PCB CNC MILLING: THE GOOD, THE BAD AND THE UGLY

DAVE VE3OOI

MARCH 2023

THE PROBLEM

1. CNC milling of PCBs is common today.

2. Its very inexpensive to fabricate a PCB from a PCB FAB House
   - ✓ They can populate the board with well known parts (e.g., Digikey # included)

3. If you compare costs of PCB FAB vs Milling, PCB FAB is much cheaper.
   - ✓ Can get 5 boards for about $12 CDN
   - ✓ CNC Mill all in is about $550.
   - ✓ Break even is 45 PCB FAB runs for cost of CNC Mill.

4. Benefits of CNC Mills (as I see it).
   - ✓ Faster to develop a board. PCB FAB House is 2 week for cheapest rate
   - ✓ Use CNC Mill for other milling work.

AGENDA

Milling for Success

1. Overview of Process
2. Mill Limitations
3. Recipe to Mills PCB Boards Successfully
4. Misc. Crap

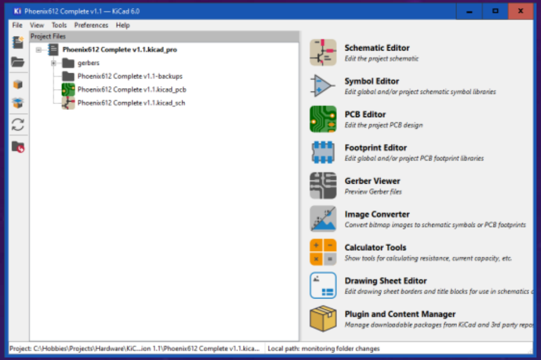
MILLING OVERVIEW

1. GENERATE PCB: NOT A TUTORIAL – See Wayne's KiCad Presentation

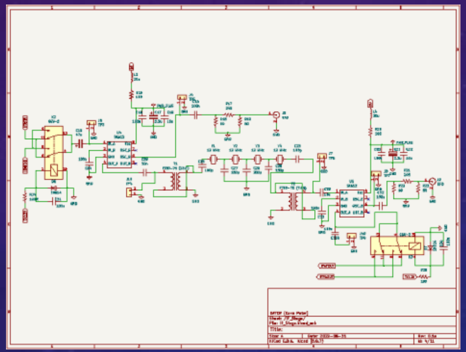
KiCad

KiCad is an open source software suite for Electronic Design Automation (EDA). The programs handle Schematic Capture, and PCB Layout with Gerber output. The suite runs on Windows, Linux and macOS and is licensed under GNU GPL v3.

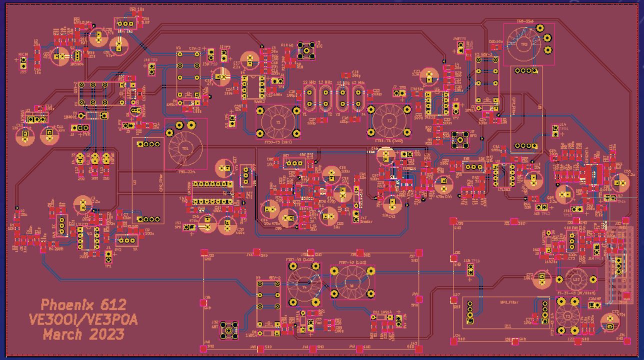
DipTrace is quality Schematic Capture and PCB Design software that offers everything to create simple or complex multi-layer boards from schematic to manufacturing files.



Generate Schematic



Generate PCB



Generate gerber files

Need to set custom clearances
(Mill trace size, spacing, etc.)

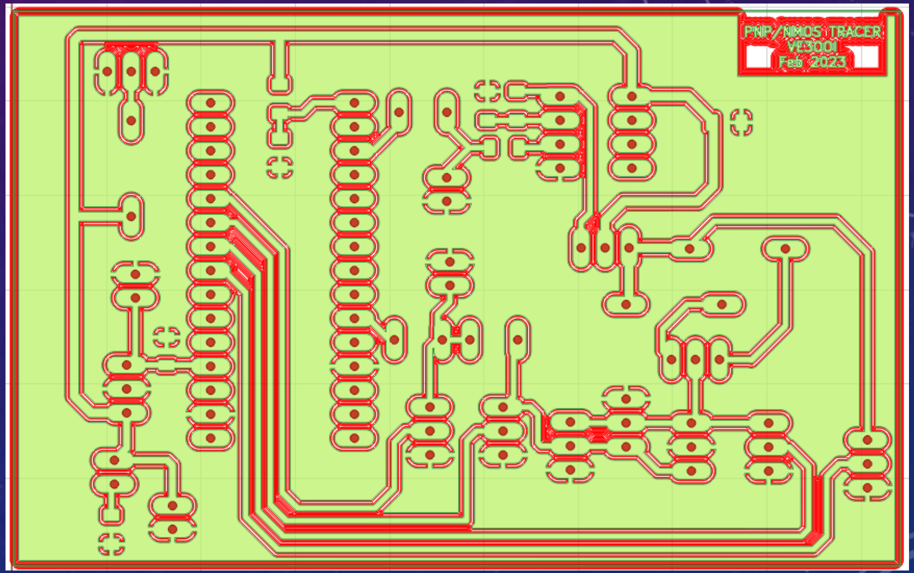
EXACTLY same process for FAB House
(Published Clearances)

MILLING OVERVIEW

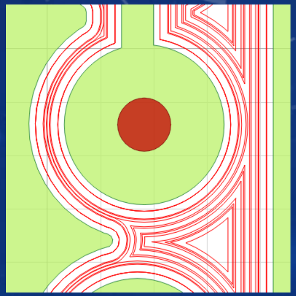
2. GENERATE Mill “G-Code”

G-code (also **RS-274**) is the most widely-used computer numerical control (CNC) programming language. It is used mainly in computer-aided manufacturing to control automated machine tools, as well as from a 3D-printing slicer app. It has many variants.

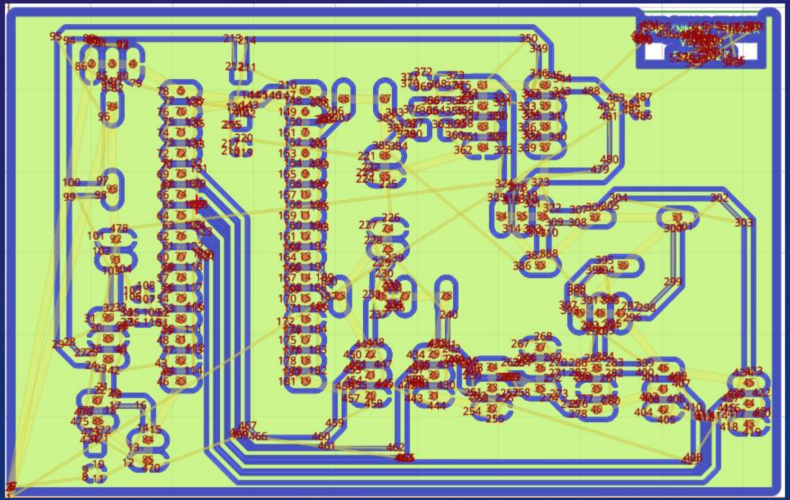
Generate Geometry
(i.e., Where to mill out)



Set Milling Parameters
(e.g., plunge depth, passes, etc.)



Generate G-code
(i.e., Code for CNC Mill)



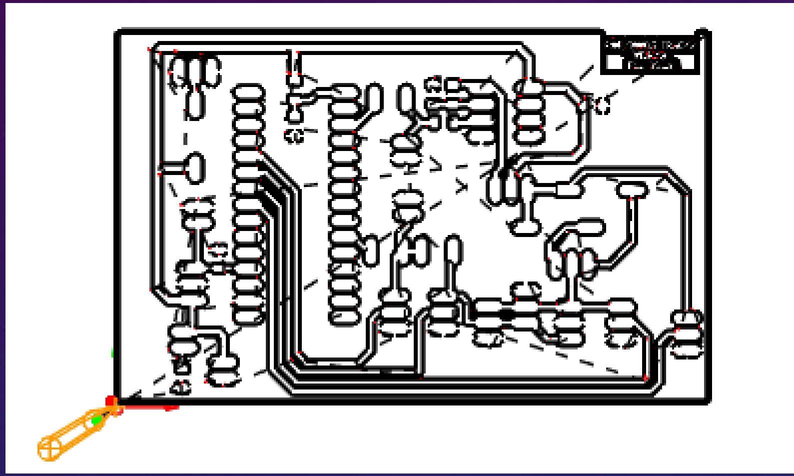
```
G01 F60.00
G00 Z2.0000

M03
G00 X-0.1037 Y1.2492
G01 F40.00
G01 Z-0.1200
G01 F60.00
G01 X-0.0923 Y1.1969
G01 X-0.0875 Y1.1762
G01 X-0.0822 Y1.1557
```

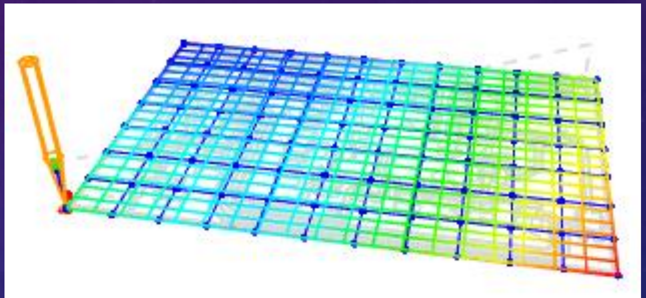
MILLING OVERVIEW

3. Load PCB on Mill. Double sided tape

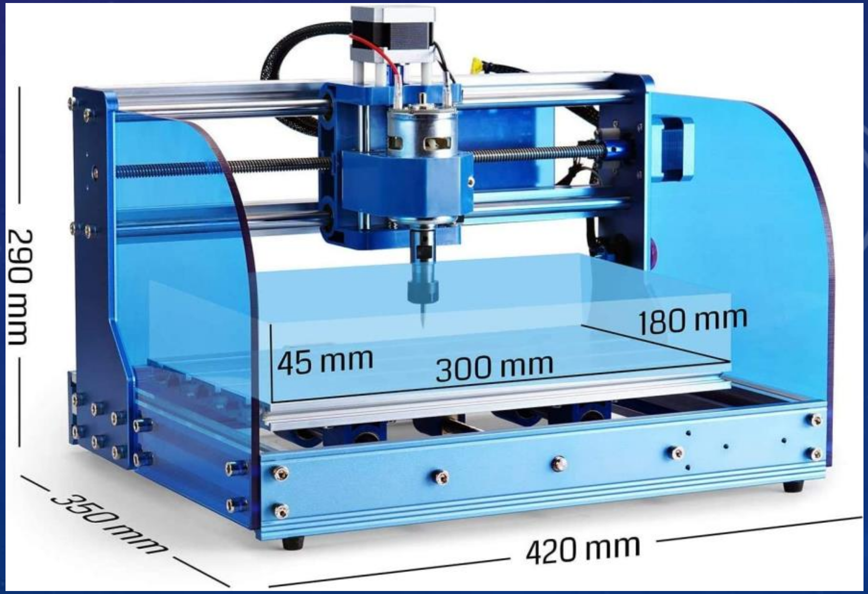
4. Send G-Code to CNC Mill. Open-Source Candle Program



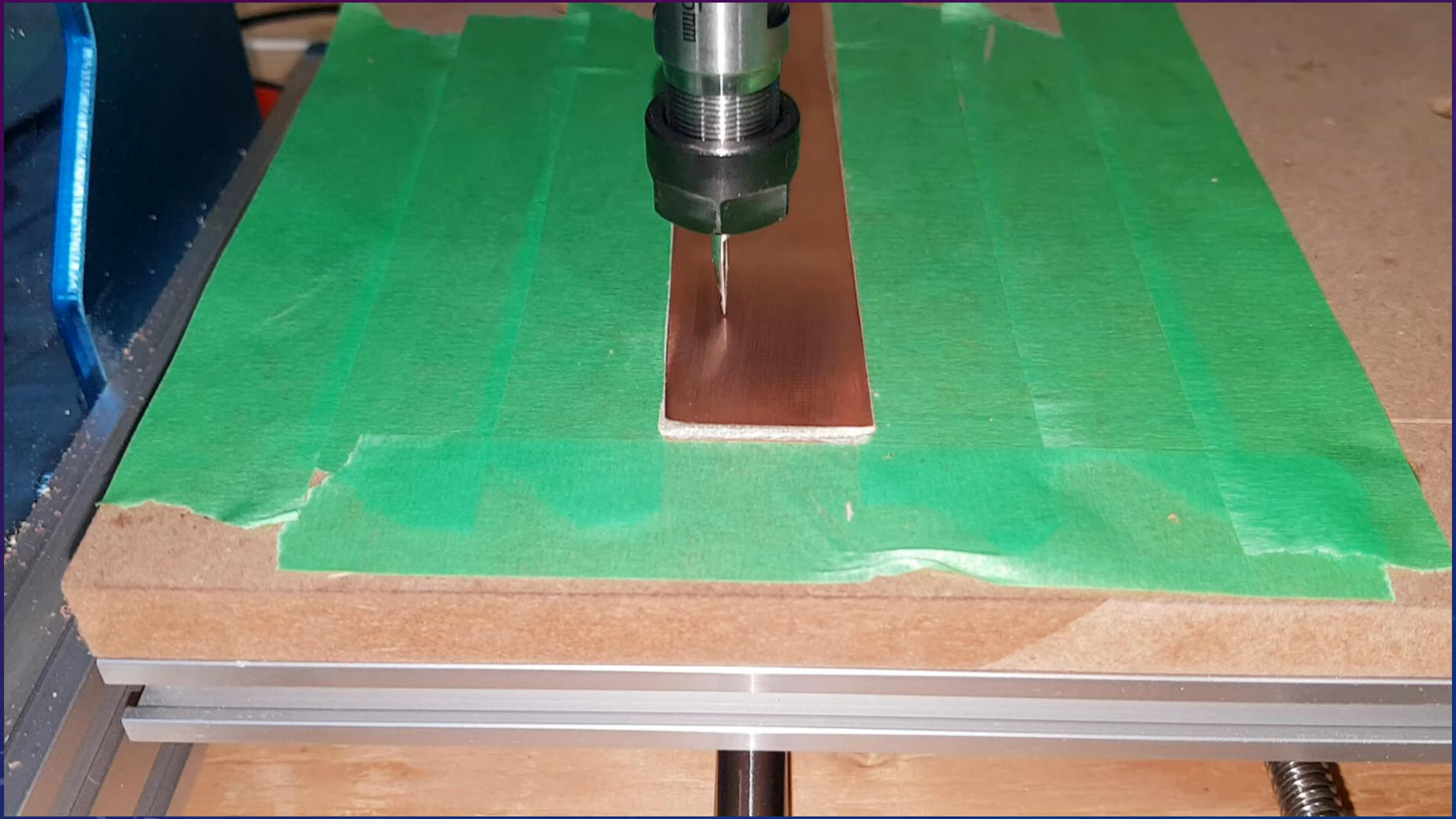
Generate Height Map



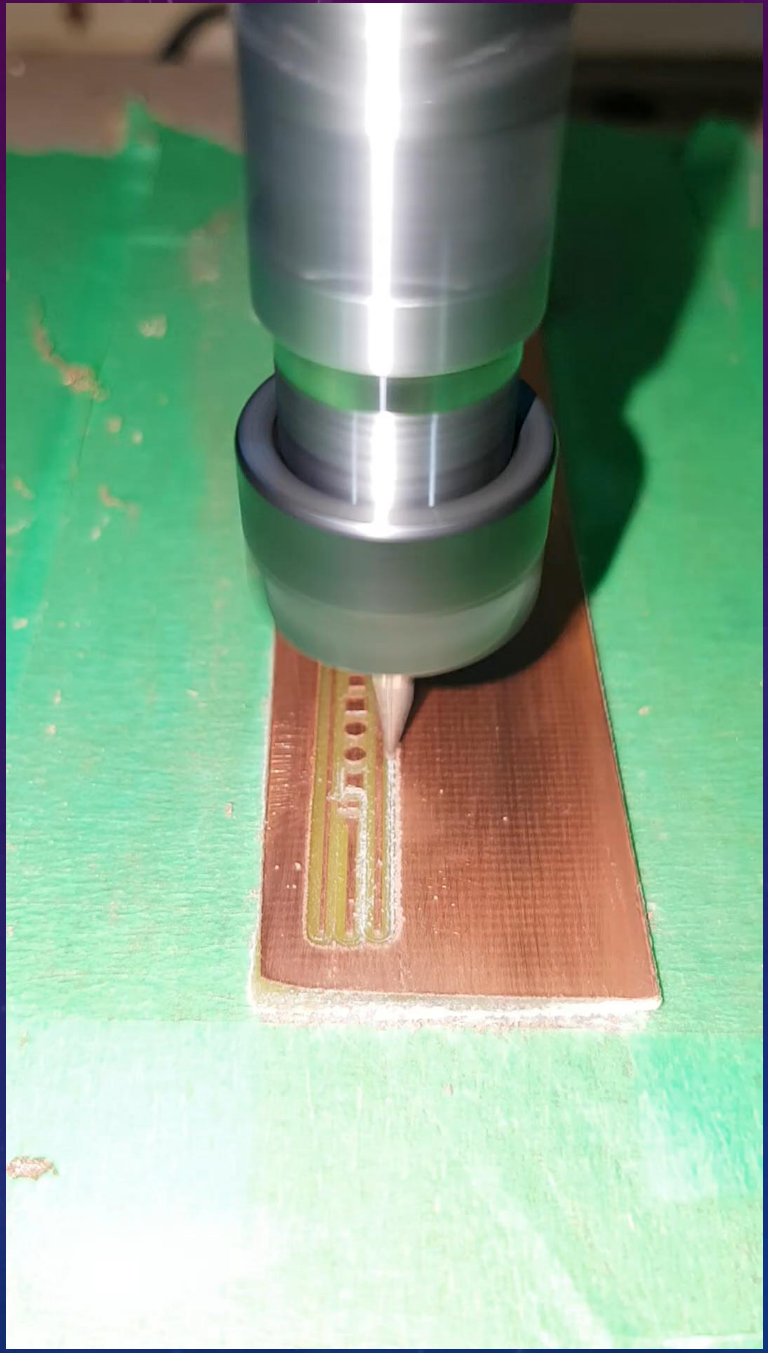
Let 'er Rip…



HEIGHT MAP



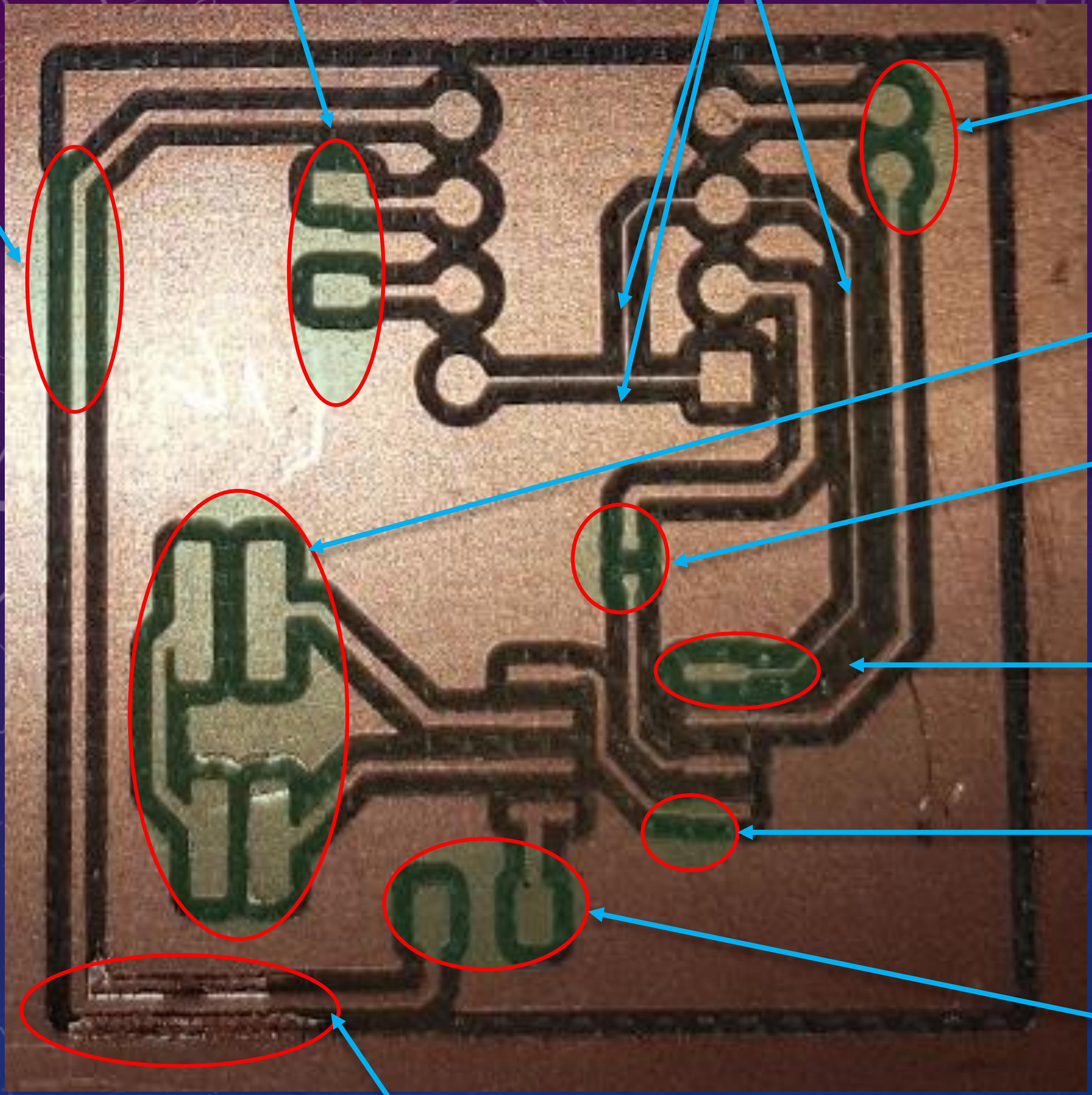
MILLING



DRILL HOLES

Not too much
FR4 removed

Small amount of
Power left on surface
after milling



1206 SMD

0.25mm Trace

Thr Hole Resistor
(Upright)

Larger 1206 SMD

405 SMD

SOIC with 0.25mm Trace

SOIC with 0.5mm Trace

1206 SMD

Height Map issue – need more points to map

CNC SETUP: SUMMARY

- CNC X Error: 0.01mm per mm (2mm per 200mm i.e., send G-code to move 200mm)
- CNC Y Error: 0.007mm per mm (1mm per 150mm)
- CNC Z Error: Measured in next slide. About 0.01mm based on variation in height maps
- 0.1mm x 30 deg V bit
  - 0.1536mm width at 0.1mm depth
  - 0.1589mm width at 0.11mm depth
  - 0.1697mm width at 0.13mm depth
- Initial 0.1mm deep cut which left gaps. Increased to 0.11 based on 0.01 error in Z axis
- 7 passes with 40% overlap (based on min, max gap around pads/traces)
- Ran one job slow (XY: 60 and Z: 40) at 40% overlap and second job faster (XY: 120, Z: 60) at 20% overlap and 4 passes
- 8mm spacing height map

3. **Feedrate:** The speed of the cutting tool while cutting in inches/minute of mm/minute depending on the project settings.

HEIGHT MAPS: Z AXIS ERROR

Height Map Testing Map 2.map

```
0.14;0.00;5.42;96.76
3;10;-1.00;1.00
0;20;20
0.0040000000000013358;0.022000000000002018;0.032000000000000028
0.027000000000001023;0.039999999999999147;0.050000000000000711
0.054000000000002046;0.067000000000000171;0.0740000000000000162
0.073000000000000398;0.080000000000001847;0.088000000000000966
0.094000000000001194;0.10699999999999932;0.10800000000000054
0.10800000000000054;0.11700000000000088;0.118000000000000021
0.10699999999999932;0.11700000000000088;0.118000000000000021
0.097000000000001307;0.10399999999999992;0.10800000000000054
0.089999999999999858;0.098000000000000253;0.10000000000000142
0.080000000000001847;0.086999999999999744;0.089999999999999858
```

Run height map 3 times, on same board, and compared variation

Map 1

Column1	Column2	Column3	Column4
0.14	0	5.42	96.76
3	10	-1	1
0	20	20	
0.003	0.028	0.033	
0.036	0.05	0.06	
0.06	0.073	0.08	
0.08	0.092	0.096	
0.1	0.11	0.113	
0.116	0.123	0.123	
0.116	0.123	0.122	
0.105	0.111	0.113	
0.098	0.106	0.106	
0.088	0.1	0.098	

Map 2

Column1	Column2	Column3	Column4
0.14	0	5.42	96.76
3	10	-1	1
0	20	20	
0.004	0.022	0.032	
0.027	0.04	0.05	
0.054	0.067	0.074	
0.073	0.08	0.088	
0.094	0.107	0.108	
0.108	0.117	0.118	
0.107	0.117	0.118	
0.097	0.104	0.108	
0.09	0.098	0.1	
0.08	0.087	0.09	

Map 3

Column1	Column2	Column3	Column4
0.14	0	5.42	96.76
3	10	-1	1
0	20	20	
-0.007	0.016	0.026	
0.02	0.038	0.048	
0.046	0.062	0.066	
0.066	0.078	0.081	
0.09	0.1	0.102	
0.103	0.112	0.112	
0.1	0.108	0.108	
0.092	0.102	0.103	
0.08	0.092	0.093	
0.076	0.086	0.088	

		Column1	Column2	Column3
Average		0.009067	0.008133	0.0078

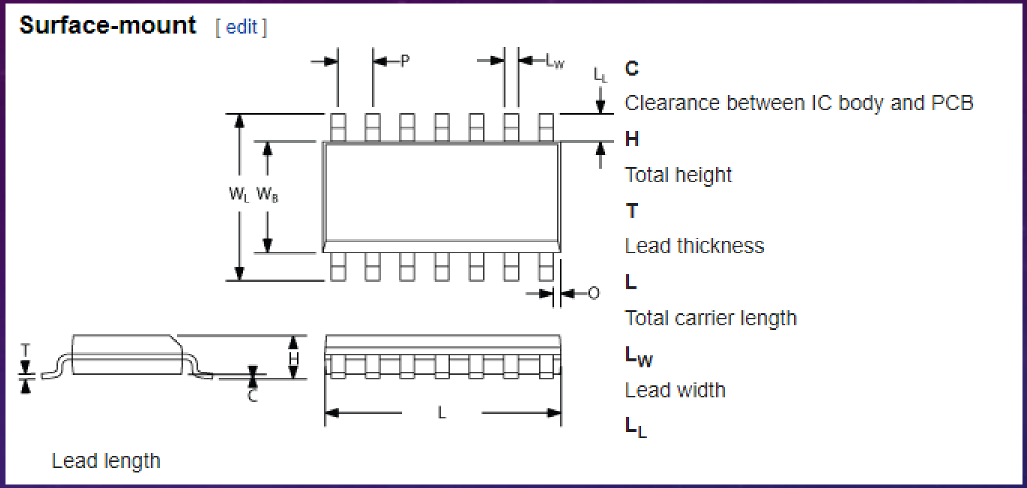
Why is the tolerance of the CNC Machine Important?

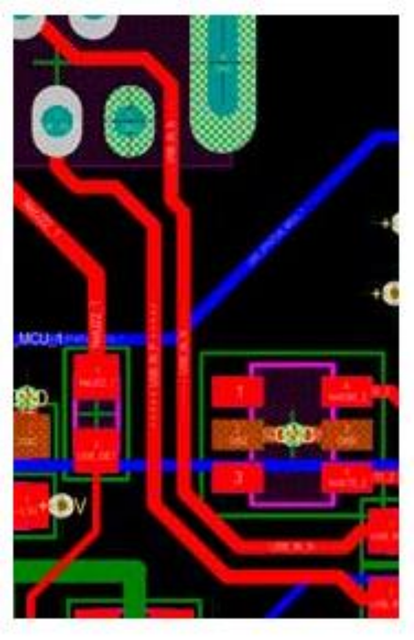
1. If you focus on the trees, you miss the forest

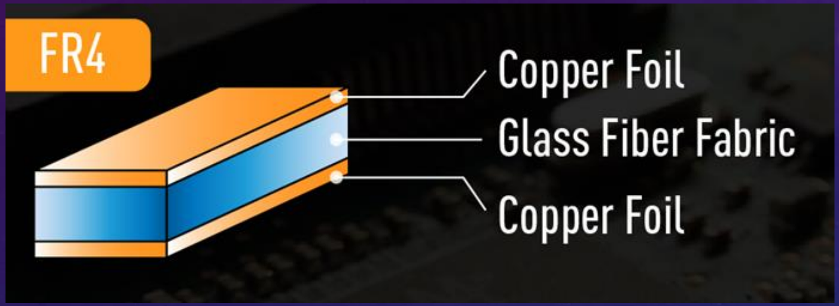
CNC CLEARANCE

Need to ensure CNC Bit removes material between copper traces and pads to reduce changes of bridges when soldering

Need to mill away sufficient copper in X,Y as well as Z







Copper Thickness Conversion Chart

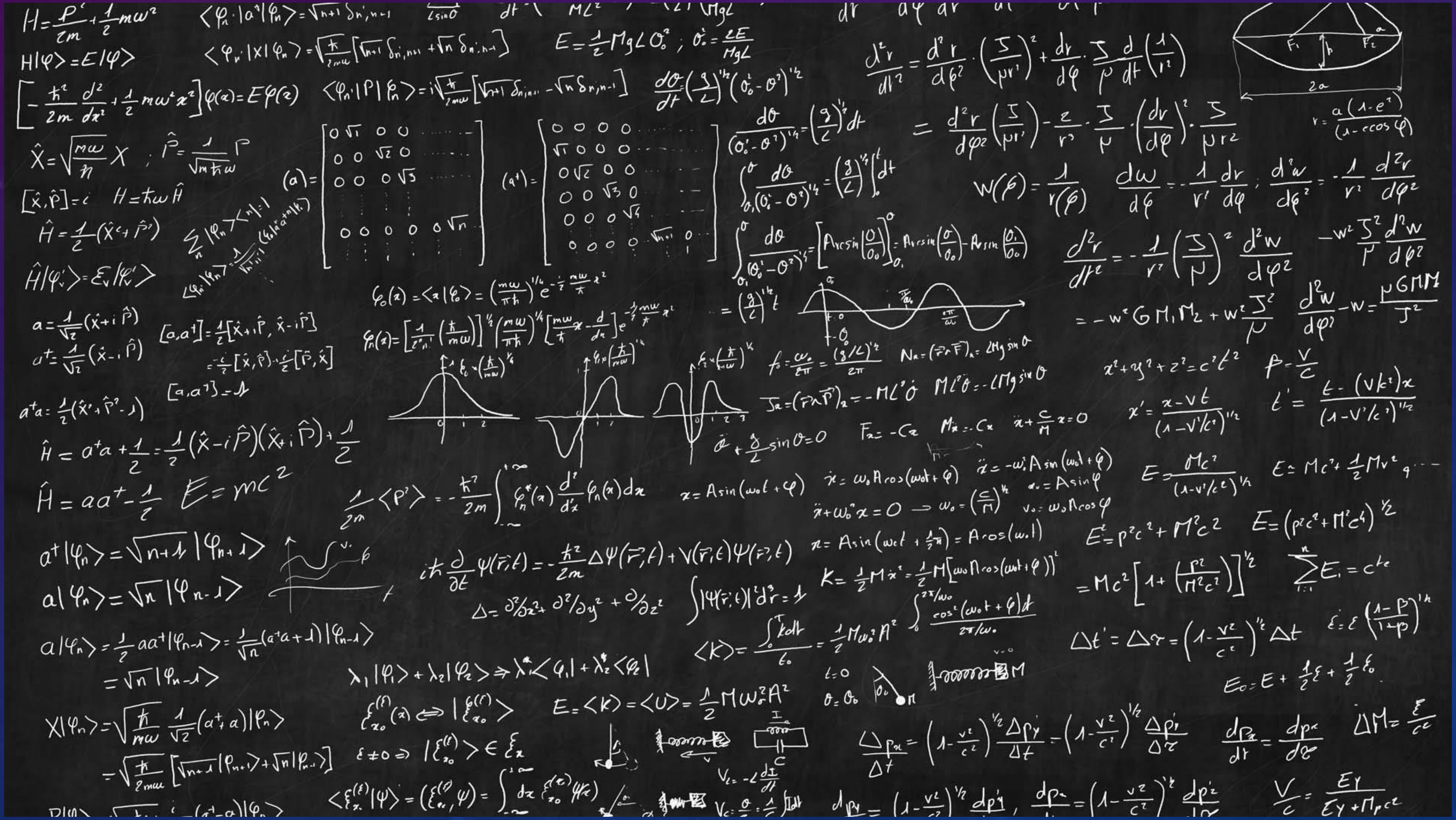
oz	1	1.5	2	3	4	5	6	7	8	9
mils	1.37	2.06	2.74	4.11	5.48	6.85	8.22	9.59	10.96	12.33
inch	0.00137	0.00206	0.00274	0.00411	0.00548	0.00685	0.00822	0.00959	0.01096	0.01233
mm	0.0348	0.0522	0.0696	0.1044	0.1392	0.1740	0.2088	0.2436	0.2784	0.3132
µm	34.80	52.20	69.60	104.39	139.19	173.99	208.79	243.59	278.38	313.18

PCB Board is about 1.5mm Thick.

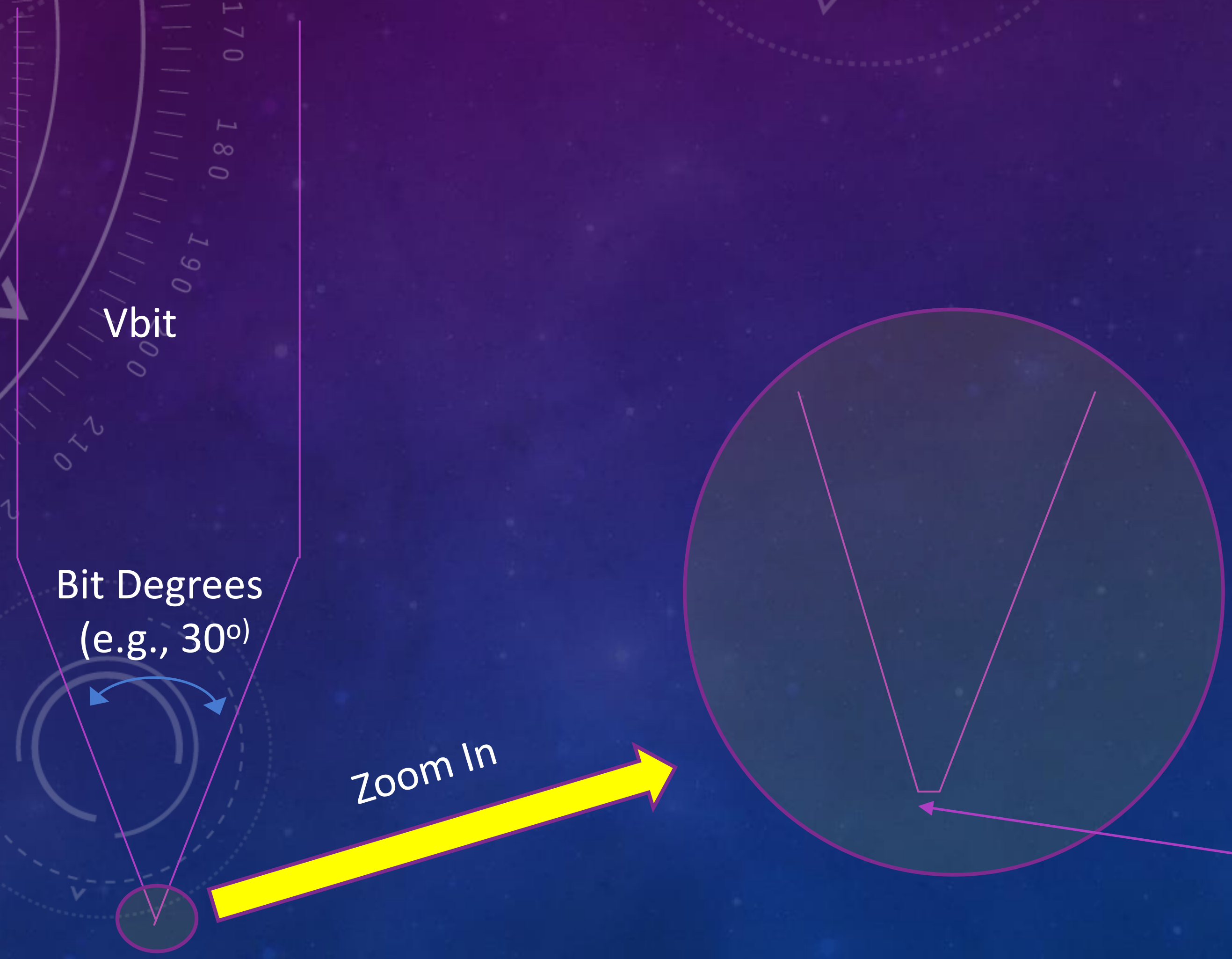
1. Milling to 0.11mm is removing 0.07mm of FR4!! (forest from the trees)
2. Error for X, Y, Z axis is about 0.01mm.

You need to understand how a V Bit works

2. Math is necessary



V BIT: SPECIFICATIONS

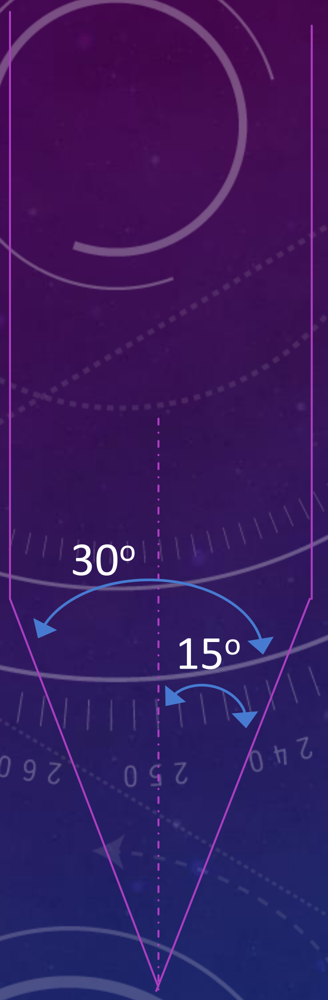


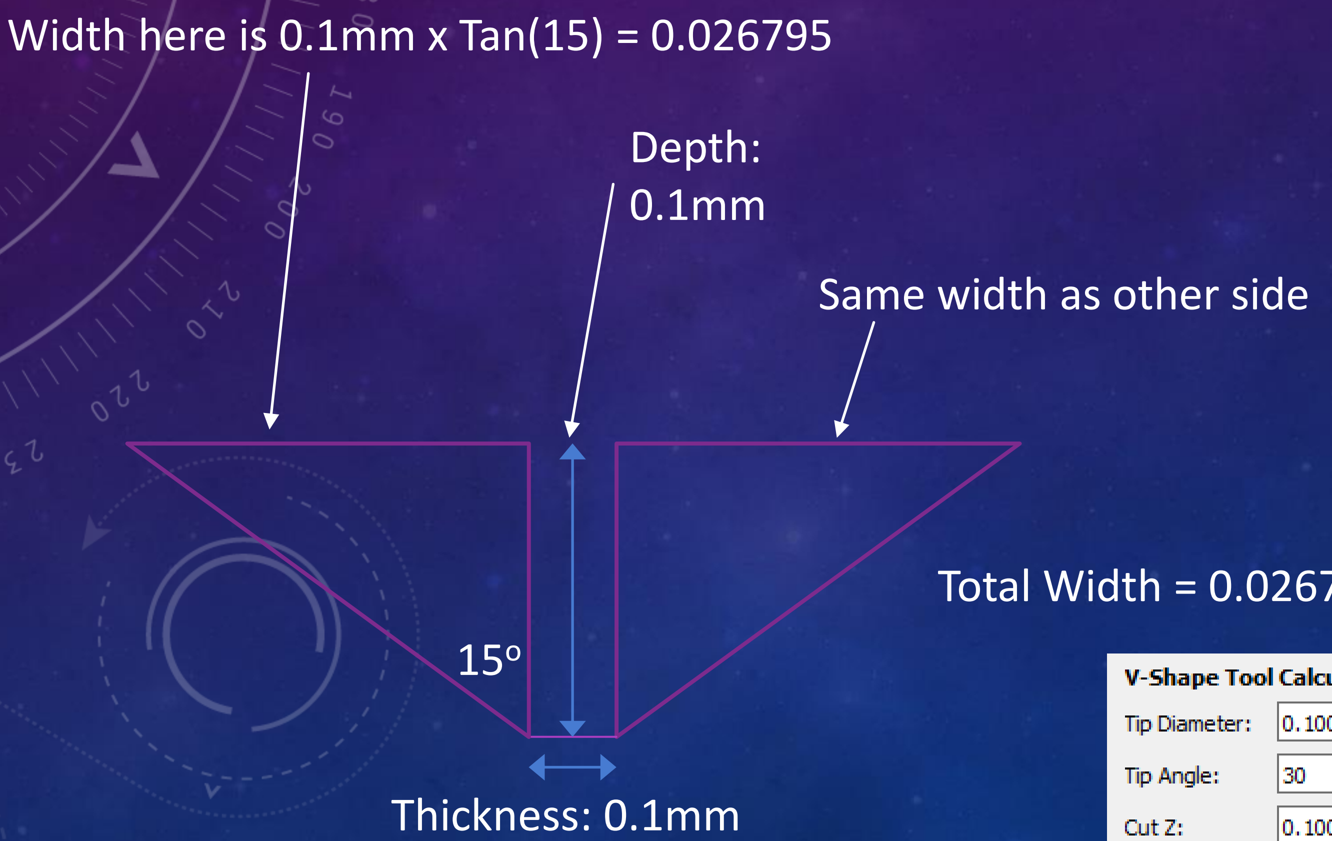
Not a sharp point.
Metal has edge/thickness
(e.g., 0.1mm)

i.e., thickness of metal

V BIT: HOW IT WORKS

V BIT: CALCULATE CUT DEPTH PER PASS



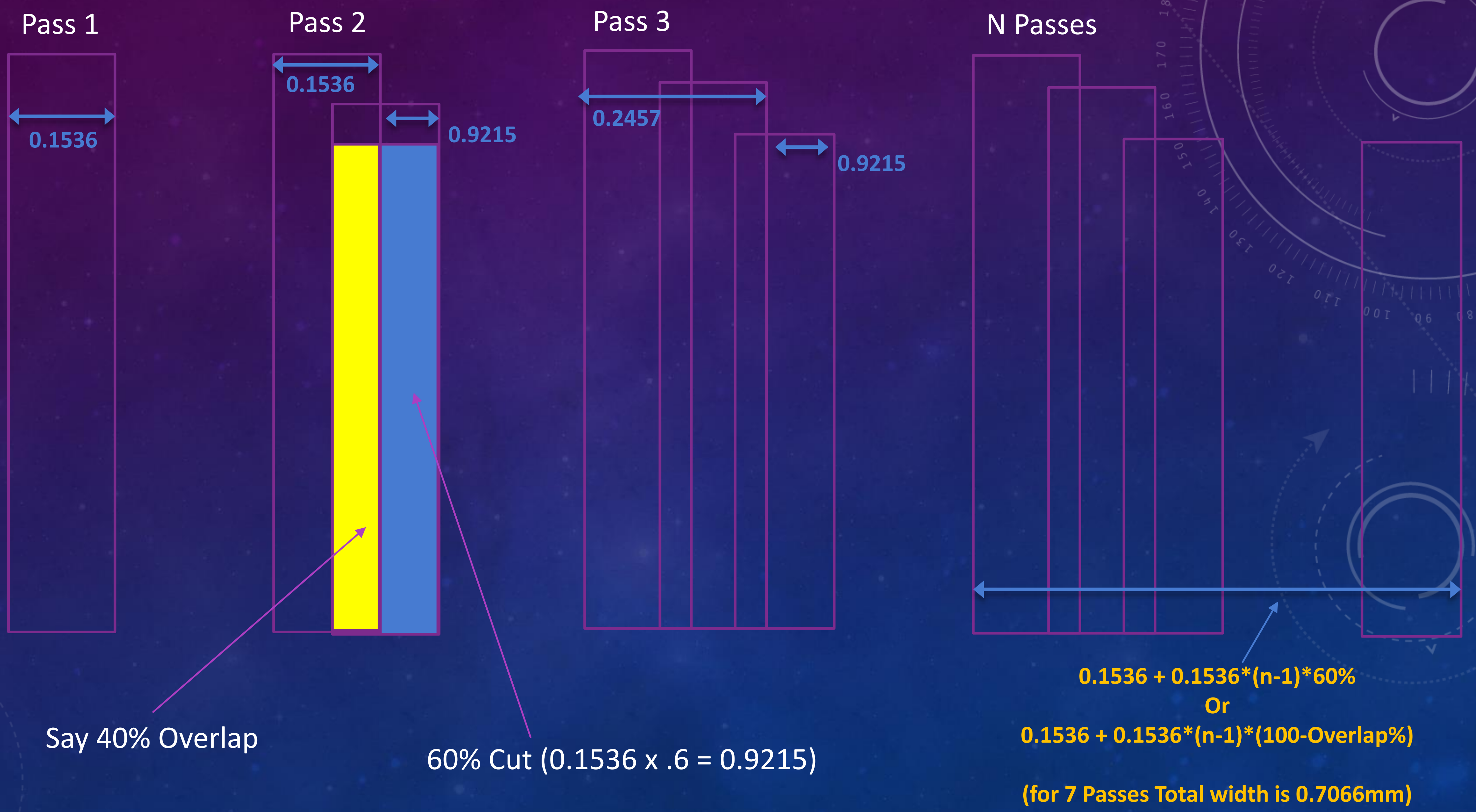


Total Width = 0.026795 + 0.1 + 0.026795 = 0.1536

V-Shape Tool Calculator

Tip Diameter:	0.1000
Tip Angle:	30
Cut Z:	0.1000
Tool Diameter:	0.1536

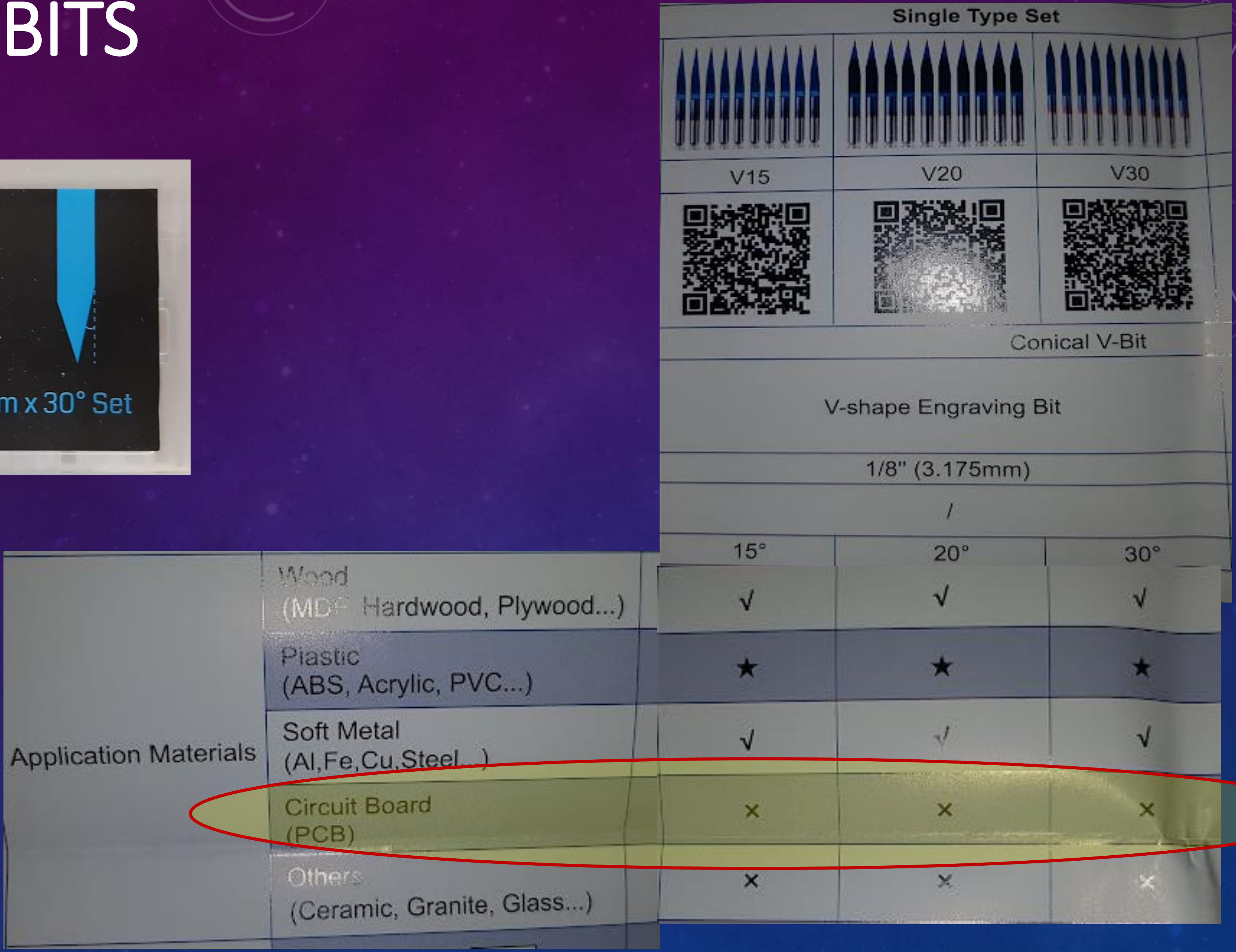
V BIT: CALCULATE PASSES



You need to know a CNC work

3. Bits and Chips

WRONG BITS

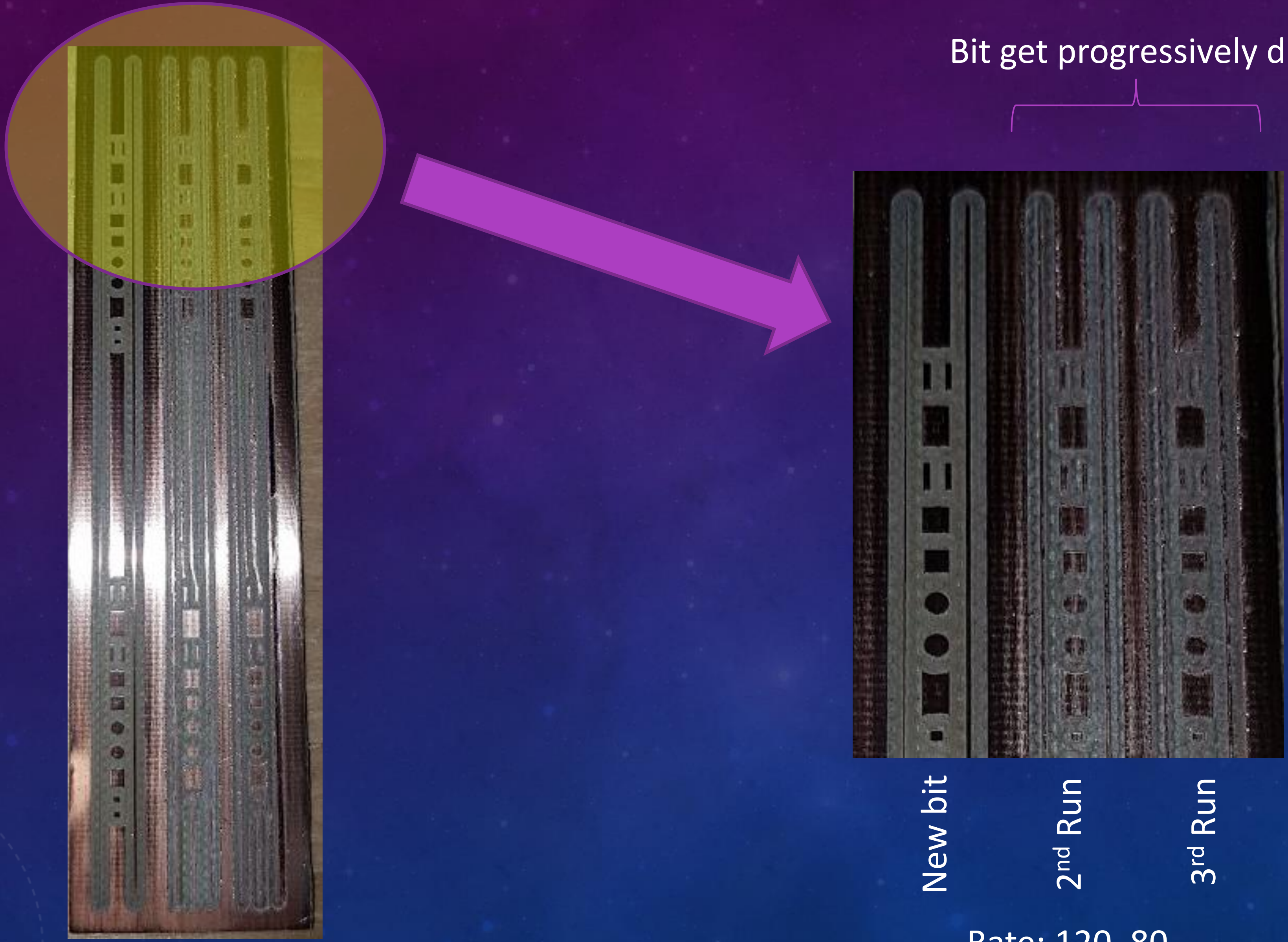


	Single Type Set		
	V15	V20	V30
	Conical V-Bit		
	V-shape Engraving Bit		
	1/8" (3.175mm)		
	/		
	15°	20°	30°
Application Materials: Wood (MDF, Hardwood, Plywood...)	√	√	√
Plastic (ABS, Acrylic, PVC...)	★	★	★
Soft Metal (Al,Fe,Cu,Steel...)	√	√	√
Circuit Board (PCB)	×	×	×
Others (Ceramic, Granite, Glass...)	×	×	×

Need to ensure you are using bits designed for PCBs

BIT ISSUES: WHAT HAPPENS

Bit get progressively dull with use



New bit

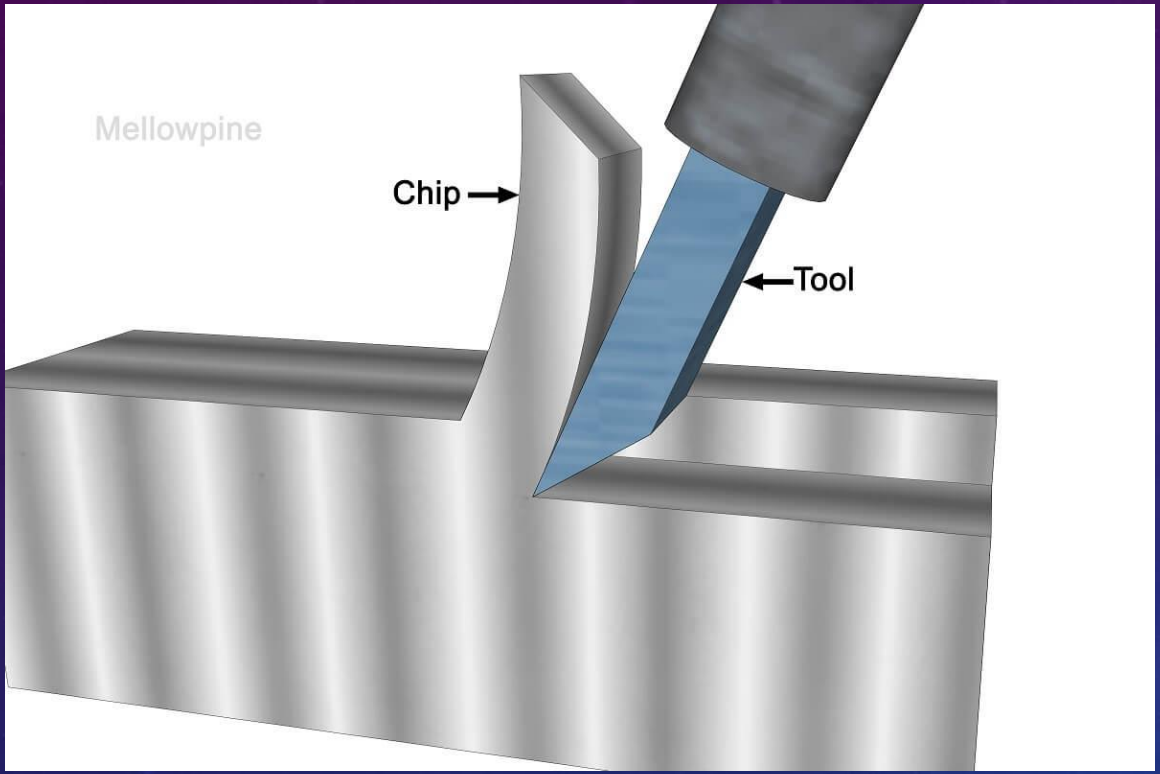
2nd Run

3rd Run

Rate: 120, 80

Cheap Bit?
Wrong Bit?
Wrong Setup?

CHIP LOAD AND FEED RATE



Speeds and feeds normally provided by manufacturer of bit. These CNC machines use cheap bits

$$\text{Chip Load} = \frac{\text{Feed Rate (inches per minute)}}{\text{RPM x No. of Flutes}}$$

OR

$$\text{Feed Rate} = \text{Chips Load (RPM x No. of Flutes)}$$

Bottom Line:

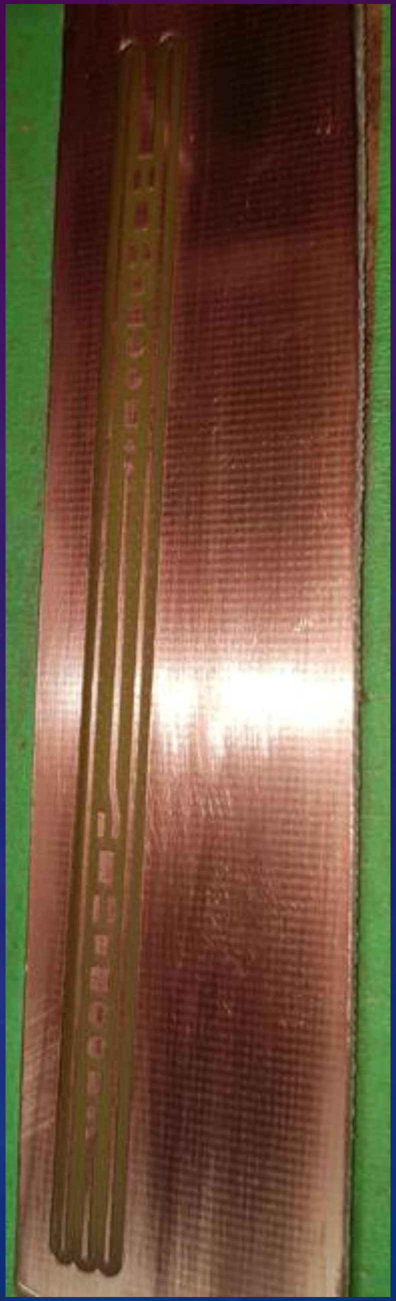
- ☠ If the bit gets too hot it will dull
- ☠ If chip size is too large, bit gets dull
- ☠ If feed rate is too high, bit gets dull
- ☠ If chip size too small and feed rate too low, bit gets dull
- ☠ …Your bit is going to get dull!!!

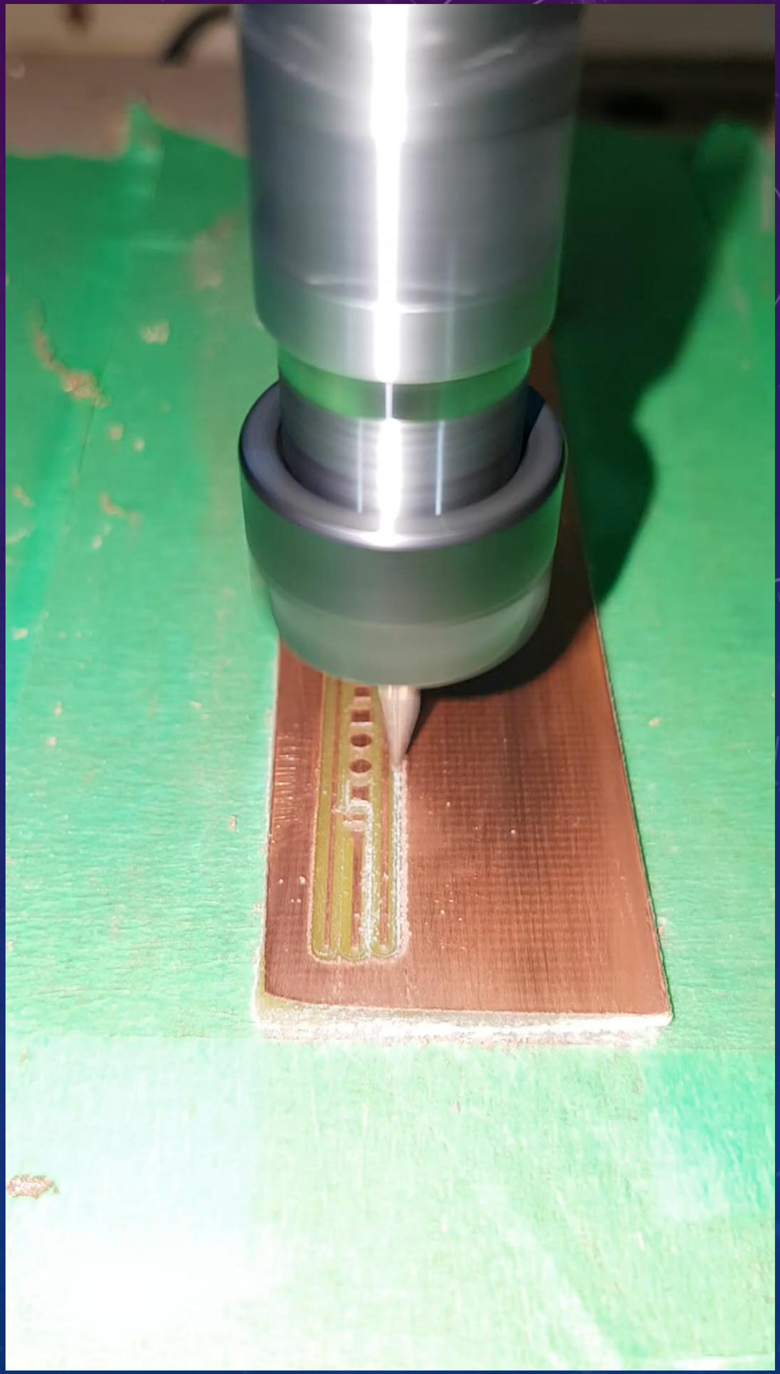
What can you do to minimize dulling of bit

- √ Slow down the feed rate and use lubrication

SOLUTION: MULTIPLE "RUNS"

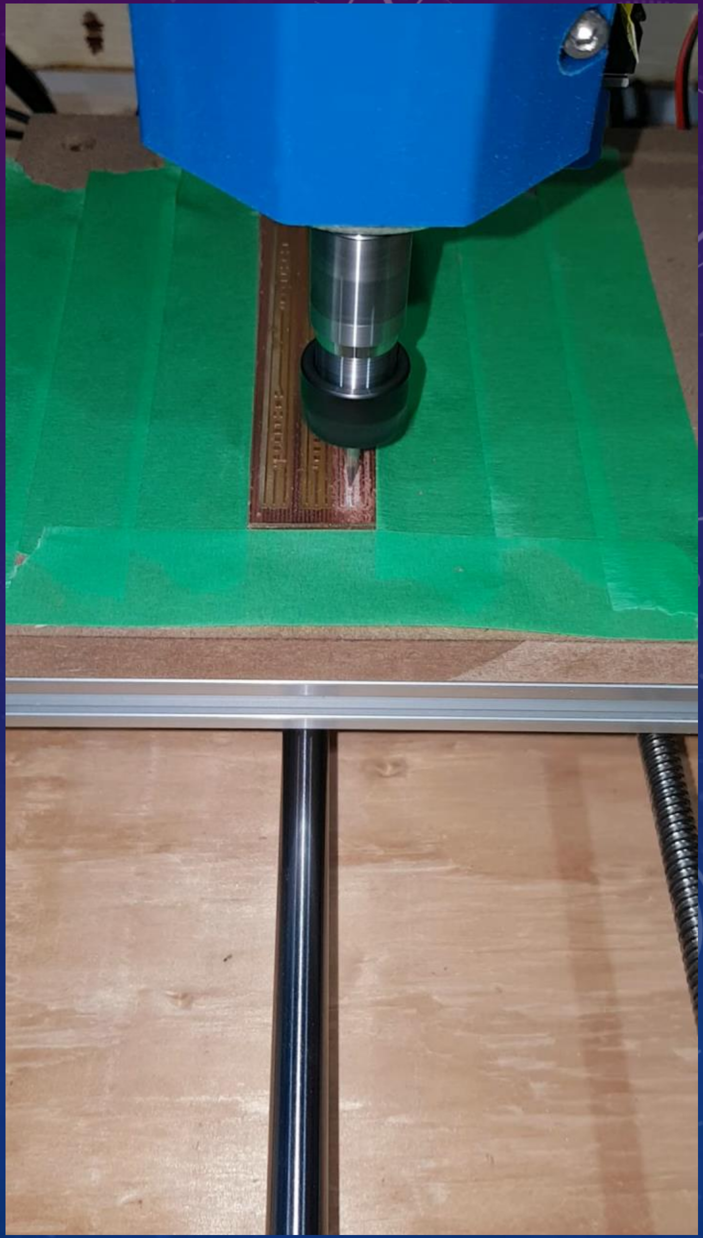
1. Normal 1st Run. Oiled surface
    - 0.11mm deep,
    - Rates XY: 50mm, Z:30mm,
    - 7 passes@40% overlap
2. Faster 2nd Run. Oiled surface
    - 0.11mm deep,
    - Rates XY: 120mm, Z:80mm,
    - 5 passes@25% overlap
3. Optional 3nd Run. Oiled surface
    - 0.12mm deep,
    - Rates XY: 120mm, Z:80mm,
    - 5 passes@25% overlap





THE RESULT

- ✓ 20º Bit
- ✓ 2 Runs
- ✓ First Run 60/40 rates, 40% 7 Passes
- ✓ Second Run (120/60) rates, 20% 5 Passes
- ✓ Lots of Oil
- ✓ Green Tape protects spoil board



DRILL HOLES

- Generate CNC Job for each drill bit. Use Z rate of 30, multiple passes each 0.7mm deep. Total depth 0.2mm larger that thickness of PCB (I used 1.7mm, board was 1.5mm)
- Always home (i.e., 0,0) after each job.
- Change bit, Re-zero Z using probe, reset Z=0
- Use last heightmap

SUMMARY OF STEPS: ISOLATON ROUTING

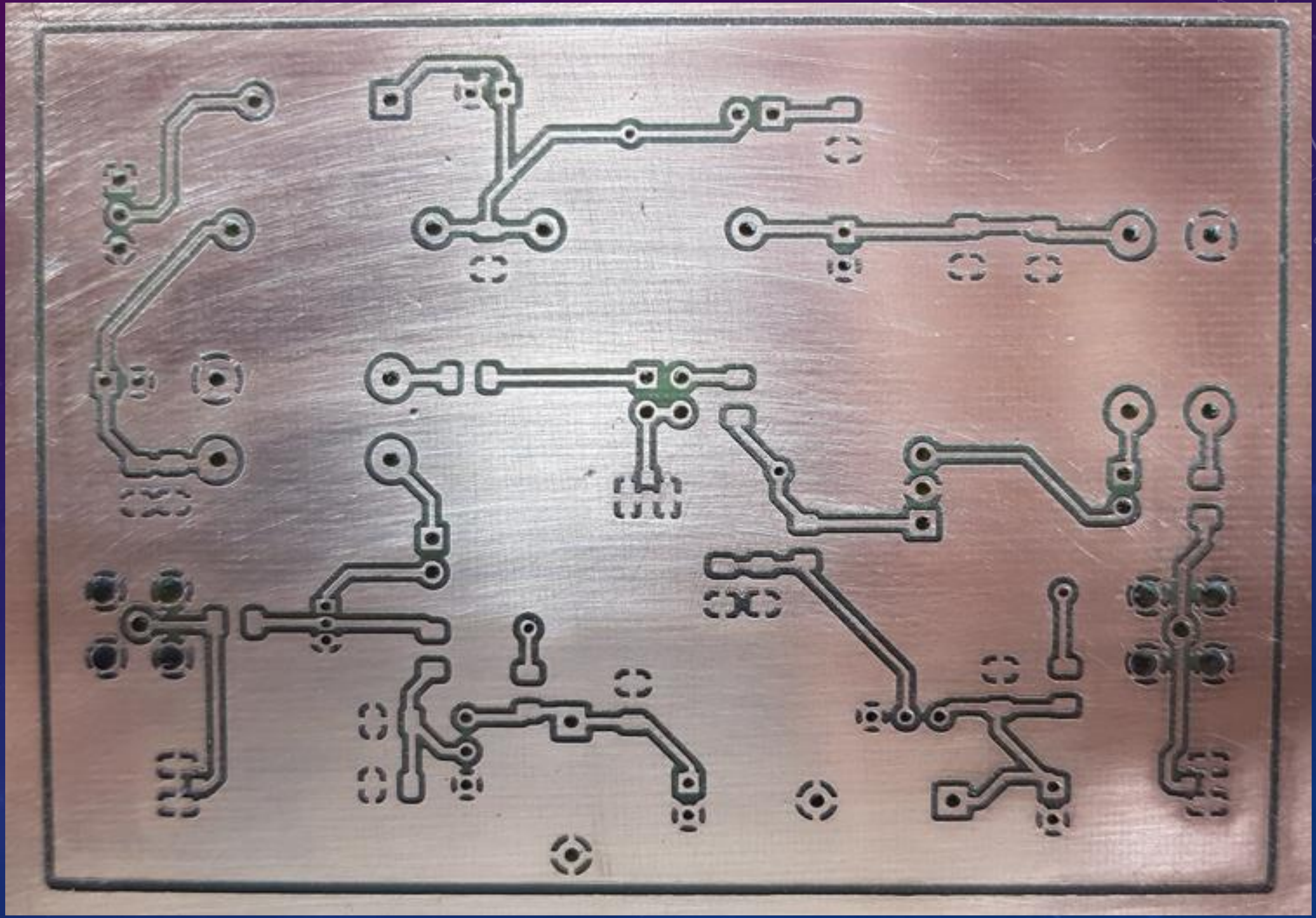
- From my PCB min separation for smallest footprints: 0.219, 0.445 and 0.635mm. Therefor I need a bit to cut much less that 0.219mm
- PCB is 1.5mm, with copper about 0.03 to 0.05mm thick. Error for my CNC was about 0.01. Its not practical to remove 0.0x mm consistently. Decided to go with 0.1mm with 0.01 error total of 0.11mm. However, during a run at this depth, the bit barley scratched the surface and went with 0.13mm depth (i.e., increased the error to an additional 0.02mm)
- Using a 30deg 0.1mm V-bit (which is not recommended for PCBs) with a depth of 0.13mm, each pass would be 0.1697mm which is under 0.219mm. Used 7 passes with 40% overlap to give max clearance of 0.1697(1+60% x 6) = 0.7806mm which is close to 1mm gap around traces and pads.
- Created two CNC jobs both cutting depth of 0.13mm. One job ran with XY rate: 50, and Z rate: 30 and the other with XY: 120, Z: 80 both with return to origin (0,0) after job. The second job was 25% overlap with 5 passes. The idea was to run the slower job first then run the faster job to clean up any errors. Different overlaps would remove any gaps from prior pass.
- The board was placed on CNC with double sided tape with painters' tape all around the spoil board to protect the spoil board from oil debris. The CNC was homed and then moved to the staring point on the PCB. With the bit fitted, the Z axis was probed. Then X,Y and Z axis was zeroed to establish an origin. The Machine X, Y coordinates were written down just in case I had to redo the job. I created a user button which would home the machine and they move to the coordinated I wrote down
- The slower CNC job was loaded.
- The surface was probed twice to confirm deviation and ensure consistent results. I used a spacing of about 8mm between probe points. Both results were saved
- I placed a thin layer of oil on the board to help cool the bit. These bits are notorious for getting dull or breaking. Possibly due to heat.
- With the Map height loaded I started the job. I noted that mainly copper was being milled and a small amount of FR4 was being milled away.
- After the job finished, I cleaned the surface with rubbing alcohol and a paper towel. I "VERY lightly" sanded (with 320 wet-dry paper) the surface to remove any burrs. Again, <u>lightly</u> sanded to not screw up the height map.
- Oil was applied to the surface again and the second faster job was loaded. The height map was reloaded, and the job was started. I noticed a <u>very</u> small amount of material was removed. The second job cleaned up any errors left by the first job.

SUMMARY OF STEPS: DRILLING

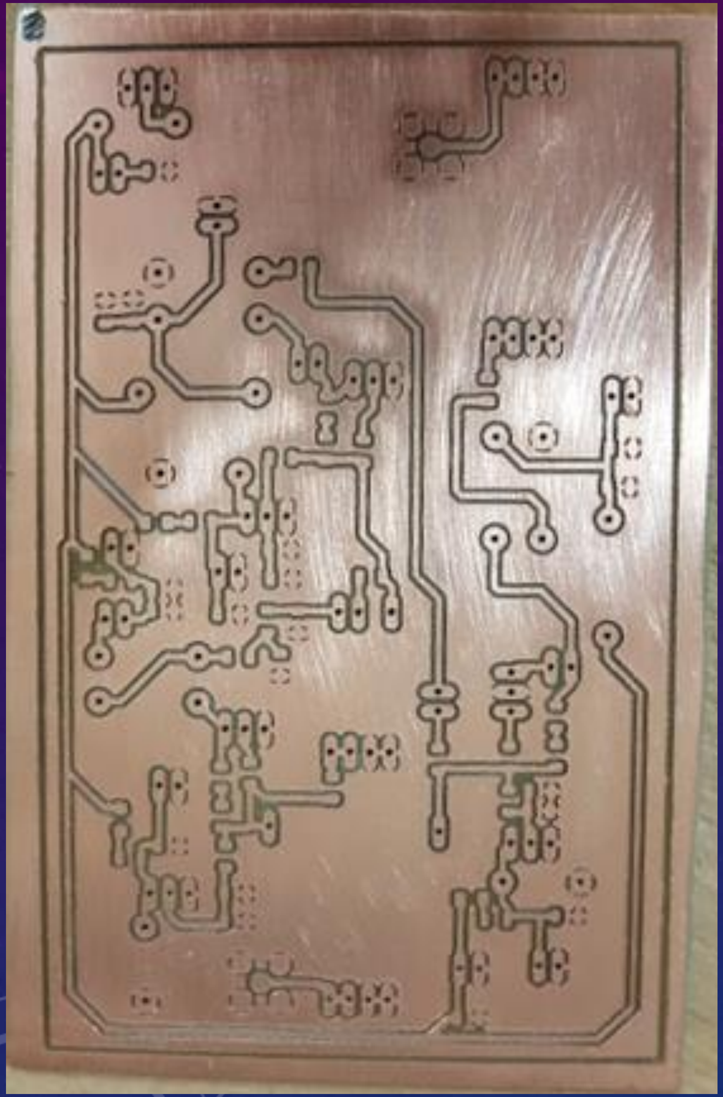
- I created a CNC job for each drill size (0.7, 0.75, 0.8, 0.9, 1.0, 1.1, 1.2, 1.7). I doubled up on some jobs (e.g., used the 0.7mm bit for the 0.75mm job, used to 1.0 mm bit for the 1.1 mm job, etc.). Since the board was 1.5mm thick I decided to drill to 1.7mm to provide margin. Drilling was done in 3 steps at 0.7mm each at a Z rate of 30. Each job would return the CNC to 0,0 (very important).
- Before each job was run, I re-zeroed the Z axis by probing the surface (at 0,0). The Z height was zeroed.
- The height map was reloaded for each job and the job was started. Between jobs I vacuumed the surface of any debris

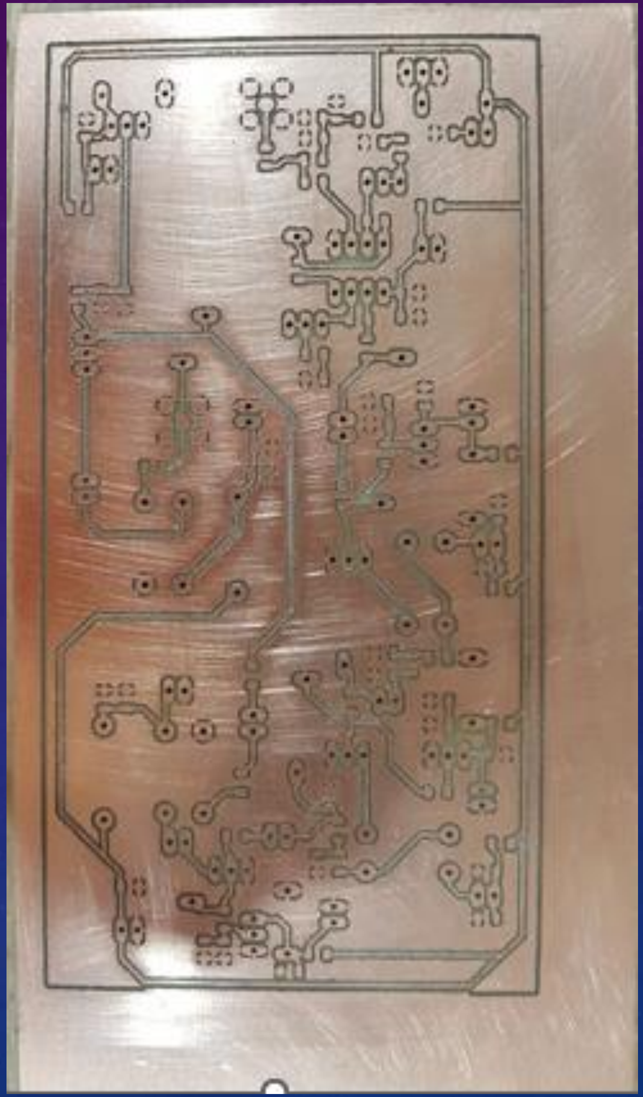
The Finished Product

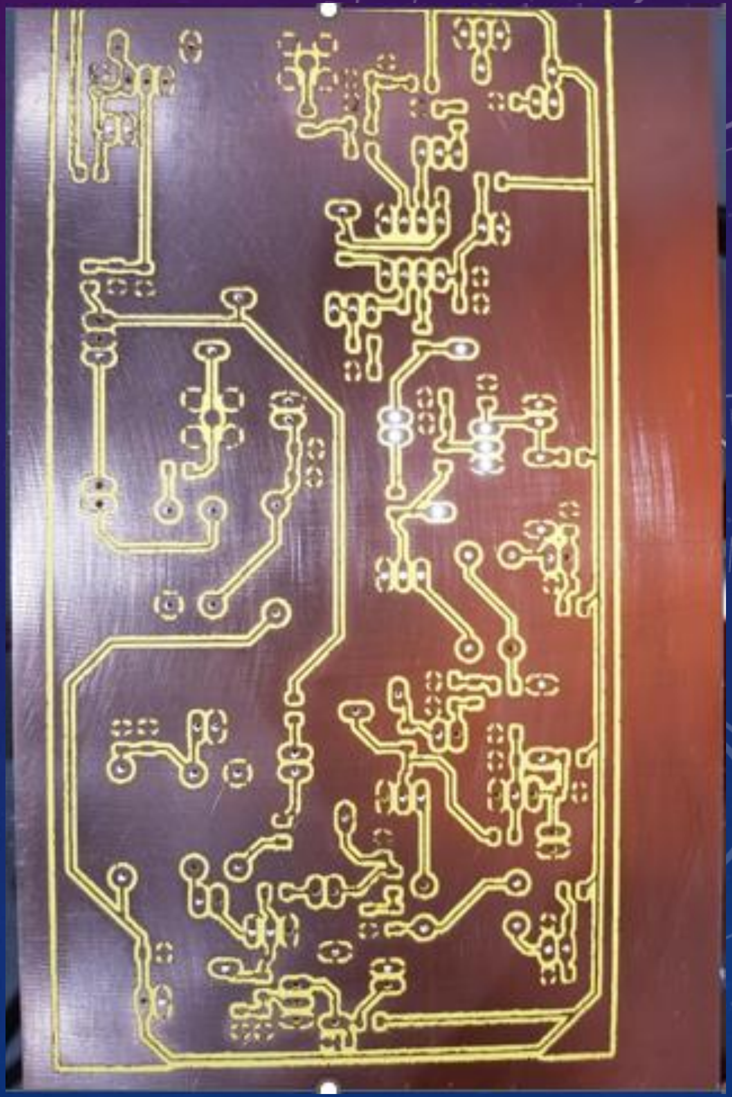
EXAMPLE: FINISHED BOARD



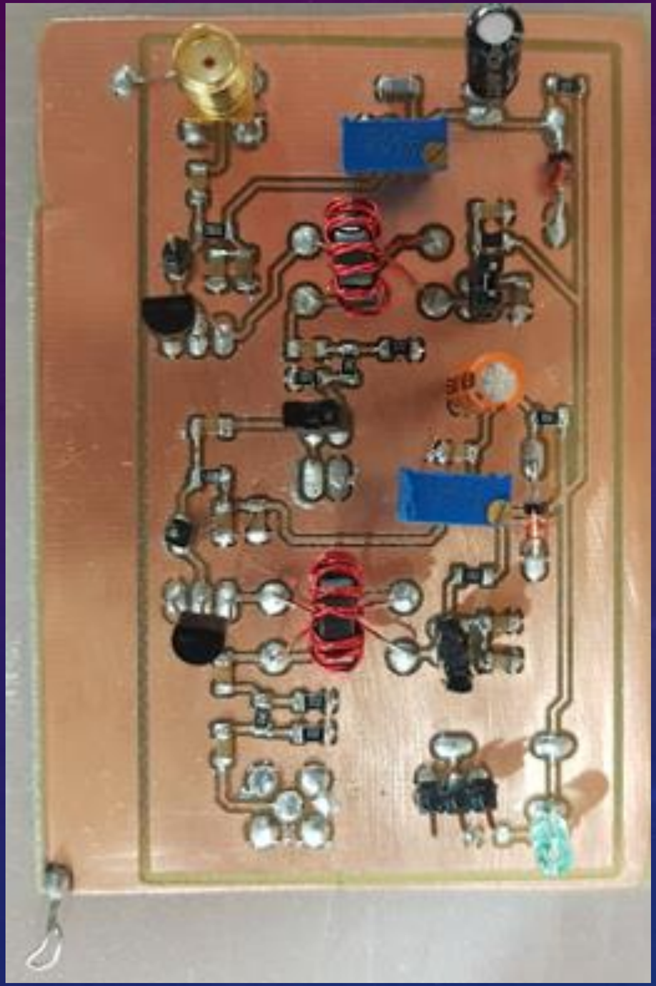
EXAMPLE: FINISHED BOARD



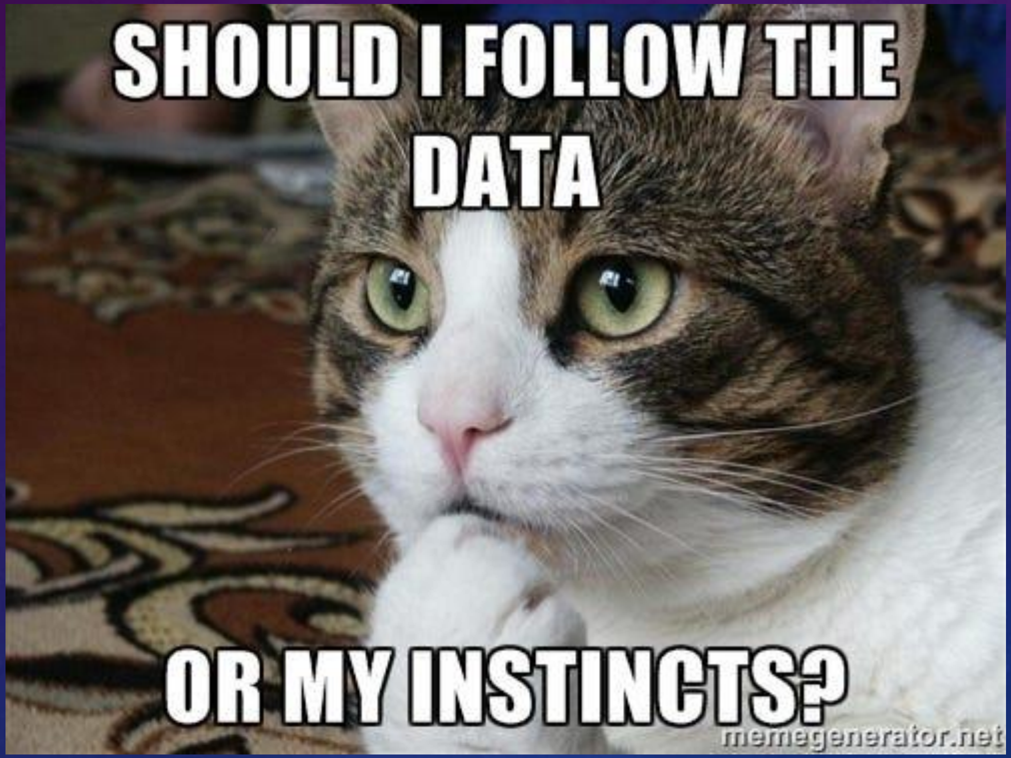




EXAMPLE: FINISHED BOARD



Reference



SETTING PARAMETERS

Common Parameters

Tool change Z: 15.0000

End move Z: 10.0000

End move X,Y: 0,0

Preprocessor: default

User commands

Button 1:

$H;G90;G0 X-207 Y-140

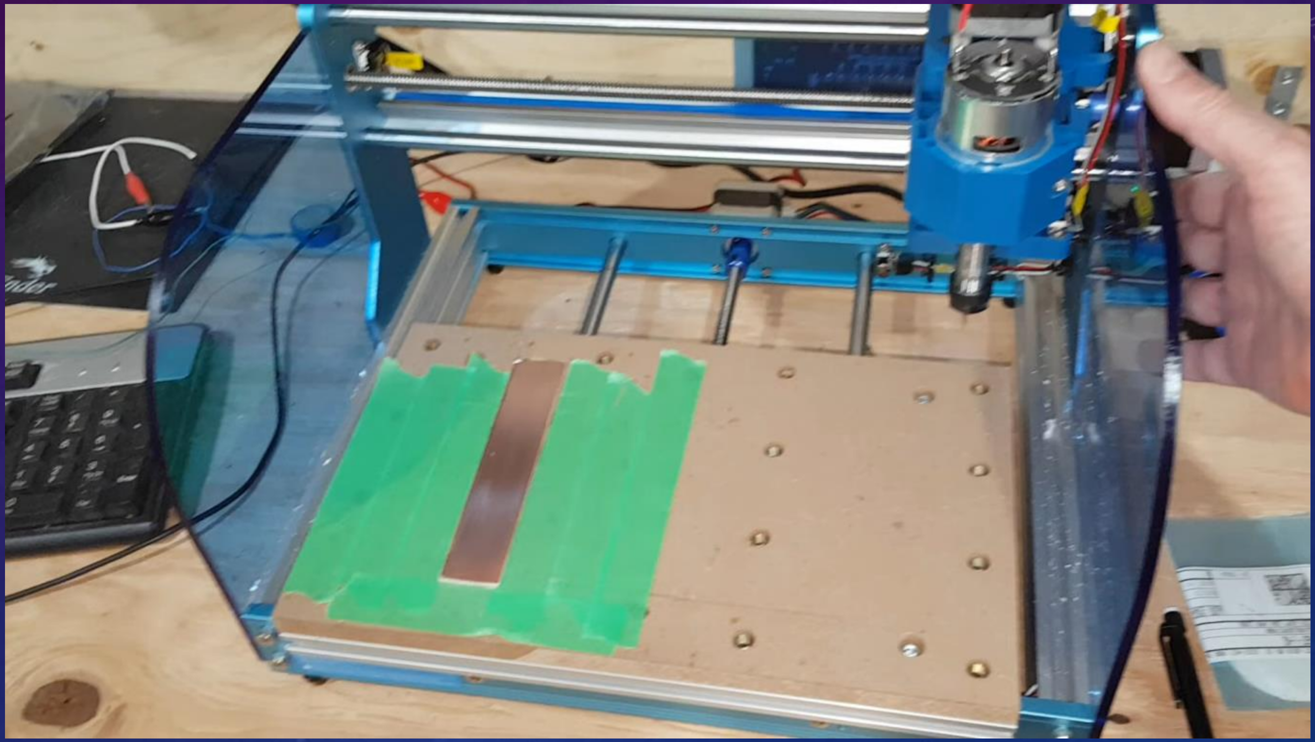
Set as user command button:

$H;G90;G0 X-207 Y-140

..but keep hand on kill switch just in case

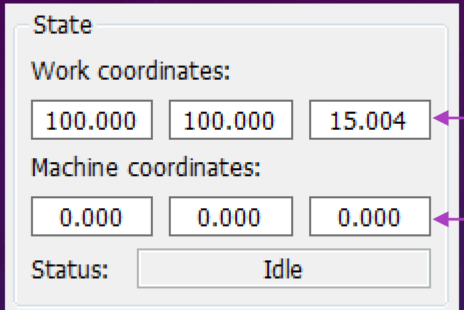
Plot Area | Code Editor

```
G01 X90.9251 Y57.2476
G01 X90.9086 Y57.2400
G01 X90.8859 Y57.2318
G01 X90.8624 Y57.2259
G01 X90.8415 Y57.2227
G01 X90.8174 Y57.2213
G01 X90.7273 Y57.2212
G01 X90.7273 Y57.1898
G01 X90.9602 Y57.1898
G01 X90.9814 Y57.1886
G01 X91.0053 Y57.1851
G01 X91.0287 Y57.1792
G01 X91.0459 Y57.1732
G01 X91.0569 Y57.1687
G01 X91.0759 Y57.1593
G01 X91.0940 Y57.1483
G01 X91.1111 Y57.1358
G01 X91.1292 Y57.1198
G01 X91.1396 Y57.1088
G01 X91.1396 Y57.7784
G00 Z2.0000
G00 X91.6320 Y57.6660
G01 F40.00
G01 Z-0.1200
G01 F60.00
G01 X91.7086 Y57.6660
G01 X91.7102 Y57.6668
G01 X91.7110 Y57.6684
G01 X91.7110 Y57.6950
G01 X91.7102 Y57.6966
G01 X91.7086 Y57.6974
G01 X91.6320 Y57.6974
G01 X91.6320 Y57.6660
G00 Z2.0000
M05
G00 Z2.0000
G00 Z10.00
G00 X0.0 Y0.0
```



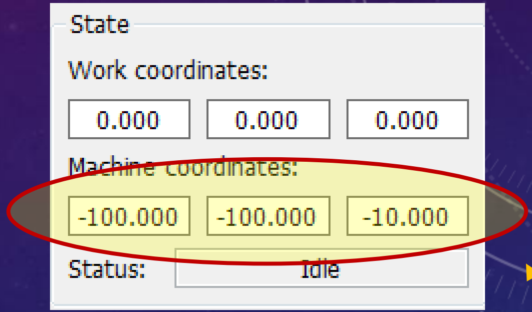
SETTING HOME

Home Position ($23=0)



Apparent position
Relative to last axis zero

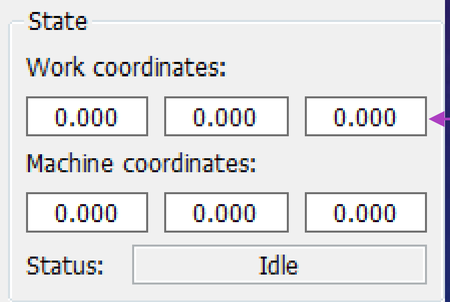
Real position
(0,0,0) is home



Axis Zeroed

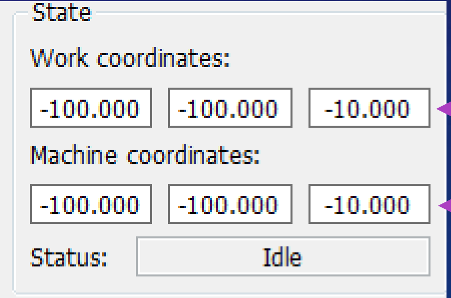
Note this
Home Location

Home Position ($23=0)



Apparent position
Axis Zeroed

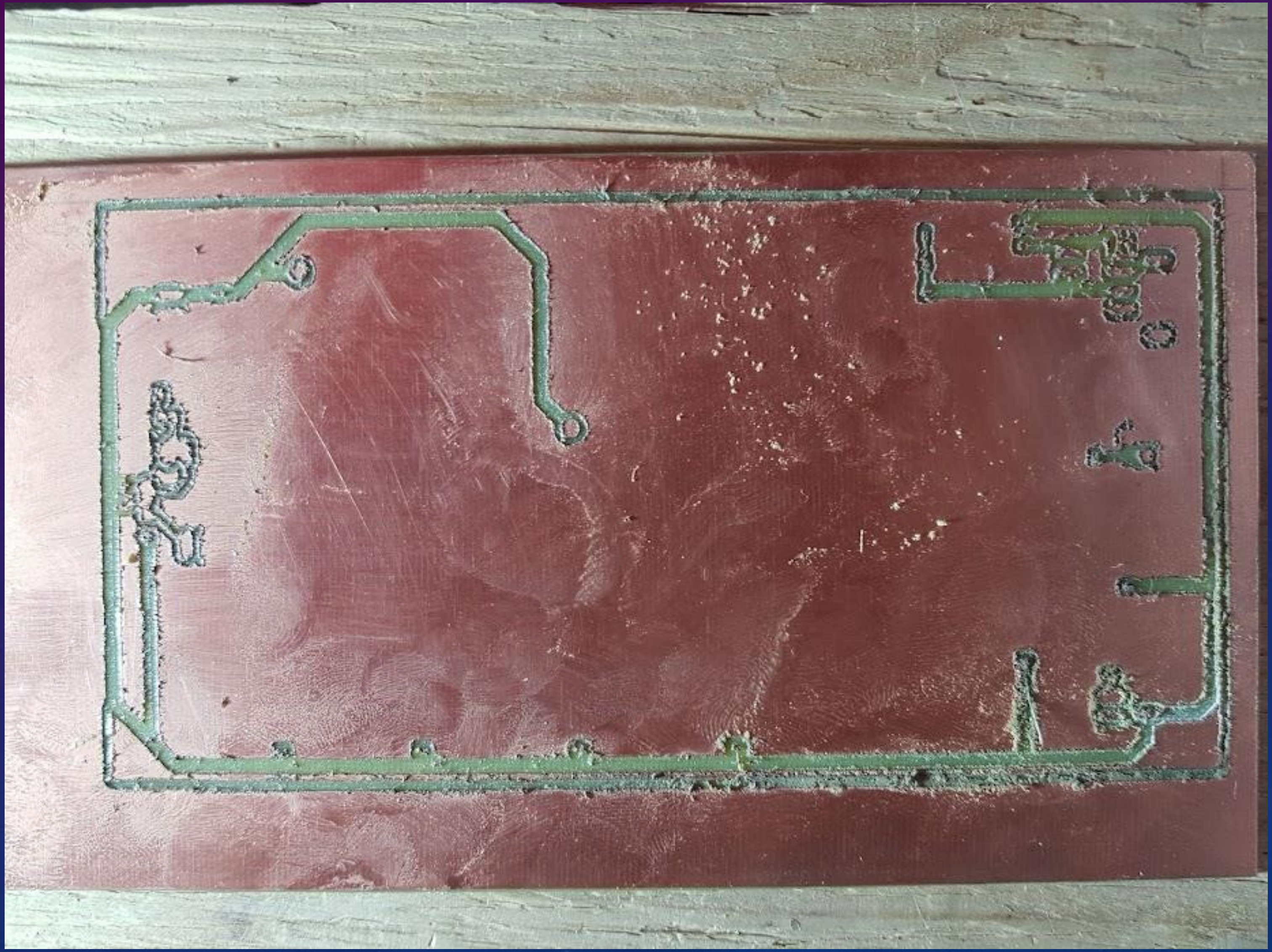
Moved
100mm x 100mm x 10mm
From Home



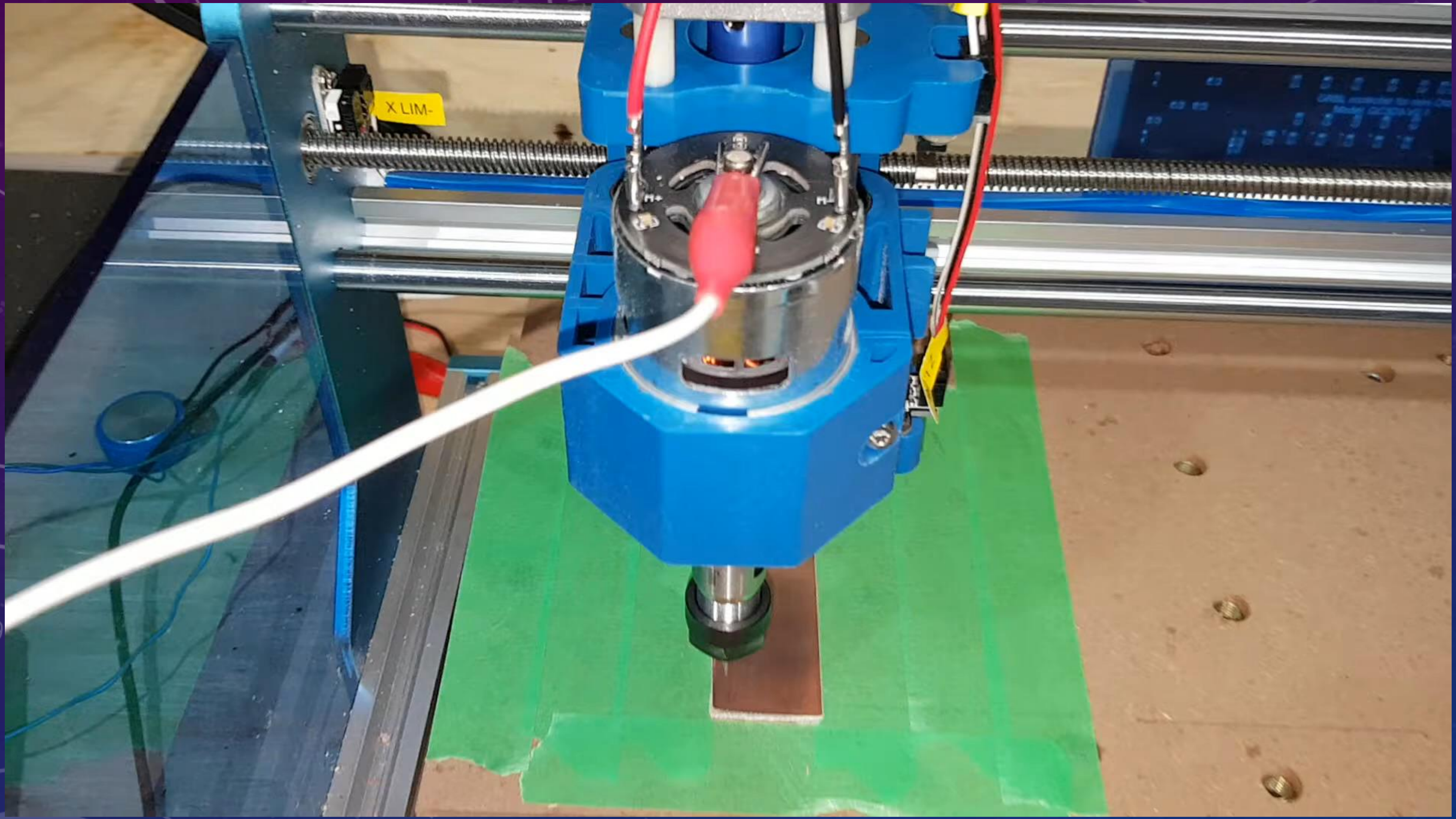
Apparent position
Relative to last axis zero

Real position

CENTER BIT IN COLLET. NO WOBBLE



MAKE SURE YOU CHECK CONDUCTION



Verify probe leads are connected and voltage seen between **PCB** and **Bit**

DIY Machining - GRBL Settings - Pocket Guide

The following settings are specific to GRBL v0.9 - For more information visit www.DiyMachining.com/GRBL

Command	Definition	Explanation
$$	View Settings	Displays current GRBL settings stored in EEPROM (memory) of the Arduino
$0=10	Step Pulse Length (µsec)	This sets the length of the step pulse delivered to the stepper motors. The goal is to have the shortest step pulse your motors can reliably recognize. The data is available on some stepper motor data sheets otherwise 10 is a good default.
$1=25	Step Idle Delay (msec)	Sets the time delay in milliseconds that GRBL will power the stepper motors after a motion command is complete. A setting of 255 tells the motors to stay powered on to hold position.
$2=0	Step Pulse Configuration	Defines the step signal sent to the stepper motor drivers. By default the step signal starts low and goes high to denote a step pulse event. See **Axis Config. Table** below.

Axis Config. Table

Setting Value	Reverse X	Reverse Y	Reverse Z	Setting Value	Reverse X	Reverse Y	Reverse Z
0	NO	NO	NO	4	NO	NO	YES
1	YES	NO	NO	5	YES	NO	YES
2	NO	YES	NO	6	NO	YES	YES
3	YES	YES	NO	7	YES	YES	YES

Command	Definition	Explanation
$3=6	Axis Direction	Changes axis motion direction without changing wiring. See **Axis Config. Table above.**
$4=0	Step Enable Invert	Controls the signal sent to the enable pin of your stepper drivers. $4=1 sets the enable pin to high. (Invert)
$5=0	Limit Pins Invert	This refers to the limit switch pins which by default are set to high using the Arduino's internal pull up resistors. Grounding the pin tells GRBL the limit switch is tripped. For the opposite behavior use the setting $5=1 which tells the system that a high is the limit switch trigger. You must also install external pull down resistor with the $5=1 setting.
$6=0	Probe Pin Invert	This refers to the probe pins which by default are set to high using the Arduino's internal pull up resistors. Grounding the pin tells GRBL the probe is tripped. For the opposite behavior use the setting $6=1 which tells the system that a high is the probe trigger. You must also install external pull down resistor with the $6=1 setting.
$10=3	Status Report	Defines the real time data sent to the user. By default GRBL reports running state which cannot be turned off, machine position & work position. The table to the right details the settings. Note to send a combination of status reports, simply add the values of the desired report types and send this value to GRBL. For Example, say I want Work Position (2) & Limits (16), I would send $10=18.

Report Type	Value
Machine Position	1
Work Position	2
Planner Buffer	4
RX Buffer	8
Limit Pins	16

Command	Definition	Explanation
$11=0.020	Junction Deviation (mm)	Think of this as cornering speed. A high values allows for fast motion around corners but increases the risk of missed steps resulting in decreased accuracy. Conversely, lower values reduce the speed around a corner decreasing the risk of missing steps while potentially improving accuracy.
$12=0.002	Arc Tolerance (mm)	GRBL treats curves as a collection of small straight lines. This setting defines how smooth the curves will be. The default is .002mm and will not likely need to be changed as this value is below the accuracy of most machines.
$13=0	Feedback Units	Sets position feedback units from mm to inches. $13=1 for inches or $13=0 for mm
$20=0	Soft Limits (Enable/Disable)	Requires "Homing" be enabled and checks to see if gCode commands will exceed the travel limits of the machine. $20=1 Enable $20=0 Disable
$21=0	Hard Limits (Enable/Disable)	Requires limit switches be installed and looks for one of the limit switches to be activated which triggers "Alarm" mode. In this mode, all machine motion, the spindle and coolant are shutdown.
$22=0	Homing Cycle (Enable/Disable)	Requires limit switches be installed. Enabling this will lock out all gCode commands until a "Homing" cycle is run.
$23=1	Homing Cycle Direction	Allows the user to change the direction of the homing cycle us the values from the **Axis Config. Table** on page 1.

For more information visit www.DIYMachining.com/GRBL

Double click to view settings

DEFAULT GBL CONFIG

$$ < $0=10

$1=25

$2=0

$3=2

$4=0

$5=0

$6=0

$10=3

$11=0.010

$12=0.002

$13=0

$20=0

$21=1

$22=1

$23=3

$24=25.000

$25=500.000

$26=250

$27=1.000

$30=10000

$31=0

$32=0

$100=800.000

$101=800.000

$102=800.000

$110=2000.000

$111=2000.000

$112=600.000

$120=10.000

$121=10.000

$122=10.000

$130=500.000

$131=400.000

$132=100.000

ok


Fin