

Setting up a desktop CNC router

Jack Summers, PhD

Smoky Mountain Scientific
4989 Tilley Creek Road
Cullowhee, NC 28723

Department of Chemistry and Physics
231 Natural Sciences Bldg
Western Carolina University
Cullowhee, NC 28723

Introduction: A CNC router is a very handy tool for the DIY Maker since it will allow you to accurately cutting parts out of wood, plastic or soft metals. We recently purchased a kit and built a CNC router. This manuscript records some of our experiences and observations we had setting up a CNC router.

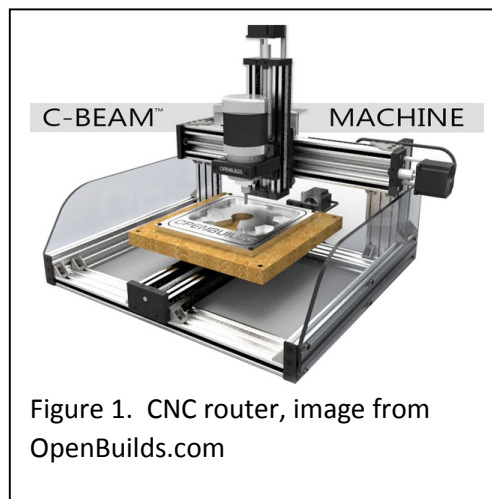


Figure 1. CNC router, image from OpenBuilds.com

- Costs.** I think the common view of CNC (computer numeric controlled) routers is that they are very expensive and difficult to build and operate. While the costs have dropped substantially recently, they are still not insignificant. Two available hobbyist level machines are the Shapokoⁱ and C-Beam Machineⁱⁱ. My experience is with the C-Beam Machine, designed and sold as a hardware kit by Open Builds. The kit we purchased has a 12" x 12" build surface and cost about \$610 USD (including three NEMA 23 stepper motors). In addition to the hardware kit, we purchased a reconditioned router (Bosch Colt), a collet to hold 1/8" bits, and a number of end millsⁱⁱⁱ. We also bought electronic hardware from RepRap Champion, including a package that includes an Arduino UNO clone, motor shield and four motor drivers,^{iv} and a 24 volt power supply^v (We chose the 24 volt version over the 12 volt version to use the lighting accessory from Open Builds^{vi}). All together, we spent about \$800 USD to build what I think is a pretty good CNC mill. I recommend budgeting another \$100 to \$150 to cover end mills, tools, electronics, plastic, plywood, etc.
- Complexity.** Complexity is definitely an issue. You need to have some mechanical aptitude, some familiarity with electronic software, and a fair bit of patience. Almost everything you need to know is described on (or linked to from) the Open Builds website.
 - Mechanicals:** The C-Beam Machine kit has pretty good video showing assembly.^{vii} While the tendency will be to rush through the video, there are many points where order and direction matter and it is easy to get things wrong. I recommend you go through the video slowly, pausing it while you work on the build. You want to get things done right the first time. Read the part of this note about rigidity below (section 3) and keep these cautions in mind while assembling your kit.
 - Software:** Of course, having the mechanicals put together is just the tip of the iceberg. The software is really the heart of complexity. We use the following freeware:

- i. **Firmware:** The microcontroller firmware for Arduino clones is called grbl. Grbl is at the heart of open source 3d printers. It is pretty good. You can download and flash grbl onto your Arduino UNO (or clone) following available instructions.^{viii}
 - ii. **User interface:** We use grbl panel.^{ix} Works well, no problems. Allows you to change parameters, jog the tool, open G-code and run the tool through the interface.
 - iii. **CAD/CAM software:** Computer aided design (CAD) and computer aided manufacturing (CAM) programs can be quite powerful. They can also be quite expensive. We don't do enough manufacturing to justify the expense of professional grade software so we make do using freeware. We use Sketchup Make for general design work and EagleCAD for pcb designs. To use Sketchup, you will need the SketchUCam plugin.^x To generate G-code from Eagle designs you need to download and install the pcb-gcode plugin^{xi}. You should know that G-code is bone-head simple and can be edited with any text editor. Look around the internet and you will find a number of resources that will get you up to speed on G-code.^{xii}
3. **Rigidity.** The accuracy of your CNC machine can only be as good as the rigidity of your machine. If you can move the cutting bit without the drive screws turning, you will not get straight line cuts or round hole cuts. Problems can arise from stripped threads or loosened bolts. When screwing into tapped aluminum, stripping holes can be a problem. The machine will generate very high frequency vibrations. While we have not experienced this problem (yet), I suspect that bolts will want to loosen with time. In general, we recommend tightening things down as much as possible, using nylon lock nuts whenever possible, and the liberal use of blue lock-tight^{xiii} whenever nylon lock nuts cannot be used.
4. **Holding things down.** Cutting a piece of aluminum with a router involves a certain amount of lateral force. Therefore, you need to hold the piece down to the cutting table while working on it.

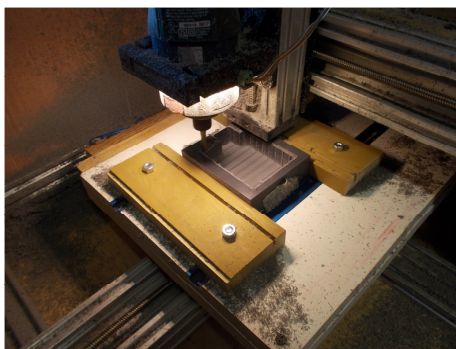


Figure 2. Wood blocks bolted to slotted track hold this work piece.

Double sided tape is something that you will see a lot of people use. While I still use it for some applications, double sided tape is not something I recommend for all cases. First, it comes with a lot of glue. When you cut into it, the glue gums up your end mills (bits). Second, the tape has a limited amount of stickiness and will release, usually at the time most likely to cause irreparable damage to the piece, or the end mill. Tape is especially unwilling to stick to a work surface that has some sawdust on it. I have seen a report where someone used two pieces of masking tape and superglue to hold the work piece down. The user put one piece on the work piece and another on the work surface then applied glue to one side and catalyst to the other to stick them together nearly instantaneously. I have not tried this method, but the author was adamant that it worked well and I have no reason to doubt him.

Right now I am experimenting with hold-downs based on aluminum T-track. Figure 2 shows a method I used while milling a plastic box to hold the electronics for this mill. As shown in this figure, we cut dados in our work surface to accept two T-tracks. The tracks accept $\frac{1}{4}$ " bolt heads and the bolts are used to hold wooden blocks that, in turn, hold the work piece. Since we did not intend to mill through the piece, we used double sided tape to assure the piece stayed in place.

5. **Dealing with swarf.**

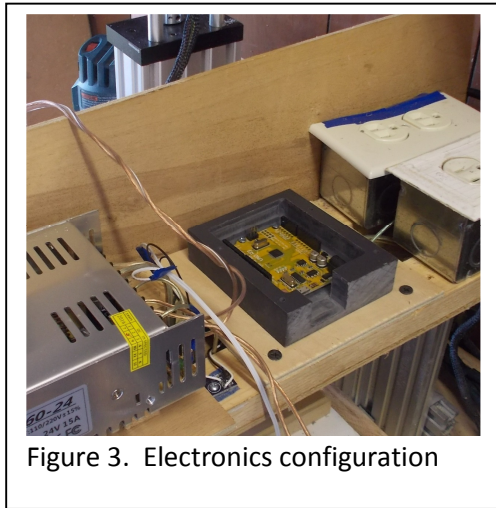


Figure 3. Electronics configuration

While I am not particularly squeamish about breathing sawdust, paint fumes or industrial chemicals, keeping swarf out of the electronics is an absolute must. Swarf is like sawdust but it is composed of the material you are milling. We found that milling aluminum without dealing with swarf is akin to throwing a handful of metal chips into any other electronics project. We are working on two strategies for dealing with swarf, isolation and vacuuming. Figure 3 shows the configuration of our electronics zone early in our build. In the top left, you can see the top of the router. The electronics are positioned up and away from the swarf zone. They are separated from the router by a piece of $\frac{1}{4}$ " plywood. While this configuration did a good job of keeping swarf out of the electronics, we continued to build a more extensive enclosure, as described in the next section. Right

now, vacuuming swarf involves me holding the nozzle of my shop-vac. We plan to investigate using shields to isolate the swarf and vacuum it up from a static position. That's in the planning stage.

6. **Noise.** Routers make a lot of noise. Vacuum cleaners make a lot of noise. So do hound dogs. While there is not currently a plan to incorporate hound dogs in this project, it does use a router

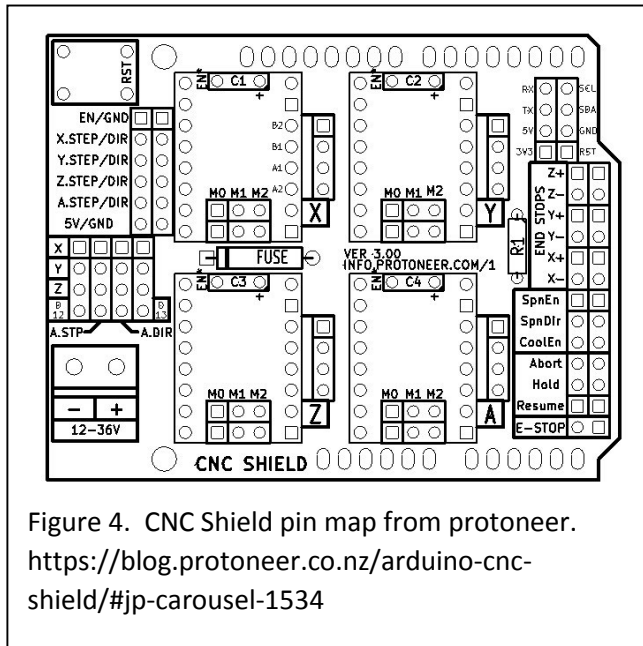


Figure 4. CNC Shield pin map from protoneer. <https://blog.protoneer.co.nz/arduino-cnc-shield/#jp-carousel-1534>

and sometimes a vacuum cleaner. To isolate the sound of the router I built a plywood box to contain the router and lined the inside with Styrofoam insulation. Building the box was a pretty extensive project and I am not sure the gains in sound justify the effort. Perhaps it would be better to operate the router in the garage instead of the basement.

7. **Router switching.** You may want to consider adding hardware to switch your router on and off automatically. It may not seem like a big deal to switch off the router after the project is milled, and the first few times you run it you will want to watch while it cuts out the design you made. There will, however, come a time when you have a two-hour milling job and you have other things to do besides babysit your project (that and the sound of that

screaching router is making you nuts). Since you don't want to leave the router running for the entire work day, you will want a switch to automatically turn the router (and perhaps the vacuum) off.

The grbl program can be set up to control the spindle in either of two ways. The default setting assumes you can control spindle speed with a pwm pin. The alternative setting assumes your spindle only has on and off settings. You can use either to control your router. If you use the default method, you will need to (1) specify a spindle speed every time you power up the tool and (2) switch the router using the Z+ end stop pin (top right in Figure 4) rather than the SpnEn pin (middle right in Figure 4). Using the alternative method, you will not need to specify a spindle speed and you will use the SpnEn pin from Figure 4 (the silkscreen on the CNC shield assumes this option). The down side is that you will need to modify the grbl code (as described in Appendix 1). We should note that specifying the speed should not be a problem when you run a bunch of G-code written by a program generated by sketchucam, which sets a spindle speed for you. It only becomes an issue if you want to turn on the motor and jog the router using the grbl panel user interface. In this case, you will need to set the spindle speed manually by typing "S1000" into the Grbl Panel manual command text box and hit enter. Once the S value has been entered, you should be able to use the spindle enable dropdown on the right side of Grbl Panel to turn the spindle on and off.

Either way you chose to switch the spindle, the motor is supposed to turn on when your Spin Enable pin goes high (+5 volts) and trun off when the pin is low. To switch a large AC load like the router, you will need a solid state relay rated for at least 20 amps. We used one from Sparkfun.^{xiv} Figure 5 shows the solid state relay mounted in an outlet box and wired to a standard electrical outlet. The outlet goes in



Figure 5. Solid state relay allows software control of the router.

the same box, on top of the relay (for the wiring diagram, see Figure 6). While the comments section of the Sparkfun page indicate that this relay can be switched using an Arduino, that did not work for us. Perhaps we got a defective relay (or a defective Uno clone), but the relay would not switch with the current sourced through the IO pin of the Arduino clone. If you have this problem, I recommend using an NPN power transistor to switch the solid state relay. The transistor has three terminals: connect the Arduino pin to the gate, connect the emitter to the ground terminal of your 24 volt power source, connect the collector terminal to the negative control terminal of the relay, solder a 10 Kohm resistor to the positive control terminal and run a wire from the other side of the resistor to the +24 volt terminal of your power source. That is not what we did^{xv} but it should work.

the same box, on top of the relay (for the wiring diagram, see Figure 6). While the comments section of the Sparkfun page indicate that this relay can be switched using an Arduino, that did not work for us. Perhaps we got a defective relay (or a defective Uno clone), but the relay would not switch with the current sourced through the IO pin of the Arduino clone. If you have this problem, I recommend using an NPN power transistor to switch the solid state relay. The transistor has three terminals: connect the Arduino pin to the gate, connect the emitter to the ground terminal of your 24 volt power source, connect the collector terminal to the negative control terminal of the relay, solder a 10 Kohm resistor to the positive control terminal and run a wire from the other side of the resistor to the +24 volt terminal of your power source. That is not what we did^{xv} but it should work.

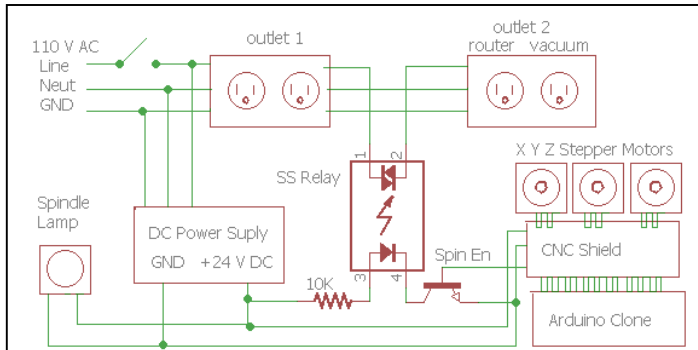


Figure 6. Wiring schematic. When switch at the top left is closed, all 24 volt DC electronics power up and the spindle lamp turns on. Power is supplied to the router and shop-vac on outlet 2 when Spin Enable pin is high. Note that the identity of the Spin Enable pin is either labeled Z+ or SpnEn in Figure 4, depending on whether VARIABLE_SPINDLE is enabled in grbl.

8. Lighting. When preparing to mill something, you want to align the cutter with some point on the piece. This means that you want to be able to see what you are doing. While you could rig a bunch of leds in the work area we purchased the spindle lamp from Open Builds. Notice the lighting of the work piece in Figure 2. Note that this accessory requires 24 volts to operate. We were curious to see if it would give some light with a 12 volt power source. It does not. While I suspect that the leds could be made to run on 12 volts by switching out the resistors on the bottom of the accessory, we did not try that.

Appendix 1, modifying grbl to switch without supplying spin rate: Download and open a free C++ editor like notepad++.^{xvi} Before starting, change the name of the file you are going to modify, so if you screw it up you can reset. Do this by navigating to Arduino\libraries\grbl and renaming the config.h file. In Notepad++, click on files-open and navigate to your Arduino\libraries\grbl folder and select what was once config.h. On or around line 339 there should be code that says “#define VARIABLE_SPINDLE // Default enabled. Comment to disable.” Comment out this line by adding // to the beginning. Now save the file as config.h and you are done. You should be able to compile the grblUpload program and flash it to your Arduino. If the program does not compile, you probably screwed up the config.h file.

Appendix 2, how we actually got the relay working: We had a number of boards for a small solid state relay laying around so we used one of them to switch the big relay. You can have the boards fabricated really cheaply by OSH Park. Three copies will set you back \$1.55. Go to oshpark.com and click on Shared Projects. There will be a google search box where you type in killSwitch2Gerbers.zip. That will get you to a screen where you will be able to order the pcb shown in Figure A2. In addition to the circuit board, you will need the small relay, which has the part number CPC1017NTR and a 500 ohm surface mount resistor (603 package). Solder the small relay to the cpb with the dimple above the white dot on the board. Solder the resistor to the pads where it says 500 Ohm. Solder the control wire to the wire pad marked KILL. Connect one of the wire pads marked GND to the digital ground (return wire for the Spin enable). Connect one of the wire pads marked RE to the low control side of the large relay. Connect one of the wire pads marked CE to the ground on the DC power supply. Be sure to include the resistor between the +24 volt supply and the positive control connection of the large relay (Figure 6). The small relay is rated to handle 100 mA, so the resistor needs to be at least 240 ohm. We used a 10 K ohm resistor and it worked fine.

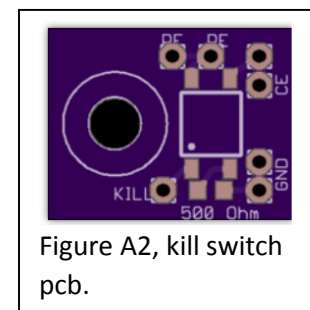


Figure A2, kill switch pcb.

ⁱ <http://carbide3d.com/shapeoko/>

-
- ⁱⁱ <http://openbuildspartstore.com/openbuilds-c-beam-machine-1/>
- ⁱⁱⁱ There are a lot of places on the internet where you can buy endmills. For cutting aluminum, we use 1/8" diameter 2 flute, single end carbide end mills with a square end. If you look around you can find these for less than \$6 each. One place that has a good searchable web site is drillsandcutters.com
- ^{iv} <https://reprapchampion.com/collections/electronics/products/arduino-cnc-v3-shield-for-mill-router-engraver-robot-4x-drv8825-uno-r3>
- ^v <https://reprapchampion.com/collections/electronics/products/12v-dc-30a-360w-power-supply-reprap-3d-printer-led-strip-light-cnc-roboics>
- ^{vi} <http://openbuildspartstore.com/spindle-router-led-light-ring/>
- ^{vii} <https://openbuilds.com/builds/c-beam%e2%84%a2-machine-plate-maker.2020/>
- ^{viii} <https://github.com/grbl/grbl/wiki/Flashing-Grbl-to-an-Arduino>
- ^{ix} <https://openbuilds.com/threads/grbl-panel-control.3037/>
- ^x <https://openbuilds.com/threads/ask-grbl-cnc-questions-here.3000/page-13#post-46602>
- ^{xi} <http://www.pcbgcode.org/>
- ^{xii} <https://makezine.com/2016/10/24/get-to-know-your-cnc-how-to-read-g-code/>
- ^{xiii} Lock-tight is a glue that binds metals together. It comes in two colors: blue is for things that you may want to disassemble someday, red is for things that you want to join permanently. You buy it at your local auto parts store. Not all parts stores will have it, so call around.
- ^{xiv} <https://www.sparkfun.com/products/13015>
- ^{xv} We had a bunch of low current switches laying around from another project so we used one of those. See appendix 2 for details.
- ^{xvi} <https://notepad-plus-plus.org/download/v7.5.1.html>