

# Learning Eagle Day 2

Heyo welcome to learning Eagle Day 2! Sorry for the break between the tutorials, I've been really busy with some college stuff :)

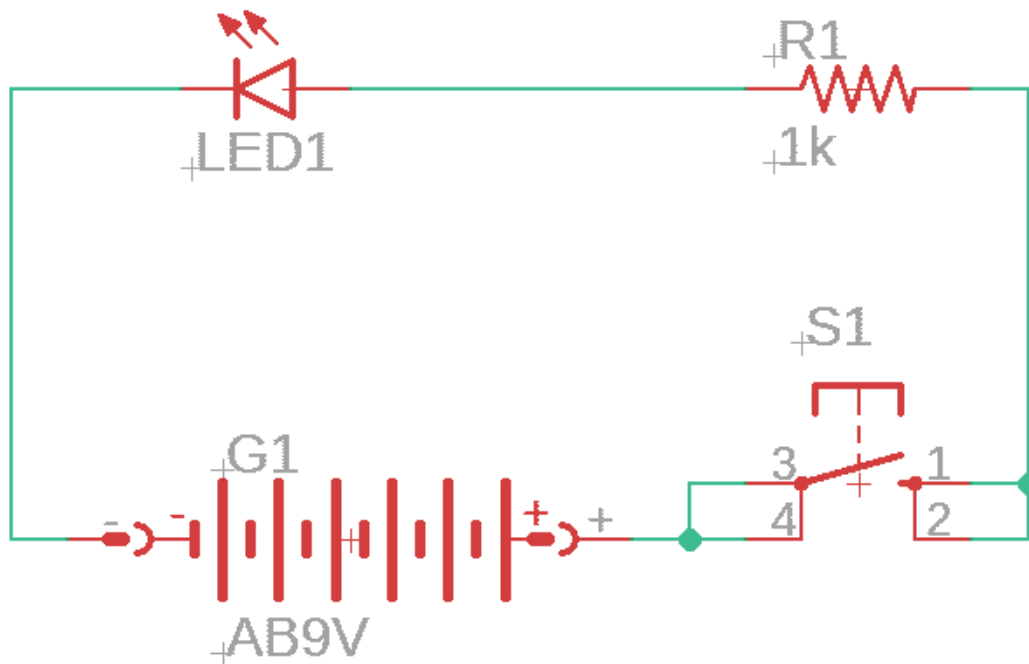
Anyways, today we are going to cover:

- Advanced schematics stuff
  - Layout style/how to make it not a big mess
  - Eagle internal vs external libraries and finding new components
- Board design!
  - Board layers/stackup
  - Terminology
  - Selection filter, design manager, inspector tools
  - Basics of efficient component placement
  - Routing - manual vs. autorouter
  - Text layers/silkscreen
  - DRC & ERC checking
- Moving towards future
  - One more lesson on some finer points required to build real things
  - Next project: After the next meeting, you'll be tasked with making your own arduino clones! More info coming next week. Basically, this should be the last instruction document for y'all

*End of day goal:* Able to create a relatively basic design in eagle from beginning to end (idea -> properly exported file ready for production)

## Part 1: Advanced Schematics

Ok, let's start with a recap from last week.



This is the simple schematic we created, using different libraries built into eagle. Today, we'll be creating a more complex circuit and routing it. The circuit that we'll be creating is designed to light up an LED for a specified amount of time using a specialized timer chip.

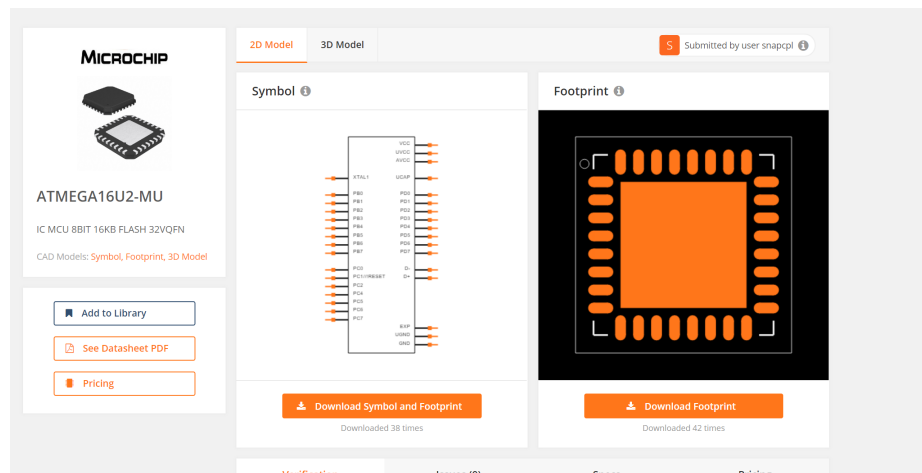
## Adding External Libraries

If you don't remember, Eagle libraries contain the parts you use to build up the design. Eagle comes with many components in the built in libraries, most commonly things like battery connectors, LEDs, switches, and a decent selection of chips and specialty components. However, it's a good thing to know how to install external components.

How to install:

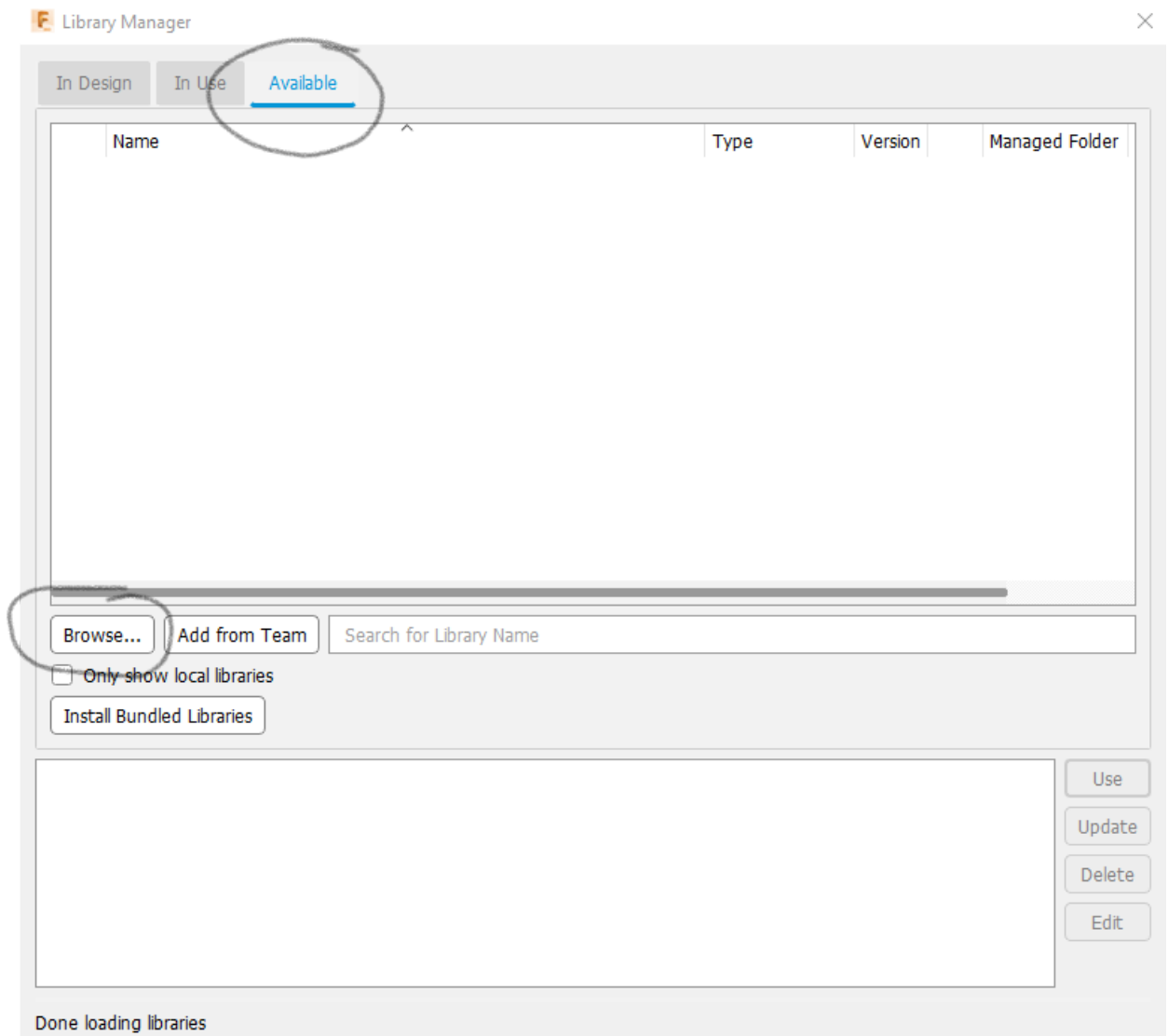
- 1) Highkey just search in google "<insert part name here> eagle". Then find a link to either SnapEDA or Octopart

For example, searching for “atmega16u2 eagle” gives me <https://www.snapeda.com/parts/ATMEGA16U2-MU/Microchip/view-part/>



- 2) Download the LBR (stands for library) file
- 3) Go into fusion and open your library by going under library in the top bar then click open library manager
- 4) Then go to the available tab and click on “Browse...” You can then navigate to where you downloaded your folder and open it (you can

also upload multiple libraries just by highlighting multiple)



5) You should now see the library in the available tab, click on the libraries and press use and your al set.

Y'all should go ahead and install these libraries now:

Adafruit Eagle Library - any product that adafruit sells, find the components in this!

<https://github.com/adafruit/Adafruit-Eagle-Library/zipball/master>

SparkFun library - literally TONS of part footprints its amazing - Code top right corner->Download zip to download all of them at once

<https://github.com/sparkfun/SparkFun-Eagle-Libraries>

Good more specialty chips

<https://www.diymodules.org/eagle>

Another good resource for power supplies in particular for some reason

[https://github.com/DangerousPrototypes/Eagle\\_Part\\_Library](https://github.com/DangerousPrototypes/Eagle_Part_Library)

You only want dp.devices\_v6.lbr, not the regular one (we're using Eagle V6)

If you would like to make your own library, the process is a bit too long to cover here but very useful for when you're using a rarer part. Autodesk themselves has a good tutorial on this:

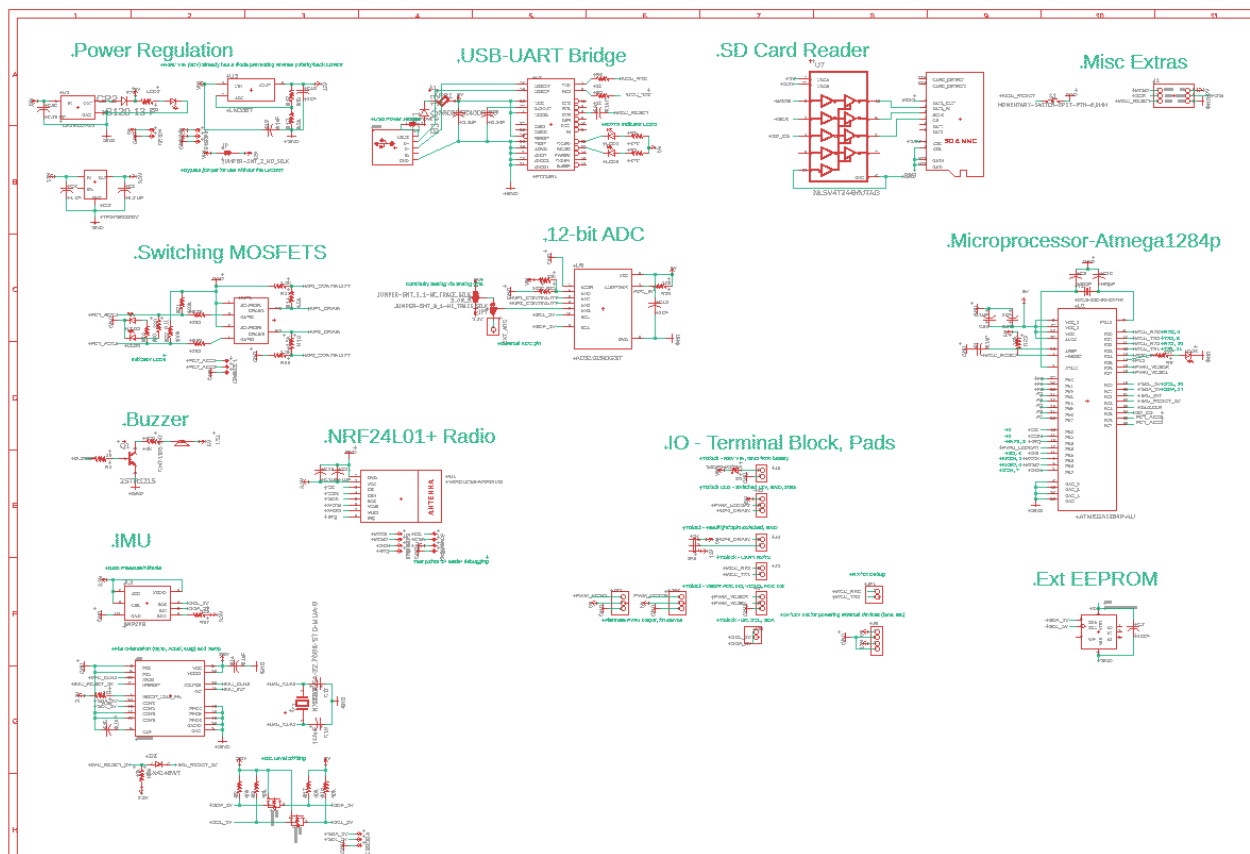
<https://www.autodesk.com/products/eagle/blog/library-basics-part-1-creating-first-package-autodesk-eagle/>

Bonus points if you make a component we'll use today yourself

## Quick note on schematic styling

Remember the frame from the "frames" library we placed on the schematic last time? The purpose of that is that it helps you organize your schematic

into more readable pieces. Here's a picture of a complex board:



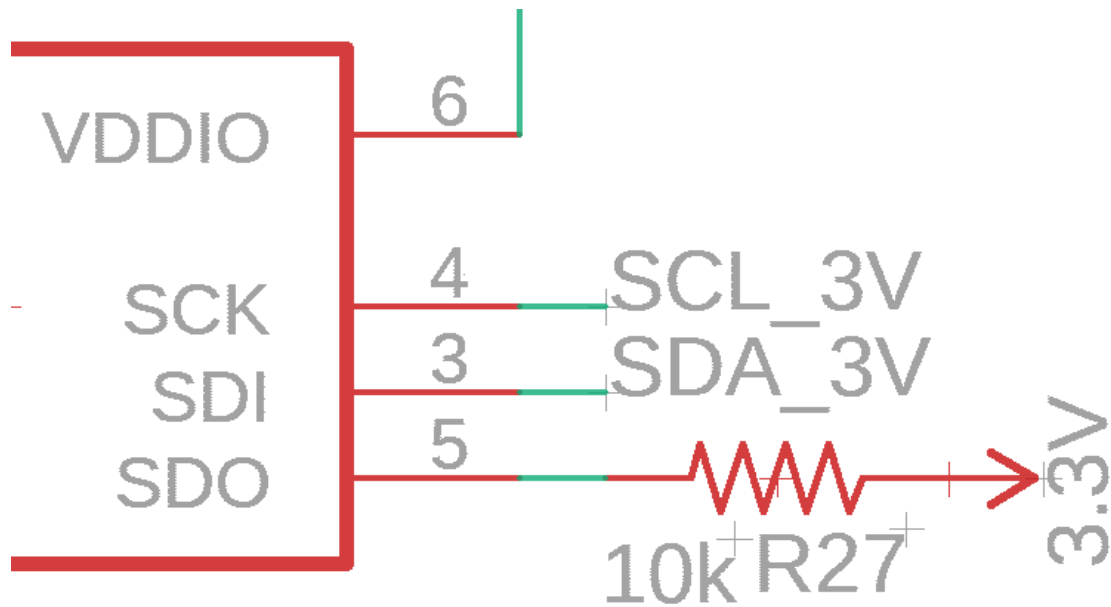
As you can see, there are a lot of different components. However, by grouping them into what each does, I can stay organized and not get lost in the mess

Also, using the text tool in the Document tab to help you stay organized

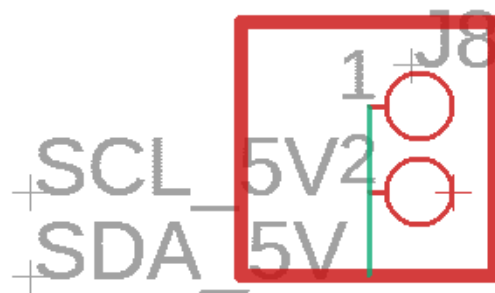


**Question: how to make a connection go all the way across the schematic without it looking like spaghetti?**

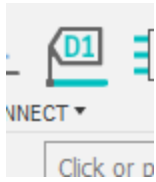
This is where the Label tool comes in handy. Basically, I can label connections across the board instead of directly connecting them.



Here are a few of the connections of one of the chips in the complex design. You can see that instead of being connected to something else, the wire has a grey label



All the way across the board, another label connects the traces. You can use these wherever you want in the schematic to make it look cleaner by using the label tool

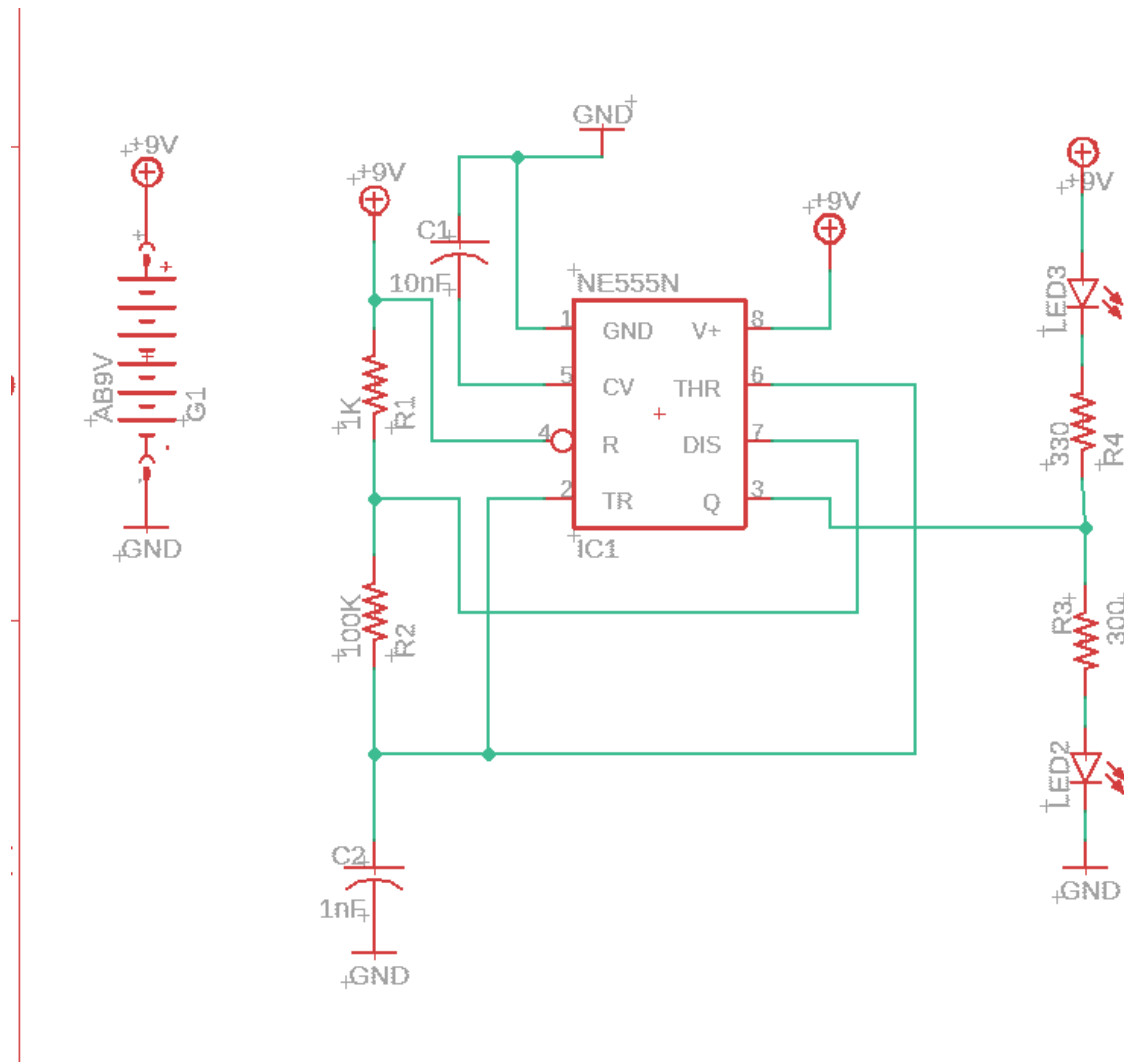


And labelling a blank piece of wire that you've placed. Good to know for future designs

## Part 2: Physical board and layouts!

To start, we're going to be building a more complex circuit than last time. Here's the schematic:





The chip in the center is called the “555 timer”, and it’s responsible for controlling the LEDs you see on a certain time interval.

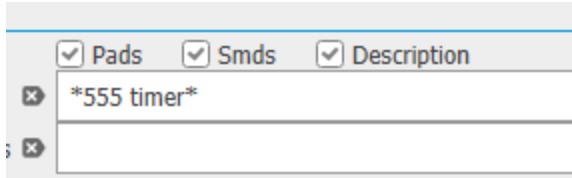
**Your first task is first to create a new schematic and copy this design into it, using what you learned from day 1.**

Hints: The resistors/capacitors are all from the RCL library, like last time.

LED from LED library etc

The things that say 9v can be found in the Supply4 library

When finding the 555 timer IC (and searching for anything, really) wrap it in asterisks to widen the search criteria, like this



Otherwise the search engine excludes too much (it's designed to be quite specific)

Make sure to set the component values (like resistance and capacitance) correctly! (right click-> properties -> value)

Refer to day 1/ask me for help if you need it

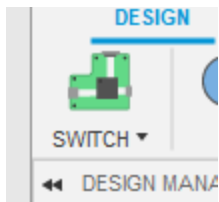
Day 1 link

[https://docs.google.com/document/d/1C0UMrlp4nEwJ-xFcjFp\\_fFkr\\_P2jDH9SwjsYwmBxTNI/edit#](https://docs.google.com/document/d/1C0UMrlp4nEwJ-xFcjFp_fFkr_P2jDH9SwjsYwmBxTNI/edit#)

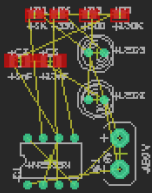
You might be wondering, what's the point? In fact, most stuff is just too difficult to design 100% on your own and you're actually supposed to follow manufacturer recommended circuits in most cases (they give you a reference design or schematic to use). So it's actually quite useful

## Intro to boards

When you're finished with the schematic, go to Switch to board in the top left



You should have a screen that looks like this

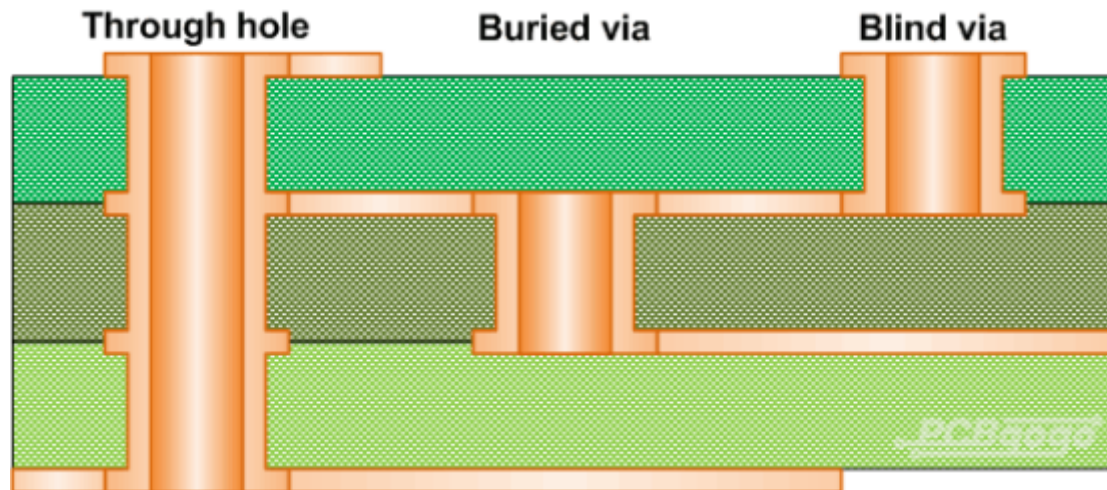


To the left are all your components. You can see a bunch of yellow lines connecting different parts of the circuit. These are called “air wires” because they represent connections that should be made but aren’t yet


# Terminology overview

Ok, here's some important terminology for y'all that will make sense as we go along. Just keep it in mind

**Via** - connection between the two sides or layers of the board that passes a signal through. If this represents the board, the Via is the hole on the left that connects the copper on the bottom and top of the board. Buried and blind vias are not important just yet since we won't be using them (they cost \$\$\$)



Stackup - represents the physical composition of the board. In the photo, the board has 4 layers of copper (3 layers of substrate, or nonconductive material). The copper and substrate alternate layers. We'll be making 2 layer boards, meaning there's only copper on the top and bottoms and a single layer of substrate in the middle

| 2 Layers 1.6mm(0.062inch)Green for 1 Oz   |                    |               |
|---|--------------------|---------------|
| Layer stack up  | Layer              | Thickness(mm) |
|  | Top Solder Mask    | 0.01          |
|  | Top Layer          | 0.035         |
|  | Core               | 1.5           |
|  | Bottom Layer       | 0.035         |
|  | Bottom Solder Mask | 0.01          |

Here's a stackup for the kind of board we'll be making

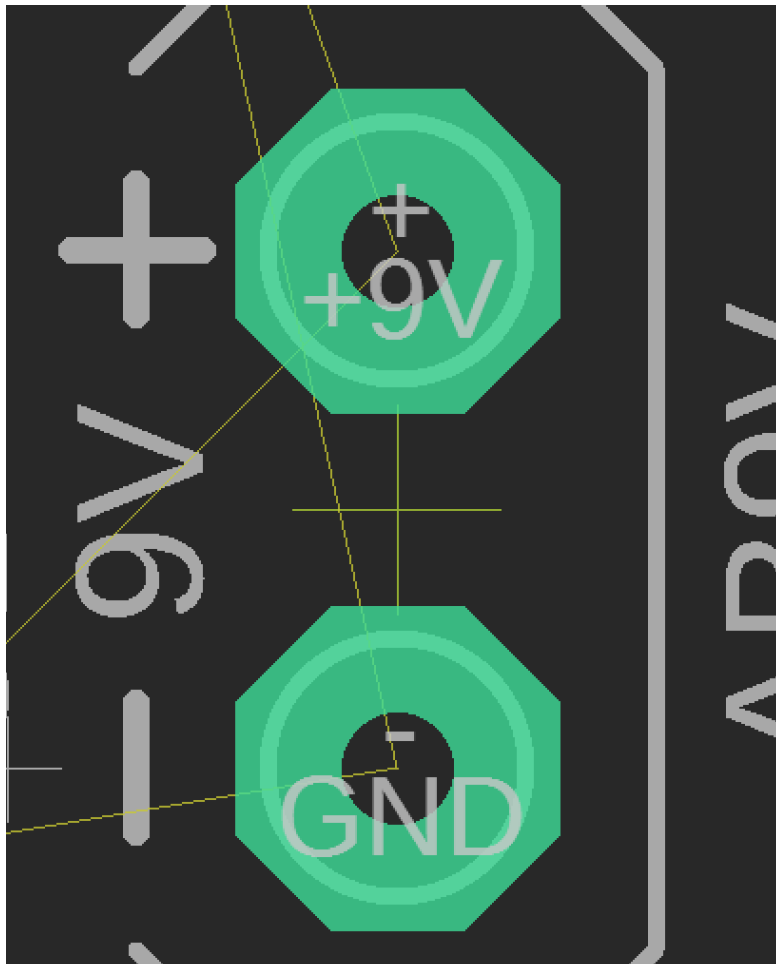
**Solder mask** - Goes over the raw copper traces and protects them from each other + oxidation. Usually tinted green which gives boards their distinctive look

Trace/wire/route - the actual wires on the board. These are WAY smaller than the wires that you're probably used to - they are commonly only

0.005in!! Wide which is crazy. This is why pcbs can be really dense if they're organized right

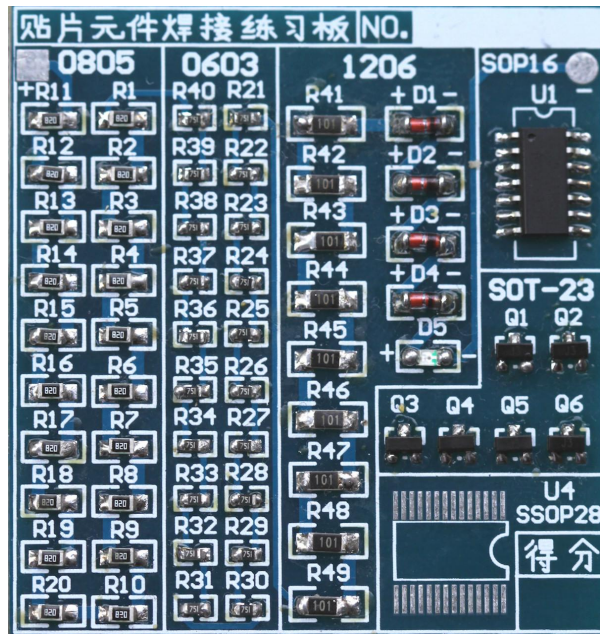
**Silkscreen** - labels for components are printed on the board. These are added by default in Eagle to represent the name of the component to make it easier to identify what goes where while assembling

**Through-hole** - components that go all the way through the board



This battery plug is through-hole

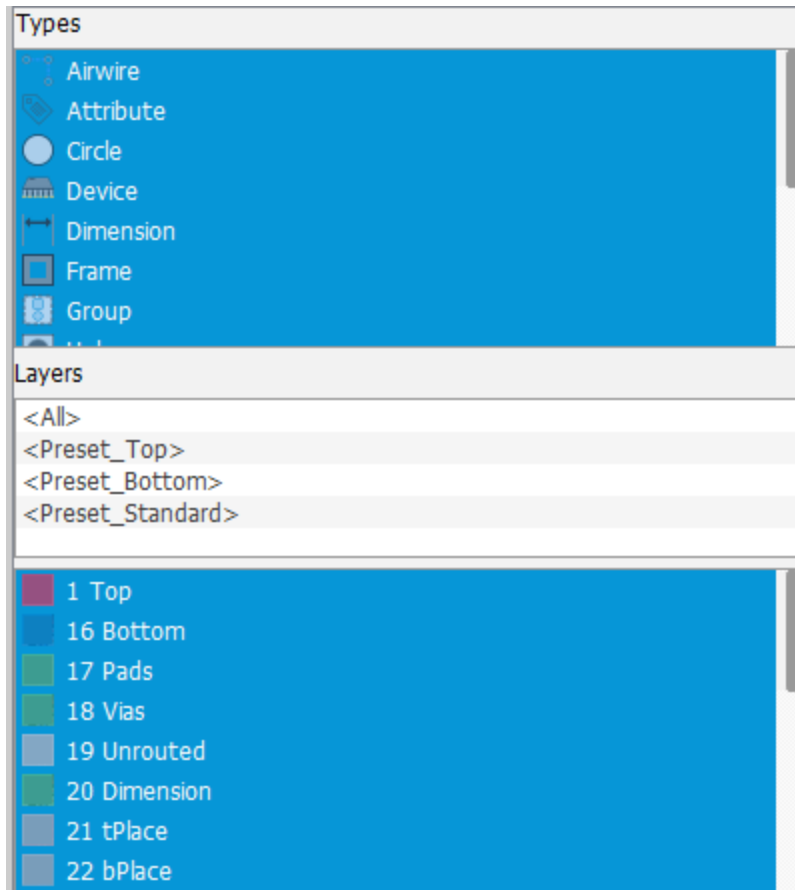
**SMD or surface mount** - components that go on top or bottom of the pcb



## User Interface (UI) Overview

Okay, a quick tour of the UI features

Bottom Right: selection filter



Controls what parts you can select/move around at once. For example, if I'm trying to route a signal I don't want to be moving the parts (known as "devices" around)

Second tab: Design manager

Browser

Filter



View

Devices



Device Sets 3 of 3 shown (0 selected)

Search



## Device Set

<All Devices>  
<Bottom Side Devices>  
<Top Side Devices>

Devices 0 of 0 shown (0 selected)

Search



| Name | Footprint | Value |
|------|-----------|-------|
|------|-----------|-------|

Items 0 of 0 shown (0 selected)

Search



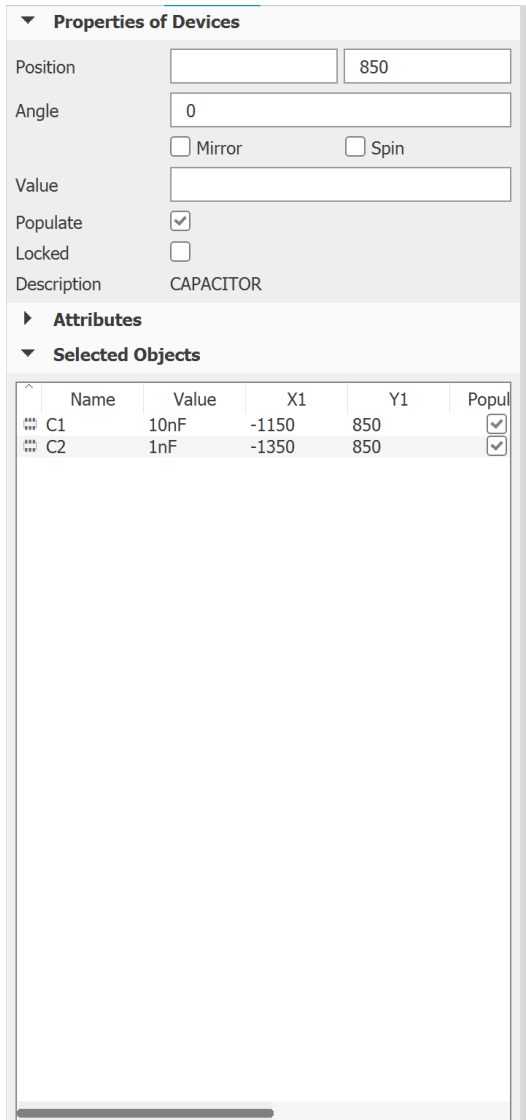
| Type | Name | Signal | Layer |
|------|------|--------|-------|
|------|------|--------|-------|



Lets you search for devices and find good info on them

3rd tab: Inspector

Empty right now, but if you select 2 or more components you get this



**Properties of Devices**

Position:

Angle:

☐ Mirror ☐ Spin

Value:

Populate: ☒

Locked: ☐

Description: CAPACITOR

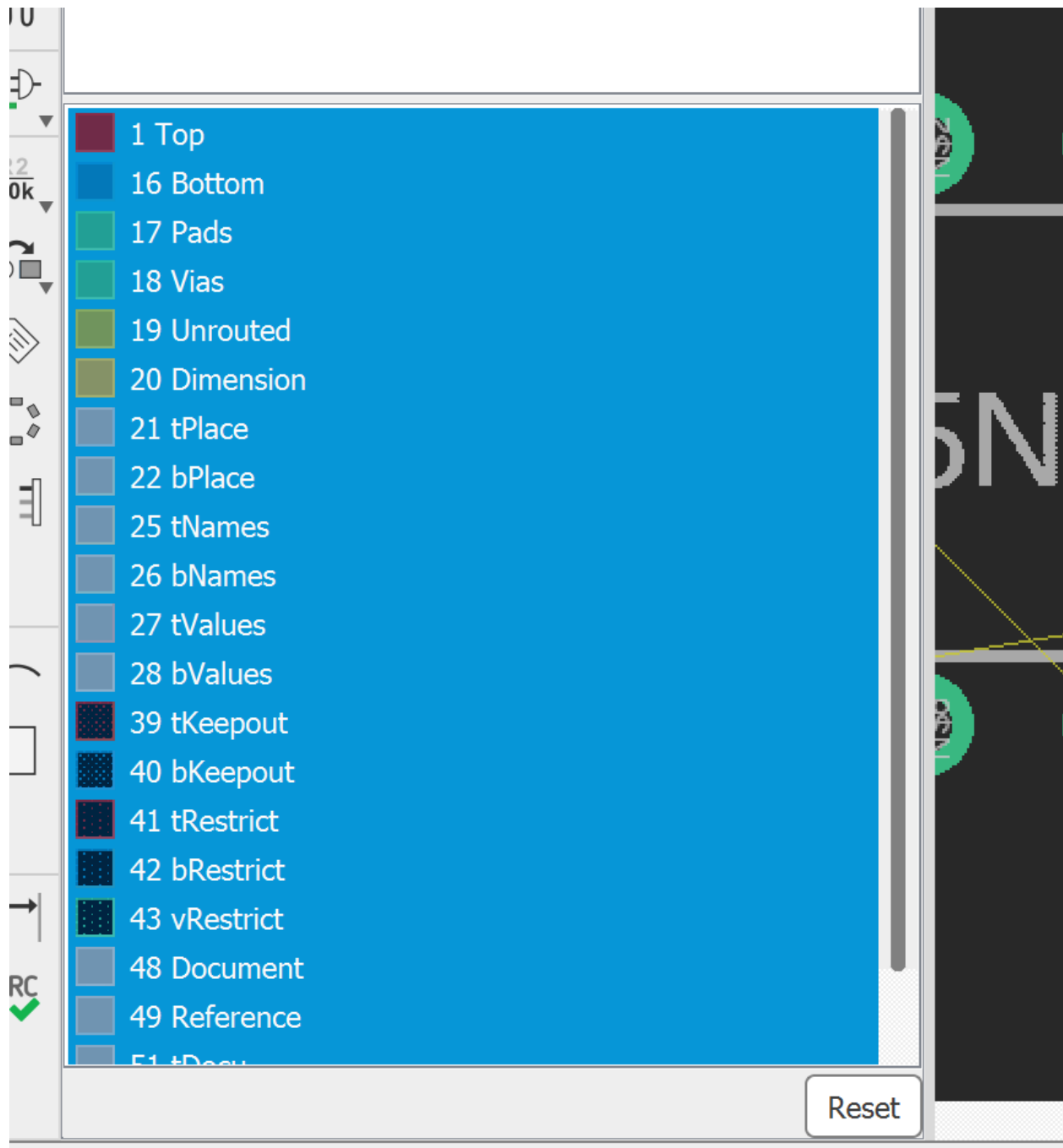
**Attributes**

**Selected Objects**

|     | Name | Value | X1    | Y1  | Popul                               |
|-----|------|-------|-------|-----|-------------------------------------|
| ### | C1   | 10nF  | -1150 | 850 | <input checked="" type="checkbox"/> |
| ### | C2   | 1nF   | -1350 | 850 | <input checked="" type="checkbox"/> |

## Layer Overview

Fusion organizes your layout into layers, where different parts of the information about your board are stored



Quick explanations for each and when to use them:

Top and bottom: controls where the copper goes/the traces

Vias: layer to hold just the vias (where the two sides of the board connect)

Unrouted: holds airwires, will be empty in final version (since all traces are routed)

Dimension: you can set the dimensions of the board here

tPlace/bPlace: Silkscreen text/graphics for the top and bottom of the board

tNames/bNames: Component names

tValues/bValues: component values

tKeepout/bKeepout: component hitbox kinda thing that determine what space they take up (i.e. no other components can go there)

tRestrict/bRestrict: component hitbox thing that determines if a trace can fit there

After 43, they're not as important but you can find a full description here if you're interested: [Boom](#)

We'll be working mainly with layers 1 and 16 (top and bottom)

## Actually placing components yay

The way this goes is in 3 main steps;

Placing components, then routing the wires, then doing a design check

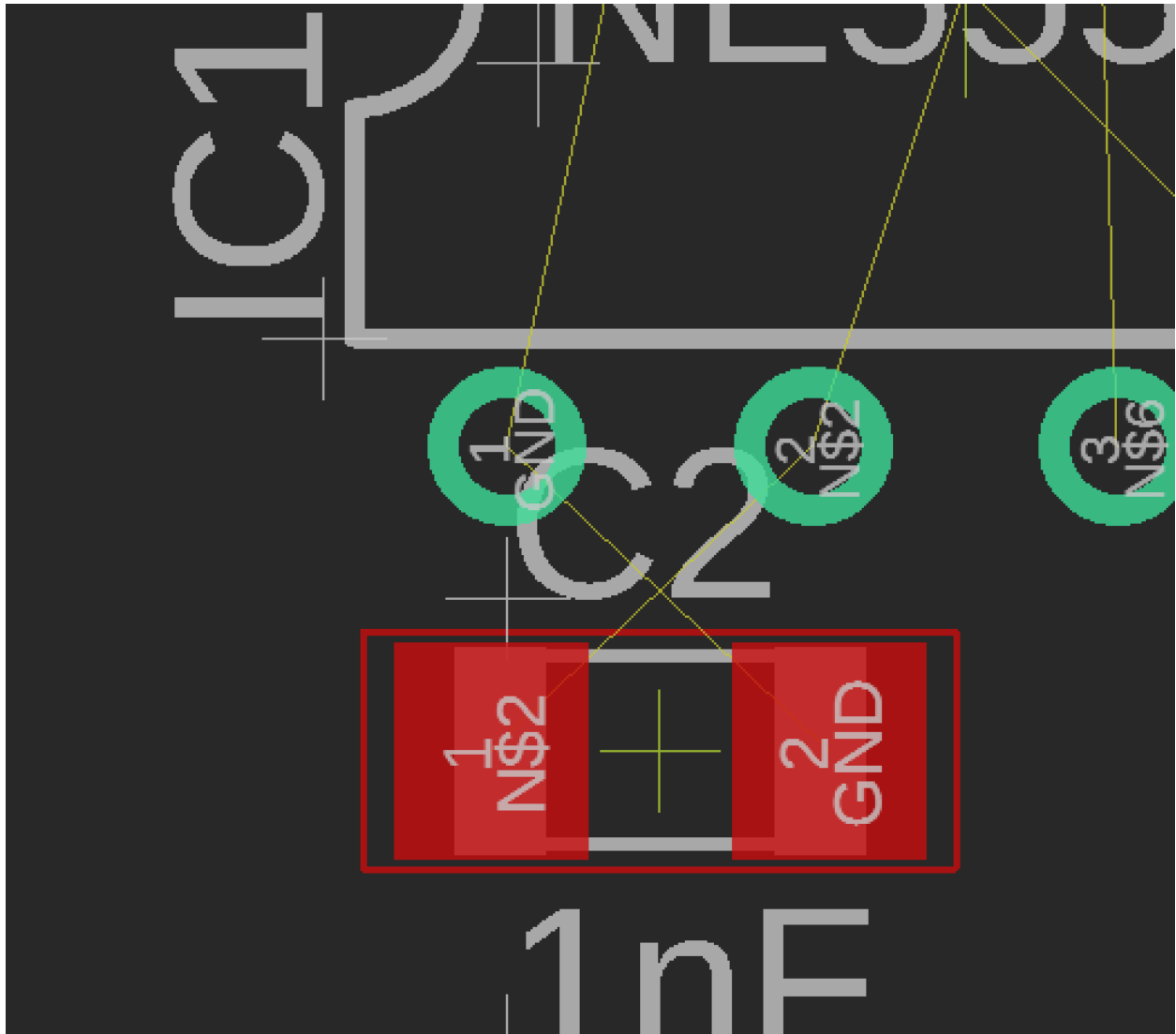
Ok, now that y'all know the terminology we can get to placing components

Basic guidelines: try to fit as much stuff as you can into as little space as you can, while making everything line up.

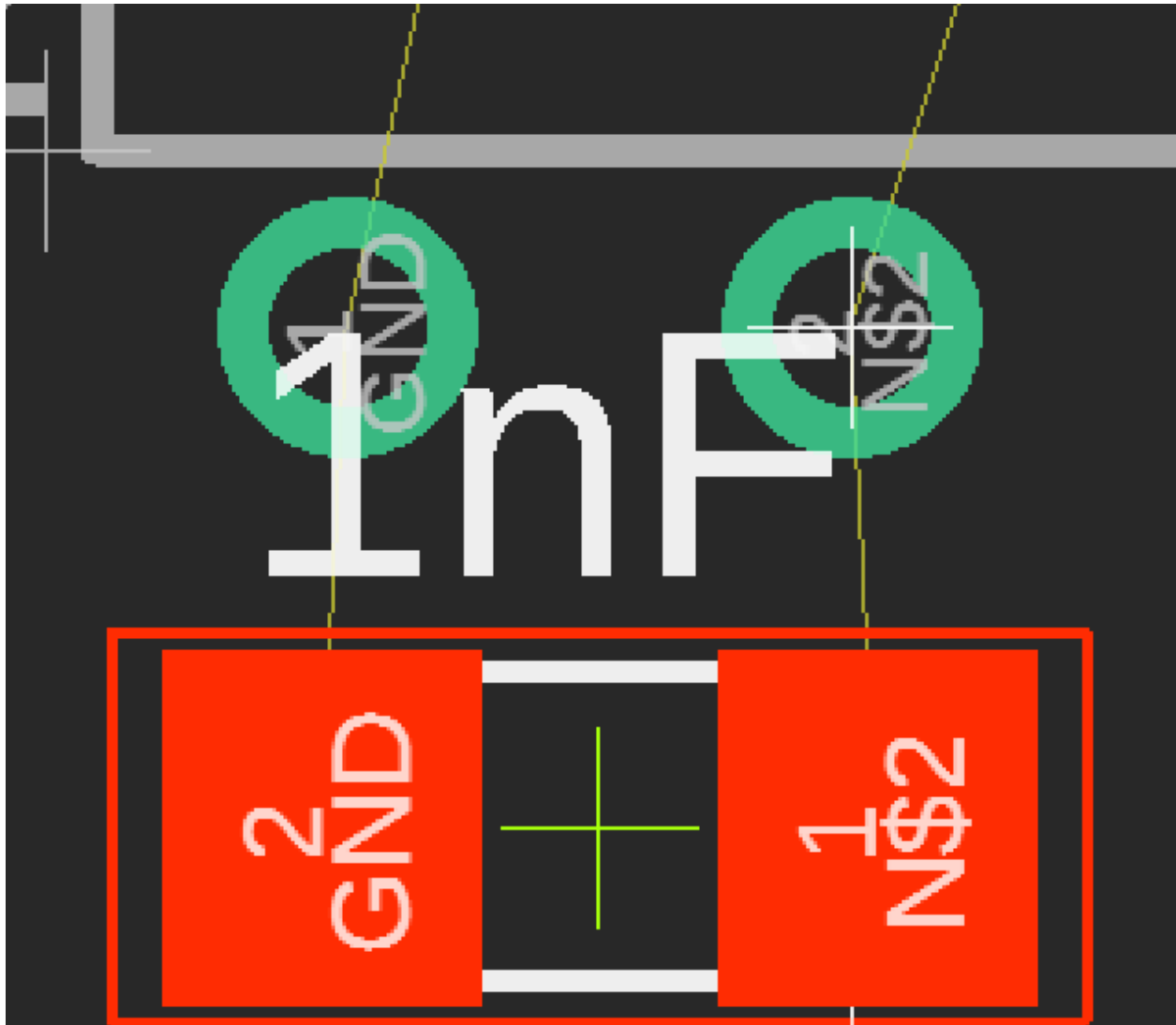
This is more of an art to a science, but basic things to keep in mind:

- Try to group devices that need the same things together. For example, if I have a bunch of sensors that use the I2C bus (i.e. are communicating with the microcontroller directly), I want to place them closer to the MCU to be more efficient

This is an exaggerated example of not efficient placement



As you can see, the wires crisscross. In the schematic, this doesn't matter, but in real life, the wires can't cross or your circuit won't work



Doing this is far more efficient.

## Controls/basic strat

I usually follow this strategy:

Place the most complex things first followed by your things from your rcl library then the rest.

Then use the route manual tool to click on a piece of airwire and connect

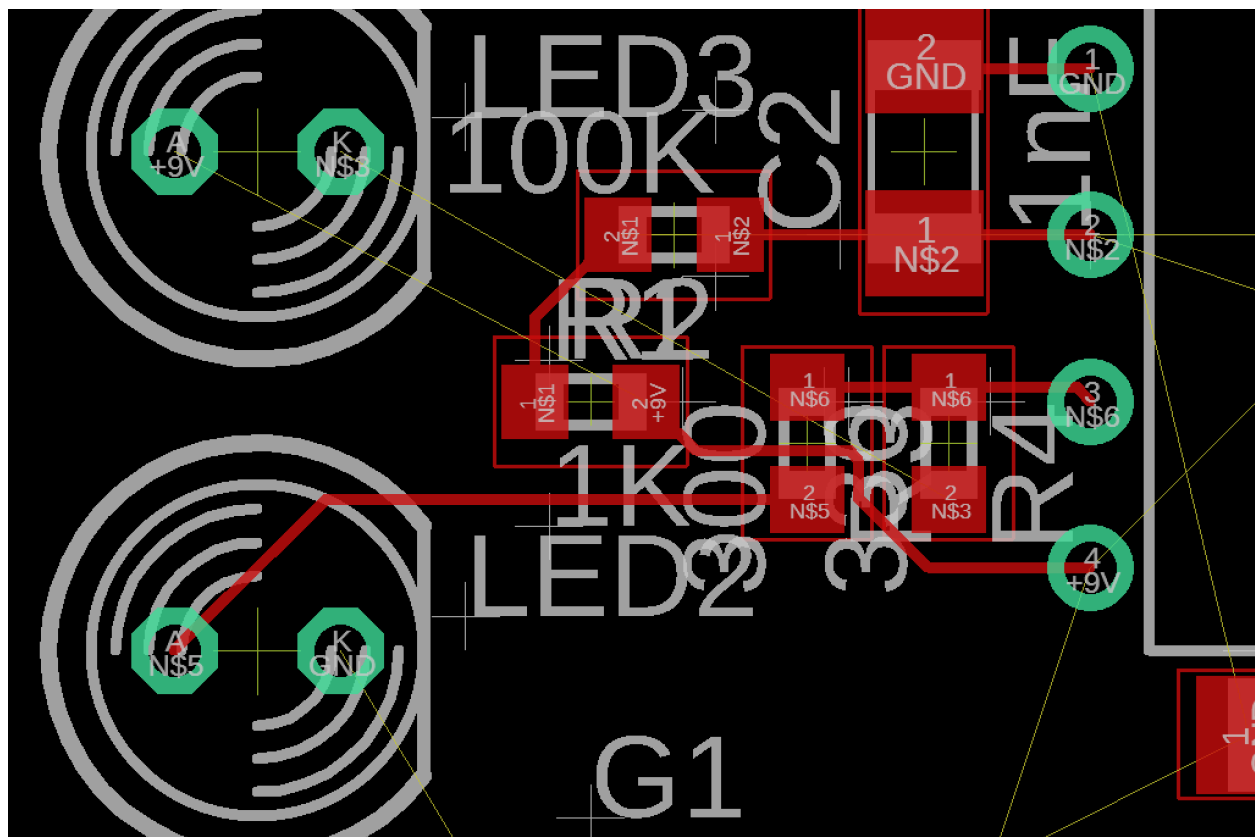


the components.

Or you can try the autorouter. Start by routing a few by hand to get the hang of it.

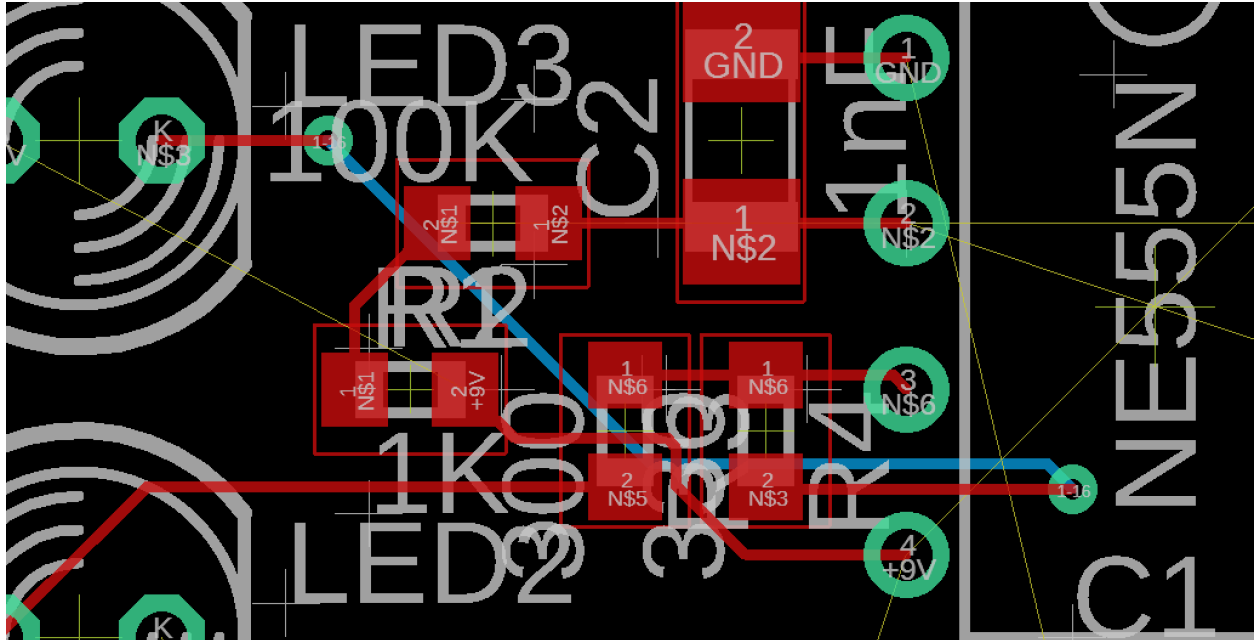
The route airwire tool lets you route connections from one pad to another.

All you have to do is click on two pads, and the connection should be made



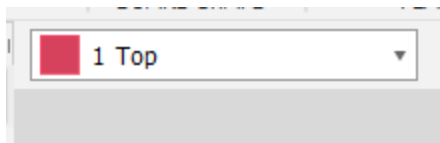
However, in situations like this, I'm trying to go from N\$3 near the chip to N\$3 on the LED. I can't place a signal on the top of the board, indicated in the red, because it's trapped. So to fix this, I need to use a via.

A via lets you move a signal to the bottom of the board like so



To use a via, drag your connection out to where you want the via to be and click the middle mouse button. It'll create one for you and transition to the bottom layer. When you want to come back to the top layer, click the middle mouse button again

If you're having trouble selecting your components/the trace isn't on the bottom, make sure the layer is set correctly



In the top right corner, this is the layer you're actively working on

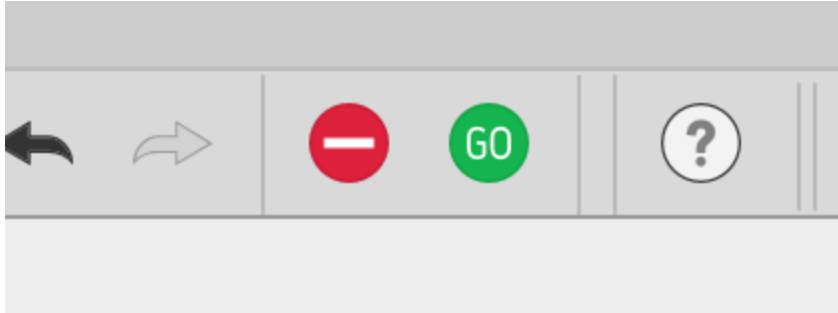
## There's gotta be a faster way, this takes so long! Aka autorouter time bb

Manual routing can honestly be really tricky and it's very time consuming for larger circuits, which can sometimes need hundreds of individual connections. For this reason, the Autorouter exists, which speeds up the process a lot

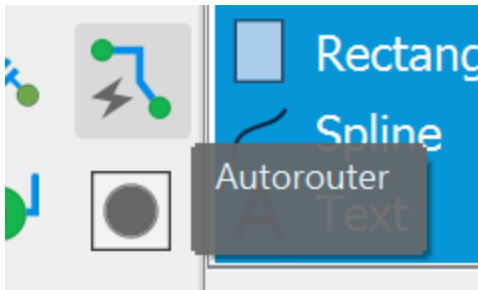
To start, we're going to get rid of all your work so far :(

Click the ripup tool (can use this to remove individual traces or vias by clicking on them)

To remove all, click the green go button on the top

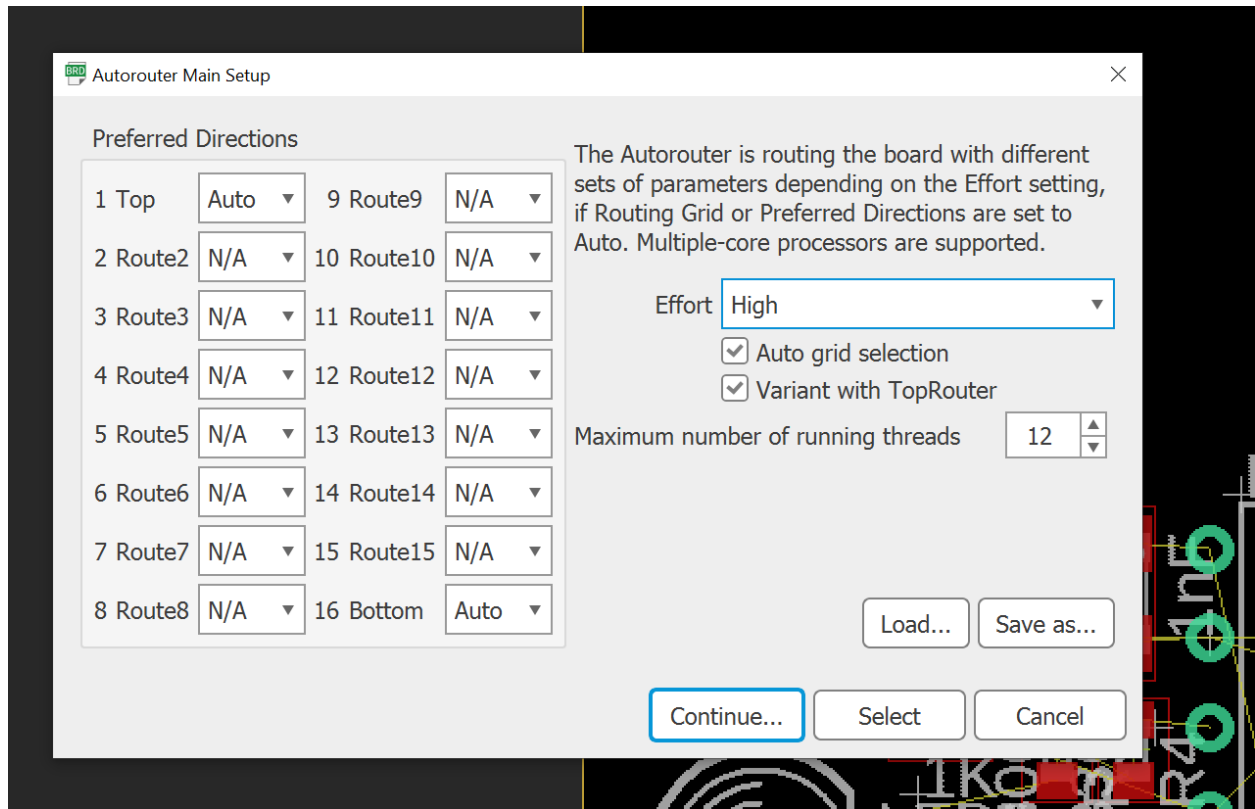


Then, click on the autorouter



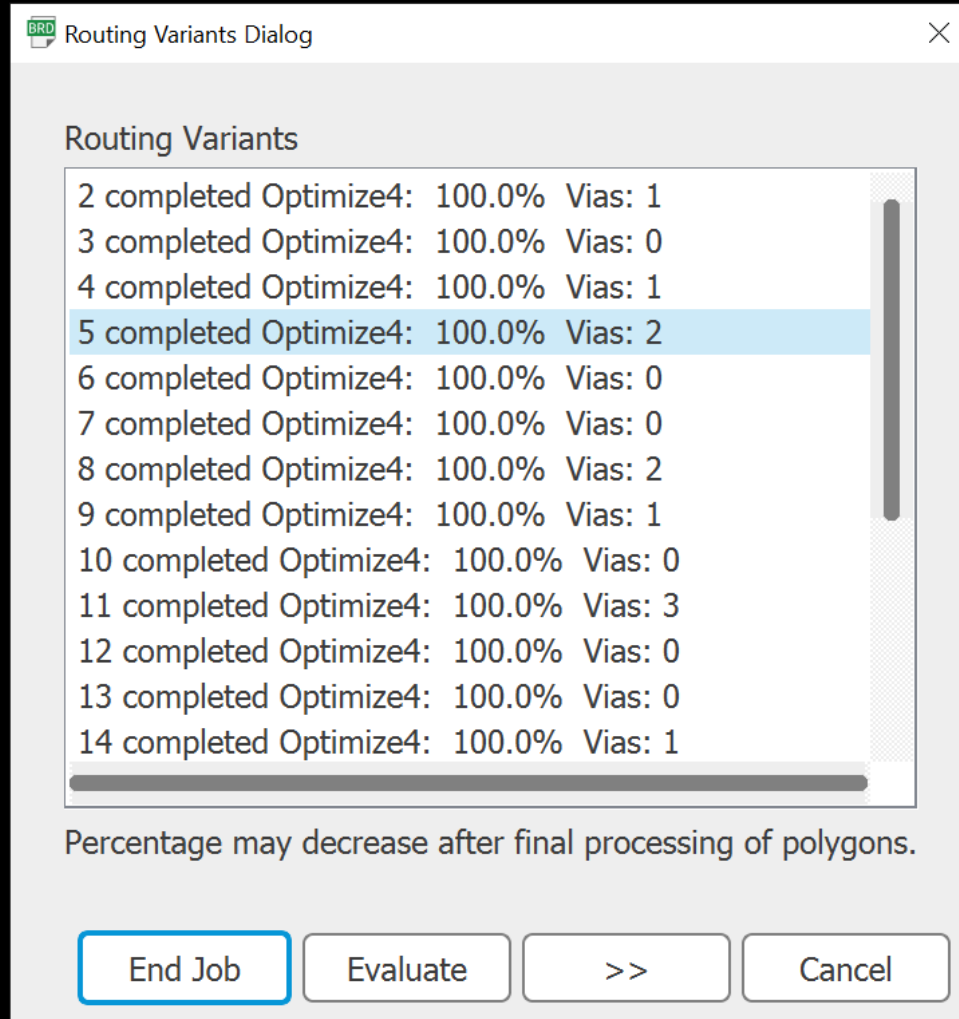
Configure your autorouter as shown



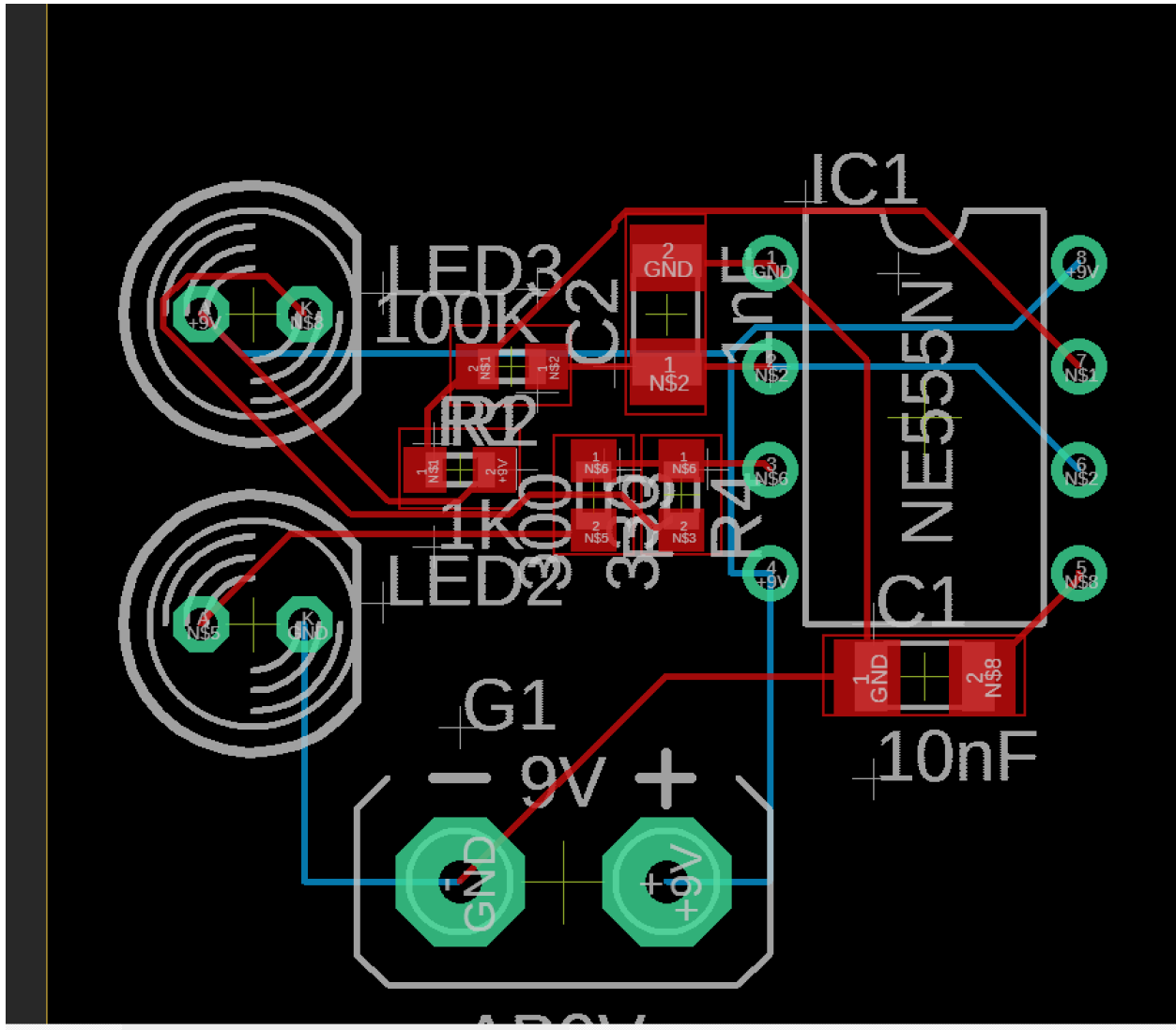


Make sure to set all layers other than top and bottom to N/A (because we're only doing 2 layer board, remember) and set effort to high  
After you're run the autorouter, it gives you a list of options to choose from

*Note: copy all the settings here except for the max number of running threads, my PC has a high core count and you probably won't have 12. Just set it to the max that your computer allows*



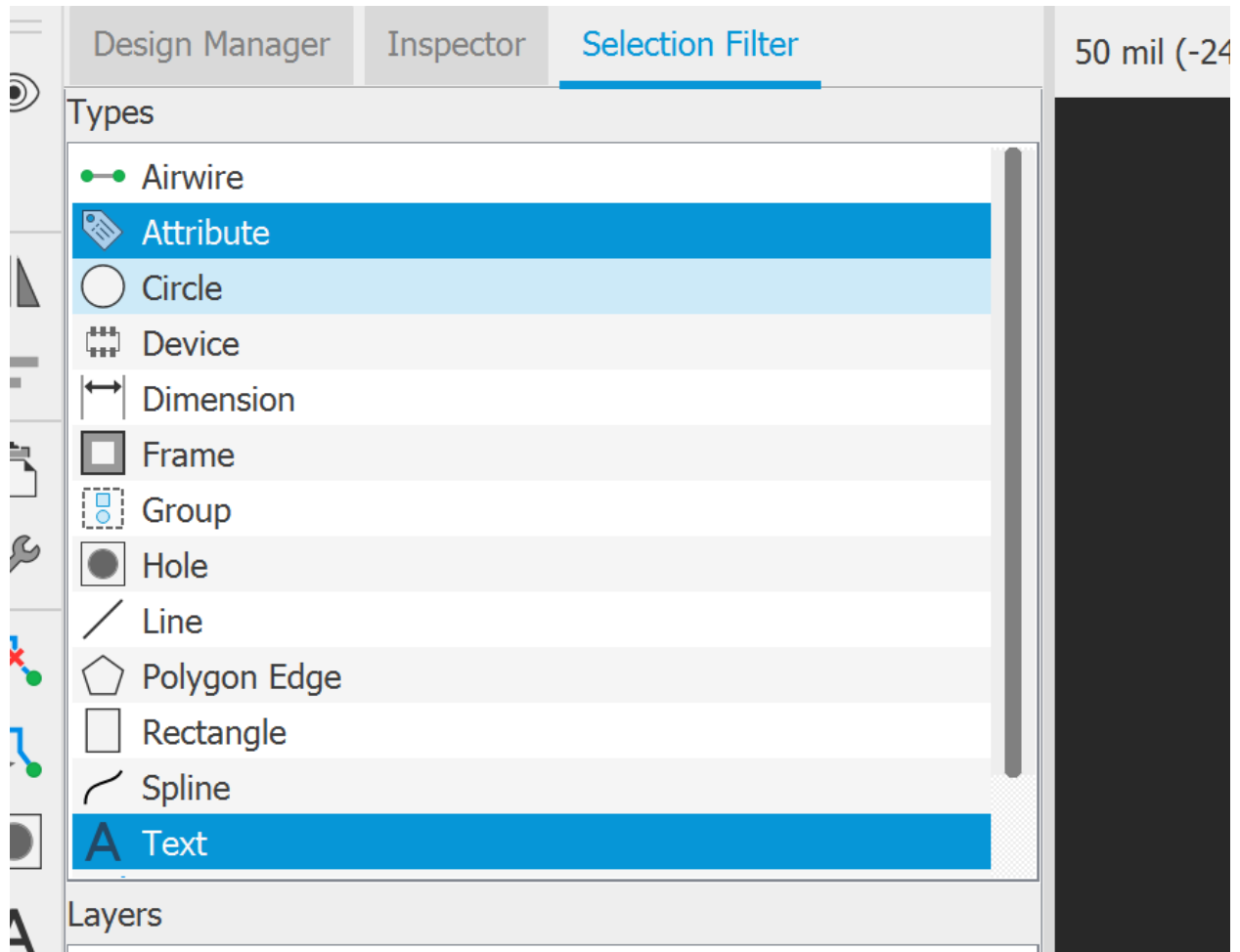
In this case, the circuit is very simple, so it's able to route all of the connections itself without your input. Most of the time tho, it only gets 70-80% of them and you have to do the rest manually



Generally you want to select the autorouter option that has the highest % finished w/ the least number of vias, I had an option that had 0 vias and 100% finished which is ideal. You may have to route some connections manually tho

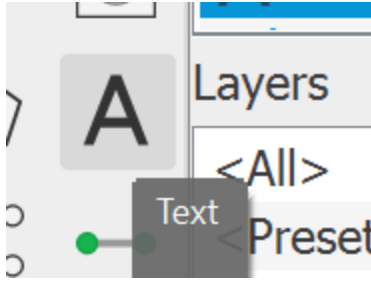
## Silkscreen

The last stage of a project for me is always dealing with the silkscreen, which is the text on the top of the board

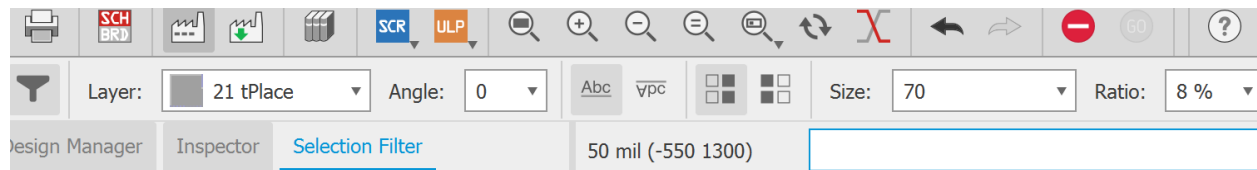


Ensure you've set your selection filter to just text and attributes so that you don't mess w the newly created wiring or devices on the board  
Then, you can drag the text and attributes (value, name etc) around until they don't overlap



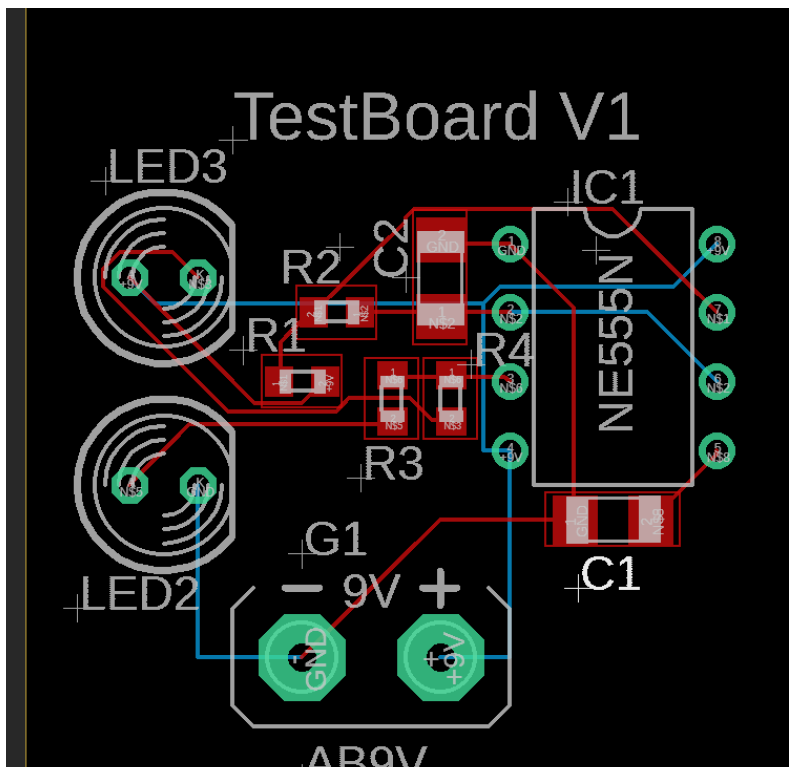


Enter your text and click ok



**Make sure your text is on layer 21 tPlace, NOT layer 1 which it'll try to go onto by default. It won't appear as text if you put it on layer 1.** Change this by going to the dropdown pictured in the top left of the above screenshot

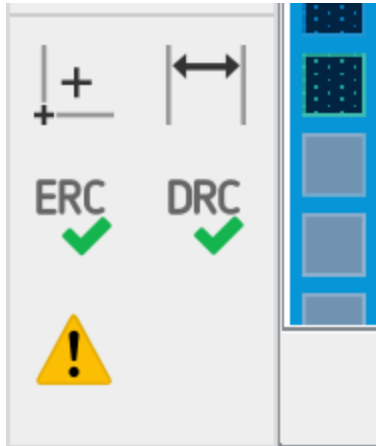
Then, set the text size to whatever you want (I chose 70mil) and place it!



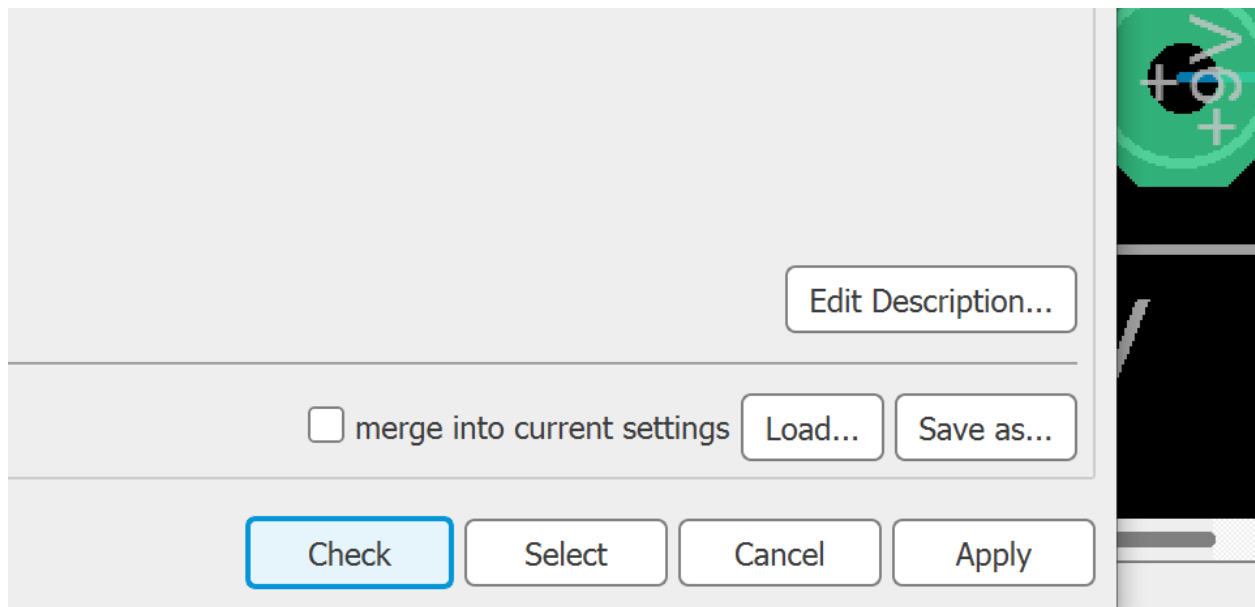
# Wrapping up: DRC Checking

The final step is to have Eagle run an automated test on your board to try to check things like clearance etc

The DRC stands for Design Rule Checker, it performs a number of checks to make sure your board will work



You can find it down in the bottom left corner



Click Check, ensure there are no errors and correct them if there are  
Errors are things like devices overlapping, wires touching etc

Note you may have to set your selection filter to include devices if you want to move things around/you might have to rip up traces

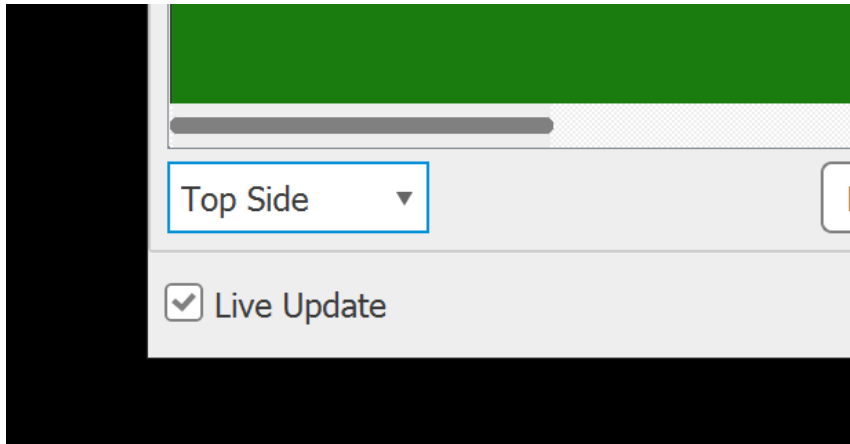
We'll mess with this more next time, these settings don't reflect reality just yet (i.e. different manufacturers have different standards that you need to input into this)

Alright, now that you're done, go to the manufacturing tab in the top right to get a nice-looking preview image

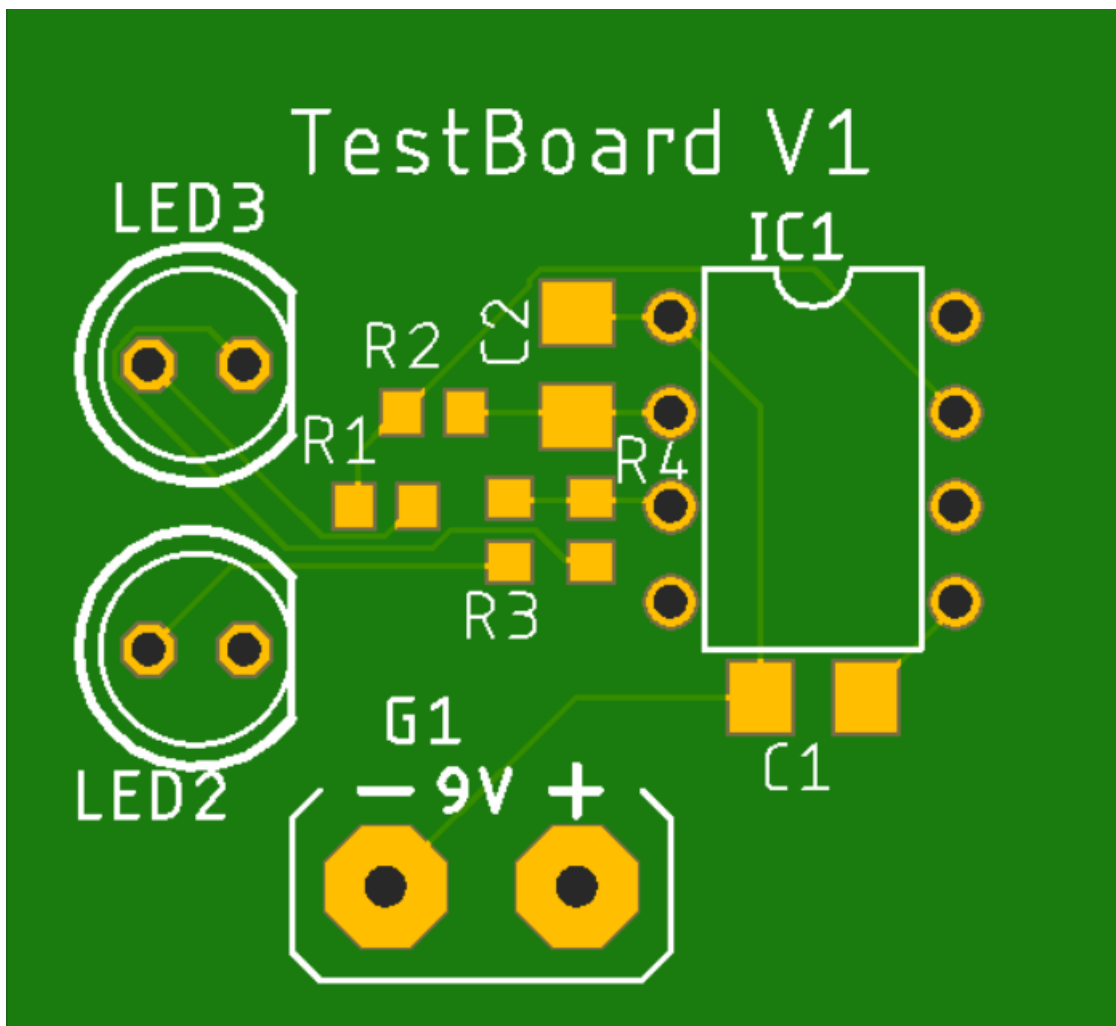


Open it and set the dropdown to "top side"





Then, zoom in on the final image and take a screenshot (or go to the export image option and crop it after exporting)



There should be no silkscreen issues and the traces should look pretty clean. The pictured board is far from perfect but is a decent example of

silkscreen placement and component placement which makes the traces connect without any vias (i.e. all my connections can fit on one side)

Congrats, you've just finished your first layout!

Post an image of the screenshotted board in the slack thread

If you made it this far thanks for following along :)