

Mach3 Tutorial

Setting up a basic three axis milling machine.

Based on Mach3 2.0

Purpose.

The purpose of this tutorial is to help and to guide the user to, step by step, set up and tune the Mach3 CNC controller application for use on a basic 3 axis milling machine. We will go thru the steps of setting up the emergency stop, the charge pump circuit the main axis motors, the spindle and coolant as well as the home switches and the software based over-travel limits. The tutorial is based on the Mach3 v2.0 series and is meant to be used as complement to the existing *Using Mach 3 Mill* manual.

Let's get started.

When first installing the software it's vital to reboot the computer as the installer prompts. If you don't do this the software will not work and you will have to manually remove the Mach3 driver from the system. After installing the software and rebooting the computer you should have four icons on your desktop. These icons all launches Mach3 but it also loads a different set of screens depending on what kind of machine it is we are running. Since this tutorial is aimed at getting a milling machine up and running we'll start the software with the one labelled Mach3Mill. When the software is started you may be presented with the following dialog but if not don't worry.



Figure 1: Hardware plugin.

Since we are going to use the computers printer port as the electrical interface between the computer and our machine we make sure that the *Normal Printer Port Operation* mode is selected and since we don't want to tell Mach3 this information each time we start the software we put a checkmark in the box next to *Don't mask me this again* and then click OK.

Metric or imperial.

The next thing we need to do is to select the native units of choice. To do this we go to the *Config* menu and click *Select Native Units* and select either inches or mm and click OK. Since I'm a metric guy I choose mm.



Figure 2: Default Units

Please note that, as the message says before you get to the windows in figure 2, this selection is not for switching the actual displayed coordinates between inches and mm nor is it meant to switch between running part programs written in inches or mm. It is ONLY for setup and tuning of the motors.

The hardware interface and connections.

Now we need to do is tell Mach3 how many parallel ports and at which addresses they are located. If the port is onboard your computers motherboard the standard address is 0x378. But other addresses are sometimes used as well. We tell Mach3 this information by selecting *Ports and Pins* under the *Config* Menu.

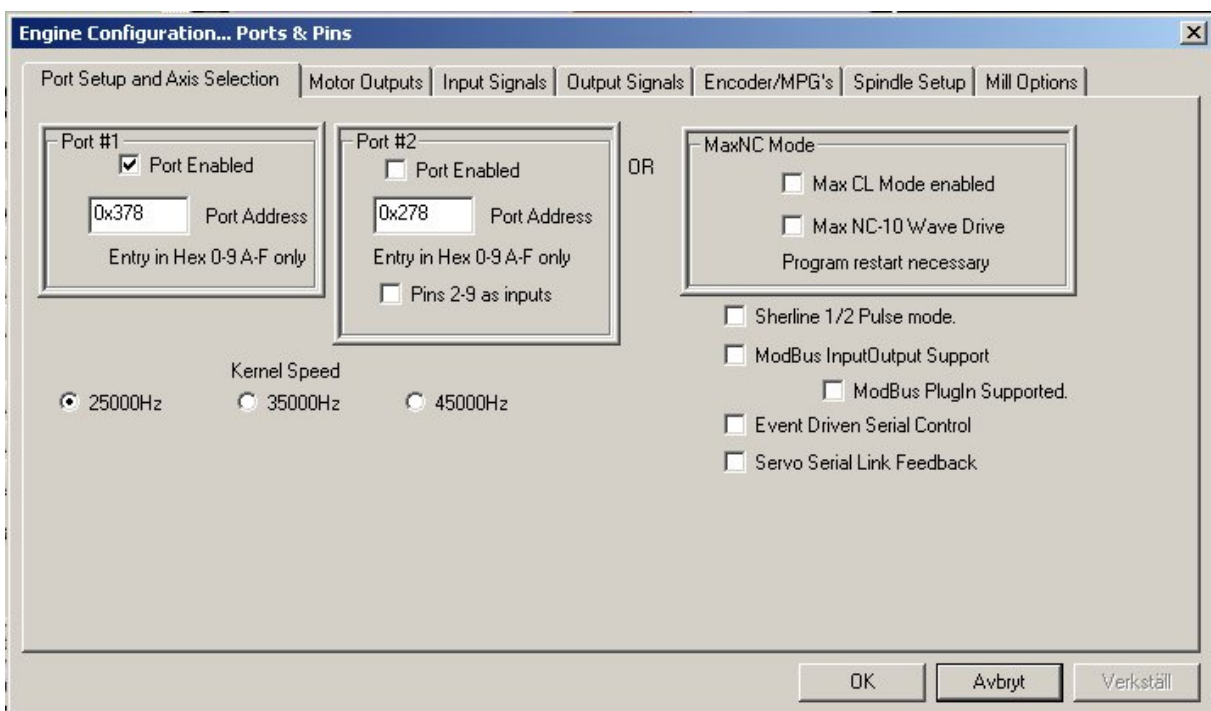


Figure 3: Engine Configuration, Ports & Pins

First check that the address of Port#1 corresponds to the actual address of your printer port and that it is enabled. 0x378 is the address most onboard parallel ports use. Since we won't be using a second port here we make sure that Port#2 is disabled. (No checkmark in the *Port Enabled* box under Port#2).

Next thing to do is to select our Kernel speed. This is the frequency at which the Mach3 driver operates and is also the maximum frequency the software will output to your motor drives. We will use 25000Hz here. Also make sure that none of the options on the right side is enabled and then click *Apply*.

Next in line is the *Motor Outputs* tab. This is where we tell Mach3 how many motors we want to control and to which pins in our printer port the each motor driver is connected. In this case it's three axis, X, Y and Z so we enable those three by making sure that there is a green checkmark in the first column of those axis.

The second column sets the pin to which the step input on our motor drive is connected. In this case the X axis drive step input is connected to pin 2, Y to pin 4 and Z to pin 6. The third column is just like the second but for the drives direction inputs, wired to 3, 5 and 7 on this machine. The actual pin out of your machine may be different. If you have wired it yourself you probably know the pin out and if you bought the machine and/or driver box please consult the documentation for that particular machine or check with the hardware manufacturer.

The setting of the forth and fifth column depends on how the drives are built and how we have them wired. The most common drives uses optically isolated inputs and are usually supplied with a constant +5V DC from the PC. The drives step and direction inputs are then wired to the computers parallel port which internally switches the pin to ground making the current flow thru the LED in the drives opto isolator which in turn makes the drive move the motor one step. This makes our step-line *active low* – that is it's 'on' when the output is low. Makes any sense?

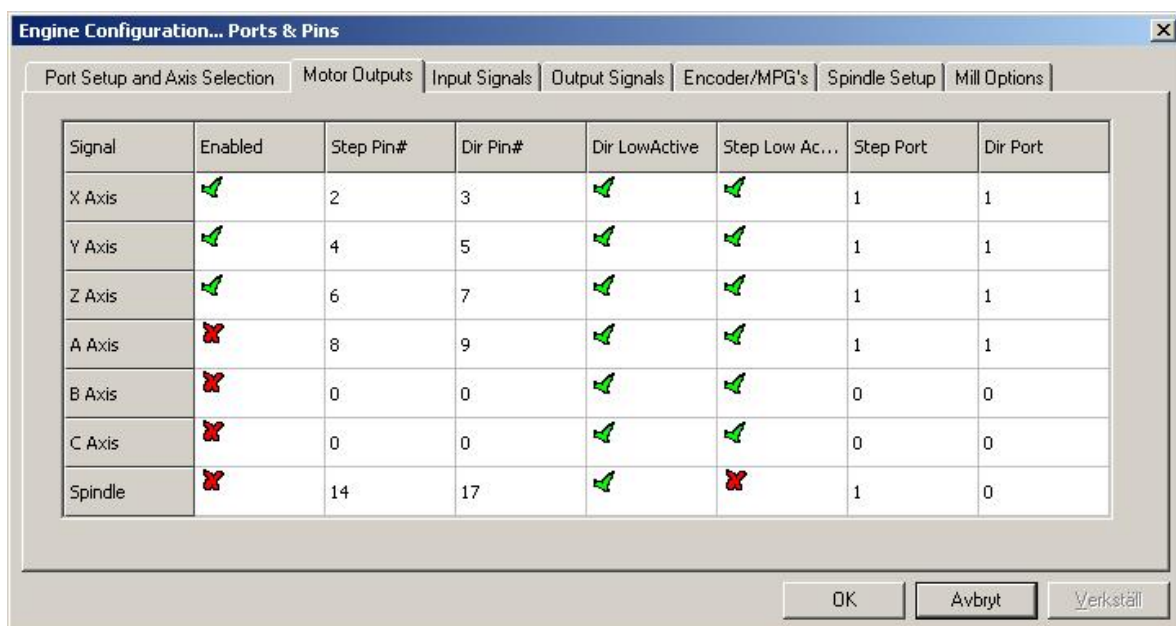


Figure 4: Engine Configuration, Motor Outputs tab.

OK, that was the physical connection for the motors. Next thing to take care of would be to set how many steps per unit we have and then go on to actually tune the motors. But before we do that we need to tell the system about one more important thing.

The Emergency Stop button.

Any decent machine tool should have an Emergency Stop button that, in the safest possible way, halts all machine movement and prevents damage to the operator in the first place and/or the machine – so will this one. This tutorial will not go into the actual hardware design of a proper E-stop system but will focus on getting Mach3 to understand that the big red button has been pushed. The various input signals to Mach3 are setup on the *Input Signals* tab under *Ports and Pins* in the *Config* menu:

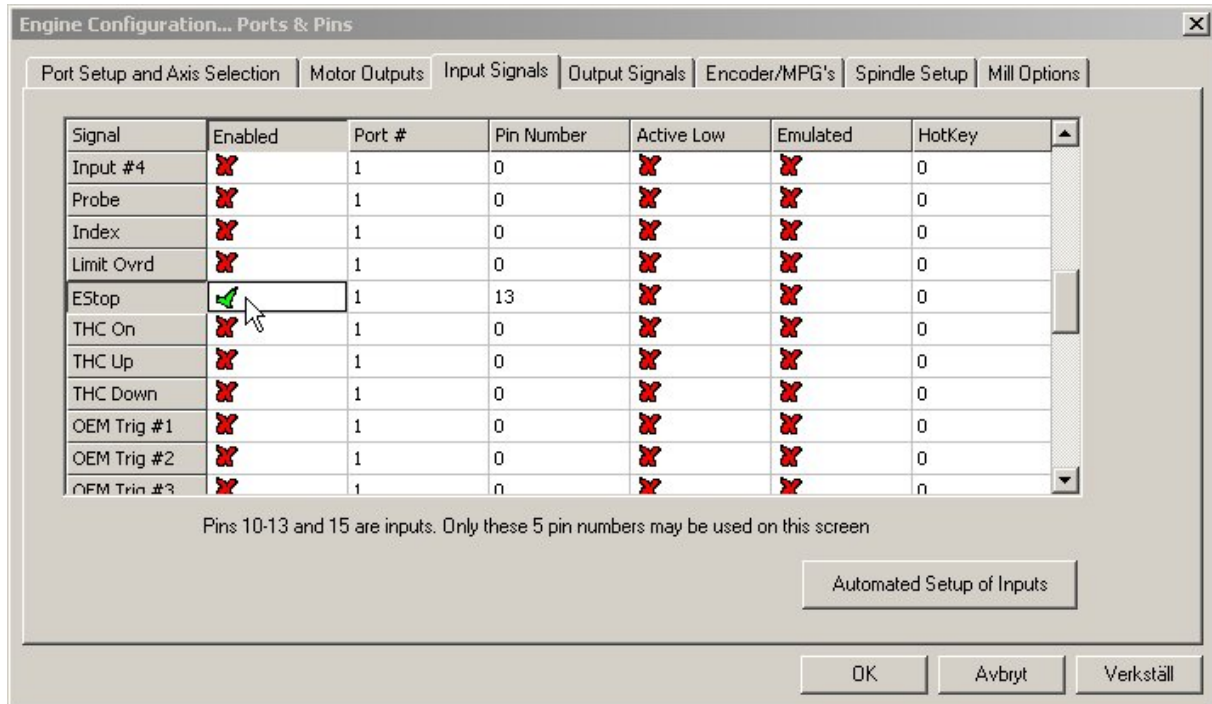


Figure 5: Engine Configuration, Input Signals tab.

About halfway down the list we should be able to find the EStop signal, make sure it is enabled. In this case the hardware is wired in such way that the input is brought high when the switch is pushed so the *Active Low* setting should not be active. In case you don't have an E-Stop switch you will probably have to enable the *Active Low* option to be able to take the software out of E-stop mode but more on that later. As the E-Stop is the only input we'll use for now we make sure that none of the other inputs are enabled. Click OK to save the settings and close the dialog.

Now lets see if we can get the system out of E-Stop mode. Make sure your hardware E-stop button is not pressed then, on the main screen click the big red button labelled *Reset*. Hopefully the blinking frame around the button will stop blinking and turn in to a nice solid green. If it doesn't turn green go back to the dialog in figure 5 and reverse the *Active Low* setting for the E-stop input as described above and try again.

With the E-stop input out of the way and working we'll move on.

Motor Tuning:

Finally we'll get to actually spin some motors. The first thing in the tuning process is to calculate how many steps per unit of travel we have. This depends on a few things:

In case of a step motor:

- The amount of steps per revolution, most commonly 200.
- The step resolution of the motor drive, full step, half step, 5, 10, 100 micro steps etc.

In case of a servo motor.

- The amount of quadrature counts produced by the encoder on the motor
- The 'encoder mode' of the motor drive. 1, 2 or 4 encoder counts.

In both the above cases:

- The reduction ratio between the motor shaft and lead screw.
- And finally, the thread pitch of the screw. (How far the table moves with each rotation of the screw).

It should be noted that other actuating systems could be used like belts, rack & pinions etc but here we will use a lead or ball screw as the basis for our calculation. We will do one metric and one imperial example.

The metric one:

Let's assume we have a standard step motor with 200 steps per rev. This motor is driven by a driver set to 5 micro steps per full step. A Gecko G210 from Geckodrive for example. The motor is directly coupled to the lead screw which has a pitch of 5mm per revolution. That means the axis will move 5mm for each revolution of the screw.

So we'll take the motors 200 steps multiply that by the drives 5 micro step. ($200 \times 5 = 1000$) The drive needs 1000 pulses (or steps) to turn the screw one revolution thus making the axis move 5mm. So now we take those 1000 steps and divide by the pitch of the screw, which is 5. ($1000 / 5 = 200$). In other words we need 200 steps to move one *unit* or mm.

The imperial one:

Let's assume a DC-servo with a 500 line encoder and a drive that uses all four quadrature counts of the encoder effectively making it a 2000 counts per rev motor/encoder combo. Let's also assume we have 3:1 belt reduction driving our screw which has a 5TPI pitch.

The drive needs 2000 pulses, or steps, to turn the motor one revolution. But since we have the 3 to 1 belt reduction between the motor and the screw we need to multiply the 2000 by 3 to make the screw turn one revolution.

6000 steps will make the screw turn one revolution, making the axis move 1/5 of an inch. To make the axis move one inch we need the screw to make five revolutions so $6000 \times 5 = 30.000$ steps per unit or inch.

In reality a steps per unit value as high as 30.000 will greatly reduce the speed at which the machine can move.

So now we have determined how many steps the computer needs to send out to the drives in order to move the machine one of our selected units, be it inch or mm. Let's tell Mach3 what we've come up with. In the *Config* menu, we select *Motor Tuning* and the following dialog should present itself:

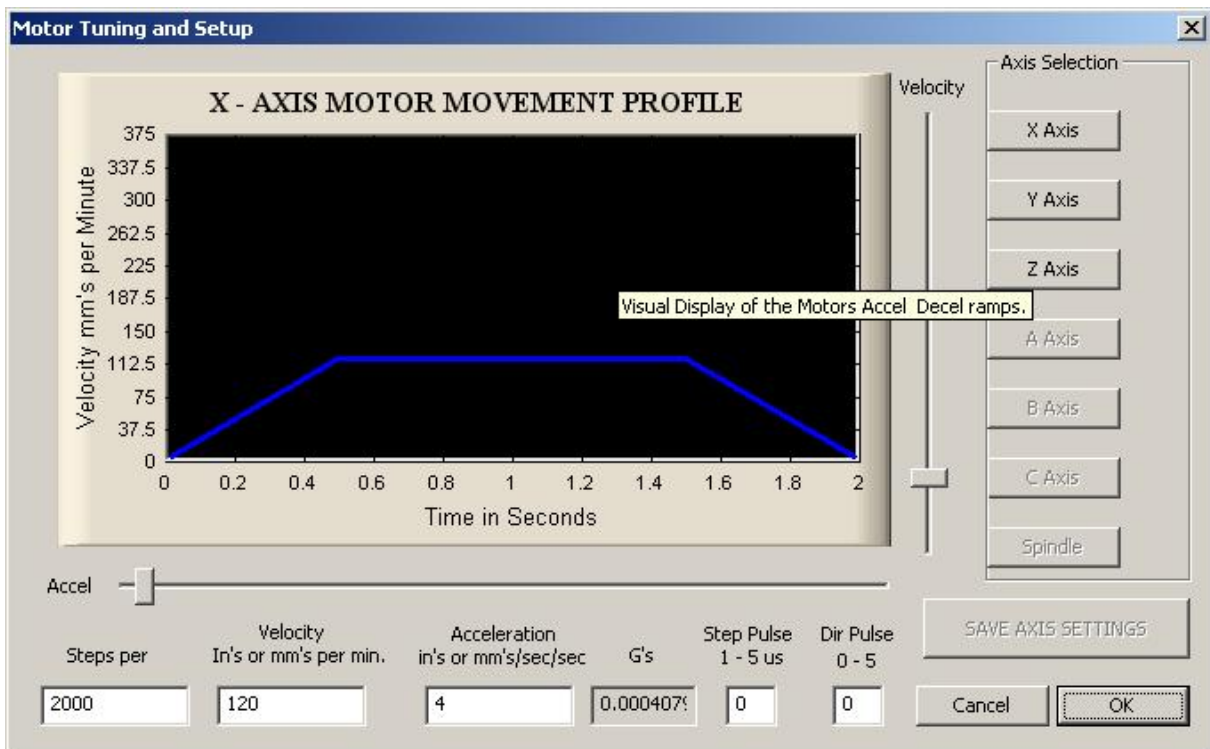


Figure 6: Motor Tuning and Setup.

The first thing to do is to enter the steps per unit we found in our calculation. We do this in the lower left corner. We'll use the metric calculations above as base for this setup so 200 steps per unit it is. Now comes the fun part. By dragging the sliders to the right and below we'll adjust the speed and the acceleration of the motor. This can of course be calculated based on motor torque, machine inertia, mass etc but is mostly done on a trial and error basis.

Start by moving the velocity slider up a bit and then press the up or down arrow key on the keyboard. The motor should now spin, if not make sure that the system is enabled and not in E-stop mode.

By adjusting the Velocity and Accel sliders try to find a speed and acceleration that feels comfortable and where the motor run smoothly without any tendency to jerk, halt, or loose steps. Then, and this is important, click *Save Axis Settings*. Now click the Y Axis button and repeat the process for that axis and then do the same for the Z axis. Don't forget to press *Save Axis Settings* before switching axis. If you don't your setting will be lost.

Please note: It's in perfect order to have a different step per unit setting as well as Velocity and Accel values for each axis. Mach3 will keep them in perfect sync either way. You may even have fractional steps per unit set for one or more axis, for example 201.3 if that is what you need.

If you can't seem to get the motors to run smoothly there are a few things to check, first the easy one: Some motor drives need to have a longer step pulse and this can be set by changing the *Step Pulse* setting. See the manual for your particular drive.

The other thing to check is the voltage at parallel port pins. Most drives inputs are optically isolated and designed to accept a 5V input. Some of the newer PC motherboards, especially laptops, only outputs about 3.3V and this can be problem for some drives. To address this, the easiest way is to either fit a PCI LPT-port card in the PC or to get one of the various breakout boards available that supports boosting the voltage to 5V.

OK, to put the tuning to the test lets go to the MDI screen and make the machine do some moves. Click the MDI button or press ALT-2 and then click the MDI input or hit enter to enable it for input. Now type something like G0 X10 Y10 Z10 and press enter. The machine should move all three axis from its current position to X10 Y10 Z10. The actual values used for this test depends of the size of your machine but you'll get the idea. Try a couple of different moves back and forth to make sure the motor tuning is OK. If one or more axis stalls or loses steps the tuning needs to be tweaked a bit more.

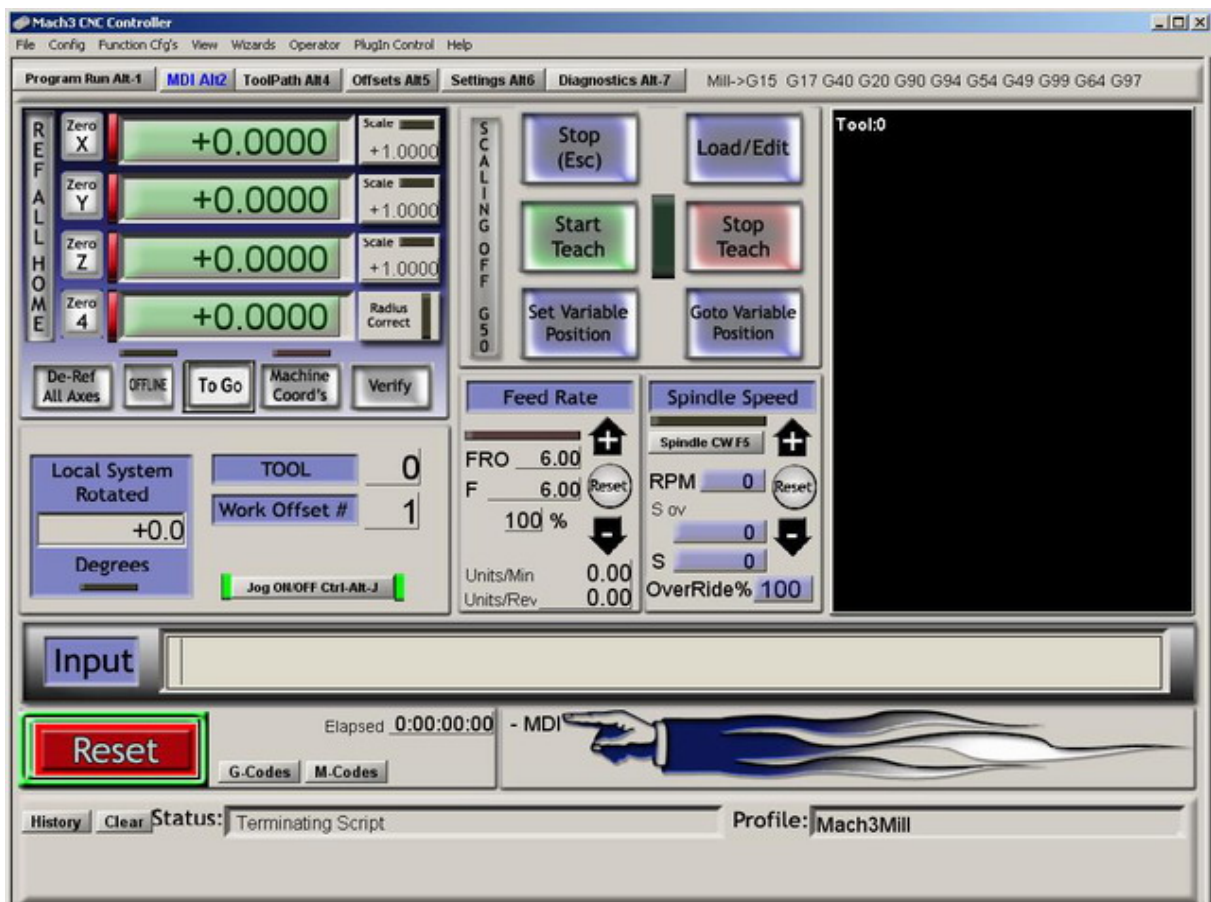


Figure 7: The MDI-screen.

Making sure the machine and DRO's moves in harmony.

On a standard three axis milling machine the X axis moves from left to right, the Y axis moves towards and away from you and the Z moves up and down. In the following section all movement is considered to be of the actual tool – that is the tool moves to the right even though it's actually the table that moves to the left.

Go back to the Program Run screen by pressing the *Program Run* button or hitting ALT-1. Make sure the LED around the Jog ON/OFF button is green, if not press that button to enable the machine to be jogged. (The button is located in the lower middle section of the screen).

Now hit the TAB-key on the keyboard to bring out the Jog-control screen. It should come out from the right side of your screen. If you can't see all the controls just grab the grey line and drag it out as far as you wish. Now we need to tell Mach3 at what speed we want it to jog and we do that by entering a *jog rate* in the DRO labelled *Slow Jog Rate* on the jog control screen. Let's start slow with something like 10%. Click the DRO, type in the value of your choice and hit enter.

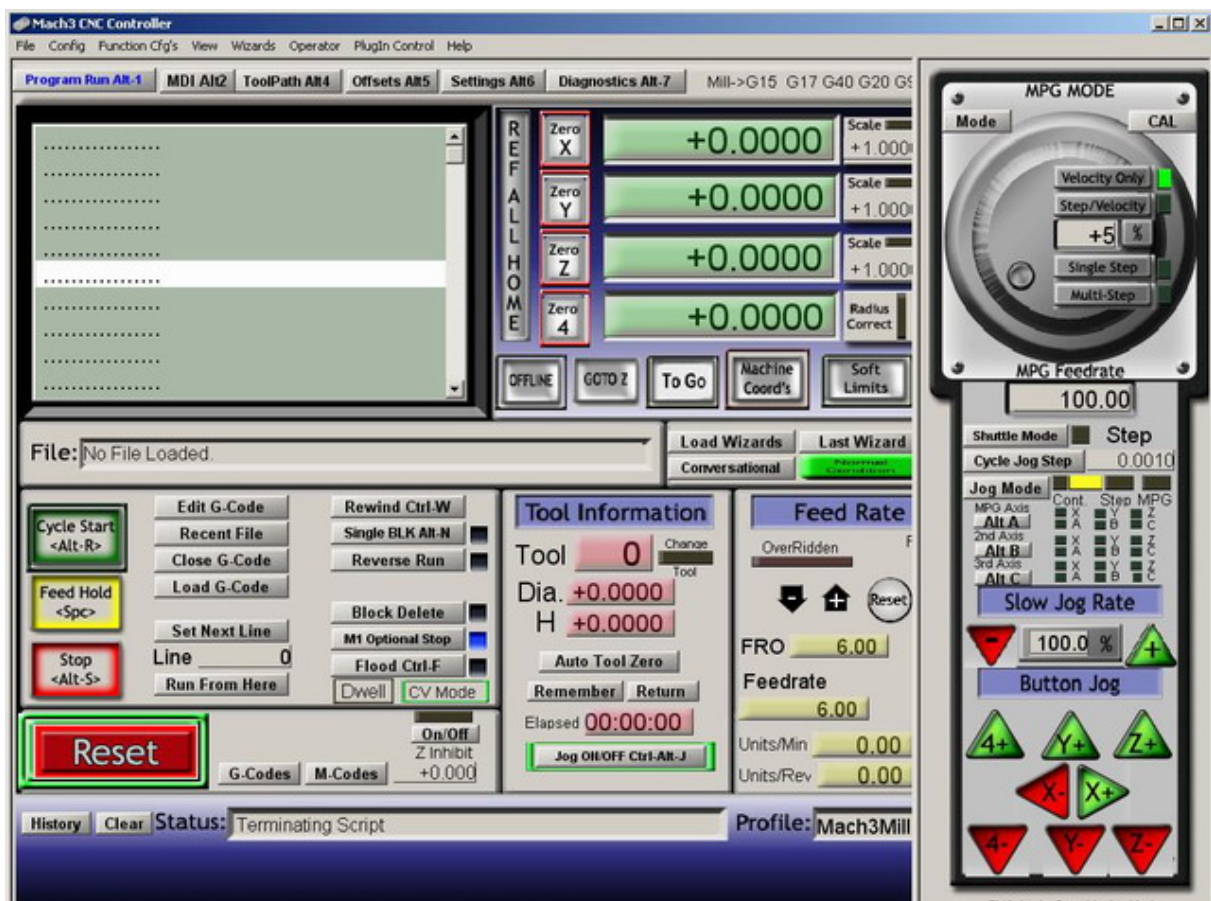


Figure 8: The Program Run screen, with jog-panel fly-out.

Now press the right arrow key on the keyboard. If all is well the X-axis DRO should count up and the tool on the machine should move in the positive direction (to the right) (table moves from right to left). If the DRO counts up but the tool moves in negative direction we need to reverse the motor direction. We do this by reversing the *Dir Low Active* setting in *Ports and Pins, Motor Outputs*. If the tool moves to the right but the DRO counts down we have to change the hotkey for the axis jog. This is done by selecting *System Hotkeys* in the *Config* Menu. When the X-axis is OK repeat the steps for the Y and Z axis.

The Y axis DRO should count up and tool should move away from you (table moving towards you when standing in front of the machine) when pressing the up arrow key.

The Z axis is jogged with PageUp and PageDown buttons. The DRO should count up and the tool should move up when pressing the PageUp button.

So, that was the most fundamental parts of setting up the Mach3 CNC control application. In the following sections we will go thru how to set up the charge pump, spindle, coolant, home switches and the software over travel limit.

The charge pump function.

When booting up the computer the output pins of the parallel port may be in an unknown state. If for example we have our spindle motor wired thru a relay that is activated when there's 5V present on pin 8 of the port the spindle may start at any time while Mach3 is not in control and this is a very dangerous condition. To prevent this we can use the charge pump function in Mach3 along with a bit of hardware. We will not discuss building the hardware part here but schematics are available from ArtSofts web site. (www.machsupport.com)

To setup the charge pump function in Mach3 we, once again, open the *Ports and Pins* in the *Config* menu and then select the *Output Signals* tab.

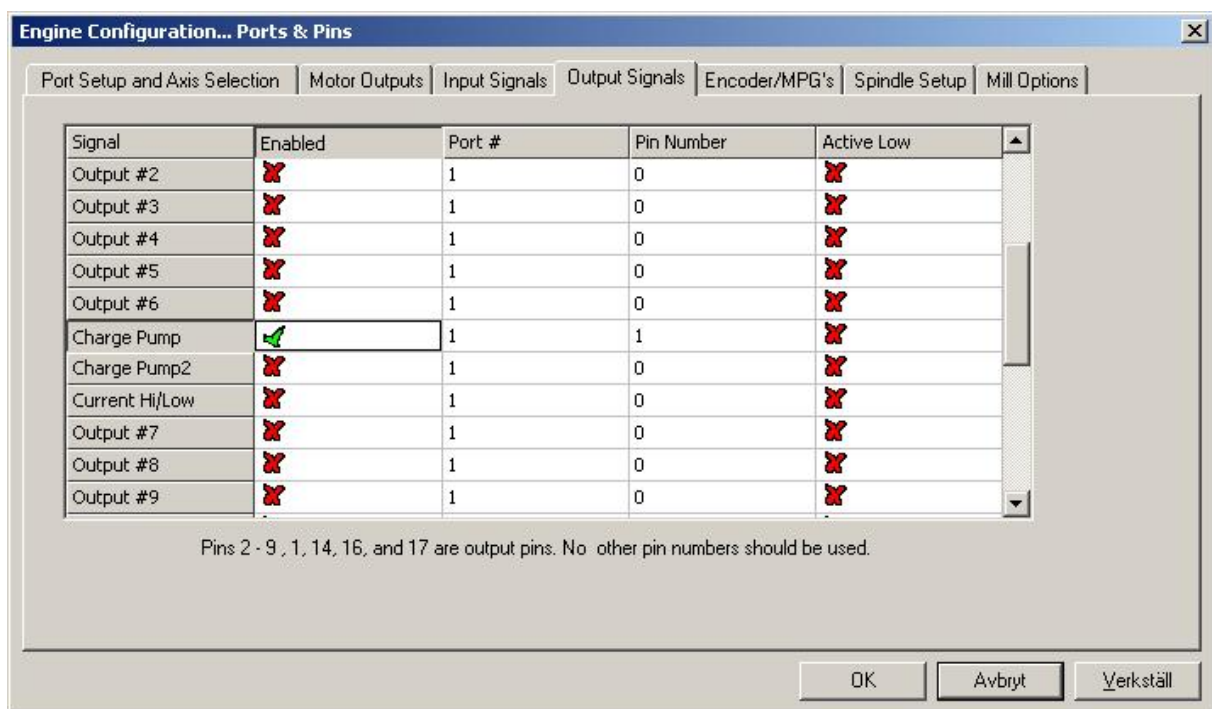


Figure 9: The Output Signals.

About halfway down there's a signal called *Charge Pump*. Put a 1 in the *Port #* and the *Pin #* to which the circuit is connected, in this example pin #1. Then make sure the function is enabled (a green checkmark in the Enabled column). Check that no other output signals are enabled. Now, when ever Mach3 is running AND is not in E-stop mode there will be 12.5kHz square wave on pin 1 of the parallel port. If you want the charge pump to run all the time Mach3 is running, even in E-stop mode go to the *Config* Menu and select *General Config* and enable the *Charge Pump on in E-stop* option.

Spindle outputs.

Now that we have a way to prevent the spindle from starting when Mach3 is not in control we'll go on and setup a relay to start and stop the spindle from within Mach3. As before, we will not discuss the actual hardware interface but it is assumed it is wired thru the charge pump circuit and is activated by a logic low level on the pin of the port. There's also a lot of different ways that Mach3 can control the spindle, the simplest being plain ON/OFF control and that's what we are going to do here.

You may have noticed the spindle as one of the axis in the *Motor Outputs* tab under *Ports and pins* when we set up the X, Y and Z axis. Those settings, however, is when we want Mach3 to control the speed of the spindle by using either PWM or step and direction output and that's not what we are doing here. We are aiming for a simple ON/OFF control.

So the first thing to do is to enable an output. Let's go to *Ports and Pins* and select the *Output Signals*. In this case we are going to use Output #1 as our spindle relay output and we are mapping that output to pin 16 of the parallel port by simply putting 1 in the Port# and 16 in the Pin# number column and enabling the output. (Green checkmark in the Enabled column). Then, finally, check the active low setting, hit apply and go to the *Spindle Setup* tab.

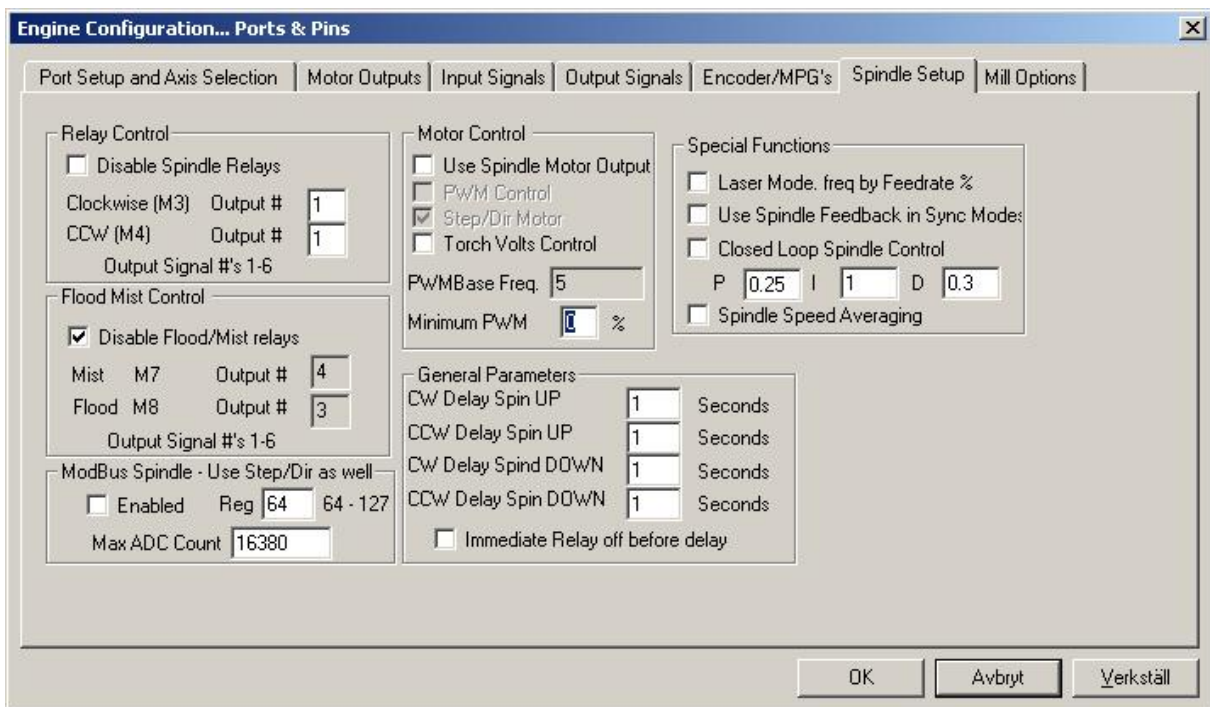


Figure 10: The Spindle Setup dialog.

In the top left corner, uncheck the *Disable Spindle Relays* and then enter a 1 in both the *Clockwise* and *CCW* boxes. Remember that we mapped Output #1 to pin 16 where we have our relay connected. This tells Mach3 that whenever we want to run the spindle, clockwise or counter clockwise it activates Output #1. It is assumed in this tutorial that the actual reversing of the spindle, if at all possible, is done either mechanically or electrically on the machine itself and is not controlled by Mach3. You could of course enable a second output and use that to run the spindle in the counter clockwise direction if you want.

If you have a high speed spindle or one that for other reasons takes time to come up to speed when turned on or down in speed when it's turned off this can be set under *General Parameters*. Enter the time it takes for the spindle to reach the maximum speed. This will prevent Mach3 to start running a program until the spindle is up to speed. Click OK to save the settings.

Now go to the *Diagnostics Screen* in Mach3 and make sure that the control is out of E-Stop mode then click the *Spindle Toggle* button. If Mach3 is setup correctly you should see the red led next to Output #1 flashing and if your hardware is correct the spindle should start. Pressing the button again will make the spindle stop. Yeah, we have spindle control !

Coolant.

The control of the coolant is done in the same way as the spindle so we'll only go thru this briefly. Again, the hardware is assumed to be connected thru the charge pump and to be activated by a logic low level on the port pin. We will only use flood coolant and it will be controlled by Output #2 thru pin 17 on the parallel port.

First enable Output #2, set the Port# to 1 and the pin# to 17, the output should be active low. Then go the *Spindle Setup* tab again en make sure the *Disable Coolant Relays* is not checked and set the outputs for both *Mist* and *Flood* to #2.

Check your setting by pressing the *Flood Toggle* button on the *Diagnostics* screen. The led next to Output #2 should start to flash and the relay should activate.

Home switches.

As you probably have noticed Mach3 is very flexible in its use of inputs and outputs. It is possible to combine home and limit switches in many different ways but in this section of the tutorial we will go thru how to set up one home switch for each axis, wired in series. As usual we will not discuss the hardware and actual interface but it is assumed that the home switches are normally closed, wired in series between ground and pin 10 on the parallel port.

To set up the home switches we need to enable the home inputs. In the *Config* menu select *Ports and Pins* and go to the *Input Signals* tab. Enable the inputs *X Home*, *Y Home* and *Z Home* and set the port # to 1 and pin # to 10 and check the *Active Low* setting on all three.

Here you can see the power of Mach3's I/O capabilities. Even though we have three axis we wire them in series and connect them to one input. When Mach3 homes the machine it will do it one axis at the time and when the switch is hit it reverses the direction of the motor until the switched is cleared and then it does the next axis. It is even possible to use same switch as limit switch – Mach3 will know that the switch is a home switch when homing and from there on it's considered to be a limit switch.

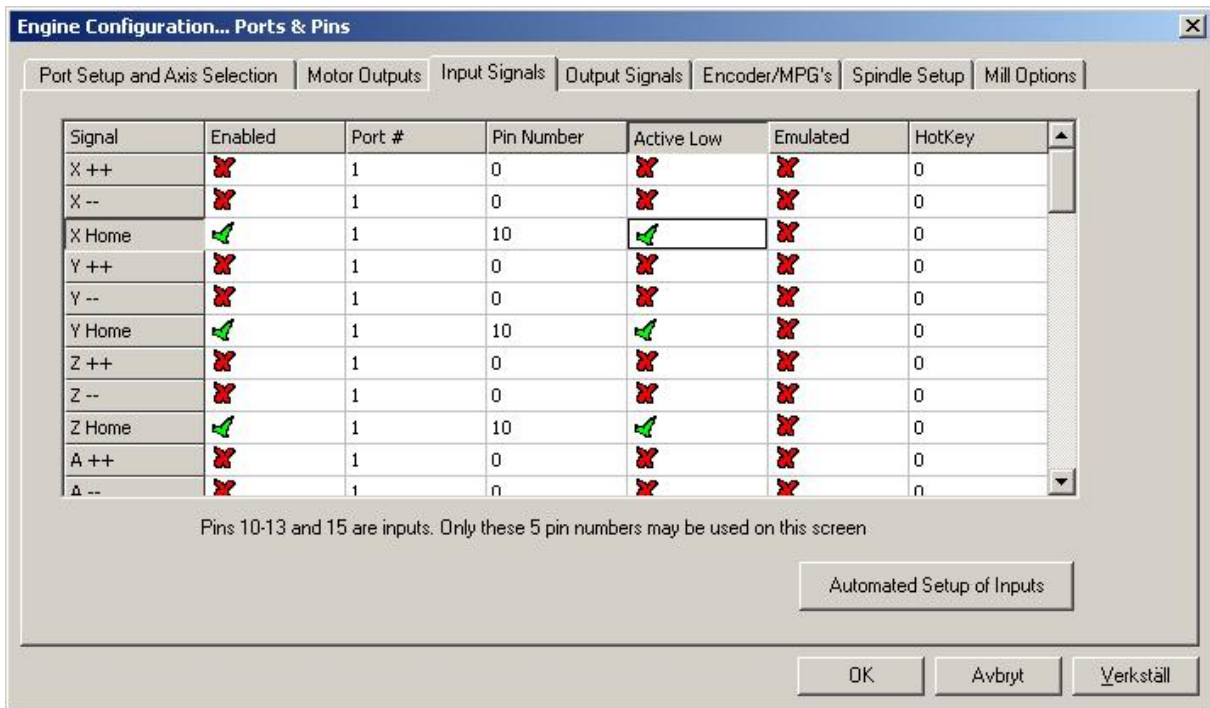


Figure 11: Setting up the home switches.

To test the settings and hardware go to the *Diagnostics* screen and observe the LED's next to M1 Home, M2 Home and M3 Home while pushing either one of the switches. The LED's (all three of them) should light up when pressing any one of the three switches.

Homing parameters.

Now it's time to tell Mach3 where we have the home switches mounted. The most common location for the X- and Y-axis is probably at the most negative position of travel. The Z-axis is commonly homed to its most positive position of travel. Please note, though that this position is still called machine zero for the Z-axis thus making all moves in the negative direction. This is how we will set it up here.

To setup the homing we need to select *Home/Limits* in the *Config* menu. When we do that we will see the dialog in figure 11.



Figure 12: The Home and Software limits dialog.

Since we want to home the X- and Y-axis in the negative direction we enable the *Home Neg* option for X and Y and leave Z unchecked. If we enable the *Auto Zero* option the Machine coordinates will automatically be set to zero when the axis is homed. Since we have our switches at the extreme ends of our travel we want to have this enabled. If the switch is placed at any other position than at the extreme end the distance can be entered in the *Home Off* column. This tells Mach3 to set the Machine Coordinate for the axis to this value instead of zero – good thing to have. The *Speed %* setting is at what percentage of the motors full speed the axis is homed and may or may not need adjustment depending on your particular machine. Generally, the slower it goes the more accurate it gets. You may need to experiment a bit.

Homing the machine.

With the home switches wired and tested it's time to home the machine. With the control out of E-Stop mode, go to the *Program Run* screen and with one hand on the E-stop button press the *Ref All – Home* button. The machine will now start to move Z up until it hits the switch then it will reverse the axis until it clears it. Then it does the same with the Y-axis and finally with the X-axis. When each axis is homed the DRO's should read 0.00 or whatever value you put in as the *Home Off* value. If any axis starts to move in the wrong direction press the E-Stop button and go to the *Home/Limits* setup and change the *Home Neg* setting.

Software limits.

Mach3 has the capability to prevent you from damaging your machine by constantly monitoring its position. If you or a G-code program tries to move it outside its working envelope it won't go there but instead present you with an error message. To set this up we go back to *Home/Limits* dialog in figure 12.

As we have our home switches located at extreme ends of travel in the negative direction for the X- and Y-axis we need to set the *Soft Min* value to 0. Doing so will prevent the machine from going beyond 0 in the negative direction. In the *Soft Max* we put whatever amount of travel we have available. In this case it's 300mm in X and 100mm in Y. Since we home Z in the positive direction but still call the home position 0 we need to put 0 as the *Soft Max* value.

The *Soft Min* value is again what amount of travel we have available, 100mm in this case so -100 (note the negative value) goes into the *Soft Min* column for the Z-axis.

The *Slow Zone* setting is the amount travel there's left in the axis when the machine automatically starts to slow down when you are approaching a limit. For example, with a value of 3.00 in the *Slow Zone*, if we jog the X-axis at high speed towards the negative limit, which is at 0.00 the machine will start to slow down when it passes 3.00. This prevents the machine from losing its home position due a too sudden stop of the motors. However, this automatic braking is only available when jogging and not when running a G-code program.

To test the software limits the first thing we need to do is to home the machine. Then we need to enable the software limits by pressing the button labelled *Soft Limits* on the Program run screen. When the LED next to the button is green the limits are turned on and we test the limits by trying to jog past them. Hopefully you want be able to do that and you will get a message at the bottom of the screen saying: *Soft Limits System Movement Aborted*.

Final words.

By now we have set up and tuned the main axis of the machine, we have made spindle and coolant control possible from Mach3 in a safe way due the E-stop and charge pump circuit. We have established a way to home the machine by the means of home switches and a way to prevent it from damaging itself with help from the software limits. I think this more or less covers the basic setup of a three axis milling machine. However.....

There is a LOT more that can be done with the Mach3 CNC control application . For example, spindle speed control, laser scanning of 3D objects, probing, tool changers, plasma cutting, tangential knife cutting and so on. The list can be made very long.

Welcome to the world of CNC and Mach3, this is just the beginning!