
EAGLECAD TUTORIAL

ECE 445

*University of Illinois at Urbana Champaign
Department of Electrical and Computer Engineering*

Instructions

This documentation includes explanations of EagleCAD functions/features and ECE 445-specific pointers for both the Eagle assignment and the PCB required for each Senior Design project. This is meant to be supplementary material to a step-by-step guide, like the one linked in the assignment.

Appendix A has Eagle's layer definitions and Appendix B includes PCB layout guidelines.

Overview

Schematics are the core of all electrical engineering. They represent a universal list of symbols and notations that can be followed by anyone in order to present some sort of circuitry. In fact, one can even think of some as a sort of language that is used to communicate our electrical ideas. Just like a language, we can present an idea in a clear and specific manner, or we can obfuscate it through a dense and complex layout of thoughts. The goal of this tutorial is to help you learn tips and methods to be in the former camp.

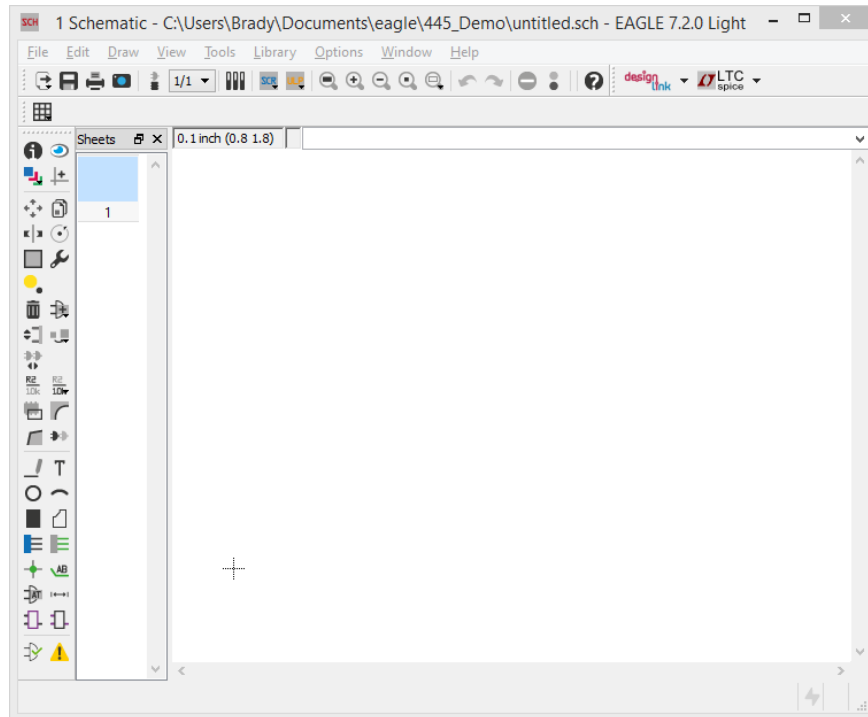
While schematics are nice from an analytical standpoint, real circuits don't often follow the same shapes or layout. As you have most likely seen in previous classes, even trying to prototype on a breadboard often has one moving parts around in order to get the proper connections. In this class, you are going to learn how to translate designs from schematics to a realizable **printed circuit board**, or PCB.

EagleCAD is one of many available Electronic Design Automation (EDA) software packages. It allows us to easily design and transition between schematics and circuit layout. EagleCAD has a large amount of popularity with hobbyists, and due to the nature of the projects in this course, you will most likely find that to be of use. You can download the freeware version of EagleCAD here: <http://www.cadsoftusa.com/download-eagle/>. The lab computers have the full professional versions installed. **WARNING: DIFFERENT EAGLE VERSIONS MAY NOT BE COMPATIBLE.** This applies both forward and backward. We advise sticking to one version for the entire semester.

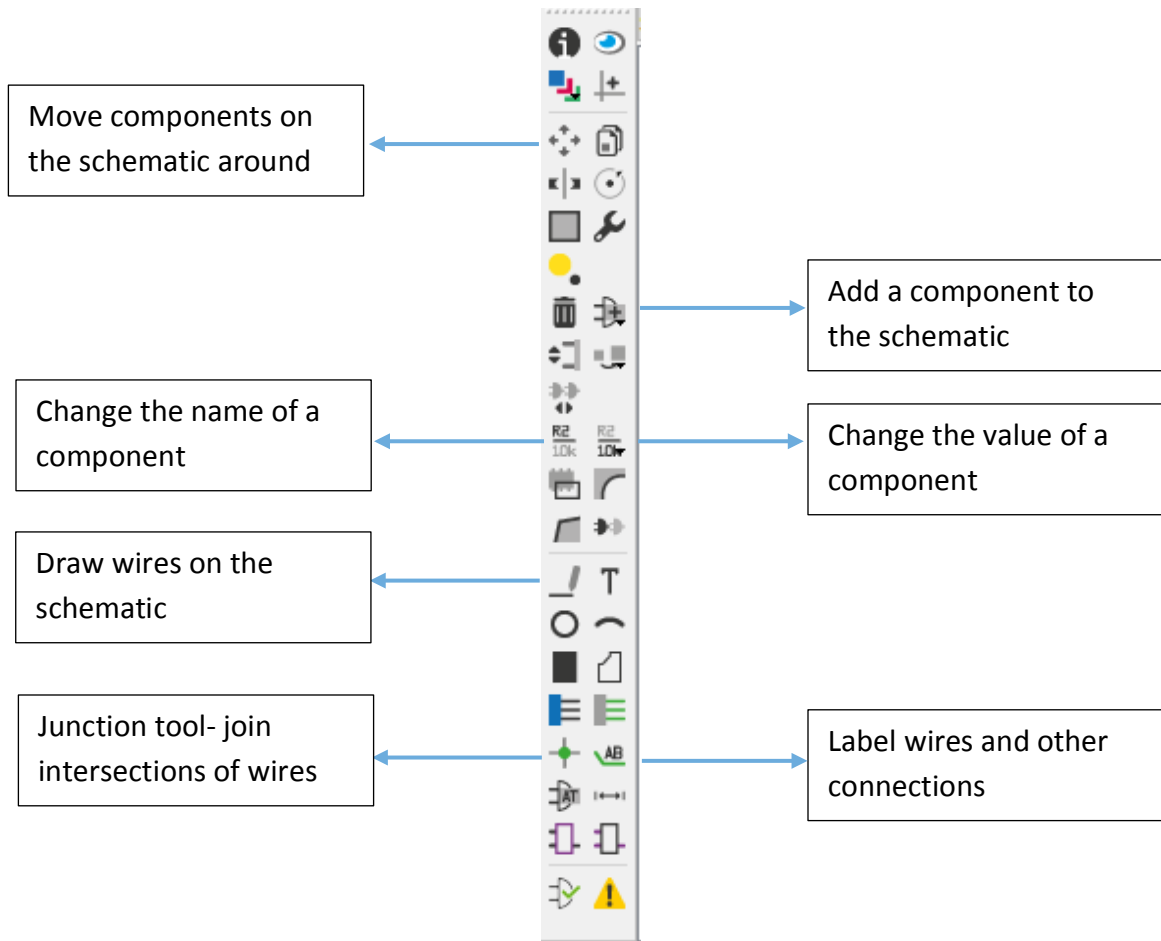
Schematic Capture

Let's start with some of the features of Eagle's schematic capture tool. When you open EagleCAD, the Project Viewer screen pops up, and you can create a new project by going to **File -> New -> Project**. Name it whatever you like, but make sure the green dot is visible next to the project – this indicates it is the active project.

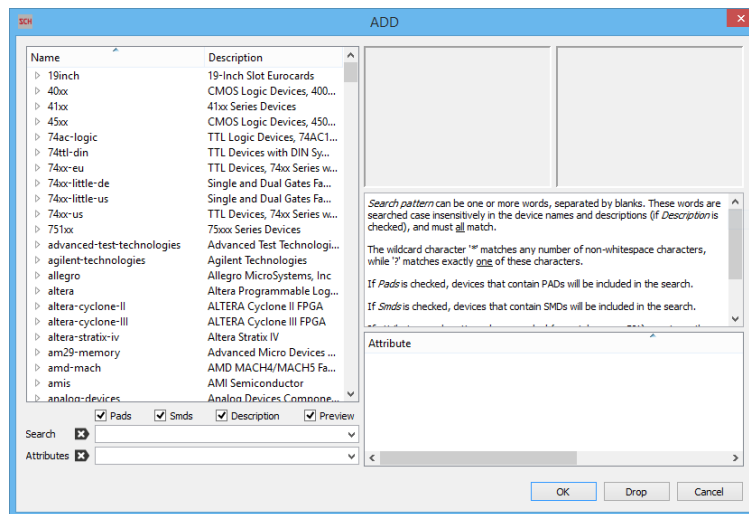
Right click on the project, and select **New -> Schematic**. This should pop up:



While all the buttons available on the left have some use case, we'll really only be using a few of them for 90% of the work. The next picture shows what functions these buttons command.



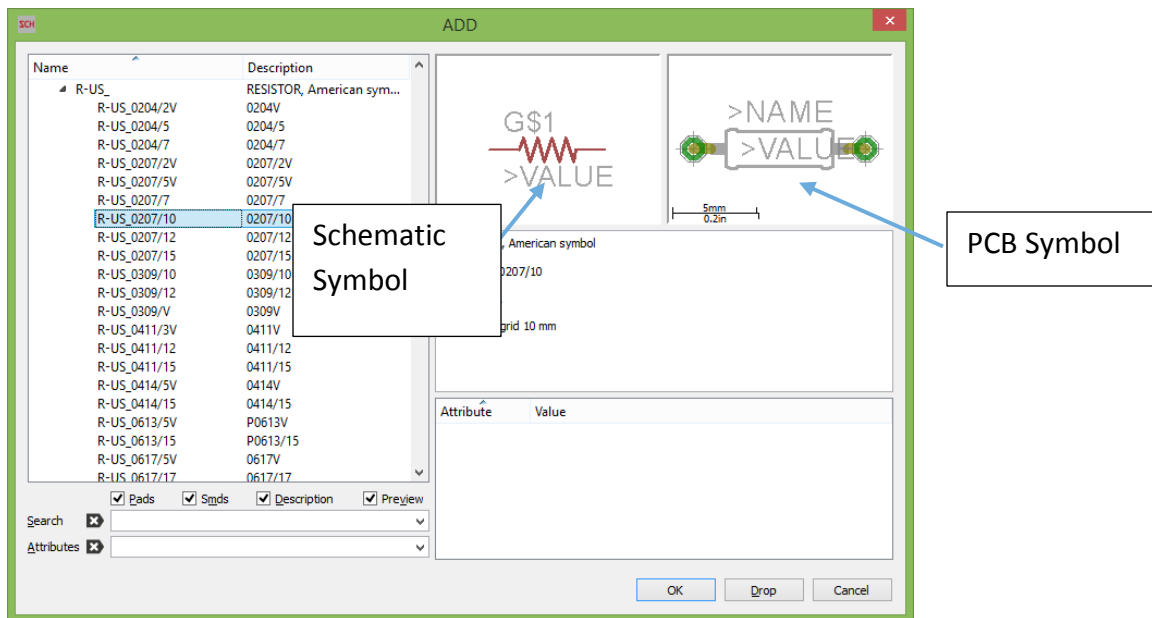
Adding Components: Press the add part window, and you'll see a list of libraries appear.



These libraries are supplied from a variety of sources: CadSoft (the creators of EagleCAD), numerous semiconductor companies, and even just regular hobbyists. Since the files

are very simple to generate for new components (just simple SVG drawings), you can find a library for nearly any part. The downside is that the organization leaves a bit to be desired, but luckily the search feature is simple enough to use. For now, we'll start off using standard libraries.

Scroll down to the library called **rc1** – this is your standard passives library. It contains symbols/footprints for the basic resistor, capacitor, and inductor parts. Opening it up, it'll have several sub folders. Since this is America, open up **R-US**. Everything contained in that sub list is a resistor, but with a different package.

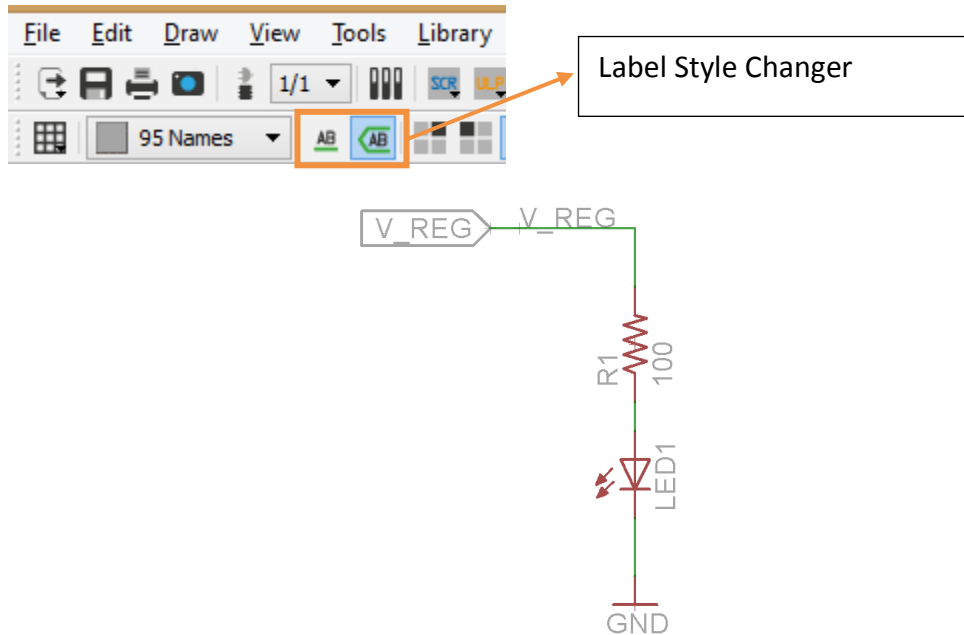


You can tell a part is “through-hole” by looking at the PCB symbol. If it has green circles, it's a through-hole part as those indicate vias in EagleCAD. If it has red squares instead (bottom half of list), those represent pads, so it is a surface mount part.

To add a part, double-click and drop it anywhere on the schematic. To keep adding copies of it, continue clicking on the schematic. To “let go,” just click the Move tool. Using Move, left-click to select the part to move and right-click to rotate parts.

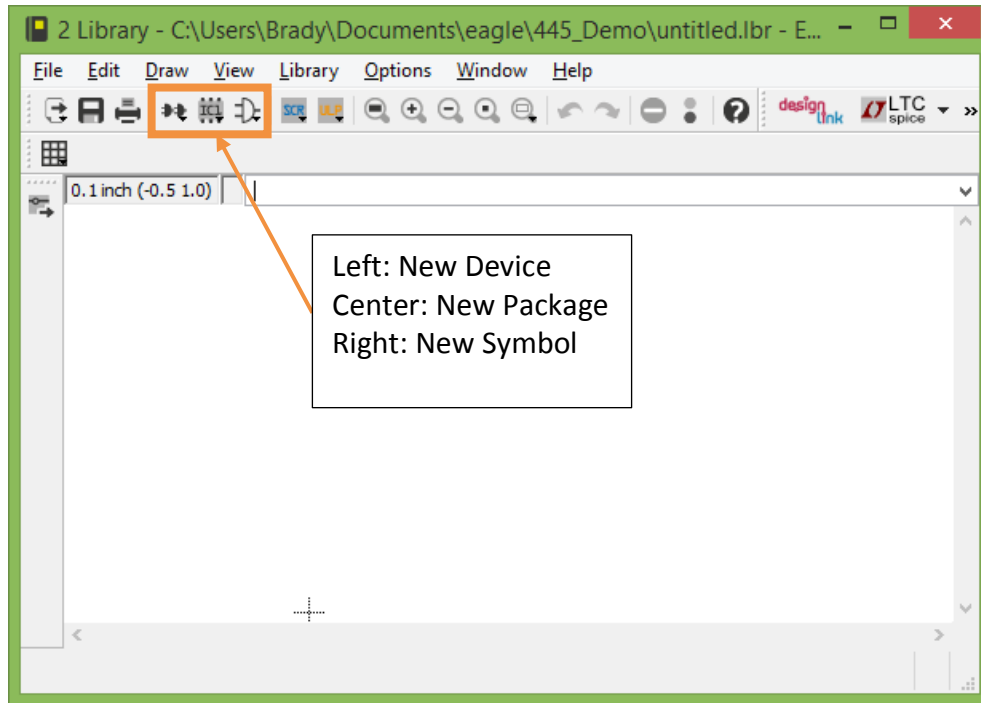
Connecting Components: Use the Line Tool to draw a wire (or “net”) between the components. Don't forget to add a ground, or reference node, to every schematic. You can find the ground symbol under the **supply1** or **supply2** libraries. If you look, you'll see that ground doesn't have a board view – which makes sense, ground is an electrical reference voltage, not a tangible, physical *thing*.

Use the Name Tool to give your nets an appropriate name: V_REG, V_OUT, VDD_5V, etc. You'll notice that doesn't really do anything by default, we need to also show the label as well. Use the Label Tool and click on the connection, then place the label somewhere nearby. You can change the style of your label via the buttons on the top bar, shown below where I've chosen both labeling styles.



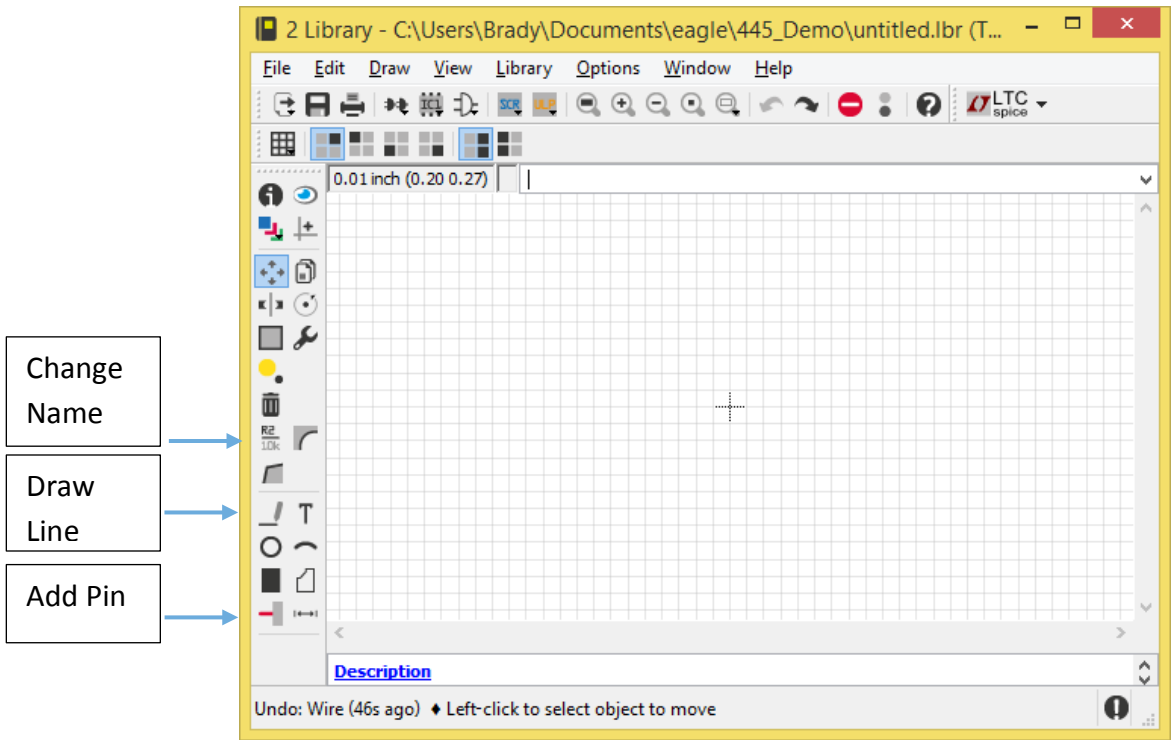
Creating a New Part

Since the class assignment uses the TPS799 voltage regulator, this is the example we will cover in detail. You can search in the libraries, but it's not there (part deliberately chosen this way). You can find the datasheet here: <http://www.ti.com/lit/ds/symlink/tps799.pdf>. For the simplicity of the assignment, we're going to use the "Fixed Voltage Version" on the bottom of the first page. Now the first thing to do is to create a new library in Eagle for this part. Go to the project manager window, select your project, and right click **New -> Library**.

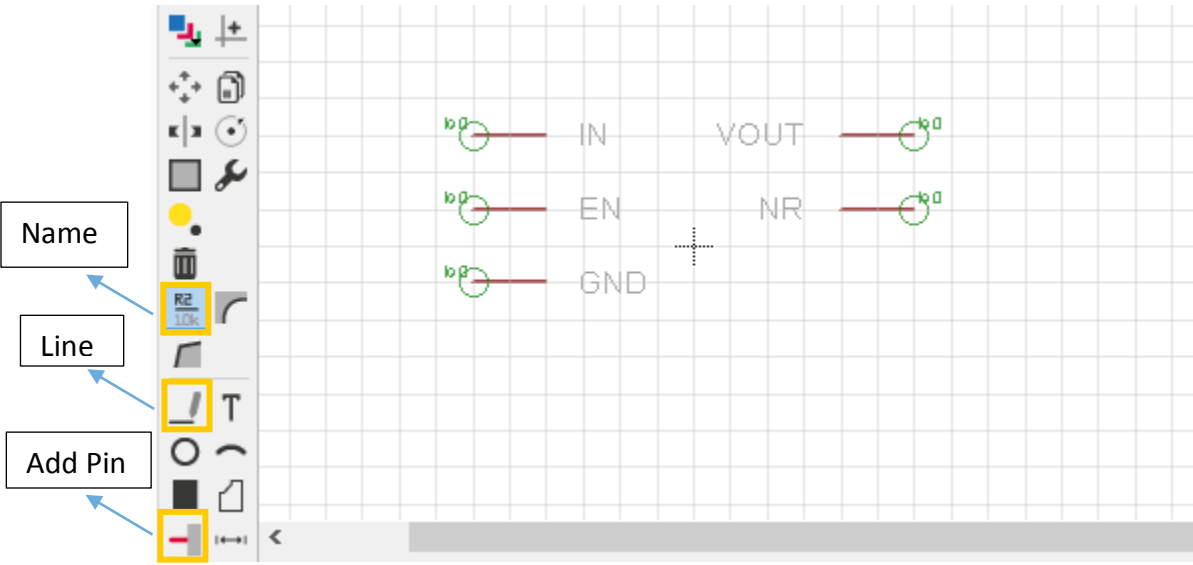


Now is a good time to talk about how EagleCAD stores parts. Like you saw when adding parts, components generally have both a schematic and board view. The schematic, or **symbol** view, is the agreed upon electrical symbol, and the board, or **package** view, is the physical layout. Some items – such as supply lines (e.g. +5V, GND, VSS) – only have a symbol view, where items like vias, traces, and wells only have a board view. When creating a new component, we generate the symbol and package views asynchronously, then combine them together with a **device** view. **Note that EagleCAD is not a simulation software of any kind**, we can call a part a resistor and give it three inputs and one output, or a capacitor with only one pin. EagleCAD doesn't care about what the part is electrically doing, only physically.

Let's start with making a new symbol. In the pop-up, give your part a name, presumably something along the lines of "TPS799". Press **Sym** to verify we're creating a symbol, then create it.



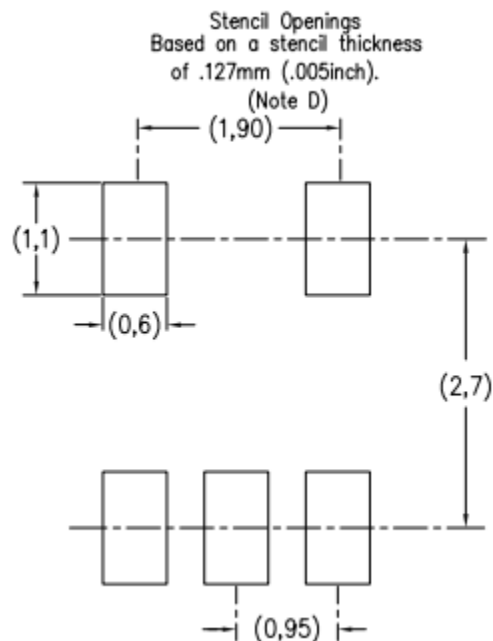
We now have the symbol editor view open. Begin by adding in some pins. For most schematics, people find left to right the easiest to read (like a book), so we'll put the ground pin on the side rather than the bottom. Add 5 pins using the Add Pin Tool, then rename them using the Name Tool. The green circle on the pin is the I/O point, so we'll want those facing outwards. Again, we tend to put inputs on the left and outputs on the right. This makes for a logical reading as we decipher a schematic.



Finish up with the Line Tool to box it off. Try to center the part as much as possible to make your life easier when adding it and moving it around. Save the symbol (as well as the library if you haven't already).

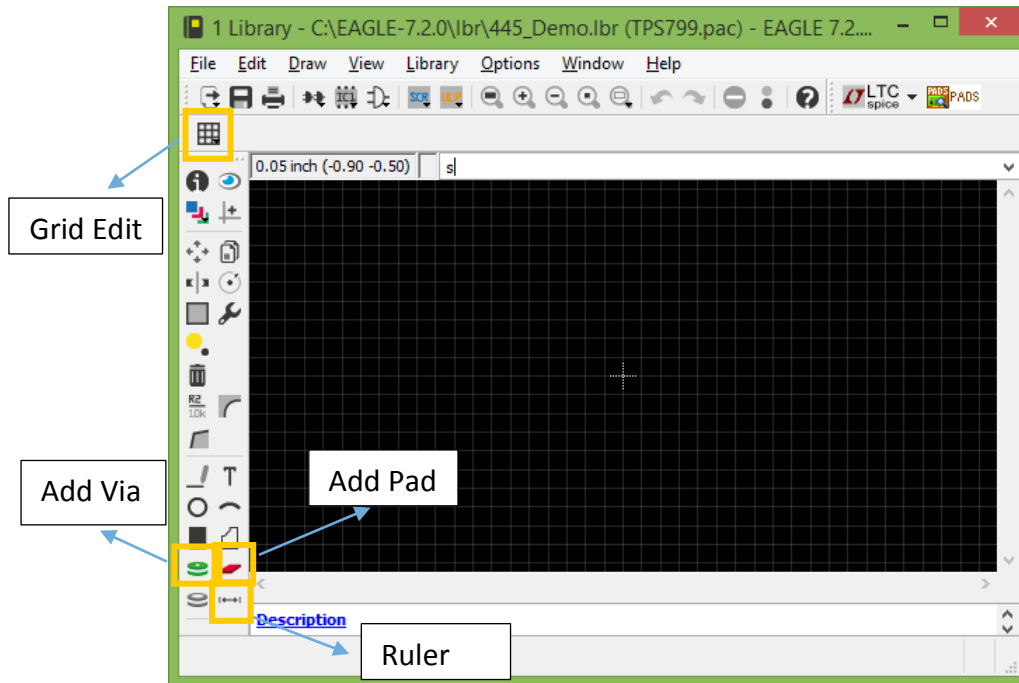


Now we'll handle the package. There are several ways to do this, but we'll do it the most manual way for now, just in case we ever get a new part that's not in Eagle at all. The part has several different packages, but we'll use the biggest one in this case, since if we were to actually create the board, it would be the easiest to use. Packages come in many different types and names, often redundantly so as you'll find out. We'll be using the *DDC* package, also known as *Plastic Small Outline*. While they give many views, we're most interested in the footprint – this is how much space we should leave on the board for the part to sit in. Dimensions shown below in millimeters (from the datasheet).

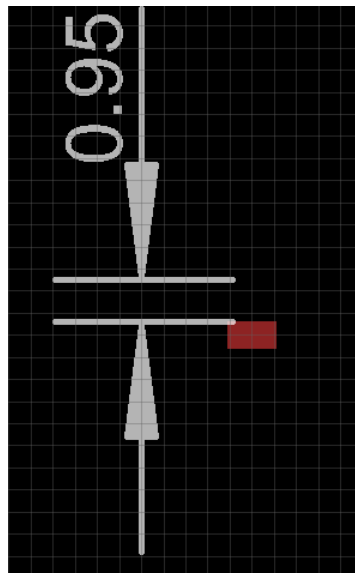


Let's Hit **New Package, PAC**, and name it "TPS799" or something similar. The following window should open. Once again, very similar to what we have seen before, but with two critical new options. The Add Via Tool allows us to add a hole to the footprint that is used for leads or pins. The Add Pad Tool allows us to add a pad for any surface mount part. Lastly, we'll

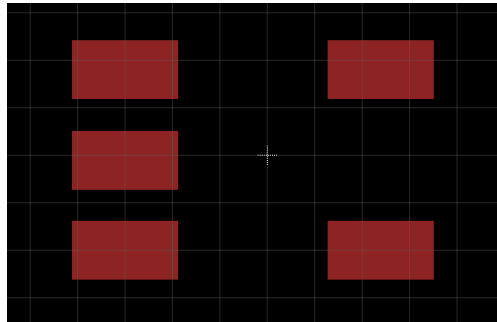
need to use the Grid Tool to make sure that the minimum grid spacing lines up with the smallest significant figure we have – here 0.05mm.



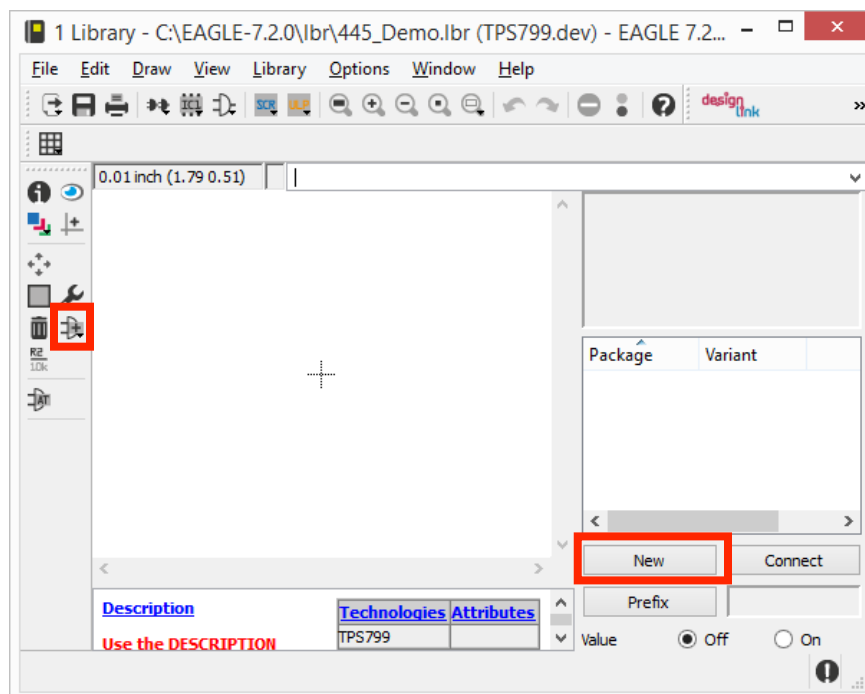
I suggest using a 0.5mm grid, with a 0.05mm alt setting. Add a pad, and place it relatively near the center. Right click on it, **Properties**, and change **Smd Size** to “1.1 x 0.6”. This automatically sizes the pad to the correct dimensions – if it’s off by 90 degrees, rotate it. Now use the Ruler to go and measure out the appropriate distance for the next pad. Remember to hold ALT as you do so – otherwise we’ll be on the main grid dimension, not the finer one. I’ve attached a picture of what the output should like below. Once that is done, copy and paste the previous pad and place it along the same line.



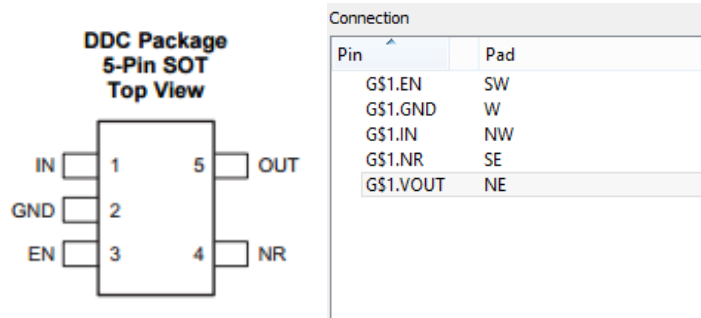
Now this is clearly going to get pretty painful if we want to do this for every pad. Fortunately for us, we can get around this with a little math. Delete the ruler drawing for now, we won't need it. Add another new pad via copy paste, just putting it anywhere. Right click, select **Properties** on the first pad, and note its **Position**. Now do the same to the newly added pad, but just move it to the appropriate point – either 0.95mm up, or 2.7mm out. Absolute position doesn't mean anything here – only relative. Once that's done, go back and name the pads! Any pattern works: [1 2 3 4 5], [TopLeft, MidLeft, BotLeft, TopRight, BotRight], [NW W SW SE NE], as long as you can tell which is which. Final image should look like something below. Don't forget to verify your layout with a ruler – never hurts to double-check.



Save it, then go back to the top and open a **New Device**. Once again, let's call it TPS799. The following window will show up. This should be the easiest of the previous steps – all we do is connect the dots! The button highlighted on the left shows the Add Symbol Tool, and the highlighted new button is the Add Package Tool. There should only be one of each in this library, go ahead and add the schematic and package we created.

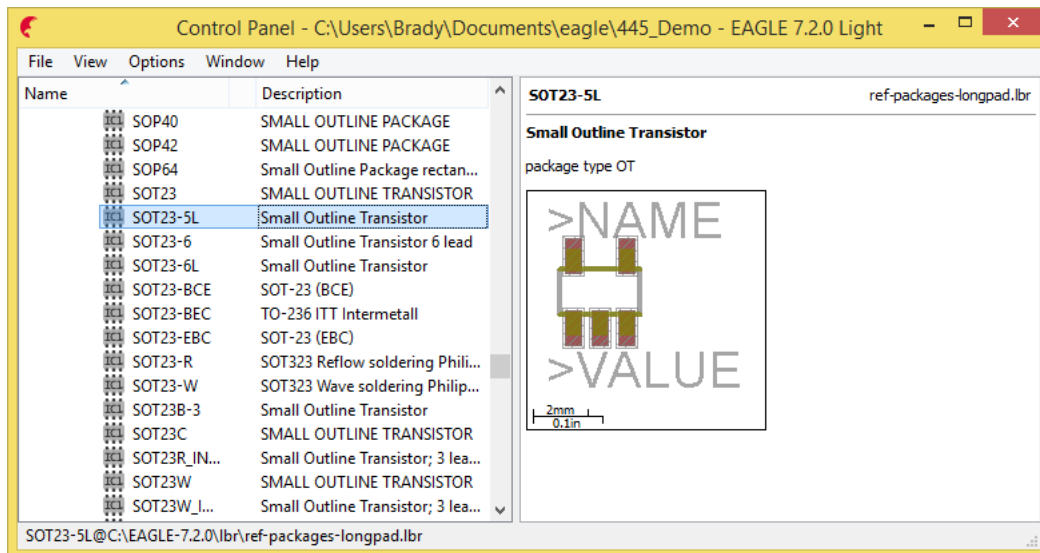


Now hit **Connect**, and let's *connect* the dots (pardon the pun). You can find the order in the datasheet, generally under a section called "Pin Mapping" or "Pin Configuration". Go through, select the appropriate Pin Name and Pad Name, and then hit Connect. Proceed for all five options, and you should end up with a configuration shown below.



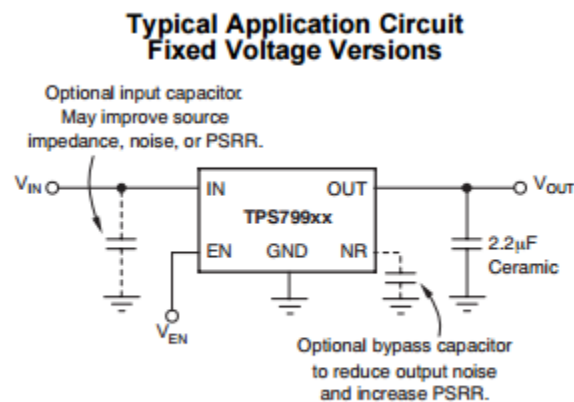
Finish it off and we're (almost) good to go! Before we finish this, let's go ahead and learn one more way of doing this. The large majority of parts you'll be making will probably be IC's. Nearly all IC's come in common packages, so there's no sense in re-inventing the wheel when it comes to drawing packages. EagleCAD stores all its reference IC footprints in **ref-packages.lbr**, or if you aren't comfortable with your soldering skills, **ref-packages-longpad.lbr**. The only difference is the longpad variant has, no surprise, larger pads in order to give more soldering area.

The tricky part about this is that you have to know that many IC footprints are the exact same physically, but they have plenty of name variants. For example, the part we have here is given to us as DDC or R-PDSO-G5, but is found in ref-packages as SOT23-5L. While you have your old library open, go back to the project viewer. Go to **Libraries** -> **ref-packages-longpad.lbr** -> **SOT23-5L**. Right click, **Copy to Library**. This will copy a version of the footprint to your library, so when you go to make a new symbol, you can see you now have two footprints to choose from – the one we designed, and the one given to us.



The choice is yours on which method to use. Both have their pros and cons, and unfortunately unless you get lucky with the footprint, take a little time. Regardless, now that we have our part created, it's time to add it to the schematic and move on to the PCB layout.

To add the newly made part to your schematic, go to the top toolbar and hit **Library** -> **Use**. The library you just created should be in the default folder, open it and the parts in it will be available to us. Add it in and wire it up. **Don't forget to finish the full voltage regulator schematic for the assignment, including the passive components required for the TPS799 to work.** See the screenshot from the datasheet below:

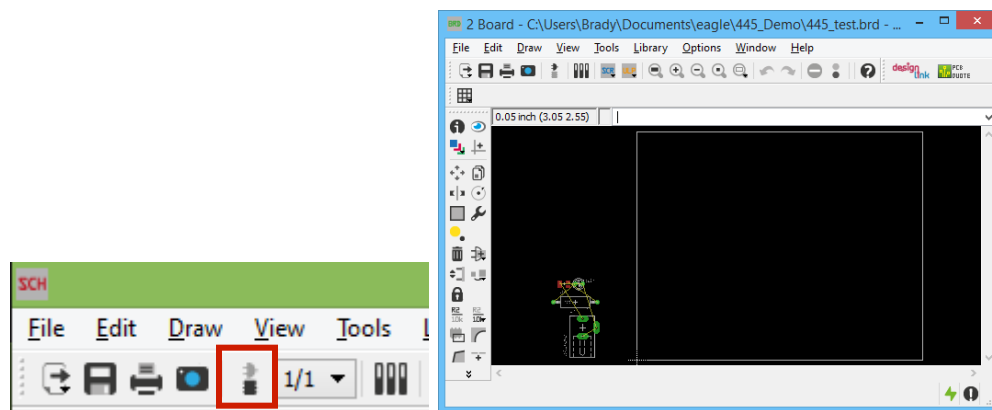



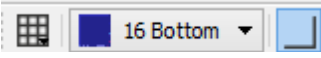
One last important note is the green “junction” dots when wires cross. These are done with the Junction Tool, labeled in our buttons figure at the beginning. Whenever we cross wires in EagleCAD, by default these are NOT connected. We need to add the green dot to ensure they are indeed touching. It's a nice visual aid, as well as a way to confirm that we do intend to connect wires when we cross them. We can check for this and other small problems with the **ERC Tool**, Electrical Rule Check. It's the bottom left most button, press it and see what errors you get. Feel free to ignore any irrelevant ones and deal with the important ones.

Making the PCB


PCB design is just a big puzzle. All we try to do is match our schematic by connecting the pads and vias, with a general goal of *minimizing* area. Each board consists of several layers to “route” (or connect) on, see Appendix A. The free version of Eagle allows you to use two layers, the paid version on the EWS machines allows four.

The first step is to switch from schematic view to PCB view. Press the highlighted button in the toolbar to create the board view. You’ll be presented with the screen below. What EagleCAD has done is save the netlist (or nodelist) from the schematic, and connected all the parts that need to be connected with yellow wires. The white square is the work area, so it can be an arbitrary size. To define the actual size of the PCB, draw in the border with traces width=0 on the **Dimension** layer.

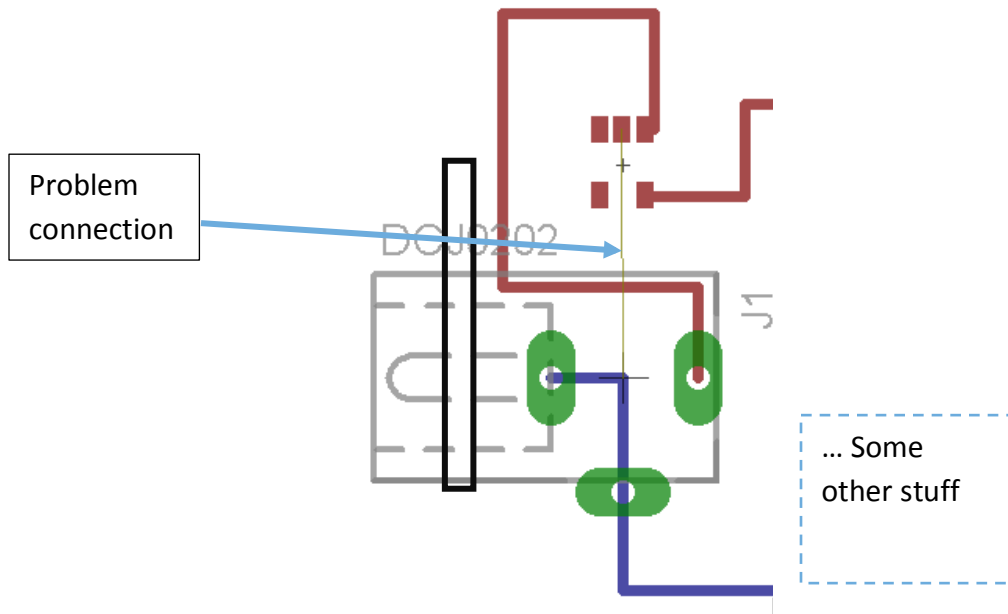


Begin by moving some of the parts into the work area. The Route Tool  is what we use to make these connections. When you select it, you’ll see one of the toolbars show a menu like so . The dropdown bar chooses what layer we route on, and additionally, each layer is color coded so you can always visually confirm as well.

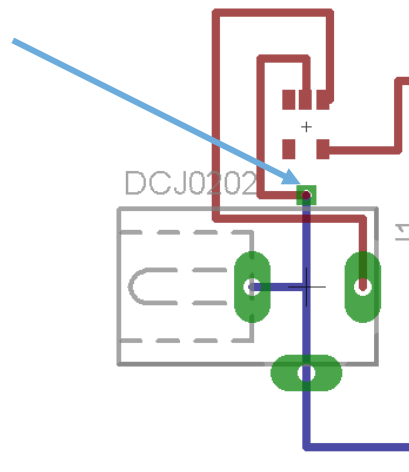
You can only route between nodes, so pick any part with a yellow line and begin. When you click one pad/via, the nodes it needs to connect to light up. A green via represents a hole that goes all the way through the board, so you can connect on any layer. A red or blue pad is created only on its corresponding side, so you will need to draw on the appropriate side. These lines we draw represent copper traces, which is a conductive material. Therefore if you overlap these lines, or accidentally overlap a line with a pad/via that you’re not supposed to, you’ll be creating an electrical short. Watch out for that, and only connect what’s supposed to!

Continue adding parts on and connecting them. Every once in a while, hit the Ratsnet Tool . This tool redraws the yellow electrical lines to the shortest configuration possible.

Below you'll see a partial layout where a via is required. The problem here is my closest connection is on the bottom, but my pad is on the top.




For this case, I would start off Route Tool on the top side, selecting the pad. When I get to a not crowded spot, *with the Route Tool still selected*, change the dropdown menu from "Top" to "Bottom" and finish the trace. See that small green square in the middle? EagleCAD automatically placed a via for me to short + connect my two traces.




How wide should one make these traces? Where should they wind up going? The first rule of thumb is draw what you can manufacture. If using the ECE shop, you should import their DRC rules and use their min-spacing/width/etc guidelines, both found here:

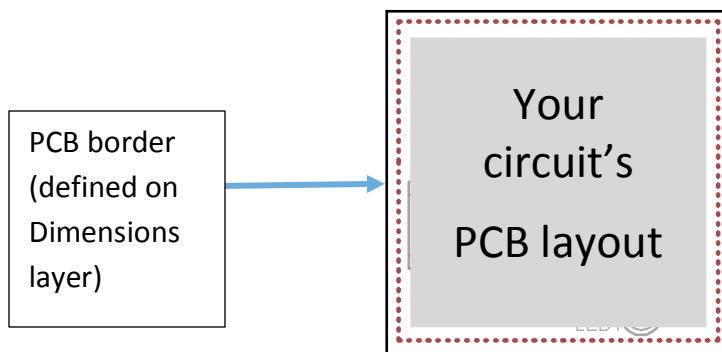
<http://eshop.ece.illinois.edu/pcbdesign/designreq/designreq.php>

After you've met these requirements, then you can reference some of the recommended practice in Appendix B.


Let's get into a bit more detail about some of the other tools in EagleCAD. For one, you can't delete traces, you rip them using the Rip-Up Tool . You can delete parts and holes and other added components, but not traces.


Making Planes: Eagle also allows us to use shapes/planes on our PCB. Nodes like GND and VDD (power) are best drawn as planes, for many reasons. If you're curious, check with your TA.

To do this, make a box (usually just inside the PCB edge) using the Shape Tool . Don't overlap end point and start point, you might get some ugly errors if you do, just connect the end right to the start. You should have a nice dotted box around your circuit like so.



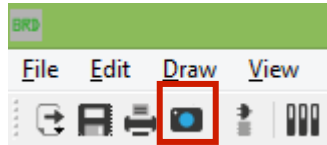
Now use the Name Tool to name the box. The name is going to be whatever signal you want to fill that area with, so you have to choose a net name that you assigned in the schematic view. Hit RatsNest to see the outcome. A large majority of the board should become entirely red (if top side) or blue (bottom side). Right click on the box (the outline, not the filled in area), and select **Properties**, then under **Polygon**, change **Isolate**. This is the parameter that determines how much separation exists between the fill and the other signals on that level. Be careful about making it too small, as you'll find it harder to solder to, as well as increasing the risk of coupling signals together. A decent value is 50mil.

Almost done! Just like in the schematic how we had the Electrical Rule Check, here we have the Design Rule Check . While the ERC looks our circuit over and sees if we have anything wired oddly, the DRC not only tells us if we laid out anything strangely, it also warns us if we did anything that isn't possible to be manufactured. While Eagle may allow you to place parts apart with nanometer, good luck trying to find someone who will actually make that! Most services, like our ECE shop, offer a design rule listing so you can go in and find out exactly how wide to make your traces, or how far apart to spread your parts. Follow the link earlier in this section to download the shop's DRC rules.

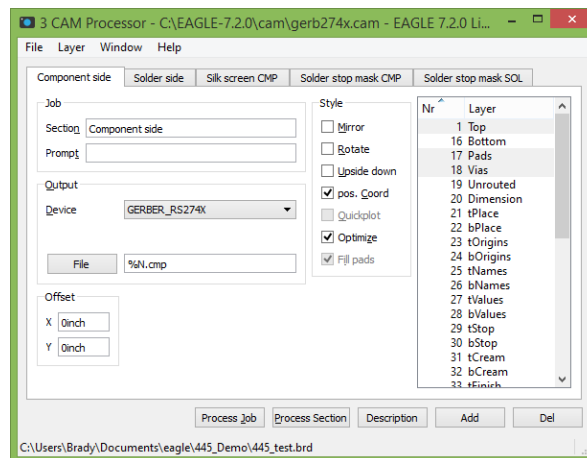
The last tool we haven't really talked about is the AutoRouter . This is for good reason – it tends to suck. PCB layout, and EDA in general, is a huge unsolved problem in terms of automation. While with digital solutions there is enough reduction that the problem is more manageable, analog layout is still a bit of an art that hasn't been completely mastered. The AutoRouter will most likely find a solution, but it's far from optimal most of the time.

Creating Gerbers

While not required for the assignment, it's good to learn how to go from the EagleCAD board file to gerber files. The [gerber format](#) is a binary vector image filetype that is the common medium for manufacturers. Plenty of free and online gerber viewers exist if you wish to view what is actually being outputted, but it effectively just slices the PCB layer by layer and lists each x by y point as a region.



Select the CAM Processor (highlighted above) to get started. Select **File, Open, Job** and select **gerber274x**. The window should populate with several tabs, each representing a new output file. Each output file is a new “slice” of the board. Pressing Process Job at the bottom will create all of these files in the same directory as your project.



Why are there so many files for a two layer board? Well you can see what each “slice” actually is in the layer tab on the right. We have our top and bottom solder, as well as soldermask, silkscreen, and a drill file to notate all the holes in the board. Fancier boards require even more layers, with items like glue points and solder stop points.

EAGLECAD LAYER REFERENCE

*ECE 445
University of Illinois at Urbana Champaign
Department of Electrical and Computer Engineering*

Many layers are prefixed with a t<X> for Top <X> or b<X> for Bottom <X>. The layer

Layer Number(s)	Name	Purpose
1-16	Top -> Bottom	The wire routing metal layers for the PCB. Also called “traces” on the board. For two layer boards, only use the Top (red) and Bottom (blue) layers to draw connections. Also shows the surface mount pads for respective sides.
17	Pads	Somewhat confusingly, EagleCAD defines pads as every through-hole connection of a part. While we often call this a “via”, those have a separate definition.
18	Vias	When a trace from layers 1 to 16 switches to another layer (ie, from [1] Top to [16] Bottom), a green via is created to vertically short these layers. By default, this is a full board (top to bottom) short. While it is possible to do blind vias (short only [3] to [5], for example), the cost is extremely prohibitive.
19	Unrouted	This is the airwire layer that shows all the “birds-eye connections” for all the parts. A finished PCB design should have nothing on this layer. The Ratsnest button will tell you how many connections are left.
20	Dimension	This sets the physical size of the board. Anything outside of this shape (usually rectangular, but can be anything) will not be manufactured.

21-22	Place	This is a decorative layer for showing parts physical profiles. This layer is drawn with silkscreen and is generally used to make sure parts do not physically overlap.
23-24	Origins	The origin is the small cross (+) signifying the center of the part. This is mostly used to help the user select part they intend to in the design phase, and is rarely manufactured.
25-26	Name	Self-explanatory. Can be moved with the “Smash” tool.
27-28	Value	Self-explanatory. Can be moved with the “Smash” tool.
29-30	Stop	For service shop boards, this layer does nothing. But when making a board at a professional manufacturer, the entire board is covered in soldermask by default. This layer specifies the soldermask free sections (soldermask STOP), and effectively shows the free to solder to areas. This is in general slightly larger than the pad or via itself.
31-32	Cream	For service shop boards, this layer does nothing. If you’re making a stencil for the board, this layer defines the stencil outline. This should be the same as all the top and bottom pads.
33-34	Finish	For service shop boards, this layer does nothing. For high quality boards, this sets the plating and finish used.
35-36	Glue	For service shop boards, this layer does nothing. When using advanced surface mount soldering techniques, such as wave soldering, manufacturers often specify placing glue to keep some parts in place during the extreme heat stress. This is generally placed in the center of parts.
37-38	Test	For service shop boards, this layer does nothing. This is for establishing automated testing points.
39-40	Keepout	This layer is used with polygons. If parts are placed within this shape, the DRC will throw errors.
41-43	Restrict	This layer is used with polygons. If traces are placed within this shape, the DRC will throw errors.
44	Drills	This layer shows all conducting (plated) holes in the board.
45	Holes	This layer shows all non-conducting holes in the board. Generally for mechanical for mounting.

GUIDELINES FOR LAYOUT

ECE 445

University of Illinois at Urbana Champaign
Department of Electrical and Computer Engineering

1. FOLLOW THE PCB SHOP'S RULES:
<http://eshop.ece.illinois.edu/pcbdesign/designreq/designreq.php>
2. Figure out if your signal is high-speed. A good rule of thumb is described below:
 - a. Figure out the T_{rise} of your fastest signal (usually a clock)
 - b. Calculate the T_{delay} of that signal – distance traveled divided by velocity (velocity will depend on your dielectric material- an internet search will usually turn up common effective dielectric constants, ϵ_r . You can approximate FR-4 to be 4.7, although this varies with frequency. $\text{Velocity} = \frac{c}{\sqrt{\epsilon_r}}$, c = speed of light)
 - c. Ratio of $T_{\text{rise}} / T_{\text{delay}} > 6$: your circuits will behave as expected from your schematic (i.e. wires can be treated as shorts). **Go to #10.**
 - d. Ratio of $T_{\text{rise}} / T_{\text{delay}} < 2.5$: you will need to think of your wire as a transmission line, **go to #3.**
 - e. If your ratio is somewhere in between, you should discuss with your TA. To be safe you could still abide by the practices for high-speed layouts.
3. You would want to control impedance, usually to 50ohm if single-ended. You could do this by figuring out the thickness of the PCB dielectric and its material, and then using an online calculator for determining the width of your trace. Talk to the PCB shop to let them know you need controlled impedance- any GND all around the trace will matter.
4. Ideally you want solid GND plane underneath the entire length of the trace from transmitter to receiver.
5. Give yourself plenty of separation between the traces (especially if you have other noise-sensitive circuits). A couple of mm minimum, greater space where possible.
6. Avoid changing materials between transmitter and receiver (try to stay on one PCB, try not to solder wires in-between or go through a connector).

7. Avoid via transitions where possible. If you have to dive between layers make sure the return path has a way to follow the signal.
8. Terminate your lines- usually a series resistor close to the transmitter is used. Talk to your TA about the proper value in your application.
9. Design your power delivery very carefully- you need plenty of bulk and high frequency (usually smaller-value) capacitors in parallel for decoupling. These need to be close to VDD pins on your chips spewing out high-speed signals.
10. For noise-sensitive circuits (analog audio, sense lines, anything extremely low voltage or high impedance, etc), group these as far as possible away from digital logic with high clock rates.
11. Consider isolated GND planes if you have an analog portion and a digital portion. True electrical isolation can be done with optocouplers and transformers, but a simpler solution is a GND moat, where the quieter analog GND is a copper island enclosed by a “moat” of no-fill, separating it from the surround digital GND. These are connected at one point only, to prevent too much DC voltage drift. All analog signals (on their layer above or below the plane) are referenced to the analog GND and vice versa for digital signals- no signals cross over the moat. Talk to your TA about the implementation.
12. Consider using separate power sources for the sensitive circuits- an LDO is a good option.
13. If you have a part that’s in a small or difficult to solder package, try using a socket or making a breakout board to solder into your main board. This way, if you mess up soldering the small connections, you won’t break your main board.
14. If you have the room, consider making power planes. Create shapes just like for the GND plane, but this time, name them your VDD net names. Each VDD with substantial current consumption (DC or transient) should ideally have its own plane, with plenty of bulk decoupling at the output of its own regulator.