Software Defined Microwave for Network Analysis

A Technical Report submitted to the Department of Electrical and Computer Engineering

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Connor Roos

Spring, 2020.

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Harry Powell, Electrical and Computer Engineering

Project Overview

Our project consists of a software-defined microwave device and open-source software that will replicate the capabilities of a Vector Network Analyzer (VNA). Much of the research for microwave technologies requires VNAs as they are extremely sensitive over multiple frequencies. The project will allow for microwave research that previously required VNAs to be conducted only using only a software-defined microwave and our open-source software. This project will make it such that microwave research can be more accessible to a larger number of research institutions and smaller companies that would not be able to afford VNA devices.

This Capstone Project focuses on testing this Software Defined Radio (SDR) Device will by comparing it to two different theoretical values for each test board, one from AWR simulations and another from the VNA in the laboratory. Once the theoretical values from AWR have been calculated, the boards will be sent out, and parts will be ordered so that they can be populated. After the boards have been soldered, the VNA will be used to find the “true” values for each board that the SDR Device can be compared to. Finally, the SDR will be used on each board and will be compared to both the AWR results and the VNA results. This should allow for the accuracy of the SDR to be thoroughly tested.

The ability for the SDR to be run remotely will be how this project handles the Microprocessor requirements for the Capstone Project, allowing a MyRIO to start the SDR and receive results remotely. This will let for the SDR results to be easier to transfer between devices, which should enable the device to be more useful in the coming future. All of these different portions of the project were conducted over Winter Break.

Circuit Design

We have tested 18 different boards to experiment with the software-defined microwave device that we believe would be beneficial in testing our current designs. The designs tested include both natural, imaginary, and complex impedances. For the natural impedances, six different resistor designs chosen for different magnitude responses over the range of negative five to negative thirty Decibels over the resistor. This wide range of magnitudes assists in testing the limits of the software-defined radio device to detect small signals. Three capacitors were chosen based on the frequency response they gave and how their frequencies responses changed as the capacitance increased. Three inductors were chosen following the same criteria as the capacitors, in order to better test the ability for the device to detect phase shifts.

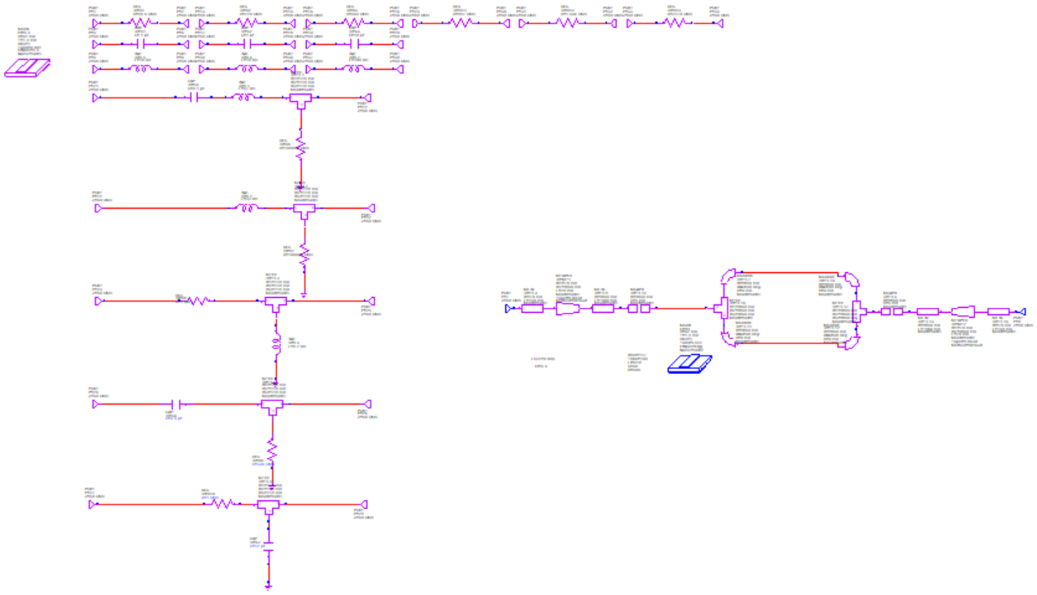


Figure 1: Test Circuits AWR Simulation Layout

Complex impedances were always used as filters. Two RC filters have been designed, one a high pass filter and one a low pass filter. Two more RL filters also were designed one as a high pass filter and one as a low pass filter. All of these filters have been designed to have a corner frequency of around 500 Megahertz. An RLC Band Pass filter was designed that allows signals in the range of 2200 to 2300 Megahertz. All of the filters will be used to test the ability of the software-defined radio to detect changes in magnitude over frequency. Finally, a Ring resonator was designed to test the ability for the software-defined radio to identify a circuit at resonance.

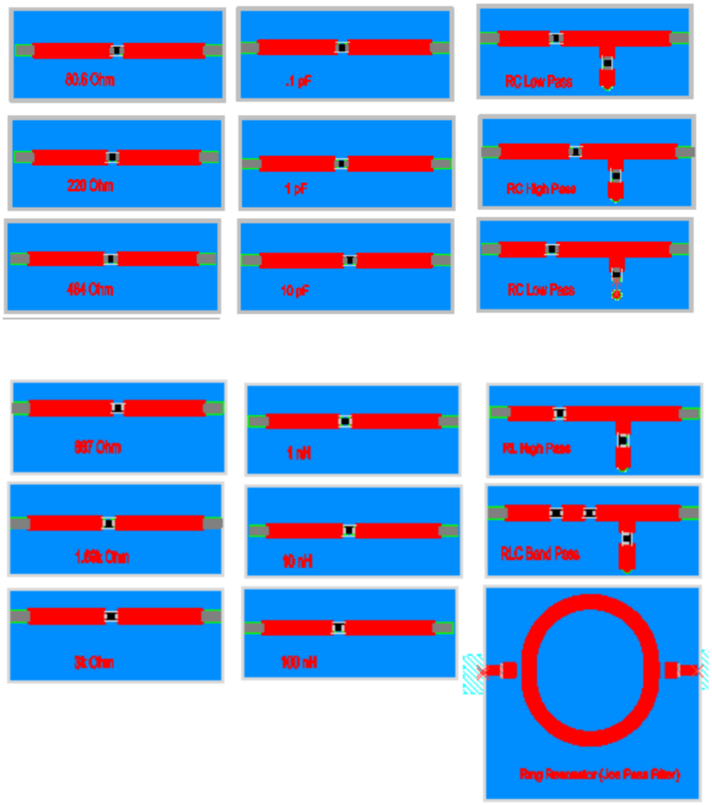
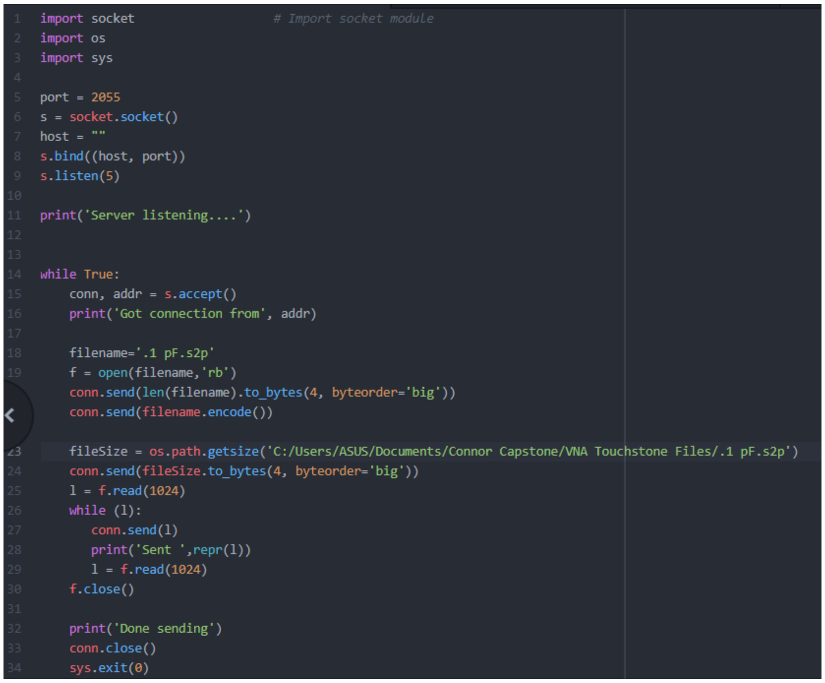


Figure 2: Test Circuits PCB Layout

Project Code

On the SDR side of TCP communication, this project will utilize the Python socket server TCP implementation. This allows for us to idle to SDR in python, and only run it when the message is sent from the MyRIO. It will run the code that sweeps through radio frequencies and returns the magnitude of the radio waves as they are seen by the device.



```python
import socket                     # Import socket module
import os
import sys

port = 2055
s = socket.socket()
host = ""
s.bind((host, port))
s.listen(5)

print('Server listening....')


while True:
    conn, addr = s.accept()
    print('Got connection from', addr)

    filename='.1 pF.s2p'
    f = open(filename,'rb')
    conn.send(len(filename).to_bytes(4, byteorder='big'))
    conn.send(filename.encode())

    fileSize = os.path.getsize('C:/Users/ASUS/Documents/Connor Capstone/VNA Touchstone Files/.1 pF.s2p')
    conn.send(fileSize.to_bytes(4, byteorder='big'))
    l = f.read(1024)
    while (l):
       conn.send(l)
       print('Sent ',repr(l))
       l = f.read(1024)
    f.close()

    print('Done sending')
    conn.close()
    sys.exit(0)
```

Figure 3: Client TCP Snippet

The MyRIO side of the TCP communications utilizes Labview in order to connect to the TCP server on the SDR. The MyRIO will connect using the IP Address of the SDR's computer. Once the two devices are connected, the SDR will run and will return the results that will be sent to the MyRIO, where the results will be displayed on a laptop that the MyRIO is connected to.

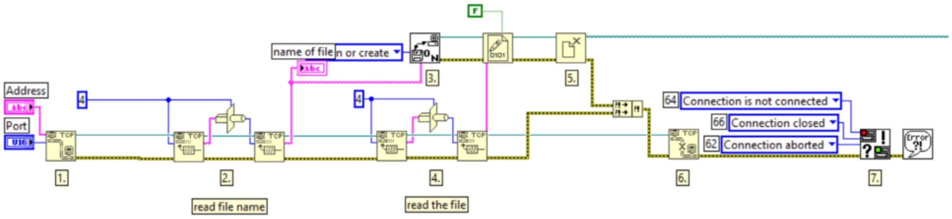


Figure 4: First Section of MyRIO Code

The MyRIO code can be split into three sections; the first section deals with the communication of the testing protocol. First, the VI creates a handshake with the server on the SDR computer and accepts the length of the file name in a TCP packet. Then the file name is sent using TCP, with the String Variable using the length given from the SDR computer. The VI then receives the length of the Touchstone files in Byte, followed by the file itself, which is then saved locally.

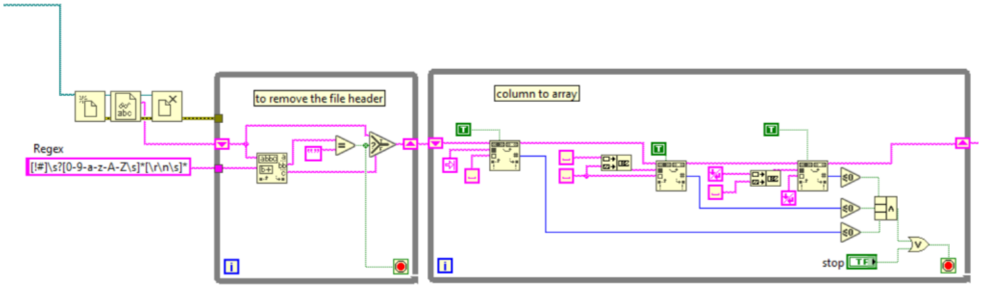


Figure 5: Second Section of MyRIO Code

The Second section of the code formats the Touchstone file into a 2-D array with each of the individual measurements given in Real and Imaginary Columns. This code was mostly designed using the code from https://forums.ni.com/t5/LabVIEW/Check-and-plot-data-of-an-array-of-touchstone-files/td-p/3816419?profile.language=en.

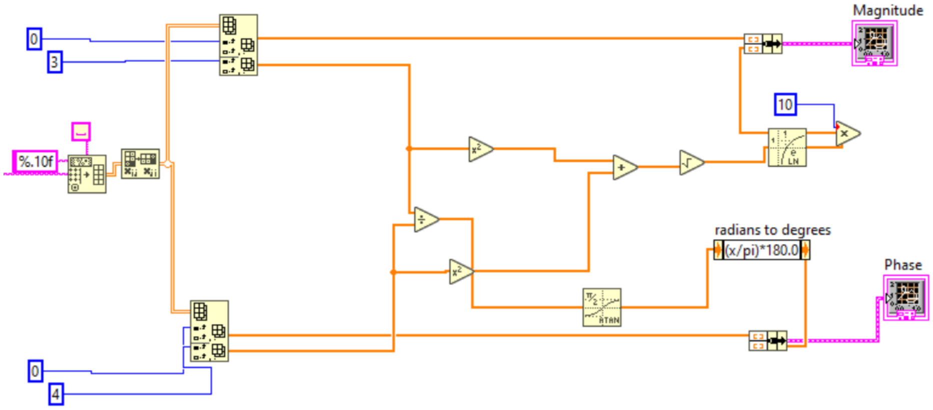
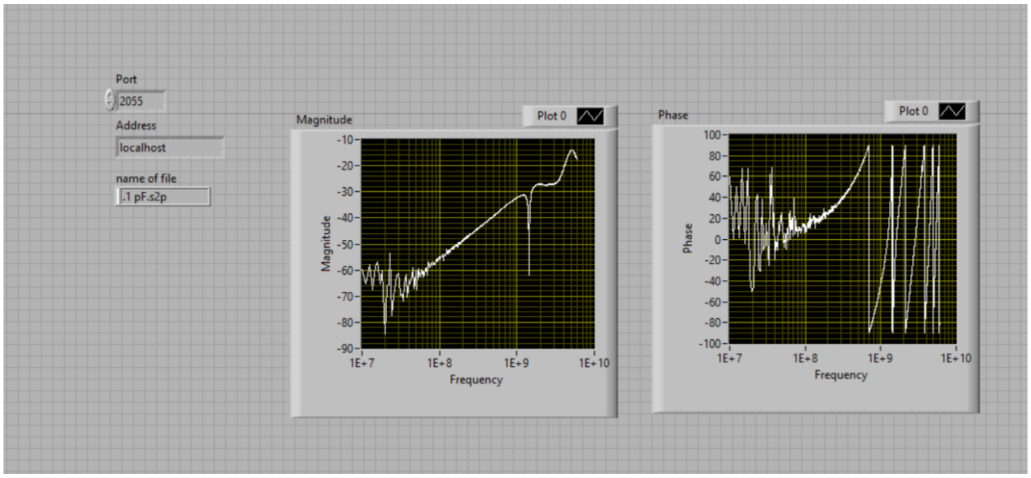


Figure 6: Final Section of MyRIO Code

Finally, the last section of MyRIO code extracts the S_{21} real and imaginary columns and converts them into a Magnitude and Phase array using mathematical operations in Labview. From this, graphs are produced on the front panel that allows users to see the plots that have been sent from the SDR.



Port

2055

Address

localhost

name of file

.1 pF.s2p

Figure 7: MyRIO Code Front Panel

Results

Many of the issues this project had stemmed from the shorter time frame that this Capstone Project was conducted during. Because of this, some of the pieces for the SDR device were not received, and the SDR device was not finished before the deadline for this research. As such, only the AWR and VNA measurements were produced for the research. This means that only around 2/3rds of the intended measurements for this project were able to be created. Unfortunately, this was not the only problem that occurred over the capstone project.

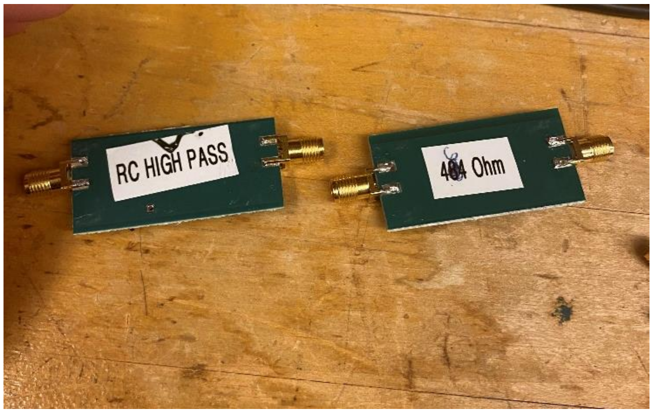


Figure 8: Test Board Examples

All of the test boards were populated with components and SMA connectors, however, it was discovered that the Verticle SMA ports, which were used due to how thick the boards were, were not very strong and could not withstand supporting the board on the VNA. Because of this, there were some broken connectors, such as the one of the left that was on the Ring Resonator Board. These were fixed by using a glue gun in order to strengthen the connection between the connectors and the boards, as seen below on the left.

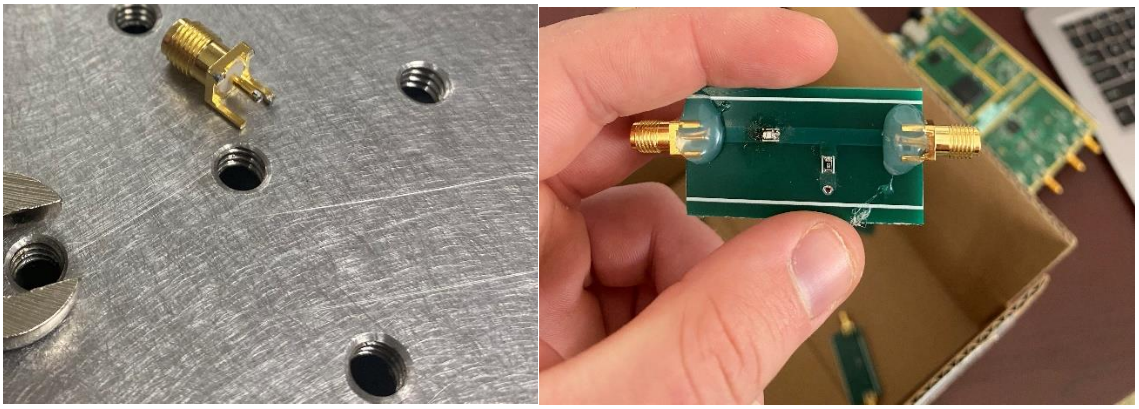


Figure 9: Broken SMA and Super Glue

Another problem occurred with the LabView MyRIO code. As much of it was designed with a non-real-time Operating System in mind, the final VI, while able to be deployed on the MyRIO, would not run as intended due to not being able to produce a static file. If more time were available for this project, a shared folder would be produced on the PC connected to allow for the MyRIO to save and read its files there.

Conclusion

While there were many portions of this project that did not reach the expected level of completion seen in the project proposal, a great deal of progress was made in designing a Test Protocol for the SDR device once it has been completed, the work done in this Capstone will allow for the SDR to be thoroughly tested as soon as it has been set up and connected in the FIR lab. The Test Boards and Test Code will be used in this project moving forward, so future work has been completed while the necessary parts were being shipped for the SDR.