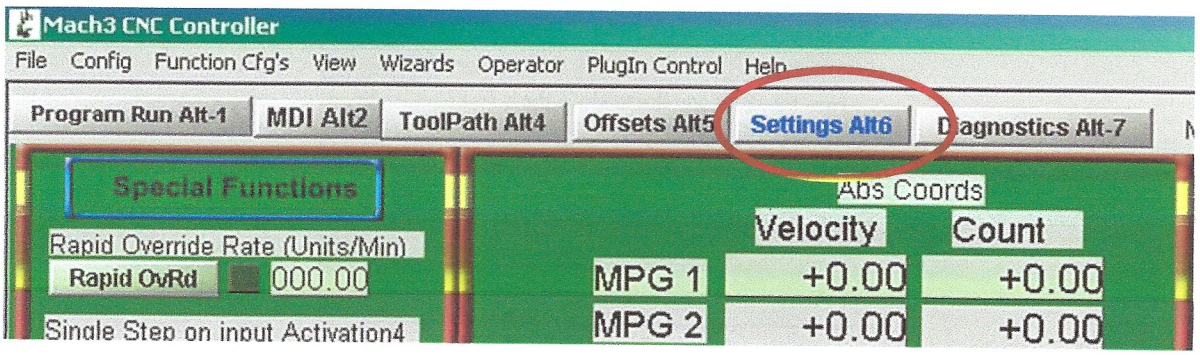
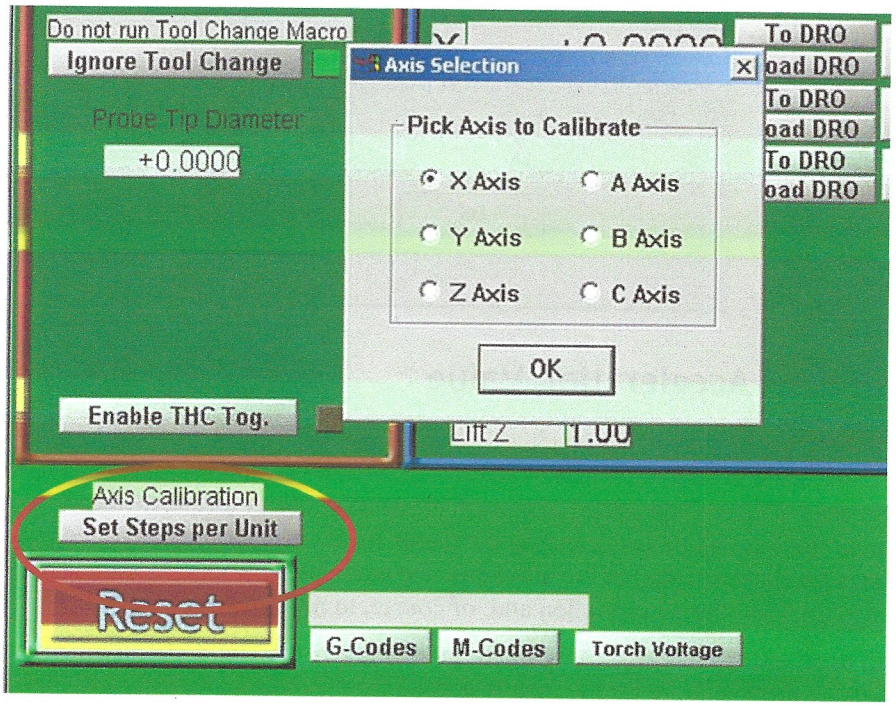


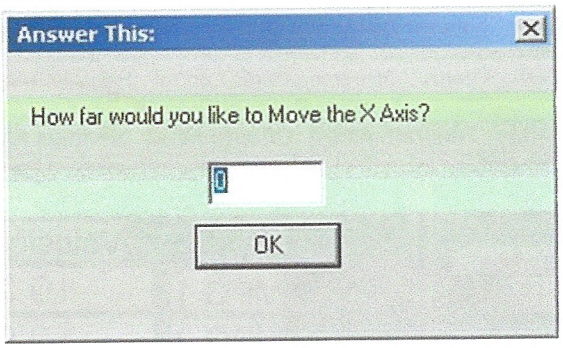
Figure 5-13 shows the *Set Steps per Unit* button on the **Settings Alt6** screen to initiate the automatic settings process. You will be prompted for the axis that you want to calibrate. Select an axis on the **Pick Axis to Calibrate** pop-up menu and click OK.

Figure 5-13: Automatic Setting of Steps per Unit



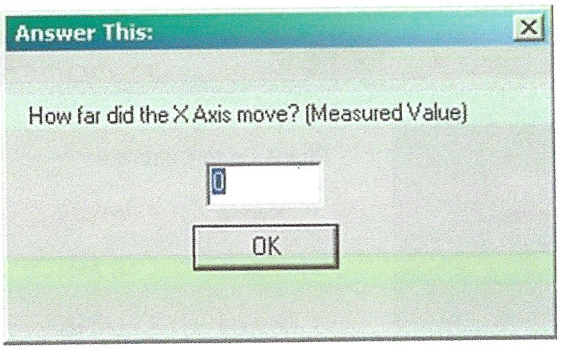
Another pop-up menu, shown in Figure 5-14, asks you to enter a nominal move distance. Mach3 will make that move based on its current settings, which of course may be wildly incorrect. Be ready to press the EStop button if the machine seems about to crash because your existing settings are too far off.

Figure 5-14: Window to Enter Nominal Move Distance



Finally, after the move you will be prompted to measure and enter the exact distance that was moved, as shown in Figure 5-15. This will be used to calculate the actual Steps per Unit of your machine axis. Section 5.5.4 describes how to measure axis travel.

Figure 5-15: Window to Enter Actual Move Distance



5.5.3 Choosing an Acceleration Value

5.5.3.1 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 or workpiece. 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, friction, and inertia forces to be provided at the specified acceleration rate, then all is well. If the torque is insufficient, then the motor will lose steps or stall (if a stepper), or the servo position error will increase (if a servo). If the servo error gets too great, then the drive will probably signal a fault condition, but even if it does not the accuracy of the cut will suffer. This will be explained in more detail shortly.

5.5.3.2 Testing Different Acceleration Values

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

5.5.3.3 Why You Want To Avoid a Big Servo Error

Most moves made in a part program are coordinated 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 also ensures that the speed-required relationship applies during acceleration and deceleration by accelerating all motions at a speed determined by the "slowest" axis.

If you specify an acceleration for a given axis greater than what the machine can deliver, Mach3 will nevertheless assume it can use that value. If in practice the axis motion lags behind what is commanded (i.e. the servo error is big), then the path cut in the work will be inaccurate.

5.5.3.4 Choosing an Acceleration Value

It is 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.

Unless you must have the ultimate in performance from your machine, however, ArtSoft USA recommends simply setting the acceleration value so that test starts and stops sound "comfortable." That may not be very scientific, but it usually gives good results. It is also a lot easier than doing all the calculations.

5.5.4 Saving and Testing Axis

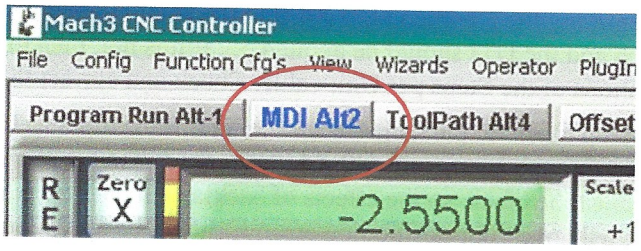
Finally, don't forget to click *Save Axis Settings* to save the acceleration rate before you move on.

You should now check your calculations by using manual data input (MDI) to make a defined G0 move and check the results. For a rough check you can use a steel rule. A more accurate test can be made with a Dial Test Indicator (DTI) and a gage block. Strictly, the DTI should be mounted in the toolholder, but for a conventional mill you can use the frame of the machine as the spindle does not move relative to the frame in the X-Y plane.

Suppose you are testing the X axis and have a 4" gage block.

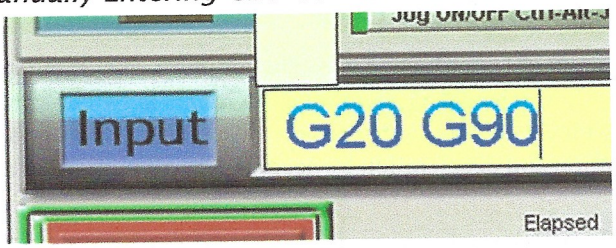
Select the MDI screen (Figure 5-16).

Figure 5-16: Selecting MDI



Click in the Input box (Figure 5-17) and enter commands (G20 G90) to select inch units and absolute coordinates.

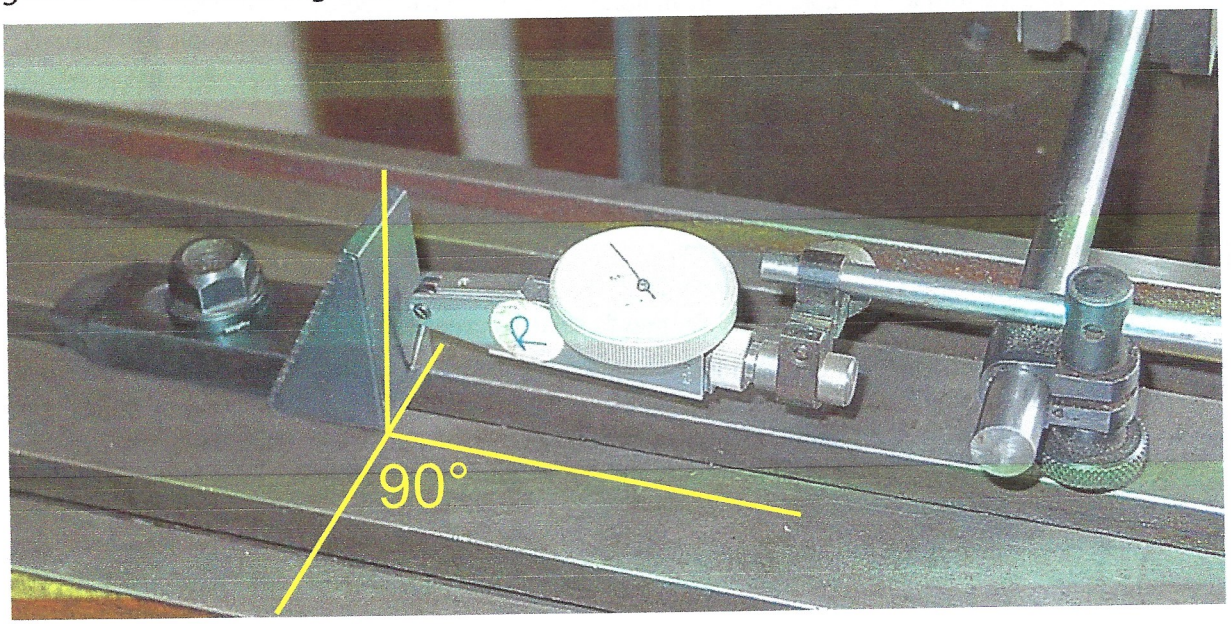
Figure 5-17: Manually Entering G20 G90 Commands



Set up a clamp on the table with its surface perpendicular to the table and to the table travel. Jog the axis so the DTI probe touches it. (See Section 3.2, *Jogging*, for information about jog control buttons and jog mode settings.) Ensure you finish by a move in the minus X direction.

Rotate the DTI bezel to zero the reading. This is illustrated in Figure 5-18.

Figure 5-18: Establishing a Zero Position

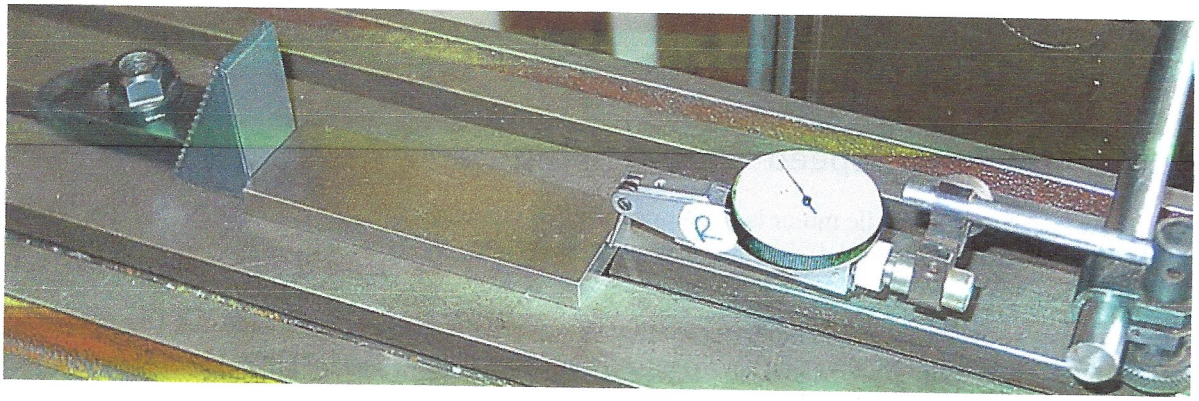


Using the Mach3 MDI screen, click the **Zero X** button to zero the X axis DRO.

Move the table to X = 4.5 by entering the command **G0 X4.5** in the MDI screen's Input box. The resulting gap between the block and the DTI should be about 4.5". If it is not, then there is something badly wrong with your calculations of the Steps per Unit value. Check and correct this.

Set the gage block against the stop block on the table (recall that in this example we're assuming a 4" block) and move to X = 4.0 by using the command **G0 X4.** This move is in the X minus direction as was the jog to zero the DTI against the block, so the effects of backlash in the mechanism will be eliminated. The reading on the DTI will give your positioning error. It should be only up to a thousandth of an inch or so. Figure 5-19 shows the gage block in position.

Figure 5-19: Gage Block in Position



Remove the gage block and issue the command G0 X0 to move back and check the zero value. Repeat the 4" test to get a set of perhaps 20 test values. See how reproducible the positioning is. If you get big variations, then there is something wrong mechanically. If you get a consistent error, then you can fine tune the Steps per Unit value using the technique described in Section 5.5.2.3 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 the MDI screen to issue the G0 X0 command, and check the zero on the DTI.

Click the **Start Teach** button. Click in the **Input** space, and type in 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 (stop)
O1234
G1 X4
G1 X0 (do a feed rate move and move back)
M99 (return)
```

The commands will be executed as you type them, but they will also be saved. When you have typed in all the program commands, click the **Stop Teach** button. Click the **Load/Edit** button. Select the **Program Run** tab. Click the **Cycle Start** button to run the program. Check that the motion sounds smooth.

When it finishes, the DTI should of course read zero. If it does not, adjust the maximum velocity and acceleration of the axis (downward!) and try again.

If the program does not run properly, check to be sure you didn't make a typing error. You can edit the program by clicking the **Edit GCode** button on the **Program Run** tab.

5.5.4.1 Repeat Configuration of Other Axes

With the experience you will have gained configuring the first axis, you should be able to quickly repeat the process for the other axes.

5.5.5 Spindle Motor Speed Control Setup

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

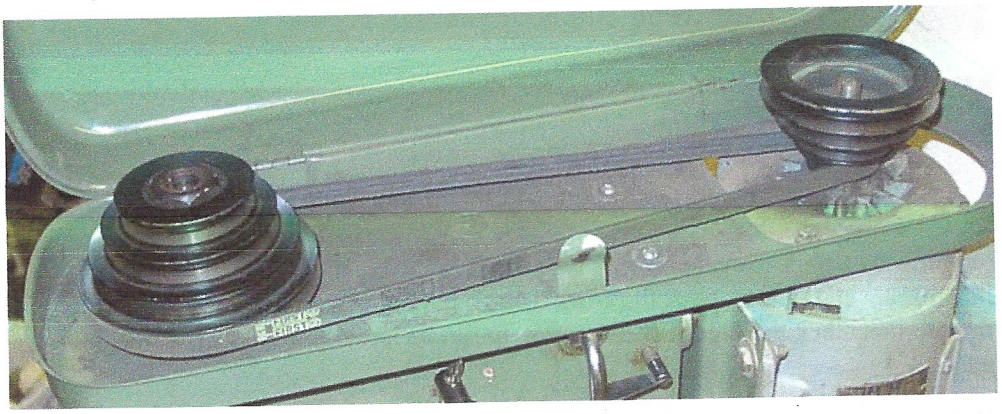
If Mach3 is to control the spindle speed either by a servo drive that accepts Step and Direction pulses, or by a Pulse Width Modulated (PWM) motor controller, then this section tells you how to configure your system.

5.5.5.1 Motor Speed, Spindle Speed, and Pulleys

Step and Direction and PWM both allow you to control the speed of the **motor**. When you are machining, what you and the spindle speed command (the S-word) in the part program are concerned with is the speed of the **spindle**. The motor and spindle speeds are, of course, related by the pulleys or gears connecting them. This section explains how to define the motor/spindle relationship to Mach3.

We use the term "pulley" to cover both pulley and gear drive in this manual. Figure 5-20 shows a step pulley system.

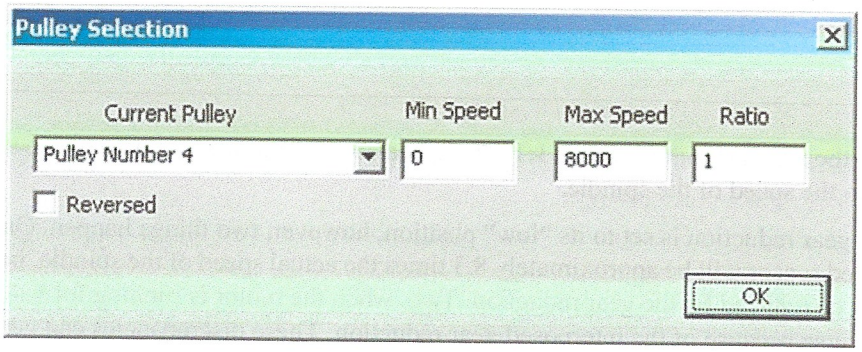
Figure 5-20: Step Pulleys



Mach3 cannot know without being told what pulley ratio is selected at any given time so you, the machine operator, are responsible for supplying that information to Mach3. The information is given in two steps. When the system is configured (i.e. what you are doing now), you define up to fifteen available pulley combinations. These are set by the physical sizes of the pulleys or by the ratios in the geared head. Then, when a part program is being run, the operator specifies which pulley (1 to 15) is in use.

The machine's pulley ratios are defined to Mach3 using the **Config>Spindle Pulleys...** dialog. Figure 5-21 shows a sample dialog window. The dialog lets you define up to fifteen pulley sets.

Figure 5-21: Config>Spindle Pulleys... Dialog



The maximum speed (**Max 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 and at the set *Vel* value on Motor Tuning "Spindle Axis" for Step and Direction. If a speed greater than the Max Speed is requested, Mach3 will display a warning and use the Max Speed value.

If the **Min Speed** feature is used, its value for each pulley should be calculated as a percentage of the maximum speed, with the percentage determined by the minimum speed rating of the motor or controller. It is, of course, also the minimum percentage PWM signal ratio. For example, if a PWM of less than 20% gives unacceptable motor performance, then the minimum speed should be calculated as 20% of the maximum speed. If a speed lower than the minimum is requested (by the S-word etc.), then Mach3 will display a warning and use the minimum acceptable speed. For example, if there is a maximum speed of 1600 rpm on pulley 4 and a minimum speed of 320 (20% of 1600), an S200 command would display a warning and the minimum speed of 320 would be used. This feature is to avoid operating the motor or its controller at a speed below its minimum rating. If you do not want to set up the Min Speed feature, enter a value of 0 for each pulley's Min Speed.

Mach3 uses the pulley ratio information as follows:

1. 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.
2. Otherwise, the percentage of the maximum for the pulley that has been requested is used to set the PWM pulse width, or Step pulses are generated to produce that percentage of the maximum motor speed as set in Motor Tuning for the "Spindle Axis."

For example, suppose the maximum spindle speed for Pulley #1 is 1500 rpm. S1600 would be an error. S600 would give a PWM pulse width of 40% (600/1500). If the maximum Step and Direction speed is 3600 rpm, then the motor would be "stepped" at 1440 rpm (3600 x 0.4).

The **Ratio** value can be set if the actual speed of the spindle is for any reason different than the speed being seen by the spindle speed sensor. This may occur if, because of physical constraints, the speed sensor must be mounted at a point in the spindle drivetrain that has additional gearing between its location and the output spindle.

Check **Reversed** if, because of gearing, spindle rotation of a pulley setting is opposite to the rotation of other pulley settings.

As an example, consider the Bridgeport® Series 1 step pulley J head. It provides a total of eight speeds through a four-step pulley and a two-speed internal gear reduction. Because of the design of the J head, the only reasonable place to mount a spindle speed sensor is on the spindle pulley. When the gear reduction is used in its "high" position, this causes no difficulty. The ratio between the spindle pulley

speed and the actual spindle speed is 1:1. The speed seen by the speed sensor on the spindle pulley will be equal to the speed of the spindle.

When the gear reduction is set to its "low" position, however, two things happen. One, the speed seen by the speed sensor will be approximately 8.3 times the actual speed of the spindle, because the spindle speed will be reduced by the gear reduction. Two, when the motor is running forward, the spindle will turn in reverse because of the interposed gear reduction. These discrepancies can get compensated for in Mach3 by appropriately configuring the pulley entries that correspond to the low speed range. Set the Ratio to be equal to the gear reduction. Mach3 will divide the speed seen by the speed sensor by the Ratio and display the correct spindle speed. Also check the "Reversed" box, so Mach3 will interchange the notion of "forward" and "reverse" for those pulley steps.

Table 5-3 lists the appropriate settings.[1]

Table 5-3: Pulley Configuration for Bridgeport Step Pulley J Head

Pulley	Max Speed	Min Speed	Ratio	Reversed
1	660	0	1	
2	1115	0	1	
3	1750	0	1	
4	2720	0	1	
5	80	0	8.3	X
6	135	0	8.3	X
7	210	0	8.3	X
8	325	0	8.3	X

Min Speed values are not given in Table 5-3 because they will depend on the operating characteristics of your particular motor and driver. (A Min Speed of 0 will always work, but you will not get the overload protection offered by a correctly configured Min Speed.)

5.5.5.2 Pulse Width Modulated Spindle Controller

To configure the spindle motor for PWM control, check the *Use Spindle Motor Output* and *PWM Control* boxes on the **Config>Port and Pins>Spindle Setup** tab (Figure 5-7).

Locate the *PWMBase Freq* box. The value you enter here is the frequency of the squarewave whose pulse width is modulated. This is the signal which appears on the Spindle Step pin. The higher the frequency you choose here, the faster your controller will be able to respond to speed changes but the lower the "resolution" of chosen speeds. The number of different speeds is the *Engine pulse frequency* divided by the *PWMBase freq*. For example, if you are running at 35,000 Hz and set *PWMBase* to 50 Hz, there are 700 discrete speeds available. That is almost certainly sufficient on any real system, as a motor with maximum speed of 3600 rpm could, theoretically, be controlled in steps of less than 6 rpm.

Enter the minimum acceptable percentage PWM signal in the *Minimum PWM* box. Refer to your vendor's product documentation for guidance.

Define an output pin on the **Motor Outputs** tab (Figure 5-5) for the Spindle Step. This pin must be connected to your PWM motor control electronics. You do not need one for Spindle Direction, so set this pin to 0.

1. May not be correct for all step pulley J heads; verify for your particular machine.

Define External Activation signals in **Config>Ports and Pins>Output Signals** to switch the PWM controller on/off and, if required, to set the direction of rotation.

Don't forget to *Apply* all changes.

5.5.5.3 Step and Direction Spindle Controller

To configure the spindle motor for Step and Direction control, check the *Use Spindle Motor Output* and the *Step/Dir Motor* boxes on the **Config>Port and Pins>Spindle Setup** tab (Figure 5-7). Leave *PWM Control* unchecked. Define output pins on the **Config>Ports and Pins>Motor Outputs** tab (Figure 5-3) for the Spindle Step and Spindle Direction. These pins must be connected to your motor drive electronics. *Apply* the changes.

Define External Activation signals on the **Config>Port and Pins>Output Signals** tab to switch the spindle motor controller on/off if you wish to turn off the motor when the spindle is stopped by M5. The motor will not be rotating, of course, as Mach3 will not be sending step pulses but, depending on the driver design, the motor may still be dissipating power.

Now move to **Config>Motor Tuning** for the "Spindle Axis." The units for this will be one revolution. So the Steps per Unit are the number of pulses for one rev (e.g. 2000 for a 10 times micro-stepping drive or 4 x the line count of a servomotor encoder or the equivalent with electronic gearing).

The *Vel* box should be set to the number of revs per second at full speed. For example, a 3600 rpm motor would need to be set to 60. A high line count encoder may limit maximum speed because the maximum pulse rate from Mach3 will be insufficient to drive it (e.g. a 100 line encoder allows 87.5 revs per second on a 35,000 Hz system). The spindle will generally require a powerful motor whose drive electronics is likely to include electronic gearing which overcomes this constraint.

The *Accel* box can be set by experiment to give a smooth start and stop to the spindle. If you want to enter a very small value in the Accel box, you can do this by typing a value rather than using the Accel slider. A spindle run-up time of 30 seconds is quite possible.

5.5.5.4 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 judge it by eye using your experience.

On Mach3 Settings screen, choose a pulley that will allow 900 rpm. Set the belt or gearbox on the machine to the corresponding position. On the Program Run screen, set the spindle speed required to 900 rpm and start it rotating. Measure or estimate the speed. If it is wrong, you will have to revisit your calculations and setup.

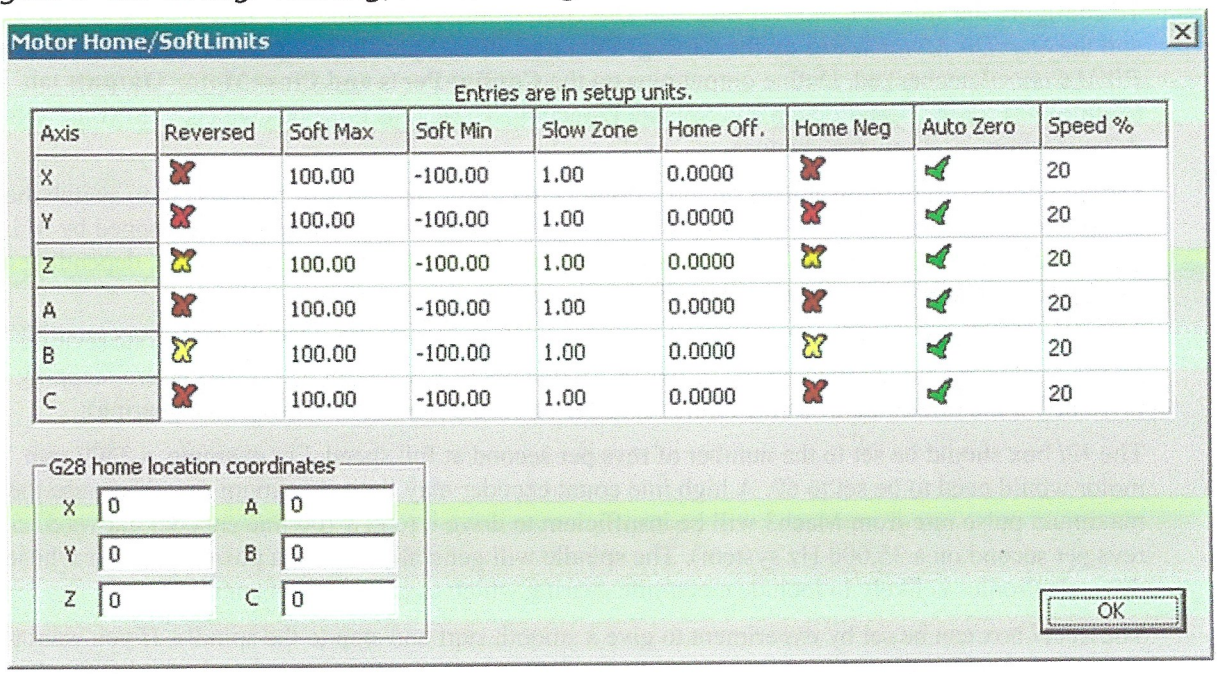
You may want to check the speeds of all the pulley steps in the same way using a suitable set speed for each step.

5.6 Other Configuration

5.6.1 Configure Homing and Soft Limits

The **Config>Homing/Limits** dialog defines what happens when a reference operation (G28.1 or a screen button) is performed. Figure 5-22 shows the dialog.

Figure 5-22: Config>Homing/Limits Dialog



5.6.1.1 Referencing Speeds and Direction

The *Speed %* entry is used to avoid crashing into the stop of an axis at full speed when looking for the reference switch. Mach3 will move the axis at the percentage of full speed that you enter here.

The *Home Neg* entry determines the initial search direction. When you are referencing, Mach3 has no idea of the position of an axis. The direction it moves depends on the *Home Neg* setting. 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 axis will move in the plus direction. Similarly, if the box is unchecked, then the axis moves in the plus direction until the input is active or the minus direction if it is already active.

5.6.1.2 Position of Home Switches

If the *Auto Zero* checkbox is checked, then the axis DROs will be set to the Reference/Home Switch location values defined in the *Home Off.* (offset) column (rather than actual Zero). This can be useful to minimize homing time on a very large and slow axis.

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

5.6.1.3 Configure Soft Limits

Most implementations of limit switches involve some compromises. 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.

The software will refuse to allow the axes to move outside the declared range of the soft limits of the X, Y 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 a *Slow Zone* which is defined for the table.

If the *Slow Zone* is too big, then 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 *Soft Limits* toggle button – see Limits and Miscellaneous control family for details.

If a part program attempts to move beyond a soft limit, then it will raise an error.

The soft limits values are also used to define the cutting envelope if Machine is selected for the tool-path display. You may find them useful for this even if you are not concerned about actual limits.

5.6.1.4 G28 Home Location

The *G28 coordinates* define the position in absolute coordinates to which the axes will move when a `G28` is executed. They are interpreted in the current units (G20/G21) and not automatically adjusted if the units system is changed.

5.6.2 Configure System Hotkeys

Mach3 has a set of global hotkeys that can be used for jogging or to enter values into the MDI line etc. These keys are configured in the **Config>System Hotkeys** dialog shown in Figure 5-23. Click on the button for the required function, then press the key to be used as its hotkey. Its value will be displayed on the dialog. Take care to avoid duplicate use of a code, as that can cause serious confusion.

This dialog also lets you define the codes for external buttons used as OEM Triggers.

Figure 5-23: Configure System Hotkeys Dialog

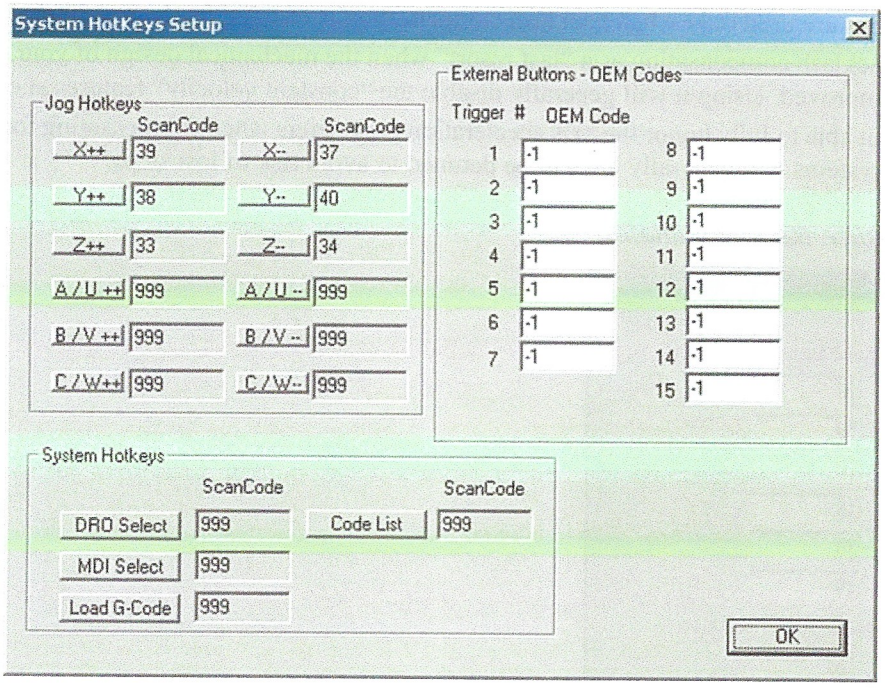


Table 5-4 lists the defaults used for the Jog hotkeys.

Table 5-4: Defaults for Jog Hotkeys

Function	Key	Code	Function	Key	Code
X++	Right Arrow	39	X--	Left Arrow	37
Y++	Up Arrow	38	Y--	Right Arrow	40
Z++	Page Up	33	Z--	Page Down	34

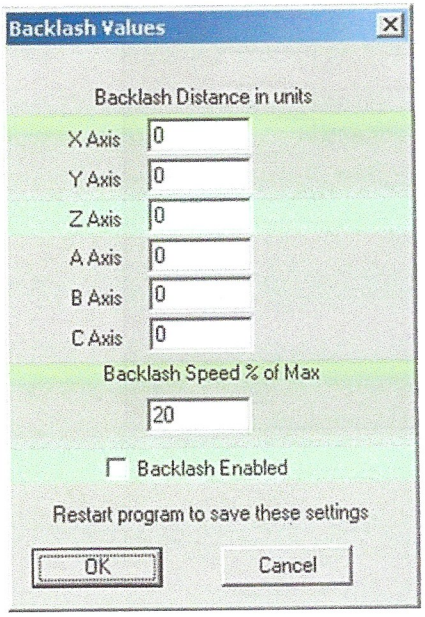
5.6.3 Configure Backlash

The **Config>Backlash** dialog shown in Figure 5-24 lets you provide an estimate of the distance the axis must back up to ensure any backlash is taken up when the final "forward" movement is made. You can also specify the speed at which this movement is to be made.

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 where changes in direction occur. Also note:

- These settings are used only when backlash compensation is enabled by the checkbox.
- Consider backlash compensation as a "last resort" when the mechanical design of your machine cannot be improved. Using it will generally disable the "constant velocity" features at corners.
- Mach3 is not able to fully honor the axis acceleration parameters when compensating for backlash, so stepper systems will generally have to be detuned to avoid risk of lost steps.

Figure 5-24: Configure Backlash Dialog



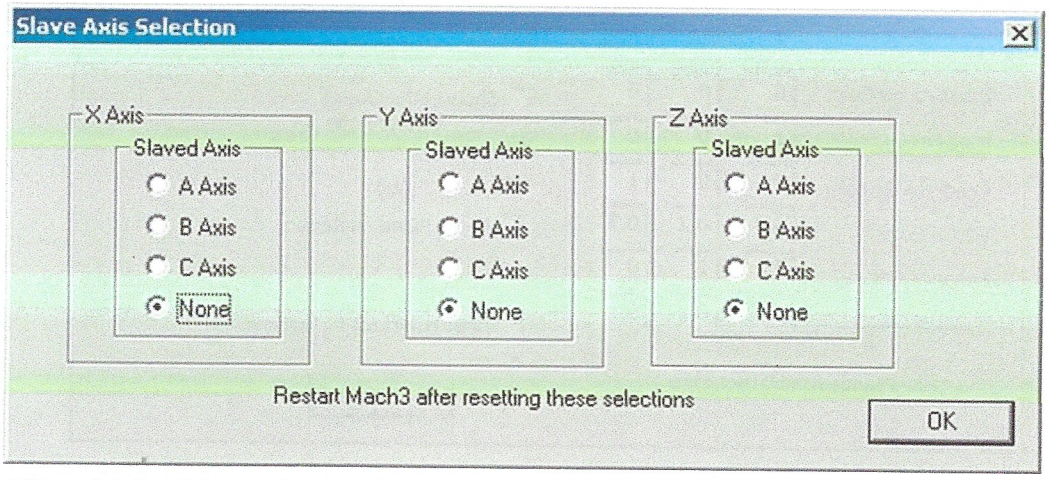
5.6.4 Configure Slaving

Large machines such as gantry routers or mills often need two drives, one on each side of the gantry itself. If these become out of step the gantry will "rack," and its cross axis not be perpendicular to the long axis.

You can use the **Config>Slave Axis** dialog shown in Figure 5-25 to configure Mach3 so one drive (say the X axis) is the main drive and slave another to it (perhaps the C axis configured as linear rather than rotary). During normal use, the same number of step pulses will be sent to the master and slave axes with the speed and acceleration being determined by the "slower" of the two.

When a reference operation is requested, they will move together until the home switch of one is detected. This drive will position just off that switch in the usual way, but the other axis will continue until its switch is detected, then positioned off it. Thus the pair of axes will be "squared up" to the home switch positions and any racking which has occurred be eliminated.

Figure 5-25: Configure Slave Axis Dialog

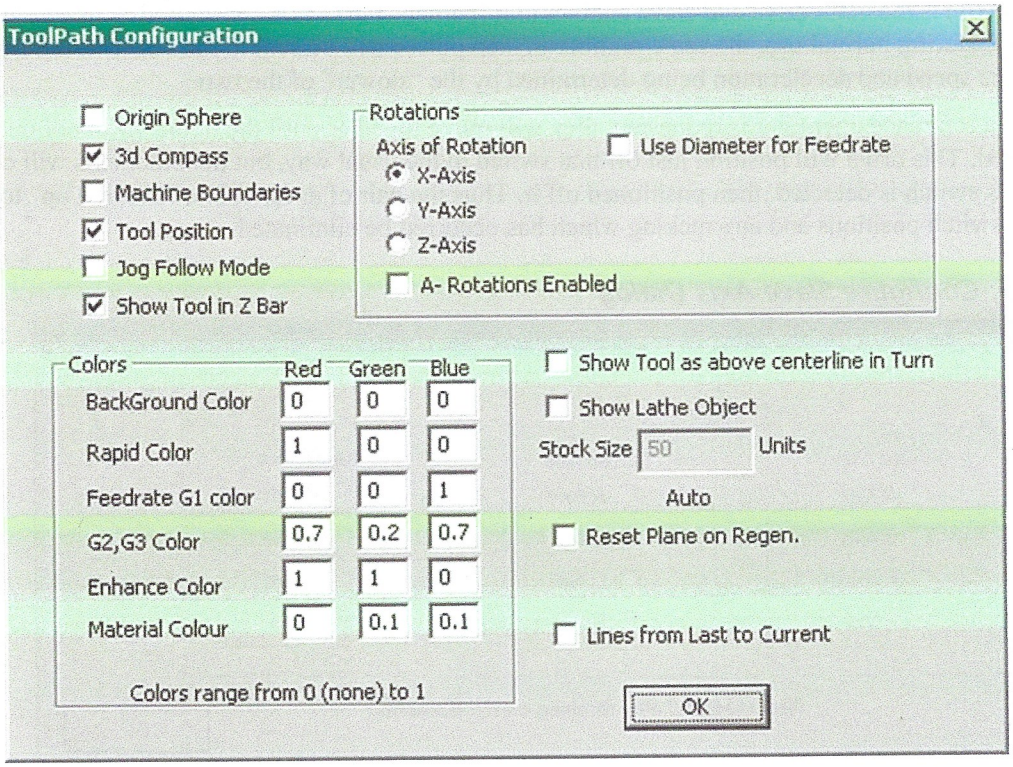


Although Mach3 keeps the master and slaves axes in step, the DRO of the slave axis will not display offsets applied by the Tool table, fixture offsets etc. Its values may thus be confusing to the operator. ArtSoft USA therefore recommend that you use the Screen Designer to remove the slave axis DRO and related controls from all the screens except Diagnostics. *Save As* the new design with a name other than the default, and use the **View>Load Screens** menu to load it into Mach3.

5.6.5 Configure Toolpath

The **Config>ToolPath** dialog shown in Figure 5-26 lets you to define how the toolpath is displayed.

Figure 5-26: Configure Toolpath Dialog



Origin sphere, when checked, displays a filled circle at the point of the toolpath display representing X=0, Y=0, Z=0

3D Compass, when checked, shows arrows depicting the directions of positive X, Y, and Z in the toolpath display.

Machine boundaries, when checked, displays a box corresponding to the settings of the soft limits (whether or not they are switched on).

Tool Position, when checked, shows the current position of the tool on the display.

Jog Follow Mode, when checked, causes the lines representing the toolpath to move relative to the window as the tool is jogged. In other words, the tool position is fixed in the toolpath display window.

ShowTool as above centerline in Turn relates to Mach3Turn (to handle front and rear toolposts).

Show Lathe Object enables the 3D rendering of the object that will be produced by the toolpath (Mach3Turn only).

Colors for different elements of the display can be configured. The brightness of each of the additive primary colors Red, Green, and Blue are set on a scale 0 to 1 for each type of line. **Hint:** Use a program such as Photoshop to make a color you like and divide its RGB values by 255 (it uses the scale 0 to 255) to get the values for Mach3.

The *A-axis* values let you specify the position and orientation of the A-axis if it is configured as rotary and the display is enabled by the *A Rotations* checkbox.

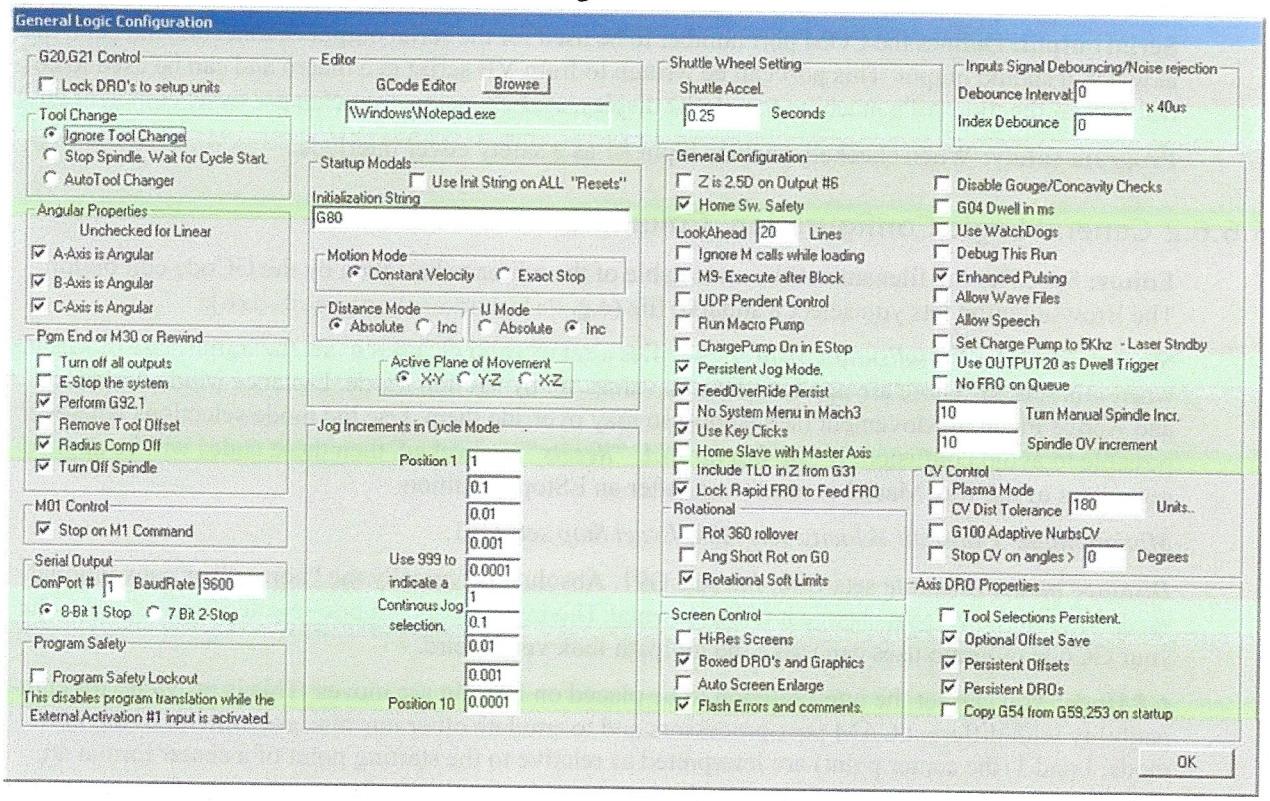
Reset Plane on Regen reverts the display of the toolpath display to the current plane whenever it is regenerated (by double click or button click).

Boxed Graphic displays a box at the boundaries of the tool movement.

5.6.6 General Configuration

The **Config>General Config...** dialog shown in Figure 5-27 lets you define the modes that are active when Mach3 is loaded (i.e. the initial state of the system). The entries are described column by column.

Figure 5-27: General Configuration Dialog



5.6.6.1 General Logic Configuration, Column 1

G20/G21 Control: If *Lock DROs to setup units* is checked, then even though G20 and G21 will alter the way X, Y, Z etc. words are interpreted (inch or millimeter), the DROs will always display in the Setup Unit system.

Tool change: If *Ignore Tool Change* is checked, M6 tool change requests will be ignored. If unchecked, M6 will call the M6 tool change macro.

If *Stop Spindle, Wait for Cycle Start* is checked, an M6 tool change request will stop the spindle and wait for manual restart.

If *Auto Tool Changer* is checked, then the M6Start/M6End macros will be called.

Angular properties: An axis checked as angular is measured in degrees (that is to say G20/G21 do not alter the interpretation of A, B, C words). If A, B, or C is unchecked, that axis will be measured in linear units similar to X, Y, and Z.

Program end or M30 or Rewind: Defines action(s) to take place at end or a rewind of a part program. Check the required functions. ***Caution:*** Before checking the items to remove offsets and to perform G92.1, you should be absolutely clear 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.

M01 Control: If *Stop on M1 Command* is checked, the optional program stop command M1 will be enabled.

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.).

Program safety: When checked, enables Input #1 as a safety cover interlock.

5.6.6.2 General Logic Configuration, Column 2

Editor: Specifies the filename of the executable of the editor to be called by the GCode edit button. The Browse button lets you select a suitable file (e.g. `C:\windows\notepad.exe`).

Startup Modals: ***Initialization String:*** Specifies a set of valid GCodes to set the initial state of Mach3 when it is started. These are applied after the values set by Motion Mode, Distance Mode, IJ Mode, and Active Plane of Movement (see below) so may override them. Use the mode selections wherever possible to avoid confusion. If *Use Init on ALL "Resets"* is checked, then these codes will be applied regardless of the way Mach3 is reset – e.g. after an EStop condition.

Motion mode: *Constant Velocity* sets G64, *Exact Stop* sets G61.

Distance mode: *Absolute* sets G90, *Inc* sets G91. Absolute is typically the "normal" or conventional mode. Use caution if you change it to incremental. Having the wrong mode selected in relation to what your GCode program uses can make the toolpath look **very** weird.

I/J Mode: You can set the interpretation to be placed on I & J in arc moves. This is provided for compatibility with different CAM post-processors 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 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 your part program. **An error in this setting is the most frequent cause of questions from users when trying to cut circles.**

Active Plane: *X-Y* sets G17, *Y-Z* sets G19, *X-Z* sets G18.

Jog Increments in Cycle Mode: The *Cycle Jog Step* button will successively load the values from this list into the *Step* DRO. This is often more convenient than typing a value into the *Step* DRO. Enter the special value 999 to switch to Cont Jog Mode. The list is always ten entries long. When stepped to the end, it will cycle to the beginning of the list. A sequence of values can be duplicated to fill out the ten entries required in the list.

5.6.6.3 General Logic Configuration, Column 3

Shuttle Wheel Setting: *Shuttle Accel* controls the responsiveness of Mach3 to the MPG when it is being used to control the execution of lines of GCode. *Shuttle Accel* is also used by Backlash.

General Configuration: *Z is 2.5D on output #6*, if checked, controls Output #6 depending on the current position in the program coordinate system of the Z axis. If Z is greater than 0.0, then Output #6 will be active. You must have a Z axis configured to use this feature, but its Step and Direction outputs can be configured to a non-existent pin, for example Pin 0, Port 0.

Home Sw Safety, if checked, will prevent motion of an axis during homing if the home switch is already active. This is useful to prevent mechanical damage on a machine that shares limit switches at both ends of an axis with Home.

LookAhead determines the number of lines of GCode that the interpreter can buffer for execution. It does not normally require tuning.

Ignore M calls while loading disables M call execution while the GCode is being interpreted to draw the tool path. Some M calls do things like start spindles, turn on oil pumps, etc., which should execute only when the part program is actually run.

M9 Execute after Block causes a block containing an M9 command to execute the M9 last. M9 turns off coolant. Some postprocessors write a line like: `M9 G01 X##`, which might perform a cut. If M9 was turned off when it was seen, not at the end of the block, the last cut would be made without coolant.

UDP Pendent Control Special OEM option for Ethernet control pendant. Most users can ignore.

Run Macro Pump, if checked, will on startup look for the file `MacroPump.mls` in the macro folder for the current profile and will run it every 200 milliseconds.

Charge pump On in EStop, if checked, retains the charge pump output (or outputs) even when EStop is detected. This is required for the logic of some breakout boards.

Persistent Jog Mode, if checked, remembers the Jog Mode you have chosen between runs of Mach3Mill.

FeedOveride Persist, if checked, retains the selected feed override at the end of a part program run.

No System Menu in Mach3, if checked, eliminates the menu selection bar from the Mach3. This may be useful in a production environment to prevent unauthorized changes to the Mach3 setup parameters. The option will take effect when you restart Mach3. **Use CAUTION when selecting this option!** It will keep you from making any further changes to the Mach3 setup. Be sure you have a way to re-enable the menu bar if you need it. One way to do so is to replace your configuration `.XML` file with a fresh copy, but that will reset all the other configuration parameters as well.

Use Key Clicks, if checked, enables key click sounds for data entry into DROs and MDI. (Not implemented in current version of Mach3.)

Home Slave with Master Axis, if checked, homes both motors of a master/slave pair at the same time. This will not do axis straightening, however.

Include TLO in Z from G31, if checked, will include the tool offset in the probe data.

Lock Rapid Feed FRO to Feed FRO, if checked, will apply any percentage feed rate override you make to the rapid feed (up to 100% of the motor tuning speed) as well as to the standard feed rate.

Rotational: *Rot 360 rollover,* if checked, will measure a rotary axis modulo 360 (0 to 360 then restart at 0). Otherwise, it will keep counting up (for example, two revolutions would be 720).

Ang Short Rot on G0, if checked, makes any rotary axis treat the position given as an angle modulo 360 degrees. Moves will be by the shortest route to that position. For example, if the axis were at 0 degrees and a request was made to rotate to 359 degrees, it would rotate –1 instead of +359.

Rotational Soft Limits, if checked, applies software limit switches to rotary axes.

Screen Control: *Hi-Res Screens,* if checked, will draw the screen twice to help eliminate pixelization. Use this only if you have a good video card and a fast computer.

Boxed DROs and Graphics, if checked, draws a small border around the GCode, MDI, and tool path and a small raised edge around the DRO.

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, ensuring that it fills the entire screen area.

Flash Errors and comments, if checked, will cause any displayed error messages and comments to flash.

5.6.6.4 General Logic Configuration, Column 4

Inputs Signal Debouncing/Noise Reduction: *Debounce interval/Index Debounce:* Specifies the number of Mach3 pulses for which a switch must be stable for its signal to be considered valid. For example, if a system runs at 35,000 Hz, 100 would give about a 3 millisecond debounce time (100 ÷ 35000 = 0.0029 sec). The Index pulse and the other inputs have independent settings.

General Configuration (continued from Column 3): *Disable Gouge/Concavity checks*, if unchecked, Mach3 will check during cutter compensation (G41 and G42) if the tool diameter is too large to cut "insider corners" without gouging the work. Check the box to disable the warning.

G04 Dwell param in Milliseconds, if checked the command G4 5000 will give a Dwell in running of 5 seconds. If the control is unchecked, the dwell value will be treated as seconds. (G4 5000 would give a dwell of 1 hour 23 minutes 20 seconds.)

Use WatchDogs, if checked, triggers an EStop if Mach3 seems not to be running correctly. You may need to uncheck it if you get spurious EStops on slower computers with operations like loading Wizards.

Debug this Run, if checked, gives extra diagnostics to the program designer. ArtSoft USA may ask you to enable this option when you request support.

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 generally select this option.

Allow Wave files, if checked, allows Windows `.WAV` sound clips to be played by Mach3. This can be used, for example, to signal errors or attention required by the machine.

Allow Speech, if checked, allows Mach3 to use the Microsoft Speech Agent for system information messages and "right button" Help text. See the Speech option on the Windows Control Panel to configure the voice to be used, speed of speaking, etc.

Set charge pump to 5kHz - Laser Stndby, if checked, charge pump output or output(s) are a 5 kHz signal (for compatibility with some lasers), rather than the standard 12.5kHz signal.

Use OUTPUT 20 as Dwell Trigger, if checked, will turn on Output 20 whenever Dwell is active.

No FRO on Queue, if checked, will delay the application of feed rate override until the queue of commands waiting to be implemented is empty. This is sometimes necessary to avoid exceeding permitted speeds or accelerations when increasing the FRO above 100%.

Turn Manual Spindle Incr: This box lets you enter the increment to raise or lower the spindle RPM using OEM buttons 350 and 351.

Spindle OV increment: This box lets you enter the percentage increment to raise or lower the spindle RPM using OEM buttons 163 and 164.

CV Control: *Plasma Mode*, if checked, controls Mach3's implementation of constant velocity moves to suit the characteristics of plasma cutters. It does anti-dive and tries not to round corners in some circumstances. For the most part, ArtSoft USA suggests you not select this option unless your machine has very poor acceleration and poor step resolution.

CV Dist Tolerance lets you define the permissible tracking error when operating in constant velocity mode. This will affect the amount of rounding on corners. Setting this to a high value will allow faster movement but increase rounding.

G100 Adaptive NumbsCV (Obsolete.)

Stop CV on angles > n CV attempts to maintain constant velocity during all angular or arc moves while obeying the acceleration parameter. This is not possible during some moves, and this option lets you specify the amount of angular movement permissible in CV mode before automatically switching over to Exact Stop mode. Setting to 90 degrees is usually a good compromise.

Axis DRO Properties: *Tool Selections Persistent*, if checked, remembers the selected tool at shut-down of Mach3.

Optional Offset Save, if checked, will prompt to verify that you want to actually do any save requested in *Persistent Offsets*.

Persistent Offsets, if checked, will save the work and tool offsets in the permanent tables you have selected between runs of Mach3Mill. See also *Optional Offset Save*.

Persistent DROs, if checked, the axis DROs will have the same values on startup as when Mach3 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.

Copy G54 from G59.253 on startup, if checked, will re-initialize 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.

5.7 How The Profile Information Is Stored

Note: ArtSoft USA **STONGLY RECOMMENDS** that you create your own profile(s) by cloning one of the supplied default profiles. If you do so, you will not modify the default profiles when you configure Mach3. That way, the default profiles will always be available to you as a known starting point.

In addition, creating your own profiles will help to keep your configuration information from being overwritten if you install an update to Mach3. If you modify the supplied default profiles (Mach3Mill or Mach3Turn), they will be overwritten by the program update process...and you will lose all your settings. Refer to Section 2.3, *Mach3 Profiles*.

When the `Mach3.exe` program is run, it will prompt you for the Profile file to use. Profiles 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 or other web browser. (XML is a mark-up language used on web pages.)

The Mach3 program installer sets up shortcuts to run `Mach3.exe` with default Profiles for a Mill and for Turning (i.e. Mach3Mill and Mach3Turn). You can 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` (use the Mach3Loader shortcut) 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. To see an example, inspect the Properties of the Mach3Mill shortcut. This can be done by right clicking the shortcut and choosing Properties from the menu.

An .XML file for a profile **can** be edited by an external editor such as Notepad, but you are **very strongly** advised not to do this unless you are fully conversant with the meaning of each entry in the file. Some users have encountered very strange effects with mis-formatted files. Notice that some tags (e.g. the screen layout) are created only 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.**

When a new profile is created, then a folder for storing its macros will be created. If you are "cloning" from a profile with custom macros, then you must take care to copy any such custom macros into the new profile.

5.8 Documenting Your Setup

You can use Table 5-5 and Table 5-6 to write down the settings you have made.

Table 5-5: Document Your Motor Outputs Settings Here

Signal	Enabled	Step Pin #	Dir Pin #	Dir Low Active	Step Low Active	Step Port	Dir Port
X Axis							
Y Axis							
Z Axis							
A Axis							
B Axis							
C Axis							
Spindle							

Table 5-6: Document Your Input Signal Settings Here

Signal	Enabled	Port #	Pin #	Active Lo	Emulated	Hotkey
X ++						
X --						
X Home						
Y ++						
Y --						
Y Home						
Z ++						
Z --						
Z Home						
A ++						
A --						
A Home						
B ++						
B --						
B Home						
C ++						
C --						
C Home						
Input #1						
Input #2						
Input #3						
Input #4						
Probe						
Index						
Limit Ovrd						
EStop						
THC On						
THC Up						
THC Down						

This is a blank left-hand page for two-sided printing.

Appendix A Torch Height Controls in Mach3

There are several settings, LEDs, and DROs that affect plasma usage and torch height controls (THC) in general in Mach3. Table A-1 lists those controls and their effect on program run-time behavior.

Table A-1: Torch Height Controls in Mach3

Item	Meaning	Value
DRO #25	THC Speed	0-100% rapid Z speed
DRO #27	THC Max Correction	Max height
DRO #65	THC Min Correction	Min height
DRO #26	THC Current Correction	Current correction
DRO #177	Pierce Delay	
LED #35	THC On/Off indicator	
LED #36	THCON signal (torch OK) condition	
LED #37	THC UP signal	
LED #38	THC DOWN signal	
BUTTON#123	THC On/Off control	
BUTTON#124	Calibrate THC – Sets Z to current location	Zeros correction
BUTTON#379	Turns on THC without THCMODE (realtime Z correction)	
BUTTON#380	Turns above mode off	

Notes:

- Turning spindle ON sets the THC correction to zero. Min/Max step counts are then effective from the current Z.
- Only licensed versions can enable THC mode.
- When in THC mode, motion will not begin until THCON (Torch OK) signal becomes active. Motion may be delayed by Pierce Delay.
- Pierce Delays tells the system not to move until *delay* seconds have passed since spindle (torch) was turned on.
- Config/spindle Torch AutoOff setting shortens delay in takeoff, and also turns off torch automatically if motion stops.

This is a blank left-hand page for two-sided printing.

Index

A

Acceleration 5-22
  configuring 5-22
  importance of configuration for accuracy 5-23
Active Hi convention 4-2
Active Lo convention 4-2
Active plane
  configuring default 5-36
Addons, installed with Wizards 2-2
Allow
  speech 5-38
  wave files 5-38
Ang Short Rot on G0 5-37
Angular axes
  configure 5-35
Arcs display or cut incorrectly
  wrong IJ Mode setting 5-36
Auto Screen Enlarge 5-37
Axis drive design
  design calculation 4-6
  limitation by Mach3 step rate 4-7
  minimum possible move 4-6
  rapid speed 4-7
Axis jogging 3-3
  by MPG 3-4
  joystick 3-4
  keyboard 3-3
    continuous jogging 3-4
    step jogging 3-4
  rate override 3-4
  rate override by joystick throttle 3-4
  slow speed 3-4
Axis slaving
  purpose and referencing 4-15

B

Backlash
  configure 5-32
  try to avoid 5-32
  uses Shuttle Accel setting 5-36
Belt or chain pitch 5-17
Block, name for a line of GCode 3-5
Blue Screen of Death
  action to take after, 2-9
Boxed DROs and Graphics 5-37
Breakout board 4-3
Button
  control on screens 3-2

C

Charge pump
  on during EStop 5-37
  setting frequency 5-38
Circles display or cut incorrectly
  wrong IJ Mode setting 5-36
Concavity checks 5-38
Configure
  acceleration 5-22
  action at M30 5-35
  action at program end 5-35
  angular axes 5-35
  backlash 5-32
  belt or chain pitch 5-17
  DROs locked to initial units 5-35
  initial state 5-35
  Mach3 steps per unit 5-18
  max motor speed 5-19
  motor revs per unit 5-16
  motor steps per revolution 5-18
  permanent DROs 5-39
  pulse widths 5-20
  referencing 5-29
  rotary axis units 5-18
  screw revs per unit 5-16
  serial output 5-36
  slaving 5-33
  steps per unit 5-16
  tool change action 5-35
Coolant
  configuring 5-11
Coolant control 4-20
Coordinates of ref switches 5-30
CPR (cycles per revolution) 5-18

D

Debounce
  configure 5-38
Digital Readout See DRO
Digitize probe interfaces 4-20
Direction & Step interface See Step & Direction
Downloading Mach3 2-1
Driver (parallel port)
  manual installation 2-9
  test program 2-5
DRO
  cancelling entry in 3-3
  caution when changing axis 3-3
  control on screens 3-2
  entering data to 3-3
DROs locked to initial units 5-35
Dwell time, for spindle 5-12

E

Editor program
  configure filename for 5-36
Emulated input signals 5-7
Encoder inputs
  defining 5-9
Encoder/MPGs tab 5-9
Engine frequency

choice of suitable value 5-3
defining 5-3
Enhanced pulsing
processor requirement for 5-38
EStop button
emergency stop 1-4
function 4-4
lockout until reset 4-5
not involving software 4-4

F
Fault finding
port addressing and connections 5-14
Flash Errors and comments 5-38
Flood coolant 4-20

G
G04, dwell
units of P word in 5-38
GCode window
control on screens 3-2
Gouge checks 5-38
Ground
signal 4-3

H
Hi-Res Screens 5-37
Home Slave with Master Axis 5-37
Home switch
not near axis limit 4-15
purpose 4-9
Home switch See also Limit switches
Hotkeys
configuring 5-31
defaults for jog 5-32

I
Ignore M calls while loading 5-37
IJ Mode
defines how G02/G03 are interpreted 5-36
Include TLO in Z from G31 5-37
Index
defining pin for pulse 5-7
interface for spindle 4-22
Initial state
configure 5-35
Initialization string 5-36
Input pins
shortage of 5-7
Input signals
interfacing 4-23
Input Signals tab 5-5
Inputs
defining which to be used 5-4
Installation
errors after 2-9
Installation of Mach3 2-1
Interlock
switch for guards 5-36

J
Jog Follow Mode 5-34
Jog step
setting size 3-4
Jogging See Axis jogging

K
Keyboard
shortcuts 3-3
Keyboard emulator
for additional input signals 5-7

L
LED
control on screens 3-2
Light Emitting Diode See LED
Limit switches 5-7
cabling 4-13
defining override switch input 5-7
external implementation 4-10
logical OR for electronic switches 4-11
manual override 4-14
microswitches
accuracy of 4-12
overtravel 4-12
mounting 4-13
purpose 4-9
sharing Mach3 inputs 4-10, 4-14
Limits, soft
Soft limits 5-30
Linear glass scale
not part of servo loop 4-22
quadrature interface 4-20
Lock Rapid Feed FRO to Feed FRO 5-37
LookAhead 5-37

M
M01 Control 5-36
M30, configure action at 5-35
M9 Execute after Block 5-37
Mach3
charge pump monitor See Charge pump
Components of 2-4
computer requirements 1-4
demonstration version 2-1
how distributed 2-1
on laptop 1-4
pulse monitor See Charge pump
what features it supports 1-4
what machines it can control 1-4
Mach3 Addons for Mill, installed with Wizards 2-2
Manual data input (MDI) 5-23
Manual Data Input See MDI
Manual Pulse Generators (MPGs) 1-3
defining inputs 5-9

for jogging 3-4
Maximum spindle speed 5-26
Mcalls, ignore while loading 5-37
MDI
control on screens 3-2
screen 3-5
teaching function 3-5
use of history 3-5
MDI (manual data input) 5-23
Mill Options tab 5-13
Mist coolant 4-20
Motor
reversing direction 5-20
revs per unit 5-16
steps per revolution 5-18
Motor Outputs tab 5-4
Motor pulleys See Pulleys
Motor Tuning dialog 5-19
Mouse
using Mach3 without 3-3
MPG
See Maual Pulse Generators (MPGs)

N

Native Units dialog 5-14
NC machine
parts of 1-2
No System Menu in Mach3 5-37
Noise
on signal ground 4-3

O

OEM Trigger inputs 5-7
OEM triggers
configuring 5-31
Offset save dialog 5-39
Output signals
interfacing 4-23
Output Signals tab 5-8
Outputs
defining which to be used 5-4

P

Parallel port
D25 connector pinout 4-2
general background 4-1
Parallel port driver
manual install 2-9
test program 2-5
Part Program
running a sample 3-5
PC
configuration required 1-4
Permanent DROs
configure 5-39
Persistent feed override 5-37
Persistent jog mode 5-37
Persistent offsets 5-39
Plasma Mode 5-38
Port addresses, finding with Windows Control Panel 5-3
Port Setup and Axis Selection tab 5-2
Ports and Pins dialog 5-1
Probe - design requirements 4-20
Profiles
how stored 5-39
multiple to allow control of several machine tools 5-39
persistence when upgrading Mach3 2-10
selecting on rselected by the /p argument in shortcut target 2-4
selecting on running Mach3.exe 2-4
specified in /p argument 5-39
Program end
configure action at 5-35
Pulleys
explanation of 5-26
ratios for spindle drive 5-12
setting max speed of 5-26
Pulse width modulated control
of motor speed 4-18
Pulse width modulated spindle See Spindle, PWM
Pulse widths
configuring 5-20
PWM
PWM speed control See also Spindle
PWM minimum speed limit 5-12
PWM See Pulse width modulated

R

Re-boot during installation
how to manually uninstall driver if you fail to do it 2-10
reason for 2-4
Referencing
configure 5-29
details of Mach3 actions 4-14
Retrofitting old CNC machines
caution 1-4, 4-5
Reversing motor direction 5-20
Rot 360 rollover 5-37
Rotational Soft Limits 5-37
Run Macro Pump 5-37

S

Save offsets 5-39
Screen
automatic enlarge 5-37
boxed DROs and graphics 5-37
example of LED 3-2
Hi-Res 5-37
sample layouts 3-1
Screw revs per unit 5-16
Secondhand equipment, a caution 4-5
Serial output
configure 5-36

Servo motor drives
properties 4-5
Setup units
choosing between inch and millimetres 5-14
Shortcuts See Keyboard shortcuts
Signal
ground 4-3
Slave axis See Axis Slaving
Slaving
configure 5-33
Soft limits See Limits
Soft limits, rotational 5-37
Special Mach3.sys driver
need for 2-9
SpecialDriver.bat 2-9
Speech, allow 5-38
Spindle
control options 4-18
CW and CCW control configuring 5-11
delay dwell time 5-12
drive, configuring 5-10
PWM speed control 5-12, 5-28
Step & Direction speed control 5-29
Step and Direction control 5-12
Spindle control
clockwise/counterclockwise 4-18
interlocking requirements 4-18
Spindle Setup tab 5-10
Spindle speed
maximum, defined for pulleys 5-26
Step & Direction
spindle drive 4-18
Step & Direction interface
caution about active hi/lo 4-8
waveforms 4-8
Stepper motor drives
limit of capability 4-5
properties 4-5
Steps per unit 5-16
S-word, spindle speed command 5-26
System hotkeys
configuring 5-31

T

Tangential control
of knife etc 4-20
Teaching
to store sequence of MDI commands 3-5
Testing
axis calibration 5-23
configuration settings 5-14
for lost steps 5-25
spindle drive 5-29
THC, torch height controls A-1
Timing
defining pin for pulse 5-7
Tool change action
configure 5-35
Toolpath
display configuration 5-33
Toolpath display
control on screens 3-2
Torch height controls (THC) A-1
TTL
current sourcing and sinking 4-2
signal levels 4-2

U

UDP Pendent Control 5-37
Un-installation of driver
manual 2-10
Use Key Clicks 5-37

V

Vista, may require a registry patch 2-4

W

Wave files, allow 5-38
Wizards, installation 2-2