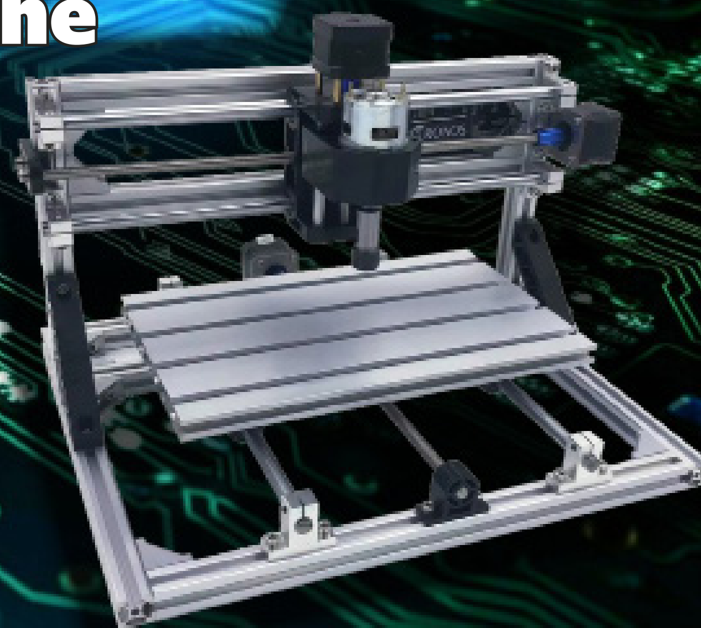
FEATURES

Create Your Own PCBs with a CNC Milling Machine

Using KiCad, CopperCAM, and Candle Software



By
Matthew Oppenheim

A Computer Numerical Control or CNC machine is a computer-controlled device that uses pre-programmed software to cut, shape, and create parts and prototypes. In this article, Matthew demonstrates how to use a small CNC milling machine at home to make a simple two-sided PCB, using KiCad, CopperCAM and Candle software.

A CNC (Computer Numerical Control) machine is controlled by a computer and uses pre-programmed software to cut and shape parts and prototypes. They are used with CAD (Computer Aided Design) and CAM (Computer Aided Manufacture) software. In this article I describe my workflow for manufacturing a simple, two-sided printed circuit board from a blank piece of copper-clad FR4, using a Genmitsu 3018 PROVer V2 CNC milling machine [1].

Why, in the age of cheap PCB production, would anyone take the time to mill out their own PCB at home? I asked myself this question a few times during this project.

I don't suggest that it is practical to mill a fully functional PCB that has a complex design at home, one with many vias, or component footprints with finely spaced surface mount pads. Some artisans do manage to do this and I credit them for it. However, modifying a complex PCB design so that it retains the board outline, mounting holes, and footprints required for the board connectors does not take long, and allows us to manufacture a "looks like" prototype that enables us to check two things:

- That the PCB fits inside the intended case correctly.
- That any connectors on the board align correctly with the openings in the case.

For this article, I milled a complete, two-sided PCB that connects to a LILYGO T-Embed module [2] using 0.1" spaced header pins. T-Embed is an IoT embedded panel designed for programmable development. It provides 3D drawings and reference drawings for products. The completed and populated board is shown in **Figure 1**. I used the completed assembly as a part of an assistive technology project called "handshake" [3]. The project is designed to allow people with limited dexterity to interact with assistive technology devices and software.

The board enables the T-Embed to act as an access switch to operate assistive technology. The standard interface for these assistive technology devices is a 3.5mm plug and socket. When the tip and sleeve of the plug are shorted, a switch closure is detected by the hardware that is being controlled.

The PCB I built has a 3.5mm socket that can short the tip and sleeve of an attached 3.5mm plug, using an opto-isolated relay controlled by the T-Embed. I hope to present a more detailed article on this project in the future. See **Figure 2** for the schematic of the PCB.

The workflow for creating the PCB has three steps, which are discussed in detail in the next sections. They are:

- Creating Gerber and drill files from my PCB design using KiCad [4] software. A Gerber file is a vector image file containing information about a PCB layer.
- Creating G-code files using CopperCAM [5] software. G code is the programming language most commonly used for CNC and 3D printing. CopperCAM manages the drilling, cutting, and engraving of PCB prototypes.)
- Sending the G-code to the CNC using Candle [6] software.

CREATING GERBER AND DRILL FILES USING KICAD

I use KiCad for schematic capture and PCB layout. To prepare the PCB for use with CopperCAM, we need to move or copy the board edge line from the Edge.Cuts layer to the front copper F.Cu layer. Why? It is easier to tell CopperCAM that a trace on the top copper layer is the edge cut than to try and align a separate edge-cut layer with the component layers.

The next step needed is to create Gerber and drill files to import into CopperCAM. To create the Gerber files:

1) Go to the menu option: File, Fabrication Outputs. Plot F.Cu, B.Cu.
2) Unselect "Plot footprint text" (see **Figure 3**).

To create the drill file:

1) Go to the menu option: File, Fabrication Outputs, Drill Files.
2) Tick the PTH and NPTH in a single file option; this option defaults to not being selected. All other options are default (see **Figure 4**).

USING COPPERCAM SOFTWARE

The CNC's spindle is told how fast to spin and how to move, using a language called G-code. CopperCAM software uses the Gerber and drill files that we created with KiCad to calculate the G-code needed to mill out the traces and to drill the holes on the PCB. CopperCAM software is designed for Windows, but will run smoothly on Linux using the Windows emulation software called Wine [7].

A demo version of CopperCAM can be downloaded from galaad.net [8], but is limited

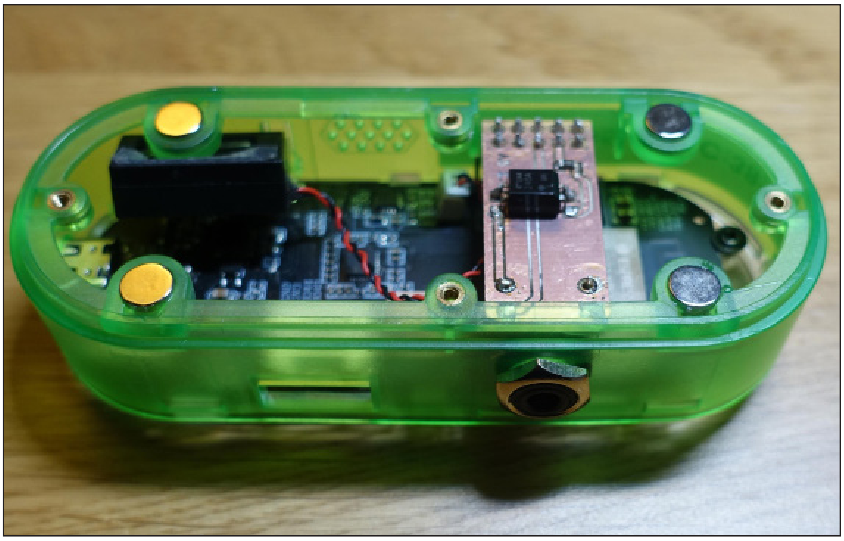


FIGURE 1
Completed PCB connected to a LILYGO T-Embed.

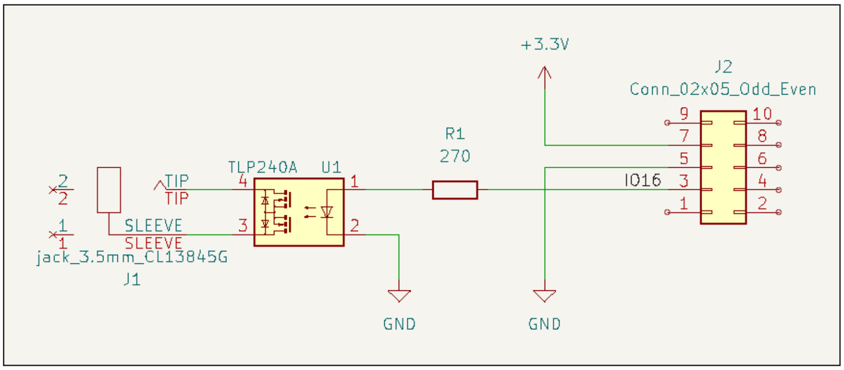


FIGURE 2
Schematic for the access switch adapter board that connects with a LILYGO T-Embed.

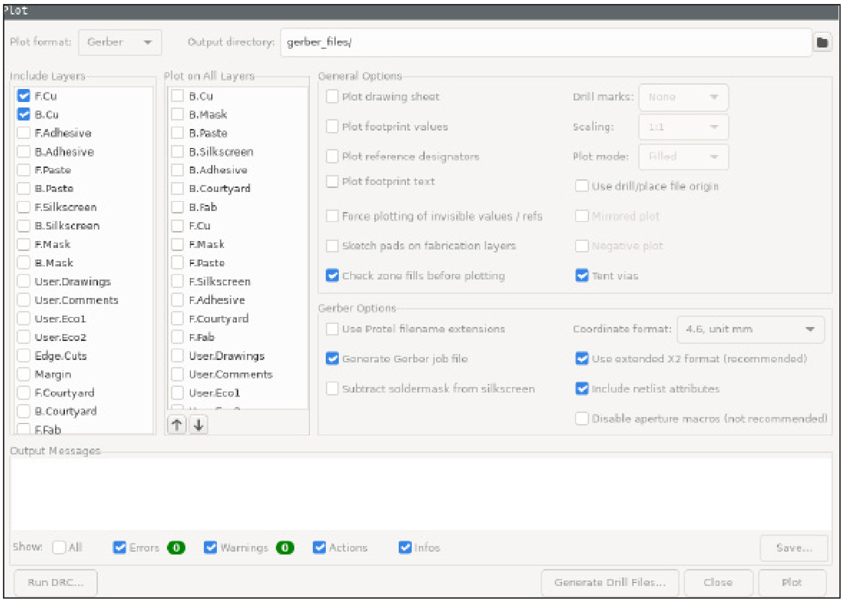


FIGURE 3
KiCad Gerber export options.

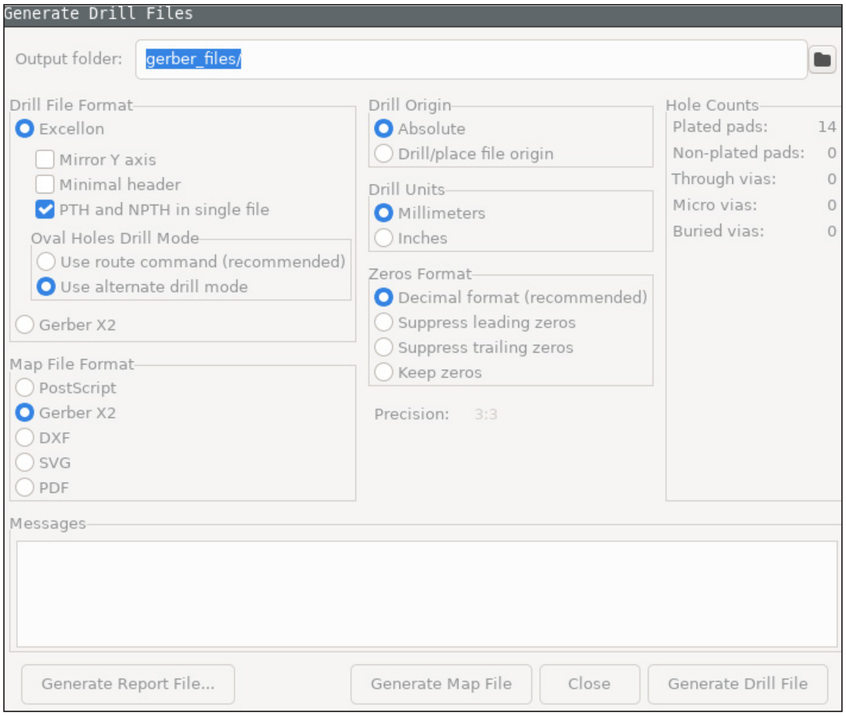


FIGURE 4
Creating drill files using KiCad.

in the number of pads for which the software will produce G-code. Unlimited capability, but time-limited licenses, also are available, so that you can fully test the software. I paid for a license, which was €80 (about $84) [9]. The license is "lifetime use anywhere." The other software used in this article is free and open-source.

I don't begrudge paying for software that works well and allows me to own it, but I'm not a fan of having to pay a subscription for software. CopperCAM is not sponsoring me for this article. A colleague mills out his PCBs using the free bCNC [10] software. I had a look at bCNC, but found it less user friendly than CopperCAM.

INSTALL COPPERCAM USING WINE

I'm including details on how to install and use CopperCAM using Linux, because this is how I did it, and I anticipate that a lot of the readers of this esteemed publication also use Linux.

These are the commands necessary to install Wine [7] on Debian 12, mostly copied from a helpful website [11].

First, we need to install the 32-bit version of Wine. Avoid the Flatpak installation, I couldn't get that to work. These are the commands I used to install Wine:

```
sudo mkdir -pm755 /etc/apt/keyrings

sudo wget -O /etc/apt/keyrings/
winehq-archive.key https://dl.winehq.
org/wine-builds/winehq.key

sudo wget -NP /etc/apt/sources.
list.d/ https://dl.winehq.org/wine-
builds/debian/dists/bookworm/winehq-
bookworm.sources

sudo apt update

sudo apt install --install-recommends
winehq-stable

export WINEARCH=win32
```

For whatever reason, to get the 32-bit version of wine to install, I had to remove the existing ".wine" directory in my home directory. Rename the directory for safety instead of removing it, mv ~/.wine ~/wine_backup

Now install CopperCAM from whatever directory to which you downloaded the .exe installation:

```
wine coppercam.exe
```

The installation process will ask to install .net. Ok this.

To run the CopperCAM software use this command:

```
wine ~/.wine/drive_c/CopperCAM/
CopperCAM.exe
```

CREATING G-CODE

The menu icons along the top bar of

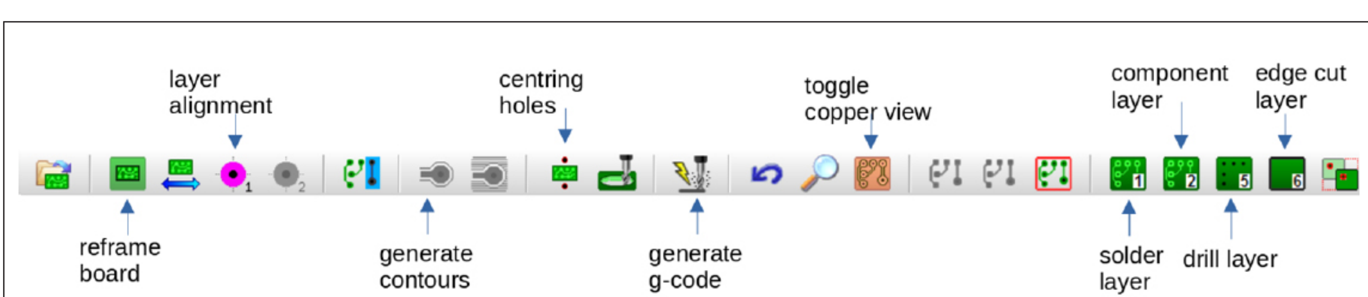


FIGURE 5
Annotated CopperCAM menu bar.

CopperCAM don't have any pop-up tool-tips to clarify what each one does, at least not when running it under Wine on Linux. **Figure 5** is an annotated screenshot of the menu icons, with the tools I used in this article labeled. This is an illustration I wish I'd had when I started!

The best online resource that I found for using CopperCAM to prepare a PCB is a YouTube video, "CopperCAM Tutorial," by Jamesong10 [12]. It is worth watching this video before reading further. The steps are given below.

Import the Gerber and Drill Files: First, we need to import the Gerber and drill files that we created in KiCad.

Go to the menu option File, Open, new circuit.

In the pop-up menu, we need to define which of our Gerber files corresponds with which construction stage of the PCB. Select these options:

Solder side, select the B.Cu Gerber file. This defines what to mill out on the solder side of the PCB, which is probably the bottom side of your board.

Component side, select the F.Cu Gerber file. This defines what to mill out on the component side of the PCB, which is probably your top side.

Drills, select the drill file. I use a single drill bit size and drill out any larger holes using a drill press.

Cut-out, leave blank. This is the outline of the board. Your cut out should be in the F.Cu layer as detailed earlier in this article. We will select the trace to use as the edge cut presently.

If you are using Linux, continuously navigating from your Wine directory to the Gerber files directory can become tedious. My tip is to make some soft links in the CopperCAM .wine directory to three locations: Gerber and drill files directory; CopperCAM project file directory; and G-code directory.

Once you've imported your Gerber and drill files, you will see your design in CopperCAM. Have a good look at all the pad shapes. because sometimes some of these are changed from what you had in KiCad. Right click on any incorrect pads, and select the correct shape and size. From what I've read, there are different standards for Gerber files; this leads to the occasional confusion on pad shape when importing Gerber files.

Right click on the trace that you want to be your edge cut, and select this option as "Define as cutting contour" from the menu that appears. CopperCAM is pretty good at working out what the outline is. Select that the milling tool that cuts out the board will go outside this path, or you'll end up with a smaller board than you anticipated.

If the top and bottom layers become misaligned, the layers can be realigned using the layer alignment tool (Figure 5). Click on a pad on one side of your board, activate the other layer, and click on the corresponding pad on that layer. Watch the YouTube video mentioned earlier [12] for a demonstration of how to do this.

Define Drill Holes for Side 2 of the PCB: The next step is to define where two drill holes go that are used to align the board when you flip it over to mill the second side.

I use M4 acrylic bolts. I found a 4.1mm hole to be the right size. I use acrylic bolts for the unlikely scenario where I miscalculate where the outline edge cut goes, and end up shaving the bolt head with the CNC cutting tool when I cut out the finished PCB. Acrylic doesn't mind having a bit of it shaved off, whereas steel does. Guess how I found this out? I found that 4.1mm holes allow for a snug fit between the hole and an M4 acrylic bolt.

Offset an M4 bolt hole by at least 1.6mm from the edge of the board to allow for the extra radius of the bolt head. This works if you are using a 2mm diameter cutter to route out the board.

Place the two drill holes around about the center of your board. Then accurately recenter the board around these holes by using the menu option "Machining, Add centering holes, Recenter circuit."

Now define the bottom hole as your coordinate origin by selecting the Reframe button on the menu.

Click on the crosshairs icon, select "Plot manually," then click on the bottom hole. This hole should flash red. In earlier versions of the software, a white cross was plotted over the hole. In the version of CopperCAM that I used for this article (Version: 20/09/2024), the white cross did not appear on the display.

If we set up the two drill holes in this way, when the board is flipped to machine the second side, the board is mirrored around the origin. This means that the origin is in the

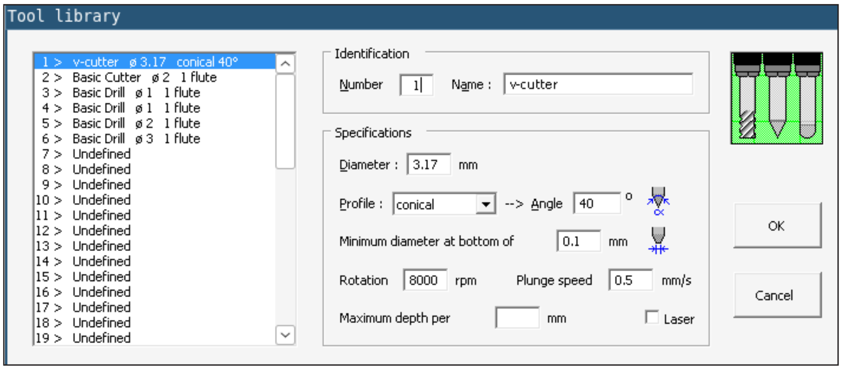


FIGURE 6
Example of CopperCAM tool library setup.

same place for machining both sides of the board. If we use four holes and define, for example, the bottom left hole as the origin, after flipping the board, the origin will move.

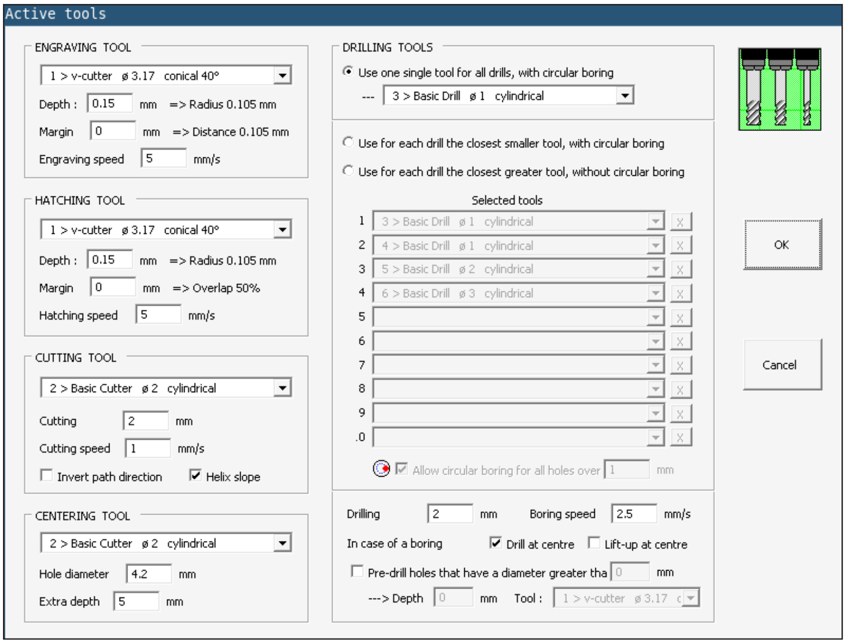


FIGURE 7
Example of CopperCAM tool settings.

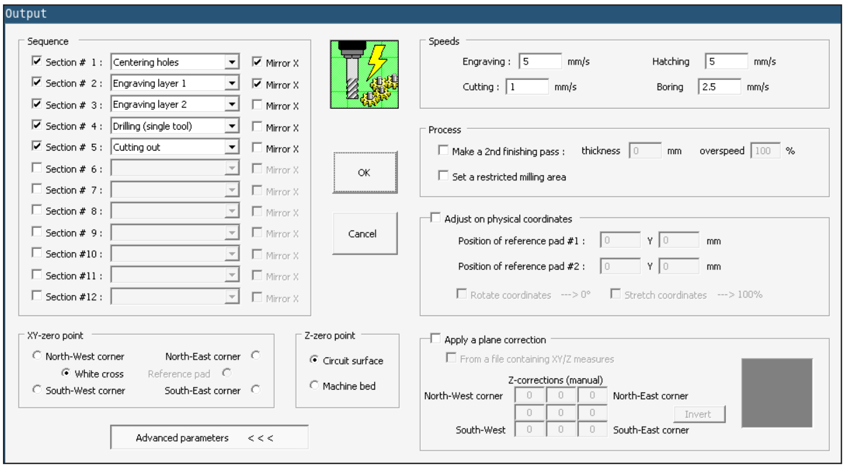


FIGURE 8
CopperCAM screenshot showing G-code setup.

Unless you keep track of this and redefine the origin for each side of the board, the traces will be not be milled on top of each other. Guess how I found this out?

Define the Settings for Milling and Drilling: Now we need to define the settings for each of the tools that are used to mill and drill the PCB.

Go to the menu option Parameters, Tool library. To create a PCB we need an engraver for milling out the traces and pads, a cutter for cutting out the board and at least one drill for drilling through holes. I use a 40° conical cutter for milling out the traces and pads, a 2mm cylindrical basic cutter for cutting out the PCB and a 1.0mm drill for all holes. Please see **Figure 6** for a screenshot of my Tool library settings. Obviously, you need to set this up with the parameters of the tools that you use.

Assign Tools to Stages of PCB Manufacture: We now need to assign these tools to each of the stages of PCB manufacture. In the menu, go to Parameters, Active tools. The first time that you use CopperCAM, the options will have default settings. For the little CNC that I own, the default movement speed for each of the tools is too high, leading to poor results. I at least halve the default movement speeds. The Engraving Tool, Depth value is critical. This defines how deep the tool will cut to mill out you pads and traces. Too shallow and there will be a short between your trace and the surrounding copper. Too deep and the pad shapes will be distorted and the traces may be completely milled out. I tested different values and settled at 0.15mm for the v-cutter that I use. A screenshot of the settings that I use is shown in **Figure 7**.

Create G-Code for CNC Spindle: Now, we need to create the G-code that tells the CNC spindle where to move and how fast to spin.

Click on the Generate G-code icon (Figure 5). The first time you use this option, a screen comes up with information about the G-code standard that is to be used. Just click on OK.

In the next screen, we define and export G-code for each of five construction stages. I created a screenshot showing all five stages

LISTING 1
Bash script to edit G-code produced by CopperCAM, so that it runs on Candle software.

```
# Clean g-code so that it is accepted by Candle software
# run ./clean_code.sh <filename>
# the input is edited in place
#!/bin/bash
# Use the sed command to edit the file
sed -i '
# Remove lines starting with the % symbol
/^%/d
# Remove lines starting with the T symbol - these are tool definitions
/^T/d
# Remove lines starting with the ( symbol - these are comments
/^(/d
# operate on the file name supplied at run-time
' "$@"
```

at once (**Figure 8**). Only select one sequence at a time when you create your g-code.

Things to check when exporting the G-code:

Click the "Mirror x" for the first two stages, which are the stages that create the centering holes and engrave the bottom layer.

Click on "White cross" for the XY-zero point. The white cross is the bottom drill hole that we defined as the coordinate origin earlier. Note that the version of CopperCAM I used for this article did not display this white cross, though an earlier version of the software does.

Select each construction stage shown in Figure 8, then click "OK." I use the default naming convention for the G-code files, since this name includes the manufacturing step number and a brief description of what that step is.

Bash Script for Editing G-Code File: Now we have the G-code to run our CNC. Almost. As with Gerber files, there is some flexibility as to what valid G-code is. I had to edit the G-code files that CopperCAM outputs using a small bash script to make it run on Candle software (**Listing 1**). This script uses the `sed` command to edit the G-code file.

Run the script with the G-code file as an argument. To run this script on all G-code files in the directory containing the script, use the bash command:

```
find . -name ‘*.iso’ -exec ./clean_
gcode.sh {} \;
```

This will also run the script in any sub-directories. To limit the script to the current directory use this bash command:

```
find . -name ‘*.iso’ -maxdepth 1
-exec ./clean_gcode.sh {} \;
```

```
# Clean g-code so that it is
accepted by Candle software

# run ./clean_code.sh <filename>

# the input is edited in place

#!/bin/bash

# Use the sed command to edit the file

sed -i ‘

# Remove lines starting with the %
symbol

/^%/d

# Remove lines starting with the T
symbol - these are tool definitions

/^T/d

# Remove lines starting with the (
symbol - these are comments

/^(/d

# operate on the file name supplied
at run-time

‘ “$@”
```

Listing 1: Bash script to edit G-code produced by CopperCAM, so that it runs on Candle software.

We now have the G-code to send to our CNC to mill and drill our PCB. Before milling and drilling, however, we need to set up the CNC milling machine.

CNC MACHINE SETUP

My father trained as a machinist, and kept a traditional lathe and milling machine in his basement. I spent some hours learning the basics of machining using them. Setting up and clamping the material in the milling machine can be one of the trickiest steps. An old friend of mine, who is now a physics professor, has a much larger and more powerful CNC than I do. I spent one happy morning helping him build a laser on his kitchen table, by attaching various components to an aluminium base plate he machined in his garage with his CNC.

I now follow his advice of sticking the blank PCB to a flat piece of 5mm acrylic with

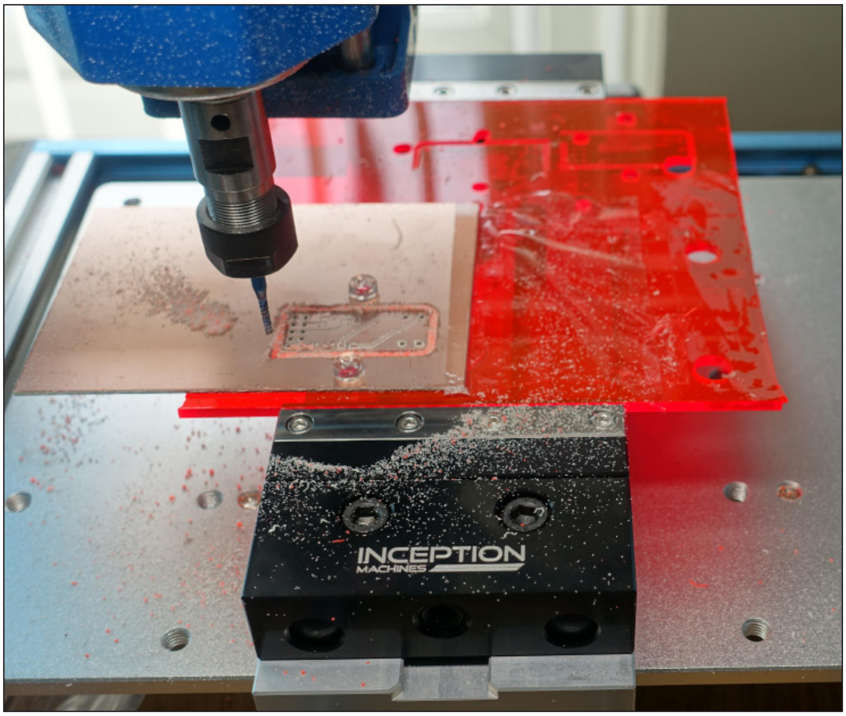


FIGURE 9
PCB after completing milling.

double-sided tape. This keeps the PCB flat, so the milling tool stays at the same cutting depth when the traces are cut out. I hold the acrylic sheet in a low-profile vice [13]. It is possible to secure the acrylic sheet to the base plate of the CNC, but it is a bit more fiddly than using a vice, and there is a risk of running the cutting tool into clamps that protrude above the surface of the PCB.

Make sure to put some double-sided tape under the milling area of your PCB before you machine the second side, because it is the only was you can secure the board when you do the final edge cut process. **Figure 9** shows a completed PCB ready to remove from my Genmitsu CNC. The board is attached with double-sided tape, as described above.

Take a minute to think through the necessary safety precautions when running a CNC. Always wear safety glasses. If your tool path is incorrect and the tool collides with a clamp or the base plate, it could shatter. The dust kicked up from carving the FR4 in a PCB is basically powdered glass, which you don't want in your lungs, so wear a filter mask. Use a vacuum cleaner to clean up the dust. Keep an eye on the machine at all times when it is operating, and be ready to hit the emergency stop.

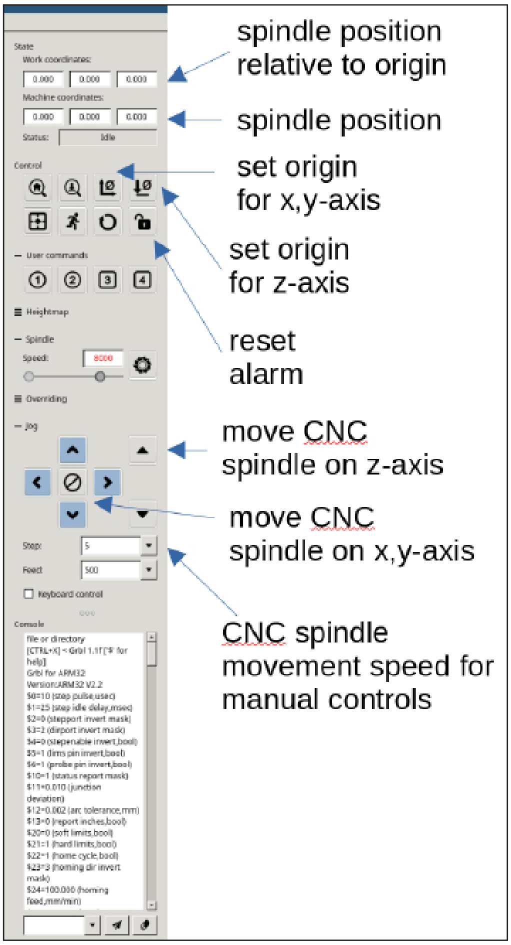


FIGURE 10
Screenshot of Candle software, with CNC spindle controls annotated.

USING CANDLE TO CNC MACHINE

Candle software sends the G-code to the CNC through a USB cable. The software shows the path that the tool tip will take for the G-code file. A panel on the right side of the screen allows you to control the CNC spindle position and to set the origin for either the x, y coordinates or the z position of the your tool-tip. The x, y origin and the z-axis origin controls tell Candle where the origin point of the G-code file will physically be on the PCB. **Figure 10** shows these controls with labels. It is important to get the origin point right, or you'll end up milling somewhere you didn't want to, potentially driving your tool into the vice or CNC bed. Guess how I found this out!

To set the x, y axis origin position, manually move the tool tip to where you want the bottom hole drilled for your alignment holes. This is the position we defined as the coordinate origin in CopperCAM. Click on the x, y origin button on Candle. You will see the x and y coordinates in the top row on the right hand panel go to zero.

The origin position for the z-axis is where the tip of the tool touches your PCB. Obviously, you need to change the tool in between some of the milling stages, so you will have to reset the z-axis origin after each tool change. Use the z-axis knob on your CNC to lower the tool until the tip just touches your PCB. How do you know when this is? Use a multimeter with two bulldog clips, one on the tool and one clipped onto your PCB. The instant you get continuity on the resistance setting, you know you have the tool tip at the z-axis origin. Click the z axis origin button on Candle. You only need accurate z-axis positioning for the milling stages. For drilling and cutting out the board, lower the tip of the tool using the manual control on your CNC until you feel it touch the PCB.

Important: Once you have set the z-axis origin, use Candle to raise the tool. The first operation during milling, cutting, or drilling can be to move the tip of the tool over your board without raising it, which will leave a long scratch on your board. Guess how I found this out? Using Candle to raise the tool means that the software will move the tool, then lower it to the board surface to start machining.

To change the tool and to reset the z-axis origin without losing the x, y positioning, use Candle's controls to move the spindle somewhere that you can change out the tool, and then change it. Use Candle's controls to move the tool tip back over your PCB milling area. While you move the spindle using Candle software, the position of the spindle remains

Additional materials from the author are available at:
www.circuitcellar.com/article-materials
Resources [1] to [13] as marked in the article can be found there.

known to the software. After changing the tool, move it back to around the center of your PCB milling area, then manually reset the z-axis origin as detailed above.

RECOVERING FROM MISTAKES

If you lose the positioning of your spindle, for instance by manually moving the spindle until one of the end stop alarms is triggered, you can recover. Manually move the spindle until the tip of your tool is just above the center of the bottom drill hole for your alignment bolts. Clear the alarm using the control on the Candle menu. If you use a cross-head bolt, then the center of the cross on the bolt head will be your x,y origin. You can recover the z-axis origin using the same method used each time that you replaced the tool, as discussed above.

Check that milled traces are isolated from the surrounding copper while the PCB is on the CNC, using a multimeter on the continuity setting. If you find continuity, then the milling is not deep enough. You can reset the z-axis origin so that the cutting tool is placed a little more firmly against the PCB, then rerun the milling stage.

If your tool breaks, replace it, reset the z-axis origin and rerun the G-code for that machining step.

LESSONS LEARNED

I made mistakes. It turned out that the contacts on one side of my 3.5mm audio socket disconnected when a 3.5mm plug was inserted. I hadn't accounted for this, so I needed to update the footprint and routing for the socket. I needed to lengthen the board, so that the 3.5mm socket protruded enough to allow for the retaining nut to go over the thread on the socket. This created mechanical stability and reduced the chance of the solder joints on the PCB cracking when a plug was inserted into the socket.

I learned a lot during this project, and have an increased respect for the PCB manufacturers who produce flawless PCBs at a fraction of the price I had to pay when starting out in electronics. I hope that this article is of interest, and that I have encouraged you to give milling your own PCB a go.

ABOUT THE AUTHOR

Matthew Oppenheim works in marine geophysical survey and offshore construction. When not working at sea he loiters at InfoLab21, Lancaster University, where he works on assistive technology projects and how to deploy them into the real world. Visit www.mattoppenheim.com for more details of these projects. Please email matt@mattoppenheim.com with any comments.