Portable Field Radio Based on LIME Software-Defined Radio Receiver

Brian Taylor (EE), Elier Bermudes (CPE), Noah Madison (EE), Daniel Sypioe (CPE)

Dept. of Electrical Engineering and Computer Science, the University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — This conference paper elaborates on the methodology and design implemented in order to develop a portable software-defined field radio based on the LimeSDR, an embedded system, as well as any necessary components that are required to complete a functional prototype. This project is funded up to $500 by the Amature Radio Club in hopes that we can not only develop the system but create one that is easily accessible and upgradeable by the average entry-level Amature Radio enthusiast without the use of specialized components or software. We hope to not only exceed the expectations of our sponsors and advisors with the final prototype but to create an affordable device that will encourage generations of engineers of all backgrounds to enter the field of Amature Radio.**

***Index Terms* — Radio Frequency (RF), LimeSDR, Portable Radio, Rx/Tx Transmission, Panadapter**

I. Introduction

During the start of Senior Design I, we were selected to do a sponsored project by the University of Central Florida Amateur Radio Club. The project plans they had laid out for us described a portable, modular, and open-source software-defined radio. The components we would use for the project were not important, however, there was a preference for the LimeSDR mini since the club is already in possession of a LimeSDR. Aside from the required components, the system needed to meet certain goals by the end of the prototyping phase so that the Amateur Radio Club could use and develop the system further.

In order to achieve such a task, we aim to design a software-defined radio based on the Raspberry Pi and Lime SDR mini, alongside complementary components which cost no more than $500 to produce. The device will feature a speaker, microphone, and push-to-talk button which will enable the consumer to listen to their favorite AM/FM stations as well as communicate with others on FCC-regulated Amateur wavelengths.

The system will also feature a touch screen to visually display the necessary functions and features of the radio. This will include separate pages to access regular listening, communications, a waterfall display, a settings page, as well as a signal bar on the right-hand corner of the screen. In the final stages of prototyping, measures will be taken to make the product water-resistant to sweat and mild water droplets allowing the device to be taken outdoors without risk of damage due to weather or accidents.

The most critical aspect of our device is the combination of our embedded system, radio receiver, and amplifier. The Raspberry PI and Lime SDR must be integrated seamlessly and in a way that is computationally efficient in order to reduce any disruptions to service. The amplifier on the other hand must be selected carefully since a number of factors can affect its performance such as the input/output impedance, heat dissipation, and the type of filter it uses.

II. System Overview

The following figure provides a generalized diagram of all of the features of the device and how they should interact at a hardware level. The embedded system is further elaborated on in the upcoming sections. The battery control circuit, although necessary for the functionality of the system, will not be a part of the actual handheld and will be external to the system. This diagram in particular is for a high abstraction explanation of different features that will exist and how they are to interact with other project features and systems. It is not meant to be an exhaustive description of every component, but more of a summary. Provided below the figure is the legend which outlines the division of labor for this project in all stages including research, design, integration, prototyping, and development.

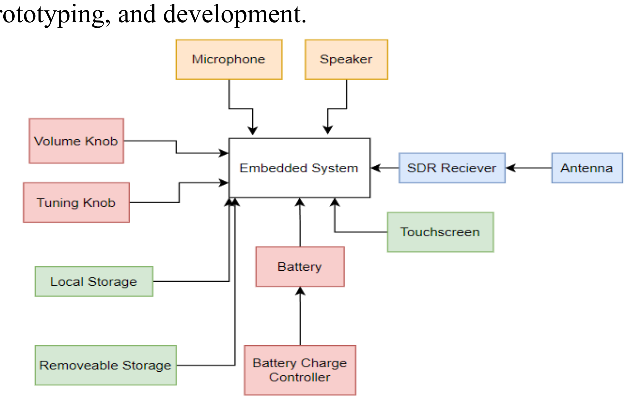


Figure 1: System Block Diagram

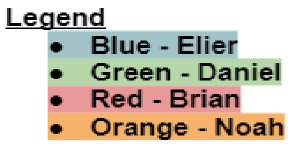


Figure 2: Work Distribution Legend

III. Project Plan

Since Senior Design I and II are very labor-intensive classes and many of us in our group hold jobs and responsibilities outside of education and are taking other classes; we needed to devise a plan of success to make sure we meet the goals of our sponsors/advisors as well as to ensure we turn in our work on time. In order to plan for such a large-scale project, we adapted the timeline of submissions for Senior Design I and II by incorporating several decision-making and workflow processes outlined by organizations such as the IEEE.

IV. Component Selection

In order to design and construct our system, we had to undergo extensive research for part selection as our limited budget of $500 ensured that we could not buy multiple units of major components such as the radio receiver, embedded system, and RF components.

A. Radio Receiver

We had originally designed our system based on the Lime SDR mini, but due to the global component shortage, all manufacturing of the part was placed on hiatus. Instead of choosing a different radio receiver, we were able to use the full Lime SDR provided by the Amateur Radio Club. Due to this, we will have an increased throughput on core radio functionality at the cost of resources to process radio communications.

B. Embedded System

We are currently designing our system on the Raspberry Pi model 3B+ since our original plan of using the Raspberry Pi Zero faltered when we needed to upgrade to the full-sized Lime SDR. Despite this, during prototyping, we encountered bottleneck issues while developing the software and using multiple programs at the same time. This proves and will continue to be an issue until the Lime manufacturer releases the Lime SDR mini at a reasonable price again. To alleviate this predicament, we have implemented multiprocessing to utilize the full power of the Raspberry Pi model 3B+ instead of once again upgrading the model of the embedded system.

C. Touchscreen

For the touchscreen, we decided that the use of the GPIO pins of the Raspberry Pi was more essential than having the best quality screen. Since we are also working on a limited budget, we were only able to budget $30 towards the touchscreen and opted for one of the only highly rated options that featured a touch and video connection via USB and HDMI, respectively. It is advertised to have a refresh rate of 60Hz, resolution of 480x320px, power consumption of 130mA/5V, and a resistive touchscreen.

D. Antenna

Our easiest component to choose was the antenna since they are very simple devices. The six criteria the antenna met in order to be a candidate for the system were: power, price, frequency, VSWR, durability, and size. Our target audience is a beginner to intermediate Ameature Radio enthusiasts and students so instead of opting for the most feature-dense antenna, we will provide a ¼ wave whip antenna, although it is noted that a multiband antenna will increase the performance of the system.

E. Amplifier

Intermediate amplification is necessary since the output of the Lime SDR is meager, at only about 15mW, and has the potential of not meeting the minimum input power for the 5-watt power amplifier for the system. Additionally, if time permits, we could implement our stretch goal of adding variable gain functionality since the Lime SDR does not natively support it.

F. RX Conditioning

RX conditioning for our system is set to a Half Duplex mode in order to reduce the necessity of using large and inefficient filters for each band to get the Full Duplex system operational. Despite our efforts to mitigate complexity and meet our goals for project completion, we are still faced with the challenge of handling the high voltage peaks of the system which have the possibility of exceeding 40V PkPk. To have a fully operational and efficient system, the design plan notes that the addition of RF limiters, circulators, duplexers, and T/R switches will be beneficial. Currently plans to implement DSP in the second iteration of the device are underway as we continuously develop software for the system.

G. Battery Cells

Our selection for battery cells was an easy one given modern battery technology. Although pricy due to inflation, the 18650 Lithium-Ion Battery Cells were the optimal choice. With the power delivery of 20A at 3.7V nominal, two of the battery cells are able to power all of our components for an estimated battery life of at least three hours. With upgraded 18650 Lithium-Ion cells, the battery life of the system can easily exceed the desired measurement of a five-hour standby time.

H. Speaker Subsystem

We had many form factors to choose from when it came to selecting our audio system, but ultimately the decision came down to what would be seamlessly incorporated into the final system. We have chosen a kit from Amazon that provided us with a PCB containing a stereo class D audio amplifier, a volume knob, and necessary wires and headers to connect the speakers, power, and a 3.5mm audio jack so as to not take up any potential GPIO space.

Microphone

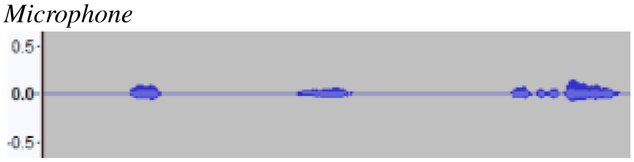


Figure 3: Audio Input Recording From Laptop Microphone

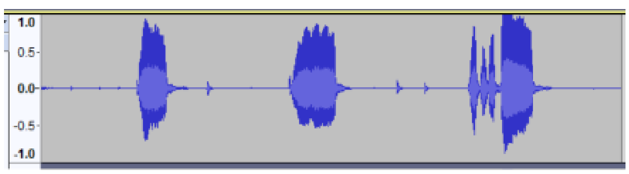


Figure 4: Audio Input Recording From USB Microphone

The microphone of the system followed the same rule as most of the other components, which was to minimize the usage of the GPIO pins on the Raspberry Pi. We decided to use an ultra-slim USB microphone receiver in order to handle incoming audio since it fit our rule and was also cheaper than most other options with different form factors. The two figures above show the audio sensitivity and quality measurements of a conventional laptop microphone (top picture) vs the measurements recorded from using a USB microphone connected to the Raspberry Pi (bottom picture).

Low-Level Software

We regard the low-level software of the system as the operating system and any firmware needed in order to integrate our components. Since we originally planned to develop our system based on the Raspberry Pi Zero, we have selected Raspberry Pi Os Lite as our core operating system and will upgrade to the full version only if it is absolutely necessary for final production.

High-Level Software

The system's high-level software consists of all the programming languages and add-ons we plan to use during the software development stage of prototyping. Since our project is designed to be as open source as possible, we have selected Python and C++ as the primary programming languages since they have prevalent documentation and a dense community base users can rely on. Since we plan to develop and optimize our system until we run out of time so as to create the best product we can, our list of high-level software is incomplete and will remain so; so that the Amateur Radio Club can edit the document at their discretion with whatever they choose to integrate to the system.

V. Block Diagram

Our systems' hardware aspect (depicted below) is a relatively simple one, as the components are standard for what you would need to integrate in order to make a portable computer out of the Raspberry Pi. Integrating the radio receiver and RF components was also a simple task since we underwent the necessary research and preparations. As we will discuss later in the document, though, many of the challenges we faced were after the integration phase and occurred in the software development phase.

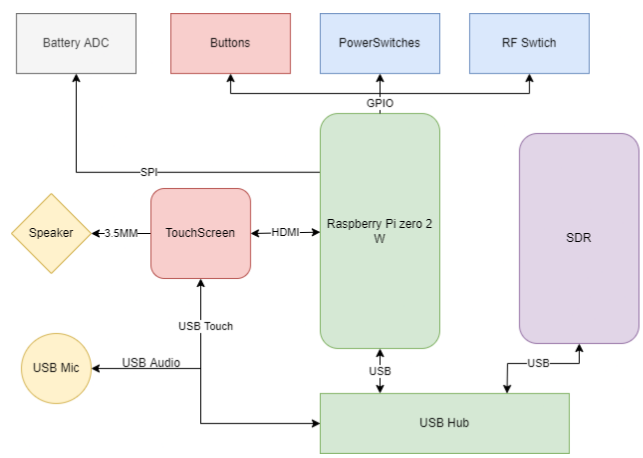


Figure 5: Hardware System Block Diagram

VI. Battery Management Subsystem

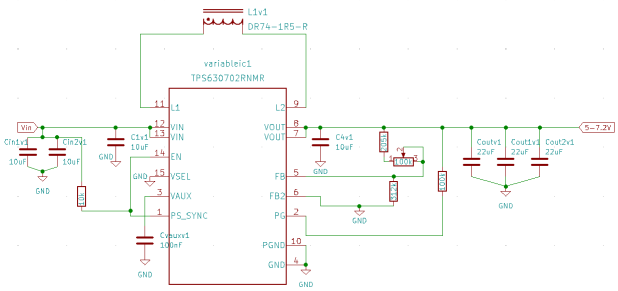


Figure 6: Variable Output Boost-Buck Converter

Our power subsystem provides energy to every subsystem of our software-defined radio. The voltages and currents were determined during our part requirements and we found that the cylindrical form factor 18650 Lithium_ion battery cells were the best candidate for the task. The power entering the amplification subsystem is controlled by two dual MOSFET modules connected to the Raspberry Pi. We used these MOSFETs as a switch allowing the GPIO on the Raspberry Pi to turn on or off voltages on different parts of the RF subsystem. We have selected a rather efficient model of generic MOSFET in which the on-resistance at our target current draw will provide a lower power dissipation and therefore a low power drop of fewer than 0.01 Volts compared to other solutions.

In order to monitor the battery subsystem's overall voltage, we used an SPI ADC connected to the Raspberry Pi which allowed for a low complexity solution to measuring the battery cells and charge control. We also avoided unnecessary complexities when designing our charging subsystem by making the circuit compatible with a hot-swappable set up so that the battery cells could be charged outside of the actual system, and the case in which they are contained will be directly able to be charged with a USB-C cable for simplicity and accessibility.

VII. Antenna Subsystem

Despite the antenna being our simplest component in terms of complexity, it still had to undergo a rigorous testing procedure to ensure the one we have selected performs as advertised and will meet the needs and requirements of our system. We tested the antenna inside the Radio Room at the University of Central Florida in the back of the Senior Design Lab. A nano VCA will be used for VSWR and an RF power meter will be used to test the output power. An ICOM radio with controllable output power was also used for power and transmission testing.

Listed below is the specific procedure we used to test our antenna:

1. The antenna will be tested with a portable RF meter to see transmitted vs absorbed power
2. The antenna will be tested with a CNA to see if it meets the impedance and VSWR characteristics.
3. The Antenna will have 5 Watts of RF power inserted into the system at each band separately to ensure that we are properly dissipating heat radiation.

VIII. RF Subsystem

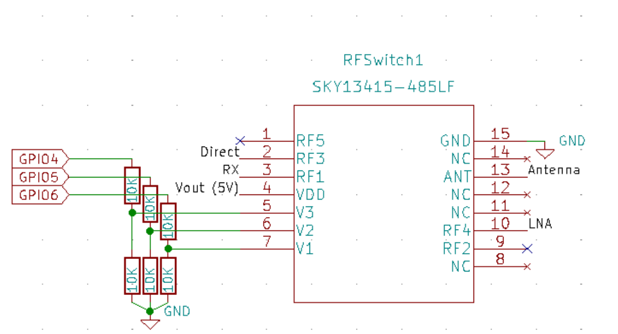


Figure 7: RF Switch Schematic

For our RF amplification control circuit, we used a combination of two methodologies. Firstly, the actual power supplied to the amplifiers could be disabled in order to allow us to make the system more efficient by saving power. If this is not the best case at the given time, the other option the circuit provides is switching the power draw from the lower RF section directly to the antenna subsystem. By doing this, we are able to have a high output power when it comes to protecting the software-defined radio.

During testing, we tested each external amplifier with an oscilloscope so that we can verify that the noise sidebands and EMI generated and received by the system are within an acceptable range for quality assurance. Each component of the RF system must also be tested to ensure that they meet the minimum requirements, standards, and restraints provided by the FCC, our advisory board, our sponsors, and ourselves.

IX. Radio Receiver Subsystem

The chosen software-defined radio receiver that we selected for our system is the Lime SDR. It plays a critical role within the system itself, as our RF capabilities are completely limited by the capabilities of our SDR receiver. As mentioned before, the sponsor preferred the LIME SDR mini for the project but we were not able to get one within our price range and confirmed with both the sponsors and the Senior Design Advisory board that it was okay to proceed with the full-sized LIME SDR. This did prove to be beneficial during the initial prototyping and development because the full-sized LIME SDR has four channels as opposed to only two on the mini. Not only that, it comes with an enhanced FPGA, a major selling point of the board, and will be able to handle the significant amounts of digital signal processing the system will undergo during testing.

Some of the things that will benefit our system that is available from the LIME suite of SDRs are that they have the inclusion of SMA antenna connectors rather than micro U.FL antenna connectors, a frequency range of 100 kHz to 3.8 GHz along separate channels for TX and RX, multiple matching networks connected to the RF transceiver. 256 MB of DDR2 SDRAM, and a bandwidth of 61.44 MHz. The addition of SMA antenna connectors allows us to use hot-swappable antennas and amplifiers without the need for an adapter. Most importantly though, the separate connectors for TX and RX allow for the possibility of a Duplex operation and significantly easier T/R switching for Half-Duplex operations.

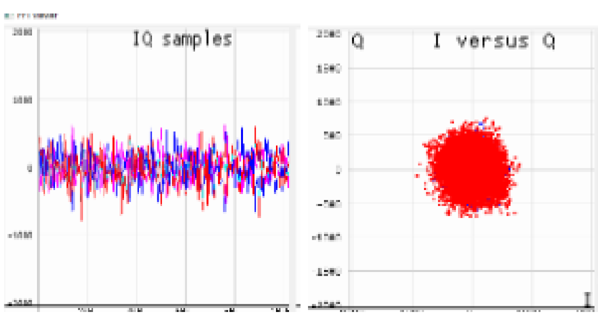


Figure 8: SDR IQ Samples (Left) I vs Q Plot (Right)

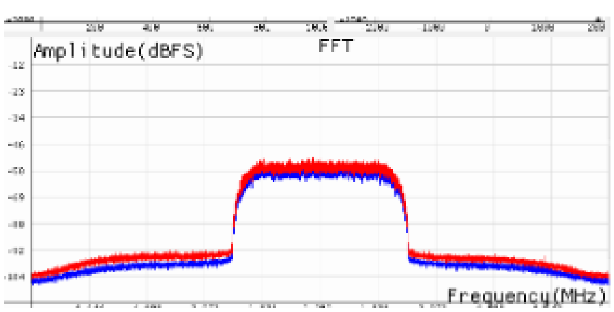


Figure 9: Lime SDR Connection to Raspberry Pi and Loopback Test FFT Graph

The grouping of figures above are graphs of measurements obtained during our prototyping and development phase of building the system. Our procedure to test the connection between the Lime SDR and the Raspberry PI to verify that the connection was clear and that communications between the two boards were stable with functionality in different modes is as follows:

1. The LIME SDR will be connected and the supporting firmware will be upgraded using the LIMESuite of tools.
2. The given configuration file will be loaded into setting the RF switched to the loop-back to the software-defined radio receiver.
3. The given play file is transmitted through loopback and an FFT is measured via mathematic software such as MATLAB or Python.

Before we tested on the Raspberry Pi, however, we did try to establish a connection to the LIME SDR via a desktop PC using the Windows 11 operating system. Our findings were that the LIME SDR was unable to connect to a system operating on Windows 11 due to driver compatibility issues. Since this version of the operating system is relatively new, this was one of our expected outcomes.

Instead of trying to establish this connection in order to develop the software in a high-performance environment, we decided that it was not within the scope of the project for these two sessions of Senior Design and that we should continue on to the Raspberry Pi since Windows compatibility can be developed by our sponsors or by another Senior Design Group. Based on our results we assured that the LIME SDR and Raspberry Pi were both fully functional and that both boards operated as advertised within a controlled environment.

X. Processing Subsystem

In order to process all of our commands and communicate with all of our major components, we selected Raspberry Pi as our microcontroller. On the Raspberry Pi, we need to use a multitude of varying communication protocols such as i2c, SPI, and serial communications. This will encompass all connections including the USB bus, the interface between the LIME SDR and the Raspberry Pi, as well as all GPIO connections.

Our digital subsystem works with four communication types, excluding the internal communication of the SDR which may have to be interfaced with later on based on how far we are able to get before delivering the system to the Amateur Radio club. Depicted below is a table representing the different states the digital subsystem can be in depending on the logic level given as input.

V1	V2	V3	State
0	0	0	External SMA
0	0	1	Direct TX connect
0	1	0	LNA
0	1	1	RX Mode
1	0	0	All Isolation/Test Mode
1	0	1	50 Ohm Sync
1	1	1	Shutdown mode

Table 1: Digital Subsystem State Levels

The first and main communication will be via UXB. With our USB hub, we are able to access the microphone, touchscreen, and LIME SDR with the Raspberry Pi. The second communication type will be over the Raspberry Pi's GPIO block and will by far be the simplest to set up. The GPIO pins will be set to only operate in one direction unless stated otherwise. Our third protocol will be HDMI, which will handle sending all video signals to the screen and the audio signal to the speakers. Our final communication type will be SPI which will be used to connect and monitor our Audio to Digitial Converter. This will monitor various functions including but not limited to: the volume, gain adjustment, output power, and battery level indicators. The battery monitor requires a voltage divider circuit in order to shift the levels to within the ADC's chosen range.

XI. Software Detail

As this project was to construct a software-defined radio, there are many interconnected components of the software and hardware that must be seamlessly integrated in order for radio telecommunications to be clear and accurate. Aside from perfecting the radio system entirely, the process we underwent to develop the system was to

1. Install Raspberry Pi Os/Lite and upgrade/update the system before use
2. Install the touchscreen and all complementary components. Update all pertinent firmware/drivers.
3. Establish a connection between Raspberry Pi and Lime SDR
4. Update Firmware and Drivers for the LIME SDR
5. Develop software for Audio I/O
6. Establish communications on LIME SDR
7. Develop GUI and integrate physical control into embedded system
8. Create a waterfall display if time permits

This can be shown in the diagram below.

An issue that we consistently ran into during development was that the Raspberry Pi model that we were using was not able to handle multiple programs open at the same time as well as run the code for testing. Routinely the system would either overheat or shut off despite being in an ambient room temperature and heat sinks installed on all high-power components. We could do nothing to solve this other than upgrading the embedded system to the most powerful Raspberry Pi available on the market, but this would lead to both size, placement, and power challenges.

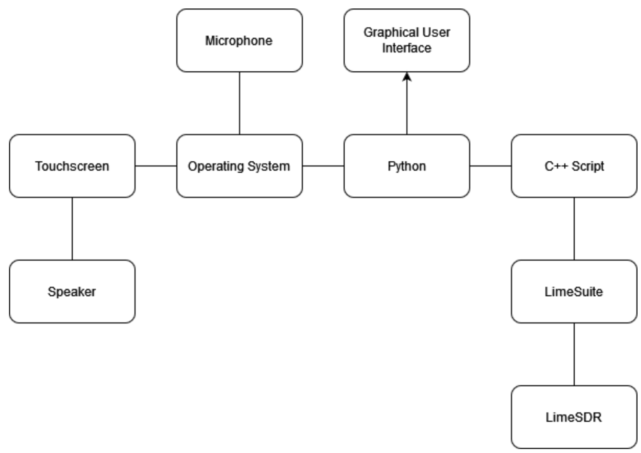


Figure 10: Software Flow Chart

XII. PCB Design

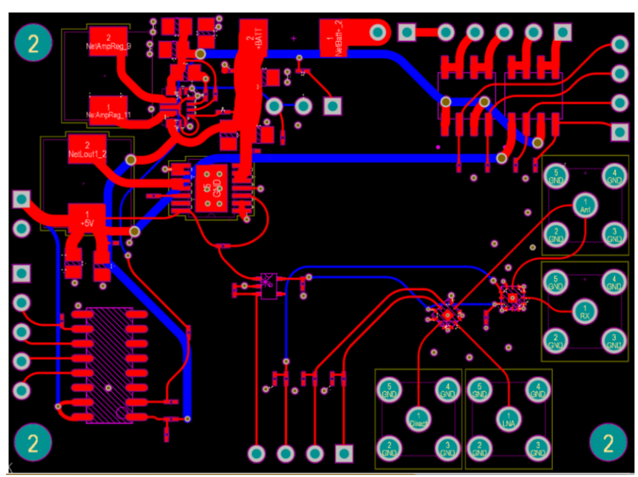


Figure 11: Printed Circuit Board Schematic

Designing our PCB was challenging yet we were able to accomplish a working prototype with our first print and design. While researching the properties our board would have for instance single or double layer, we found that RF components were more sensitive than we anticipated and that a specialty PBB board would have to be etched.

While dealing with RF components, we needed to use a high level of production, tools, and manufacturing since the components are very sensitive. All components in the RF category must be impedance matched (we chose 50-ohms as our desired impedance) so that we don't produce any undesired noise or disrupt our frequencies. We also prevent EMI by using a shared ground plane across all noise planes. Our simplest fix, however, was to place our switching regulators and inductors at the furthest point away from the RF section.

VII. Conclusion

At the point of writing this document, we have succeeded in creating a portable system concept equipped with Lithium-Ion batteries with a charging circuit, an embedded system for radio receiver communications, simple TX/RX transmissions, and a fully functional touchscreen with audio I/O support. We are currently working on finalizing the development of our transmission protocols as well as constructing a GUI capable of being updated by our sponsors once the system is delivered.Despite the numerous challenges we faced along the way ranging from trying different methods of remote software development, complications with component testing, and getting RX/TX transmission set up; we were able to overcome most of our obstacles and what we could not we improvised and allowed for slight changes in the overall requirements when new information was presented.

As we optimize our system for the final presentation and hand it off to the Amateur Radio Club of UCF, we must take into consideration the stretch goals of the project and outline how we had planned to continue so that they can finish the system either themselves or with the aid of another Senior Design group. Some of the features that we were not able to complete due to the time restrictions of a short Senior Design II semester in the summer were: waterproofing, final heat dissipation, and a portable enclosure for the device.

Once the market for LIME SDR mini's or similar boards is stable again, a portable casing could be designed and constructed. After this, waterproofing could be done in various ways. A water-tight seal on the system as well as a waterproof, audio-permeable material for the speaker grill will be optimal. For final heat dissipation, many routes can be taken to ensure that heat and radiation are maintained within the FCC limit of electromagnetic energy absorbed by the body.

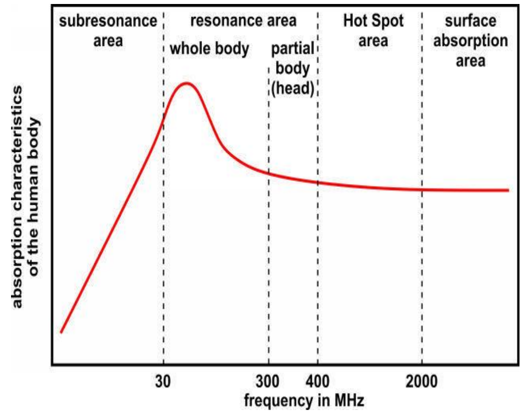


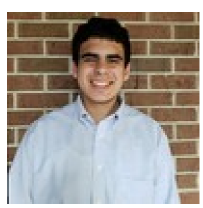
Figure 12: Heat Characteristics

This measurement is called the SAR and must be determined in instances where the device is 0.2 m or less in contact with the body and cannot exceed 1.6 watts per kilogram of tissue. In terms of temperature measurement, the device must stay between -4 and 113 degrees Fahrenheit. All of these specifications are requirements amongst devices of this nature as the system has the capacity to burn the user or components with the lack of proper heat dissipation.

Acknowledgment

The authors of this conference report gratefully and respectfully acknowledge the University of Central Florida Amateur Radio Club for their contributions in financially supporting the project as well as facilitating research and providing necessary guidance.

VII. Meet the Engineers



Brian Taylor is a 23-year- old Electrical Engineering student who is employed as an engineering intern at JBT Aerotech. Once he graduates, Brain hopes to work in PLCs.

Elier Bermudez is a 22-year-old graduating Computer Engineering student. Elier is taking a job with I-CON Systems Inc. as an Embedded Developer while pursuing his Master's Degree in Computer Engineering at UCF.

Noah Madison is a 22-year-old Electrical Engineering student in the comprehensive track. Noah has accepted an RF systems engineering position at Indyne/RTT in pax river Maryland.

Daniel Sypioe is a 24-year-old Computer Engineering student. Daniel is the CEO of an ML firm and was previously a contractor for various venture capitalists as well as the Director of Engineering at Community Tech, LLC.

References

[4] "Balun One Nine v1 - Tiny Low-Cost 1:9 HF Antenna Balun and Unun with Antenna Input Protection." *Amazon.com*, https://www.amazon.com/NooElec-Balun-One-Nine-Applications/dp/B00R09WHT6/ref=psdc_667846011_t3_B01N2NJSGV. Accessed 9 April 2022.

[8] "CEA 861-F-2013 (ANSI) - A DTV Profile for Uncompressed High Speed Digital Interfaces." *ANSI Webstore*, https://webstore.ansi.org/standards/cea/cea8612013ansi. Accessed 9 April 2022.

[12] "Design and Analysis of a Continuously Tunable Low Noise Amplifier for Software Defined Radio." https://www.researchgate.net/publication/331735915_Design_and_Analysis_of_a_Continuously_Tunable_Low_Noise_Amplifier_for_Software_Defined_Radio.

[15] "DSP-Driven Self-Tuning of RF Circuits for Process-Induced Performance Variability." DSP-Driven Self-Tuning of RF Circuits for Process-Induced Performance Variability.

[26] Infineon. "OPTIREG™ switcher TLS4125D0EPV50." *Infineon Technologies*, 5 February 2021, https://www.infineon.com/dgdl/Infineon-TLS4125D0EP%20V50-DataSheet-v01_00-EN.pdf?fileId=5546d46270c4f93e01710b608b523b33. Accessed 25 April 2022.

[29] "KISEER USB 2.0 Mini Microphone." https://www.amazon.com/KISEER-Microphone-Desktop-Recording-YouTube/dp/B071WH7FC6/.[29] "KISEER USB 2.0 Mini Microphone." https://www.amazon.com/KISEER-Microphone-Desktop-Recording-YouTube/dp/B071WH7FC6/.

[28] "JniTyOpt 3.5 inch Display." https://www.amazon.com/JniTyOpt-Raspberry-Interface-Resolution-Supports/dp/B0931Z7MX5/.

[30] "LimeSDR vs LimeSDR Mini." *Lime Microsystems*, https://limemicro.com/products/boards/limesdr-mini/. Accessed 26 April 2022.

[31] "LoveRPi 4 Port MicroUSB to USB Hub." *Amazon.com*, https://www.amazon.com/LoveRPi-MicroUSB-Port-Black-Raspberry/dp/B01HYJLZH6/. Accessed 9 April 2022.

[43] "Raspberry PI OS Lite vs Desktop: comparison between the 2 distributions." *peppe8o*, https://peppe8o.com/raspberry-pi-os-lite-vs-desktop/. Accessed 26 April 2022.

[44] "Raspberry Pi Zero 2 W – Raspberry Pi." *RaspberryPi.com*, https://www.raspberrypi.com/products/raspberry-pi-zero-2-w/. Accessed 26 April 2022.

[45] "RF Amplifier Ultra Wideband Gain 20dB Medium Power Amplifier Board 5M-6GHz." https://ebay.to/3CLVugK.

[46] "RF Power Amplifier 20M-512MHz 5W Linear Amplifier for FM Radio Remote Interphone." *eBay*, https://www.ebay.com/itm/174362238519. Accessed 9 April 2022.

[47] "RS-UVPA 5W UHF/VHF Power Amp." *HobbyPCB*, https://www.hobbypcb.com/index.php/products/uhf-vhf-radio/rs-uvpa. Accessed 9 April 2022.

[51]Texas Instruments. "LM2621Low Input Voltage,Step-UpDC-DC Converter." November 2015, https://rocelec.widen.net/view/pdf/y3iuum7fts/snvs033d.pdf?t.download=true&u=5oefqw. Accessed 10th April 2022.

[52]Texas Instruments. "TPS63070 2-V to 16-V Buck-Boost Converter With 3.6-A Switch Current." *ti.com*, Texas Instruments, March 2019, https://www.ti.com/lit/ds/symlink/tps63070.pdf?ts=1650925377945&ref_url=https%253A%252F%252Fwww.google.com%252F. Accessed 21st April 2022.

[60]"Used Lithium-Ion Batteries | US EPA." *US Environmental Protection Agency*, https://www.epa.gov/recycle/used-lithium-ion-batteries. Accessed 9 April 2022.

[61]"Vega Barebones - Ultra Low-Noise VGA Module for RF & SDR." https://www.amazon.com/Vega-Barebones-Low-Noise-30MHz-4000MHz-Capability/dp/B08LNYKHSM/.