CSU Sacramento
Department of Engineering

Class : CPE 191 and EEE 193B
Team 2

End of Project Documentation

Modulation Classification - Emergency Services

Authors:
Nicholas Bergamini
Cory Diehl
Yuriy Dzyuba
Alexander Han

Professor:
Dr. Tatro

Date Submitted: April 29, 2019

SACRAMENTO
STATE

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

Executive Summary

In the event of an emergency situation such as a natural disaster, life threatening event, or any other hazardous and/or dangerous situation, effective communication is imperative for first responders and emergency rescue teams to mitigate damage and help provide relief. Communication within a single emergency team is relatively straightforward due to compatible communication technology. It is when multiple teams from different districts and levels of government find themselves having to collaborate that the issue of impaired interoperability arises. Our system serves to resolve this precise issue of deficient interoperability among emergency response teams and individuals.

To create an effective system that serves as a solution to the problem, we had to analyze the problem at its current state. We found that modern wireless radio technology employs software defined radio. This enables easy reconfigurability of radio devices even in the field. Emergency teams have their own standards and operate on predetermined channels and with predetermined modulation schemes. Teams, therefore, have thick handbooks that provide information pertaining to existing emergency teams and their respective configuration (frequency channels and modulation schemes) for the communication devices. This seemed to us as something that can be automated and enhanced. We designed our system to build off of the implementation of software defined radio in today's communication devices. Effectively we used software defined radio, DSP, and computer networking to remove the handbook and provide quick and easy access to the information that the handbook now provides.

Using the timeline we are able to track our progress and make adjustments where necessary. The timeline was a flexible document that was designed to accommodate changes and unforeseen setbacks. Coupled with the timeline was our work breakdown structure (WBS). This document was used to allocate tasks to each team member. The work breakdown structure was instrumental in helping us see the big picture of our system and how each individual module interacted with the other modules. The tasks in the WBS was superimposed onto the timeline. When creating these documents, our team operated by the concept of prepared spontaneity. This means that while we did not know what changes we may have to make, setbacks we will have, or pitfalls that await us, we created our documents to accommodate these. We also analyzed our market in order for us to tailor our design and presentation of our design. During the course of the market review, we analyzed the needs of our consumer, took a look at our competitors, and highlighted the uniqueness of our product. We performed periodic testing throughout the duration of the one year project. This was to ensure that we are on the right track and used to make necessary adjustments. The unit testing made integration of our system tremendously easier than if we had done without the testing. Ultimately, we created a deployable prototype that serves as a proof of concept of a full scale, commercialized system.

Abstract

This document outlines the problem that emergency response agencies experience with wireless communication interoperability when having to collaborate with other agencies at the scene of a natural disaster or other potentially dangerous situation. Detailed analysis is provided concerning the approach that our senior design team selected in addressing this issue. This analysis includes our design philosophy, technology used, and justification for the use of said technology. Our system aims to perform spectrum sensing in order to collect radio frequency signals in a emergency affected area. Upon receiving the signal, mixing it down to baseband, and digitizing it, the digitized signal is processed using algorithms for frequency compensation and symbol timing synchronization. After these signal processing algorithms are applied to the signal, the signal is classified by its modulation scheme. The carrier frequency,

bandwidth, and modulation scheme of the particular signal are all displayed on a dynamic user web page. At the heart of the system is a software-defined radio. The software-defined radio serves to enable signal processing that was traditionally done using analog hardware to now be performed using software. The use of this emerging technology allows for easy reconfigurability for our customer. This entire project's aim is to give our customer easier and faster access to information pertaining to other emergency response agencies and their respective wireless communication devices. Our system, however, is a mere prototype, but it is intentionally designed so that it can be scaled up to an industry grade product.

Index Terms

Radio, Emergency, Communicate, Communication, Challenges, Federal, Local, State, International, Police, Firefighters, Red Cross, Blue Cross, Interoperability,Signal Classification, Database Table, SDR, Javascript, CSS, HTML, HTTP, MySQL, MATLAB simulink, GNU Radio.

I. Introduction

The creation of a modulation classification system from end-to-end was a daunting and time consuming process. This documentation sums up our complete design from the societal problem down to the engineering diagrams. It covers, in detail, a description of our societal issue, issues with previous solutions, other approaches we could have taken, and the level of effectiveness to which our design solution solves the needs to real life emergency services. It will also detail the design idea contract to which we agreed to with our professor, as well as length of which our first semester and current semester prototypes meet our feature set. These topics will include changes in architecture, feature set adjustments, tool set adjustments, and cost benefits from the switching of platforms.

Funding, both internal and external, will be laid out in the form of a chart. We will clearly show from which sources we received funding, and what was covered internally. Financial sponsorship will be addressed as well. In addition, we will highlight key project milestones (both start and end dates), as well as the duration to which these milestones took to complete. The complete work breakdown structure will be highlighted separately, demonstrating our ability to cleanly distribute the workload throughout our group.

Following the work breakdown structure, we will list our risks and mitigation plans for various components. These risk and mitigation plans will be coupled with a clear report showing the associated level of risk. Next, we will give a broad design overview, giving a taste to the underlying technology that made our project possible. This overview will cover the entire project (end-to-end), from receiving the radio signal to the end user experience. With a full overview in place, we will demonstrate via testing reports the status of our deployable prototype. This will come with various testing methodologies, industry grade tools, and simple input to output checks. This will be followed by a full marketability forecast. The forecast will include details related to the viability of our product in the real world, key features that put us apart from possible competitors, and the ease of use that our product provides.

Lastly, we will include useful tools for end users such as a user manual, hardware block diagram, software block diagram, vendor contacts, and useful information about the team members. This information will be placed in the appendices at the end of the paper.

II. Societal Problem

Communication between first responders and emergency officials is a matter of public safety. Recently in California fire season has grown to an expected 200 days of the year. The most recent fire in California, the Camp Fire, was devastating to many towns and is considered the largest fire in California History [1]. Major natural disasters are a natural part of current society today and require a cooperative effort from all levels: Local, State, Federal, and sometimes International. The fundamental issue here is that each agency is operates on different bands to communicate within their own agency. However, this means that they can not effectively communicate with each other. This highlights a second issue that these disasters or attacks can happen anywhere, at any time. The usual solution for this communication issue is for a large corporation such as Verizon or AT&T to bring in a dedicated tower for everyone to communicate with via cellphone. While the dedicated tower is an effective solution, the tower comes at a cost of data rates and time to set up. In the even of a natural disaster such as the Carr fire evacuations must be immediate, and communication for getting citizens out of danger is vital. Currently hand mobile radios on multiple carrier waves is used for communication before a reliable tower is temporarily put in. By designing a system to mitigate communication over these bands such that their is minimal interference. This system also needs to be mobile friendly to allow deployment quickly and efficiently.

Given the wide variety of frequency and modulation options, restrictions, and varying equipment, communication has broken down to the point where agencies spend more time trying to communicate then they do resolving the issue at hand. This was evident in one of the most notorious emergencies in history, the Columbine High School shooting. Viktor Mayer-Schönberger, after discussing the brutal details of the shooting, writes, “Yet as it turned out, the biggest challenge on that Tuesday afternoon was not battling the two attackers. They had already killed themselves when the first law enforcement team entered the school. The biggest challenge was coordinating heavily armed and ready-to-fire police forces from half a dozen sheriff’s offices and twenty area police departments, forty-six ambulances, and two helicopters from twelve fire and EMS agencies, as well as personnel from a number of state and federal agencies” [2]. This is an identified problem, and as of today, there is not concrete, efficient solution.

Often, it is the beginning of a natural disaster that is the most dangerous time for civilians as they are for the most part untrained on how to react to these situations. The government has a system in place for alerting citizens in danger of a natural disaster or man made disaster such as a shooting called WEA (Wireless Emergency Alert). Communication of dangers to citizens is broadcast in the form of a 90 character text to alert them of danger nearby. Although, the message doesn’t get out to everyone in the area, and often it gets out to too many people causing traffic jams and further escalating the problem since first responders can’t get to the scene as soon as possible. The government has put law in place that carriers need to make the range of texts more accurate to resolve the problem. However with no financial gain in it for corporations the progress is relatively slow compared to other corporate projects.

Federal, State and Local agencies all struggle with emergency communication. Time does not permit analysis of all levels of government. Thus, we briefly focus on the federal level. “Growing exponentially over the last decade, the federal government has acknowledged and supported the need to improve compatibility issues for increased radio interoperability.” says Nancy Huyek in Interoperability challenges among public safety radio systems: surveyed research addressing these challenges [3]. She also stated that “...the federal government has been procedurally and technologically investigating methods by which radio communications between first responders may be improved for disasters requiring cross-organizational and jurisdictional efforts”, representing an immediate and direct problem that’s in drastic need of a solution [3]. Huyek later goes on to explain that as first responders such as police, fire, 911 services, medical personal, etc keep expanding, old channels will become full and new frequency channels will be created [3]. Thus, the “public safety radio spectrum has become divided with diverse organizations communicating on multiple frequency bands” [3]. These bands range from as low as 25 Mhz all the way up to nearly 6Ghz [4]. Below is an included table representing the public safety Spectrum bands.

Table I

FCC Public Safety Spectrum [4]

Frequency	MHz Available for Public Safety
25-50 MHz (VHF Low Band)	6.3 MHz
150-174 MHz(VHF High Band)	3.6 MHz[non-contiguous]
220-222(220 MHz band)	0.1 MHz
450-470(UHF Band)	3.7 MHz[non-contiguous]
470-512 MHz(T-Band)	6 to 12 MHz blocks[contiguous in specified markets][Licensing freeze in effect]
758-769/788-799 MHz(700 Broadband)	22 MHz(11 MHz x 11 MHz)[contiguous]
768-775/798-805(700 Narrowband)[1]	14 MHz(7 MHz x 7 MHz)[contiguous]
806-809/851-854 MHz(NPSPAC Band)	6 MHz(3 MHz x 3 MHz)[contiguous]
809-815/854-860 MHz(800 MHz Band)	3.5 MHz(1.75 MHz x 1.75 MHz)[non-contiguous]
4940-4990 MHz(4.9 GHz Band)	50 MHz[contiguous]
5850-5925 MHz band(5.9 GHz Band)	75 MHz[contiguous]

As you can see above, there are eleven different frequency ranges all within the 25 MHz to 6000 MHz range. Beyond the transmission frequency, there are still modulation schemes that need to be used in order to perform demodulation, or the decryption of the signal. The Federal Communications Commission has stated that “Some public safety channel assignments provide for exclusive use by a single licensee within its service area whereas others are shared” [4]. Within these bands, wireless communication can be received in many ways. The public safety institutions are known for using a tool called land mobile radios or LMR, which is based on outdated technology [3]. These tools cannot dynamically demodulate signals over a wide range of frequencies, resulting in the need for both safety agencies to communicate ahead of time on what frequencies and modulation schemes are being used. Many of us can think of these land mobile radios as something similar to walkie talkie we used to use as a kid. The federal government has spent, between the year of 2001 and 2004, over $5.4 billion in support of interoperability between agencies [2]. In an attempt to unify standards for which federal responders communicate, the federal government created what's called the Integrated Wireless Network (IWN) [2]. Huyck writes, "As a model for national interoperability, the federal government, through the Department of Justice, the Treasury, and Homeland Security, created the Integrated Wireless Network (IWN) in the hope that the program could be a collective radio communications solution for federal first responders." [2]. This system failed shorty after due to its incompatibility with local and state agencies. The current solution that is being implemented is that of relying on a handbook that contains the information about emergency agencies and their respective frequency channels and radio configurations. Until a new and robust solution is presented, this terrible handbook solution will continue to be used.

III. Design Idea Contract Review

Automatic modulation classification is the heart of the project and what makes our process unique. The goal of the algorithms presented in this paper is to assist emergency service communications operators to interact with the various teams in the field without having to go through an extensive manual. The system allows them to determine the modulation scheme being used by each team in the local area. The system classifies the modulation scheme within 30 seconds and then restarts the classification process in the case the team's scheme has changed. Currently the only schemes that can be classified are BPSK,QPSK, and 8PSK. However, it is possible for more schemes to be classified using this algorithm.

The entire system is executed in a custom GNU Radio block that interfaces with the rest of the project. It has also been successfully tested and run on a single board computer (Raspberry Pi 3B) that connects to the BladeRF. This is significant progress towards having a deployable prototype that simply needs to be plugged in for power.

Future work to be done on this feature includes: improved accuracy with a signal to noise ratio metric, and local generation of a server for the device to output data on. The system currently classifies it's 3 modulation schemes more than 60% of the time. Including the signal to noise ratio in this algorithm will increase the range to which the project meets this metric.

Modulation schemes are an integral part of modern communication systems. They are effectively the solution to the problem "how does one transmit data". There are three modulation schemes that this paper will focus on are Binary Phase Shift Keying (BPSK), Quadrature Phase Shift Keying (QPSK), and Eight Phase Shift Keying (8PSK). Each of these schemes' IQ constellation plot can be seen in the figures below. Automatic Modulation Classification

(AMC) allows the user of a communication system to keep in contact with the other members on their team without having to constantly switch manually between different communication methods. It is often the case that different equipment will be hardwired for different communication schemes. The following figure shows the ideal plots and what data each point in the constellation plot (known as a symbol) corresponds to.

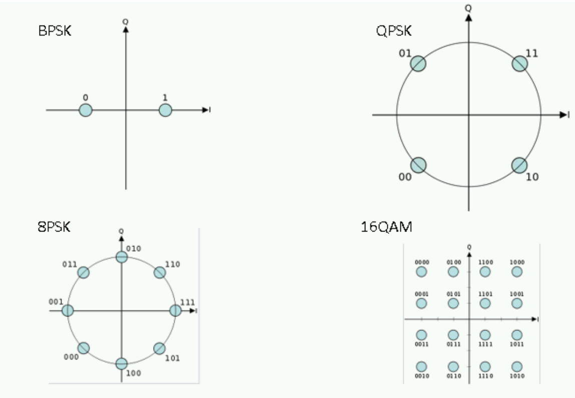


Figure 1: Constellation plots of 4 different schemes [8]

BPSK, QPSK, and 8PSK are the only schemes focused on in this study, showing that higher order schemes exist for the curious reader. The more symbols, the more bits can be transmitted corresponding to that symbol. For BPSK, 1 bit per symbol, for QPSK, 2 bits per symbol, and for 8PSK 3 bits per symbol. If you know what you are receiving, then it is easier to classify what data corresponds to the received symbol. However, what will be shown is a method for how to determine the type of scheme being used. Such that the user can tune their device to the right scheme to receive messages.

A. *Signal Conditioning*

The purpose of the signal conditioning is to apply several algorithms that converge in order to provide the cleanest possible data to the classification stage. For the purposes of our project, we used 3 total stages of signal conditioning. In the first semester, we implemented the algorithms in MATLAB and Simulink. The first stage included the automatic gain control (AGC) in the digital domain for fine signal scaling. The second stage included symbol timing synchronization algorithm. This is used to minimize the effects of intersymbol interference (ISI) and maximize the signal-to-noise (SNR) ratio of each sample. The third stage included algorithms for phase and frequency correction. When the team moved away from Matlab and into GNU Radio for the signal processing during the second semester, the AGC stage was eliminated. Also in GNU Radio, we introduced a constant modulus algorithm (CMA), which is an algorithm utilized for blind equalization. This algorithm works on digital signals with constant amplitude, which includes MPSK signals.

B. *User Experience*

The user interface is crucial for displaying the Carrier Frequency, Bandwidth, and Modulation Schemes to the user in a table like format. When a user clicks into one of the table options, they will be presented with additional data for the signal along with accuracy measurements. The user will be allowed to select what range of frequencies they would lake to scan over to fill the table with data. This will be accepted using a form type section of the site either above or below the main table. This data will then be pushed into the database for later use. The user, should potentially, be able to re-scan the data on command or add to the existing data by scanning another section. In addition, they will be presented with a map, showing what areas bladeRF devices are deployed. Lastly, the authentication piece of the user interface allows the end user to know the signals they are seeing as available are secure from public viewing. The metric that were met using this feature were ”Quick User Interface Load Times” and ”Display of useful information to the user(s)”.

C. Networking

In order to meet the network feature metrics, TCP(Transmission Control Protocol) communication is critical. Although TCP ensures data security and checks client side data packets, it does require more process time then UDP(User Datagram Protocol). However, under right management and efficient timing, this problem can be solved by maximizing server communication throughput, minimizing input/output transaction time, and minimizing database update time.

D. MAP Detector

This section shows sufficient conditions for classifying BPSK, QPSK, and 8PSK signals using a MAP detector. The map detector effectively acts as a means to classify for which region each signal belongs too. Assuming that when a message is sent x(t), that it picks up some additive white gaussian noise while it’s moving through the air, and the received message y(t) will be a result of both added together.

$$y(t) = x(t) + n(t) \tag{1}$$

Graphically, what was sent as a 0-data bit, for example, ends up being received in the wrong location.

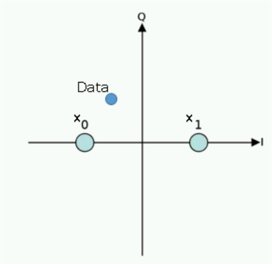


Figure 2: Data with some noise [8]

The way the algorithm works is if both the points are equally likely to occur, then the data bit corresponds to the ideal point that it is closest to. In reality there are millions of data bits being received, so each bit must be classified where it belongs. The following is an example of a noisy received in a QPSK message. The red crosses indicate where the symbols should ideally be received at. Note the following figure is using MATLAB to visualize the noisy signal.

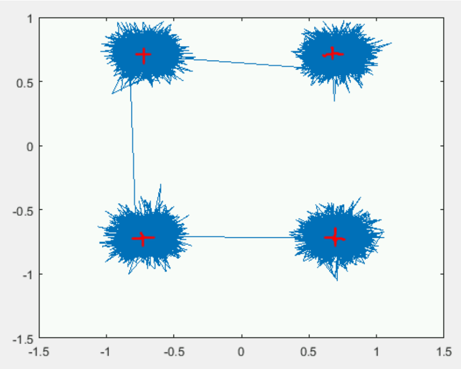


Figure 3: Noisy QPSK message [9]

Proof sketch:

Consider all of the symbols ideally sent to be labeled "$i \in X$" for our desired symbol, and "$j \in X$" if it's not our desired symbol, and let "$v \in Y$" be the received symbol. The assumption is that the probability of the symbol v showing up where it should is more likely than all the other places it could be. Note that the probability of noise occurring $P_{Y|X}(v|i) = P_n(v-i) = e^{\frac{1}{2\sigma^2}||v-i||^2}$. Where σ is the deviation from the ideal symbol point.

$$\begin{aligned} P_{Y|X}(v|i)P_X(i) &\geq P_{Y|X}(v|j)P_X(j) \\ P_n(v-i)P_X(i) &\geq P_n(v-j)P_X(j) \\ e^{\frac{1}{2\sigma^2}||v-i||^2}P_X(i) &\geq e^{\frac{1}{2\sigma^2}||v-j||^2}P_X(j) \\ ||v-i||^2 - 2\delta^2 ln(P_X(i)) &\leq ||v-j||^2 - 2\delta^2 ln(P_X(j)) \end{aligned}$$

Assuming all of the symbols that could be received for the scheme are equally likely to occur, then $P_X(i) = P_X(j)$ which is really nice, because the complicated term drops out and we are left with.

$$||v-i||^2 \leq ||v-j||^2 \tag{2}$$

Thus we have shown that the distance between the received data symbol v and it's closest ideal symbol is enough to classify what type of data it was supposed to be. This is what is known as a MAP detector for the maximum likelihood case.

E. *Modulation Scheme Classification*

Using the ideal symbols of 8PSK as a reference. Note that 8PSK has the look of a BPSK scheme and a QPSK scheme with more ideal symbols. However in the BPSK version only two of the symbols would show up from the 8 that are in 8PSK, and similarly only 4 would show up in the case of QPSK. Numbering off each of the ideal symbols in the 8PSK constellation plot we can use this as a way to classify the scheme.

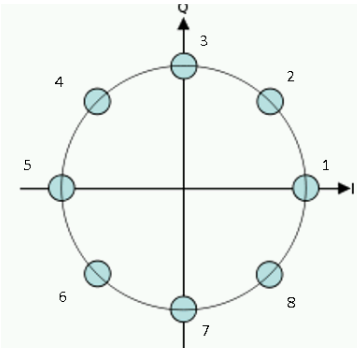


Figure 4: Classifier [8]

Assuming that the send data is equally likely for each of the sent symbols. We would expect for the case of 8PSK that each symbol would get 12.5% of the data mapped to it by the MAP detector. In the case of QPSK we would expect that symbols 2,4,6,8 would have about equal number of symbols mapped to them roughly

25% each. In the case of BPSK we would expect symbols 1 and 5 to have about 50% of the data mapped to them respectively. To be more explicit in how we are determining these probabilities. The following equation is an example for the 3^{rd} symbol.

$$P_3 = \frac{\text{Numbers of symbols classified as 3}}{\text{Total number of symbols received}} \tag{3}$$

Using this as our algorithm we can classify each scheme by the probability of symbols landing in their respective regions. Running a simple Python script in GNU Radio the case of Figure 2. The following probabilities were obtained.

Table II
Results of Algorithm [9]

P_1	P_2	P_3	P_4
0	0.2446	0	0.2489
P_5	P_6	P_7	P_8
0	0.2531	0	0.2534

Since the probabilities are around 25% for the symbols 2,4,6,8, then the modulation scheme must be QPSK. A very simple threshold algorithm can be used to automate this decision for the user.

IV. Funding

We received no outside funding for this project. All purchases were split between group members as seen in Table III below.

Table III
Internal Budget [9]

Contributor	Part	Model	Quantity	Price ($)
All	Software Defined Radio	bladeRF 2.0 micro xA9	1	820
Cory Diehl	Software Defined Radio	bladeRF 2.0 micro xA4	1	480
All	Antenna	ANT500	2	67
Total				1367

V. Project Milestones

Are project milestones this semester roughly reassemble a combination of assignments throughout the semester, as well as big achievements throughout our project. Our first milestone came in the form of identifying our societal problem. The societal problem is a key component to which we surround a technical solution. Identifying this problem took many 2 - 3 hour meetings. We had to find something that was a serious issue, but also had innovative solutions. Next, we had to create a design idea to address this societal problem. This was our second milestone. Identifying a solution had the struggles of finding a technology that allowed all four of our group members to work for two semesters, but wasn't too difficult to the point of in-completion. Along with this design idea, we had to create a list of feature metrics that we were to be graded on come the end of our second semester. Next we had to identify funding and the purchasing of our hardware. This was a huge milestone as it allowed us to actual begin writing and testing code in a real working environment. After we started, we had to create a WBS or Work Breakdown Structure. This allowed us to separate out the work and accomplish tasks in a timely, well uniform manner. This was a milestone in itself because it shed light on who had more work, and who would need help down the line. It also helped us to identify what dependency's we had, and how they would line up upon completion. Completing our Prototype first semester was also a milestone. It showed us that we were capable of getting a product tested, configured, and running by a due date. Lastly, the market review and analysis was key because it showed how our product would fit in the real world. In addition, Table IV below shows some more key milestones throughout the project on the technical level.

Table IV
Major Technical Project Milestones [9]

	Technical Milestone	Owner
1	Timing Recovery	Yuriy D
2	Carrier Recovery	Yuriy D
3	Signal Bandwidth Estimation	Yuriy D
4	Matched Filter	Cory D
5	Pooling	Cory D
6	Decision Tree	Cory D
7	Move DSP Algorithms to BladeRF GNU Radio	Yuriy D
8	Optimizing DSP Code	Yuriy D
9	Setting up GNU Radio	Cory D
10	Converting Matlab code into C++/python	Cory D
11	Optimizing Code	Cory D
12	Reduce backend node.js interaction with database	Nicholas B
13	Efficient Jade, CSS, & Javascript Code	Nicholas B
14	Front-End Interface	Nicholas B
15	Visual Representation of Accuracy	Nicholas B
16	User Interfacing	Nicholas B
17	Apply the input from/to user interface to/from SDR control	Alex H
18	Reflect the desired output to user	Alex H
19	Ensure the security during user interaction	Alex H
20	Website Authentication	Nicholas B
21	GPS coding & transmission of data	Alex H

VI. Work Breakdown Structure

The work breakdown structure consists of seven features from our project punch list and the class assignments, each with their own sub tasks and activities to break down the work. These features range from timing requirements to displayed features to a user. The class assignments, which are used to help promote project management. For sub-tasks, which typically range from two to three per feature, show broad points that need to be addressed in order to complete the feature. The activities, which fall under the sub-tasks, specify exactly what needs to be completed for the sub-task to be completely integrated into the project. By laying out these features and the corresponding tasks, the group will be more equipped to allocate there time for weekly tasks to fit the scope of the entire project. With all the features properly integrated, we hope to progress interoperability of public safety communication systems to a higher level of efficiency. This will be done by first in-taking signals using our software defined radio. This will move over to a neural network that will work to classify the signal modulation scheme. After the Signal Classification process, the next part is the transmission of processed signal output to online database. Once the data is pushed into the MySQL database, it will be extracted using JavaScript on a node.js back-end server. The front-end JavaScript will make an API call to the back-end requesting the updated data. This data will be present on a table interface. By selecting a signal, an agency will be able to see more specific details regarding the data.

A. Identify Modulation Scheme for Emergency Services

The identification of modulation schemes for emergency services requires classifying data derived from the signal. For the algorithms to yield a high probability of accurate classification of the modulation scheme, signal processing algorithms are required at the receiver to produce discernible data for the classification algorithms. These signal processing algorithms include timing recovery and carrier recovery. Carrier frequency and signal bandwidth will also be determined.

1) Timing Recovery: The purpose of timing recovery is to obtain symbol synchronization. Two things are to be determined by the receiver in order to achieve symbol synchronization. Sampling frequency is to be determined by estimating the symbol period so that samples can be taken at the correct rate. Sampling phase is also to be determined by finding the correct time within a symbol period to take a sample. In real communication systems, binary data pulses have a peak in the center of the symbol period. To increase SNR, it is best to sample the symbol in the middle of this peak.

2) Carrier Recovery: In a real world communication system, there exists a frequency and or phase offset between the oscillator at the transmitter and the oscillator at the receiver. Also, if there exists relative motion between the two oscillators, the Doppler effect yields a frequency offset. The frequency offset causes the symbols to rotate at the rate of the offset. This can be visualized on the constellation plot. This is why it is imperative to write a carrier recovery algorithm. The key to the carrier recovery algorithm is a phase-locked loop (PLL). This enables locking onto the carrier frequency.

3) Signal Bandwidth Estimation: Signal bandwidth estimation will be done through implementation of either parametric or non-parametric methods. The signal will be treated as a random process with finite average power, and the power spectrum will be estimated to capture the distribution of power in the signal. From this, the bandwidth can be estimated. Non-parametric methods assume no underlying process and are Fourier based. Parametric methods are model based include observing the data, estimating the model parameters using that data, and expressing the power spectrum as a function of those model parameters.

B. Accurate Identification of Modulation Scheme

Once a signal has been identified to be not just noise, the identification process can begin. There are several modulation schemes, the big 3 being Amplitude Modulation (AM), Frequency Modulation (FM), and Phase Modulation (PM). Being able to determine which modulation scheme is being used by another emergency service team and automatically adjusting for the scheme will allow for easier communication between teams. Without the user needing to know technical knowledge regarding modulation schemes. The overall method being used to automatically identify and classify the modulation schemes is a convolutional neural network.

1) Matched Filtering: A matched filter to a modulation scheme is the ideal version of the the modulation scheme. Sweeping the matched filter IQ plot through the IQ plot of the the data will provide a heat map for each type of modulation scheme being used. Each modulation scheme has different characteristics, and the purpose of a matched filter is to match those characteristics. Since each matched filter of the characteristics of the modulation scheme will.

C. Pooling

A normal pooling method is to take the max of a region of the IQ plot and force the negative values to zero. Rather than blindly doing this rather than just throwing out terms based on their negative value the system will statistically determine whether or not the signal is statistically significant compared to Gaussian noise before throwing it out.

1) Decision Layer: The decision layer will weight each set of pooled data accordingly to the modulation scheme that it's trying to identify. AM based schemes will have a higher weight ,and the weight of the FM and PM pooled data will be weighted less when deciding if the signal is AM. Similarly for the PM and FM schemes.
Whichever has the highest value will be the output of the Decision Layer. Since the weights are not well defined, this is the section of the neural network that back propagation can be done to optimize the choice in weights. Which will in turn make the neural network more accurate.

D. Efficient Processing Time for Signal Separation and Detection

This feature is to be accomplished the second semester of senior design. For the first semester, Matlab/Simulink will be used for the signal detection and conditioning. This means that data will be processed on a laptop or PC that will be running Matlab and Simulink. For the second semester, the goal is to convert the Simulink models and Matlab code into C++/Python code that can be executed on a DSP, Laptop, or microcontroller.

1) Signal Separation & Detection: The Simulink models and Matlab code will have to be either all manually translated into a C++ or python program, or they will have to be automatically translated using some pre-existing environment and compiler. If the models and Matlab programs will be translated automatically, the code will still have to be optimized.

E. Efficient Signal Classification Processing Time

A convolutional neural network is a very computationally heavy method for classifying things in general. The reason for the large amount of computation is that each feature that is needed to identify the modulation scheme requires a convolutional matrix operation. To handle the large number of matrix operations and improve the performance, the convolutional neural network will be implemented on the DSP or laptop.

F. Quick User Interface Load Times

Improving load times on the user interface is crucial in user acceptance and adoption. If a user cannot quickly access and interact with a service, it is likely that they will stop using it and the service will die off. To promote quick load times, the backend JavaScript code will need to make efficient calls back to the database using separate functions that can be selected using parameters. Also efficient Jade and CSS coding will allow for the website to load quickly. If, for example, a CSS property tried to make changes on all objects, (*), it could cause load times to drastically increase for each additional div in the Jade.

1) Reduced BackEnd Node.JS Interaction With Database: The backend Node.js based server will be required to interact with the database. In order to interact with the database, a file called database.js bust contain the correct API calls. By creating this file the front end can interact and grab data from the MySQL server. To create this as efficiently as possible, it's important to separate out the database calls into different functions that can be varied based on the parameters passed to them.

2) Efficient Jade, CSS, & Javascript Code: Jade, CSS, and JavaScript code can potentially create massive load times if not configured correctly. To help this, all commonly ran JavaScript functions will be separated out into different functions, allowing for common functions calls to take less time. In addition, CSS changes to the visuals of the website will be done optimally, by only addressing the specific classes that need to be changed.

G. Display of Useful Information to the User(s)

Displaying data to the user involves a variety of subtasks. The front-end interface will need to properly present this data, in a reasonable size, and within a very easy to read structure. When thinking about what a user wants, Jeff McWherter & Scott Gowell in Professional Mobile Application Development introduce the questions, "Who are the users, what do they need and why, and how, when, and where will they access and use information?” [4]. To answer this question we addressed the following criteria for Emergency Services; Carrier Frequency, Bandwidth, Modulation Scheme, & an Accuracy Metric. The accuracy metric will be presented visually using some type of color map, accuracy bar, or other easy to see visual representation. Lastly, user interfacing is a crucial part of the project as it allows the emergency services agency to calibrate what frequencies they would like to scan over.

1) Front-End Interface :

The user interface will be crucial for displaying the Carrier Frequency, Bandwidth, and Modulation Schemes to the user in a table like format. Ideally, when a user clicks into one of the table options, they will be presented with additional data for the signal along with accuracy measurements.

2) Visual Representation of Accuracy :

To visually represent the accuracy of our measurements, we will display to the user a visual diagram. This could range from overall accuracy using a progress bar, individual accuracy by displaying the signal and our corresponding measurements, or anywhere in between. This should be presented to the user when clicking into a signal from the table, or directly below the navigation bar.

3) User Interfacing back to the Database:

The user will be allowed to select what range of frequencies they would lake to scan over to fill the table with data. This will be accepted using a form type section of the site either above or below the main table. This data will then be pushed into the database for later use. The user, should potentially, be able to re-scan the data on command or add to the existing data by scanning another section.

H. User Controls to Adjust Settings

User interaction is important part for our senior project since we are developing Modulation Classification for Emergency Services. If our system can’t interact with users such as firefighters, police, and ambulance operators it is pointless to have a such expensive and time consuming system. In order to establish the user interaction, we have to make a system that can take an input from a user and return the desired output. In other words, This system has to be able to communicate between user and SDR. Not only for that, it also have to have a robust security system that can prevent unauthorized activities in the database. In detail, our user interaction system is partition to 3 distinct sections apply the input from user interface to SDR control, return the desired output to user, and ensure the security during user interaction.

1) Apply the input from user interface to SDR control:

Followed by Nick’s part, the user interface will displaying the Carrier Frequency, Bandwidth, and Modulation Schemes to any user who have access to it. Moreover, the users have a option to pick the desired Carrier Frequency with adjustable Bandwidth. User also have a choice to select the Modulation Schemes. Depends on the users inputs (alteration of

Carrier Frequency and Bandwidth with Modulation Schemes selection) there will be different stages of changes to database table. If user input error occurs, re-prompt to user and discard the current user input. The database table (excel format linked with mySQL) will leave a mark what is going to be changed and send a request to SDR. Then, SDR will operate whatever the user is asking for.

2) Return the desired output to user: Once SDR done its operation successfully, output will be sent to the database and update it. If an error occurs from SDR side, report to user and restart the process. (SDR error checking part will be on Yuriy's section) Whenever database detects the alteration of the table, it will update the table and print out the changes to user (Making user friendly table output will be on Nick's side).

3) Ensure the security during user interaction: This part is going to be the final step for our project. Once user input and output system is complete with maximum efficiency and correct error checking, we are going to establish security for each direction which is from user to SDR vise versa. Our system will be able to accept different inputs from multiple users ideally. For that matter, we are adding the system that prevents users to access to other people's operation. Furthermore, we are applying security at the user interface part mainly for blocking unauthorised user activities such as datatable reset, spamming random inputs, and channing other user's information. This part will be improvised later during second semester.

Table V
Work Breakdown Package for Emergency Service Communications [9]

Feature	Subtask	Activity	Work Package
Feature 1: Identify modulation scheme for emergency services.			
	Subtask 1.1 Timing Recovery		
		Activity 1.1.1 Determine sampling frequency by estimating the symbol period	
			• The signal will be resampled in Matlab/Simulink and timing error detector will be used
		Activity 1.1.2 Lock the sampling phase by determining the correct time within a symbol period to take a sample	
			• A PLL system will be used in the Matlab/Simulink environment
	Subtask 1.2 Carrier Recovery		
		Activity 1.2.1 write a PLL based algorithm to keep correct for frequency/phase offset and keep symbols from rotating	
			• Matlab's "coarse frequency compensation" block will be used to find an estimate of the frequency offset. • Matlab's "fine frequency compensation" block will be used. It uses a PLL which incorporates a direct digital synthesizer to compensate for the remaining frequency and phase offset.

Feature	Subtask	Activity	Work Package
	Subtask 1.3: Signal Bandwidth Estimation		
		Activity 1.3.1 Employ a parametric or non-parametric method for power spectrum estimation	
			• The periodogram and/or welch method may be used for spectrum estimation. • Another possible approach would be an Auto-regressive model for spectrum estimation.
Feature 2: Accurate identification of modulation scheme			
	Subtask 2.1 Matched Filter		
			• Create database of matched filters for modulation scheme of known plots to correlate to. • Create a database of deterministic noise such as thermal noise to increasing the accuracy of the Hypothesis test.

Feature	Subtask	Activity	Work Package
	Subtask 2.2 Pooling		
		Activity 2.2.1 Refine statistical hypothesis for testing for noise.	
			• Reduce the size of the convolution layer via pooling. – Use a hypothesis test to determine if the subsection is just noise or not.
	Subtask 2.3		
		Activity 2.3.1 Decision tree	
			Create a fully connected layer of the decision tree to decide the scheme. Optimize the weights of the decision tree via trial and error from testing.
Feature 3: Efficient processing time for signal separation & detection.			
	Subtask 3.1: Achieve signal detection and conditioning in GNU Radio		
		Activity 3.1.1: Study GNU Radio environment and learn how to develop algorithms in it.	
			• Simulink has the capability through Xilinx to convert system blocks to c code. Code can then be optimized.

Feature	Subtask	Activity	Work Package
Feature 4: Efficient signal classification processing time.			
	Subtask 4.1: Setting up the bladeRF		
		Activity 4.1.2: installing binaries	
			• Install binaries
	Subtask 4.2: Converting-Matlab code to C or Python		
		Activity 4.2.1:Translate decision tree code from MatLab into python	
			• Write a python block in GNU Radio Companion.

Feature	Subtask	Activity	Work Package
Feature 5: Quick user interface load times.			
	Subtask 5.1: Reduce backend node.js interaction with database		
		Activity 5.1.1: Write backend node.js file (database.js) that connects to database.	
			• Create database.js file, include the mysql npm module. • Initiate connection settings.
			Create connection to grab data from databaseSimilarly, create a connection for entering data into the database.
		Activity 5.1.2: Seperate out API calls using parameters.	
			• Write separate API calls – Input – All Output – Focused Output
	Subtask 5.2: Efficient Jade, CSS, & Javascript code		
		Activity 5.2.1: Create efficient Javascript code, separate out commonly used functions.	
			• Create simple interfacing javascript. – Tabs – Pages – Clicking on Table
			• Functions for various needs – Refreshing table – Showing map information – Showing accuracy metric
		Activity 5.2.2: Simplistic Jade code with reduced CSS changes.	
			• Produce CSS code for each object • Focus on reducing margin changes or other small changes.

Feature	Subtask	Activity	Work Package
Feature 6: Display of useful information to the user(s)			
	Subtask 6.1: Front-End Interface		
		Activity 6.1.1: Grab and display Carrier Frequency	
			API call to databaseStructure data into table
		Activity 6.1.2: Grab and display Bandwidth	
			• call to database • Structure data into table
		Activity 6.1.3: Grab and display Modulation Scheme	
			• call to database • Structure data into table
	Subtask 6.2: Visual Representation of Accuracy		
		Activity 6.2.1: Identify Accuracy Metric	
			• call to database • Structure data into table
			• Massage data into visible metric – Using API – Showing Signal
		Activity 6.2.2: Visual represent metric in easy to read fashion	
			• Identify best way to show data • Write Javascript and CSS that represents an accuracy metric.

Feature	Subtask	Activity	Work Package
	Subtask 6.3: User interfacing		
		Activity 6.3.1: Create form for user input	
			• Create tab option labelled “Input Parameters” • Let the user input data via – Form – Adjustable text – etc.
		Activity 6.3.2: Digest data and push to database	
			• Receive user input, and immediately make call to database. • Test to make sure data is moved. • Test to see any characters that have trouble moving to database.
Feature 7: User controls to adjust settings			
	Subtask 7.1: Apply the input from user interface to SDR control		
		Activity 7.1.1: Accept the user input from user interface	
			• MySQL Installation in Anaconda
			• MySQL Workbench Installation
			• Write a code that accepts datas from user interface

Feature	Subtask	Activity	Work Package
		Activity 7.1.2: Apply the input to database table from/to SDR to/from user	
			• Write a code that accepts datas from SDR and user interface • Python-text file export test • Python-MAT file export testPython-CSV file export test
			• Python database coding (Create DB, Insert data to DB, Display DB)
			Simulation work • Python-text file export test • Python-MAT file export test • Python-CSV file export test – If error occurs, go back to export text and do it again
			Simulation work • Python-text file import test • Python-MAT file import test • Python-CSV file import test
		Activity 7.1.3: Depends on what have inserted to database table, order SDR to generate the desired output	
			• Write a code that sends a request to SDR(Using Python -Data Analysis & Plot Tools)

Feature	Subtask	Activity	Work Package
	Subtask 7.2: Reflect the desired output to user		
		Activity 7.2.1: Bring the output from SDR to database table and save it. If there is an error, report to user and discard the changes.	
			• Write a conditional code the checks errors and categorise user input options(Using Python -Data Analysis & Plot Tools)
		Activity 7.2.2: Print out the user friendly output to user interface	
			• Download tkinker (GUI software)
			• Display Python-DB using GUI
			• Python-HTML (Basically, sending output to Nick)
	Subtask 7.3: Ensure the security during user interaction		
		Activity 7.3.1: Apply security during user interaction from user interface to SDR.	
			• Making sure data interaction is based upon encrypted transaction
		Activity 7.3.2: Create the secured environment while SDR is applying user inputs.	
			• TBD, however, if we do ended up needing external security, I'm planning to use the Blockchain DB (database security software)

Feature	Subtask	Activity	Work Package
Feature 8: Assignments			
	Subtask 8.1: Problem Statement		
		Activity 8.1.1: Identify a significant problem within our society	
		Activity 8.1.2: Each member research about this problem	
		Activity 8.1.3: Write a 10 to 15 pages IEEE format report as a team	
	Subtask 8.2: Design Idea Contract		
		Activity 8.2.1: Describe senior design idea with detailed information	
		Activity 8.2.2: Explain the nature and feature set of team design	
		Activity 8.2.3: Write a 15 to 20 pages IEEE format report as a team	
	Subtask 8.3: Work Breakdown Structure		
		Activity 8.3.1: Create a lists of tasks to outline the broad features	
		Activity 8.3.2: Create the subtasks under feature set which is required to implement that feature.	
		Activity 8.3.3: Write a 15 to 20 pages IEEE format report as a team	
	Subtask 8.4: Project Timeline		
		Activity 8.4.1: Create a Gantt chart and PERT diagram of the anticipated tasks of senior project	

Feature	Subtask	Activity	Work Package
	Subtask 8.5: Risk Assignment		
		Activity 8.5.1: Create a risk chart to show the level of project risk vs the level of project impact if that risk occurs	
		Activity 8.5.2: List possible mitigation if that risk occurs	
	Subtask 8.6:Technical Review		
		Activity 8.6.1: Demonstrate the integrated components of your design idea in a complete laboratory prototype	
		Activity 8.6.2: Present the technical details of your laboratory prototype	
	Subtask 8.7: Laboratory Prototype Presentation		
		Activity 8.7.1: Demonstrate a working laboratory prototype that performs the functions of team design idea feature set	
	Subtask 8.8: Problem Statement Revision		
		Activity 8.8.1: Revise the problem statement	
		Activity 8.8.2: Revise design idea and Timeline	
		Activity 8.8.3: Presentate a short overview of your revised problem statement/design idea	

Feature	Subtask	Activity	Work Package
	Subtask 8.9: Device Test Plan		
		Activity 8.9.1: Develop a test plan that proves the laboratory prototype works as expected over a convincing range of factors	
	Subtask 8.10: Market Review		
		Activity 8.10.1: Develop a strategic plan for the prototype to impress people from non- engineering field.	
		Activity 8.10.2: Write an IEEE format report as a team	
		Activity 8.10.3: Give a short overview of market review to the entire senior design group.	
	Subtask 8.11: Feature Presentation		
		Activity 8.11.1: Write an IEEE format report individually that include every assignments	
		Activity 8.11.2: Give an individual presentation that summarize your feature	
	Subtask 8.12: Midterm Progress Review		
		Activity 8.12.1: Update team test plan that shows all tasks are on-target and no insurmountable obstacles remain.	
		Activity 8.12.2: Report the test results in a professional and typewritten document.	
		Activity 8.12.3: Demonstrate your project at your workstation	

Feature	Subtask	Activity	Work Package
	Subtask 8.13: Engineering Ethic Quiz		
		Activity 8.13.1: Complete a quiz on engineering ethics.	
	Subtask 8.14: Deployable Prototype Review		
		Activity 8.14.1: Demonstrate the essentially complete deployable prototype if there is a part that cannot be presented in lab, create a video for that matter	
	Subtask 8.15: Deployable Prototype Documentation		
		Activity 8.14.1: Fully document all aspects of your senior design project	
		Activity 8.14.2: Write a full IEEE (two column format) format report as a team	
	Subtask 8.16: Deployable Prototype Presentation		
		Activity 8.15.1: Final task, demonstrate a working deployable prototype.	

Table VI
Table of hours [9]

Team Member	Primary Task(s) for Fall 2018	Hours
Nicholas Bergamini	• Research Societal Problems & Solutions • Research Societal Problems & Solutions for Northrop Grumman & Contribute to Team Societal Report • Implementation Research & Initial Setup of MySQL Database • Wrote CSS & Jade code to create the shell for our web interface • Establishing back-end Web Server connection to the Database & creating appropriate MySQL accounts. Writing WBS. • Web server navigation and table display of data. WBS • Web interface table details option & Maps.Gantt and Pert Charts. • User input system development from webserver to database. • User input system using parameter frame with input hazard prompt. • User input system integration with Alex Han Project optimization • User Authentication	∼21 ∼8 ∼30 ∼16 ∼16 ∼15 ∼21 ∼14 ∼30 ∼53 ∼55 Total 279 Hours
Alexander Han	• Research Societal Problems & Solutions • Research Societal Problems & Solutions for Northrop Grumman & Contribute to Team Societal Report • Implementation Research • Research into MATLAB/Simulink interfacing with Python Script for compression • Compression research, User interaction research, & writing WBS part • Script for processing data into database from Simulink & WBS • Python program to write data back to database • Gantt Chart Establishing sending connection between main server and BladeRF • Simulation of established connection. Take input from user and carry through server and BladeRF board. • Continuation of communication coding & Bidirectional commutation research • User input system integration with Nick Bergamini • Output system integration with Cory Diehl Integration from Nick's part to Alex's part	∼16 ∼8 ∼30 ∼15 ∼13 ∼17 ∼15 ∼20 ∼45 ∼66 ∼29 Total 274 Hours

Team Member	Primary Task(s) for Fall 2018	Hours
Yuriy Dzyuba	• Research Societal Problems & Solutions • Research Societal Problems & Solutions for Northrop Grumman & Contribute to Team Societal Report • Implementation Research & Initial Setup of SDR Ordering Parts • Ran simulations consisting of spectrum analysis in Simulink • Performed I/Q constellation analysis of received signals • Learned about timing recovery, carrier recovery, and channel equalization • Creating, researching, and testing recovery with timing recovery algorithms • WBS Implementing timing recovery and carrier recovery algorithms for received signals where the receiver has no a priori knowledge of the incoming signal • Carrier Recovery with timing Receiving a real-world transmitter signal Channelization work Receiving part of the communication Performing energy for the local oscillator sweep using Parseval's energy • GNU radio function integration with Cory Diehl	∼30 ∼30 ∼25 ∼16 ∼41 ∼65 ∼20 ∼23 ∼15 ∼45 Total 330 Hours

Team Member	Primary Task(s) for Fall 2018	Hours
Cory Diehl	• Research Societal Problems & Solutions • Research Societal Problems & Solutions for Northrop Grumman & Contribute to Team Societal Report • Implementation Research & Ordering Parts • Research Modern Spectral Estimation theory • Proof of concept for the first 3 nodes of the convolution neural network • Development of a convolution neural network for IQ plots • Worked on the matched filter technique and then convolving it with IQ data using MATLAB(GNU radio for 2nd Semester) • Continue to work on the classification algorithm then testing constructed algorithm • GNU radio control system through Python code • GNU radio output integration with Alex Han • GNU radio function integration with Yuriy Dzyuba Improving the algorithm for classifying the modulation scheme	∼20 ∼25 ∼15 ∼15 ∼18 ∼10 ∼15 ∼44 ∼20 ∼48 ∼30 Total 270 Hours
		Team Total 1,153 Hours

VII. Risk Assessment and Mitigation

Possible mitigation is pretty straight forward. To meet the industrial level requirements and expectations, always work hard and try your best to impress sponsors. However, if we do ended it up having this risk, only way to fix this problems is going through apologies towards sponsoring companies and never make the same mistake.

For financial risks, it is not the end of the day. Nevertheless, it is always better to stick with the budget that is given. To mitigate this, we already moved to GNU Radio under python environment during second semester which saved good amount of money.

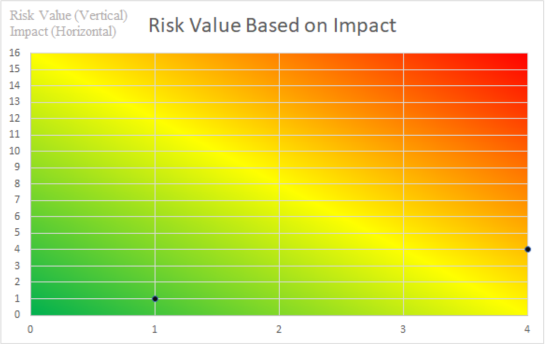


Figure 5: Meeting Industrial Integrity and Financial Management Risk [9]

Table VII
Meeting Industrial Integrity and Financial Management [9]

Persons conducting the assessment: Alexander Han, Yuriy Dzyuba, Nicholas Bergamini, Cory Diehl Date: 11/5/2018					
Task	Hazards (Existing and Potential)	Risks	Risk Value (Priority)		
			Likelihood (1-4)	Impact (1-4)	Risk Level (Likelihood x Impact)
Purchasing essential boards or parts	Due to price requirement we might ended up exceeding team budgetSince MATLAB cost so much, it is possible to pay lots of money to keep the license	Have to use personal moneyMight cause financial problem if it goes too long	1	1	1
Meeting industrial level requirements / expectations	Team couldn't meet up on time.Some teammates late Couldn't meet the deadline (If they do request some documentation)	Losing trust from sponsors. The worst case, might discontinue the sponsorship	1	4	4

To mitigate the issue of noisy data, a lot of simulations were run in Matlab/ and Simulink to ensure the system works. The simulation utilized models of real-world channels. For example, additive white Gaussian noise (AWGN) channels were used to simulate the type of data that would be present at the bladeRF receiver. This allowed for better system development. The real receiver algorithms implemented in Simulink were already expecting noisy data. There is not much we can do about the effects that the receiver has on the bandwidth estimation.

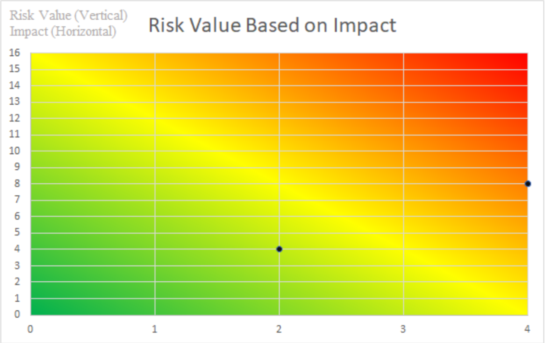


Figure 6: Identify Modulation Scheme for Emergency Services [9]

Table VIII
Identify Modulation Scheme for Emergency Services [9]

Persons conducting the assessment: Alexander Han, Yuriy Dzyuba, Nicholas Bergamini, Cory Diehl Date: 11/5/2018					
Task	Hazards (Existing and Potential)	Risks	Risk Value (Priority)		
			Likelihood (1-4)	Impact (1-4)	Risk Level (Likelihood x Impact)
Task 1.1 Timing Recovery	Timing recovery algorithm does not achieve symbol synchronization	Data will be more noisy and symbols will be smeared	2	4	8
Task 1.2 Carrier Recovery	Carrier recovery algorithm is not able to properly compensate for phase/frequency offset	Data will contain a lower signal to noise ratio and matched filtering will be more difficult	2	4	8
Task 1.3: Signal Bandwidth Estimation	Bandwidth cannot be accurately estimated because of receiver and algorithm limitations	The user will not know what the bandwidth of any particular signal is	2	2	4

To help mitigate these issues, we can try to check if the database table query gives an error. If it does, we can generate code that tells the users there has been an error and to contact one of us for a fix. Moreover, during second semester, we made two more changes to our database which is GPS location data and authentication. This made us difficult to mitigate the possible hazards because these two data should return nothing until user inputs the valid ID and password. However, since Nicholas Bergamini used OKTA API as an authentication protocol, the risk of having database initialization is close to zero.

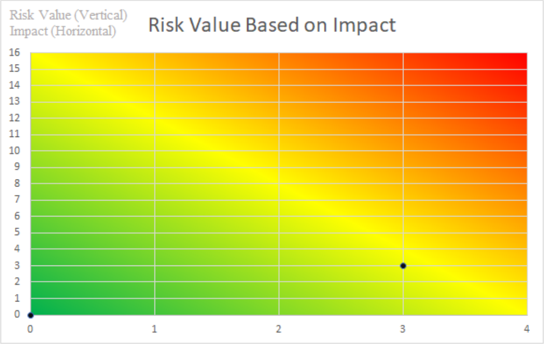


Figure 7: Quick User Interface Load Times Risk Assessment [9]

Table IX
Quick User Interface Load Times [9]

Persons conducting the assessment: Alexander Han, Yuriy Dzyuba, Nicholas Bergamini, Cory Diehl Date: 11/5/2018					
Task	Hazards (Existing and Potential)	Risks	Risk Value (Priority)		
Task 2.1: Reduce back-end node.js interaction with database	Data from database could be in the wrong format. Table could be corrupted.	Would bring the entire web server down which wouldn't come back up on it's own.	1	3	3
Task 2.2: Efficient Jade, CSS, & Javascript code	None	None	0	0	0

Mitigation plans regarding syntax or UI element errors can be resolved through thorough testing. Also by not allowing access to the front end web code to the entire group, it's likely we will not encounter many issues. User interfacing issues are going to be a bit harder to deal with. The best plan is to alert the user that what they entered is invalid and have them re-type a number. We can also extract incorrect characters. In our case the user will be entering two numbers over which they would like to scan. We could remove all alphabetic characters from this input.

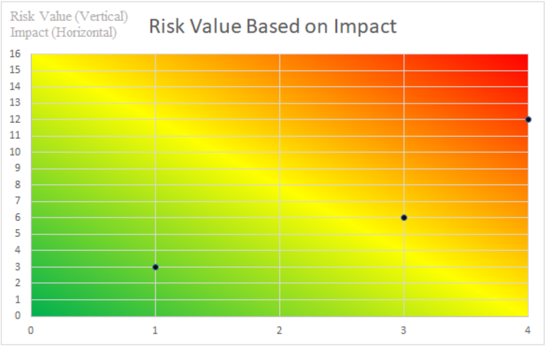


Figure 8: Displaying Useful Information to User(s) Risk Assessment [9]

Table X
Displaying Useful Information to the User(s) [9]

Persons conducting the assessment: Alexander Han, Yuriy Dzyuba, Nicholas Bergamini, Cory Diehl Date: 11/5/2018					
Task	Hazards (Existing and Potential)	Risks	Risk Value (Priority)		
Task 3.1: Front-End Interface	Syntax errors in the jade file.	Page can redirect to "Page not Found". It could also fail to load if to large.	2	3	6
Task 3.2: Visual Representation of Accuracy	Overlapping UI elements.	Could cause the user to not be able to click on an object.	3	1	3
Task 3.3: User interfacing	User enters incorrect format.	The page could break due to error connecting back to database.	3	4	12

In order to mitigate input problems, only way to solve this is going through simulation and testing. Our project is going to be simulation heavy project. Debugging means reassuring the correct simulation that is ultimately contributes to correct outcome to user. Same as the input risk factors, if this system do reflect wrong output from user to SDR vise versa, setting multiple testing case and debugging the error is the only way to solve this problem. Security part is mainly encoding the whole interoperable transaction. If there is a risk that causes security hazard, the mitigation plan is going to changing the encryption program.

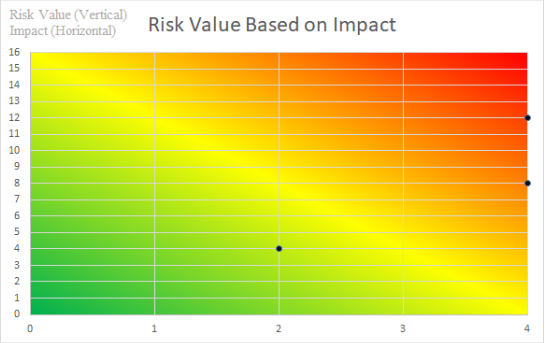


Figure 9: User Controls to adjust settings risk value assessment [9]

Table XI
User Controls to Adjust Settings [9]

Persons conducting the assessment: Alexander Han, Yuriy Dzyuba, Nicholas Bergamini, Cory Diehl Date: 11/5/2018					
Task	Hazards (Existing and Potential)	Risks	Risk Value (Priority)		
Task 4.1: Apply the input from/to user interface to/from SDR control	User can't access to database Database fails to read the output from SDR Connection problem between user interface and SDR	Whole project design is failing, which means we might have to restart from designing connection that will take lots of time.Risk of changing the core idea of interoperability design that will cause lots of time loss due to extra make-up documentation	2	4	8
Task 4.2: Reflect the desired output to user	Applies input incorrectly outputs	User can't read the outputs due to SDR and user interface communication error User gets the wrong output which means our project fails the core objective	2	2	4

Task 4.3: Ensure the security during user interaction	Security fails to protect the data transaction between SDR and user	Other user can ended up getting someone else’s information Unwanted people can access to interoperable system and gets the data	3	4	12

A possible mitigation is the prevention of the security risk factor. Unlike any other part of our Senior Design Project, only way to mitigate this risk is prevent such hazard before it even become an issue for our system. The methods of prevention is going to be installation of external encryption program and using different languages to manually encrypt the small part of the coding from user interface.

During second semester, we successfully implemented OKTA authentication API to our main server system. This API eliminates most of the possible hazard from security issue.

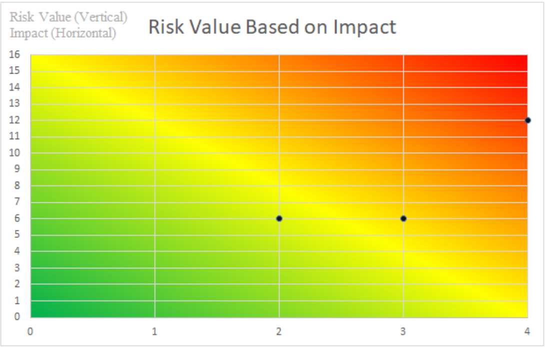


Figure 10: Security Application Risk Value Assessment [9]

Table XII
Security Application [9]

<table>
<tr><td colspan="6">Persons conducting the assessment: Alexander Han, Yuriy Dzyuba, Nicholas Bergamini, Cory Diehl Date: 11/5/2018</td></tr>
<tr><td>Task</td><td>Hazards (Existing and Potential)</td><td>Risks</td><td colspan="3">Risk Value (Priority)</td></tr>
<tr><td>Task 5.1: Secure Database Connection</td><td>Other unauthorized users can access to database.</td><td>Data loss or unwanted changes on database</td><td>2</td><td>3</td><td>6</td></tr>
<tr><td>Task 5.2: Encrypting Passwords</td><td>Once unauthorized users acquires access to database, they can also take passwords of actual user in this system.</td><td>User information leak</td><td>3</td><td>4</td><td>12</td></tr>
<tr><td>Task 5.3: Website Authentication</td><td>Any user can join into the system</td><td>Server down</td><td>3</td><td>2</td><td>6</td></tr>
</table>

First of all, I put this whole section as 1 likelihood because any of these risk should not be happening at the end of the project. If we do get either of these risk factors, this means we are severally failing the entire course. However, we still have one way to mitigate these possible risks. Although personally do not want to take this path, but if it happens we, as a team, have to crunch through time like there is no tomorrow. Start from the beginning, check every single individual parts while peer checking with others and run the simulation for insuring the objective of the project. Again, any of these risk factor should not be happening.

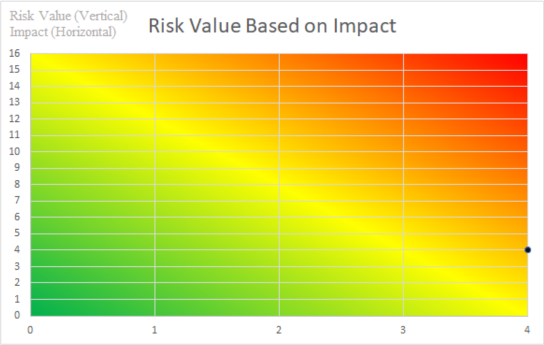


Figure 11: Project Finalization Risk Value Assessment [9]

Table XIII
Project Finalization [9]

Persons conducting the assessment: Alexander Han, Yuriy Dzyuba, Nicholas Bergamini, Cory Diehl Date: 11/5/2018					
Task	Hazards (Existing and Potential)	Risks	Risk Value (Priority)		
Task 6.1: Project Objectives	Fails to secure the objective of the whole project	Directly fails the whole Senior Project course	1	4	4
Task 6.2: Quality Deliverables	Fails to ensure quality of overall project design	Won’t be able to impress industryLost the objective of getting the sponsorship from industry	1	4	4
Task 6.3: Effort and Cost Tracking	Fails to keep the efficiency alongside with affordable cost range	Project becomes inefficient that is going to affect our overall performance	1	4	4
Task 6.4: Project Performance	Fails to Perform	Directly fails the whole Senior Project course	1	4	4

A. *Risk Value Calculation*

Likelihood Probability

4 = Very likely to occur
3 = Somewhat likely
2 = Remote

1 = Improbable

Potential Impact
4 = Severe
3 = Substantial
2 = Minor
1 = Minimal

Risk Value = Incident Probability X Potential Impact

Risk Level $>$ 11, High Risk (Take immediate action)
= 4 – 11, Medium Risk (should take some actions)
$<$ 4 Low Risk (Continued operation is permissible with minimal controls)

VIII. Design Philosophy

The design philosophy for our project was one that was technology centric. Before our group chose a societal problem to address, we analyzed our current skills and took note of the skills that we wanted to develop for professional marketability. We looked to industry to see what technologies were currently being used in our respective fields of interest. Next, we sought a societal problem that can be addressed using the technologies that we were aiming to learn about. This may be considered a bit unorthodox for a project. Given, however, that this project is primarily an academic exercise, it was a wholly appropriate approach.

As detailed in the problem statement, the societal issue that we addressed was poor interoperability between emergency response agencies. The crucial technology that we used in our project as we aimed to address our chosen problem is software-defined radio (SDR). The system that we designed can be loosely divided into 3 main modules.

The first module is the radio frequency (RF) and DSP front end. This part consists of an antenna connected to a software defined radio. The software defined radio is a transceiver that enables much of the signal processing which was traditionally done in hardware to now be performed in software. In the purest form, an SDR would consist of an antenna connected analog-to-digital converter which feeds the digital data to a processor. All necessary signal detection and filtering would then be done in the digital domain. The SDR that we used for our project, had minimal analog filtering and an AGC in the analog domain in addition to the analog-to-digital converter (ADC). After the analog signal is digitized, in-phase (I) and quadrature (Q) digital data is streamed via Universal Serial Bus to a processor capable of performing DSP. In our case we are using our laptops to apply the DSP algorithms on the data. Our selected SDR is shown in figure 12. The SDR uses a Cypress FX3 USB 3.0 integrated circuit to push I and Q data via USB to the laptop. The use of the SDR enables the achievement of classification of multiple modulation schemes. This is because the SDR passes digital I and Q data from any and every signal that it receives. This fact provides for easy digital implementation of symbol timing synchronization, carrier recovery, and equalization algorithms. For classification of the modulation scheme, no apriori knowledge is required just a constellation tracking algorithm (such as we used second semester), matched filtering (used first semester), or any other applicable classification algorithm.

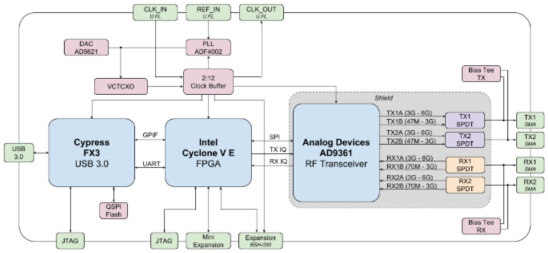


Figure 12: Block diagram of BladeRF 2.0 micro xA9 SDR[5]

The second main module provides network connectivity to our SDR and DSP front end product. The purpose of the networking stage is to give an embedded device(s) the capability to communicate on a single network. For the purposes of our system, it is the intermediary stage between the digital signal processing and the user interface. This is the server side and database system which serves as a link between the RF/DSP side and what the user sees on the user interface. The basic composition of this module consists of three TCP/IP server related programs in Python on the web server Node JS program side. Also a fourth TCP/IP python program is executed on the processor that is directly connected to the SDR. Although the user may never be cognizant of the back end programs running in order to display signal information, the networking aspect of the system is necessary for achievement of the features that focus on the user experience. A visual representation of the networking aspect of our project is given in figure 13.

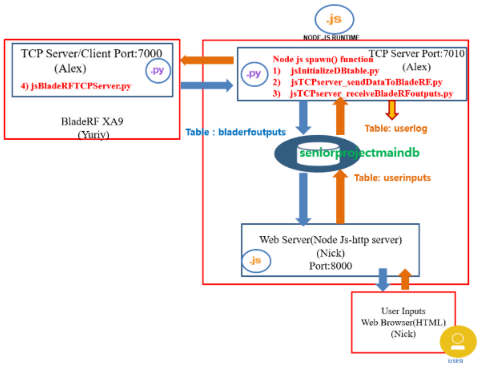


Figure 13: Networking module [9]

The third module is the user interface itself. This is where the target user is able to view valuable information. The interface consists of two parts, the back-end web development and the front end web development. Each of these systems work together to pull SQL information from the MySQL database and present it to the user in a clean, easy to work with manner. The backend web development has be done in the node.js run-time engine. The purpose and goal of this third module was to provide a pleasurable experience to the user who is not necessarily familiar with the rigor of digital signal processing or the particulars of embedded systems but knows how to access a website.

IX. Deployable Prototype Status

Next we will discuss, at length, our deployable prototype status. We will highlight the extent at which we are testing each component, the duration over which that testing will occur, has occurred, and the team member assigned to that individual test. Tests will occur at both physical (SDR) and software (Web Server) layers. Each test will have simple pass or fail measurements unless otherwise stated. All timeline dates will fall within our project deadline, and should give enough leeway incase of emergency under which we have time to restart testing. At this point all tests prior to March 25th have been successful. Most testing will be done in parallel by different group members, however, some test will require that members work together to assess the situation.

A. Classification Testing Results

Currently the system is able to classify the correct signal from down the hall of the 3rd floor of Riverside to the senior design lab. Approximately 50 meters. Since the classification seems to work through walls this is a good sign that it will work in the field. In upcoming tests we will test transmission at a greater distance. Thus determining exactly how far the team can reliably meet our metric from. It was a surprise that it still worked at the distance that it did.

A custom GNU radio block has been written that successfully interacts with both Yuriy and Alex's code. It can be seen in Figure 1 as "Classification Block" Thus the classification part of the project has been successfully implemented from MatLab to GNU radio. This significantly reduces the cost of the project from needing a $10000 MatLab License to a free install of GNU radio.

In the process of converting the algorithm to GNU radio a problem occurred that significantly hinders the accuracy of the method. Unfortunately GNU radio has a subprocess that restarts the loop for it's custom blocks every 2048 data points. Which is a significant drop from the 107 data points that were being used in the monte-carlo algorithm in the MatLab version. Since all monte-carlo methods rely on large data, the accuracy of the algorithm has decreased significantly. In the coming weeks we plan on constructing a logging method such that each packet of 2048 data points can be stored and we can retrieve the accuracy of the system from last semester. On a positive note more modulation schemes including BPSK,QPSK,8PSK,16PSK, 16QAM, 7 and different phased versions of the previously mentioned schemes have been added. Due to the drawback of the low number of data points. The team is sticking to only classifying QPSk and 8PSK for our deployable prototype until the limit on data stored can be resolved. Being able to classify two schemes is still within our feature set.

B. Detecting the modulation scheme of a signal

The implementation of the signal conditioning algorithms necessary for more accurate modulation classification was a success. The testing results show that each requisite algorithm is functioning as desired and yielding data that has a higher signal-to-noise ratio (SNR). The testing process took a long time because each algorithm was first tested using simulated data and then using real-world, transmitted data that is received by the SDR. IN the simulation environment, things are always easier.

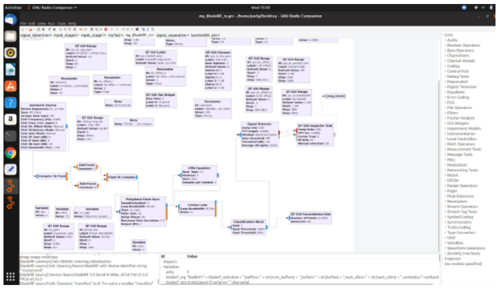


Figure 14: GNU Radio Companion DSP flow graph on Yuriy's computer [9]

Figure 2.1 shows the carrier compensation, timing synchronization, and energy detection algorithms implemented in GNU Radio Companion on Yuriy's computer.

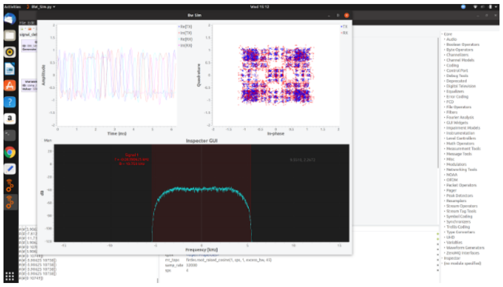


Figure 15: Simulation of Energy Detection Algorithm on Yuriy's computer [9]

Figure 2.2 shows the results of the signal conditioning algorithms applied to a QPSK signal in a simulated environment. The simulation modeled channel effect by introducing noise to the QPSK signal. It can be seen from the constellation plot (right-hand corner, blue plot) that the algorithms greatly increased the SNR. Also, the bottom quadrant shows successful energy detection and bandwidth measurement of the signal.

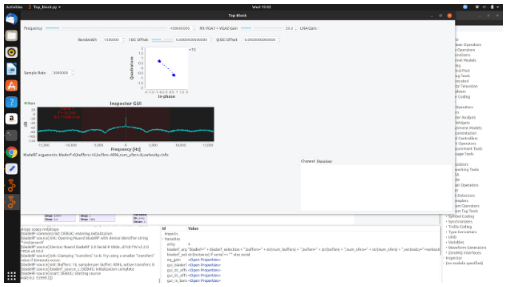


Figure 16: Results of algorithms applied to real-world QPSK signal [9]

C. *Efficient Processing Time for Signal Separation & Detection Testing Timeline*

To achieve efficient processing for the signal conditioning algorithms is a little bit difficult because of the limitation of the processors used. Also, "efficient" is a relative term. Thus far, for testing efficiency we have been transmitting a signal using a transmitter and using the bladeRF with GNU Radio to detect the signal's presence.We still need to accurately record the time it takes from receiving the signal to performing all necessary signal conditioning before actual classification. This testing will be carried out the first week of April.

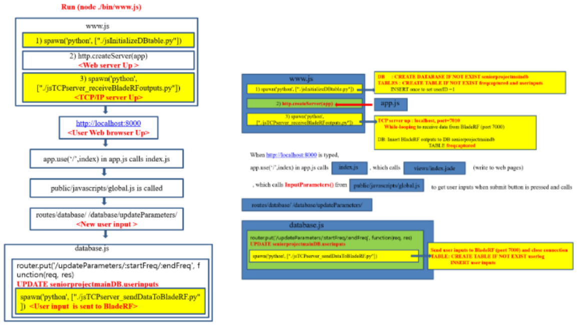
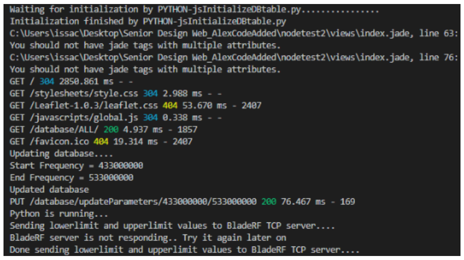

Run (node ./bin/www.js)
www.js
1) spawn('python', ["./jsInitializeDBtable.py"])
2) http.createServer(app)
<Web server Up >
3) spawn('python', ["./jsTCPserver_receiveBladeRFoutputs.py"])
<TCP/IP server Up>
http://localhost:8000
<User Web browser Up>
app.use('/',index) in app.js calls index.js
public/javascripts/global.js is called
routes/database/ /database/updateParameters/
<New user input >
database.js
router.put('/updateParameters/:startFreq/:endFreq', function(req, res)
UPDATE seniorprojectmainDB.userinputs
spawn('python', ["./jsTCPserver_sendDataToBladeRF.py"])
<User input is sent to BladeRF>
www.js
1) spawn('python', ["./jsInitializeDBtable.py"])
2) http.createServer(app)
3) spawn('python', ["./jsTCPserver_receiveBladeRFoutputs.py"])
DB : CREATE DATABASE IF NOT EXIST seniorprojectmaindb
TABLES : CREATE TABLE IF NOT EXIST freqcaptured and userinputs
INSERT once to set userID =1
app.js
TCP server up : localhost, port=7010
While-looping to receive data from BladeRF (port 7000)
DB: Insert BladeRF outputs to DB seniorprojectmaindb
TABLE freqcaptured
When http://localhost:8000 is typed,
app.use('/',index) in app.js calls
index.js
, which calls
views/index.jade
(write to web pages)
, which calls InputParameters() from
public/javascripts/global.js
to get user inputs when submit button is pressed and calls
routes/database/ /database/updateParameters/
database.js
router.put('/updateParameters/:startFreq/:endFreq', function(req, res)
UPDATE seniorprojectmainDB.userinputs
spawn('python', ["./jsTCPserver_sendDataToBladeRF.py"])
Send user inputs to BladeRF (port 7000) and close connection
TABLE: CREATE TABLE IF NOT EXIST userlog
INSERT user inputs

Figure 17: White Box Testing Plan [9]

Followed by my white box testing plan (Figure 4), I'm checking the time interval from cold start (initial server startup) to user input ready state. After user input, second testing part is going to be timing between user input and BladeRF output to database. All timing testing is done though program runtime (real time program execution state) which means I'm using python built in timing library in order to time the TCP communication.

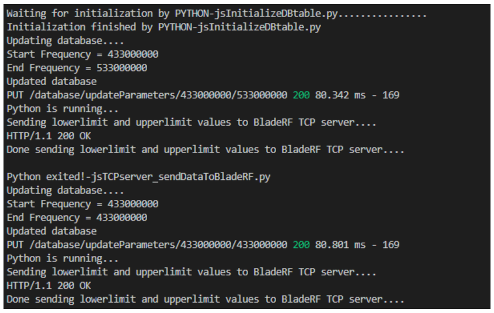
Before we go on testing results, I will point out my initial result from fall semester and what have improved over spring semester by showing TCP communication output with timing reports. You will notice huge improvements on communication speed more than 10 times faster compare to fall semester.



```
Waiting for initialization by PYTHON-jsInitializeDBtable.py...............
Initialization finished by PYTHON-jsInitializeDBtable.py
C:\Users\issac\Desktop\Senior Design Web_AlexCodeAdded\nodetest2\views\index.jade, line 63:
You should not have jade tags with multiple attributes.
C:\Users\issac\Desktop\Senior Design Web_AlexCodeAdded\nodetest2\views\index.jade, line 76:
You should not have jade tags with multiple attributes.
GET / 304 2850.861 ms - -
GET /stylesheets/style.css 304 2.988 ms - -
GET /Leaflet-1.0.3/leaflet.css 404 53.670 ms - 2407
GET /javascripts/global.js 304 0.338 ms - -
GET /database/ALL/ 200 4.937 ms - 1857
GET /favicon.ico 404 19.314 ms - 2407
Updating database....
Start Frequency = 433000000
End Frequency = 533000000
Updated database
PUT /database/updateParameters/433000000/533000000 200 76.467 ms - 169
Python is running...
Sending lowerlimit and upperlimit values to BladeRF TCP server....
BladeRF server is not responding.. Try it again later on
Done sending lowerlimit and upperlimit values to BladeRF TCP server....
```

Figure 18: Communication Output Fall [9]

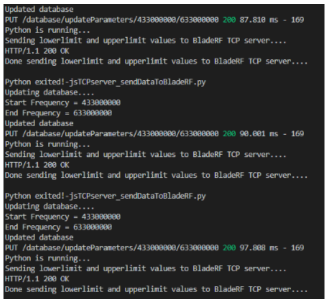
This chart is the very first communication system from fall semester at October 25th. As you can see initial start up time is 2850.861ms which is inefficient and to slow for wireless communication that meets our feature matrix.



```
Waiting for initialization by PYTHON-jsInitializeDBtable.py...............
Initialization finished by PYTHON-jsInitializeDBtable.py
Updating database....
Start Frequency = 433000000
End Frequency = 533000000
Updated database
PUT /database/updateParameters/433000000/533000000 200 80.342 ms - 169
Python is running...
Sending lowerlimit and upperlimit values to BladeRF TCP server....
HTTP/1.1 200 OK
Done sending lowerlimit and upperlimit values to BladeRF TCP server....

Python exited!-jsTCPserver_sendDataToBladeRF.py
Updating database....
Start Frequency = 433000000
End Frequency = 433000000
Updated database
PUT /database/updateParameters/433000000/433000000 200 80.801 ms - 169
Python is running...
Sending lowerlimit and upperlimit values to BladeRF TCP server....
HTTP/1.1 200 OK
Done sending lowerlimit and upperlimit values to BladeRF TCP server....
```

Figure 19: Fall TCP Timing [9]

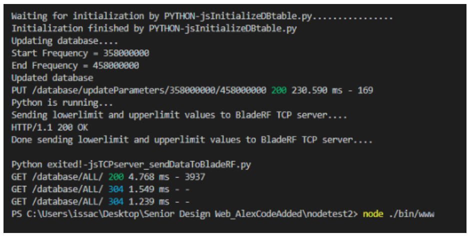
This is the time elapse from user input to bladeRF output, as you notice it only takes average 80ms to complete the one data transaction. In order to improve initial server startup time, I had to change the TCP communication topology (different way of communication order). After altering the topology I have increased communication time by approximately 7ms. However, I have improved initial startup time down to 200ms which is 10 times faster than the fall semester.



```
Updated database
PUT /database/updateParameters/433000000/633000000 200 87.810 ms - 169
Python is running...
Sending lowerlimit and upperlimit values to BladeRF TCP server....
HTTP/1.1 200 OK
Done sending lowerlimit and upperlimit values to BladeRF TCP server....

Python exited!-jsTCPserver_sendDataToBladeRF.py
Updating database....
Start Frequency = 433000000
End Frequency = 633000000
Updated database
PUT /database/updateParameters/433000000/633000000 200 90.001 ms - 169
Python is running...
Sending lowerlimit and upperlimit values to BladeRF TCP server....
HTTP/1.1 200 OK
Done sending lowerlimit and upperlimit values to BladeRF TCP server....

Python exited!-jsTCPserver_sendDataToBladeRF.py
Updating database....
Start Frequency = 433000000
End Frequency = 633000000
Updated database
PUT /database/updateParameters/433000000/633000000 200 97.808 ms - 169
Python is running...
Sending lowerlimit and upperlimit values to BladeRF TCP server....
HTTP/1.1 200 OK
Done sending lowerlimit and upperlimit values to BladeRF TCP server....
```

Figure 20: Spring TCP Timing Results [9]

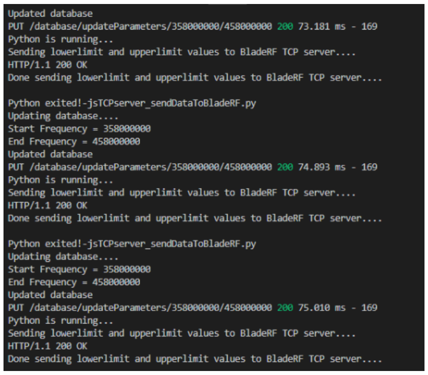
As I mentioned it, after altering TCP communication topology, user input and BladeRF output time interval has increased around 87 to 97ms.



```
Waiting for initialization by PYTHON-jsInitializeDBtable.py................
Initialization finished by PYTHON-jsInitializeDBtable.py
Updating database....
Start Frequency = 358000000
End Frequency = 458000000
Updated database
PUT /database/updateParameters/358000000/458000000 200 230.590 ms - 169
Python is running...
Sending lowerlimit and upperlimit values to BladeRF TCP server....
HTTP/1.1 200 OK
Done sending lowerlimit and upperlimit values to BladeRF TCP server....

Python exited!-jsTCPserver_sendDataToBladeRF.py
GET /database/ALL/ 200 4.768 ms - 3937
GET /database/ALL/ 304 1.549 ms - -
GET /database/ALL/ 304 1.239 ms - -
PS C:\Users\issac\Desktop\Senior Design Web_AlexCodeAdded\nodetest2> node ./bin/www
```

Figure 21: Altered Topology Results [9]

However, as a return I get the huge improvements on server startup time and database write time up to 10 times faster at the February 13th.



```
Updated database
PUT /database/updateParameters/358000000/458000000 200 73.181 ms - 169
Python is running...
Sending lowerlimit and upperlimit values to BladeRF TCP server....
HTTP/1.1 200 OK
Done sending lowerlimit and upperlimit values to BladeRF TCP server....

Python exited!-jsTCPserver_sendDataToBladeRF.py
Updating database....
Start Frequency = 358000000
End Frequency = 458000000
Updated database
PUT /database/updateParameters/358000000/458000000 200 74.893 ms - 169
Python is running...
Sending lowerlimit and upperlimit values to BladeRF TCP server....
HTTP/1.1 200 OK
Done sending lowerlimit and upperlimit values to BladeRF TCP server....

Python exited!-jsTCPserver_sendDataToBladeRF.py
Updating database....
Start Frequency = 358000000
End Frequency = 458000000
Updated database
PUT /database/updateParameters/358000000/458000000 200 75.010 ms - 169
Python is running...
Sending lowerlimit and upperlimit values to BladeRF TCP server....
HTTP/1.1 200 OK
Done sending lowerlimit and upperlimit values to BladeRF TCP server....
```

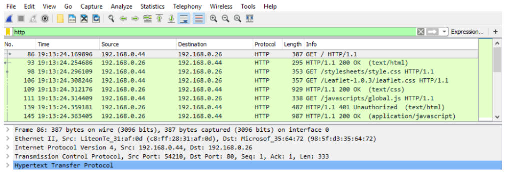
Figure 22: Current TCP Timing 10x Improvment [9]

After merging Yuriy and Cory side of the project using my TCP communication system, I also worked on maximizing time interval from user input and printing BladeRF result to web server. Unfortunately, since Yuriy and Cory changed their working environment to linux with GNU radio (python), I had rewrite the whole new communication protocol from scratch but I was lucky enough to find out one big mistake from fall semester during that reconstruction. As a result, while keeping the improved startup time and database write time, I was able to make an efficient communication protocol that only requires around 74ms average per user data transaction.

GPS module turn out to be very simple, we made a geolocator program that gets the ip address and returns the approximated GPS location using internet network either Wifi or 4G LTE network. Field testing is scheduled at April 15th. Therefore, I do not have the result at the point of submitting this report.

D. *Quick User Interface Load times*

In this testing phase, WireShark (an industry grade packet sniffer) was used to get the load times from the initial HTTP GET request to an 200 OK reply. we included two of the runs below, but a total of 10 were used for testing. You can see the initial start of the http GET as “GET / HTTP/1.1” and the reply info as “HTTP/1.1 200 OK (text/html)”. You will noticed in Figures 4.1 and 4.2 that there are multiple requests being sent.



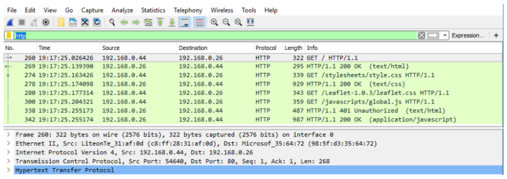
http

No.	Time	Source	Destination	Protocol	Length	Info
86	19:13:24.169896	192.168.0.44	192.168.0.26	HTTP	387	GET / HTTP/1.1
93	19:13:24.254686	192.168.0.26	192.168.0.44	HTTP	295	HTTP/1.1 200 OK (text/html)
98	19:13:24.296109	192.168.0.44	192.168.0.26	HTTP	353	GET /stylesheets/style.css HTTP/1.1
106	19:13:24.308246	192.168.0.44	192.168.0.26	HTTP	357	GET /Leaflet-1.0.3/leaflet.css HTTP/1.1
109	19:13:24.312176	192.168.0.26	192.168.0.44	HTTP	929	HTTP/1.1 200 OK (text/css)
111	19:13:24.314409	192.168.0.44	192.168.0.26	HTTP	338	GET /javascripts/global.js HTTP/1.1
139	19:13:24.359181	192.168.0.26	192.168.0.44	HTTP	487	HTTP/1.1 401 Unauthorized (text/html)
145	19:13:24.363405	192.168.0.26	192.168.0.44	HTTP	987	HTTP/1.1 200 OK (application/javascript)

```
> Frame 86: 387 bytes on wire (3096 bits), 387 bytes captured (3096 bits) on interface 0
> Ethernet II, Src: LiteonTe_31:af:0d (c8:ff:28:31:af:0d), Dst: Microsof_35:64:72 (98:5f:d3:35:64:72)
> Internet Protocol Version 4, Src: 192.168.0.44, Dst: 192.168.0.26
> Transmission Control Protocol, Src Port: 54210, Dst Port: 80, Seq: 1, Ack: 1, Len: 333
> Hypertext Transfer Protocol
```

Figure 23: User Interface GET Results 1 [9]

The multiple requests indicate the grabbing on the html, css, and lastly the javascript. This can almost be seen when rendering web pages that the javascript comes last.


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http
Expression...
+

```
No.      Time             Source         Destination    Protocol  Length Info
  260 19:17:25.026426  192.168.0.44   192.168.0.26   HTTP      322 GET / HTTP/1.1
  269 19:17:25.139390  192.168.0.26   192.168.0.44   HTTP      295 HTTP/1.1 200 OK  (text/html)
  274 19:17:25.163426  192.168.0.44   192.168.0.26   HTTP      339 GET /stylesheets/style.css HTTP/1.1
  278 19:17:25.174098  192.168.0.26   192.168.0.44   HTTP      929 HTTP/1.1 200 OK  (text/css)
  280 19:17:25.177314  192.168.0.44   192.168.0.26   HTTP      343 GET /Leaflet-1.0.3/leaflet.css HTTP/1.1
  300 19:17:25.204321  192.168.0.44   192.168.0.26   HTTP      359 GET /javascripts/global.js HTTP/1.1
  338 19:17:25.255173  192.168.0.26   192.168.0.44   HTTP      487 HTTP/1.1 401 Unauthorized  (text/html)
  342 19:17:25.255174  192.168.0.26   192.168.0.44   HTTP      987 HTTP/1.1 200 OK  (application/javascript)

> Frame 260: 322 bytes on wire (2576 bits), 322 bytes captured (2576 bits) on interface 0
> Ethernet II, Src: LiteonTe_31:af:0d (c8:ff:28:31:af:0d), Dst: Microsof_35:64:72 (98:5f:d3:35:64:72)
> Internet Protocol Version 4, Src: 192.168.0.44, Dst: 192.168.0.26
> Transmission Control Protocol, Src Port: 54640, Dst Port: 80, Seq: 1, Ack: 1, Len: 268
> Hypertext Transfer Protocol
```

Figure 24: User Interface html,css, and javascript Results [9]

In addition, you will noticed that the source and destination IP's are different. This is because we are running the web server off of my laptop which is set on IP 192.168.0.26, and my desktop computer which was used for testing was running on 192.168.0.44. This is representation in the Source and Destination columns of the results. After gathering all the load times, We estimated an average of $\frac{1}{4}$ of a second. This is counting from the initial HTTP GET for the html to the very last HTTP OK from the javascript. This testing was gathered using 10 different WireShark runs, 8 of which are not included as images above. This testing was done by Nicholas Bergamini

Information Displayed to the User The information displayed by the user was testing in a less calculatable way. We used various members of our senior design class to view the web page and give feedback on how it looks and if the text is readable. We also verified the correctness of our data based on what is in the SQL database. Out of the 4 group members we talked to, we found that all 4 were happy with the way our web page look and felt. The only comment we got was that the yellow highlight on hover was a bit bright, so we may look to adjust that. In terms of database accuracy, we didn't find any issues or strange characters. This survey-type testing was completed by Nicholas Bergamini.

Interagency operability is a time sensitive operation. The team wants to ensure that in the event of natural disasters that our system is reliable and operates within a reasonable amount of time such that first responders can focus on saving lives rather than how to communicate with each other. To ensure that our device meets this desire, the team will be testing the range at which the system is reliable, the ease of use for the user interface, and the max number of units in the area that our system can account for that meets our metrics. Through the tests described in this report, the team will be able to determine the number of devices that will need to be operational in the area in order for the system to be reliable (meet our metrics). The worst case is that the system will be updated with new information for the agents to use every 15 min. Regarding the schemes used by the other agents in the area, the accuracy to which we know the location of other teams being used in the emergency area and their communication channel and scheme. During the period of time that the GPS is being tested for long range, the device will also be tested for weathering such that of an actual disaster. The device will be used outdoors. With the device tests mentioned in this report our team has ensured that our deployable prototype is ready and functional.

X. Deployable Prototype Marketability Forecast

The Marketability Forecast serves as the foundation for planning a strategy for marketing a product in order to maximize profit. In other words the marketability forecast enables a company to develop a business plan[6]. Through our marketability forecast, we will better be able to determine what strategy has the highest probability of working and which strategies are likely to fail. In order to have a holistic view of our market, our team decided to break the marketability forecast into four distinct sections, which serve to provide a comprehensive analysis. The first section of the marketability forecast focuses on the product consumer. The consumer in our case is any emergency response team/agency. This report provides further insight into these agencies. The second section of the report focuses on the competition. This includes a look at other current and developing solutions to the problem of a lack of easy interoperability across emergency response teams. The third section of the report highlights the uniqueness of our product in comparison to existing products. The fourth and final section of the market review is the proposed advertising plan, which is developed as a result of the insights gained from the first three sections.

To gain a deep understanding of the market, we recognize that research alone will not suffice. Literature research is, of course, part of the process, but we will be interviewing personnel from organizations that serve as the target consumer of our product [6]. This insight is invaluable, as it will give us a better understanding of the customer demands. Although our team majors are not related to any part of marketing, it is possible to academically and professionally approach market review through understanding the market analysis. Still, there is no perfect way to analyze market but business can follow some rules like geographic, demographic, psychographic, and behavioral. A good market review should be sustainable, accessible, measurable, and differentiable[6] which means it has to make sense supported by many evidence from research backgrounds.

For our device to be useful to FEMA it must meet and aid the above three challenges. FEMA's current solution is to deploy temporary emergency communications vehicles (ECVs). Inside of each ECV is a small generator for charging the equipment of the first responders, radios for interoperability back to the ECV, and work stations for operators to sit at and manage the communications amongst the teams. It is in the ECV that our device has a purpose for being deployed. The following figure shows the work stations in the ECV that our system would be deployed in. The target end user of our device is the operators in the ECV. Rather than legacy radios being handed out to each team for communication back to the operators and amongst each other. SDR's can be used which will allow for each team to communicate with the other team's legacy radios and contains a GPS of where each team is at. Our system can keep an up to date system of where each team is at through GPS. Scan the area for which teams are in the local area that the operators can communicate with even if they are on legacy radios. It also generates a local database of the emergency teams in the area along with their scheme such that operators do not need to flip through a massive manual of all the teams in the

US to determine these facts. Our system reduces the time needed for determining how to communicate which gives operators and teams more time to focus on allocating their forces for saving lives in a disaster. Having a deployable field ready version of what the operators use with the power of a laptop, but on a single board computer in the field also solves the problem of teams being able to communicate with each other with a sudo operator if they can not communicate back to the ECV in the field.

When a natural disaster hits or some other emergency situation occurs, emergency response teams are forced to cross collaborate with each other. With every team and organization having their own land mobile radio system and pre-selected channels and modulation scheme, interoperability becomes very difficult. The current solution calls for each agency to have a thick manual handy that can be consulted for information about channel and modulation scheme used for different agencies. The problem here is that the handbook does not actually give information on which exact teams are actually in the area trying to mitigate the damage of the situation. In addition, the process of consulting a handbook is very time consuming. Emergency situation often cannot afford this waste of time of sifting through pages upon pages. Our product serves to automate this entire process and provide a centralized system which can be consulted by all participating agencies and teams.

Our product has the unique ability to pick up signal in the affected area and provide real time information about only the agencies that are present in the area. This information is available on a secure website, which only agencies have the proper credentials to access. No more handbooks for consulting! Our system is also scalable. Due to the software nature of our system, updates can be made to our product days, weeks, months, and even years after it has already been in use by a consumer. This scalability and reconfigurability is enabled due to the increasing used technology of software-defined radio.

The need is clear, communication interoperability takes time and reducing the amount of time needed for this allows first responders to focus on saving lives. Our system addresses this need by picking up signals in the affected area and providing near real time information about only the agencies that are present in the area. This information is available on a secure website, on a locally generated server, which only agencies have the proper credentials to access. Once FEMA's Disaster Emergency Communications team is deployed they will be able to communicate with teams already in the area and focus on saving lives rather than flipping through a manual. The emergency communications vehicle is the central part of the relief effort for team interoperability. Our device would operate in these vehicles and in the field with first responders. Which will allow the system to operate over the greatest range and not leave any teams out that are in the region. While AT&T has made a deal with the U.S. government to build First Net, combating interagency communications issues. Perhaps our system could assist them in their mission as well. At the end of the day, the goal is to design a device that helps save lives.

XI. Conclusion

This two semester senior project has come to its conclusion. The team experience of going through challenging tasks for about 8 months has been fulfilling. Sometimes we had to stay in Senior Design Lab for over 7 hours straight, work through weekends, or even sacrificing our time during holidays and breaks. Although, during the beginning of the first semester, we were not sure about our goals and abilities to reach the desired outcome to complete a deployable prototype, we were ultimately able to overcome these challenges and successfully formed meaningful goals for our feature metrics and achieved these goals.

This document provided a thorough outline of our team's selected societal problem. It was detailed how the wide variety of frequency and modulation options, restrictions, and multiple wireless devices manufacturers all contribute to a communication breakdown between emergency response teams attempting to collaborate with each other.

We also presented a justification for the use of our selected technology, namely software defined radio. SDR, or Software Defined Radio, has proven to be an excellent resource in combating outlined issues. SDR is a wireless communications device that minimizes hardware by enabling key features which were traditionally implemented in hardware to be implemented in software. There is a common consensus that Software Defined Radio may be the leading solution to the issue of a lack of interoperability between local, state, and federal agencies. In addition, security risks were discussed and an explanation of the authentication process was explained.

In order to maximize our team productivity during the course of our project, we developed certain flexible documents that helped us anticipate unforeseen setbacks and pitfalls. We created a work breakdown structure that describes the implementation of seven features that are to be incorporated into the system. Every individual on the team had a unique part to work on, and everyone divided their part into subtasks and activities. These features range from signal processing of data to timing requirements to displaying useful information to a user through a web server. The activities serve to describe the implementation of subtasks necessary for the achievement of the features. By organizing each feature and detailing the associated subtasks and activities, the group was more effective in time management and troubleshooting. In accord with the work breakdown structure, we developed a timeline for the duration of the project.

Our team's End of Term Documentation also includes a risk assessment which was an attempt to identify the aspects or components of our project that have the potential to cause a deviation from the overall goal of the project. These aspects or components were rated as to the likelihood of hazard occurrence and the impact that will be incurred. This provided for the overall quantification of the risk level. The aspects that were assessed were effectively the features that comprise our product. These features are neatly divided among the electrical engineering students and the computer engineering students. The electrical engineering aspect involves all the signal processing. This includes signal conditioning and modulation classification. For the modulation to have a good level of accuracy, there are signal conditioning algorithms that must first operate on the received data. The hazards and risks associated with these algorithms are analyzed. The modulation identification algorithms are also analyzed for their potential pitfalls and the impact of those pitfalls.

This document also provides a design overview which details the design philosophy of our project. In addition we included a deployable prototype marketability forcecast detailing what is still needed to achieve a marketable device and a status update detailing the performance of our system. This performance was determined through various testing schemes.

References

[1] “Camp Fire: The Local Police and Firefighters Who Responded to the Deadliest Wildfire.” KXTV, KXTV, 14 Nov. 2018, www.abc10.com/article/news/local/paradise/camp-fire-the-local-police-and-firefighters-who-responded-to-the-deadliest-wildfire/103-614571187.

[2] Bush, E. (2015, August 21). Type of person drawn to wildland firefighting might be called a ’tactical athlete’. Retrieved from https://www.seattletimes.com/seattle-news/northwest/type-of-person-drawn-to-wildland-firefighting-might-be-called-a-tactical-athlete/

[3] Huyck, N. (2015). Interoperability challenges among public safety radio systems: Surveyed research addressing these challenges. International Journal of Business and Public Administration (IJBPA), 12(1), 12.

[4] “Public Safety Spectrum.” Federal Communications Commission, 18 Sept. 2018, www.fcc.gov/public-safety/public-safety-and-homeland-security/policy-and-licensing-division/public-safety-spectrum.

[5] https://www.nuand.com/product/bladerf-xA9/?gclid=Cj0KCQjwwODlBRDuARIsAMy_28WnKeZ1o6qzZfuSaDY95UkMofo4wfPvbJik-qG7cY42Nky15SLmiygaAmamEALw_wcB

[6] Gowell, J., McWherter, S., Gowell, Scott, & McWherter, Jeff. (2012). Professional mobile application development. Indianapolis, Ind.: Wiley.

[7] Raghu Chandra Garmimella,Dr.V.R.Sastry,Mohammed Shoeb Mohiuddin(2015)”Piezo-Gen An Approach to Generate Electricity from Vibrations”, Procedia Earth and Planetary Science.

[8] ANDREW S. TANENBAUM. COMPUTER NETWORKS. PRENTICE HALL. pp. 131–132. ISBN 0-13-212695-8.

[9] Bergamini, Nicholas; Han, Alexander; Dzyuba, Yuriy; Diehl, Cory. Produced by one or more members of CSUS 2018 - 2019 Senior Design Team Two.

.

Glossary

HTML- Hypertext Markup Language
CSS- Cascading Style Sheet
HTTP- Hypertext Transfer Protocol
SDR - Software Defined Radio
PSD - Power Spectral Density
RF - Radio Frequency
FPGA - Field Programmable Gate Array
API - Application Programming Interface

APPENDIX A
USER MANUAL

The interagency operability system is relatively easy to set up. There are 3 parts that come in the kit: a laptop, bladeRF, and antenna. The laptop is the main processing unit that manages the server which you will use, the bladeRF manages the data being received, and the antenna receives the communications signal from the local area. All must be connected and powered on in order for the system to run properly.

Steps for set up:

1) Connect the antenna to the Rx1 (receive SMA connector) of the blade RF. For optimal lifetime of the product: turn the antenna left (loose) until a click is heard or felt, then screw the antenna onto the connector until it is finger tight.

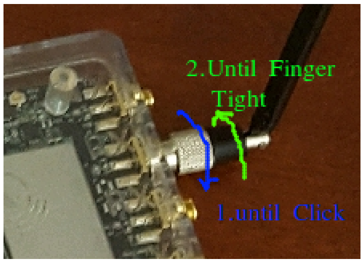


Figure A.25: Connecting Antenna [9]

2) Plug the BladeRF into the laptop's USB port.

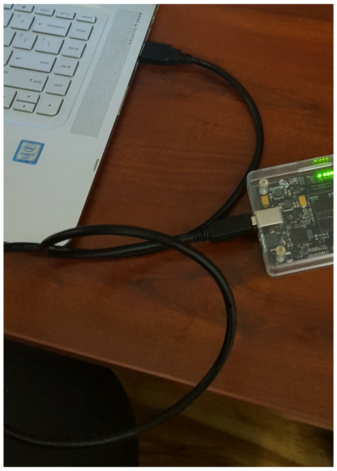


Figure A.26: Connecting BladeRF to Laptop [9]

3) Plug the Laptop into a wall charger.

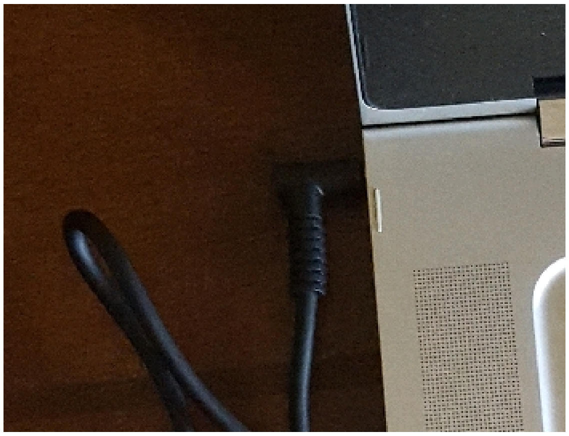


Figure A.27: Connecting Laptop to Wall Charger [9]

The system is now set up and ready to be turned on. Simply press the power button of the laptop and let the system run in the background. There is no need to interact with the device until the emergency event is over.

User Interface

The user interface can be accessed on any device.

User Interface Steps:

1) Go to your favorite internet browser and type the following into the URL (10.117.170.50.8000) . The google chrome browser is shown in the example below.

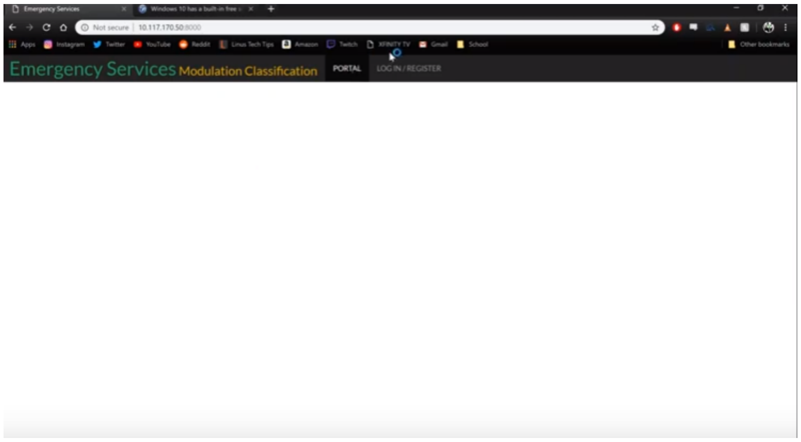
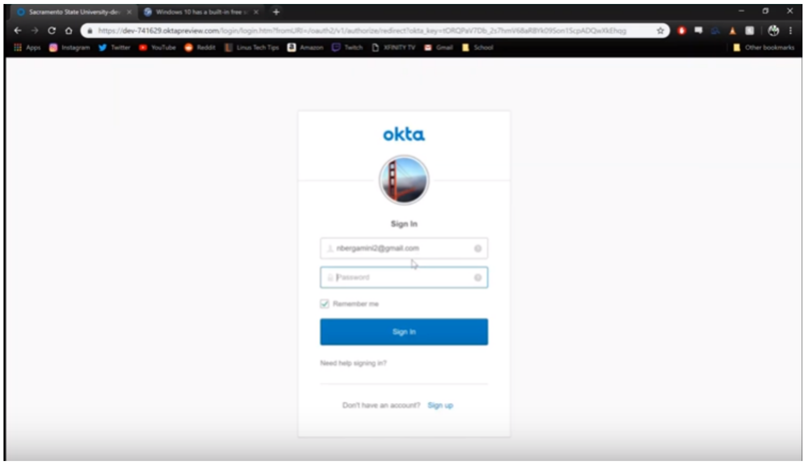


Figure A.28: Getting to the user interface [9]

2) Click Login/Register and login to your account.



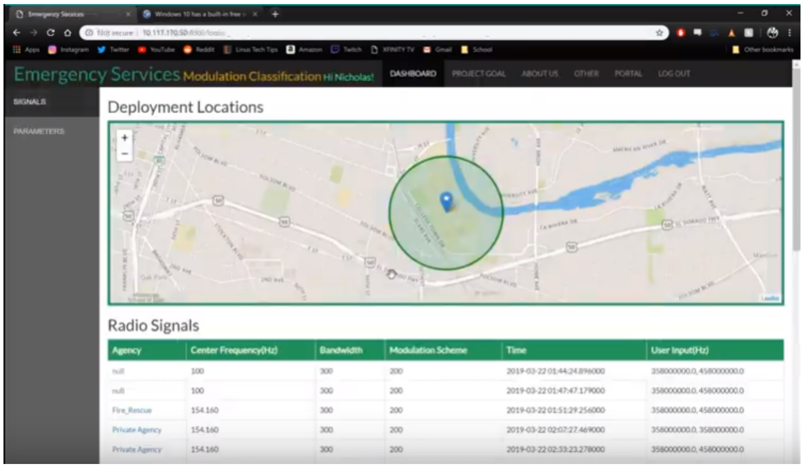
nbergamini2@gmail.com

Password

☑ Remember me

Figure A.29: Log into your account. [9]

3) If everything is successful you will be looking at the following main interface. Which has a map of the area displaying where other devices are deployed, and a list of communication schemes.



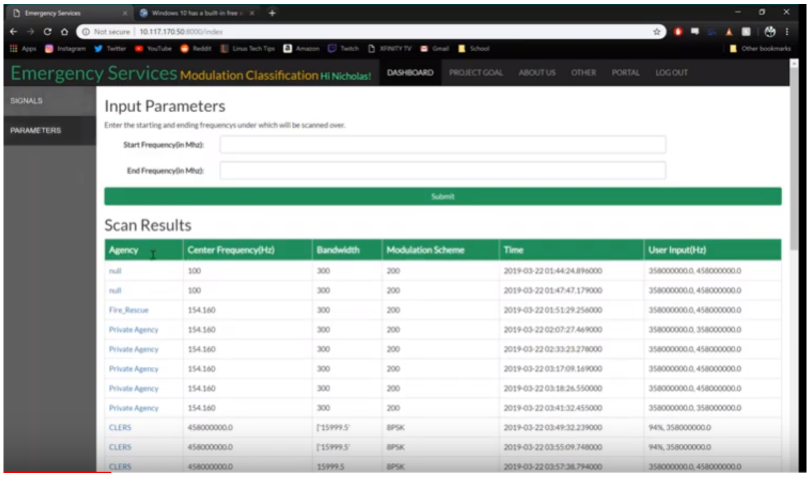
Deployment Locations

Radio Signals

Agency	Center Frequency(Hz)	Bandwidth	Modulation Scheme	Time	User Input(Hz)
null	100	300	200	2019-03-22 01:44:24.896000	358000000.0, 458000000.0
null	100	300	200	2019-03-22 01:47:47.179000	358000000.0, 458000000.0
Fire_Rescue	154.160	300	200	2019-03-22 01:51:29.256000	358000000.0, 458000000.0
Private Agency	154.160	300	200	2019-03-22 02:07:27.469000	358000000.0, 358000000.0
Private Agency	154.160	300	200	2019-03-22 02:33:23.278000	358000000.0, 458000000.0

Figure A.30: User Interface Main Screen [9]

4) Click on Parameters on the top right of the page. This will replace the map with two boxes to enter in the frequencies that you would like to sweep across.


Emergency Services
Windows 10 has a built-in free
Not secure | 10.117.170.50:8000/index
Apps Instagram Twitter YouTube Reddit Linus Tech Tips Amazon Twitch XFINITY TV Gmail School
Other bookmarks
Emergency Services Modulation Classification Hi Nicholas!
DASHBOARD
PROJECT GOAL
ABOUT US
OTHER
PORTAL
LOG OUT
SIGNALS
PARAMETERS
Submit

Input Parameters

Enter the starting and ending frequencys under which will be scanned over.

Start Frequency(in Mhz):

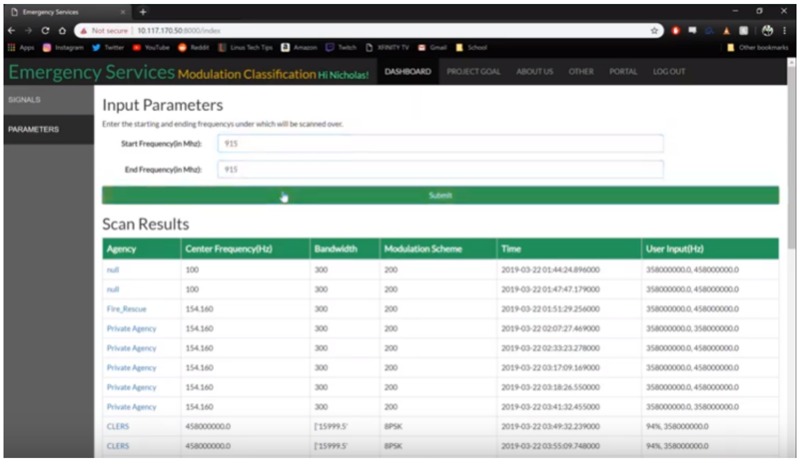
End Frequency(in Mhz):

Scan Results

Agency	Center Frequency(Hz)	Bandwidth	Modulation Scheme	Time	User Input(Hz)
null	100	300	200	2019-03-22 01:44:24.896000	358000000.0, 458000000.0
null	100	300	200	2019-03-22 01:47:47.179000	358000000.0, 458000000.0
Fire_Rescue	154.160	300	200	2019-03-22 01:51:29.256000	358000000.0, 458000000.0
Private Agency	154.160	300	200	2019-03-22 02:07:27.469000	358000000.0, 358000000.0
Private Agency	154.160	300	200	2019-03-22 02:33:23.278000	358000000.0, 458000000.0
Private Agency	154.160	300	200	2019-03-22 03:17:09.169000	358000000.0, 458000000.0
Private Agency	154.160	300	200	2019-03-22 03:18:26.550000	358000000.0, 458000000.0
Private Agency	154.160	300	200	2019-03-22 03:41:32.455000	358000000.0, 358000000.0
CLERS	458000000.0	['15999.5'	8PSK	2019-03-22 03:49:32.239000	94%, 358000000.0
CLERS	458000000.0	['15999.5'	8PSK	2019-03-22 03:55:09.748000	94%, 358000000.0
CLERS	458000000.0	15999.5	8PSK	2019-03-22 03:57:38.794000	358000000.0, 458000000.0

Figure A.31: User Interface Parameters Screen [9]

5) Enter the frequency you would like to start the sweep at and the frequency you would like to stop the sweep at. Then click submit. Note: the same frequency can be used for the start and stop frequency.


Emergency Services
Not secure | 10.117.170.50:8000/index
Apps Instagram Twitter YouTube Reddit Linus Tech Tips Amazon Twitch XFINITY TV Gmail School
Other bookmarks
Emergency Services Modulation Classification Hi Nicholas!
DASHBOARD
PROJECT GOAL
ABOUT US
OTHER
PORTAL
LOG OUT
SIGNALS
PARAMETERS
Submit

Input Parameters

Enter the starting and ending frequencys under which will be scanned over.

Start Frequency(in Mhz): 915

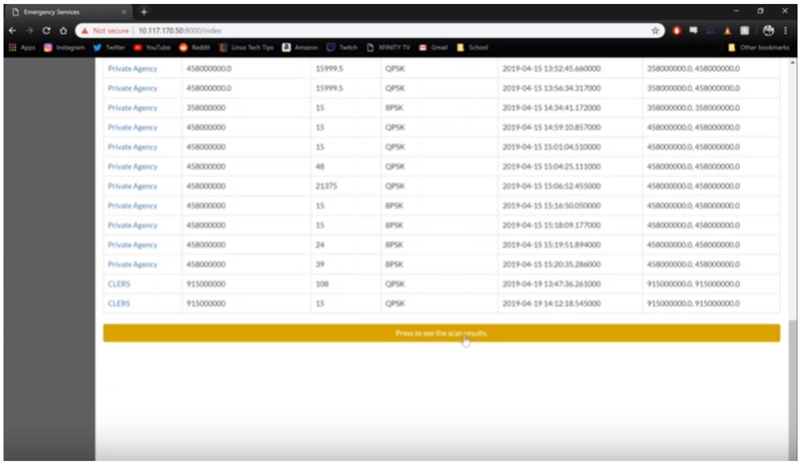
End Frequency(in Mhz): 915

Scan Results

Agency	Center Frequency(Hz)	Bandwidth	Modulation Scheme	Time	User Input(Hz)
null	100	300	200	2019-03-22 01:44:24.896000	358000000.0, 458000000.0
null	100	300	200	2019-03-22 01:47:47.179000	358000000.0, 458000000.0
Fire_Rescue	154.160	300	200	2019-03-22 01:51:29.256000	358000000.0, 458000000.0
Private Agency	154.160	300	200	2019-03-22 02:07:27.469000	358000000.0, 358000000.0
Private Agency	154.160	300	200	2019-03-22 02:33:23.278000	358000000.0, 458000000.0
Private Agency	154.160	300	200	2019-03-22 03:17:09.169000	358000000.0, 458000000.0
Private Agency	154.160	300	200	2019-03-22 03:18:26.550000	358000000.0, 458000000.0
Private Agency	154.160	300	200	2019-03-22 03:41:32.455000	358000000.0, 358000000.0
CLERS	458000000.0	['15999.5'	8PSK	2019-03-22 03:49:32.239000	94%, 358000000.0
CLERS	458000000.0	['15999.5'	8PSK	2019-03-22 03:55:09.748000	94%, 358000000.0

Figure A.32: Entering Parameters for Sweep [9]

6) Scroll to the bottom of the page and click update to see if any new signals that were captured by the scan or other scans from other deployed devices.



Private Agency	458000000.0	15999.5	QPSK	2019-04-15 13:52:45.660000	358000000.0, 458000000.0
Private Agency	458000000.0	15999.5	QPSK	2019-04-15 13:56:34.317000	358000000.0, 458000000.0
Private Agency	358000000	15	8PSK	2019-04-15 14:34:41.172000	358000000.0, 358000000.0
Private Agency	458000000	15	QPSK	2019-04-15 14:59:10.857000	458000000.0, 458000000.0
Private Agency	458000000	15	QPSK	2019-04-15 15:01:04.510000	458000000.0, 458000000.0
Private Agency	458000000	48	QPSK	2019-04-15 15:04:25.111000	458000000.0, 458000000.0
Private Agency	458000000	21375	QPSK	2019-04-15 15:06:52.455000	458000000.0, 458000000.0
Private Agency	458000000	15	8PSK	2019-04-15 15:16:50.050000	458000000.0, 458000000.0
Private Agency	458000000	15	8PSK	2019-04-15 15:18:09.177000	458000000.0, 458000000.0
Private Agency	458000000	24	8PSK	2019-04-15 15:19:51.894000	458000000.0, 458000000.0
Private Agency	458000000	39	8PSK	2019-04-15 15:20:35.286000	458000000.0, 458000000.0
CLERS	915000000	108	QPSK	2019-04-19 13:47:36.261000	915000000.0, 915000000.0
CLERS	915000000	15	QPSK	2019-04-19 14:12:18.545000	915000000.0, 915000000.0

Figure A.33: Update the database [9]

APPENDIX B
HARDWARE

The hardware for our system is relatively simple because all the components are commercial, off the shelf (C.O.T.S) parts. Our system consists of four main components. Three components (antenna, SDR, and one laptop) are for the signal detection and processing. The fourth component (a second laptop) is used for running a server. The basic system level block diagram is given in figure B.1.

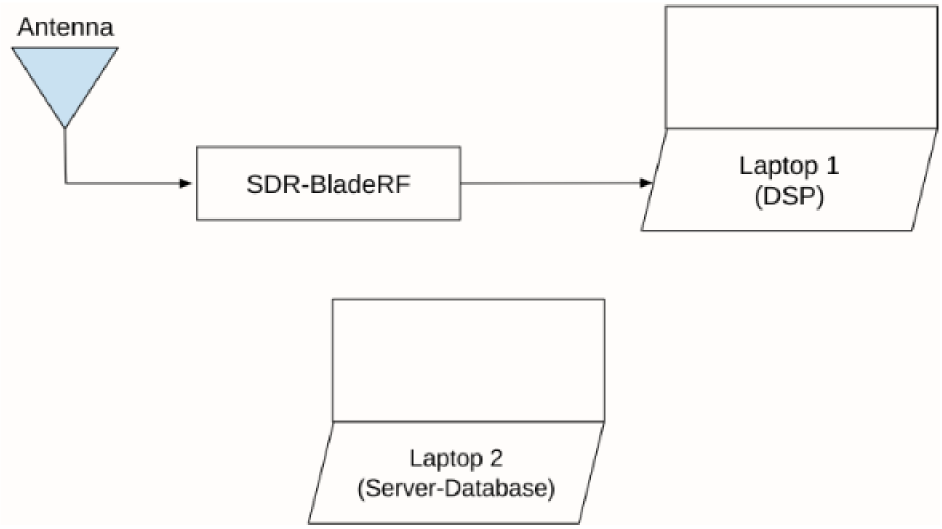


Figure B.1: System Hardware Block Diagram [9]

The antenna we are using for receiving is an ANT500 telescopic antenna which is designed for 75 MHz to 1 GHz operation. The antenna is made of stainless steel and has a male SMA connector. The ANT500 is designed to impedance match hobbyist SDRs with 50 ohms and is adjustable from 20 cm to 88 cm. When testing the antenna, we learned that it perform well in the low-band UHF (450-806 MHz) part of the spectrum. It performs poorly, however, in the high-band UHF (900-952 MHz) part of the spectrum. This affects the modulation classification stage.

The software-defined radio that we used is the bladeRF 2.0 micro xA9. The frequency range on the built-in RF transceiver is 47 MHz to 6 GHz with a 61.44 MHz sampling rate on the ADC. At the heart of the BladeRF 2.0 micro xA9 is the Cyclone V FPGA from Intel. For the purposes of our project, we did not utilize this FPGA for signal processing. Rather our signal processing is performed on a laptop. The SDR uses a Cypress FX3 integrated circuit for USB 3.0. This enables high speed and low latency data transfer to the laptop. The basic architecture of the BladeRF is given in the block diagram of figure B.2.

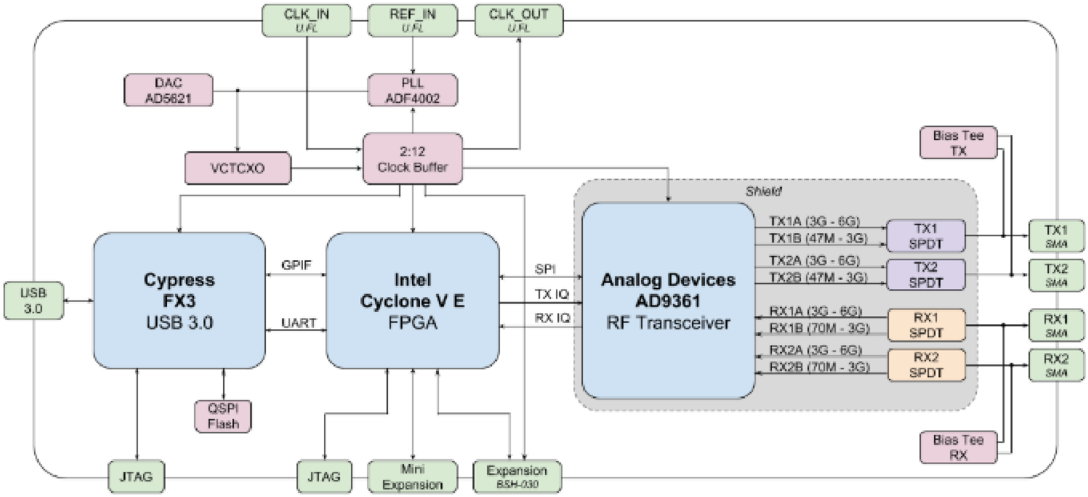


Figure B.2: Block diagram of BladeRF 2.0 micro xA9[5]

APPENDIX C
SOFTWARE

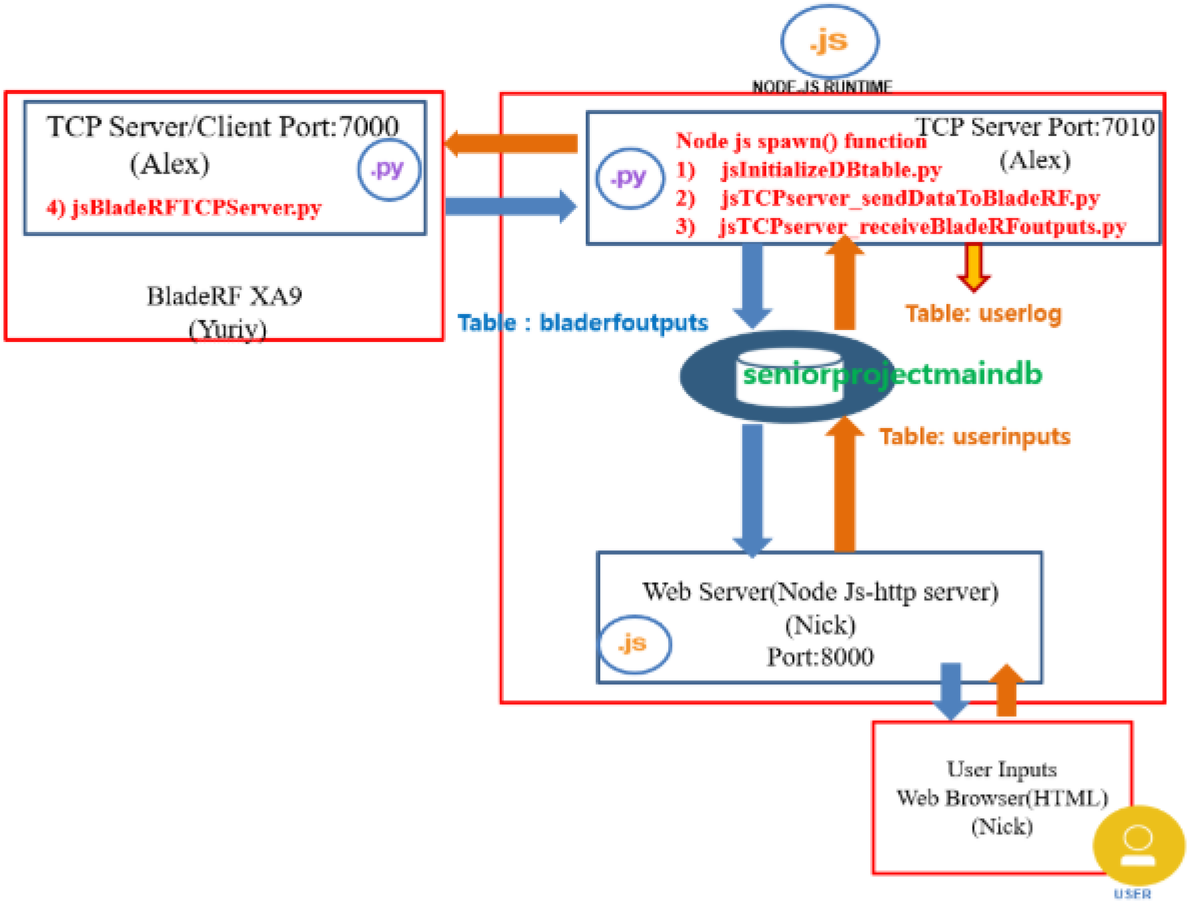
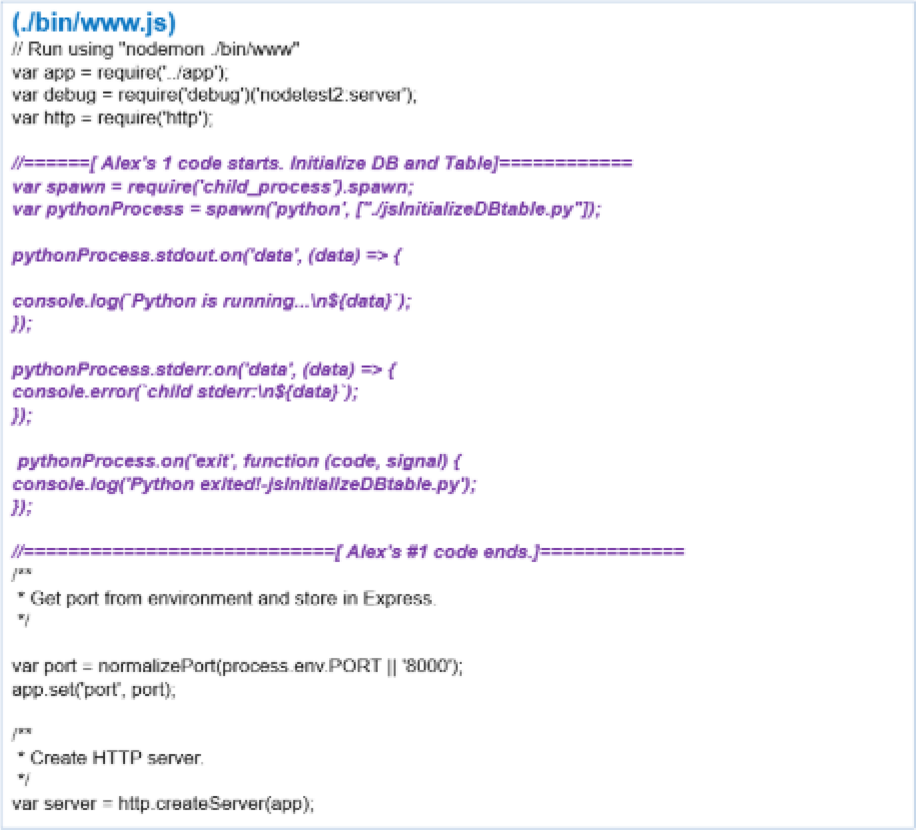


Figure C.1: Visualized deployable Prototype in software aspect [9]

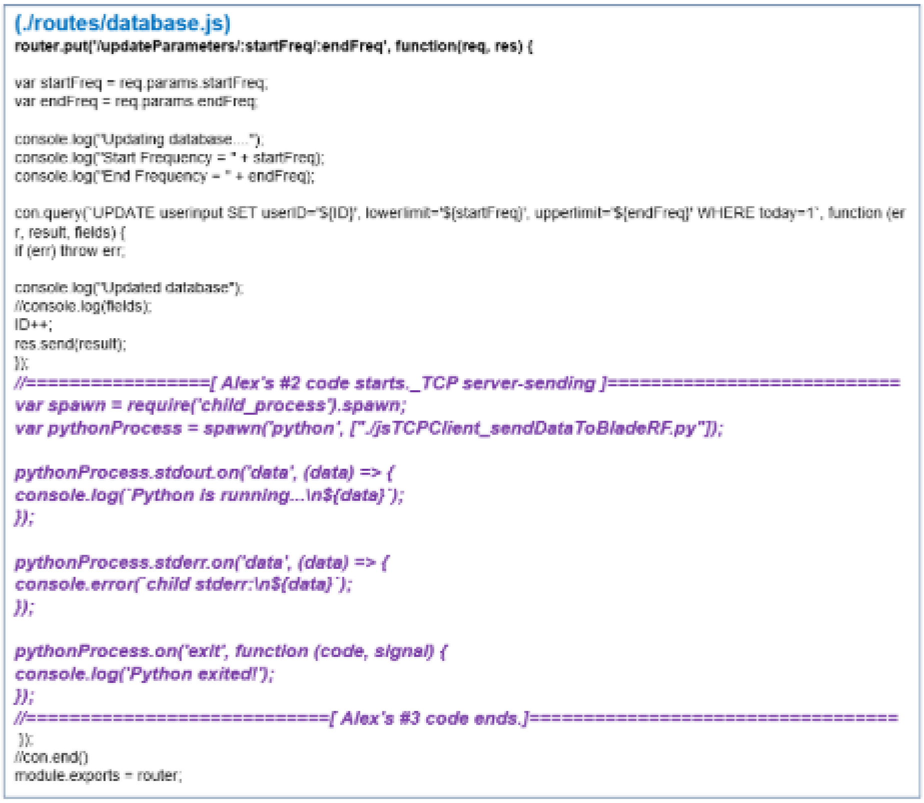
Alex's code #1 in Web server code



```
(./bin/www.js)
// Run using "nodemon ./bin/www"
var app = require('../app');
var debug = require('debug')('nodetest2:server');
var http = require('http');

//======[ Alex's 1 code starts. Initialize DB and Table]============
var spawn = require('child_process').spawn;
var pythonProcess = spawn('python', ["./jsInitializeDBtable.py"]);

pythonProcess.stdout.on('data', (data) => {

console.log(`Python is running...\n${data}`);
});

pythonProcess.stderr.on('data', (data) => {
console.error(`child stderr:\n${data}`);
});

 pythonProcess.on('exit', function (code, signal) {
console.log('Python exited!-jsInitializeDBtable.py');
});

//============================[ Alex's #1 code ends.]=============
/**
 * Get port from environment and store in Express.
 */

var port = normalizePort(process.env.PORT || '8000');
app.set('port', port);

/**
 * Create HTTP server.
 */
var server = http.createServer(app);
```

Figure C.2: NodeJS Synchronization and MySQL(Database) startup integrated with web-server [9]

Alex’s code #2 in Web server code

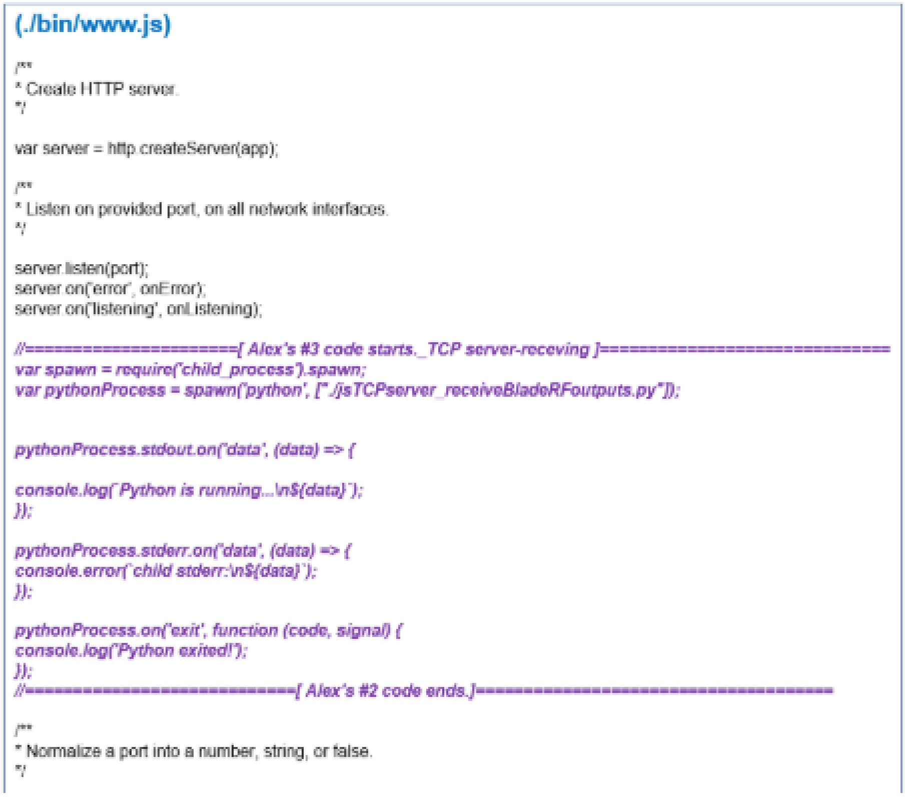


(./routes/database.js)

```
router.put('/updateParameters/:startFreq/:endFreq', function(req, res) {

var startFreq = req.params.startFreq;
var endFreq = req.params.endFreq;

console.log("Updating database....");
console.log("Start Frequency = " + startFreq);
console.log("End Frequency = " + endFreq);

con.query(`UPDATE userinput SET userID='${ID}', lowerlimit='${startFreq}', upperlimit='${endFreq}' WHERE today=1`, function (er
r, result, fields) {
if (err) throw err;

console.log("Updated database");
//console.log(fields);
ID++;
res.send(result);
});
//=================[ Alex's #2 code starts._TCP server-sending ]============================
var spawn = require('child_process').spawn;
var pythonProcess = spawn('python', ["./jsTCPClient_sendDataToBladeRF.py"]);

pythonProcess.stdout.on('data', (data) => {
console.log(`Python is running...\n${data}`);
});

pythonProcess.stderr.on('data', (data) => {
console.error(`child stderr:\n${data}`);
});

pythonProcess.on('exit', function (code, signal) {
console.log('Python exited!');
});
//============================[ Alex's #3 code ends.]====================================
 });
//con.end()
module.exports = router;
```

Figure C.3: Web-server user input protocol and TCP integration [9]

Alex's code #3 in Web server code



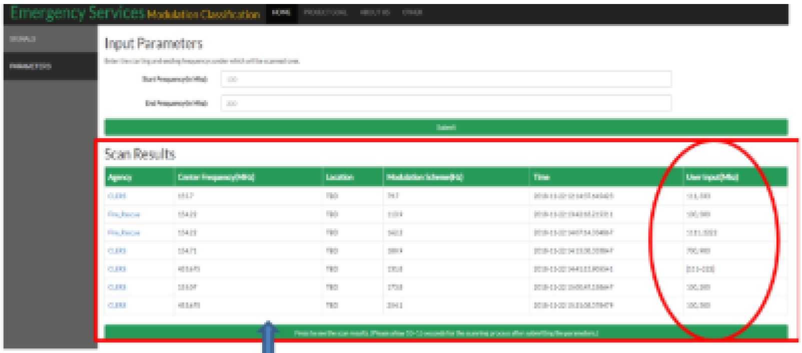
(./bin/www.js)

```
/**
* Create HTTP server.
*/

var server = http.createServer(app);

/**
* Listen on provided port, on all network interfaces.
*/

server.listen(port);
server.on('error', onError);
server.on('listening', onListening);

//======================[ Alex's #3 code starts._TCP server-receving ]==============================
var spawn = require('child_process').spawn;
var pythonProcess = spawn('python', [" ./jsTCPserver_receiveBladeRFoutputs.py"]);


pythonProcess.stdout.on('data', (data) => {

console.log(`Python is running...\n${data}`);
});

pythonProcess.stderr.on('data', (data) => {
console.error(`child stderr:\n${data}`);
});

pythonProcess.on('exit', function (code, signal) {
console.log('Python exited!');
});
//=============================[ Alex's #2 code ends.]========================================

/**
* Normalize a port into a number, string, or false.
*/
```

Figure C.4: HTTP and TCP receiving-end integration[9]

(November, 2018)

#5: Added by Alex on 11/20/2018



Input Parameters

Scan Results

Agency	Center Frequency(MHz)	Location	Modulation Scheme(Hz)	Time	User Input(Mhz)
CLERS	[illegible]	TBD	[illegible]	[illegible]	[illegible]
Fire_Rescue	[illegible]	TBD	[illegible]	[illegible]	[illegible]
Fire_Rescue	[illegible]	TBD	[illegible]	[illegible]	[illegible]
CLERS	[illegible]	TBD	[illegible]	[illegible]	[illegible]
CLERS	[illegible]	TBD	[illegible]	[illegible]	[illegible]
CLERS	[illegible]	TBD	[illegible]	[illegible]	[illegible]
CLERS	[illegible]	TBD	[illegible]	[illegible]	[illegible]

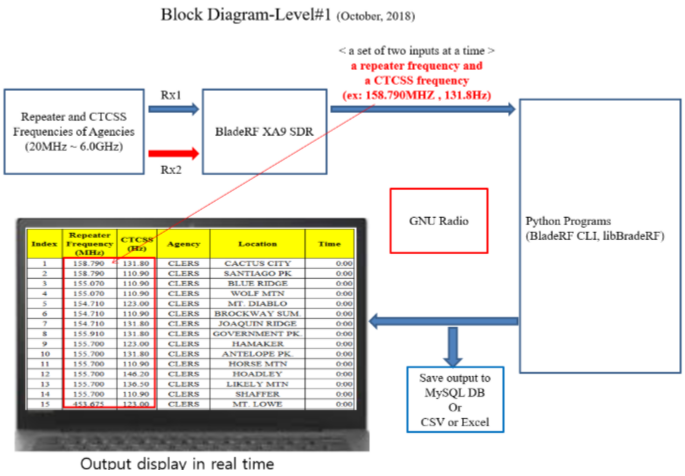
this field of the table “freqcaptured”

```
#Parameter.tabcontent
     .container-fluid
       h2 Input Parameters
       p
         | Enter the starting and ending frequencys under which will be scanned over.
       form.form-horizontal
         .form-group
           label.control-label.col-sm-2(for='startFrequency') Start Frequency(in Mhz):
           .col-sm-8
             input#startFreq.form-control(type='number')
         .form-group
           label.control-label.col-sm-2(for='endFrequency') End Frequency(in Mhz):
           .col-sm-8
             input#endFreq.form-control(type='number')
       button.btn.btn-primary.btn-block(type='button')(onclick='InputParameters()') Submit

       h2 Scan Results
       table.table.table-bordered
         thead
           tr
             th Agency
             th Center Frequency(MHz)
             th Location
             th Modulation Scheme(Hz)
             th Time
             th User Input(Mhz)
         tbody
       button.btn.btn-primary.btn-block(type='button')(onclick='gatherData()') Press to see the scan results. (Please allow 10~15 seconds for the scan
ning process after submitting the parameters.)
```

Code for this table in index.jade

Figure C.5: MySQL output and web-server display synchronization [9]

*Figures 29 to 32 show all the networking protocols merged by Alex Han.


Block Diagram-Level#1 (October, 2018)
< a set of two inputs at a time >
a repeater frequency and
a CTCSS frequency
(ex: 158.790MHZ , 131.8Hz)
Repeater and CTCSS Frequencies of Agencies (20MHz ~ 6.0GHz)
Rx1
Rx2
BladeRF XA9 SDR
GNU Radio
Python Programs (BladeRF CLI, libBradeRF)
Save output to MySQL DB Or CSV or Excel
Output display in real time

Index	Repeater Frequency (MHz)	CTCSS (Hz)	Agency	Location	Time
1	158.790	131.80	CLERS	CACTUS CITY	0:00
2	158.790	110.90	CLERS	SANTIAGO PK	0:00
3	155.070	110.90	CLERS	BLUE RIDGE	0:00
4	155.070	110.90	CLERS	WOLF MTN	0:00
5	154.710	123.00	CLERS	MT. DIABLO	0:00
6	154.710	110.90	CLERS	BROCKWAY SUM.	0:00
7	154.710	131.80	CLERS	JOAQUIN RIDGE	0:00
8	155.910	131.80	CLERS	GOVERNMENT PK.	0:00
9	155.700	123.00	CLERS	HAMAKER	0:00
10	155.700	131.80	CLERS	ANTELOPE PK.	0:00
11	155.700	110.90	CLERS	HORSE MTN	0:00
12	155.700	146.20	CLERS	HOADLEY	0:00
13	155.700	136.50	CLERS	LIKELY MTN	0:00
14	155.700	110.90	CLERS	SHAFFER	0:00
15	453.675	123.00	CLERS	MT. LOWE	0:00

Figure C.6: Hardware implementation from networking aspect [9]

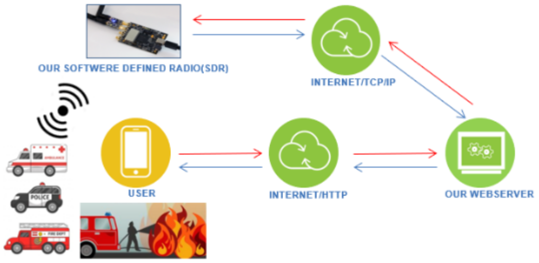


Figure C.7: Initial planning for Senior Design Project [9]

Appendix D
Mechanical

A set of antennas was produced and measured by Cory Diehl of team 2.In this section we will display the measured parameters for our antennas. There are three antenna's that were used for the project. A store bought telescoping antenna, a circular patch antenna, and a dipole antenna.

The store bought telescoping antenna was measured on the spectrum analyzer to work from $70MHz-1MHz$. The reflection coefficient S_{11} was measured to be $-3dB$ lower in the $70MHz-200MHz$ range than it did for higher frequencies. Hence why they are seen on many cars as FM radio antennas.

The circular patch antenna was determined to work for a very limited set of frequencies around 1.3GHz ,1.8GHz, and 2.1GHz. This was due to the radius of the circular patch being made just a bit too small. The following figure shows an example of measuring the reflection coefficient.

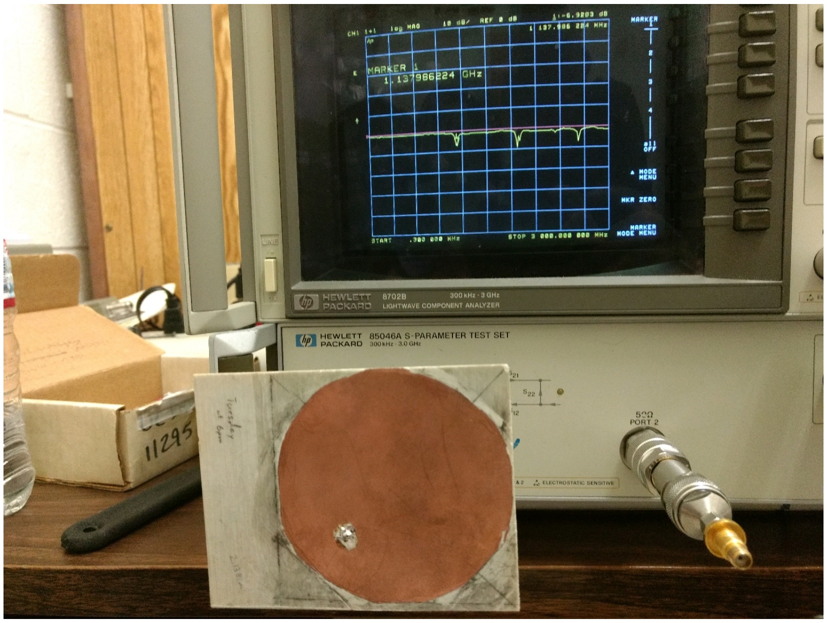

CH1 1+1 log MAG 10 dB/ REF 0 dB
1:-6.9203 dB
1 137.986 224 MHz
MARKER 1
1.137986224 GHz
START .300 000 MHz
STOP 3 000.000 000 MHz
MARKER
all OFF
MODE MENU
MKR ZERO
MARKER MODE MENU
HEWLETT PACKARD 8702B 300 kHz - 3 GHz LIGHTWAVE COMPONENT ANALYZER
HEWLETT PACKARD 85046A S-PARAMETER TEST SET 300 kHz - 3.0 GHz
50Ω PORT 2
11295

Figure D.1: Circular Patch Antenna Measurement [9]

The DiPole antenna was constructed to operate in the 915MHz since medical and technology band. Which is useful for testing as a license isn't needed. On the spectrum analyzer it was shown to have a reflection coefficient $S_{11} = -16dB$ at 915MHz with a bandwidth of 10MHz. Which means that is should work well within that band and only that band for transmission and receiving. The following figure shows the antenna being tested in the anechoic chamber.

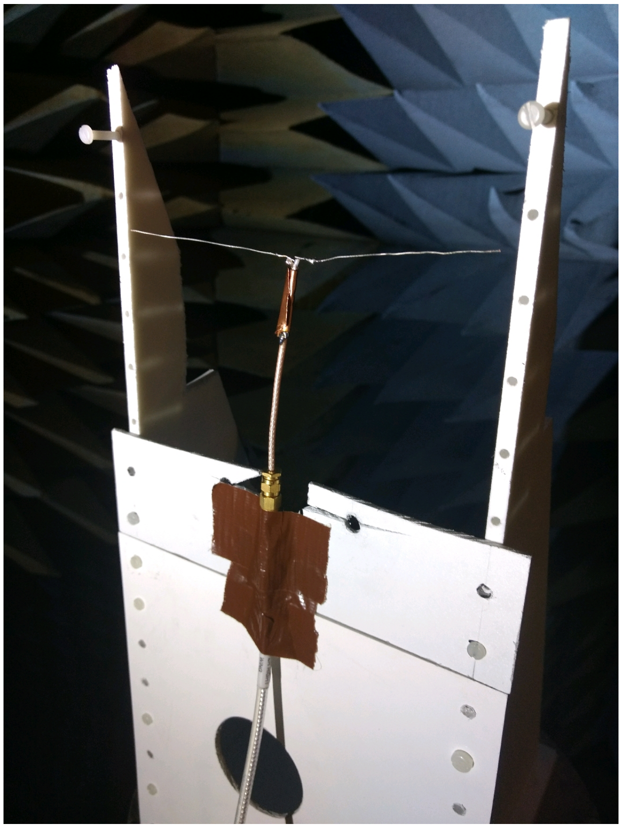


Figure D.2: DiPole Antenna Testing in the Anechoic Chamber [9]

The following figures show the beam pattern of the antenna. The shape of the beam pattern is a torus which means that the beam will be directed radially, but not perpendicularly to the antenna. Which shows why the antenna worked so well on the same floor of the building, but not when there was a height difference.

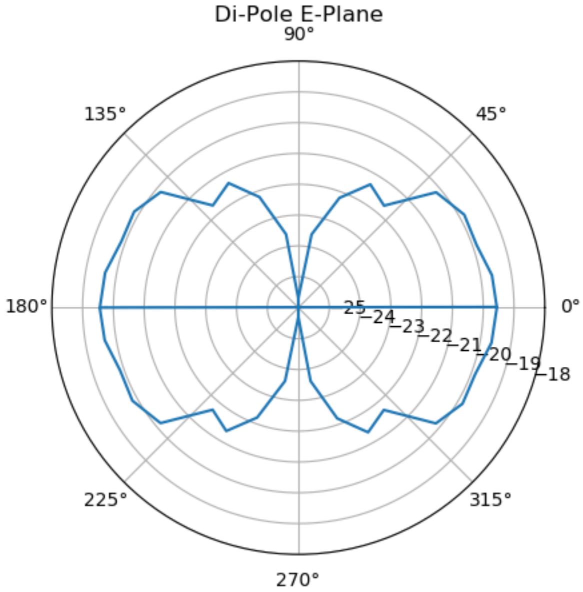


Figure D.3: Measured radiation pattern in the E-Plane [9]

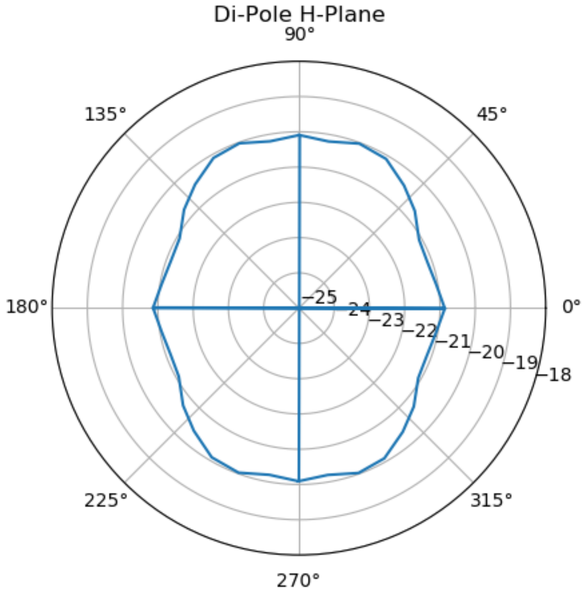


Figure D.4: Measured radiation pattern in the H-Plane [9]

Appendix E
Vendor Contacts

Nuand - Company from which we purchased two software-defined radios (both BladeRF)

Great Scott Gadgets - Company from which we purchased two ANT500 telescopic antennas

Leading Defense Contractor - A team of engineers from a leading U.S. defense contractor gave our team technical guidance

APPENDIX F
RESUMES

Alexander Han

EDUCATION

California State University Sacramento
Bachelor of Computer Engineering, expected Dec 2019
G.P.A. 3.822

Related Courses

- Hardware related
  - CMOS And VLSI
  - Advanced Logic Design
  - Advanced Computer Organization
  - Computer Hardware Design
  - Signals, Systems, and Probability
  - General Physics-Electromagnetics
  - Circuit Logics (Verilog FPGA)
  - Electronics
- Software related
  - Program Concept and Method I, II
  - Discrete Structures, Objective Oriented
  - Network Analysis
  - Computer Network and Internet
  - Computer Architecture (Assembly Intel 8086)
  - Data Structures & Algorithm Analysis
  - Senior Project Design

EXPERIENCE

GPS (Graduate Pathway to STEM)
- UC Berkeley (October 15-16, 2016)
- Stanford (October 11-12, 2018)

SKILLS

Programming Languages

- C, Assembly, Java, Javascript (Nodejs), Python

Software and Tools

- Unix/Linux, Microsoft Office

Communication and Organization

- MEP/MESA
  - Strong organizational skills developed during MEP Assembly
- EOP
  - Well-developed leadership skills through Green/Gold Certificate

* Bilingual: Korean (Very fluent in writing and speaking)

SENIOR PROJECT

- Senior Design Project: Modulation Classification System for Emergency Services.

AWARDS

Deans Honor Roll (Spring 2016, Fall 2016, Spring 2017, Fall 2017, Spring 2018, Fall 2018)

Cory Diehl

EDUCATION

California State University Sacramento
Bachelor of Electrical Engineering with a Minor Math, expected May 2019
G.P.A. 3.9

EXPERIENCE

Seirra College, Math and Physics Tutor, Rocklin Fall 2014 - Spring 2017

Helped students understand dificult concepts and misconseptions.

Engineered Medical Technologies, Lab Tech , Winter 2018
Rocklin, CA

Helped reduce statistical uncertainty in lab equipment and product.

SKILLS

Programming

- C, Python, Javascript, Assembly x86 and ARM instruction set.

Analysis

- Fourier
- Image Processing
- Statistical Error Analysis
- Least Squares.

Microprocessors

- Microprocessor Arcitecture That I'm framiliar with
  - MSP 432, MSP 430
  - ATmega
  - ARM 6,7,8
- IDEs that I'm framiliar with
  - Atmel Studio
  - Code Composer Studio
  - MPLab -X

Projects

- Wireless home lighting system with the ESP8266.
- Custom google home out of a raspberry Pi.
- Automated fans and TV with Infared communication.
- Presented Research at Senoma State Undergraduate Math Conference Fall 2018

HOBBIES

Home Automation, Kayaking, Biking, Reading

Nicholas Bergamini

EDUCATION

California State University Sacramento
Bachelor of Computer Engineering
GPA 3.98, Deans Honor Roll

EXPERIENCE

SMUD - IT Infrastructure & Operations Intern June 2017 - Present

Various projects related to Cloud Architecture, Automation & Scripting, ServiceNow Development, Web Development, File Security, & Mobility Solutions.

SKILLS

Languages

- POWERSHELL, PYTHON, JAVA, VISUAL BASIC
- JAVASCRIPT, C, SQL, PERL

Industry Tools

- AWS
- AZURE
- VMWARE
- MOBAXTERM
- VISUAL STUDIO
- NODE.JS

Projects

Cloud Architecture, AWS & Azure January 2019 Current

Worked with a team of industry engineers to proof of concept Commvault on AWS and NetApp ONTAP on Azure. These solutions were in conjunction with an effort to relocate backup targets from an on prem datacenter to the cloud.

Emergency Services - Modulation Classification August 2018 Current

Project aimed to combat intra-agency interoperability issues in the public safety domain through automatic modulation classification using Software Defined Radio. Implemented front end using Node.js, Bootstrap, jQuery, express, and external APIs. Sponsored by SMUD a leading defense contractor. (Senior Design Project)

Recycle App January 2017

Hack Davis 2017 Project that utilizes the Watson Image Recognition API to detect if an object inside of a picture is recyclable. Programmed using various tools such as Bootstrap, jQuery, node.js, express, and external APIs.

Storage Alert & Analytics System December 2017 March 2018

Used JavaScript, PowerShell, Google Charts API, a MSSQL Database. Created a system for which a group of industry engineers could monitor, complete, reassign, and sort incoming security or capacity alerts. The system also provides analytics for data growth, usage, and compression.

COMMUNITY

VOLUNTEERING - Russian to English Interpretation in Local Church & Family Services

ASSOCIATIONS - ACM (Association of Computer Machinery)

Yuriy Dzyuba

EDUCATION

California State University Sacramento
B.S. Electrical Engineering, expected May 2019
G.P.A. 3.9

EXPERIENCE

Electrical Engineering Intern, Elevator Controls 5/18 - 10/18

I worked with object-oriented C++ programs for ARM processors. The purpose is to automate testing and validation of multiple PCBs. Responsibilities include: reading PCB schematics to write a C++ program to test the integrity of the CAN bus communication, proper working of the analog and digital circuitry on the PCB, and to perform firmware updates on the processors.

SKILLS

Programming

- C/C++, Python, Verilog, VHDL, MATLAB

Hardware

- Oscilloscopes
- logic analyzers
- spectrum analyzers
- Circuit design
- Basic digital design with FPGA

PROJECTS

Building a Modern Computer from First Principles

I designed a 16-bit computer called Hack. Using a basic hardware description language (HDL) and given only a basic NAND gate and a data flip flop (DFF), I created basic logic gates. After creating additional logic blocks, I was able to create an ALU. Next, I created the RAM and the CPU. In the end, I combined the modules to form a complete 16-bit computer. Next, I built a software hierarchy for the 16-bit hardware platform. Using Python, I implemented a VM translator that translates virtual machine code from a stack machine abstraction to assembly language.

Emergency Services - Modulation Classification

My senior project group is working on modulation classification by utilizing software defined radio. My distinct part includes creating digitally modulated signals to transmit using an SDR, writing algorithms on the receiving end for carrier recovery and timing recovery. After these steps, the I/Q data is sent to a modulation classification algorithm. All algorithms were first implemented in MATLAB and Simulink and then in the GNU Radio environment.

Remotely Controlled Robot Vehicle

Designed and built a vehicle that was controlled over the internet and included RF communication capabilities. The vehicle included two onboard Raspberry Pis. A web server was set up on one Raspberry Pi, and an HTML webpage was used to control the vehicle over the internet. The second Raspberry Pi was connected to a 433 MHz receiver. A Python script was used to decode signals from a 433 MHz remote. A shell script was used to monitor this Python program to receive the information as to what button was pressed on the remote. Depending on which button was pressed on the remote, the shell script would call other python programs to carry out functions with the various sensors onboard the vehicle.

COMMUNITY

VOLUNTEERING - Russian to English Interpretation in Local Church

ASSOCIATIONS - Member, Tau Beta Pi National Engineering Honor Society