ENGR 357

SDR Project

Chrisner Garcesa

Andrew Nascimento

June 10, 2020

**Project Description**

The purpose of this project was to build a software defined radio, to which we decided that it would operate from 5 to 10 MHz. To control the device, we will be using a. Arduino Nano, and we will use a program provided to control the Arduino. We also had to build an attenuator, which would help with the SDR device.

**Circuit Description**

When looking up circuit designs, there was one that I found that I thought would be helpful. Our design wasn’t entirely based on it, but it helped serve as a guide on how we should go about designing the board. When I went about researching for the project, there was [one site](https://www.arrl.org/files/file/Technology/tis/info/pdf/020708qex013.pdf) that I found to be useful. This is where I got the basic layout for the project, and more details will be discussed later on.

Differential Op Amp

Tayloe Mixer

Bandpass Filter

Oscillator

Johnson Counter

Arduino Nano

The signal coming in will have to go through the bandpass filter, to which we decided it should operate around 5 to 10 MHz. In order to get the values to get that range, we used an [online calculator](https://rf-tools.com/lc-filter/) to get the desired values. The type of bandpass filter we decided to use was a Buttersworth, series first, 3rd order bandpass filter. For the Tayloe Mixer, there were a couple of other mux’s that we had gone through before getting to the FST3253. When designing this, it should ideally range from 0°, 90°, 180°, and 270°. There was also the oscillator, and to operate that, we used a Si5351, which in turn would be powered by the Arduino nano. Before that, we have the Johnson counter. For this stage, we chose a CD74A74E flip flop. We had originally gone with a different flip flop, but we went with the latter due to it being a bit faster than the original one we chose. The differential Op Amp was something that was implemented later on in the design phase, right before we had to turn the schematic. I had originally had just a low pass filter at the end before the signal went to the audio jack, but Andrew had added the differential op amp to help with the design.

**Software**

One of the most obvious parts of the board that needed software was the Arduino Nano. For this, we needed to download Arduino IDE and the code we used to test the oscillator and the clock was taken from [Frohne’s github](https://github.com/frohro/IQ_SDR/tree/master/Quisk). Other bits of software we used was [Python](https://www.python.org/), [WaveForms](https://store.digilentinc.com/waveforms-download-only/), and [Quisk](https://james.ahlstrom.name/quisk/), which would’ve helped us with testing when our boards were built.

**Build Instructions**

Due to the nature of this project, the board had to be built in 4 main stages. They can be done in the following order:

1. Bandpass filter
2. Tayloe Mixer
3. Johnson Counter
4. Differential Amplifier

A good place to start though is by soldering on the IC chips, since they are the most difficult to put on the board. I would suggest putting a bit of solder on one of the places where the pins should be, and after it settles, put the IC chip in its place (make sure it’s oriented correctly). Using the soldering iron, heat up the place where the solder is and hold the IC in place. If done correctly, the IC chip should be set in place, and then the rest of the pins can be filled with solder (make sure just to put a little bit and not too much, otherwise the pins may end up being connected together). After the IC’s have been placed in their places, the next thing to do is to build each stage and testing each one separately (however, I made the mistake of not doing that, so testing became a little bit difficult to do, but not impossible. Highly recommended building in stages). After each stage has been built and tested, you can move on to the next one and so on, until each stage has been built.

**Test Results**

The test results for the project are a bit mixed. There were some components that did work, and there were other areas where it did not work out as intended. One of the things that had worked out as intended was the bandpass filter, which was probably the easiest part to test out. The other stage that had worked out was the differential op amp. When I tested this part, it was recommended that I take out two resistors (R4 and R6), and I hooked up two probes at those resistors and connected a single probe to one of the capacitors at the output. The two probes connected to the resistors were connected to the wave generator, and the single probe should be the one that will measure the output. When I tested this, I had put in a small signal, one that was around 500 mV. For the output, I thought the signal coming out of the capacitors was fairly good, though I will note that the sinewave output was a bit smaller than the input.

What didn’t work was the Tayloe Mixer, the Johnson Counter, the transformer, and the crystal oscillator. When I had tested the transformer, I had connected a probe at the output of the transformer, which could either be pins 5 or 3. I had also placed probes at the beginning of the bandpass filter, which were connected to the two diodes. After running a signal through the bandpass filter and the transformer, the output of the transformer came out to be a bit messy. The output signal somewhat resembled a sine wave, but it looked a bit rough and sporadic. The next thing that I had tested was the Tayloe mixer, and the signal that came out more or less the same as the transformer. I had even tested it separately by placing two probes at the beginning of the Tayloe mixer and measured it at the end, and the signal appeared to be just as messy. The last thing that I had tried testing was the Johnson Counter and the Crystal oscillator. However, this required using the code needed for the Arduino Nano, and the problem with this was the Arduino IDE kept crashing every time I tried opening it (I had tried to install older versions, but those versions also kept crashing on startup). At this point, it was fairly late, so I couldn’t test any further. A screenshot of a computer

Description automatically generated

Bandpass Filter

A screenshot of a computer

Description automatically generated

Differential Op AmpA screenshot of a computer

Description automatically generated

Transformer Output

A picture containing computer, table, sitting, black

Description automatically generated

Tayloe Mixer Output

**Considerations for future uses**

There were a couple of things that I had thought of for trying to fix the device. So far, the only things that I think may work as intended are the Bandpass filter and the Differential Op Amp. However, there is a lot more work to do with the other parts of the board. Here are a couple of ideas on how to fix:

1. Fixing the transformer: the transformer turns ratio may be responsible for this. From how it was built, it was a step-down transformer, so probably changing it to a step-up transformer may help. Another thing that will have to be taken into consideration is the turns ratio. Due to time constraints, I had little time to calculate the turns ratio, so I ended up using an online calculator to determine the turns ratio (the online calculator required input and output impedances, where I needed the output impedance to determine the turns ratio, which I think wasn’t correct).
2. Reconfiguring the Tayloe Mixer and Crystal Oscillator: I also have reason to believe that both these stages would not work out as well, even if the I got the code working for them. Some things that I may do is try to simulate these stages again, but I will try to do it carefully. Another issue may be how I soldered the IC chips. I had soldered these chips by hand, and I had made sure that each pin was not somehow connected. With that said, there may be some pins that are accidentally connected, so I may try a different method of soldering the next time around. Lastly, the crystal oscillator may also potentially be a problem, so I may have a closer look on how to fix that part

**Conclusion**

After building and testing the board, there are still a lot more things to fix in order to get this thing working. With that said, I will try to work on this more in the future to (hopefully) get it into a working state. One of the stages that I will try to focus a bit more on detail is the Tayloe Mixer and the Johnson counter. I believe these are the more crucial parts of the board, and if I can get this working, I think that it’ll be easier to work with the rest of the board. The other stages and components should be fairly easy to work with, so I’m not too worried about the other stages.

Along with this project, I think that I had learned more about how to use KiCAD and how to design a PCB and what to take into consideration. With this board, there were some parts that were hard to solder on, so knowing that I will have to take into consideration who will be building the board, and how to make it easy as possible to build.

With that said, I will try to work on this project in the future. I think that it’ll be interesting to have this device actually work, and I will try to do more research on how to debug and fix this device.