

REV PCB Design Workshop

Max Charlamb - 2019

Goals:

1. Learn to create board designs in Eagle CAD
2. Use the OtherMill to create a simple PCB
3. Learn to solder parts
4. (Time permitting) Write code and upload to Arduino

Contents:

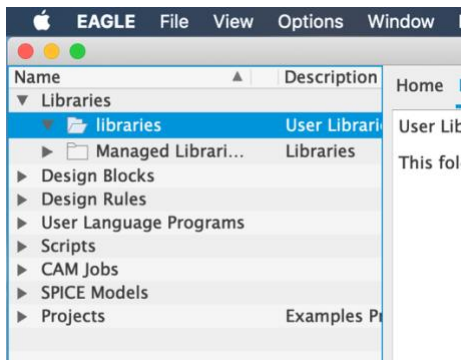
1. [Eagle Basics](#)
 - a. [How to Import Libraries](#)
 - b. [Setting up your first Project](#)
2. [Simple ATTiny Board](#)
3. [OtherMill](#)
4. [Soldering](#)
5. [Programming](#)
6. [Advanced PCB Design Considerations](#)

1. Eagle Basics

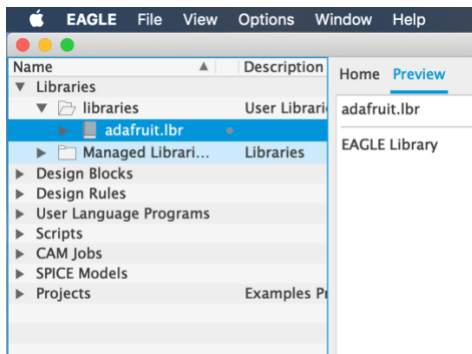
This section consists of Eagle basics: how to create a project and how to import a library. If you are familiar with Eagle feel free to skip these steps or use them as reference.

How to import Libraries

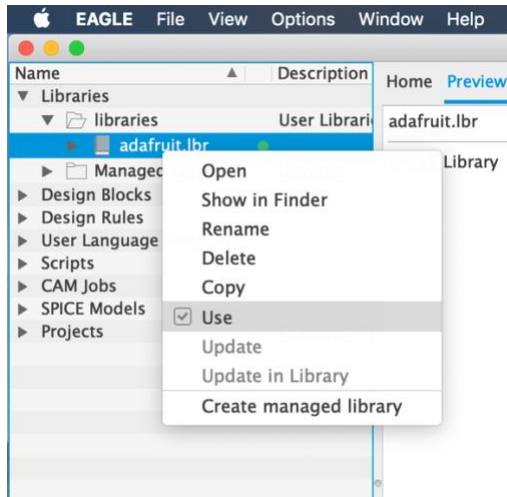
1. Open Libraries/libraries on the Eagle home screen.



2. Drag lbr file into the libraries. I am using the Adafruit library, which can be found [here](#), as an example.



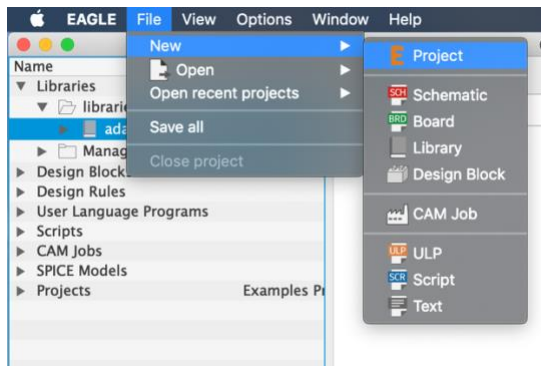
3. Right click the file and select "use." Make sure the dot to the right of the file name is green.



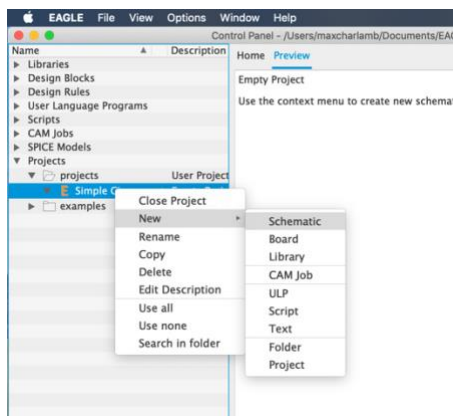
4. Now the library will be available to use in all local projects. Sometimes projects must be closed and reopened to see new libraries.

Create your First Project

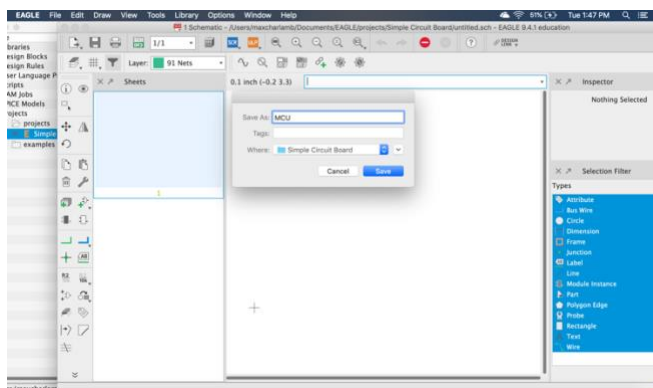
1. Select File -> New Project



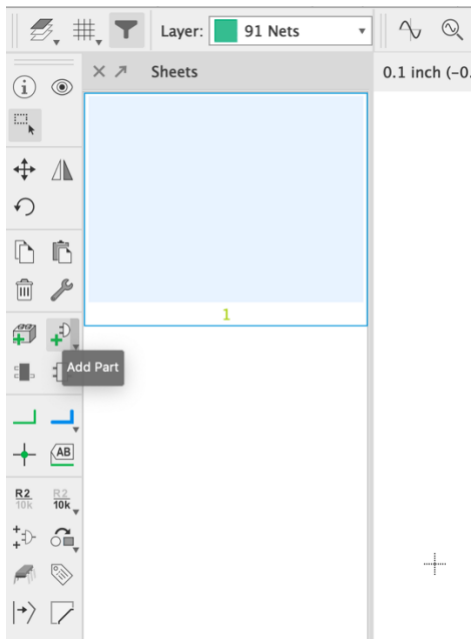
2. Right-click New -> Schematic. This is also an opportunity to rename the project.



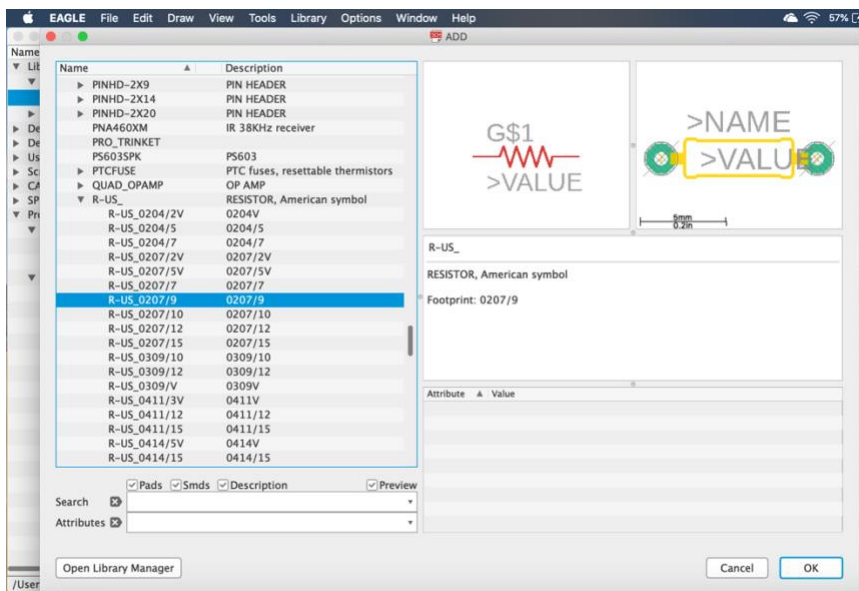
3. A new schematic window will pop up. This is where you will design the connections in the circuit. The schematic editor allows components to be added and wires to run between them. Make sure to save (cmd + s) the file!!!



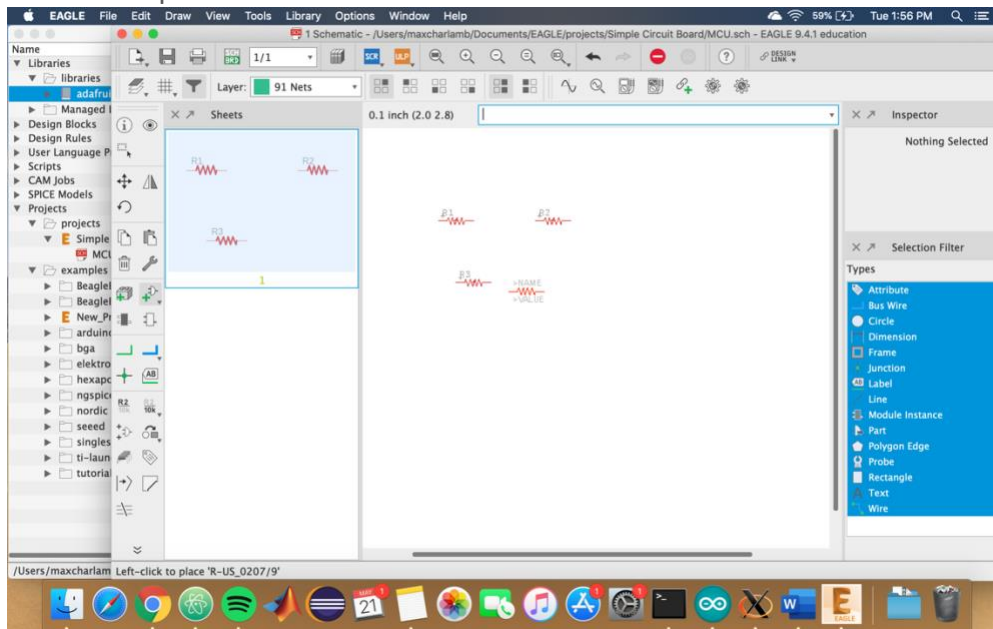
4. The first step to creating a circuit is to import a component. Left-click the "Add Part" button half way up the left bar.



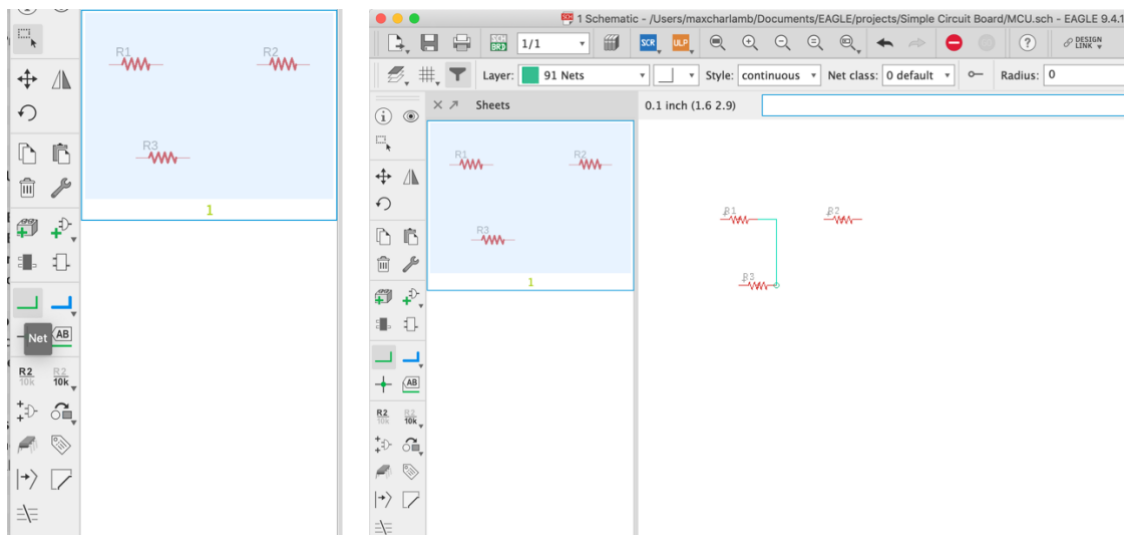
5. The part selection window will pop up. This window allows components in any library to be added to the schematic. We will add a resistor from the library "adafruit/R-US_" with the name R_US_207/9. If you do not see this library refer to this guide. Navigate to the component and select "OK."



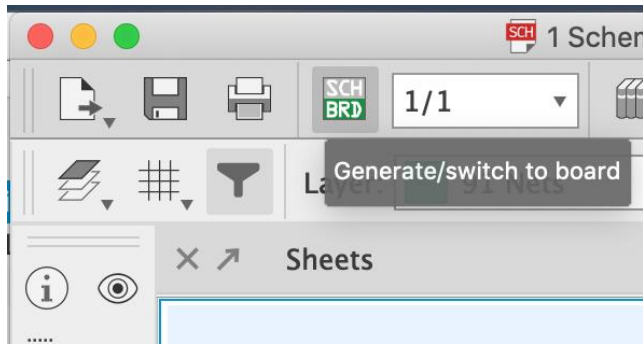
- Left-clicks place down the component. Press Escape twice to stop placing down components.



- These resistors have two pins on them. One for each side and they are identical. These pins can be connected using a "Net" which will later become a wire. Select the "Net" tool from the left sidebar and click on the end of two resistors.



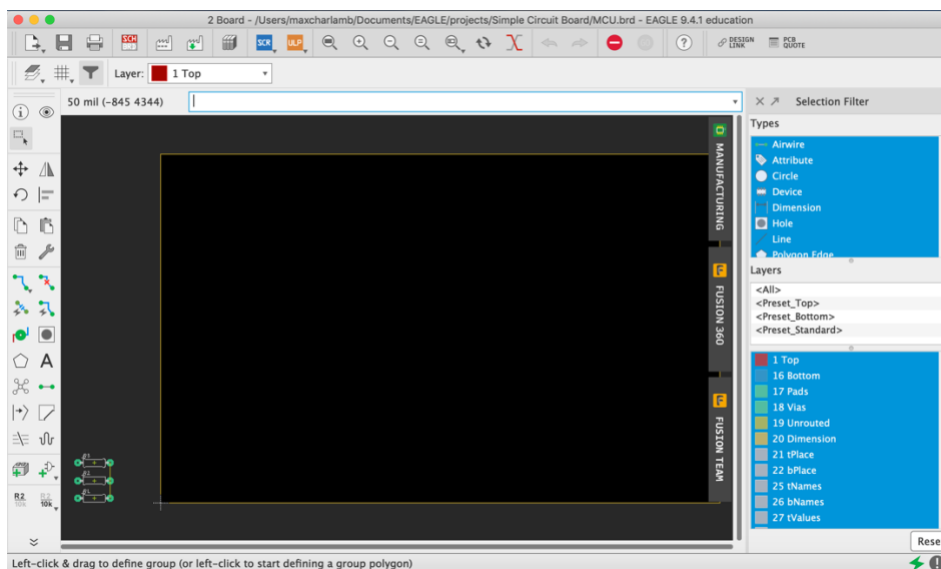
- Now that we have a schematic, we can create a matching board. The schematic is the plan for the circuit while the board is how that plan layed out on the physical PCB. Press the "SCH/BRD" button on the upper bar.



9. If this is the first time opening the board, a pop-up may ask to create the file. Select yes and continue. When editing either the board or the schematic it is important to have both open. This will ensure that they are synced and prevent issues down the line.

Important parts of the board designer

- In the bottom left you can see the three resistors that were on the schematic
- Two of them have a small yellow "airwire" running between them. This symbolizes that they should be connected according to the schematic.
- The large black rectangle with yellow outlines is the default board shape. This can be edited by selecting the move tool and dragging the lines.



10. At this point the project should be set up and you can move to the ATTiny Project tutorial.

2. Simple ATTiny Board

Now that you know the basics of Eagle, we will walkthrough making a functional circuit board. We will use an ATTiny85 microprocessor. This in the same family as the ATmega328p used on Arduino Uno's and is able to run the same software.

Parts

This circuit will include two buttons, an LED, a couple resistors, a capacitor, the battery, and the microcontroller.

ATTiny85 DIP Package
2x4 IC Holder
5mm LED
2 220 Ohm Resistors
1 4.7uF Capacitor
2 6mm Tactile Switches
20mm Button Battery Holder
CR2032 Battery

Libraries

Install the following libraries to model the components we need:

[adafruit](#)

[PTH BATTERY](#)

[_hnh_attiny25_45_85](#)

[switch-tact](#)

These can be installed by following the library installation tutorial.

Design Process

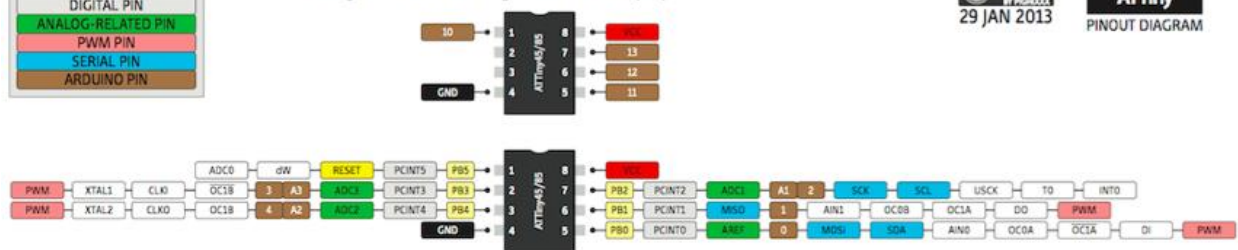
The heart of the circuit is the ATTiny85. The microcontroller will control everything about the circuit: read the input, control the output, and allow code to be uploaded.

Pins:

The pin out of the ATTiny85 is needed as a reference to design around the chip. The company that designs the chip publishes a PDF ([here](#)) that lays out every single detail of the microcontroller. That is way more information than we need right now so it is easier to find a nice graphic with the pinouts.

LEGEND	
GND	
POWER	
CONTROL	
PORT PIN	
ATMEGA328 PIN FUNC	
DIGITAL PIN	
ANALOG-RELATED PIN	
PWM PIN	
SERIAL PIN	
ARDUINO PIN	

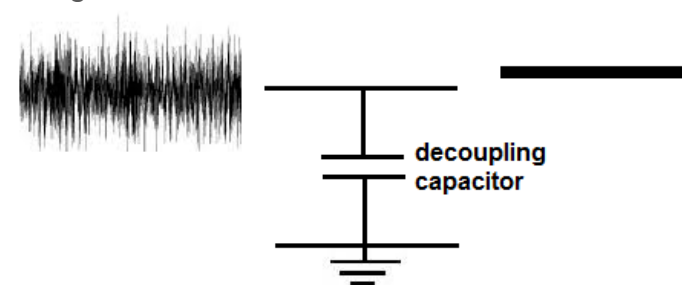
Using Arduino as ICSP Programmer for ATtiny45/85



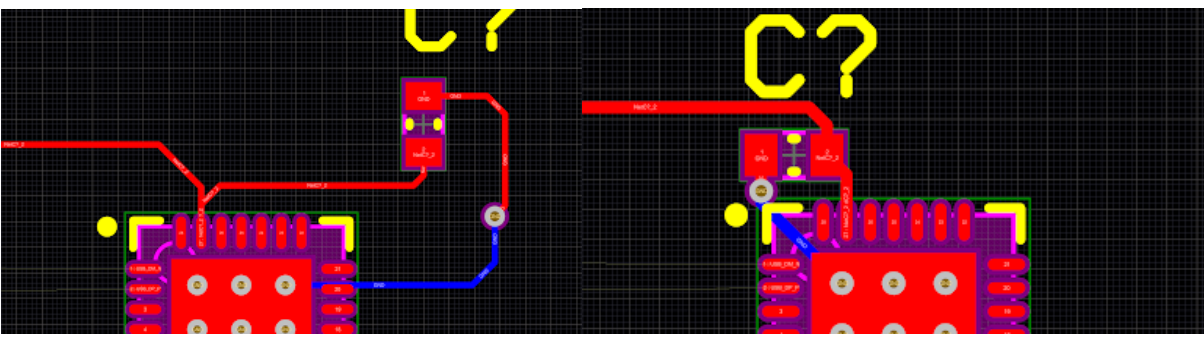
There are different kinds of pins. Pin 8 is always power, pin 4 is always ground, and pin 1 is reset. The other five pins can be used for other purposes such as input/output pins. A couple pins have a salmon colored tag with "PWM." This means that the pins can be toggled on and off to change the intensity of a light or produce a buzzing sound.

Decoupling Capacitor:

When incorporating microcontrollers into a design it is important to consider power fluctuations. Any rapid change or spike in power could cause damage or cause the microcontroller to restart. A "decoupling capacitor" filters out spikes and keeps a steady voltage.



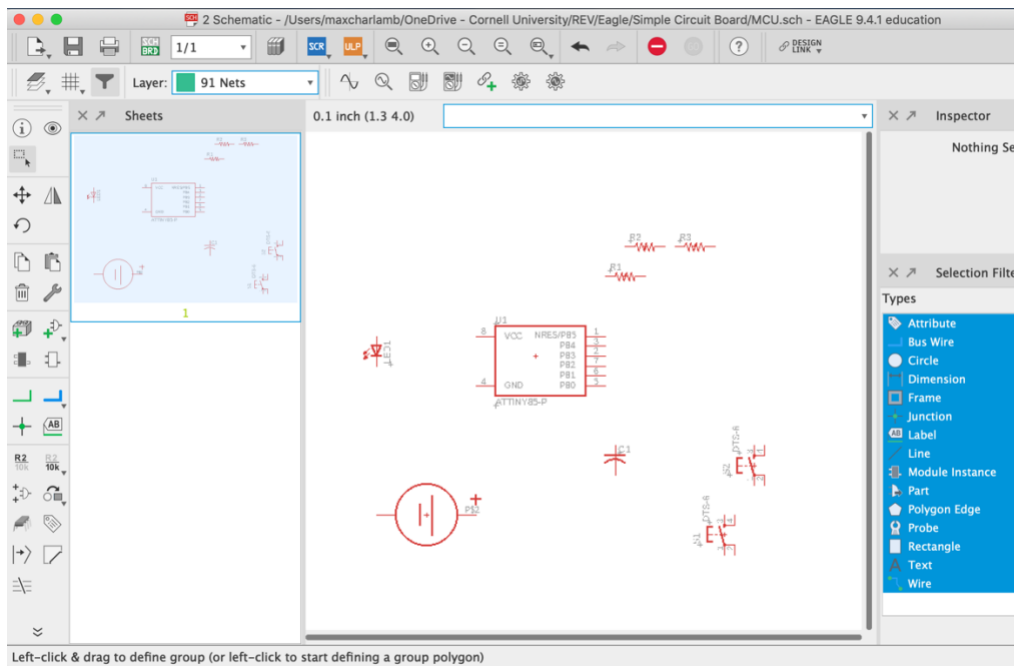
We will put a capacitor between ground and power very close to the microcontroller power pin. **The left is a bad example, the right is a good example.**



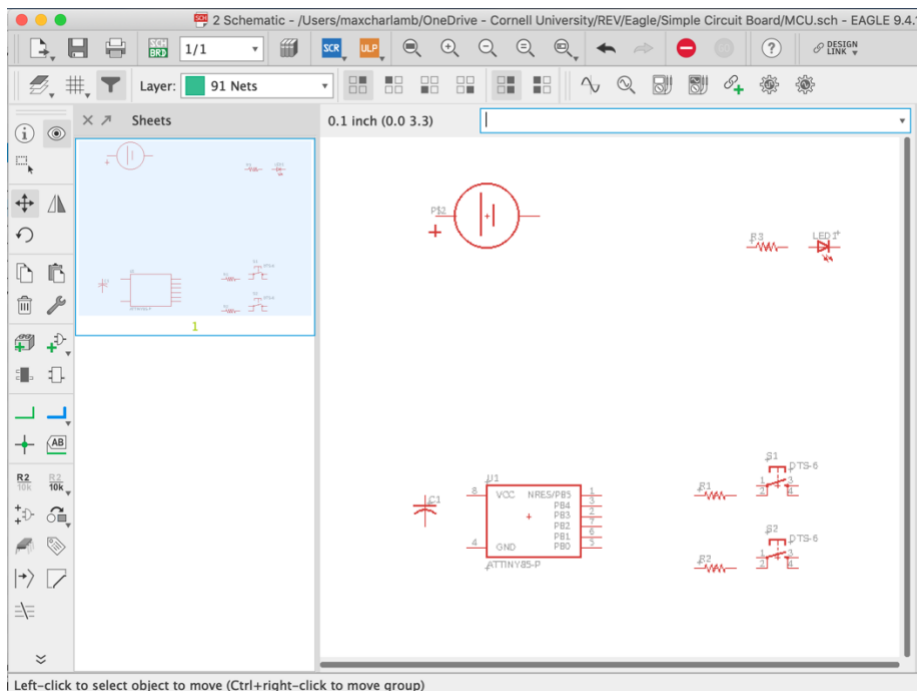
Schematic

Below is a list of the components that represent the parts we need. Add them to your Eagle schematic by copy and pasting the bold names into the part search bar. The schematic should look something like the image below.

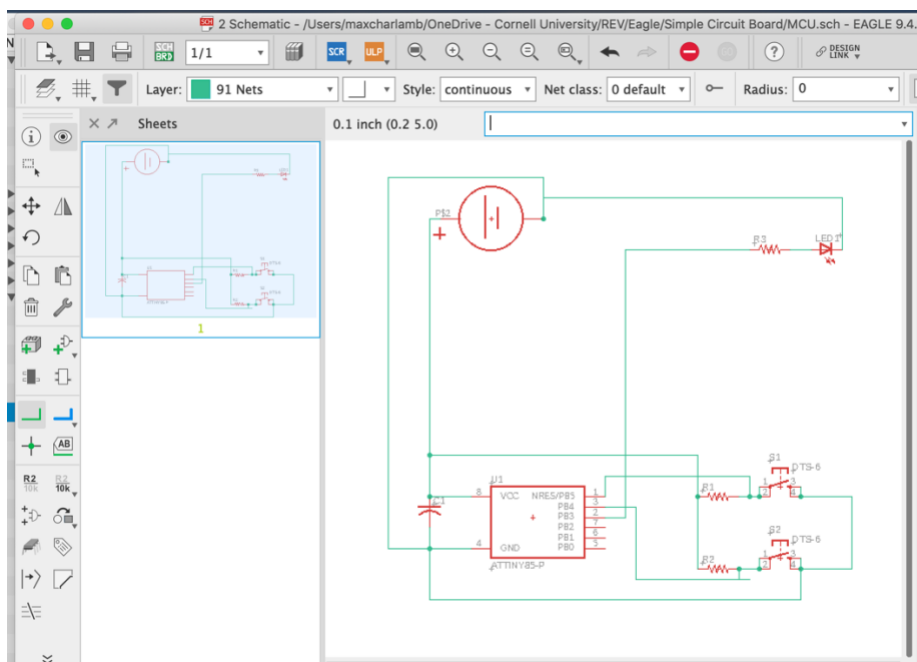
- 1x **ATTINY85-P** (IC)
- 1x **BS-7** (Battery)
- 2x **DTS-6** (Switch)
- 1x **LED5MM** (LED)
- 3x **R-US_0207/9** (Resistor)
- 1x **C-US075-052X106** (Capacitor)



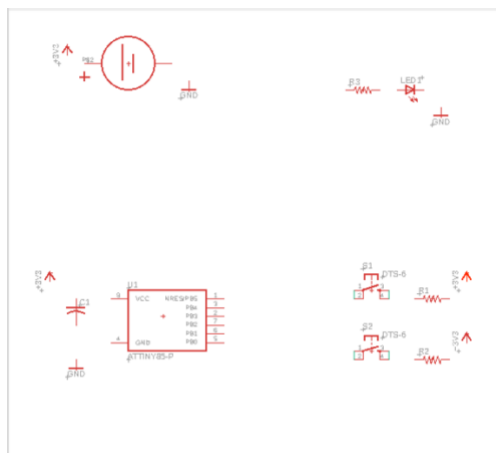
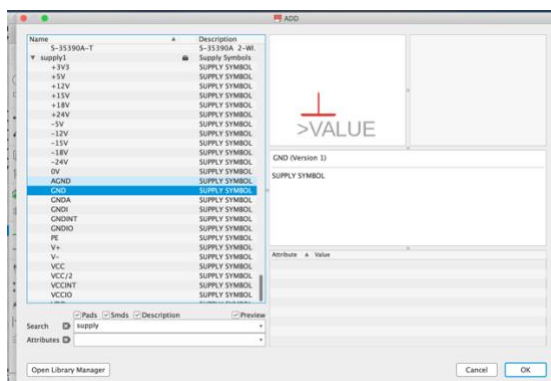
We can use the move and rotate tools to orient our components in a way that makes sense. The resistors are near what they will be connected to and the bypass capacitor is close to the microcontroller.



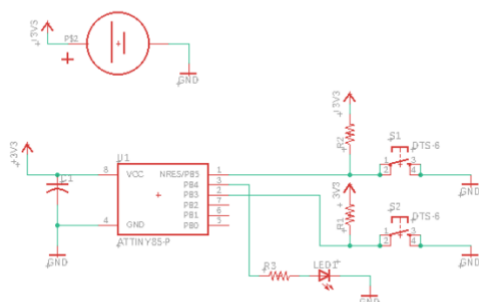
Now we can start adding "Nets" between components and actual create our circuit. However, this quickly turns into a jumbled mess of wires.



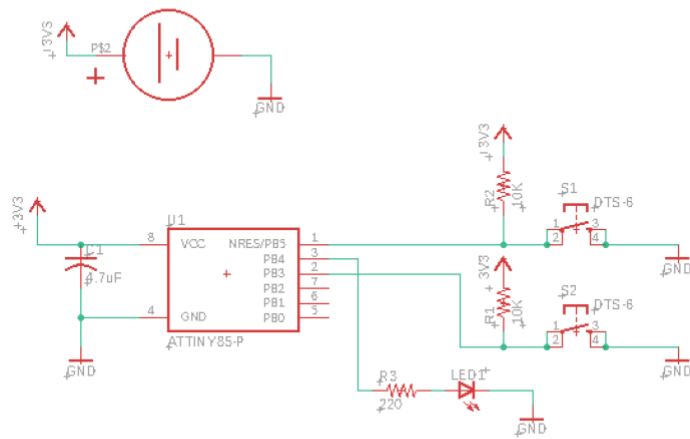
To simplify the schematic, we can add "ports." These symbols represent components that are connected but don't need a mess of wires running between them. We will add a ground port and a power port.



Now we can move a few parts around and make a schematic that is much easier to read.



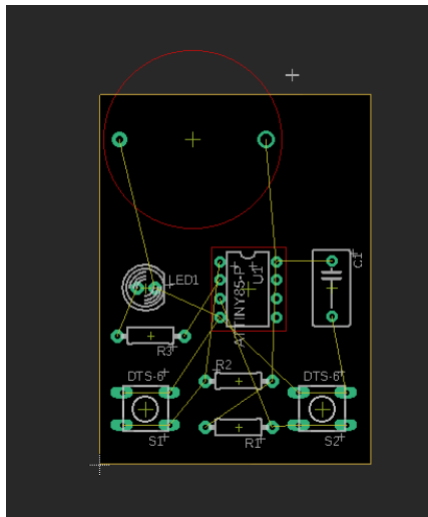
The last thing to do before moving over to the schematic and laying out the board is to add values to the components. Press the "Value" button on the lower half of the left bar. Click on the resistors and capacitors to add their value. R1/R2 should 10K Ohm, R3 should be 220 Ohm, and C1 should be 4.7uF.



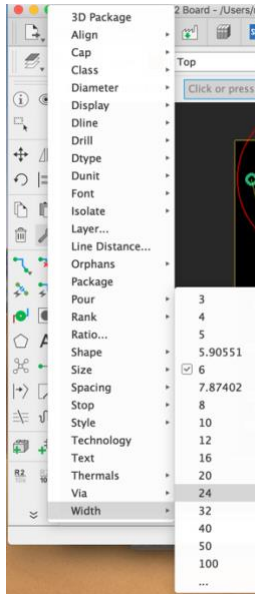
The schematic is complete!

Layout

Let's move on to board layout. Press the "SCH/BRD" button on the top bar. The default size for the board layout is WAY too big. Drag in the yellow lines to make it smaller and place the components inside.



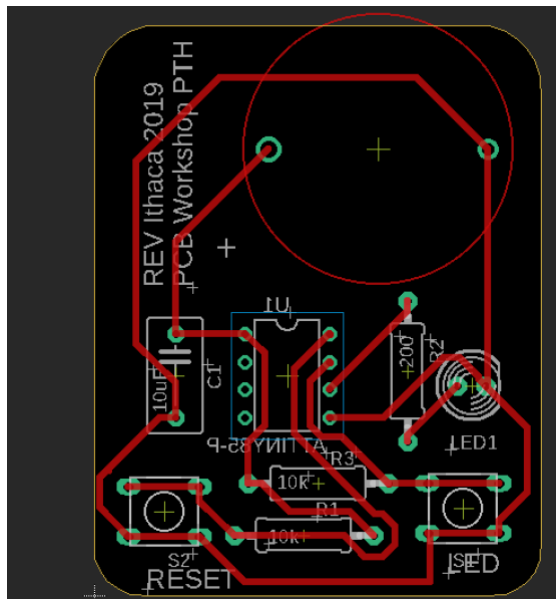
Those yellow lines mean that the components will be connected together. Be sure to press the "Ratsnest" button on the left bar which will update these lines to the shortest paths. In preparation to route the traces, make the trace size 24. Download the OtherMill DRC from [this page](#) and unzip it. Install the file named "OthermillDRC_1_32_v2.dru" by going to Tool -> DRC -> Load and navigating to the file that was downloaded in Eagle.



Flip the microcontroller component using the mirror tool on the left bar. Because this board is milled on the OtherMill we are mounting the components on the back. It doesn't matter for the other components so they can stay unmirrored.

It is common to route most of the air traces then not be able to finish the last few. In this case it is sometimes necessary to start from scratch and move the components around.

This is one solution I found. It isn't the prettiest routing, but it gets the job done.

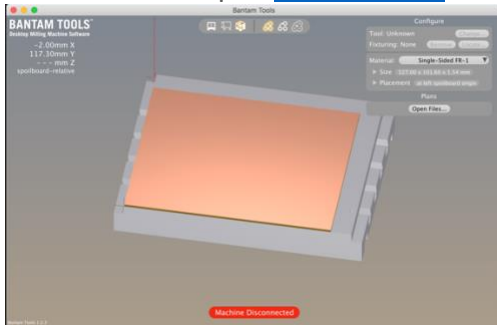


3. OtherMill

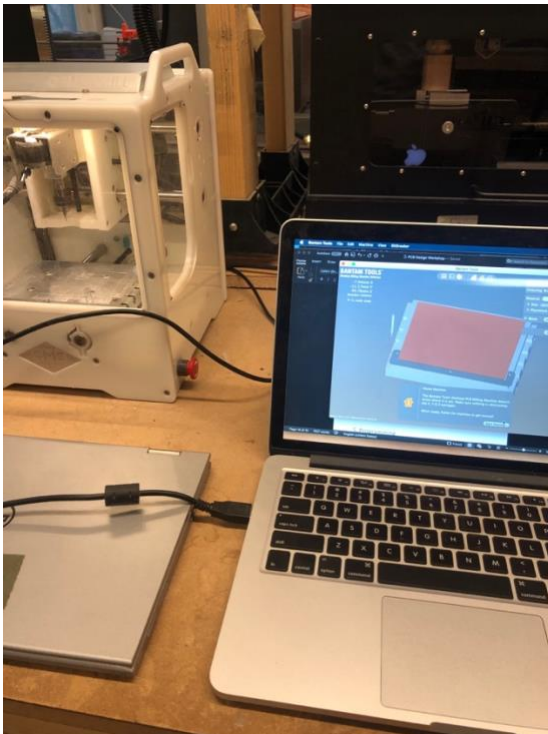
The OtherMill is quite limited in complexity of design which it is able to process. It can only produce one- or two-sided boards. Even then, with quite a few drawbacks. Trace size must be large, distance between traces must be large, and vias (holes between sides of the board) can't be plated. Even with all these drawbacks the OtherMill has an important role in the lab. It allows rapid prototyping of PCB's. Turnaround time is only minutes instead of days if the boards were shipped out to a manufacturer.

Printing

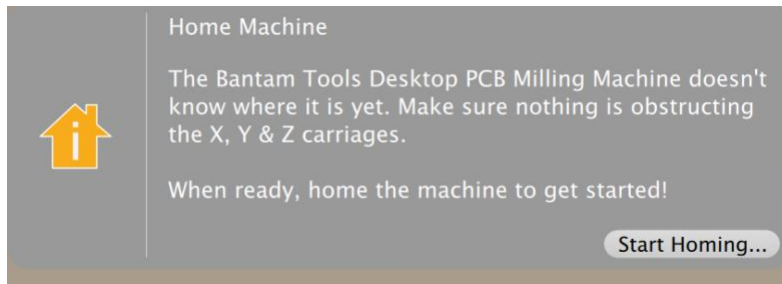
1. Download and open [BantamTools](#)



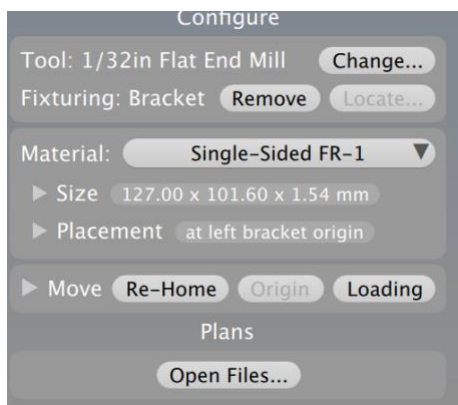
2. Turn on OtherMill with switch in back
3. Connect to the OtherMill over USB



4. Follow on screen prompt to Home tool head



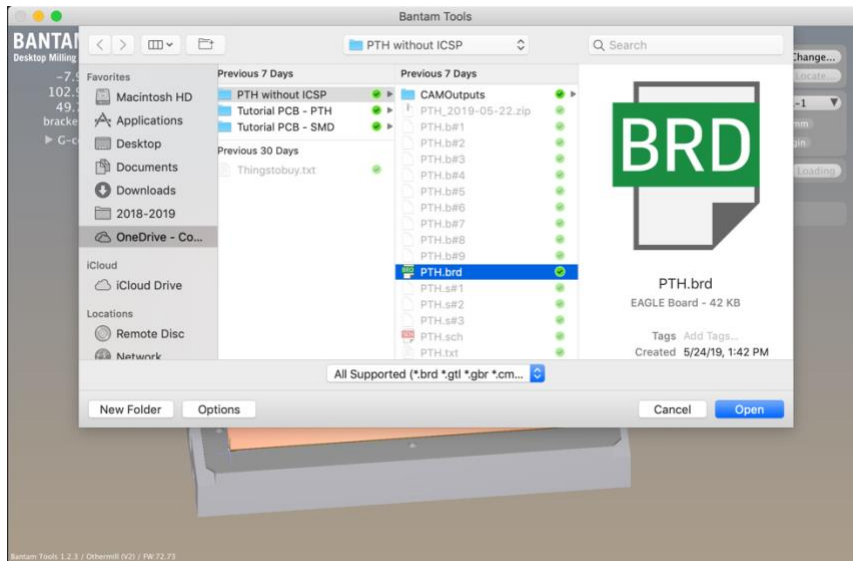
5. Press "Loading Position"



6. Load PCB Blank with double sided tape on back



7. Select "Open Files..." and navigate to a .brd Eagle file



8. Position board on the PCB blank
9. Run job!

4. Soldering

How to Solder

Check out these two YouTube videos on soldering:

- [How to Solder](#)
- [Soldering Technique](#)

Soldering is an art and takes practice! Use this simple board to get used to soldering through hole components. Most boards have solder masks which prevent solder from sticking to places it shouldn't. This board does not have a solder mask so it will be a little tricky.

Soldering Tools

- Aoyue Soldering Station – Includes a soldering iron (with suction) and heat gun
- Hanco Soldering Iron – Measures temperature in Fahrenheit



- Solder Sucker – Use the soldering iron to melt the solder and then use the solder sucker to remove solder.
- Brass Wire – Used to clean soldering tips

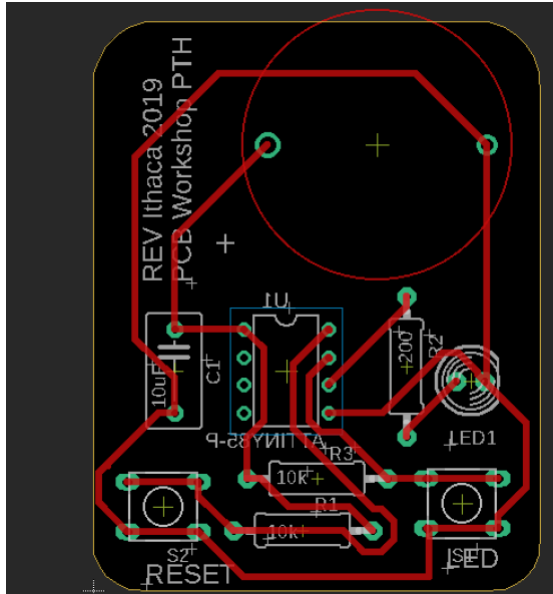


- Aoyue Smoke Absorber – Ventilation



Soldering

Use this as a reference to solder components onto the board. Keep in mind the components go on the back of the board and you solder on the copper side. The microcontroller, LED, capacitor, and battery are directional. For the LED and capacitor, the longer lead is positive. The microcontroller has a small dot that corresponds to a dot on the schematic. The batteries positive terminal has a larger hole.



Final Product

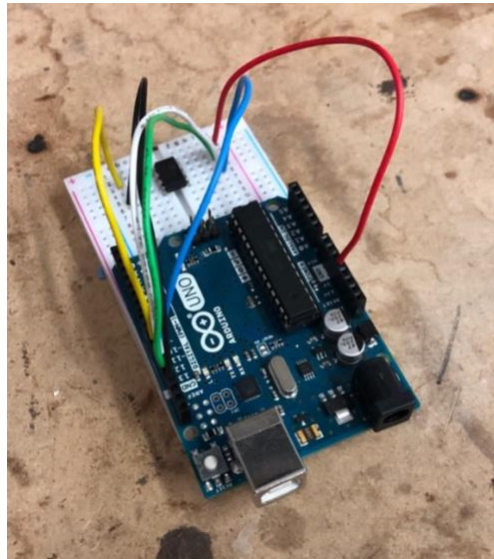


5. Programming

You can write code in nearly text editor but using the [Arduino IDE](#) is often the most convenient for small projects. Once you have software writing it can be uploaded to the ATtiny using [this guide](#). When programming for this project keep in mind that LED output is pin 4 and LED button read is pin 3.

This design does not include a way to program the ATtiny while it is on the board. This is not necessary because the whole DIP package can be removed easily. On other boards ICSP or USB headers allow a microcontroller to be programmed. I have created a small rig to assist in programming the ATtiny chip.

Feel free to make your own code or use mine which you can download [here](#).



6. Advanced PCB Design Considerations

Contents:

- [Trace Width](#)
- [Routing](#)
- [Polygon Pours/Planes](#)
- [Vias](#)
- [PCB Manufacturers](#)
- [Layers](#)
- [Differential Routing](#)
- [Mounting](#)
- [Programming](#)

Trace Width

Trace width depends on the purpose of the trace. Wider traces have lower impedance and are needed in high current applications (20-30mils). Signal traces can be smaller (6-10mils). When designing boards for the OtherMill the traces should be about 24mils for durability and tolerance during manufacturing.

Trace width will be constrained when differentially routing.

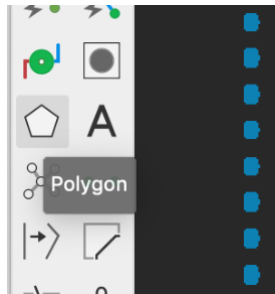
Routing

Routing often takes multiple tries and sometimes requires restarting from the beginning. Routing is just like writing. Constant revision is crucial to success!

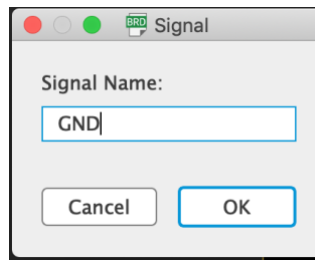
Polygon Pours/Planes

Polygon pours are an area of copper. This is most commonly used for creating a ground plane, an area of ground to reduce noise. They can also be used for very high current applications such as between a battery and voltage protection.

To add a polygon pour in Eagle, use the polygon button on the left hand bar.



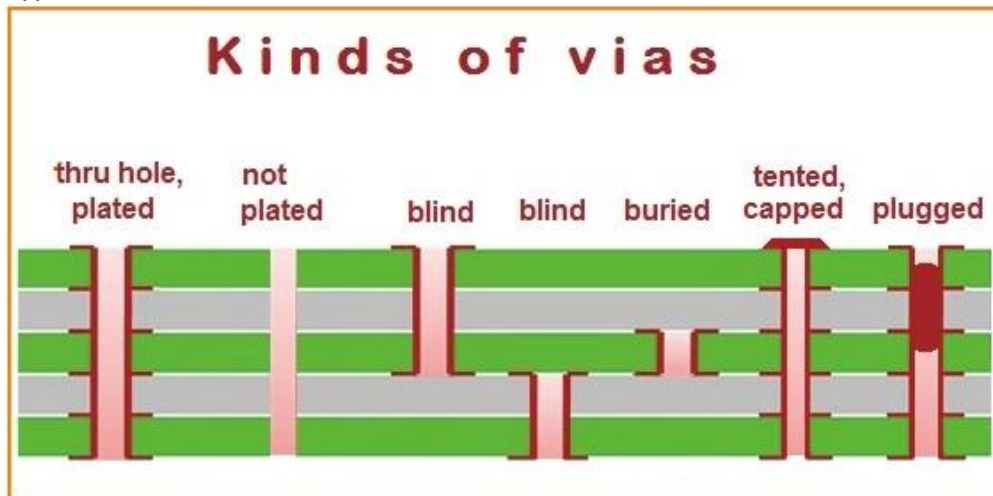
Then draw the shape of the pour. Eagle will prompt you asking for the signal name. Enter the exact name of the net that you want the pour to connect to.



Vias

Vias are small holes that connect different layers of the board. The OtherMill can only make not plated vias.

Types of vias:



PCB Manufacturers

Most manufacturers take several weeks to get a finished product back. Rush service is often extremely expensive.


- OSHPark – Quick American PCB Firm. Good for three run prototypes.
- Sierra Circuits – California based PCB Firm which is expensive but can manufacture complex designs. Allow for full turnkey service.
- JLCPCB – Chinese PCB manufacturer. Ten copies of a board for \$2 but shipping is expensive and slow.

Layers

This project only uses a single layer. Two layers will be standard for most simple boards that will be shipped out. This allows for traces on the top and bottom which makes routing much simpler. Four-layer boards allow for internal ground and power planes and allow for more complicated designs. The only downside for higher layer boards is cost of manufacturing.

Internal layers are often planes rather than signal layers. This means that the entire plane is a single net, such as ground or power. The copper is thinner and doesn't allow for traces. Some manufacturers allow for internal signal layers which are just like the top and bottom layers.

Here is an example layer stackup for a four-layer board.

SOLDER MASK			0.0005
TOP SIGNAL		1 Oz	0.0014
DIELECTRIC			0.0035
PLANE		1 Oz	0.0014
DIELECTRIC			0.0470
PLANE		1 Oz	0.0014
DIELECTRIC			0.0035
BOTTOM SIGNAL		1 Oz	0.0014
SOLDER MASK			0.0005
Total Thickness			0.0606 (inches)

Differential Routing

Two parallel traces that carry one signal encoded in the potential difference between the lines. This counteracts noise and is required for high-speed transmission such as USB or HDMI. When using differential routing it is important to also impedance match with the standard being used. Use an online calculator like [this](#) to find trace width. When routing differential traces the PCB should be four layers so that the traces can be referenced to a ground plane.

For more information on differential pairs in Eagle reference [this](#) guide.

Standards:

HDMI – 100 Ω
Ethernet – 100 Ω
USB – 90 Ω

Mounting

Usually M3 Screw holes, 3mm Diameter.

Programming

There are several ways to upload code to a microcontroller on a PCB. USB is an option, but overly complicated for most applications. The solution is ICSP or in-circuit serial programming header. The 6-pin ICSP header below is standard.

