Hound Doggies Final Report

Group Members: Zack Bennett, John Docalovich, André Magill, Joey Pye, Shane Ryan

Advisor: Jonathan Chisum

Table of Contents

1 Introduction

Our project attempted to scale down the existing RadioHound sensor while also making it more power efficient and less expensive. This system has a wide range of military and commercial applications including IED detection, cellular coverage maps, and the FCC implementing broadcasting frequency regulation. Moreover, one of the most promising areas of application is to use dozens of these sensors spread out in an array across a war zone. Figure 1 below demonstrates how an array of nine sensors might be set up for RF signal detection. Each square represents a sensor. The color red represents a high power level detected in the requested frequency range, while the lighter colors represent lower power levels. Knowing the powers detected by each sensor and their locations would allow us to find the source of the signal, represented by the blue star. Moreover, this information can be sent to and stored on a cloud server and used to produce a heat map of the surrounding areas where the colors correspond to the power levels present for a certain frequency band.

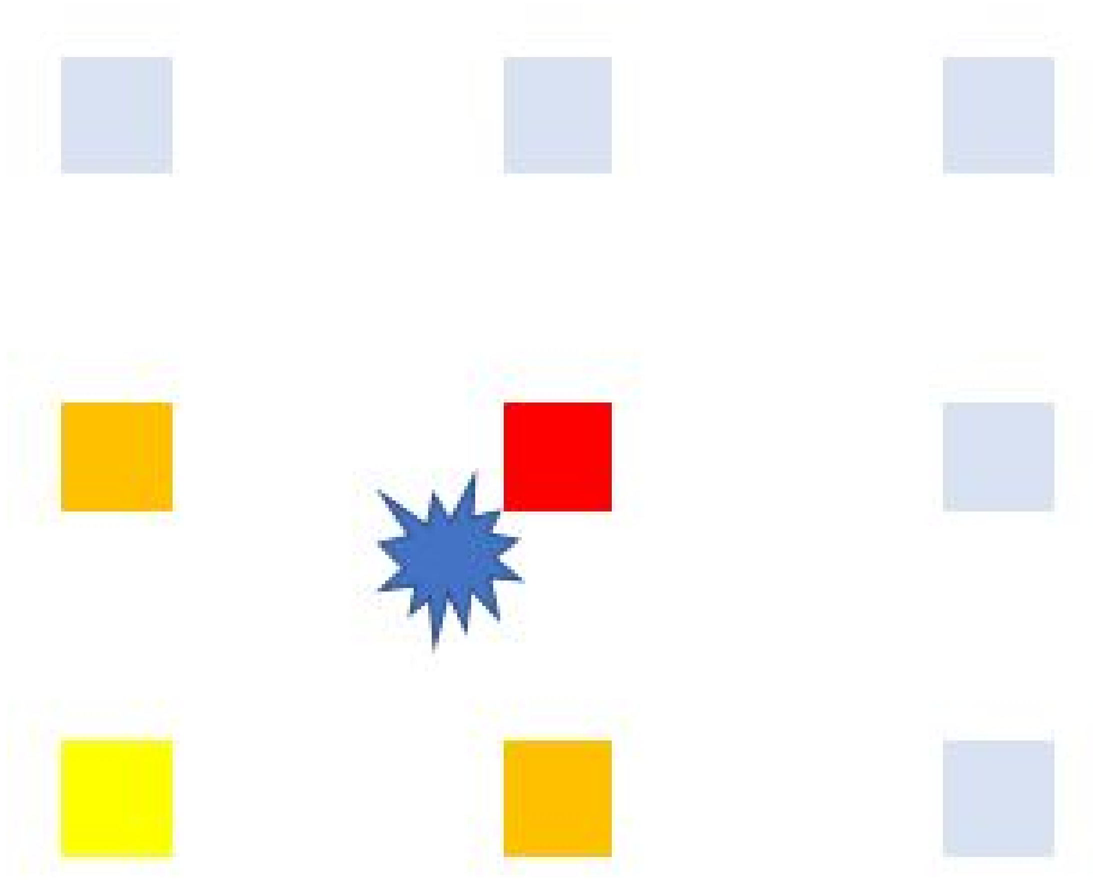

Fig 1. Diagram of an array of RadioHound sensors detecting an RF signal.

Thus, in a war zone, this technology could be used to identify the source of IED triggers or where an enemy is communicating from. It is best to produce as many of these devices as possible because the more devices in an array, the more accurate the heat map generated. Currently, the sensor is large, bulky, and costly; we tried to create a smaller system integrated onto a single PCB that was more cost efficient and provided a faster method of scanning than the present system. Through this, we wished to produce large numbers of these low cost devices to increase the coverage area monitored while also significantly cutting the overall price.

The end goal of this project was to be able to wirelessly give the device a frequency range and have the device return the power levels detected in that range. This project sought to solve that problem at a low cost with a low power device. Specifically, the goal was to create a device that costs less than $50 each and consumes less than 3 Watts of power. We intended to accomplish this by overhauling the current device to cut down cost and power consumption.

One of the most significant changes we made to the current RadioHound sensor is replacing the Raspberry Pi with a CC3220SF MCU/Network Processor chip. Reasons for doing this included the fact that the CC3220SF consumes considerably less power, costs less, and simplifies the internal communication between the different components of the board. The current RadioHound sensor has the Pi relay commands to the onboard MCU and then the MCU sends commands to the various components. Our design has only the onboard MCU sending commands. In addition, a GPS component was added to the system using the Adafruit MTK3339 chip. This allowed the RadioHound sensor the ability to send its location to the cloud as well as the power level data. The power supply was a LiPo battery plug in addition to a micro USB jack. A switch is used to select the onboard LiPo or the USB jack as the power supply. An analog circuit was used to produce a DC voltage proportional to the detected signal power level that can be sampled by the microcontroller ADC. Thus, the diode is able to integrate the incident power in the frequency bin without doing an FFT. This sped up the data analysis process significantly since no FFT is required.

In the end, we came close to but were unable to completely meet our expectations. One of the expectations that we had was the decreased power consumption. Our expectation from Chisum was to lower the power consumption to around 2.5 W. Although we were only able to bring the system down to about 4W, it is still a significant decrease from the 8W that the system consumed when we began working on it. As far as the pricing expectations, we were successfully able to save about $60 by removing the Raspberry Pi and an additional $25 by taking the RTL-SDR USB dongle out of the system. With purchasing the GPS chip, we tacked on about $35 to our project, for a net of $50 saved from the original design. This did not meet the initial expectations of making the system cost less than $10, however, we ended up shaving a significant amount of cost off the project, which is considered a success for our senior design.

The large battery that was used to power the system was another issue that we tried to solve. With the introduction of the power subsystem that we created, we were able to cut down the battery size significantly. The original battery was approximately half the size of a car battery, however our new design has a battery that is about 1” x 3”.

For the board itself, we planned on designing it so that every component could be included and functional on one PBC board. However after testing, we found out that only the GPS, power system circuit, and the analog circuit worked on the board. Therefore we were forced to use the individual evaluation boards for the microprocessor, the PLL, the LNA, the mixer, and the low pass filters.

2 Detailed System Requirements

The GPS subsystem requirements are that it must be able to send its location to the cloud to accompany the power level data. This way when transferring the node data to the GUI/web interface, the RadioHound site will able to plot the data in a heatmap that displays not only the geographical location of each node, but the signal intensity surrounding the nodes themselves. This GPS module must be in the form of a solderable chip that can be embedded onto a PCB board so that it can integrated into our design while minimizing parasitics that would come with using an external GPS device.

Our RadioHound device is designed to exist as a single node in a larger network of similar nodes. In order to coordinate the usage of these devices (receiving commands, synchronizing clocks, reporting scan data), the device runs an MQTT client thread subscribed to a broker through fmnc.cse.nd.edu. Thus the software on our device parses and packages MQTT packets in order to communicate and exchanges these over WiFi using the network processor onboard.

The MQTT process exists in a thread separated from the main microcontroller process, allowing for packets to be sent and received simultaneously and for the microcontroller to carry out tasks during communication time. As part of the established protocol, our device sends an ANNOUNCE packet every ten seconds to the RadioHound server to keep the connection alive and exist as an enabled device on the user interface.

The microcontroller requirements were to have an ADC capable of sampling a DC voltage with a sample rate of 500-1000 kS/sec. It also needed to have at least two SPI ports to communicate with the PLL/VCO and the VGA. Moreover, it needed WiFi connectivity support in order to enable it to connect to the RadioHound GUI and receive scan jobs.

The RF front end needed to be able to demodulate RF signals of a frequency range of 400 MHz to 6 GHz, down-converting them to an IF of 110 MHz. It accomplished this using the current RadioHound sensor architecture of an LNA, mixer with LO of a PLL/VCO, LPFs, and VGA.

The analog circuit needed to be able to taken in the amplified IF signal at 110 MHz and produce a proportional DC voltage which could then be sampled by the MCU's ADC. A matching network on the front end of the circuit is necessary to ensure maximum power transfer into the analog. A proper buffer to protect the ADC and allow for good input impedance for the produced DC voltage is also needed.

The power system requirements were crucial in the construction of the board. Originally, the power design specifications stated that we needed a working board that drew less than 2.5W of power. This could be powered by either a battery, USB or possibly even a solar panel; after tossing the solar panel idea around for a while we decided to drop it and stick with a battery and USB power option. Eventually, we realized that the components that were necessary to get our project to function as desired could not operate less than 2.5W. We did everything we could to get this as low as possible; in the end the power draw ended up being a little less than 4.5W total power. We needed a 3.3V and 5V rail for the components. To achieve this we used a USB power supply (5V) connected to a LD1117 linear regulator for the 3.3V supply. If we could not connect to the USB we had a LiPo battery connected to a 5V DC/DC converter. Originally we were trying to look for a linear regulator but the ones that we found that could send out enough wattage had efficiency that was much lower than was acceptable. Finally we settled on the 5V DC/DC converter that we used in our completed design.

3 Detailed project description

3.1 System Theory of Operation

The end goal of this project is to be able to remotely give our device a frequency range and have the device return the power levels detected in that range as a node in a larger detection network. To accomplish this, a remote backend communicates with all nodes and users may enter scan requests on a website GUI. These scan jobs are sent to the microcontroller in our node over WiFi and the start and stop frequencies extracted. Now a PLL/VCO tunes to each frequency in the desired sweep, a short delay occurs to allow for signal propagation, and the RF signal chain generates a voltage at the power detection circuit proportional to the incident power of that detection frequency. An ADC in the microcontroller samples this voltage and appends it to an array of voltages to report back to to the backend, where the data is then saved and plotted for visual analysis, along with a GPS locked location in order to deliver location results for the scan.

One of the most significant changes made to the current RadioHound device is replacing the Raspberry Pi with a CC3220SF MCU/ Network Processor chip. The MCU is used to send and receive information over WiFi, read the GPS data, communicate to the other ICs on the board via SPI protocol, and sample the DC voltage from the analog circuit. It also stores the frequency range to be scanned and the array of DC voltages until the data can be sent back to the RadioHound server.

In addition, a GPS component will act as a contact between our device and the GPS satellites orbiting earth. This allows the RadioHound sensor to send its location to the cloud to be read by other components of the project. Thus, the GPS component solves the task of figuring out when and where each piece of data is coming from during analysis. This data is then sent from the GPS module to the GUI / web interface to accurately plot the geographical positions of each node that is out on the field at that specific time. From there, the RadioHound web interface is able to take the plots and position data to display a heat map that shows the intensity of signal versus node location.

The power supply system is used to give reliable current and voltage sources to board components that require power. After assessing the current draw and power capabilities, we tailored the battery choice to be the smallest, lightest, and least expensive possible while still meeting requirements. We also added a USB jack so that an alternative power source could be attached if desired. A switch is used to select the onboard LiPo or the USB jack as the power supply.

The analog circuit takes in the IF signal at 110 MHz and produces a DC voltage proportional to the incident power level that can be sampled by the microcontroller ADC. A fast-switching diode replaces doing an FFT since it is a non-linear device capable of taking high frequencies and producing a DC voltage. Through calibration, the DC voltages could be mapped back to incident power levels.

3.2 System Block diagram

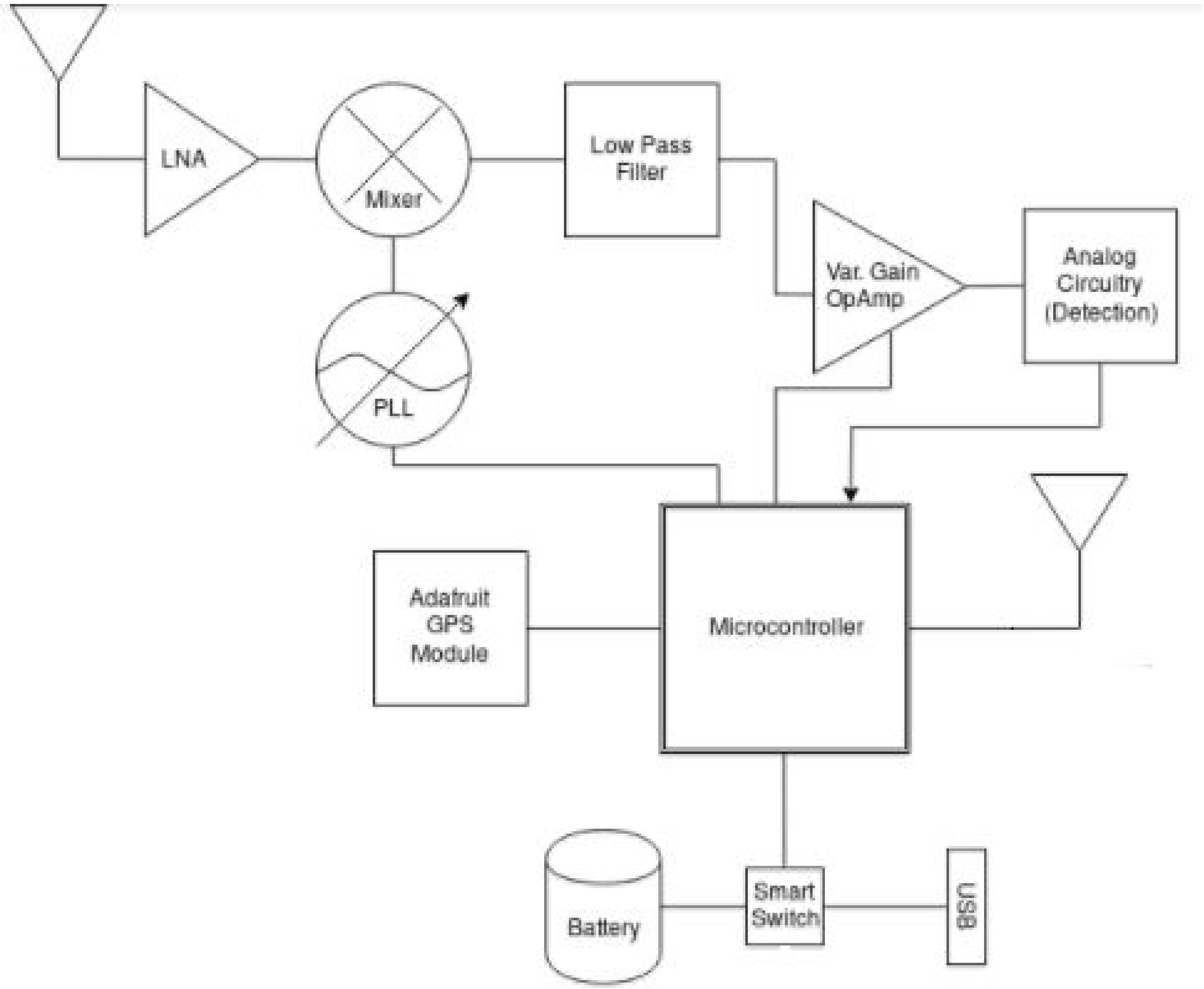

Fig. 2. RadioHound System Block Diagram.

The above block diagram shows the electronic components of the circuit board. These can be broken down into the following subsystems: the user interface, wireless communication, GPS module, filter system (including receiver and mixer), analog detection circuit, and power supply module. Those 6 subsystems will be discussed in detail in the following sections.

3.3 User Interface

Much of the user interface software has already been developed along with the pre-existing RadioHound project. The interface can be accessed at “fmnc.cse.nd.edu:9000” and lists the currently connected RadioHound nodes. In terms of inputs, the user will specify the frequency range to be scanned, the nodes over which to scan, and other optional parameters. The output is either a waterfall graph of power across the specified spectrum or a heatmap of power readings, which reveals the desired insights. We will need to calibrate our analog circuit to allow the cloud server to accurately convert the DC voltages back into power levels for displaying to the user - currently all measurements are relative.

We decided to use the bulk of the user interface software that already exists for RadioHound and build our software architecture to match the existing connections. The primary interface that we use is the heat map feature that allows the user to visually locate the RF source on a map. The only difference is that the output from our device was a single DC voltage that using our calibration can be converted to a power level, so much less backend computation is needed.

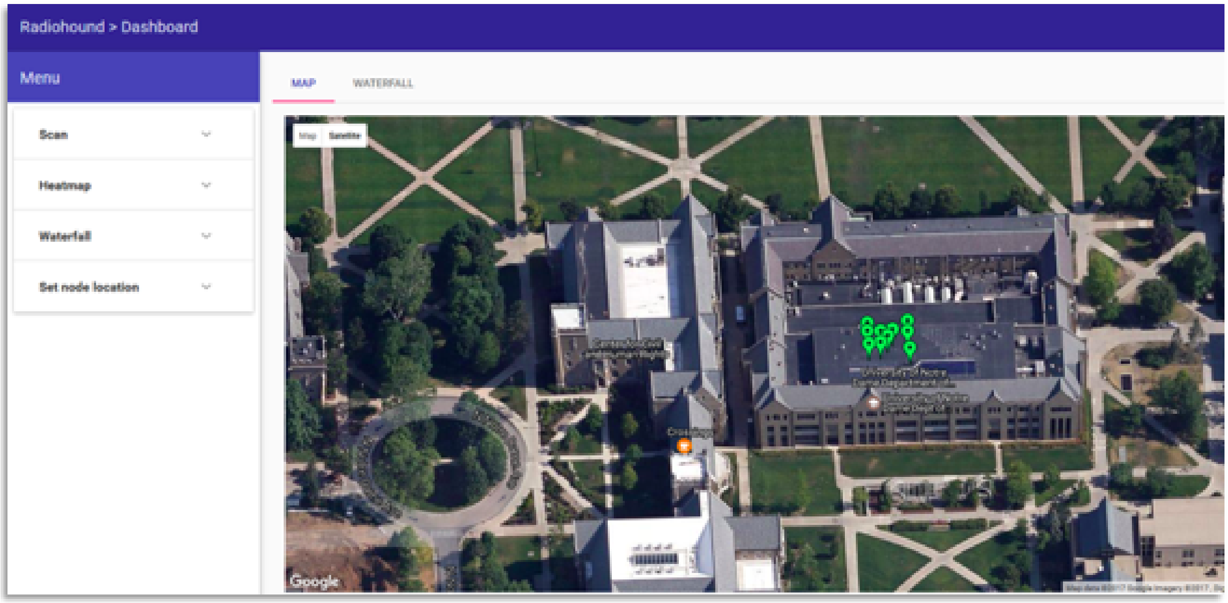

Fig 3. Website interface showing locations of RadioHound nodes.

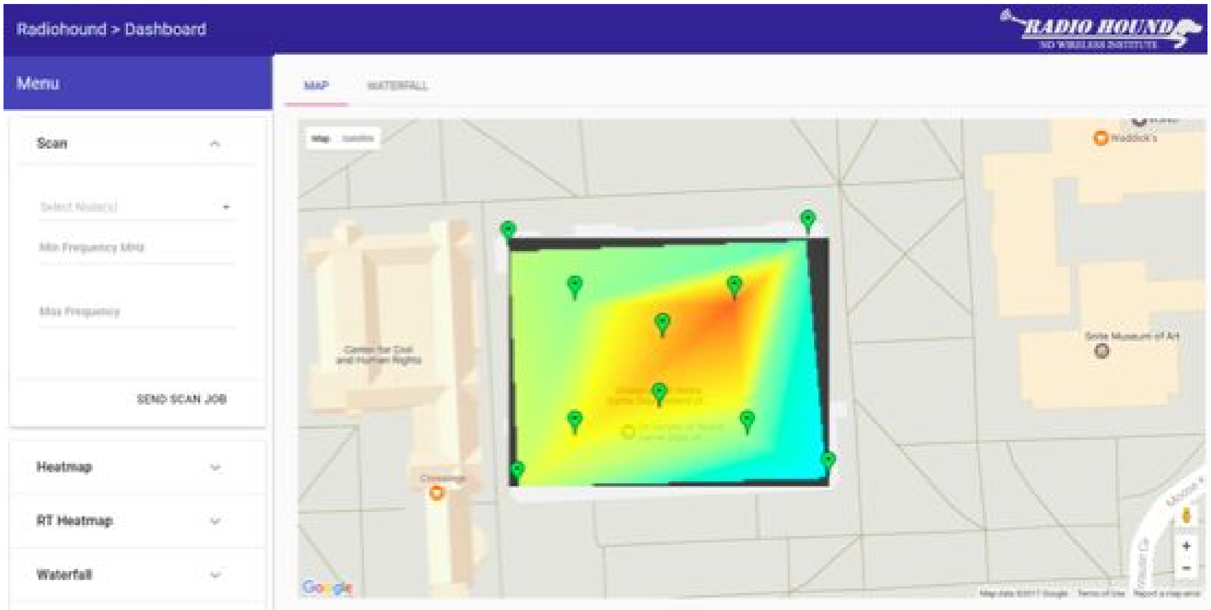

Fig 4. Heat map interface displaying the intensity of the frequency over an area.

3.4 Wireless Communication

We send the scan data back to the RadioHound backend and receive commands and scan jobs from the server using the CC3220SF microcontroller which has built-in wireless communication capabilities. Thus, we are able to wirelessly send data from each device to the server that hosts the user interface. Further, we receive the frequency range request from this backend as a scan job, along with requests for clock synchronization and reporting back system parameters and configuration.

We used a standard WiFi 802.11 b/g/n communication protocol on the network processor inside the CC3220SF chip. This separate network processor has stack protocols built in to handle transmission and reception of data using WiFi. Data is stored in a buffer before being communicated wirelessly using a 2.45 GHz antenna, and the actual communication protocols. We used the Taiyo Yuden AH316M245001-T antenna with the option of a Bluetooth antenna as well.

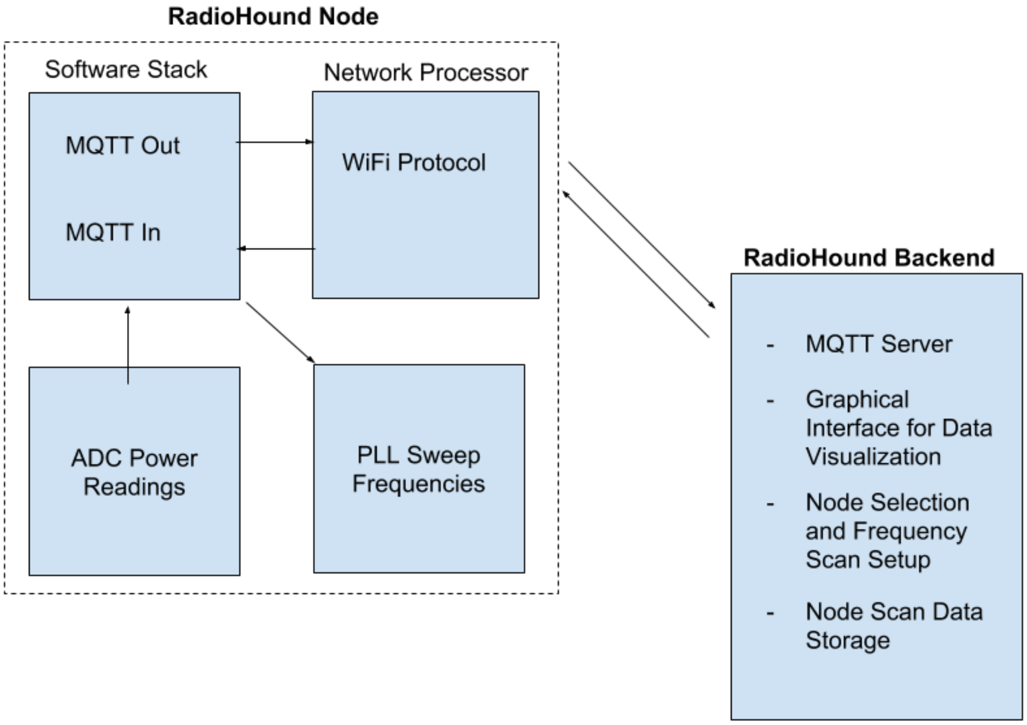

Fig 5. Block diagram of data flow through communication subsystem and interaction with the RadioHound backend.

3.5 GPS Module

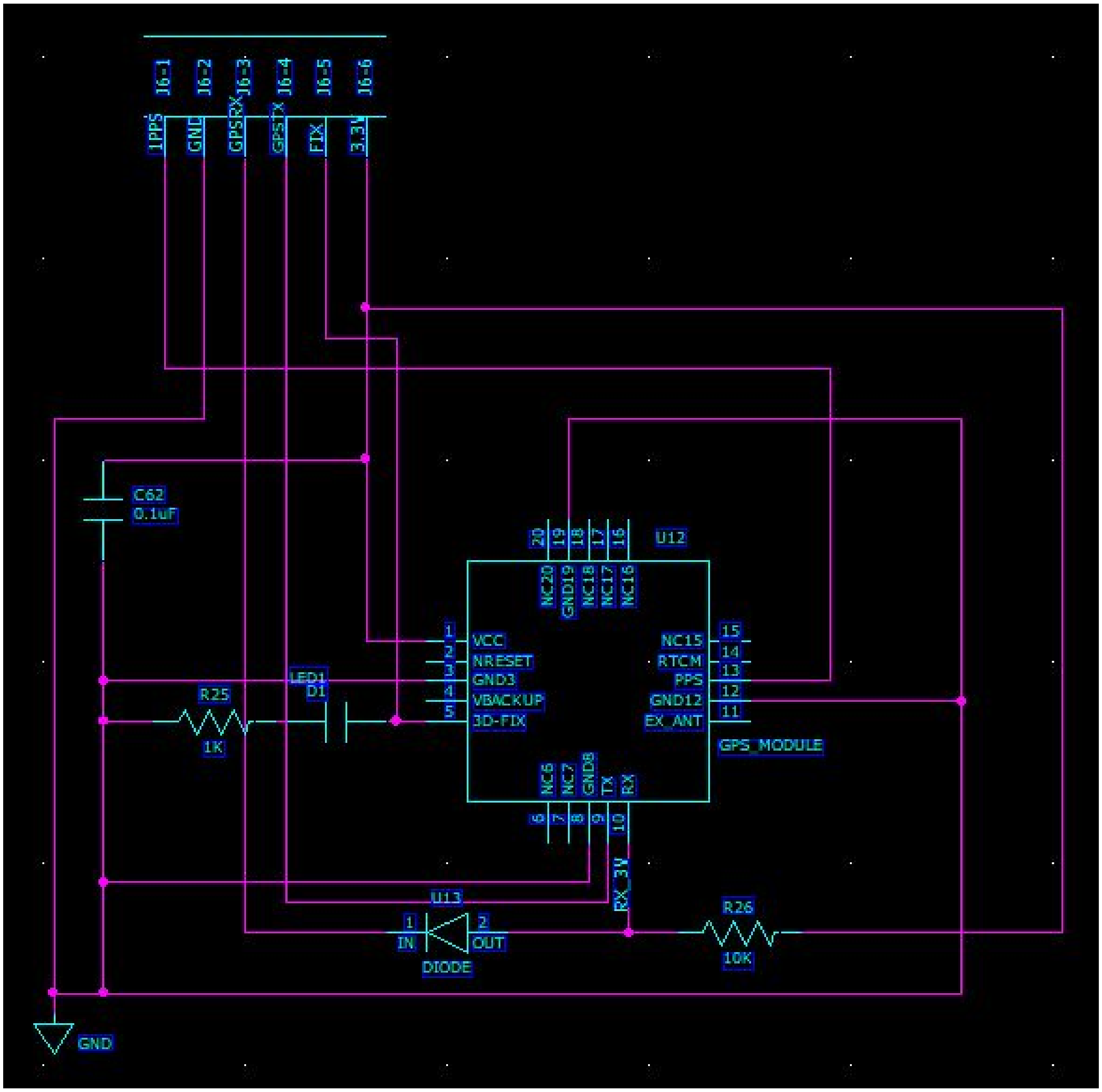

Fig 6. Schematic for the GPS subsystem.

The GPS module allows us to provide a accurate location of the device to the microcontroller. Following the build criteria for the chip allowed us to integrate the GPS module into the PCB.

We were successfully able to integrate the GPS chip directly into our PCB design. While we were inside Stinson-Remick, it is impossible to receive a GPS lock; however, outside we receive a lock within a few minutes of powering up. This location was accurate within a few feet at worst and a few inches at best. The coordinates we received came out to be 4141.6799N, 8613.8932W, which when plugged into Google Maps shows the location in the figure below.

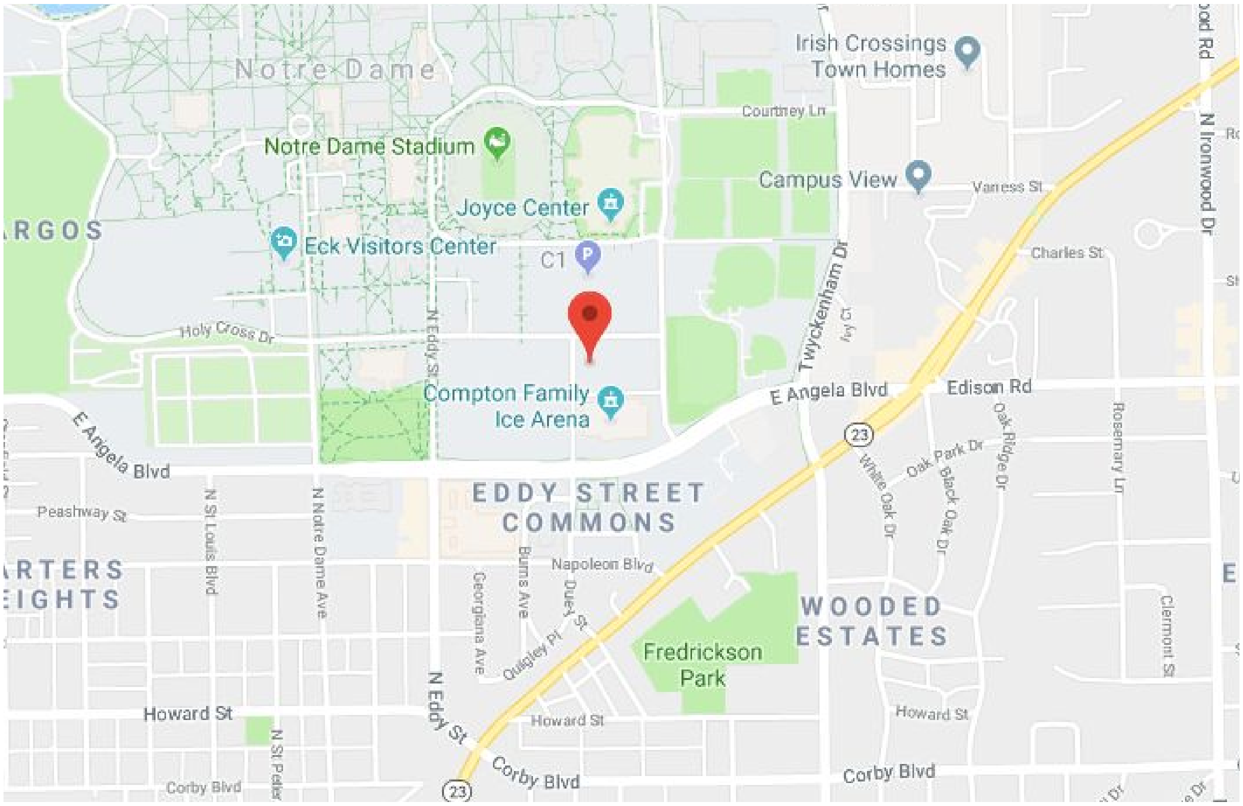

Fig 7. Location on google maps of coordinates received by GPS module.

We were consistently able to lock onto at least four satellites simultaneously which allowed us great accuracy in longitude and latitude as well as elevation and heading. Additionally, we could see the speed of the module as we were testing this while driving. We noted that the speed and heading were very accurate as well. While we don't necessarily need a speed or heading reading for our purposes, it is possible to further integrate them into a possibly mobile platform in the future.

We used the Adafruit GPS Module MTK3339 chipset. This module will meet our goals with its relevant features of:

1. -165 dBm sensitivity, 1 or 10 Hz updates
2. Ultra low power usage: 20mA current draw while tracking
3. 3.3V operation
4. Ultra small size: only 16mm x 16mm x 5mm and 4 grams

3.6 Filter System

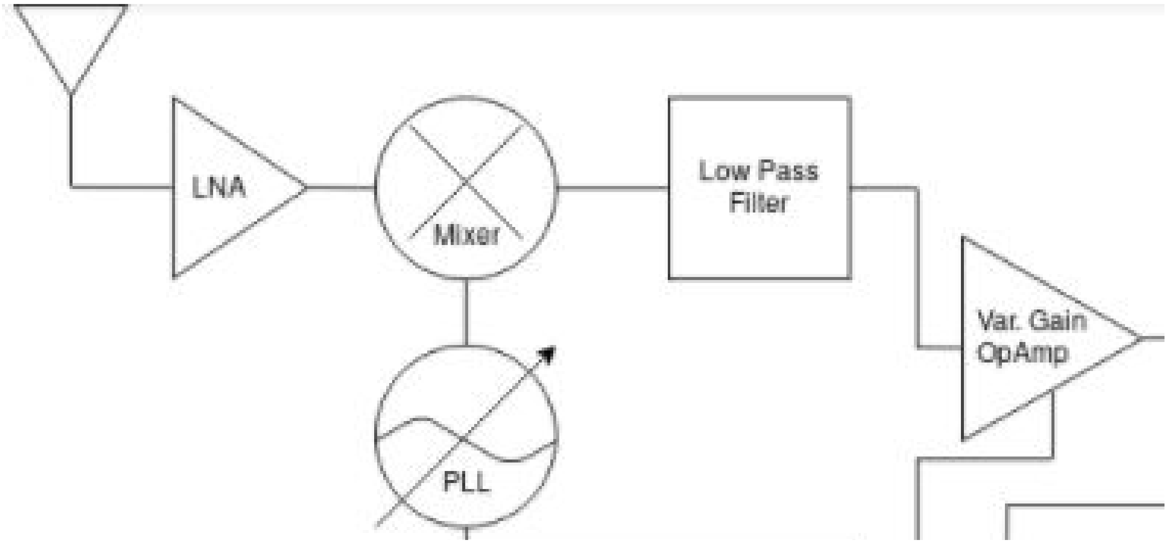

Fig 8. Filter System Block Diagram.

The filter subsystem is meant to describe the hardware and software that will be required to take the received RF signals and convert them into a lower frequency signal which then enters the analog circuit.

This system begins with a wideband telescoping antenna that can pick up the RF signal the user wishes to scan from the frequency range of 400 MHz to 6 GHz. Next, the RF signal passes through a low noise amplifier (LNA). The noise in an RF signal chain is often dictated by the first component in the chain so beginning with an LNA significantly enhances SNR. After passing through the LNA, the RF signal is mixed with the local oscillator (LO) signal that is generated by the phase locked loop voltage-controlled oscillator (PLL/VCO) chip. The mixer is set up to operate as an Image-Reject Quadrature

Hybrid. Generally, when two signals are mixed together, the output is four signals (the original two plus the sum and difference of them). Because we only wish to keep the lower image (the difference), the upper image must be eliminated which is accomplished by having the Quadrature Hybrid. The microcontroller using SPI tunes the PLL/VCO to generate an LO signal 110 MHz below the RF signal frequency the user wishes to scan. Thus, the lower image which is the difference of the RF and LO frequencies will be 110 MHz. To further filter out the other signal that result from the mixer, two low pass filters with cutoff frequencies of 110 MHz were added after the mixer. After passing through the mixer and LPFs, the demodulated 110 MHz signal passes into the variable gain amplifier (VGA). This amplifier's gain is set by the scan job GUI parameters and can be adjusted via SPI through the microcontroller. The VGA was included so that if the RF signal power is really low, it can be amplified in order to raise it above the noise floor and allow for more accurate measurements by the ADC.

All RF components were QFN packages and were chosen by the other RadioHound group of professors and graduate students. Because they had confirmed that their RF front-end performed well, we used their design and chips. See below for a table summarizing information about the parts for this filter system.

Part	Manufacturer	Manufacturer Part #	Price/ piece	Vendor Website
LNA	Qorvo	TQP3M9037	$1.79	http://www.qorvo.com/products/p/TQP3M9037
VGA	Hittite Microwave	HMC681ALP5	$9.90	http://www.digikey.com/product-detail/en/analog-devices-inc/HMC681ALP5ETR/1127-1865-1-ND/4756324
PLL+VCO	Hittite Microwave	HMC833LP6G	$12.57	http://www.digikey.com/product-detail/en/analog-devices-inc/HMC833LP6GE/1127-3034-ND/5360014
Mixer/Demod	Analog Devices	ADL5380	$5.28	http://www.analog.com/en/products/rf-microwave/iq-modulators-demodulators/iq-demodulators/adl5380.html#product-overview
Low-Pass Filter	Mini-Circuits	LFCN-255+	$2.99	http://www.minicircuits.com/MCLStore/ModelPriceDisplay?14720751193400.5870072024389166

Fig 9. Filter System Parts List

3.7 *Analog Circuit & Analog-Digital Converter*

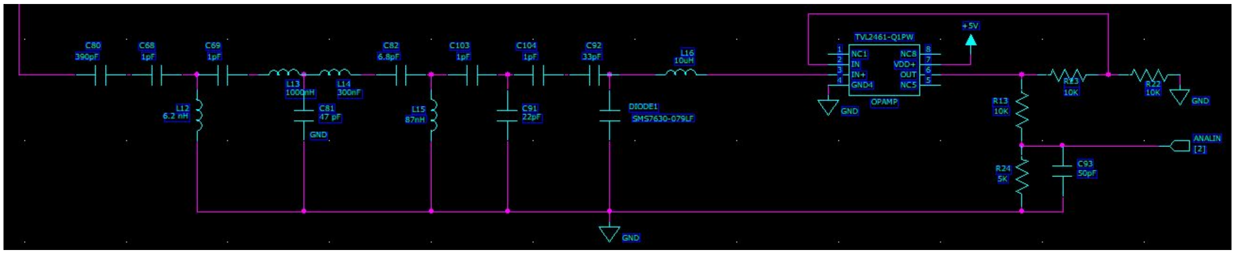

Fig 10 .Schematic for the Analog Circuit & Analog-Digital Converter

The analog circuit takes the amplified output of the filter system and produces a DC voltage proportional to the power in that frequency bin using the square-law range of detector diodes.

The first part of the analog consists of a matching network of passive components designed using ADS to provide maximum power transfer to the analog circuit. This was necessary because the diode has bad S parameters which translates to a large reflection coefficient. The matching network was designed to minimize S11 seen when looking into the circuit. Minimizing S11 means eliminating as much reflection as possible.

After the matching network is a Skyworks diode. The diode is doing the work of taking the 110 MHz signal and producing a proportional DC voltage. This is possible because of the output of a diode is modeled using an exponential function. Exponential functions are actually Taylor series whose first term is a constant. That constant represents the DC voltage produced. Next, the output of the diode is passed through an RF choke which blocks most of the high frequency output but allows the DC voltage to pass.

The DC voltage then passes into a non-inverting op amp. This was included to boost the ADC's low input impedance of 1-2 kOhms. Low input impedance would mean the ADC was only getting a portion of the DC voltage produced, resulting in a less accurate measurement. The op amp also includes a small gain of 2 but uses large resistors in order to keep the current absorbed by the op amp low. The ADC has a max input voltage of 1.8 V so a buffer was needed to protect it. The buffer consisted of a voltage divider circuit with a capacitor to smooth out any ripples seen by the ADC. The divider circuit resistor values were chosen so that the saturation output voltage from the op amp of 5 V would divide so that only 1.5 V was seen by the ADC. The capacitor value was chosen to be much larger than the input capacitance of the ADC of 2-3 pF.

The inductors and capacitors used in the analog circuit matching network and ADC buffer were Vishay Dale and Coilcraft components. These components have small package sizes and low parasitics and ESRs which are necessary for high frequency applications. At high frequencies, passive components acquire parasitics (resistors are no longer just resistors) which affect their performance. Companies like Vishay Dale and Coilcraft try to minimize and account for this. The diode was chosen by our advisor because it is one he has worked with before and is fast enough to rectify a 110 MHz signal. As can be seen below, it is commonly used in these detector type circuits.

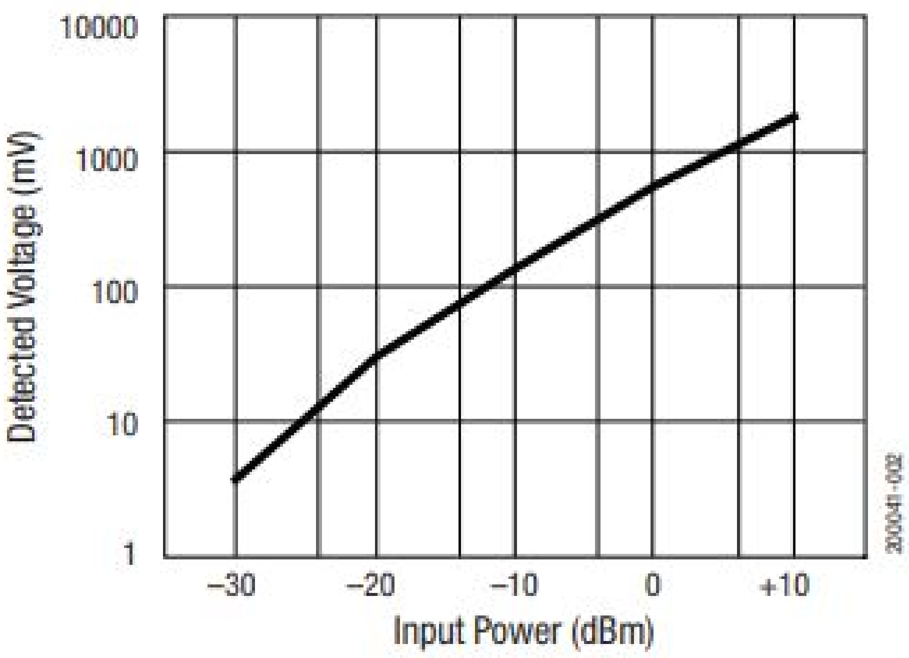

Figure 2. Typical Detector Characteristics @ 1.8 GHz

Fig 11. Diode detector performance from manufacturer datasheet.

The RF choke was chosen to have 5-10 kOhms impedance at 110 MHz and a self-resonance frequency around 110 MHz with low parasitics. A Coilcraft inductor of 2.2 uH met these requirements. The resistors for the op amp and buffer were generic 0603 resistors. The table below summarizes some of the key components.

Part	Manufacturer	Manufacturer Part #	Price/ piece	Vendor Website
Diode	Skyworks	SMS7630-079F	$0.55	http://www.skyworksinc.com/Product/511/SMS7630_Series
RF choke	Coilcraft	0603HL-222XJRB	$1.12	https://www.coilcraft.com/0603hl.cfm
Op amp	Texas Instruments	TLV2461	$2.67	http://www.ti.com/product/TLV2461

Figure 12. List of the Analog Circuit Parts, Prices, and Vendors

3.8 Power Module

We constructed a power subsystem that could use either a battery or USB power, and switch between the two manually. The schematic for that system is below:

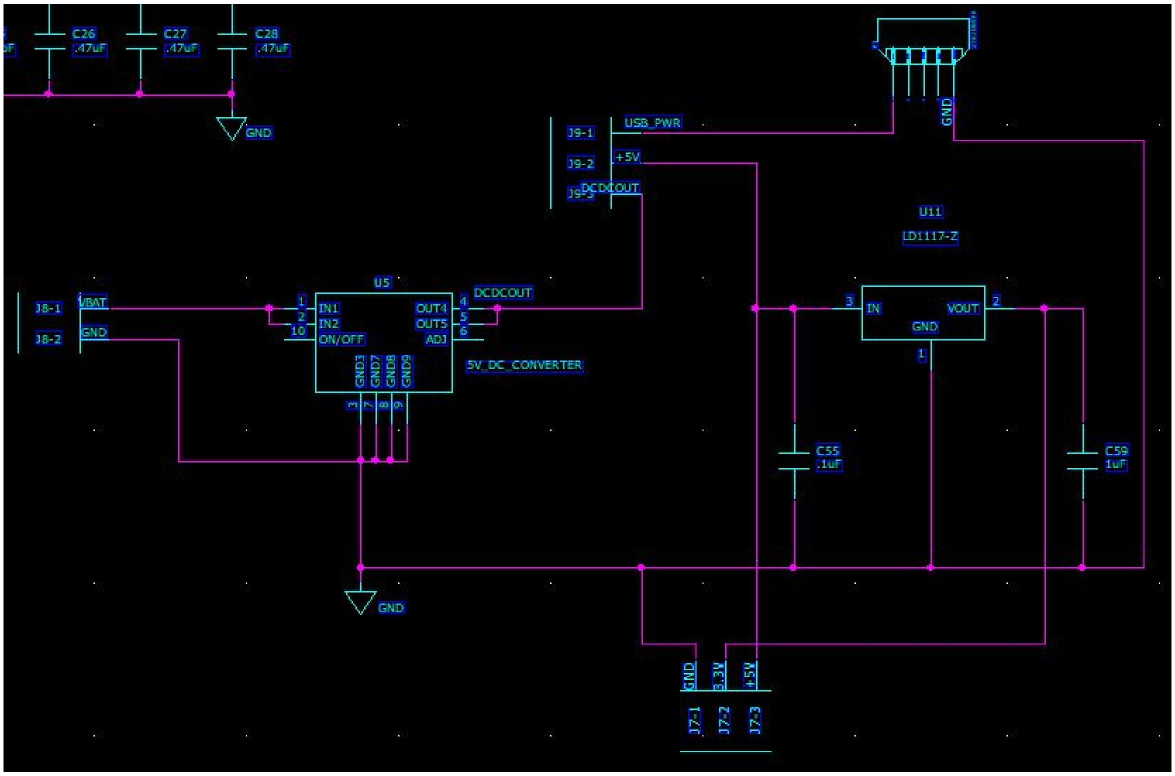

Fig 13. Power System Schematic

In this system, there is a 5V DC-DC converter, specifically the Recom Power R-78AA5.0-1.0SMD-R, that takes in power via MOLEX connections from a 2-cell 7.4V battery. The 5V output of the DC-DC converter goes to a physical switch that can be used to switch between a battery source and the 5V USB source. The switch is connected to a 5V power plane on layer 3 of the board. Also connected to the 5V rail is a linear voltage regulator, the LD1117, that steps down to 3.3V to supply a 3.3V plane, also on the 3rd layer of the board. This power system is able to supply both 5V and 3.3V to all parts on the board that require power.

We switched power sources by using an external switch on the PCB. One power source was the USB, while the other was a LiPo battery. The default setting is battery however the user could plug in the massive battery and flip the switch to use it instead. In choosing the battery, several of the components requires a voltage of 5V while others only need 3.3V. We used a 1000mAh LiPo battery that ran 7.4V, we needed a DC-DC converter circuit to step up the voltage. When using the USB to power the board the 5V DC-DC converter is bypassed and the 5V rail is powered directly from the USB.

4 System Integration Testing

4.1 Subsystems Testing Procedures

Initial testing of the subsystems required several development boards as well as one that we designed and built ourselves. For the power subsystem, we needed to first determine what the power requirements were for the rest of the board; ultimately, we decided that we would need a 5V and a 3.3V rail for our board to operate correctly. The overall power draw will be less than 5W but to allow for some extra headroom we designed the subsystem to operate at a max of 5W. For our power subsystem to be successful, we needed to show that it could handle 5W of sustained power draw from both the 5V and 3.3V rails.

The GPS subsystem used the Adafruit ultimate GPS development board connected to a computer through a USB to FTDI cable. In order to demonstrate that the GPS module was working correctly, we had to demonstrate that we could interface with the GPS chip using our microcontroller over UART protocol. Additionally, we needed to show that we can get a successful GPS lock using our module. To connect the GPS, we used PuTTy with a com serial port with a 9600 Baud Rate.

The analog circuit subsystem presented a lot of challenges for us. Looking back on the project, we would recommend creating a development board of the circuit and tweaking parameters in the matching network to get as accurate a signal as possible. We modelled and designed the analog circuit using ADS software. This would also allow for calibration, mapping the DC voltages to the incident power levels.

To demonstrate that the WiFi module was working, we needed to show that we could send a MQTT packet to the existing RadioHound backend and appear as a selectable node on the RadioHound GUI website.

For the phase lock loop (PLL) subsystem, we needed to be able to first initialize its registers and tune it to output a 250 MHz signal. After initializion, we then had to set the frequency to some other arbitrary number other than 250 MHz. This proved to be difficult mainly because we could only find broken PLL boards and due to the fact that the CC3220SF microcontroller's SPI phase and polarity settings were different than the standard meaning.

The TI CC3220SF microcontroller development boards allowed us to interact with each of the other subsystems individually before we brought them all together on our single PCB. We could also figure out how to program the microcontroller. To show that we had successfully completed this subsystem, we needed to interact with all of these other subsystems so that we can determine what GPIO pins must go to what subsystems.

5 Users Manual/Installation manual

5.1 *How to install and setup your product*

Make sure you have a program installed to interface with serial data, we recommend using PuTTY. Make sure the WiFi antenna is attached to the SMA connector. Make sure the telescoping RF antenna is attached to the SMA connector. Choose your power source using the switch by either plugging into wall powered USB source or attaching LiPo battery. Connect UART1 on edge of board to computer to view the board system log and open the serial data interface with the correct computer port number (found using Device Manager on computer) and these parameters:

Baud rate: 115200

8 Bit Data 1 Stop Bit

No parity and no flow control

First, power up the board by choosing the desired power source using the onboard switch. Wait for the board to ask for a WiFi network to connect to via the serial port and give it the required information asked for. Then go to the RadioHound GUI website: http://fmnc.cse.nd.edu:9000/ using a web browser. After seeing your board appear on the map as a node, select it from the map and send it the desired scan job parameters following the GUI format. Wait for the board to finish the scan job and return the voltage detected array to the server. A display will pop up on the website displaying the returned data.

5.2 How the user can tell if the product is working

The board is functioning properly if it can connect to the given WiFi, appear on the RadioHound GUI, receive scan job parameters, and return a non-zero array of detected voltages for that frequency range.

5.3 How the user can troubleshoot the product

Check if GPS has found a lock (a red LED by the GPS should be lit up constantly if it's found a lock). If not, it might be having trouble finding the satellites so try walking outside with the board still powered up. Make sure the board was able to connect to the WiFi. If not, try resetting board by pressing the reset pushbutton, it should then ask again. Connect your RadioHound board to a serial terminal using the labelled TX/RX pins and a MAX232 Serial to TTY USB converter in order to view the output of the node's startup process. If necessary, select an open WiFi network using the serial interface, and the board will reset using this network as the default. Repeat this process until it is able to connect.

6 To-Market Design Changes

The Wireless Institute in collaboration with Army Research labs have specified the goals for getting this product to market. To reach those goals, the device must cost less than $70 and be able to scan between 0.025 (25 MHz) and 6 GHz. Although we took large steps towards those goals with our project, there is still work to be done to reach those basic goals. To extend the range, you would need different antennas to reach below 400 MHz, and a MUX to select which antenna to use depending on the requested frequency range. This is certainly possible with the current setup, but it would cost more to add extra antennas. To reduce the overall cost to less than $70, some of the commodity RF ICs and other necessary circuit components but will decrease in price when you buy them in bulk. The board itself also would be much cheaper if bought in bulk. One big cost for this board is the power supply, which certainly could be further reduced with a different battery or DC-DC converter. With those changes, it would certainly be under $100 and very close to the price goal.

Once the frequency range and price goals are met, other steps would have to be taken before shipping thousands of these boards off to war zones. First of all, calibration and noise testing needs to be done to translate the DC voltages the microcontroller measures into actual power levels that are consistent from board to board. Without board-to-board consistency, the system will not work because it relies on plotting relative power levels. Lastly, for implementation, we would need to take away unnecessary board headers, shrink the board, and enclose it with a tactical, field-ready case. Ideally, with those changes, the product would be field-ready.

7 Conclusions

In conclusion, the results of this year's work can be considered a success. Although we were unable to meet all of the initial requirements for the project, we significantly expanded on the RadioHound project, especially in terms of scalability, power, and price. For example, we were able to eliminate the Raspberry Pi and SDR-RTL dongle but we still used the existing web-based interface. This made it possible to see our new data on the current software. Further, we are confident that our work serves as a proof of concept and knowledge of our progress should inspire future work. We found that this basic approach is feasible, and a subsequent design revision and calibration should function correctly. Nevertheless, there is much work to be done before our prototype is ready.

Considering future projects, one of the aspects of the project that we never put under the microscope was selecting the best microcontroller. We chose the CC3220SF for reasons that we no longer value as much as we initially anticipated. Specifically, the low-power processing combined with onboard wireless connectivity was a major consideration. However, the programming, flashing, and PCB design process required for this microcontroller may outweigh its benefits in the long run for project development and support. For example, there are software driver issues with the size of the MQTT payload able to be sent, and difficulties in programming the board using only JTAG connectors. The flashing process also takes a significant amount of time and is necessary for each minor revision of the software. We believe the same level of performance could be achieved with a more accessible processor and wireless chip combination, such as a PIC32 and an ESP8266, so future teams should plan according to what they are most comfortable with.

The components of the system, although they were not functional on one PCB by the time of the demo, successfully worked when hooked with jumper cables from board to board. We have since identified all of the visible issues with the ordered board version and have remedied said issues on the full schematic. With only one version to build the completed system, building our second design in the future would be helpful to test and see if any issues remain that were masked by the ones already solved.

8 Appendices

Please see the senior design website under the Documentation tab for hardware schematics, software code, and relevant data sheets. This website can be found at: http://seniordesign.ee.nd.edu/2018/Design%20Teams/rf/index.html