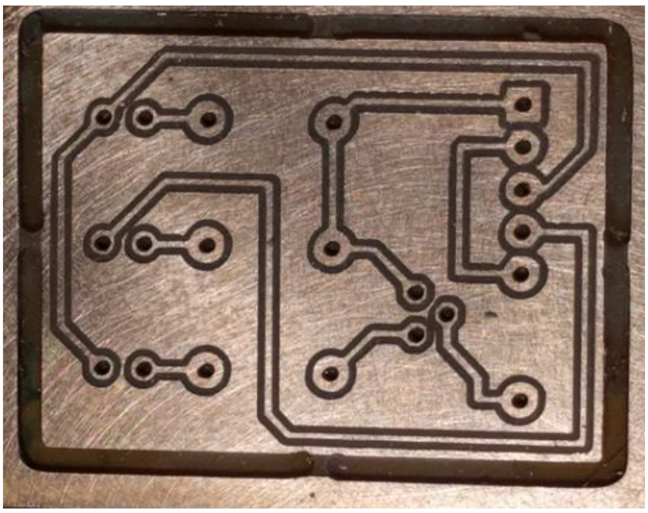
ChiliPeppr Instructions

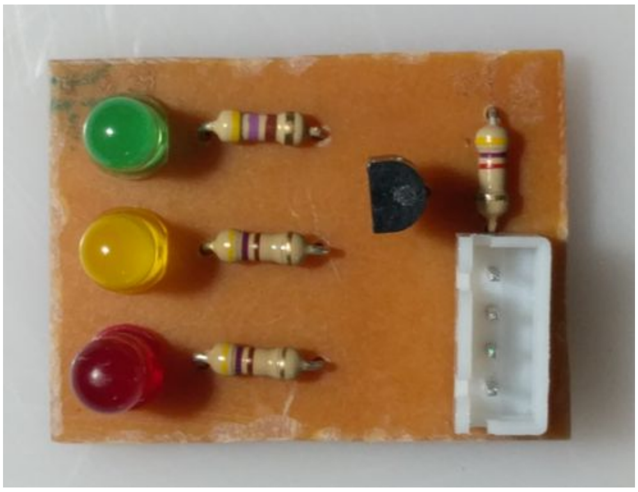
Aim: To engrave a single sided PCB from an Eagle design in Chilipeppr.

Acknowledgement: This document has been prepared to inform people how I came to have success with Eagle, Chilipeppr, GRBL and Arduino. These technologies have many authors and this document is intended to give full recognition of incredible amount of intellectual effort of the people who have contributed to the production of these applications.

Chilipeppr: http://chilipeppr.com/
Eagle: https://www.autodesk.com/products/eagle/overview (using the limited free edition)
GRBL: https://github.com/grbl/grbl/wiki
Serial Port JSON Server: https://github.com/johnlauer/serial-port-json-server
Arduino: https://www.arduino.cc/
Protoneer: http://www.protoneer.com
Machinery Suppliers: eBay search “engraving CNC machines 2418”

Preparing the Design

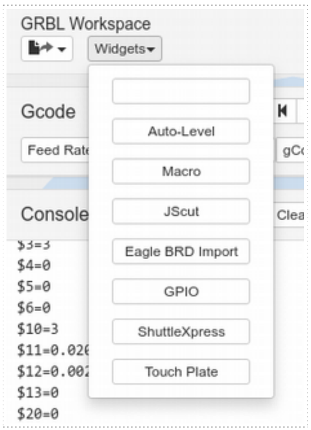




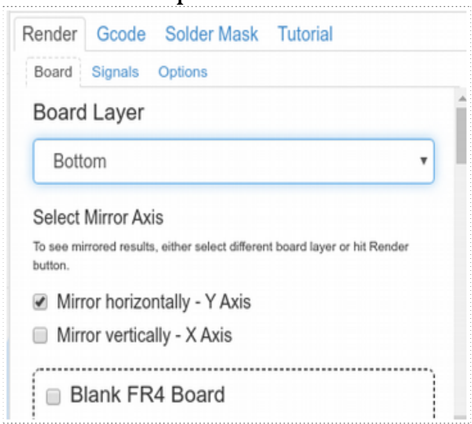
1. **ChiliPeppr** is written in Javascript so we start up ChiliPeppr(CP) in a web page.
The following refers to *jpadie work space* - http://chilipeppr.com/jpadie
Make sure you register and login as a user.
Use ChiliPeppr's *Serial Port JSON Server* See Appendice I for Ubuntu Launcher details
Attach the server and select a port that connects USB with the GRBL system.
The following applies to GRBL 1.1h on an Arduino Uno board+ Protoneer V3.0

2. Use drop down menu and choose **Eagle BRD Import** widget
This should clear the previous PCB, ready for import of the new.
Use your folder explorer, and open an Eagle Project folder.
Drag and Drop your Eagle <filename>.brd file onto the CP GUI.
The file should appear previewed in the center GUI panel.

ChiliPeppr has a drop down menu in the top lefthand corner



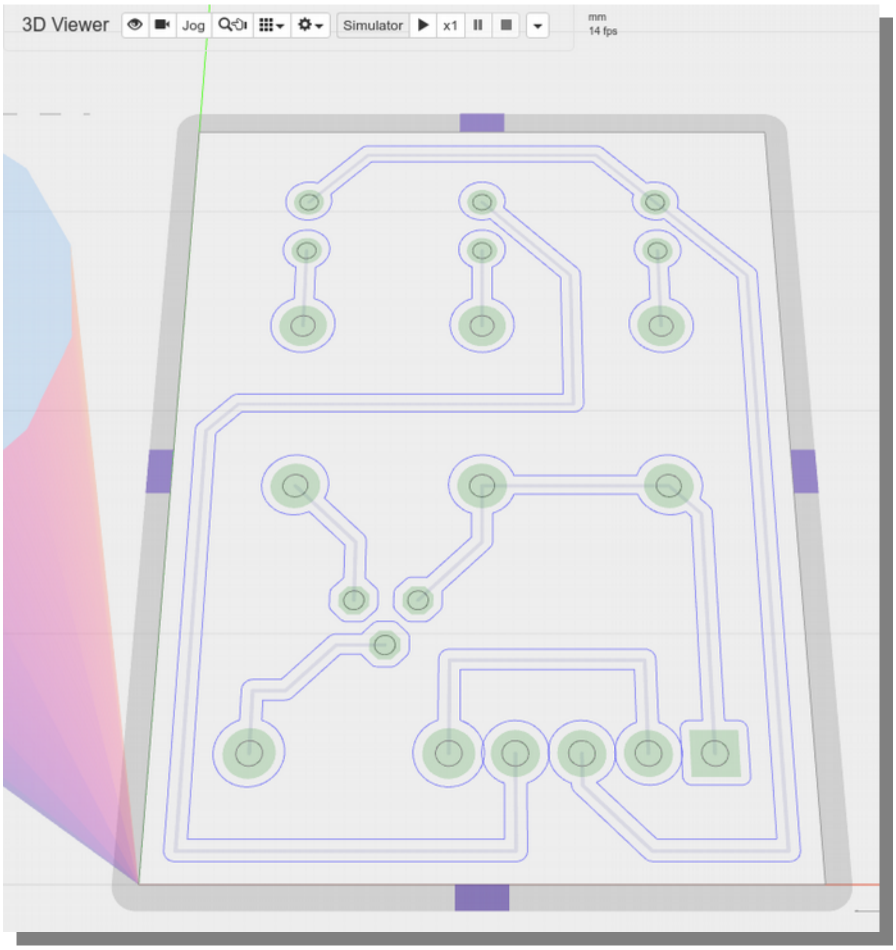
Once imported, use the tabs and menus to navigate through the options.
Choose **Board Layer** eg **Bottom** (ie Through hole)
Choose **Mirror** eg **Mirror Horizontally Y axis**
You can use the Preview button to refresh the display.
Choose other parameters such board thickness, drill hole sizes and milling options.



a. If you want to drill all your holes the same size to begin with then make the "Maximum hole diameter to drill" say 1.5mm And the milling tool 2mm. If Tool Change Pauses are occur later on you can keep drilling, until the milling is required.
b. Some parts have slightly different hole sizes so they can trigger tool changes on their own.
c. Depending on your milling capacity to cut at high speed, reduce the depth increment to 0.2 for example and make more passes. A 2mm bit at 0.2mm cut is ok for a lightweight system..

To begin with, consider these minimal changes in these tabs:
Render – choose bottom copper and y-axis mirror, rest is default.
Gcode – Traces – no change, default.
Gcode – Drilling – changed board to depth to 1.4 as I was using phenolic PCB
Gcode – Drilling – changed Maximum Hole to drill to 1.5mm
Gcode – Milling – changed Diameter of tool for cutting to 2mm
Gcode – Milling – chnged Minimum Wire Width changed to 1.5mm
Gcode – Milling – Depth of Cut Out, changed to 1.4mm
Gcode – Milling – changed Step down for Cutting Dimensions to 0.2mm (not so aggressive).
Gcode – Tabs – Tab Width 2mm (from 2mm mill)
Gcode – Gcode, default.

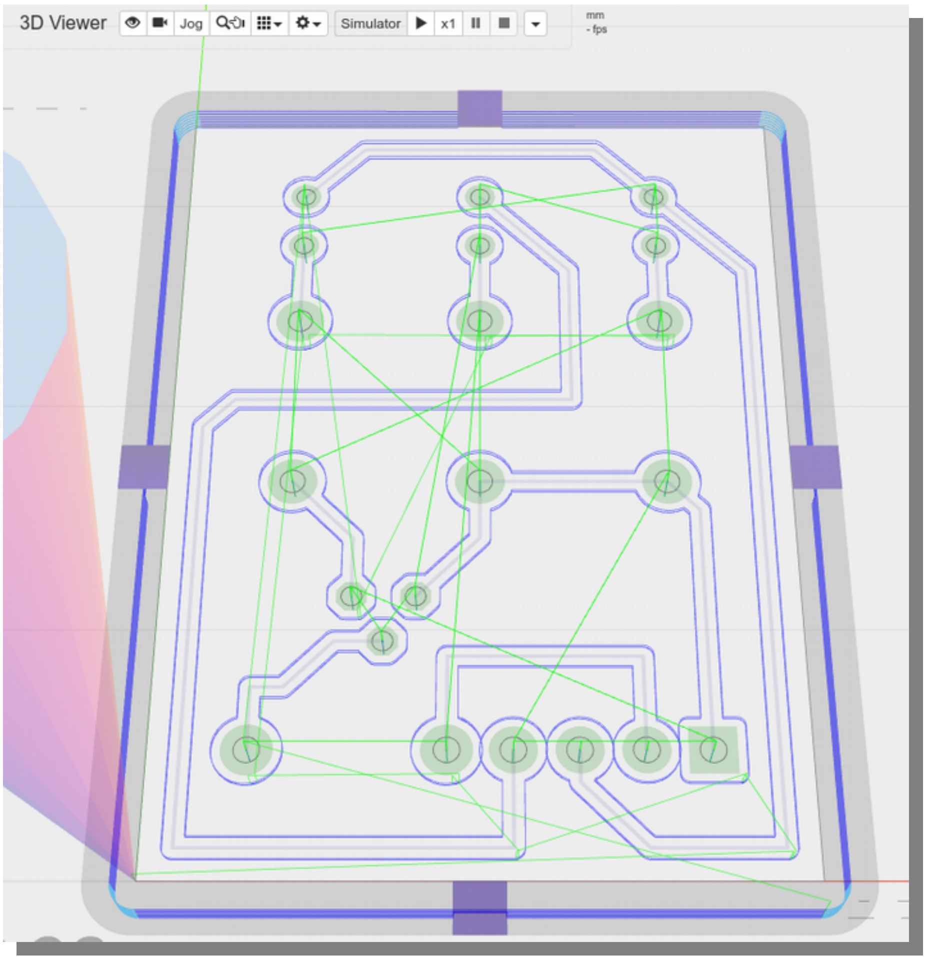
The render in the preview will look now something like this-



The outlines of the milling required is shown, but not the exact movements to achieve the end result.

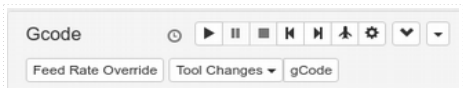
3. "Send the PCB design Gcode to the Gcode Work Space"
Use the Folded Paper Plane Icon .
Blue lines should now be present for "milling", plus
Light Green lines indicating drilling holes.
The milling for the cutout of the PCB should also be seen with anchoring tabs as well.
You can alter widget settings by making changes, then refreshing and repeating resend .

The preview should then show milling (engraving and milling bits) and drilling pathways
You cannot return to this widget's options, without starting all over again, so check thoroughly.

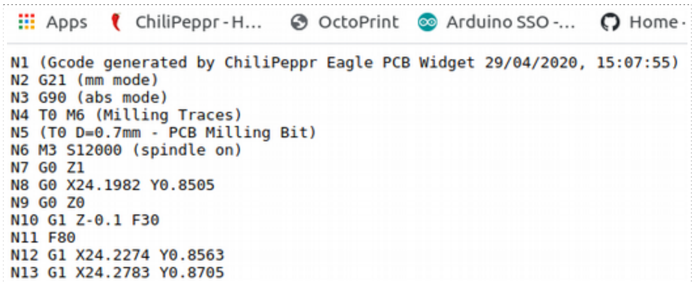


4. The Eagle BRD import widget can be removed via drop down menu. This will free up space on the GUI.

5. Using “gCode button” on the Gcode Widget should show clean Gcode,



based on your Eagle design and begin with something like this following header
Gcode generated by ChiliPeppr Eagle PCB Widget 26/04/2020, 16:04:19




```
N1 (Gcode generated by ChiliPeppr Eagle PCB Widget 29/04/2020, 15:07:55)
N2 G21 (mm mode)
N3 G90 (abs mode)
N4 T0 M6 (Milling Traces)
N5 (T0 D=0.7mm - PCB Milling Bit)
N6 M3 S12000 (spindle on)
N7 G0 Z1
N8 G0 X24.1982 Y0.8505
N9 G0 Z0
N10 G1 Z-0.1 F30
N11 F80
N12 G1 X24.2274 Y0.8563
N13 G1 X24.2783 Y0.8705
```

Check the time/date stamp the is right for your current job, so if you have updated the imported Gcode you know it is the latest version.

Preparing the CNC Machine for Milling

6. Setting the user’s [0,0,0] origin in the build space

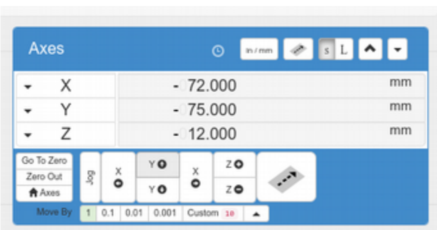
Make sure the electrical contact of the probe (GRBL, Arduino A5, Appendix 2) to the PCB is reliable.
Use a 330 (to 1k) ohm pull-up resistor to make the probe current stronger and "wet" better. Add a small red LED in series to indicate a successful circuit. Use a red LED for lowest forward voltage and logic low.
Use the machine's chassis or tool as the GND signal, and connect the PCB to A5 (was GRBL Spindle Control, don't use Min-Z Axis as mentioned for Tiny-G).

Attach PCB with strips of thin double sided tape (not foam) on the mill bed. Add a sacrifice board eg 3mm high density particle board. It is not necessary to cover the whole of the board, 10mm strips 20mm apart is sufficient and much easier to lift when finished. Position the board where lower the lefthand corner is anticipated to be [0,0,0] on the build board.

Clip on the Probe lead (from A5) to a soldered wire on the corner of the blank PCB.
Use a thick lead for a wire from CNC Shield GND to chassis GND, to reduce spurious signals or ground loops causing false triggers. If a good earth does not travel through to the tool tip, then use a flying lead with a crocodile clip to directly clip the tool to chassis GND. Do not connect the tool tip to A5 and use the PCB as GND.

7. Homing the XYZ axis (X to RHS, Y to the back of stage, Z to top).
Press reset on Arduino/GRBL shield.
Use $X to unlock the access to the GRBL from the command line.
Use $H to move the carriage to home position.
Depending on Homing Pull Off, it will show [-2,-2,-2]
Our work space on the stage relative to the homed origin is all negative.

8. Setting up the relative [0,0,0] position for the job to refer to.
Use either Gcode commands (if there is a known destination point) or use the Jog Widget buttons to position the probe (ie the cutter) over your chosen [0,0,0] point (ie usually the nearest left hand corner of the blank PCB). The jog values at this stage will be negative distances from your home origin (ie the absolute machine coordinates) Work on postioning in the horizontal Xand Y plane first and after that, lower the Z-Axis eg



So [-72,-75,Z] will become [0,0,Z],so now we need to find Z, -12mm is about 15mm too high.

The relative Z-Axis zero can be found by jogging the probe (ie the tool) down onto the PCB gently, using 0.001 if necessary, until the Red LED lights up. Back off the probe until the LED goes off.

XYZ are now at your chosen position to be declared [0,0,0].

Click on "Zero Out" button in Jog panel. This sets all three axis to zero position where the probe rests..

9.Making sure the project is progressing OK

In the Gcode widget click on the downwards pointing chevron icon  to inspect workspace Gcode so far. NB: There are no auto leveling Z corrections for any of the milling Gcode steps at this stage (there will of course be some Z movements to begin new paths etc).

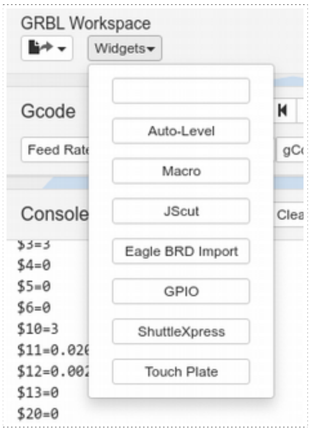
10. How to get an even cut.

If your board is small, with has minimal warpage edge to edge then it is possible to successfully cut the PCB now. (Skip to 15)

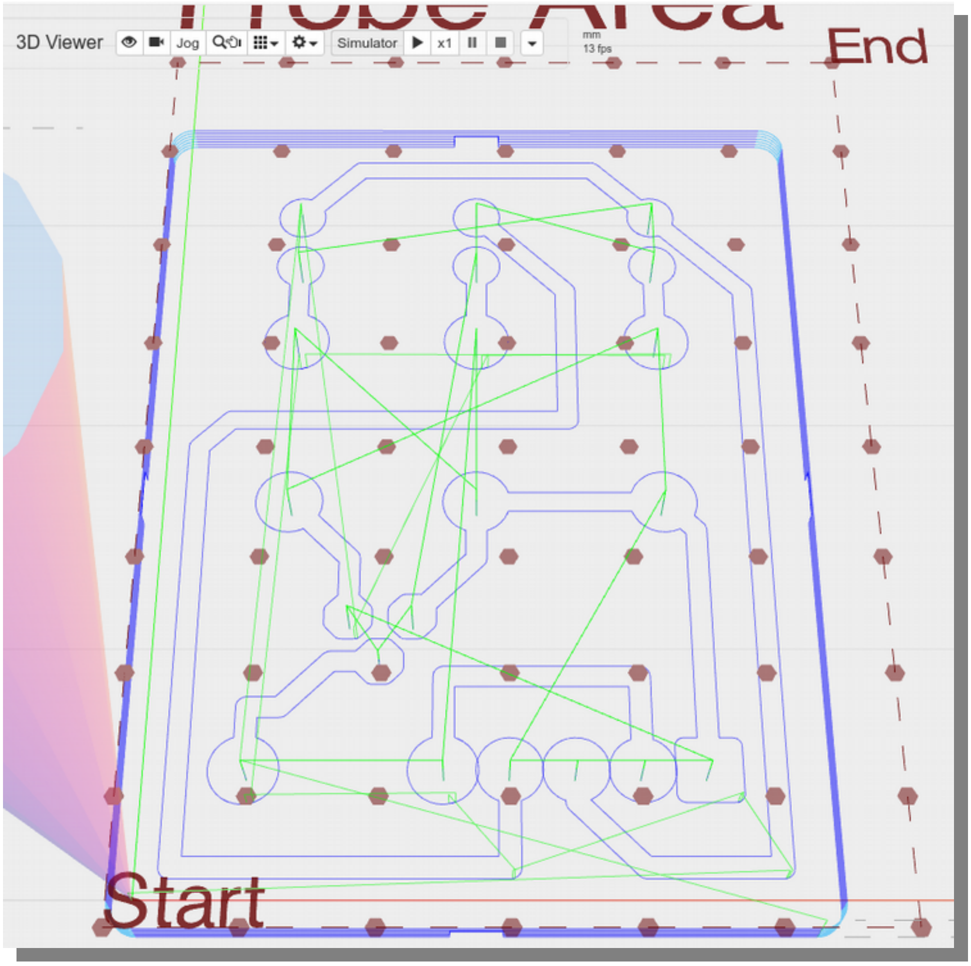
If a large PCB is being attempted or the board is known to have excess warpage, then Auto Levelling is advised.

Preparing for Auto Leveling

11.Use GRBL-Workspace for drop down menu, and load the Auto-Level Widget

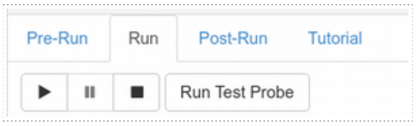


It should auto fill your PCB space with 5x5mm probe points.

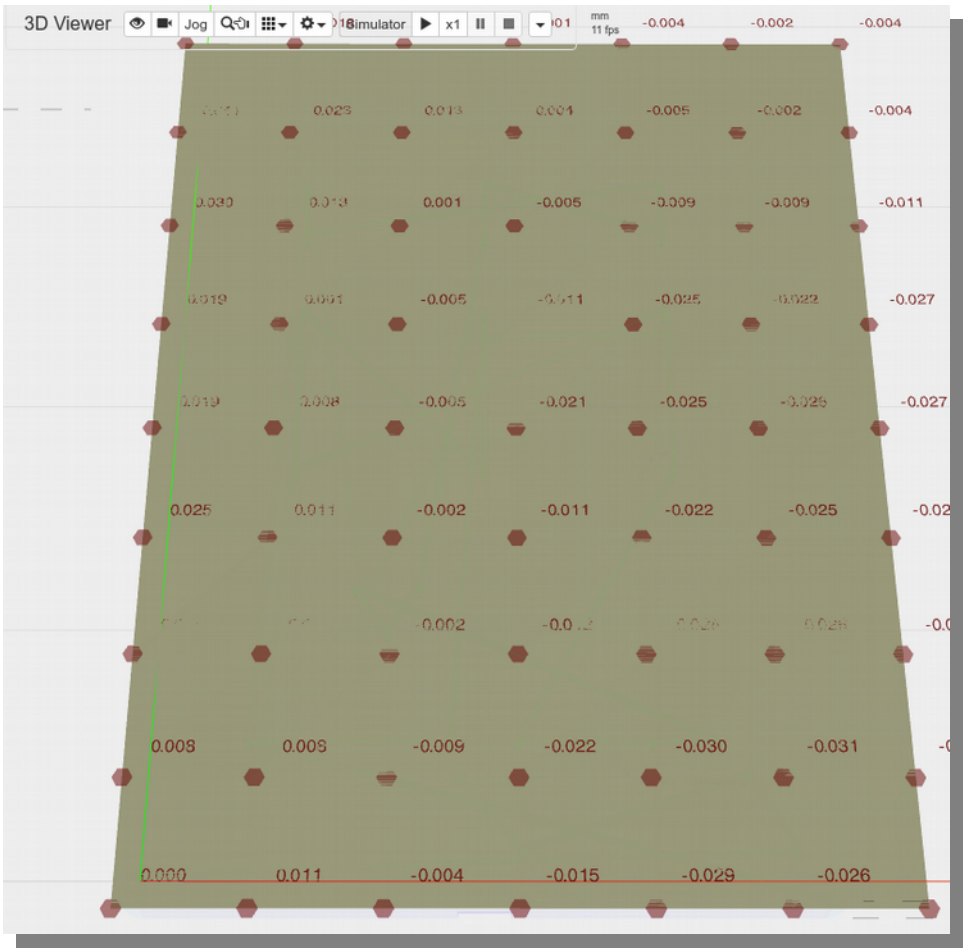


The spacing can be increased, to decrease time taken to probe, if the board is fairly flat, and super precision not expected.
If you have used the probe LED to zero out Z, then the defaults probing settings will work as the Z should be within 0.01mm, or closer to the PCB surface. This may require a little practice!
Use Auto-Level Pre-Run tab to alter these if required. It may require resetting up again.

12.Use Auto Level Run tab and use the “Run Test Probe” if you have any doubts at all about your probe to PCB connectivity/sensitivity.

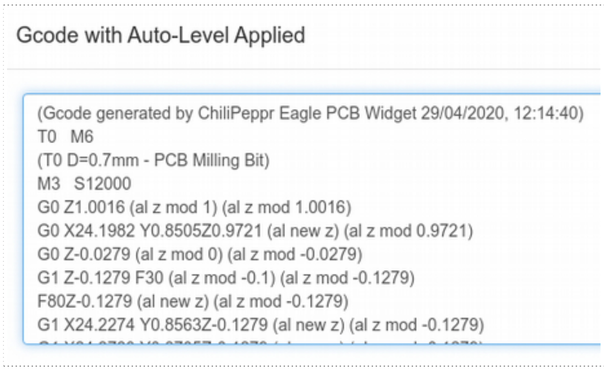


NB Many tools are destined to be broken at this stage, so be careful.
Otherwise select RUN icon ▶ to start the probe moving around the points.
Again review your readiness to the probing and if satisfied click on "Let's Get Probing".
When it has finished probing all the points, they will be shown on the GUI.

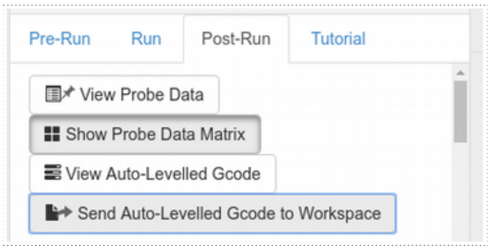


Check the reulting probe heights are reasonable, and that no erroneous bit breaking level changes have been introduced by bad probing contacts. Look for any sudden changes in Z-values that could indicate a mis-probe. If the board is reasonably flat, gradual changes from point to point should be seen. The board above has had the amplification of the elevation overlay graphic set at x1 to allow the Z-heights to be seen better. 50X amplification is the default.

13.Use the “Auto Level Post-Run” tab to inspect the leveling data and the Gcode data. There should be many changes to all Z levels added to Gcode commands. This is the auto levelling at work, note the ()’s. eg G0 X6.4182 Y0.8505Z1.0006 (al new z) It would be good to see another comment line added indicating the Auto Leveling had been applied, with a date stamp.

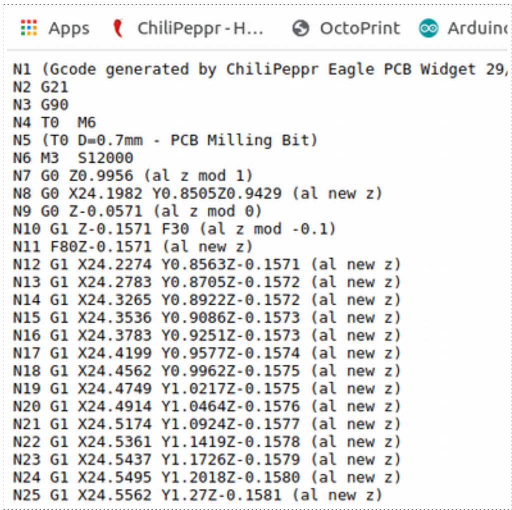


This modified Gcode is only in the Leveling Widget at this stage, so select the "Send Auto-Leveled Code to Workspace" to update the main Gcode widget with the leveled Gcode.



Remember this will also counteract beds that are not perfectly parallel with the X-Axis gantry.

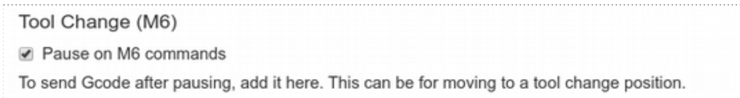
14.Check in the Gcode widget that the commands have now been transferred, and have altered Z height adjustments appended. Eg eg N12 G1 X6.4474 Y0.8563Z-0.0994 (al new z)



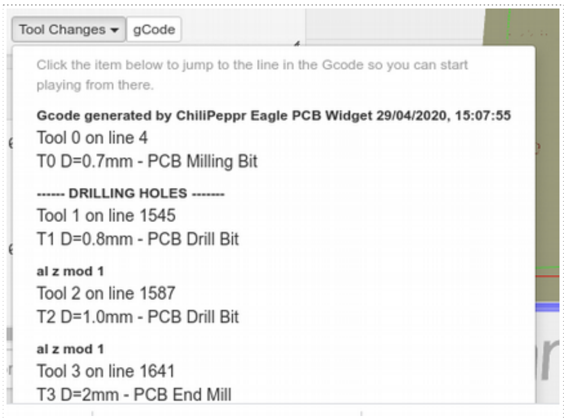
Close the “Auto Level” widget with the GRBL Workspace drop down menu. If you cannot find the Gcode in the Gcode widget, you will have to close down the browser and start CP all over again.

Milling the Board

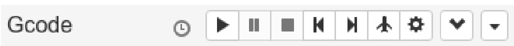
15.If you have different tools, eg engraving bit for copper tracks, drill bit for component holes and a milling bit to cut out the PCB, then click on the gear symbol and it will pop up “Gcode Sender Options”, then select "Pause on M6 commands". If your Eagle import has components requiring different size holes there will now be pauses in the milling, and you will be allowed to make tool changes. Scan down the page and find -



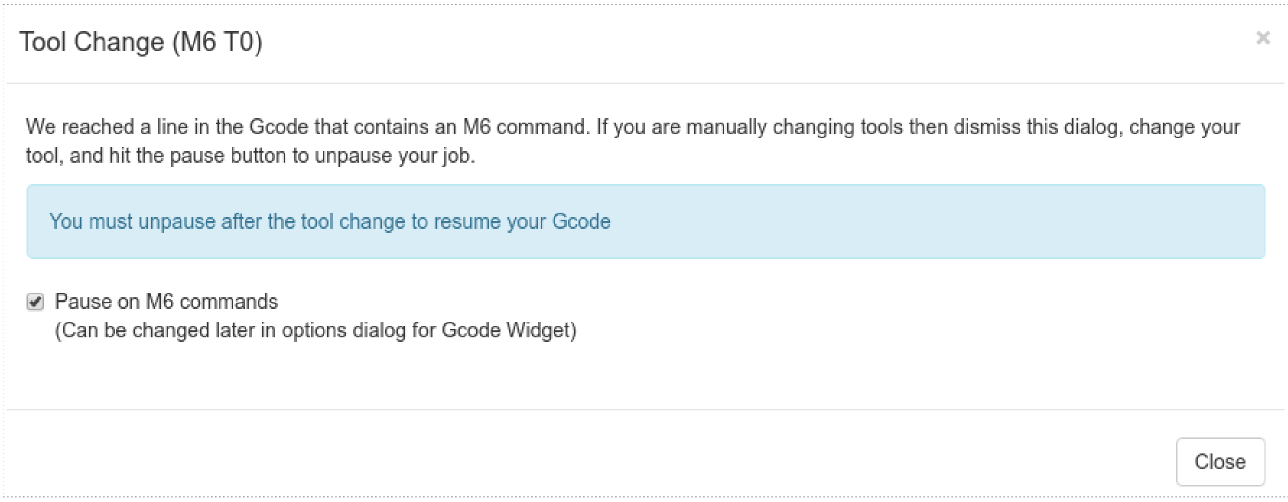
If you click on the Gcode widget "Tool Changes" button it will provide a list of tools, what they are and which line they begin on. They are numbered from 0 onwards. Tool 0 is the engraver bit, the rest that follow are either drilling or milling. This graphic below indicates 0.7mm when in fact the engraving bit is more like 0.1mm and at 30 degrees arc. The 0.8mm and 1.0mm drill were both drilled with a 0.9mm drill for both of them. They can be enlarged easy later on. If you have a big board with many holes it will be worth getting the correct drills in stock and doing right, once. Even a fraction of a millimetre in drill size can effect the durability of a pad to hot soldering iron. Smaller holes means more meat on the pad.



16.Run the Gcode in the Gcode widget using the "play" button.



The message below will pop up first as it assumes you will be changing the engraving bit in at this time, whereas most people will have it in to begin with. It will pop up again for each subsequent tool change.



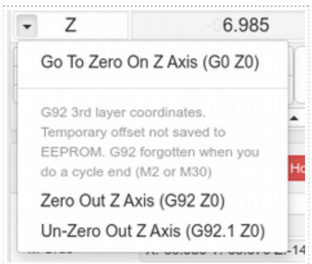
NB: The Z altered comments (in brackets) should appear in a little pop-up alert on the fly so to speak, so if you see these go past the Auto Level is working.

17.Changing tools

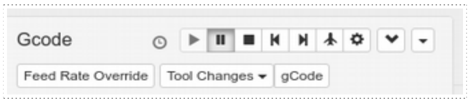
When the mill hits a tool change it pauses and puts the big notice shown above on the GUI. The spindle motor should stop now as well. After closing the big prompt you can take out one tool and put in the next one. (If you intend to use the same drill for all the holes then you can just leave that bit in).

Use only the jog buttons to lift the Z axis up high enough to facilitate swapping the bit over. Don't raise the Z axis too high or you will hit the Z-max endstop, cause an Alarm:1 and have to reset, lose your positions and have to begin all over again.

Then use the jog buttons to place the tool over a part of the copper that is still connected to the probe. Lower the tool carefully, using 0.1mm decrements until the probe red LED lights. Then back off a click (or if you touch on 0.1mm movements, back off on in 0.01mm increments), and when the red LED goes out, use the Z in the Axes Commands to Zero out (ONLY) the Z axis. These should only be the drilling of holes or milling and will not require the finess of the engraving/copper cutting.



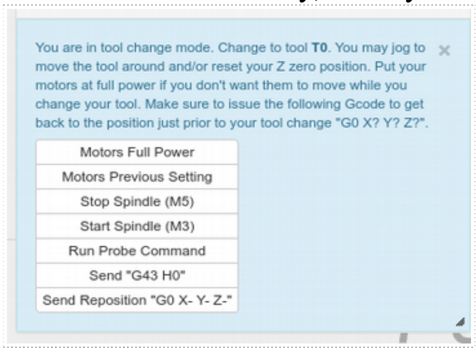
Then in the Gcode widget Un-Pause the Gcode sender and the motor should start and the milling resume.



This stage is a dangerous time for breaking bits so do things slowly and methodically. Human error is likely here. Repeat the tool change cycle, until all tools have been used, and the milling stops. Forgetting to zero the tool and the Z-axis will definitely cause problems ie A head crash!!!

The final tool change should use the milling bit to hollow out any big holes, eg 3mm mounting holes and to finally cut the PCB design from the rest of the blank board.

Hint: There is also a prompt that comes up during tool changes that give Gcode commands (well M codes!) that will allow you to stop and start the spindle and also to turn on the motors holding current if there is a chance the mill position might be moved during tool change over. If you have lead screws this is unlikely, but if you have belts is is good precaution.



18.Clean the board with a vacuum cleaner (use it while cutting preferably, PCB dust is dangerous). Then use some wet and dry abrasive paper (eg P280) and some isopropyl alcohol to debur any jagged edges, and give the board a final cleanup with an old toothbrush.

Spray the board with a very light coat of PCB laquer if the board is not going to be used immediately to stop tarnishing, and make soldering easier later on (the copper will not be oxidised).

Extra Observations

1. If the Recent Files are cleared the GUI will default to loading the CP logo file.
2. Turn off the widgets as you have used them to optimise your screen space and resist the urge to go back an modify the settings in them.
3. If you can't preview the Gcode with the right date and time, and the Z-height alterations, you may as well start again.
4. Minimise dust by flooding the work with isopropyl alcohol.

Appendix 1. Ubuntu Launcher for the serial interface

Appendix 2. Wiring up the GRBL protoneer interface.

Appendix 3. Milling build and configuration

Appendx 4. Annotated photo of some of the system

Appendix 1

Ubuntu Launcher for the Serial Interface

The serial-port-json-server-1.96 was installed into a local directory alongside other CNC necessities. The Linux version worked straight out of the box. However I had one annoying niggle with it (plus a lot of other Linux apps) . When it was launched, it did not “do anything” and leaving some doubt as to whether it ran at all. A launcher was created and saved onto the desktop. This made it more convenient, but still no telltale of activity.

The launcher was edited by using a text editor and opening it directly from the desktop directory.

```
#!/usr/bin/env xdg-open

[Desktop Entry]
Version=1.0
Type=Application
Terminal=false
Icon[en_AU]=/home/rob/Applications/UGS/ugsplatform-
linux/bin/icons/UGS_icon.png
Name[en_AU]=Serial_port_server
Icon=/home/rob/Applications/UGS/ugsplatform-
linux/bin/icons/UGS_icon.png
StartupNotify=true
Exec=/home/rob/Applications/CNC/Jason-Port-Server/serial-port-
json-server-1.96/serial-port-json-server
Comment[en_AU]=Serial_port_server
Name=Serial_port_server
Comment=Serial_port_server
```

The critical additions to the script are shown in red. This flashes the icon into the task bar while running and exits gracefully when finished, seconds later. The paths in green will no doubt be different in another installation.

This is not an “Expert’s Tip” from an expert Linux user, but it has been reassuring in use.

Appendix 2

Wiring up the GRBL Protoneer Interface

This caused a lot of grief early on. I was partly relying on old GRBL advice and applying some TinyG advice in the absence of good GRBL advice. Again my pathway is probably common sense to long time GRBL users out there, but it is included here because it is necessary if documenting all of the Chilipeppr (CP) application above that the underlying firmware be fully outlined that the CP worked with.

The Arduino and Protoneer Board V3.0 (red) arrived with the Chinese Engraver/Milling kit. The two boards make a compact system that was bit underwhelming when weighed up against the job that it was expected to do. However after the frame of the mill and the rest of the mechanicals were added, the Arduino+Protoneer was all that I had to drive it. An old GRBL version was installed. It was dutifully hooked up, ran all the motors and limit switches and could be driven from the command line setting configurations, performing homing and following the usual GCODE commands. Then it sat idle for 2 years. The Corona Virus isolation rules lead to a resurgence to tackle this thing for once and all. A couple of other projects had stalled because I needed to make some nifty PCB's to finish the projects off.

Chilipeppr had advanced and was looking very slick so I made that my focus. In the process of learning CP I also figured out some basic information that people who end up with the Arduino+Protoneer (still widely available) should be shared to save them the hassles and greatly enhance the chances of them enjoying CP as well.

First of all I tracked down the latest GRBL firmware, V1.1h and installed it. The major problems I had to over come, were all related to limit switches and probing. The GRBL programmers had swapped the Z-Axis limit switches and the Spindle PWM Pin. The Z-Axis limit switches are now on Pin12, and the Spindle Speed control is now 11. The other minor problem I had to figure out was the limit switches were wired as NO (normally open) and triggered when closed. The Protoneer has + and – marked for pairs of pins for all three axis and their upper and lower limits. I mistakenly thought each had its own pin. However I was mistaken, there are not enough pins, and XYZ only had one pin each with two switches on each pin. This means GRBL does not know whether it has hit an upper or lower switch, it only knows from the direction it is travelling in when it hit it. So the NO switches are wired in a logical OR fashion for each axis. It was a waste to use pin 11 capable of Pulse Width Modulation for the Z-Axes and use a less capable pin Pin12 for Spindle speed control so they were swapped. However it did upset the neatness of my wiring once I had figured what I needed to change.

I found I was getting way too many random *Alarm:1* from electrical noise triggering the limit switch inputs. I had twisted all the input leads initially as a precaution, but this was not enough once spindle motors etc began whirring etc. Consequently I added 1k Ohm pull up resistors to the axis limit inputs and this dampened most of it. I was able to nearly totally eliminate it by running a decent wire (salvaged from a 240V extension cable) from the negative side of the 12V input into the Protoneer, connecting back to the aluminium chassis ground. This solved a lot of "RF" problems and helped stabilize input signal levels etc

The next main problem came from mixing Tiny-G recommendations with GRBL+Protoneer reality. The Tiny-G advice suggests to use the lower Z-Limit switch as the probe point for Auto Level functions. This is not correct for GRBL, and this has become a major advantage for me. GRBL uses a dedicated probe Pin A5. This means GRBL can distinguish between a probe signal and a Z-Axis limit. Much of the setup of the system relies on finding the [X0,Y0,Z0] point to begin the milling/probing. I put a 1k Ohm resistor in as a pull up, and found that the conductivity from the chassis to through the motor to the probe/tool was very reliable, so I connected the PCB to the A5 probe input. I was probing an active PCB plane with a grounding tool tip. This worked really well.

I experimented with a multi meter on the PCB to set my Z-Axis zero by detecting when the probe contacted. This was really handy. So I formalised it by putting a red LED in series with the clip to the PCB, and increased the pull up resistor on A5 to 330Ohms. This low resistance would increase the current, lower the forward voltage of the red LED (which is the colour with the lowest forward voltage) and help make sure electrical contact was seen by GRBL as a low input. This has turned out to be a real game changer for me as I can now confidently move the tool tip in to probe easily down to 0.01 increments very reliably. It has also meant that the Auto Level feature on ChiliPeppr works like a charm as well.

So now I can run the Holy Grail of PCB engraving (and 3-D printing), Auto Leveling. It also meant realigning tools after tool changes to Z-Axis zero was a doddle. The only caveat is with the Tool Changing, one must remember to probe over copper that is still connected to the probe circuit. The milling will hopefully have isolated many tracks. Being able to detect the copper and back off 0.03mm to break the circuit, then zero Z-Axis out is bliss. The red LED is left in circuit all the time. No wires to tooltips either, just leave the PCB connected.

Lastly I added three 200nF Ceramic capacitors to the spindle motor terminals. One was wired across the motor, and then the next two were wired from + and – on the motor and the other ends both tied to chassis GND. I also got a large toroid salvaged from a switching power supply and threaded the motor power supply through 4-5 times to further attentuate any RF coming from the motor running, but especially from the start up surge.

Summary:

a. GRBL (+Arduino Uno) has only one input per X, Y & Z. Upper and lower limits are wired in a common OR function using the NO contacts. Add in 1k Ohm pullups, and capacitors most likely will not be required.
b. Z-Axis input pin is now Arduino P12 and Spindle Speed P11, these two are swapped from what is labelled on the Protoneer V3.0 GRBL board.
c. Use the Probe Pin A5 on Arduino+GRBL. Decrease pullup resistor to 330Ohms and add a red LED in series with A5 line to indicate probe contact with Probe/Tool at GND.
d. Make s a very solid GND connection from the -Ve (GND) connection on the Protoneer board to the chassis of the Milling machine. This greatly reduced false triggering.
e. Put a 200nF Ceramic capacitor across the spindle motor, and another two seperately to GND off either motor terminal to the machine chassis. Also winding the power supply wires to the spindle through 4-5 times through a toroid will also help reduce RF and false triggering of inputs.

A 3D printed end-switch holder design. It just clamps onto the bars and slides along to be adjustable.

Appendix 3

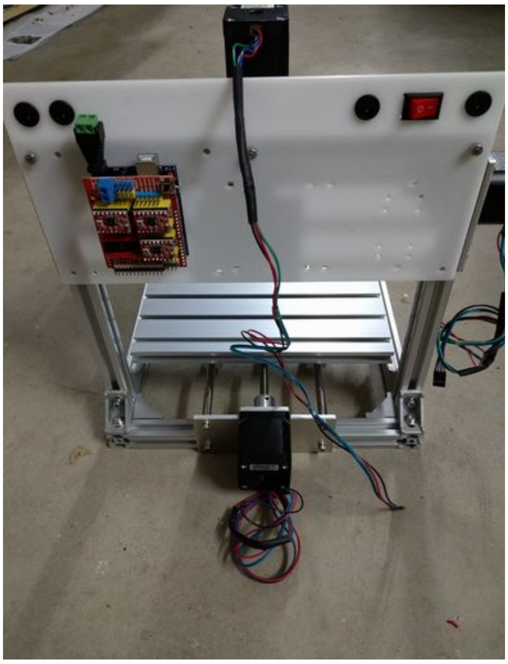
Milling Machine build and configuration

The mill I chose two years ago was a cheap and cheerful Chinese no-name brand and arrived well packed and all the essential “big-bits” ready to go. With the right skill it would be possible to set it up and run it with what was supplied. However I was a raw CNC beginner and found the jargon quite daunting, even though I had been working with Robotics in schools for many years. I bought it with the idea to learn about CNC technology and maybe somehow make a circuit board or two. Whirring razor sharp cutting blades was new territory for me.

There are many similar machines on the Chinese market today and represent very good value for money with these two caveats, they are a hobbyist system, and the person buying them must be prepared to make an effort to customise them to their own needs. With a few additions and patience a good benchtop system can be set up. I will outline the machine below so that is clear what was used in the ChiliPeppr description above.

The part list (from memory) was:

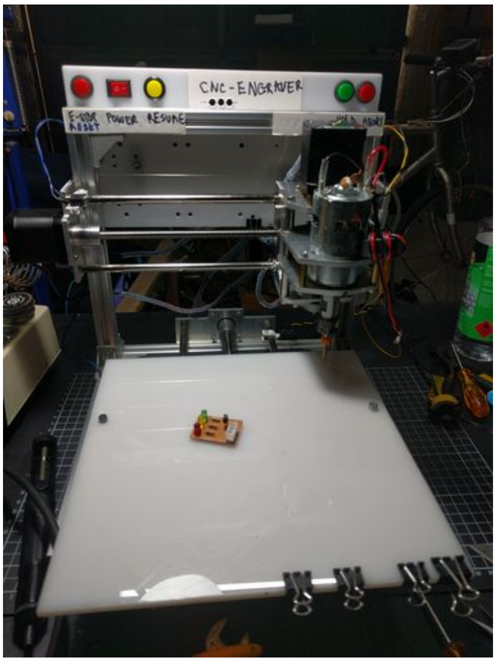
3 Stepper motors + wiring and plugs
1 Arduino Uno, Protoneer V3.0 (GRBL Board) and Pololu style drivers.
1 12V power supply (just using 12V 5A at the moment)
1 24V power supply (not used).
7 piece 2020 Aluminium Frame, with right angles and all bolts and T nuts, rubber feet.
3 lead screws, 3 flexible drive connectors, 3 thrust bearings and 3 anti-backlash nuts and springs
6mm Aluminium brackets (including metal bearings) to hold the XYZ axis rods and lead screws
1 230x110mm Aluminium bed with 4 linear bearings for the Y axis and 4 bolts to hold down work.
1 12-24V DC Spindle motor (very “tight” bearings)
Screw in barrel connectors for the power supplies and rocker power switch SPSC
Selection of Engraving tool attachments and three engraving tools.
Acrylic back mount for CPU and switches, and plastic spiral for looming.

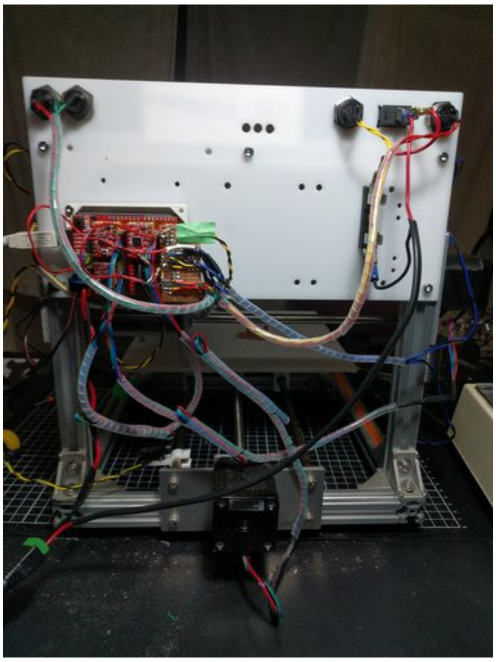


The early build stages (circa 2018?).

What I have added:

6 limit micro-switches, mounts and wiring.
1 16mm diameter Milling Tool holder with selection of proper milling collets.
Switches for E-Reset, Resume, Hold and Abort inputs back to GRBL board.
3 of 1k Ohm resistors for Axis input pull ups.
1 of 330Ohm resistor for pull up for A5 Probe input, plus clip and red LED connection indicator.
1 Toroid filter on the spindle motor power lines.
4 Large NO push buttons for control panel.
A selection of 1/8" milling tools and drills.
1 inline fuse holder, and a 5 amp fuse.
1 Acrylic sheet on the bed to extend work area to 230x230mm.





Current set up at the time of this document was written (April 2020).

Review:

The build is quite good for my purposes, I take it to be excellent value. I have looked at subsequent designs on eBay etc and feel it has stood up quite well. I paid around $250 Australian (delivered) at the time.

Main advantages:
a. Uses lead screws for all three axis, making it difficult to move the machine position while changing tools, and providing a pretty good level of rigidity for milling and power.
b. Lead screws have flexible metal connectors to the motors and thrust bearings at the other end to secure them with very little backlash. The thrust bearings are not seen so often on current designs.
c. 6mm Aluminium braces used to mount motors and bearing very strong.
d. Motors are a good size, 42H047H-054A-002 and pretty strong on 12V
e. Metal bearings premounted for X and Z travels.
f. Spacers and alignments all worked quite well.

Small disadvantages

a. Backlash system pretty crude, there are better ones out there now. These were assembled from brass mouldings, spacers and springs and a bit basic, but worked after much setting up.
b. The spacing of the guide rods on the X-Axis gantry are fairly close and the Z-Axis built onto them sticks out a fair way. This means the downwards atbility of the Z-Axis to maintain tool position is somewhat compromised. However for PCB work, not much of a problem, however if you wanted to do some heavy work eg thick wood, plastics or even metals it would really struggle. It would be hard to keep up good cuting speeds and keep accuracy. However this is not what I believe the machine was sold for. It was advertised as an engraver.

Summary: It can produce quite good milling/drilling of PCBs. Milling PCB tracks down to sub 0.01mm accuracy pretty good, and like wise drilling wire holes is accurate (0.8mm to 1.2mm). However while it can mill the PCB from the blank, a 2mm milling bit is working fairly hard with 0.2mm cuts per circuit.

Configuration

The stepper motors had plugs and leads provided and the Protoneer had motor cables provided. They were connected up with suitable twisting and soldered colour to colour. So the following configuration set up, which includes motor directions and steps/mm are based on that.

Address	Contents	Meaning	Parameter and units
$0	10	10	Step pulse time, microseconds
$1	25	25	Step idle delay, milliseconds
$2	0	0	Step pulse invert (Binary mask 0-7)
$3	3	3	Step direction invert (Binary mask 0-7)
$4	0	off	Invert step enable pin, Boolean
$5	0	off	Invert limit pins, Boolean
$6	0	off	Invert probe pin, Boolean
$10	3	Standard	Status report options (Binary Mask)
$11	0.020	0.020	Junction deviation, mm
$12	0.002	0.002	Arc tolerance, mm
$13	0	mm	Report in inches Boolean
$20	0	off	Soft limits enable, Boolean
$21	1	on	Hard limits enable, Boolean
$22	1	on	Homing cycle enable, boolean
$23	0	0	Homing direction invert (Binary mask 0-7)
$24	50.000	50.000	Homing locate feed rate, mm/min
$25	635.000	635.00	Homing search seek rate,mm/min
$26	100	100	Homing switch debounce delay, milliseconds
$27	2.000	2.000	Homing switch pull-off distance, mm
$30	1000	1000	Maximum spindle speed, RPM
$31	0	0	Minimum spindle speed, RPM

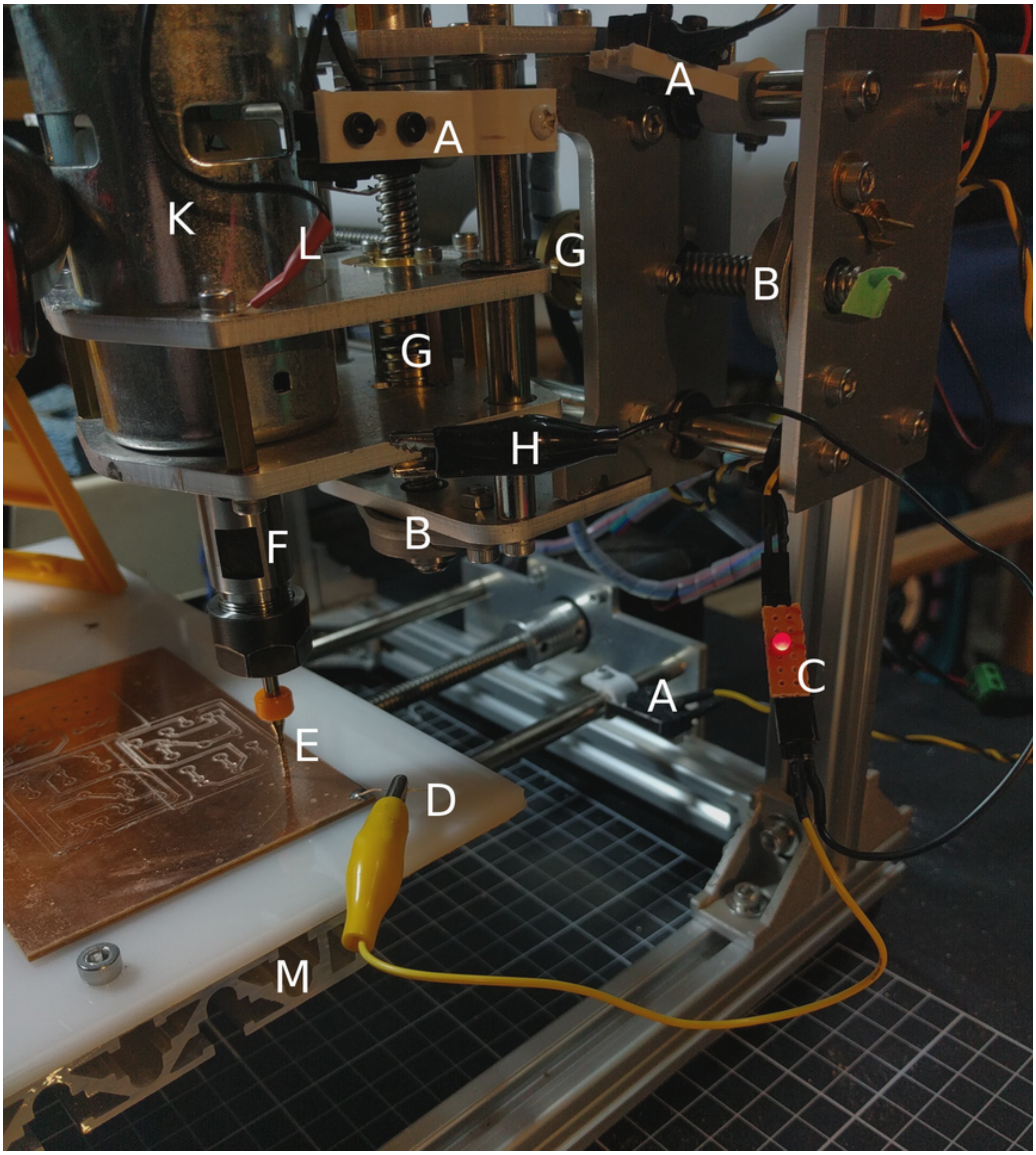
$32	0	off	Laser-mode enable, Boolean
$100	800.00	800.00	X-axis travel resolution, Steps/mm
$101	800.00	800.00	Y-axis travel resolution, Steps/mm
$102	800.00	800.00	Z-axis travel resolution, Steps/mm
$110	500.00	500.00	X-axis maximum rate, mm/min
$111	500.00	500.00	Y-axis maximum rate, mm/min
$112	500.00	500.00	Z-axis maximum rate, mm/min
$120	50.000	50.000	X-axis acceleration, mm/sec^2
$121	50.000	50.000	Y-axis acceleration, mm/sec^2
$122	50.000	50.000	Z-axis acceleration, mm/sec^2
$130	200.00	200.00	X-axis maximum travel, mm
$131	200.00	200.00	Y-axis maximum travel, mm
$132	100.00	100.00	Z-axis maximum travel, mm

On the command line values are set by $1=10 for example.

I have the Zaxis min switch available, but I am now using the Probe for setting my lower Z limits. This has proved to be a winner.

The Pololu drivers set to 0.7V reference voltage, by measuring the Voltage between the central wiper on the variable resistor and GND. Setting a small screw driver up with a wire clip can help do this comfortably.

Annotated photo of part of the system



A. Two limits switches mounted on 3-D printed clasps on the support rods (adjustable).
B. Two thrust bearings at opposite end of lead screws to motors.
C. Inline Probing LED indicator
D. A5 Probing connection to PCB
E. 2mm Milling tool being used as a probe
F. CNC style spindle motor tool holder with collets etc
G. Two backlash nuts, not so easy to set up.
H. GND connection for Probe circuit, traves through motor to the Tool. (Probably redundant)
K. 12-24V spindle motor
L. GND strap for Spindle motor bypass capacitors.
M. Extruded Aliminium build bed (With Acrylic sacrifice board).