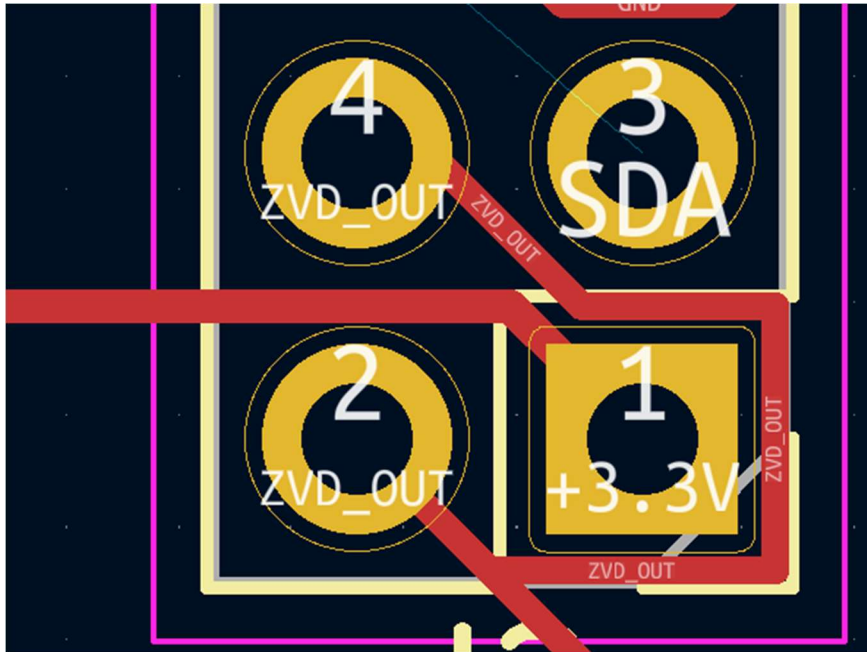


Jargon Definitions and FAQ

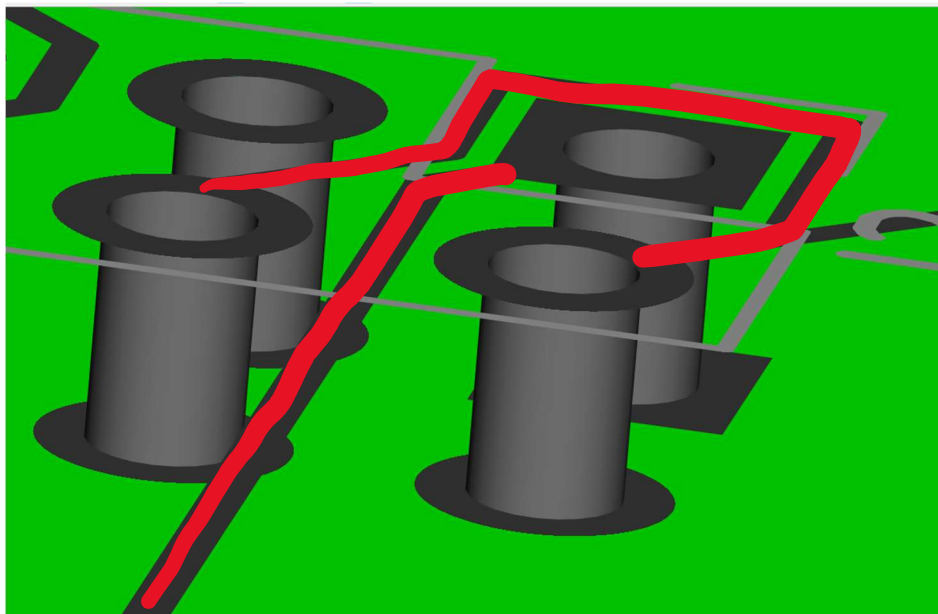
2.679 Electronics for Mechanical Systems II, Spring 2023

Bottom Copper

Printed circuit boards can have more than 1 layer! Most of the things are on the top (red), but you can easily go from the top to the bottom layer to avoid crossovers. For instance, take this layout.

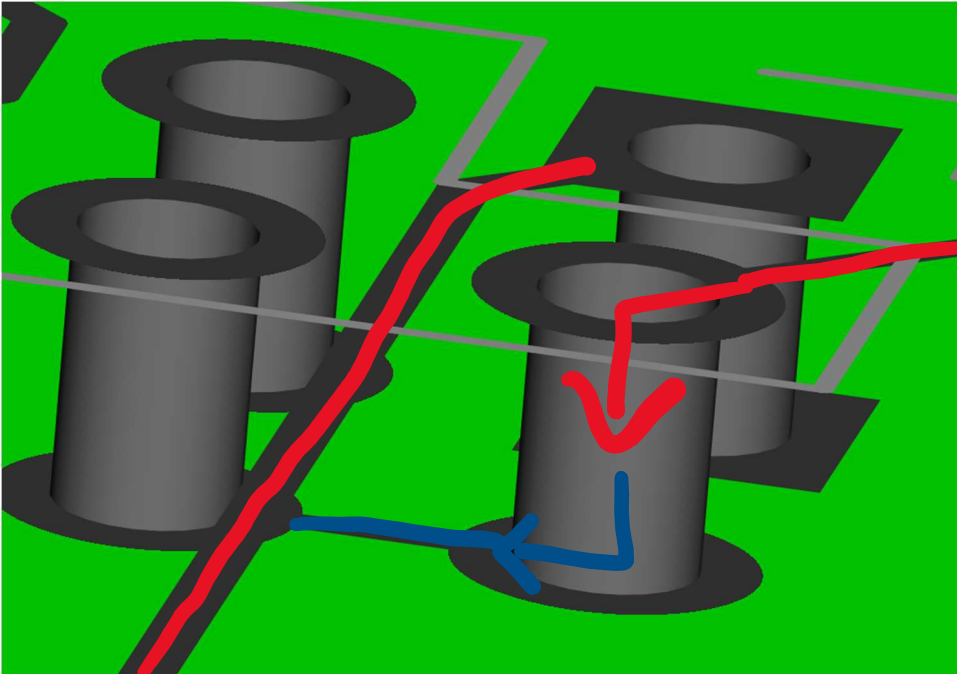
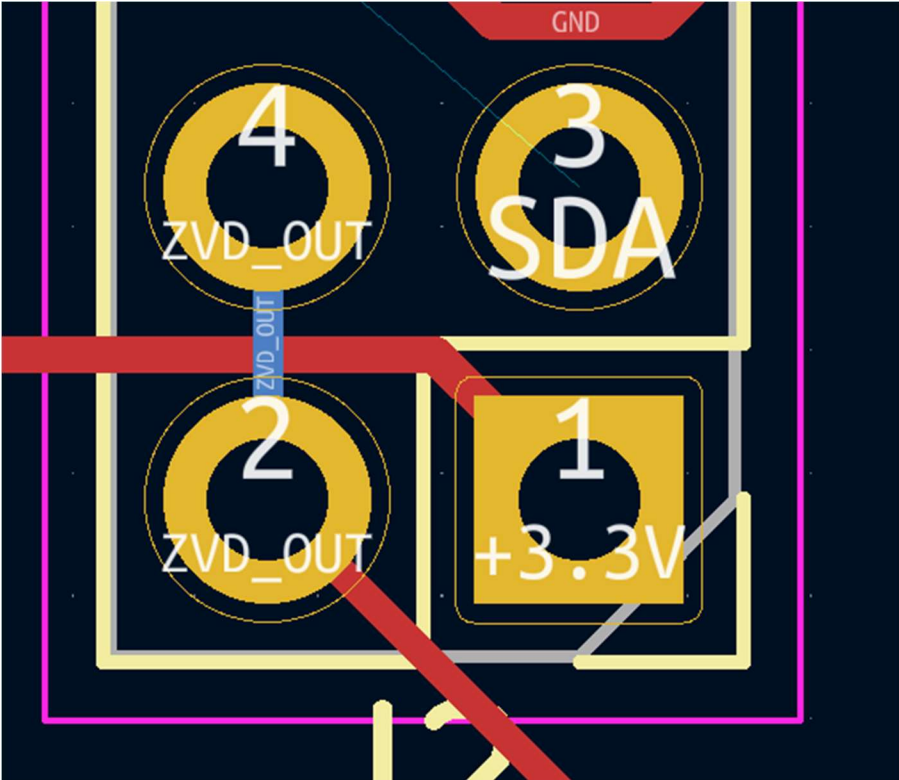


ZVD_OUT needs to go around in order to avoid the 3.3V trace.



Here's how it looks like if the board only showed the copper.

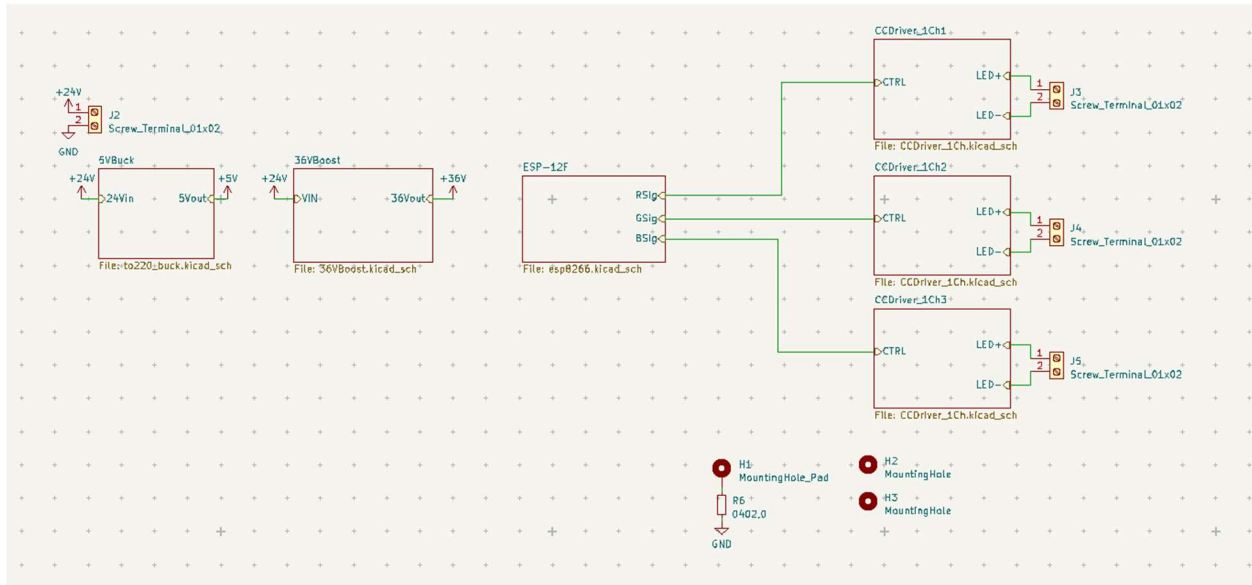
If I were to instead move the ZVD_OUT trace to the bottom layer (blue), I don't need to go around anymore. This is done by selecting the layer on the right hand side.



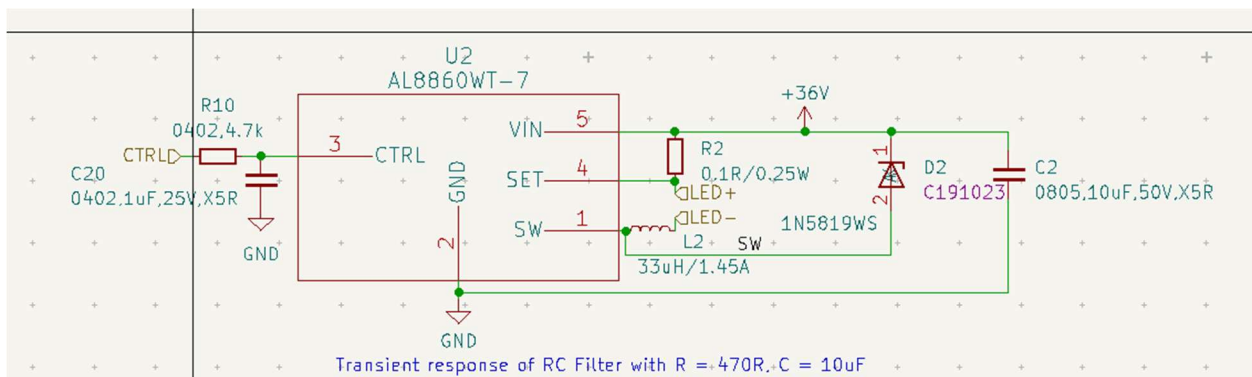
In 3D view, I've tunneled under the 3.3V trace. It's much neater and it costs the board maker nothing extra to do this. Tunneling between layers can be done using a plated through-hole (shown) or a via.

Component grouping

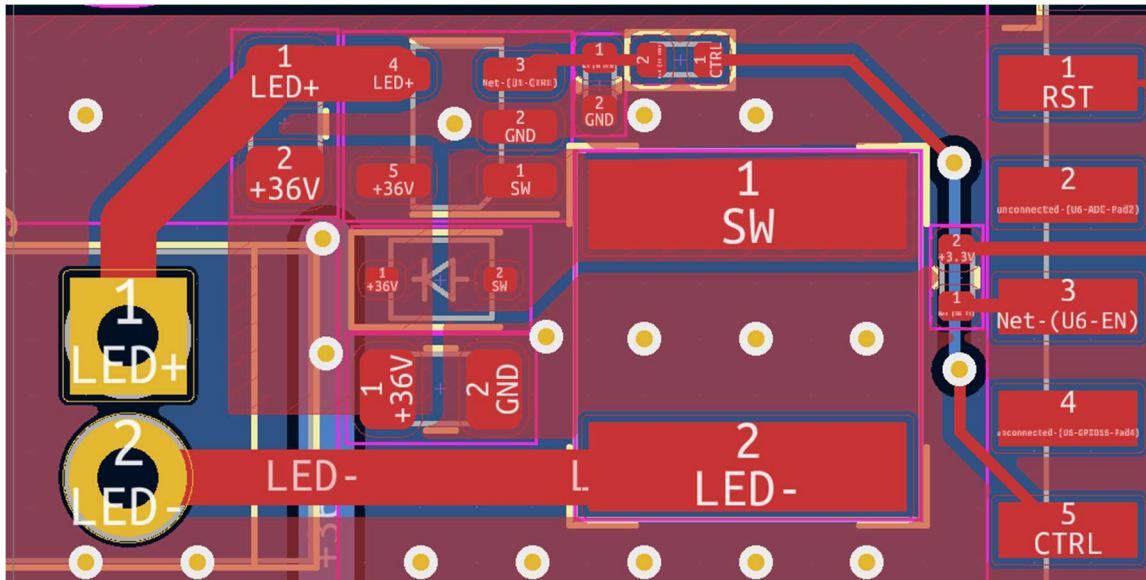
As you get more complex designs, it's more important to group components by function. Typically, there will be many nets between components inside a group, and a few traces to other groups. Take the schematic shown. Only 1 trace needs to be routed between the ESP and the CC LED Driver. One strategy for routing the board may be to optimize the routing of the CC LED driver first, and then connect it with the rest of the system. This reduces the traces that need to be optimized because I'm only concerned with a smaller section of the board. Once that section is finalized, I can then run the control signal between the sections.



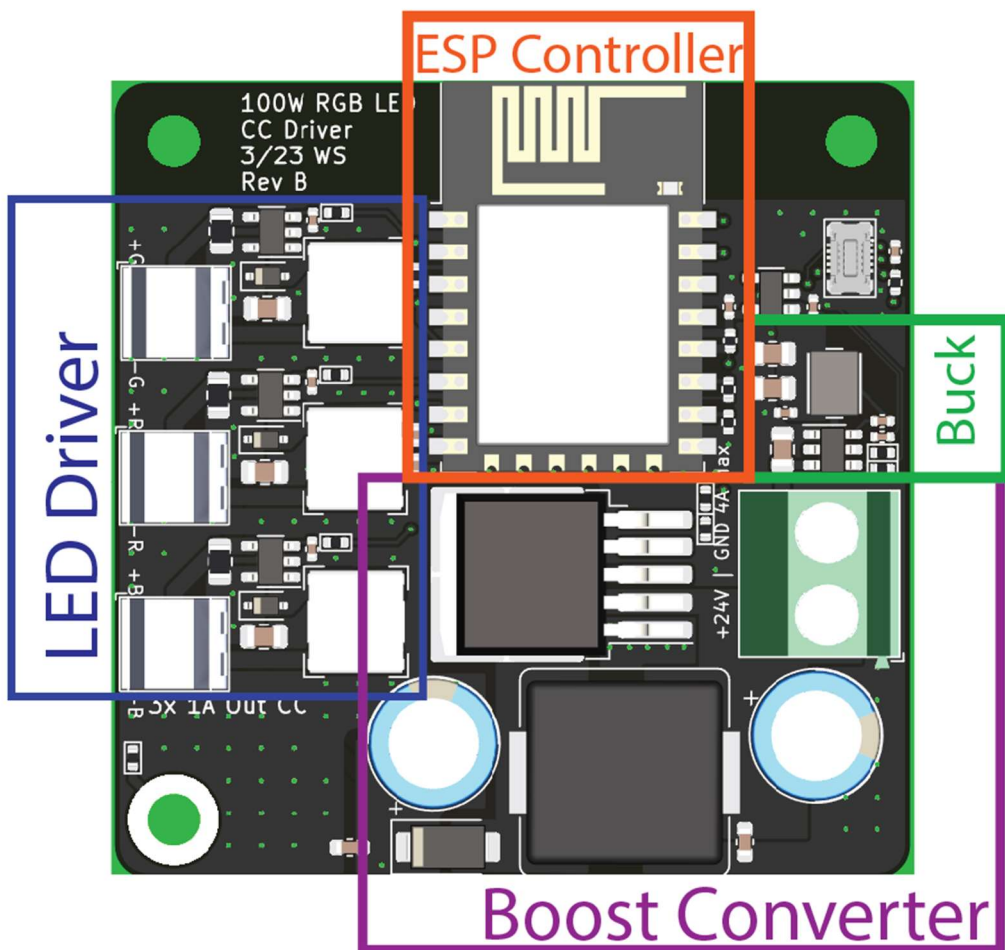
Pictured: Schematic of the LED driver board overview. Each section is split off into its own schematic, and very few connections go between sections. The “CCDriver_1Ch1” is an abstraction—it represents another schematic page, but it can be viewed as a component where a signal is input and LED power is output.



Pictured: Schematic of the CC LED driver. There are many nets between the components, but it only needs a control signal from the ESP. In this schematic page, the driver is de-abstracted and the circuitry that dictates how a control signal is converted to LED Power is shown.




Pictured: grouped layout of the CC LED Driver, with a control signal being routed on the right.



Pictured: board layout with different sections for the buck converter, boost converter, ESP microcontroller, and LED driver. Components for each section are grouped together to simplify routing.


DRC (Design Rule Checker)

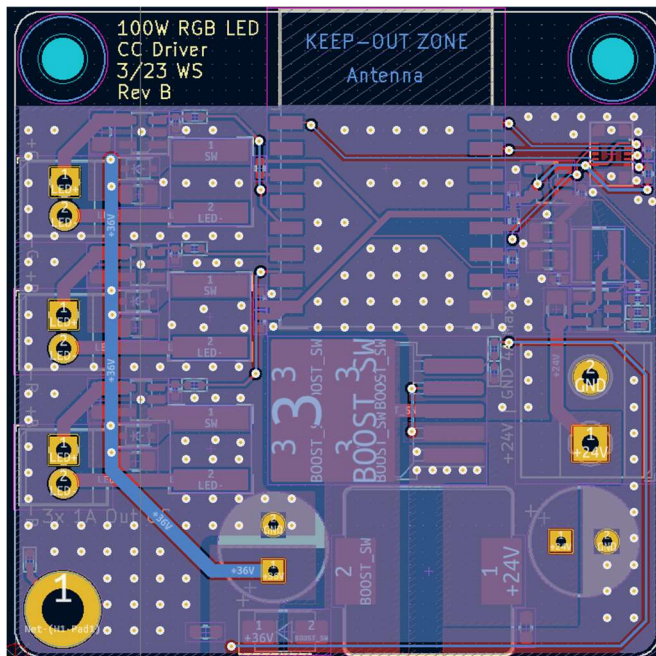
It's probably one of the most important tools you have in a PCB design software! It gives you a pretty good check on whether you've done everything correctly. It does need to be set up correctly, however. See: **Minimum trace width and clearance**. Once it's set up, you can run it by going to the top and

clicking  (Show the DRC window). Run it (repour copper if it shows up), and correct all the errors within reason. It also has a tab for unconnected traces, which is good for catching any that may slip by. If something is intentionally breaking DRC and you think it's fine, right click it and add it to an exception. You want something to show up as 0 DRC errors so someone else knows it has no glaring issues (at least, in the software's eyes).

Ground Pour

Currents that flow into an object must then flow out—otherwise, charge accumulates and you may be violating the laws of the universe. Since almost everything will draw a current, almost everything will also return a current to ground. If your ground is not well defined (ie you have long thin traces between 2 points defined as “ground”, then they may not be the same ground. It's equivalent to putting a resistor between every component and ground. If you're doing signaling or power delivery, the difference in ground potentials is especially important. This is why many boards will include a ground pour. The large amount of copper helps the return current go back to ground, and every component on the board will have the same ground potential. One way to do this is to pour a ground pour on the bottom layer.

Click  (Add a filled zone, right side of screen), select ground as the net, change thermal reliefs to solid, and draw the shape of the copper you want. Be sure to re-pour the copper before saving the board. It automatically keeps itself out of other traces.



Pictured: A ground pour on the bottom layer. Vias are used to connect grounded components on the top layer to the ground pour on the bottom layer.

Ground Return

You may have made a thick trace for voltage going into something—that’s great! But the same current also needs to leave that thing. This means the current that returns (called return current) also needs to be carried on a thick trace—typically the same size as the voltage input trace, since they should be carrying the same current. The return current is typically on the ground trace. See **Ground Pour** for one solution to ground routing.

Import Footprint



Top: Create; delete and edit footprints

File -> New Library -> Project Scope -> Save this inside the folder of your design project. Remember the name of the library.



Then go:

File -> Import footprint -> Select the .pretty file for your footprint. The footprint should show up once you import. Check to make sure it looks right.

Save -> Select the library you named -> save. You can exit now.

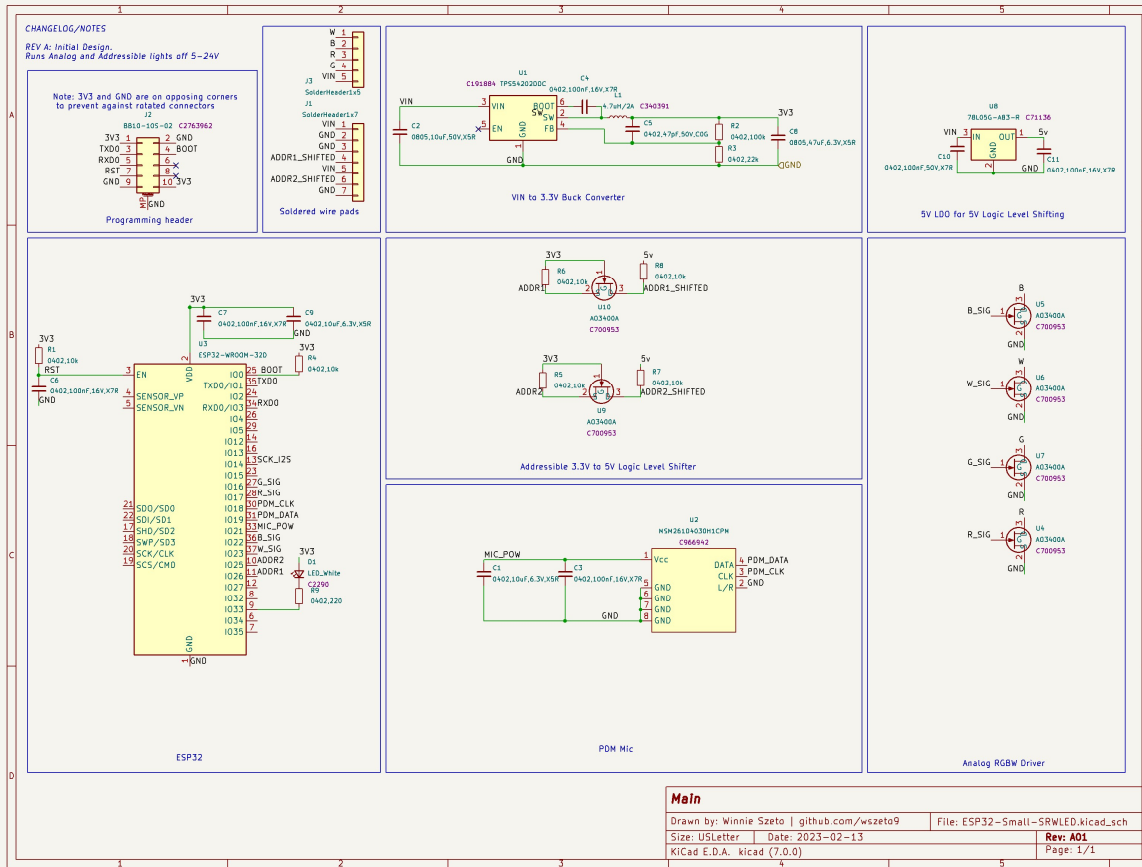
Minimum trace width and clearance

This setting is in File -> Board setup -> Constraints. The minimum size should be 0.127mm (or larger) for both—at least for this project at this fab. If you have a number that is smaller than this, then the software thinks that you can have 2 different traces touching. Effectively, the software will not catch errors where two things are shorting. Other settings exist for distances between two components, you should check those too. See: <https://jlcpcb.com/capabilities/pcb-capabilities>

Copper			
	Minimum clearance:	<input type="text" value="0.127"/>	mm
	Minimum track width:	<input type="text" value="0.127"/>	mm

Schematic Organization

Let's say you were to open someone else's schematic you've never seen before. If the functions of each part aren't labeled, it may be hard to know what each thing does. I suggest boxing stuff by function so someone else knows your design intent—especially with headers and logic shifters.



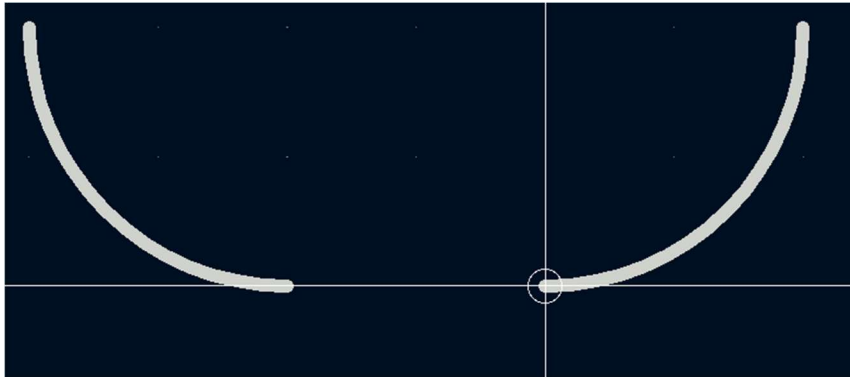
Square Corners

Remember in 2.670 or 2.007, where you have to de-burr edges so they're not sharp? Same issue here. The fabs are great at making what you ask for—edges included. Square corners are therefore quite sharp, and they can hurt if you grab the edge. Fillets can help with this issue.

First, set your grid size to be something large like 1mm – 5mm. There's a dropdown called "Grid size" on the top.

Start by drawing the corner arc on the edge cuts layer. It's done by selecting the center, then start and end point. Be sure it's on the grid and that the arc is 90 deg. Repeat for the other corners. If you choose to copy paste this for the 4 corners, be sure you move it by the center point, not the edge points. KiCAD is very picky about arcs.

Choose the line tool, and ensure that the line starts and ends at a point on the arc. Again, KiCAD is picky. Note the circle on the point indicating it's connected. Repeat until your board is enclosed.



Press Alt + 3 or View -> 3D Viewer and make sure the outline looks correct. If it pops up an error about malformed board edge, a point's probably not connected.

Square Traces

Honestly, 99% of the time, the bad attributes of square traces are purely from the judgement of fellow electrical engineers regarding aesthetics with no practical downside. In the 1% of the time, they matter if you are trying to push the speed of a signal into the high MHz range. At that point, you should probably take an RF class (6.013) to learn more about the black magic of RF. See:

<https://hackaday.com/2019/06/26/whats-the-deal-with-square-traces-on-pcbs/>

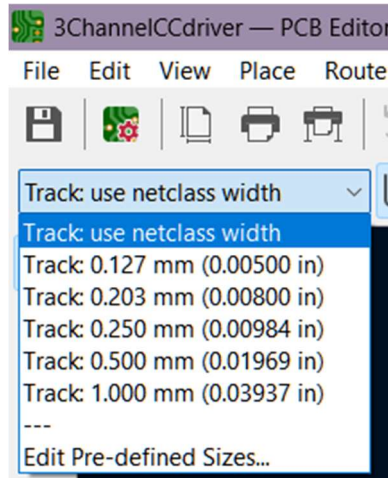
Trace Widths

Traces need to be a certain width in order to handle a certain amount of current. How wide? Use a calculator. <https://www.4pcb.com/trace-width-calculator.html> works fine.

Most boards are 1 oz copper.

Type in your current load, temperature rise, and ambient temperature. A 10-25C temp rise is standard. Trace length doesn't matter unless you want the total resistance (and thereby voltage drop) of the entire trace. For a 2 layer board, the width must be as wide as the result for "External layers in air".

Now, to add that to Kicad. Go under this dropdown and edit pre-defined sizes.



Add your track widths.

Now when routing, you can select the width on that dropdown. Alternatively, press "w" while routing a trace to change its thickness.