PCB Design Suggestions

EEC 134: Application Note

Andy Chen

I. Introduction

In this application note, I will discuss how to design, layout, and refine a printed circuit board (PCB) realization of a circuit. I will touch upon the challenges inherent in PCB designs and also comment on the particular obstacles designing for RF signals entails. This note is primarily focused on the design of a PCB for the EEC 134 radar senior design project with the KiCad software. I assume you, the reader, will already know the basic fundamentals of KiCad; this note will cover suggested design rules rather than give step-by-step instructions.

II. Before Beginning - Prerequisites

A. System

Before starting a PCB design, it is necessary to have a system and circuit schematic. Your system will most likely be close to your Quarter 1 design: there will be a baseband function generator that will control a VCO that will then be split to the transmit antenna and mixer — a receive antenna will capture the reflected signal and send that to the mixer which will output to your baseband filtering and signal processing. When thinking of your system, split it into a baseband portion and an RF portion. Do not make your RF portion overly complex because there are additional obstacles that will need to be addressed for higher frequency signals.

Your RF components will most likely have their own application circuits on their datasheets. For your baseband circuits, be sure to choose standard values for your passive components if you are going to put them on the PCB.

When picking your system components, try to match their required supply voltages. Multiple different supply voltages will require multiple circuits to supply those specific voltages which will take up significant space. In addition, each different voltage will require a different trace to carry it to the component which will increase the complexity of your design.

B. Schematic

The schematic will be the blueprint to your PCB design: your schematic should look close to what you want your PCB to end up looking like. There will be some slight differences on the actual PCB because it will have multiple copper layers but it will be much easier to copy over the schematic than to start from scratch. Thus, try to arrange so your signal has a clear path to follow and not zigzag.

It is dire that you get the schematic correct because using a netlist will be much easier than setting the net of each pad later on in Pcbnew.

C. Footprints

You will probably need to create the footprints for your components. Layouts are usually given on the datasheet or can be searched for if they are common packages. Double check to make sure that the pads are properly numbered and that the footprint is of the correct orientation — if the footprint is accidentally mirrored or incorrect, the PCB will not match your schematic and things will fail.

If your component has multiple pads with the same net, you can set those pads to the same pad number so you don't have to connect multiple pins to the same thing in the schematic. Remember, it is much easier to set the nets in Eeschema than to manually change their nets in Pcbnew.

III. Initial Design - Modularity

For the initial design, try to keep things orderly. Make sure that your traces are straight and avoid 90° turns. Try to keep your components lined up so that the pads are on a straight line, making it easier for you to draw straight traces.

When you are aligning things, I will suggest changing your grid units to millimeters and setting a small but not too small grid resolution for this part of the process. That will make it easier to read the exact XY position or move them a precise amount. You can also change the position of components by editing their XY values after pressing E on them or selecting edit from the right click menu.

The coplanar waveguides that you will be using to carry your RF signal require unbroken ground planes underneath them; try not to put any traces on the bottom layer in your RF portion or at least, make sure to not put a trace crossing your RF trace. You should also be via fencing your RF board to ward against unwanted signal propagation.

The easiest way to via fence is to create a via footprint, assign the pad to your ground net, and then make a footprint array using CTRL+N — you can select the vertical and horizontal separation and the number of footprints. Use these footprint arrays to make dense fences along your primary RF signal paths and to place vias loosely throughout your RF portion of your PCB (I recommend putting them in your baseband as well if the two are connected).

Things will most likely go wrong with your first design, either with the PCB or some circuit error that wasn't caught in time. As such, the initial design should be modular: split your design into multiple segments and create a smaller PCB for each. I recommend splitting into baseband and RF boards at least, and if possible, make your RF portion into multiple boards with one or two components and their matching network each. Give each board their own test points or SMA connections for inputs, outputs, supply voltages, and ground. Another advantage of splitting your RF portion into multiple boards is the ability to test each component using your synthesizers and spectrum analyzer. Otherwise, the only way to test is by cutting a SMA coaxial cable and using that as an antenna to hopefully pick up leakage from your RF coplanar waveguides; unfortunately, this method cannot accurately determine the amplitude of the signal and might pick up noise from other parts of the circuit.

It is very important that you provide many test points on this first design because you will need to test that each part is working correctly. Along the baseband signal traces, place 0 Ω jumpers interrupting the line and test points on either side. This will allow you to read the signal at that point, and if necessary, cut the trace at that point by removing the jumper and input a new signal at the other end using the test point. You should also do this for your supply voltage traces so you can isolate a short to ground due to incorrect soldering.

IV. Testing and Refining Your Design

When you inevitably run into a problem with your first PCB design, you should be able to isolate the problem using the various test points and 0 Ω jumpers. For your initial solder, put on everything except for the 0 Ω jumpers, then go through your signal and supply voltage traces one by one to make sure that each component is functioning as it should. With the many test points, you should be able to input test signals to each major part of your system and check that their outputs are as expected. Similarly, you should be able to supply power to individual parts of your system to check whether you are drawing the correct power. Once everything has been tested, you can solder on the jumpers to test your system as a whole.

As mentioned before, if you haven't split your RF portion into multiple boards, it will be tough to test if you are getting the correct signal power/frequency at each stage. You can cut a coax probe to use as a makeshift antenna to catch leakage from your RF traces to see if your signal is in the correct frequency range.

If everything works out perfectly, try unifying your separate boards. I would suggest keeping at least a few jumper/test points because other issues may arise (bad chips, soldering issues).

V. Additonal Notes

You can make graphics using the Bitmap2Component tool in KiCad. Just pick an image, preferably with high contrast, and the tool should be able to make it into a footprint.

The PCB calculator tool will allow you to find the correct width for your RF coplanar waveguides.

Double check that all your references are placed correctly. Decide upon an easier orientation system for them, particularly for the cluttered areas. One possibility is to always make your labels oriented in the same direction as the component is, i.e. if your resistor is longer horizontally, make the label horizontal. This will make it so you can easily identify the correct value of passive component to place in that location. You can also just read all their values off Pcbnew if you imported a netlist from your schematic as you should have.

If you must, you can manually set nets by turning of the automatic design rules checking (DRC). I don't really recommend this because there will be no netlist to fall back on and the DRC won't be able to catch your mistakes.