

Mithril - FMCW Radar

by

Tomas Esson, Ajay Thakkar, Juan Jimenez

tesson@stevens.edu, athakka5@stevens.edu, jjimene6@stevens.edu

May 16, 2024

© Tomas Esson, Ajay Thakkar, Juan Jimenez
tesson@stevens.edu, athakka5@stevens.edu, jjimene6@stevens.edu
ALL RIGHTS RESERVED

Table of Contents

1	Introduction	1
2	Project Description	2
2.1	FMCW Radar PCB	3
3	Resources	4
4	Radar Theory	5
5	Part Selection	6
5.1	Voltage Controlled Oscillator (VCO)	7
6	PCB	9
6.1	Schematic	9
6.1.1	ORCad Capture Overview	9
6.1.2	Circuitry Good Practices	12
6.1.3	Transmitter	14
6.1.4	Receiver	14
6.2	Layout	14
7	Digital Processing	15
8	Networking	16
9	Results	17
10	Issues	18

List of Tables

List of Figures

2.1	Flowchart of the Mithril system	2
5.1	Architecture Overview	6
5.2	VCO Datasheet Table	7
5.3	Tuning Voltage Graph	8
6.1	Component Database Search	9
6.2	Schematic/Footprint Examples	10
6.3	Editing Schematic Parts	10
6.4	Wires and Net Alias	10
6.5	Off Page Connectors	11
6.6	Coupling Capacitor	12
6.7	DC Blocking Capacitor	12
6.8	Capacitance of a Coupling or DC Blocking Capacitor	12
6.9	Test Pin	13
6.10	Power LEDs	13
6.11	Potentiometers	14

Chapter 1

Introduction

The following includes small biographies on all the authors as well as their research interests and projects.

Authors' Biographies

Ajay Thakkar

Ajay Thakkar Ajay Thakkar is a junior majoring in Computer Engineering. He is interested in signal processing and lower level coding. Below you can find his GitHub: <https://github.com/athakkar2>.

Tomas Esson

Tomas Esson is an aspiring Computer Engineering at Stevens Institute of technology. He is an avid surfer and enjoys elegant math proofs. Currently pursuing interests in computer chip design, digital systems implementation, mathematical optimization of computer chips, and electrical engineering.

Juan Jimenez

Juan Jimenez is a Junior Computer Engineering student at the Stevens Institute of technology. Interested in the intersection between Artificial Intelligence, embedded electronics, and software engineering. To see more projects visit the following GitHub link: <https://github.com/jjimene1>

Chapter 2

Project Description

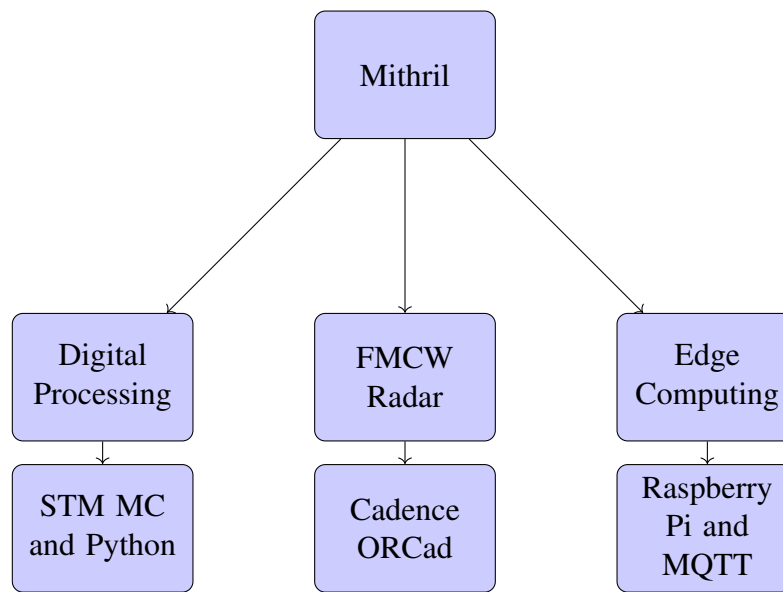


Figure 2.1: Flowchart of the Mithril system

Mithril is a nodal FMCW radar system that incorporates traditional FMCW radar, digital processing, edge computing, and distributed networking. The initial application of this project was for a military context. State of the art missile defense systems cost too much and are too big to be viable in many places, and while our system doesn't compare to systems like the Iron Dome or Patriot System it could allow for at least an alert for civilians to leave possible shelling and bombing sites. The system would include small radar nodes capable of detecting missiles and electronically steering their line of sight via phased array antennas. Multiple nodes could work in tandem to communicate their findings, allowing for a wider range of sight and active tracking of projectiles. The network would be able to function regardless of how many nodes there were, and still continue to function if nodes are destroyed.

As can be seen in Figure 2.1, the radar was designed as a standalone PCB in ORCad, digital processing was handled by STM microcontrollers, and distributed networking is done via Raspberry

Pi's and the MQTT protocol. All of these components were designed, engineered, and interfaced from scratch with a limited budget of 2000 dollars.

2.1 FMCW Radar PCB

The heart of the project is a standalone PCB capable of FMCW radar.

Chapter 3

Resources

Chapter 4

Radar Theory

include the benefits of using higher frequency signals

Chapter 5

Part Selection

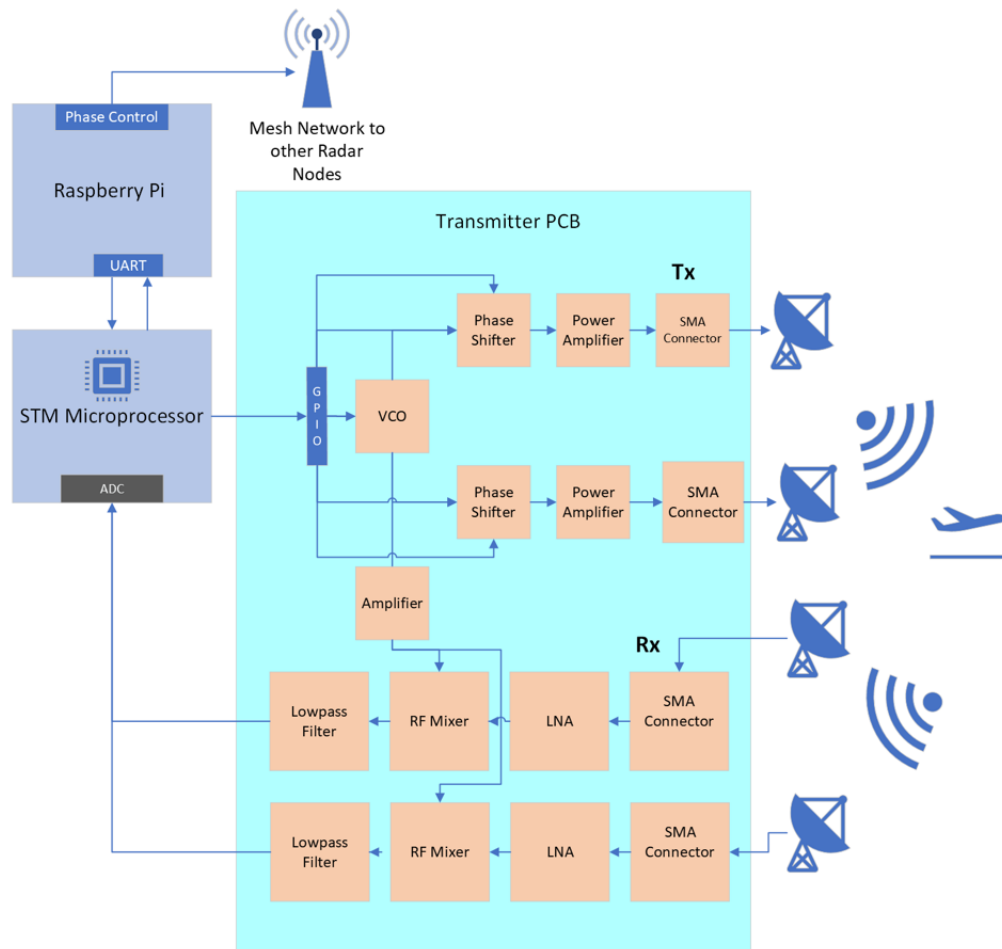


Figure 5.1: Architecture Overview

In this chapter, we will go through the architecture of the PCB and show what parts we picked and what purposes they serve. Figure 5.1 shows the overall architecture of the project.

5.1 Voltage Controlled Oscillator (VCO)

The first step in creating a radar is signal synthesis. You essentially need to create an alternating current signal that can go through your antenna and radiate out into the air. To reiterate from Chapter 4, higher frequency signals are important for a better radar resolution, and so it is desired to have a signal that is high in frequency. Naively, at the beginning of this project we thought we could use a 16 bit DAC to synthesize our signal but after finding out its max clock rate was 1 MHz, we realized a DAC was not fit for this. All we needed was something that could create a simple sinusoid at a very high frequency.

Voltage Controlled Oscillators (VCO) do just this. They take in a "tuning voltage" which corresponds to a certain frequency of sinusoid which it will output. The circuitry for this is beyond me, but for a good overall guide on VCOs you can check out DigiKey's article [here](#).

Now that we knew how to synthesize our signal, it was important to find a good VCO that had a high output bandwidth. Whatever bandwidth the VCO has will impact what parts we can get in the other stages of the RF chain since these must be within the VCOs operating regions.

PERFORMANCE SPECIFICATION	MIN	TYP	MAX	UNITS
Lower Frequency:			1530	MHz
Upper Frequency:	2700			MHz
Tuning Voltage:	0.5		10.5	VDC
Supply Voltage:	4.75	5.0	5.25	VDC
Output Power:	+4.0	+7.5	+11.0	dBm
Supply Current:		15	30	mA
Harmonic Suppression (2 nd Harmonic):		-15		dBc
Pushing:			25.0	MHz/V
Pulling, all Phases:			35.0	MHz pk-pk
Tuning Sensitivity:		140		MHz/V
Phase Noise @ 10kHz offset:		-87		dBc/Hz
Phase Noise @ 100kHz offset:		-110		dBc/Hz
Load Impedance:		50		Ω
Input Capacitance:			50	pF
Operating Temperature Range:	-40		+75	$^{\circ}\text{C}$
Storage Temperature Range:	-45		+90	$^{\circ}\text{C}$

Figure 5.2: VCO Datasheet Table

Tuning Curve (Typical)

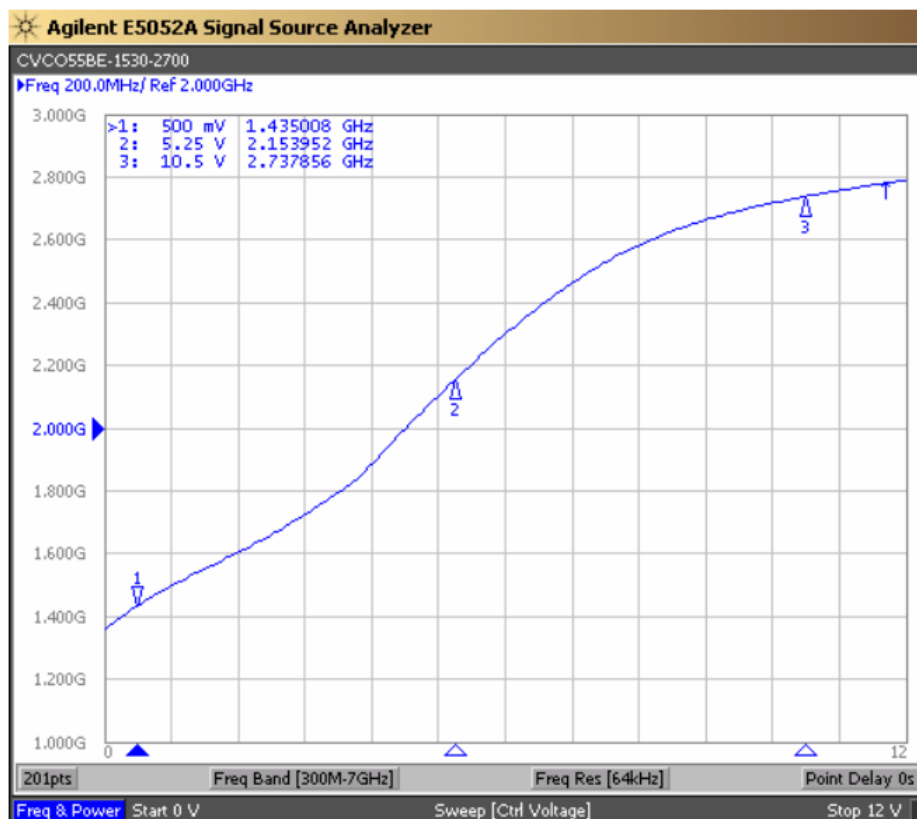


Figure 5.3: Tuning Voltage Graph

We chose a VCO from Crystek which can be found on Digikey [here](#). Looking at the specifications table in Figure 5.2, some good things to look for are first and foremost the frequency range of the part. It goes from about 1.5-2.7 GHz, which is a pretty wide range and would support a lot of other parts as it also covers the Wi-Fi band. The second thing to look for is output power. The output power can be a constraint for other parts, since they might have an absolute limit on their input power, and it is useful to know the output power for finding out how strong your signal will be once it propagates out of your antenna. Here we can see the output power is around 7.5 dBm, and this will be split four ways. Decibels are a logarithmic scale, so we cannot just divide by four but instead use a decibel calculator like [this one](#).

PCB

6.1 Schematic

6.1.1 ORCad Capture Overview

ORCad comes with a schematic software called Capture CIS which we used to create our schematics. As a reference to both us and others, this section is dedicated to providing a small walkthrough to using the software and some nice-to-know features.

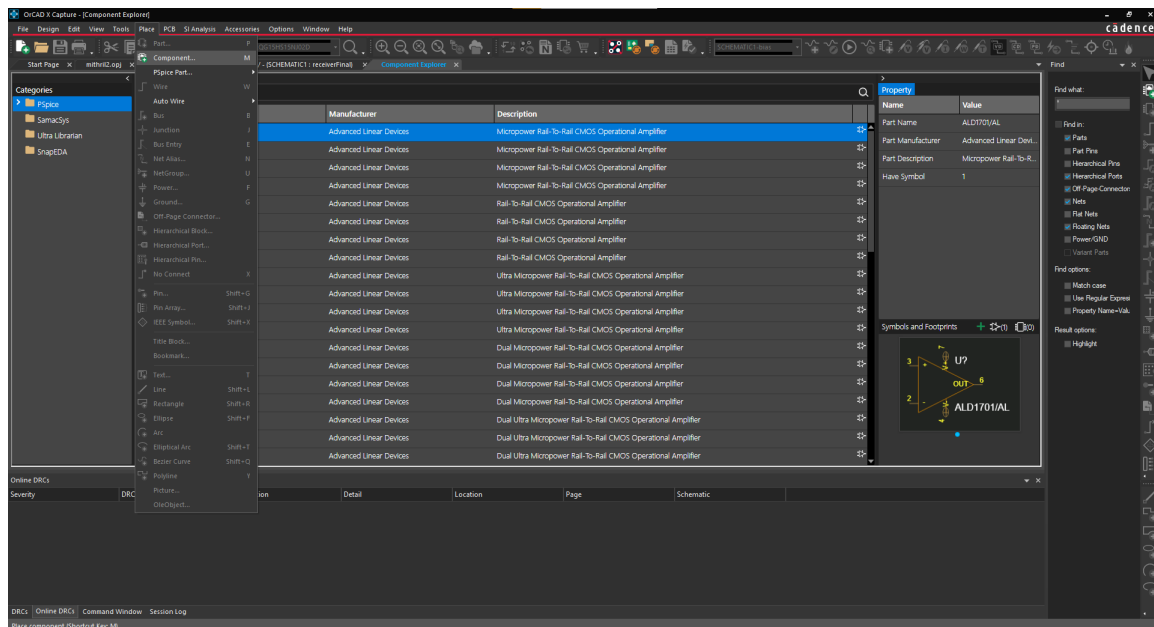


Figure 6.1: Component Database Search

The first important feature we found was the component database, which can be seen in Figure 6.1. By clicking on the icon that has a small chip and cloud will take you to this page which contains subdirectories "PSpice" "SamacSys" "UltraLibrarian" and "SnapEDA". PSpice contains parts that have simulation properties, but we ignored this as we did not know how to use PSpice.

The other three are different databases containing schematic symbols and layout footprints for parts that can be found on major retailers like DigiKey, Mouser, and Arrow.

...	SI8645BB-B-IS	Silicon Labs	SI8645BB-B-IS Silicon Labs, PCB SMT, 4-Channel Digital Isolator 150Mbps, 2.5 kV, 16-Pin SOIC	
...	TMP125AIDBVR	Texas Instruments	±2°C 2.7V to 5.5V digital temperature sensor with SPI interface and -40 to 125°C o	

Figure 6.2: Schematic/Footprint Examples

You can search for parts in the search fields for the different databases to try to find a part that has a schematic symbol and footprint available. You can see in Figure 6.2 that parts with the symbol and footprint available will have an amp and chip symbol next to them. The box means there is a 3D CAD model associated with them too. If you cannot find the symbol and footprint for a part, we recommend going on Mouser and finding the part, then requesting the symbol and footprint. Usually, the schematic and footprint will be added to SamacSys after a couple of days.

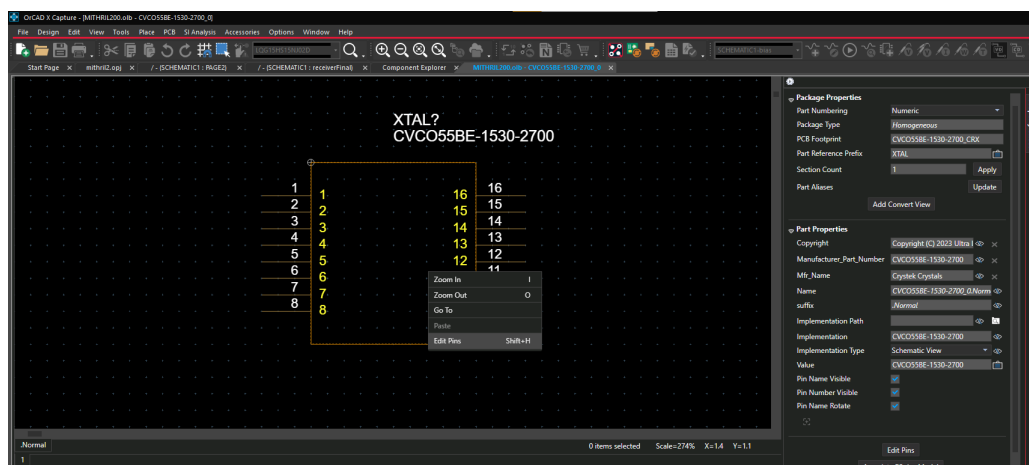


Figure 6.3: Editing Schematic Parts

Sometimes, these symbols and footprints are not correct. If the schematic symbol is incorrectly labeled, you can edit the part, then click edit pins to make sure all the pins are correctly labeled and numbered as can be seen in Figure 6.3.

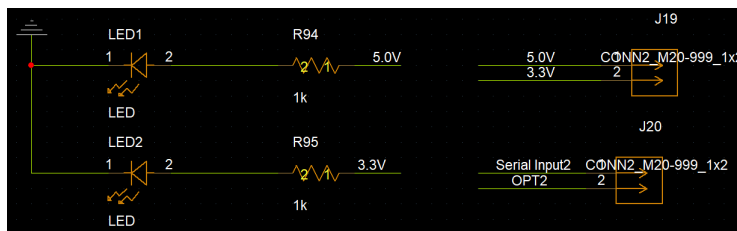


Figure 6.4: Wires and Net Alias

Once you've placed your parts down in the schematic, now comes time to connect everything together. To navigate through the schematic interface, you can use CTRL+Scroll Wheel to zoom in and out, and middle mouse button to scroll left to right.

You can use "w" to enter wiring mode, which lets you place down wires according to your grid size. After wiring, be sure to use "n" to enter net alias mode and assign aliases to your wires. As can be seen in Figure 6.4, there are two non-connected wires both with the alias "5.0V". This effectively connects them since they are under the same alias and will also label them in the layout when routing. Using the net alias helps with things like power where connecting everything that needs power with a wire to your power source would make the schematic a mess. In short, all wires with the same net alias are considered connected.

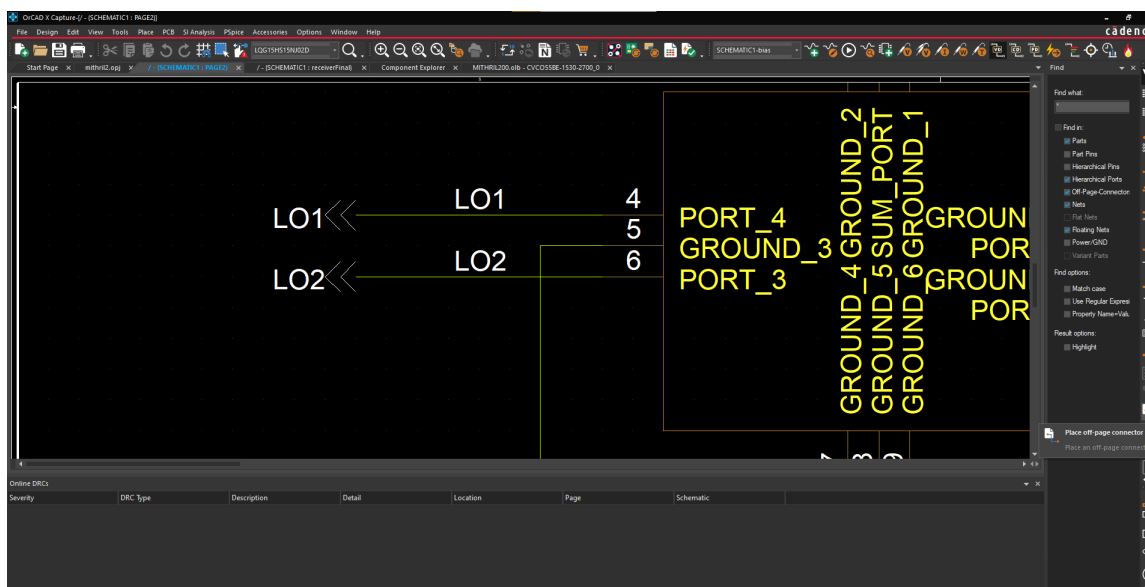


Figure 6.5: Off Page Connectors

If you run out of room or want to separate your schematics into different pages, you can connect wires from different schematic pages using off-page connectors as shown in Figure 6.5. Connect the off-page connector to a wire and name it the same on both schematic pages to have the wires connect.

6.1.2 Circuitry Good Practices

DC Blocking and Coupling Caps

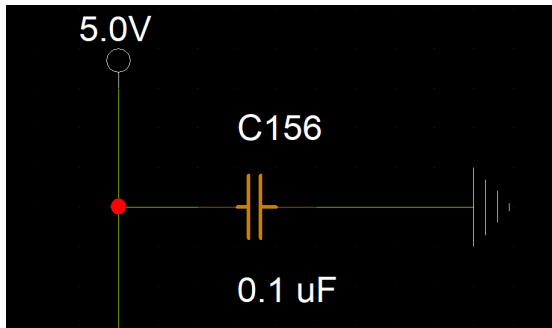


Figure 6.6: Coupling Capacitor

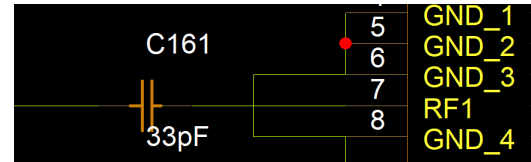


Figure 6.7: DC Blocking Capacitor

Two good practices to include in your schematics are coupling and DC blocking capacitors.

Coupling capacitors are usually used when giving power to some component or chip as can be seen in Figure 6.6. It can be thought of as a mini low-pass filter, where the capacitance of your coupling capacitor will affect the cutoff frequency of the filter. These coupling capacitors should be placed in parallel with the power wire that goes into the chip.

DC Blocking capacitors do the exact opposite, and are usually used when passing a signal from one component to another as can be seen in Figure 6.7. They can be thought of as mini high-pass filters, where the capacitance of the DC Blocking capacitor affects the cutoff frequency. These are placed in series with the signal wire.

$$C = \frac{1}{2\pi X f} \quad (6.1)$$

Figure 6.8: Capacitance of a Coupling or DC Blocking Capacitor

The equation for finding the capacitance of either the coupling or DC blocking capacitor can be seen in Equation 6.8 where C is the capacitance, X is the impedance, and f is the cutoff frequency. Essentially, a higher cutoff frequency corresponds to a lower capacitance value, and the only difference between the coupling and DC blocking capacitor is the coupling capacitor passes everything below that cutoff frequency, and the DC blocking capacitor passes everything above that cutoff frequency.

Test Pins and Grounds

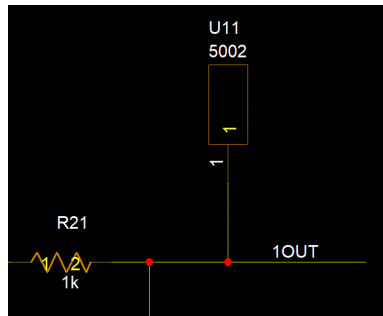


Figure 6.9: Test Pin

Other extremely important things to add are test pins as seen in Figure 6.9 and exposures to ground. Credit to Professor Muresan for telling us to add both of these as they were vital to testing and the operation of the board.

Test pins should be placed in parallel with signal wires so that you can hook in with an oscilloscope and examine what is happening. Really, after every stage in a signal chain it would be good practice to expose the signal via a test pin or if it is RF with an SMA connector. This allows you to debug analog things and find out how each stage is affecting your signal. We only added it in one place and after the fact wished we had placed more test pins so we could see what was happening from component to component.

Ground pins and pads are also really important. We overlooked adding a ground pin which was extremely inconvenient for us, and wished we had put ample ground pins and pads everywhere on the board. These are necessary if using external components with the board, since they should have a common ground.

Power LEDs

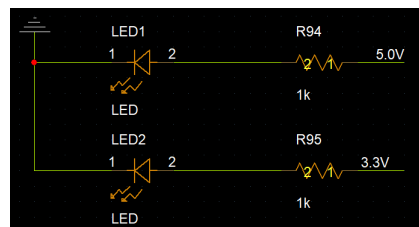


Figure 6.10: Power LEDs

Another nice thing to have are some LEDs connected to power, just so you know at the very least your board is turned on and everything is receiving power. Having a switch of some sort to block power is also something nice to have which we did not add.

Potentiometers

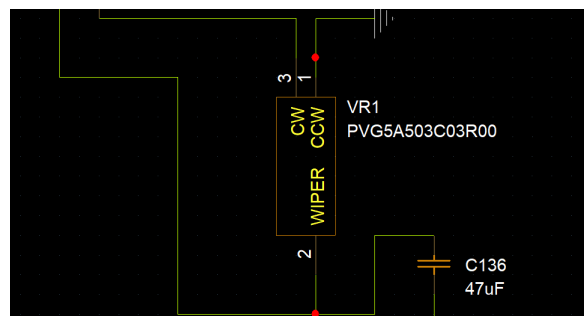


Figure 6.11: Potentiometers

Another extremely useful thing to add is potentiometers, as can be seen in Figure 6.11. In things like amplifying or filtering stages where resistors can change the cutoff frequency or gain of your circuit, potentiometers are very useful. From Figure 6.11, we can see that potentiometers have three ports: CCW, CW, and Wiper. How potentiometers work is that the wiper controls the resistance of the potentiometer, and turning it CW will make more signal go in that port, and turning it CCW will make more signal go into that port (oversimplified explanation). Therefore, it can be set up by just feeding the input to the wiper, the output to CCW, and CW to ground. This can help with changing gains and cutoff frequencies even after the board is printed and assembled.

6.1.3 Transmitter

6.1.4 Receiver

6.2 Layout

Chapter 7

Digital Processing

Chapter 8

Networking

Chapter 9

Results

Chapter 10

Issues

Index

Chapter

Digital Processing, 15

Introduction, 1

Issues, 18

Networking, 16

Part Selection, 6

PCB, 9

Project Description, 2

Radar Theory, 5

Resources, 4

Results, 17

Digital Processing, 15

Introduction, 1

Issues, 18

Networking, 16

Part Selection, 6

PCB, 9

Project Description, 2

Radar Theory, 5

Resources, 4

Results, 17