Circuit Board Tutorial

zjs

June 23, 2014

I thought it would be a good idea to learn how to make circuit boards to help make higher quality electronics projects for the group. This paper will be documentation of the process to hopefully lessen the learning curve for group members that would like to do pursue such projects in the future. This tutorial will use Eagle.

Making printed circuit boards (PCBs) offers a number of advantages including: creating more reliable, more compact, better behaved circuits with ostensibly fewer gremlins and hard-to-track-down noise sources. Obviously this would be more of an issue/potential for improvement for high frequency circuits. In addition, once a circuit is developed, it's relatively easy to make duplicates. This makes PCBs a very logical choice for circuits that we need multiple copies of. All these advantages come at a cost, the main drawback being that it takes a fair amount of time to lay out the circuit and board. Also, the resulting circuit is much more difficult to modify, although some steps can be taken to build in a little bit of flexibility.

1 Simulating Circuits

When building circuits, one smart thing to do is to try and simulate the circuit beforehand to help find parameter values that give you the desired performance. The gain, bandwidth, etc. of a single/cascaded op-amp stages is relatively easy to evaluate analytically, especially when doing rough design, but it's nice to compare such analytical investigations with the output from a circuit simulation package. The package will often include a detailed model of the actual op-amp that you might plan to use. This should give you a more realistic projected circuit performance. Although i didn't pursue it here, you can also investigate things like see how the tolerances of your parts might affect performance or try and incorporate/simulate noise sources.

The program that we have access to is Multisim and it seems to work pretty well. It's default libraries include a number of common parts including op-amps that you would use in lab circuits. One nice feature is that it includes built-in virtual instruments. One that is particularly useful is the Bode-plotter. This lets you examine the gain and phase response of your circuit model before you actually build it. It's very useful when adjusting parameter values to get the desired circuit response. Figure 1 shows a screen-shot from multisim showing a small part of a feedback circuit as well as the Bode-plotter window. This is super handy for predicting the gain and phase response of a circuit as you're building and debugging it. There are also a number of other virtual instruments like oscilloscopes, voltage meters, etc. Multisim is basically a version of another simulation program or class of programs called spice. I think you can get a free version called pspice, if one should like to simulate circuits on their own or should Multisim not be available.

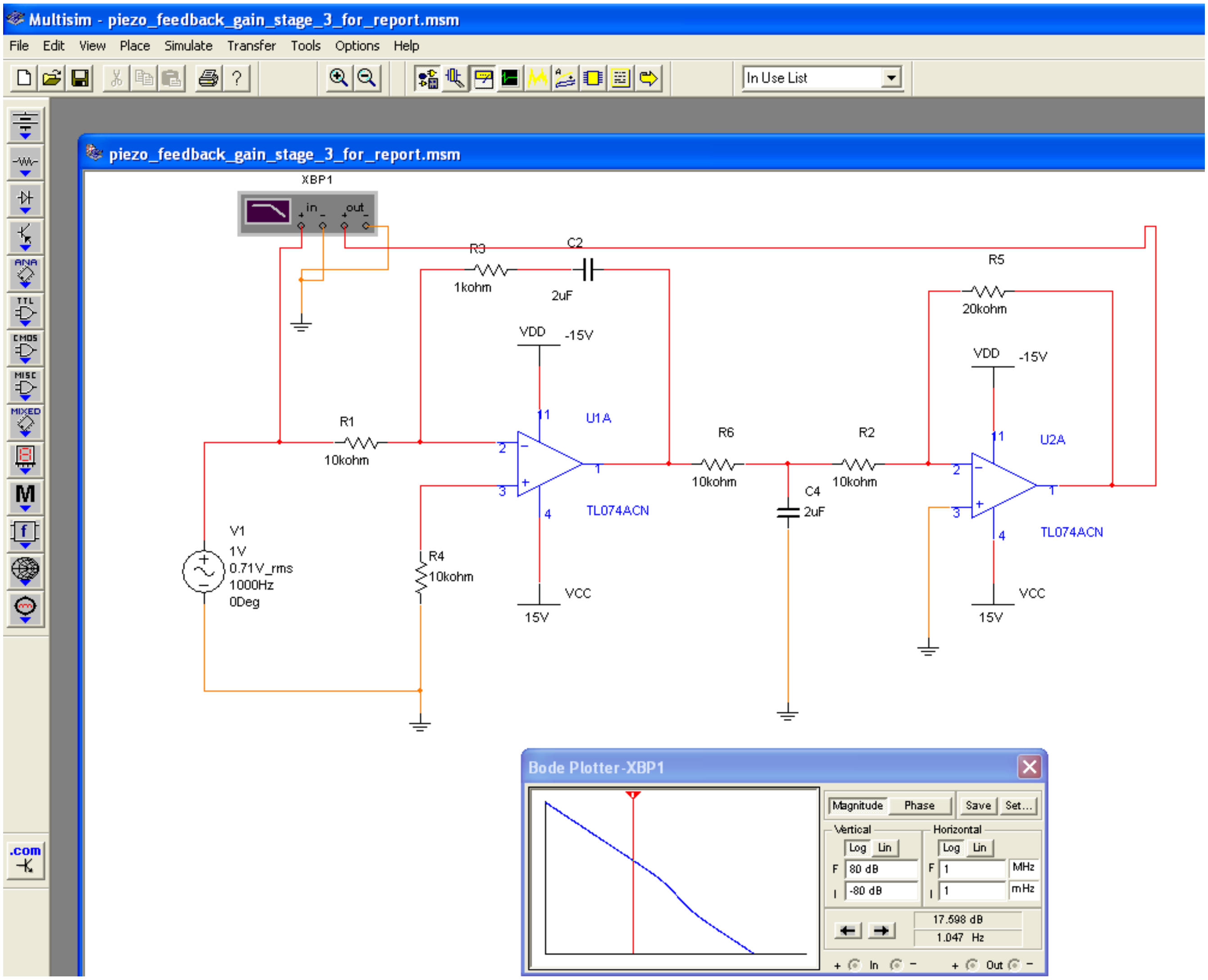

Figure 1: Multism Screen Shot showing Bode Plotter

2 Creating Circuit schematics in Eagle

In principle it”s possible to have Multism spit out files to create circuit boards but i chose to use a program called Eagle as per a recommendation from other students. Eagle has a free student version which has limited board size (~3”x4”) as well limiting the board to only 2 sides. However, this is enough capability to meet most any need we would have for electronics projects in the lab at this point. Eagle is a little bit finicky but a powerful program once you get the hang of it. There are basically three main functions that you utilize in Eagle: the schematic editor, the board editor and the library editor.The software can be found here:

http://www.cadsoftusa.com/download-eagle/freeware/

2.1 Schematic Editor

This is very similar to Multisim/Spice in appearance, although you can't simulate it. Also the layout process is slightly more awkward/a little less intuitive. The main thing to get used to is to select a button from the left-hand panel (see Fig. 2) for an operation you want to perform, say wiring two parts together for example, and as long as that button is selected, that's the operation you'll be doing. Another subtlety is that you can't just drag and select a group of parts, there's an explicit button for selecting a group of items, it looks like what you would think would be "cut". If you want to perform operations on a group of parts, say move or rotate them, that is done in the same manner. You click the group button and then right click and select the operation that you were doing, for example "move group". One detail issue: the supply pins for op-amps are not by default indicated. You can show them by right clicking on the Op-Amp and selecting "invoke", this will pull up a window that allows you to make those pins visible.

One operation that is pretty useful that you will encounter while designing a board is the Electrical Rules Check (ERC). This is located at the bottom of the columns of buttons. Just like how Word doesn't check the content of your writing but it does check your spelling, although Eagle doesn't simulate your circuit it does run some error checking to make sure that things are connected for example. Most likely there will be a large number of potential errors most of which you will probably ignore, for example components not having values specified and other non-essential issues. However, it is a useful tool for checking to see if you have missed a pin or may help you track down something that is misconnected.

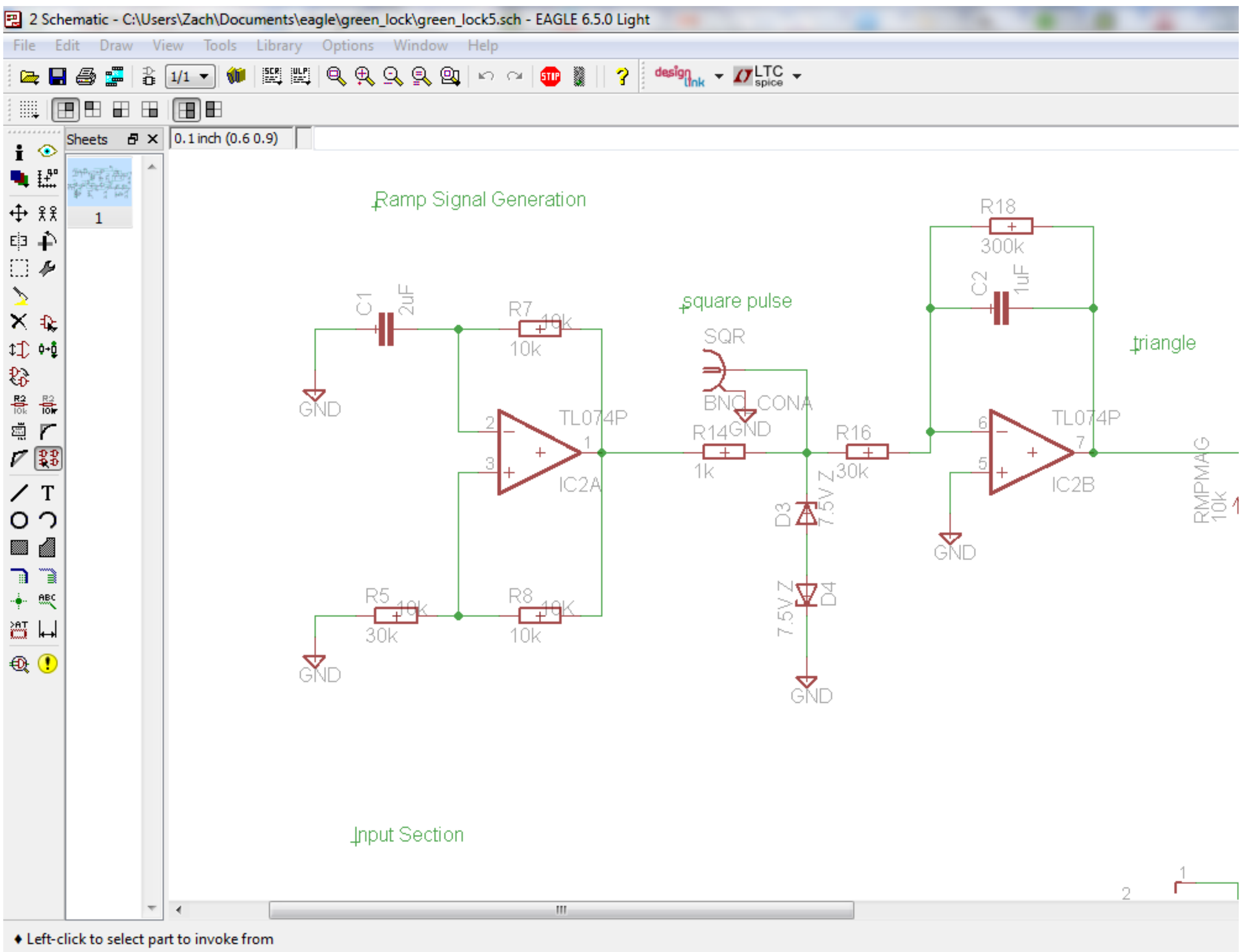

Figure 2: Eagle Schematic Editor Screen-shot

2.2 Board Editor

Once you have your circuit schematic created, the next step is to start designing the board. Go to file-switch to board. Note: one of the annoying things about the student version is that you are limited to a board about 3x4 inches. The problem is not so much that this won't be enough space for your project, but that when the board is first created from your schematic, eagle places all the parts on the page and you have to put them into the allotted space, represented by a rectangle. Unfortunately, you can't move them out of that space to have more room to see what you're doing. As a result if you try and place everything at once, it'll most likely be very busy and hard to organize. I found that it was useful to build up the schematic in stages and then work on the board, this made it easier to keep track of the different sections of the circuit and made it easier to logically group the circuit components on the board into functional blocks. See Figure for a screen-shot of the board editor. This is perhaps not the most instructive shot as the board is already complete, but we can point out some of the important functions.

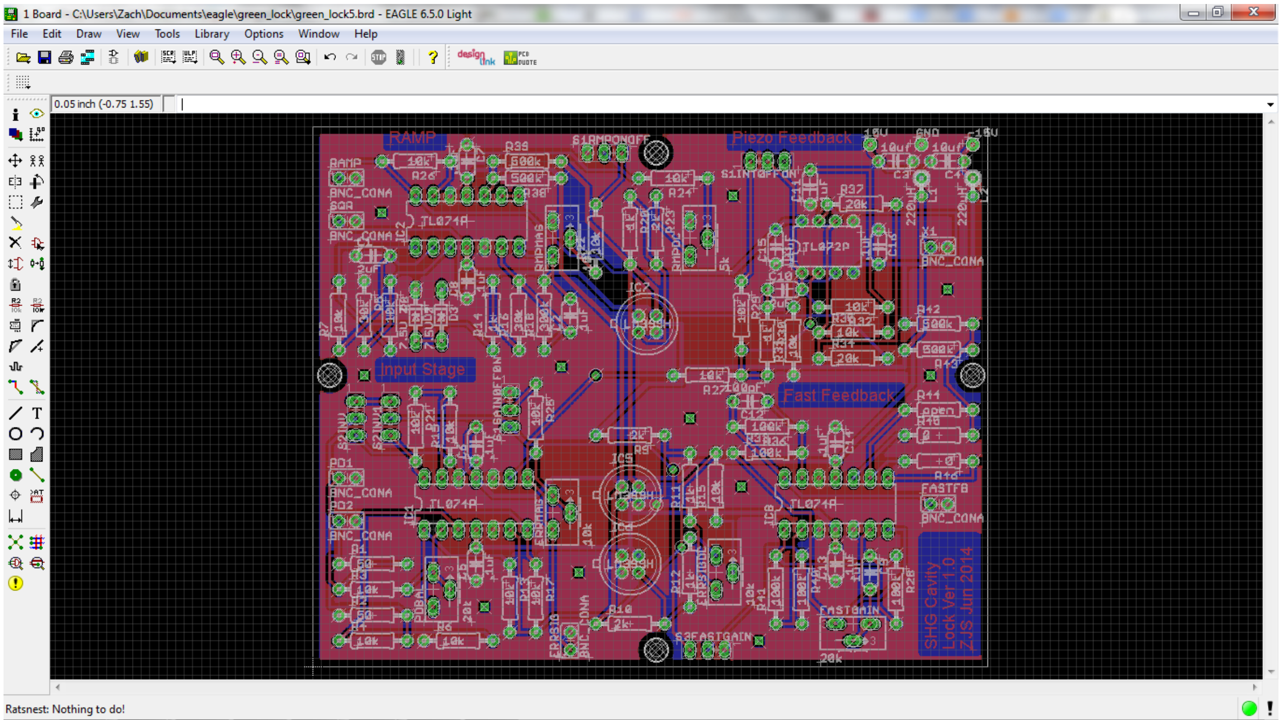

Figure 3: Board Layout

There are a number of functions that you'll want to become familiar with in the circuit board editor. Looking at the two columns on the left hand side there are a number of buttons that are the same as in the schematic editor. However there are also some new functions that you'll want to familiarize yourself with.

- Autorouter: second from the bottom in the right column. Once you get your parts in place, you can click on this and it will route all your wires, you are free to route them on your own as well, but i found this function really useful with a circuit this size. Smaller circuits you could probably do all the routing by hand. Also if you were doing RF fro example, it's very possible you'd want to route more carefully.

- Ripup: second column above the T, this let's you rip up traces once they're down. This is useful to modify your design, or if you want to rip up everything, click on Ripup and then click on the green go light at the top of the window. This will rip up all your traces.

- Holes, Text and other functions are also available that you'll need to create your board.

- Design Rule Check (DRC): just like the electrical rule check, the design rule check lets you check and see if you have parts on top of each other or things too close together for example. You want to look through all the warnings and errors it flags before your circuit is ready for prime time. Some of them may not matter or only be sort of technicalities, but others may cause you problems.

Here are a few tips on some procedures you might want to do in the board editor:

- Change trace sizes: For power rail nets vs signal nets you might want to change the size of the trace to make them larger, it's nice to be able to distinguish power and signal traces on the board. This can best be accomplished through net classes, you define two different classes of nets, one a default, another a class for higher power signals. This can be done under Edit/Net Classes. Once you have define your classes, you need to assign them to your different nets. This is most easily done on your schematic. Just right-click on your high power nets and change their class to the high power class you created. Then when you go to route the wires on your board, those nets will have different width traces.

- Create ground planes: Ground planes are one of the significant advantage of PCB over a home-built circuit board; you can fill in all the empty board area with ground plane. I would think this would be especially powerful with high freq circuits. To create ground planes:
  - First you use the polygon tool to draw a polygon around the circumference of your board.
  - Next you use the name tool to change the name of your polygon to 'GND'.
  - Next use the info tool to check the layer and clearance. You want to make sure that your polygon is the right layer, i.e. the bottom layer, layer 16. Note, here you can also modify the clearance between ground plane and your traces by changing "Isolate" under the polygon section. This is nice to be able to adjust the space between your traces and the ground plane.
  - Next you can actually create the plane, by clicking the 'Ratsnest' command. This is the third from the bottom in the 1st column. This will fill in all the areas with the ground plane and you can see what the board will look like.
  - After that, you can create an identical layer except for the top layer, layer 1. In Figure 3 , the red is the top ground plane, the blue the bottom one.
  - The last step is that it's good practice to connect your ground planes together at multiple points. This can be done by sprinkling some vias around on the board. However, once you've done that be sure to name the vias 'GND', otherwise they will be tunnels isolated from the ground planes.

2.3 Library Editor

But what if you want to change the footprint of a part on the board, you want to use a more compact switch footprint for example? This leads to one more challenging/confusing issue you may want to deal with, which is editing parts in the library. I just dabbled with this functionality to create a couple things: switch and potentiometer connections that i was unhappy with in the default library or couldn't find them. To do this, first you want to create a library; the easiest way to do this is through the main eagle menu: new->library. Next copy the part that you want to modify into that library as well as the part with the footprint you want to swap in. I think it's always a good idea to work in a new library you populate with copied parts as that way you can't mess up a part in the library and you also have a lot fewer parts to swim through in your little personal mini-library.

This starts getting into the nuts and bolts but basically what you want to do is open the part in your library and create a variant where you change the package to the package of the part with the footprint that you want. The other thing that you have to do is make sure you have the connects assigned correctly. This is done by clicking on connect and linking the pins of the schematic with the pads of the package. This wasn't too difficult for a part with just a couple, three connections, but i see how it could get complicated in a hurry. I also did not investigate how to make custom footprints, but i could see how that would be a useful thing to know as well. If one was dealing with a unique part that did not exist in the library and wanted to use it in your circuit, this would be where you could create a package as well as schematic symbol for it. See Figure for a screen-shot of the library editor. You can see the schematic in the upper left, the package in the upper right, as well as a pop-up that shows the pin assignments when you click 'connect.' This was a part i modified so that a switch would just have 3 solder pads i could wire to instead of the normal switch footprint, since our switch would be located on the front of our circuit box and connected to the board by wires.

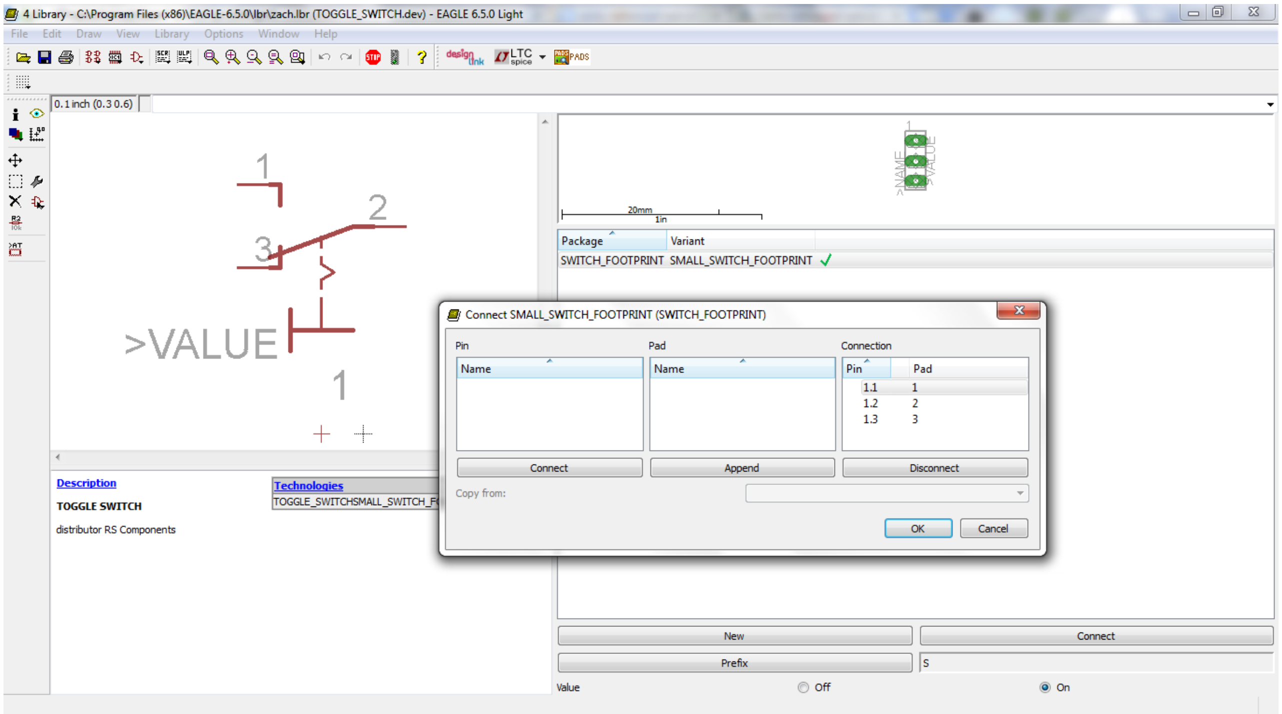

Figure 4: Library Editor

3 Having the board manufactured: Advanced Circuits

After you have your schematic made and the placement of components on your board fine-tuned, as well as checked your error rule check (ERC) and your design rule check (DRC), next it's time to get your board made. I chose to have it made by Advanced Circuits. They have a pretty sweet deal for students where you can get a 2-sided board manufactured, no minimum quantity,for \$33. With standard shipping i think it took a little over a week to arrive. The page is here: http://www.4pcb.com/pcb-student-discount.html

In order to manufacture the board, Advanced Circuits needs a group of files from you that define the different layers of the board, the copper top and bottom layers as well as silk screens, etc. In general these files are known as Gerber files. You need to have Eagle spit out the appropriate files needed by the manufacturer. In order to do this you use something called the CAM processor, under File->Cam Processor. This link gives a pretty good description: http://www.dfrobot.com/community/export-gerber-files-from-eagle.html

Here are the steps, in the Cam Processor, go to open->Job and select Gerber274x.cam. Then you hit 'Process Job' and the Cam Processor will spit out the files that you need into the directory you have your other Eagle files. Note: you need to do this procedure again, except selecting excellon.cam in order to get the files associated with drilling.

Next you zip all those files up and upload them to Advanced Circuits. Before you do so for the actual board, an extra step that you can do through Advanced Circuits is their free DFM (design for manufacturability) check. You upload your files and they will send you a report that lists any potential issues or if there are problems with the design, very similar to the integrated rule checks in Eagle. For example doing this let me know my text was too fine and wouldn't print very well so i made the lettering thicker so that it would be easier to read.

After that you upload for board manufacture. The web process will recognize some of the files, actually probably all the ones that really matter, but you may need to assign some of them as well. There was some ambiguity in the file names, but it seems that not all the files Eagle generated are essential. Basically i assigned the ones i could figure out, which seemed to cover all the most important ones and set the others as Drawings/Other. This seemed to work, the only issue that i encountered was that i did not properly specify the bottom silkscreen layer. In retrospect i think the reason i didn't have a bottom silkscreen layer was probably because i didn't have one in the circuit board design. In any case it ended up not being a problem as they went ahead and made the board anyway. In the future we could just put in the options that we don't want a bottom silkscreen layer and i think that would resolve the issue.

file extension from Eagle	Advanced Circuits Assignment
.cmp	top copper
.drd	job drill data
.dri	drill tool description, not necessary for PCB -> Drawing/Other
.gpi	Gerber Plot Information File, not necessary-> Drawing/Other
.plc	top silkscreen
.sol	bottom copper
.stc	top solder mask
.sts	bottom solder mask

Table 1: Gerber Files from Eagle and for Advanced Circuits

Wiring up the circuit board is pretty straight-forward. If you took the time to include the part values in your design, you can have Eagle export a list that has all the part numbers and their values. You can follow this and build up the circuit, basically solder by number. A few considerations: it's nice to build up the circuit in stages, this allows you to test as you go. It's also nice to include convenient signal pick-offs so that you can easily examine signals, rather than trying to wiggle a probe into a tight spot. I included sockets for ICs should one fail because if they're hard-wired in, it's very difficult to remove a chip with many pins. One thing i would consider changing is i have a number of potentiometer and switches that are hard-wired to the board. This is a bit of pain to do and makes the whole board when assembled a spaghetti of wires and connections. Headers could be employed to make the circuit more modular although I would think the connections would perhaps not be as good.

4 Additional Notes

Design with op-amps: One design issue that i ran into was how to deal with unused op-amps. It turns out that you don't want to just leave them float as it would be possible for the input pins to charge up and for the op-amp to rail. This could cause noise or other unpredictable behavior. In general the way to deal with this is to feedback and tie the input near ground with large resistors. Additional resistors or really slots for resistors can be incorporated to allow the op-amp stage to be utilized at a later time. See Figure 5 for an image describing this technique. This image is from Ch 17: "Circuit Board Layout Techniques" from Op Amps for Everyone. This is available from Texas Instruments here: http://www.ti.com/lit/ml/sloa089/sloa089.pdf. This document contains a ton of circuit-board layout information. If we needed to design a high-frequency circuit application and really wanted to carefully suppress noise pick-up loops, etc, this document would be a good place to start as far as investigating op-amp circuit board design practices.

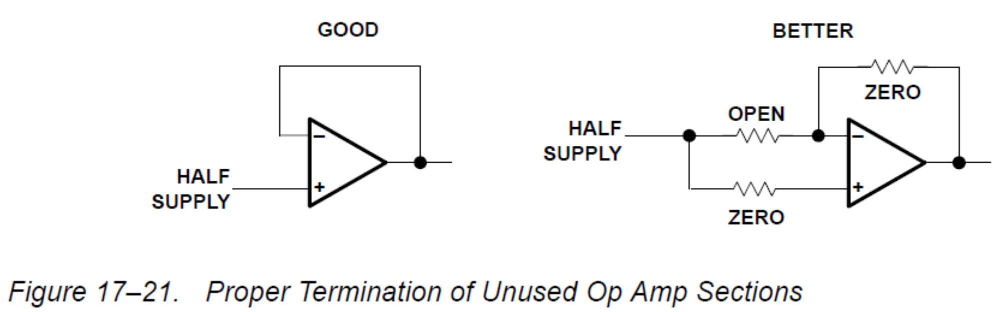

Figure 5: Unused op-amps