Milestone 5

Electromagnetic Spectrum Mapping (EMS) of a Dynamic Environment

ME 423 X2

May 17th, 2018

"I pledge my honor that I have abided by the Stevens Honor System."

John Anticev
Arunvenkatesh Aruljothi
John Grotke
Justin Holdridge
Greg McNeil
Mike Moschetti

Section - 1: Project Definition and Plan

Mission Statement

The GD Spectrum Mapping team's goal is to take advantage of the many small pockets of unused spectrum that is available in effort to maximize spectrum usage. Historical data will be collected and used to identify these white spaces. These efforts will increase the number of users that are able to use the electromagnetic spectrum at time.

Background

The electromagnetic spectrum has become increasingly congested in recent years as the number of interconnected devices has exploded. The current management of the spectrum has many issues and can be improved upon. Companies, such as cellular providers have been worried about running out of spectrum space when using the current management system.[2]

General Dynamics Mission Systems has requested a solution to map the dynamic electromagnetic spectrum environment in order to find available white space to transmit on, and use these openings to transmit data without interfering with any other users on the band. This is shown below in Figure 1.1.

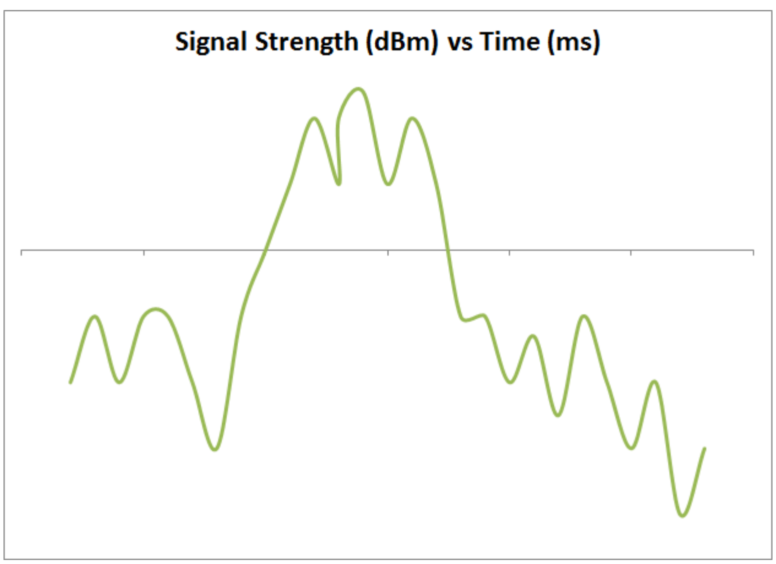


Figure 1.1: The horizontal line is a threshold to define if there is something being transmitted on a frequency or if it is idle.

Stakeholder List

- General Dynamics Mission Systems
- Department of Defense
- NSA
- CIA/FBI
- FCC

Analysis of Stakeholder Needs

A general theme between many of the companies that we researched was the need to increase bandwidth while keeping costs down. Being able to tap into the unused space in unlicensed bands would allow customers to use more frequencies while not having to purchase cost prohibitive frequency licenses. Other common requests from companies was to ensure that fast and secure communications were available for the communication between mapping units, and that the new technology would be easily integrated into existing technology.

Project Scope and Resources

The scope of this project is to construct a proof-of-concept device that will be able to be transported into a dynamic environment via UAV. Once the device has been delivered, it will begin to take samples of different frequencies that have been pre-selected and gather data on parameters such as received signal strength, time of day, location, and weather. Once a large amount of data has been recorded, the device will use software to determine the ambient noise levels at the individual frequencies, and from these observations the system will be able to recognize if a signal is being transmitted at that frequency.

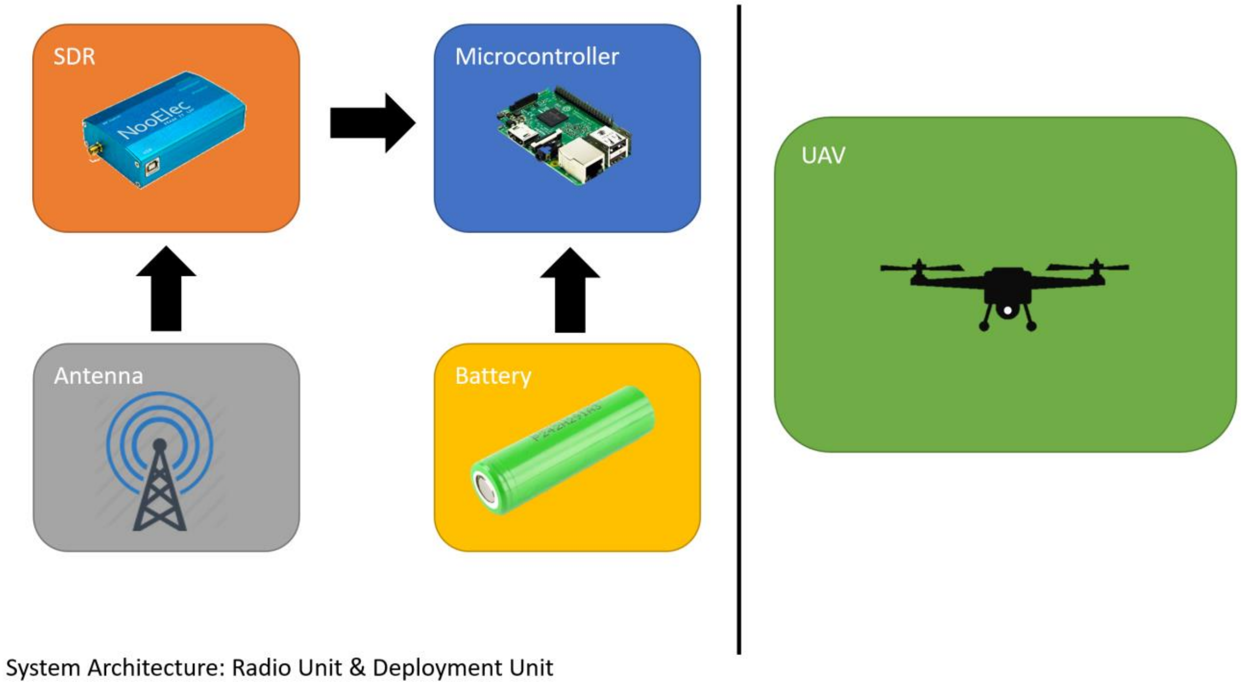


Figure 1.2: Above is the overall architecture of the project. It consists of an antenna, SDR, and microcontroller with a UAV to transport the system

Our sponsor General Dynamics Missions Systems is the team's largest resource. The team will be able to rely on the team advisor, Kishore Pochiraju, for any advice or assistance. The faculty staff and resources that are provided by Stevens Institute of Technology will also prove invaluable to this project. Additionally, there is a wealth of source code available to use through the use of the internet that will make the software challenges much easier to meet.

Project Plan

The figures 1.3, 1.4, and 1.5 below are gantt charts that display important milestones and deadlines that the group must meet. Each section was split up between the members of the project so that each member would be able to leverage their skills to sections that they are more familiar to. Some important milestones that are listed below are the design of the system, prototype construction, testing of the system, redesigning of the system, and distribution of the information collected.

	Task Name	Duration	Start	Finish	Predecessors	Resource Names
8	Design	26 days	Tue 10/3/17	Tue 11/7/17		JA,AA,JG,JH,GM,MM
9	Record data to file using laptop and RTL-SDR Mini	7 days	Tue 10/3/17	Wed 10/11/17		JA,JG,MM
10	Enclosure Preliminary Design	7 days	Tue 10/10/17	Wed 10/18/17		JH,GM
11	Drone Software Design	14 days	Tue 10/10/17	Fri 10/27/17		AA
12	Antenna Design	7 days	Tue 10/17/17	Wed 10/25/17		JA,MM,JG
13	Determine if transmission is occurring on a specific frequency	8 days	Tue 10/24/17	Thu 11/2/17		JA,MM,JG
14	Jump between frequancies at increasing rate to capture data	14 days	Tue 10/24/17	Fri 11/10/17		JA,MM,JG
15	Design enclosure for mapping units. Design drone mounts.	6 days	Thu 10/19/17	Thu 10/26/17	10	JH,GM
16	Determine useful frequencies from historical data analysis	6 days	Wed 11/1/17	Wed 11/8/17		JA,MM,JG
17	Preliminary Design Review	0 days	Tue 11/7/17	Tue 11/7/17		JA,AA,JG,JH,GM,MM
18	Prototype Construction	14 days	Tue 11/7/17	Fri 11/24/17		JA,AA,JG,JH,GM,MM

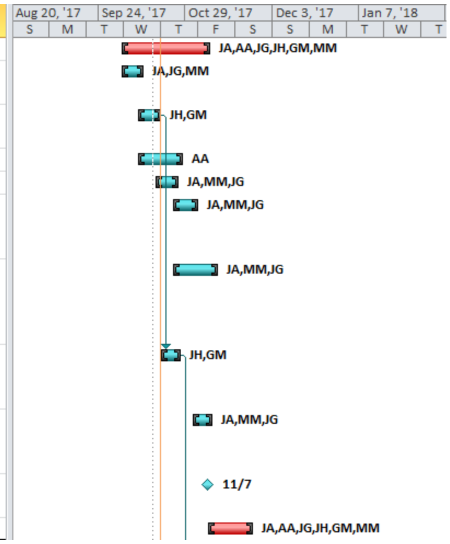


Figure 1.3: The figure above displays a gantt chart that shows important milestones for the first half of the first semester. The left section of the figure lists the important milestones and their deadlines, while the information is displayed graphically on the right.

	Task Name	Duration	Start	Finish	Predecessors	Resource Names
18	Prototype Construction	14 days	Tue 11/7/17	Fri 11/24/17		JA,AA,JG,JH,GM,MM
19	3D Print and test enclosure and drone mount	7 days	Fri 10/27/17	Mon 11/6/17	15	JH,GM
20	Purchase and integrate antennas	7 days	Tue 11/14/17	Wed 11/22/17		JG,JA,MM
21	Purchase and integrate Raspberry Pi	7 days	Tue 11/14/17	Wed 11/22/17		JA,AA,MM
22	Testing	23 days	Tue 11/21/17	Thu 12/21/17		JA,AA,JG,JH,GM,MM
23	Receive Amateur Radio Technician License	7 days	Tue 11/21/17	Wed 11/29/17		JA,AA,JG,JH,GM,MM
24	Integrate SDR Play 2 and Raspberry Pi into enclousure. Attach enclosure to Drone	16 days	Tue 11/28/17	Tue 12/19/17		JA,AA,JG,JH,GM,MM
25	Receive transmitted data on secondary receiver	16 days	Tue 11/28/17	Tue 12/19/17		JA,JG,MM
26	Alpha Prototype Presentation	0 days	Wed 12/20/17	Wed 12/20/17		JA,AA,JG,JH,GM,MM

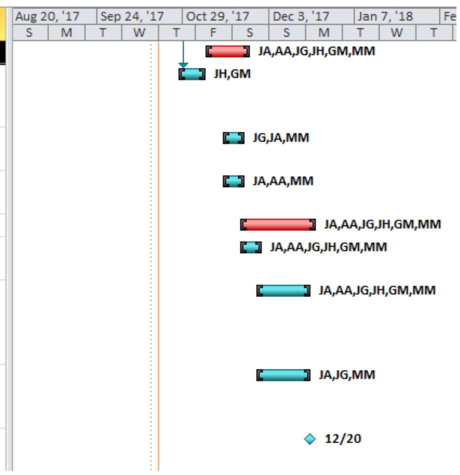


Figure 1.4: The figure above displays the gantt chart for the second half of the first semester.

	Task Name	Duration	Start	Finish	Predecessors	Resource Names
27	Redesign	42 days	Mon 1/22/18	Tue 3/20/18		JA,AA,JG,JH,GM,MM
28	Automation of drone	14 days	Mon 1/22/18	Thu 2/8/18		AA
29	Bill of Materials	6 days	Mon 1/29/18	Mon 2/5/18		AA,JH,GM
30	Create System to evaluate data	4 days	Thu 2/8/18	Tue 2/13/18		JA,GM
31	Performance and Cost Review	0 days	Tue 2/13/18	Tue 2/13/18		JA,AA,JG,JH,GM,MM
32	Obtain permit to fly on campus	6 days	Tue 2/13/18	Tue 2/20/18		GM
33	Run Optimization Test	4 days	Tue 2/20/18	Fri 2/23/18		JH,GM
34	Run Finite Element Analysis	4 days	Tue 2/20/18	Fri 2/23/18		JH,GM
35	Redesign Electronics	10 days	Tue 2/20/18	Mon 3/5/18		JA,JG,MM
36	Rebuild Prototype	17 days	Mon 3/5/18	Tue 3/27/18		JA,AA,JG,JH,GM,MM
37	Optimized Design Review	0 days	Tue 3/20/18	Tue 3/20/18		JA,AA,JG,JH,GM,MM
38	Distribution of Information	31 days	Tue 3/20/18	Tue 5/1/18		JA,AA,JG,JH,GM,MM
39	Written Analysis of findings	14 days	Tue 3/20/18	Fri 4/6/18		JA,AA,JG,JH,GM,MM
40	Create Pitch for Compitetion	16 days	Fri 4/6/18	Fri 4/27/18		GM
41	Create Presentation	18 days	Fri 4/6/18	Tue 5/1/18		JA,AA,JG,JH,GM,MM
42	Final Report and Presentation	0 days	Tue 5/1/18	Tue 5/1/18		JA,AA,JG,JH,GM,MM

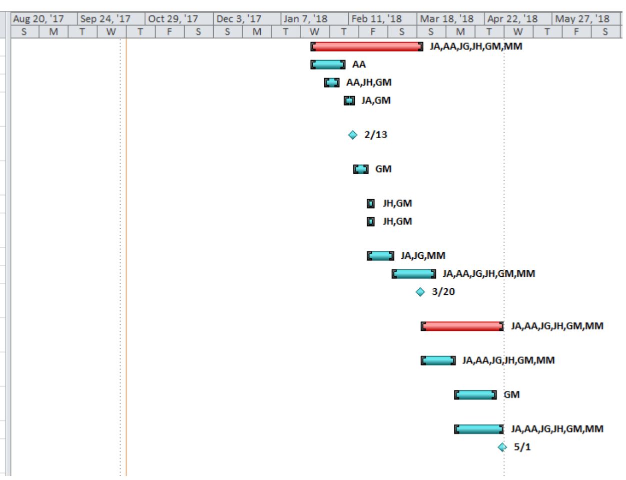


Figure 1.5: The figure above displays a gantt chart that displays important milestone information for the second semester.

Section - 2: Design, Evaluation & Optimization

Requirements

Product Perspective/Intent

This product is a system of mapping units designed to make a network. Each unit is a self contained receiver that will record historical data on unlicensed frequency bands. The collected data will be used to identify and transmit on the white spaces found in the spectrum. This system will efficiently monitor spectrum usage to transmit without interfering with existing signals. An example of white space distribution on three frequencies is shown in Figure 2.1. When the frequencies are above or below a certain threshold that will be derived through historical data, we will determine if something is being transmitted on the frequencies, or if they are idle. Each unit will be easily deployable and will withstand its operating environment. This system is intended to be used in General Dynamics communications networks, but can be utilized for other purposes such as in emergency and domestic application.

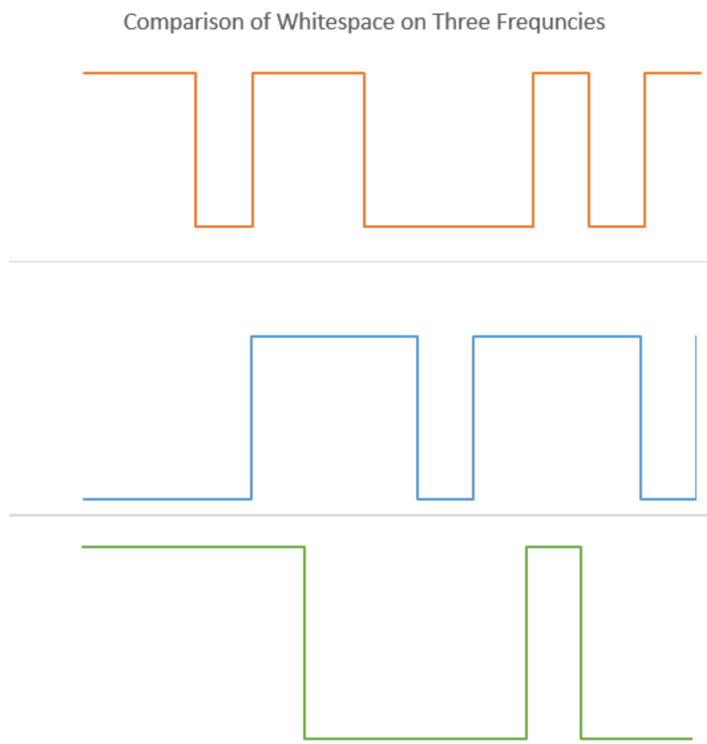


Figure 2.1: Three mock frequency bands and their available whitespace. The state will be determined by the noise threshold in Figure 1.1.

Primary Functions

The system deploys mapping units to a location through the use of a UAV. These units must be able to receive signals and record the state of the spectrum environment. The system will then be able to analyze the data in conjunction with weather, location, and time to determine optimal transmission frequencies. Information will be stored locally and will be displayed graphically. Three primary frequency bands will be selected for transmission and reception between units.

Operating and User Environments

The system is designed to be used in either civilian or military applications. The environment can be any location with a congested or contested electromagnetic spectrum. Depending on the online time of the system, a static power source may be required. The system will be all weather proof. The system will require training in custom software designed to interface with a software defined radio (SDR).

System Requirements

Table 2.1: System Functional Requirements Table

#	Functional Requirement	Associated Specification	Weight
1	System shall operate independently and have its own power source	Battery Capacity	10%
2	The units can be operated in any desired environment	Product Enclosure Rating	10%
3	System must record and analyze historical data	Data storage and processor speed	35%
4	Must transmit and receive with minimal loss of data	Transmission success rate	35%
5	System must be able to deploy radio units to non accessible areas	UAV Range and Battery Life	10%

System Functional Requirements Description

The system must be able to transmit and receive data on the EM Spectrum, and must remain transmitting and receiving for a period without interruption. The battery must be able to supply the static ground unit with enough power to operate and transmit. There shall not be any data loss when transmitting within the white spaces found in the spectrum when transmitting and receiving RF signals. In addition, the unit must be able to survive any and all weather elements occupying the given environment. Upon deployment of this device, it will start recording the environment's frequency data. The deployment system of the mapping unit must be able to gain access to areas that are otherwise unreachable by human placement. The deployment will be achieved by use of a UAV whose transmission range and battery life must withstand the distance and duration of time it takes to travel from initial UAV deployment to the inaccessible area and back to the home position.

Table 2.2: System Non-Functional Requirements Table

#	Non-Functional Requirements	Weight
1	System shall be compliant with all regulations	35%
2	Deployment and setup time shall be minimal	10%
3	System will be low maintenance and require no user input after deployment	30%
4	System will be able to store data for a one week period	25%

System Non-Functional Requirements Description

The functional requirements set above dictate the contents of the non-functional requirements. These specifications define the operating environment of the product and how it should operate. Of critical importance to the project is compliance with FCC, NTIA, and FAA regulations regarding transmission and specific laws governing any given bands. The device must also be low maintenance and easy to deploy. This means that a single operator and companion device can install the system. Then the mapping units must be operational for a week prior to needing more user interaction.

Hardware Requirements

Table 2.3: Hardware Functional Requirements Table

#	Functional Requirements	Ideal Value	Limit Value	Weight
1	Effective Transmit Range	1 mi	.25 mi	35%
2	Battery life of Mapping Unit	14 Days	7 Days	35%
3	Battery life of UAV	30 min	15 min	15%
4	UAV Range	5 Miles	1 Mile	15%

Hardware Functional Requirements Description

The software defined radio that we use is going to need to have certain hardware requirements in order to perform its function. The total system must be able to both transmit and receive, with the ability to receive for long periods under its own battery power in uncontrolled environments. The physical performance of its transmit and receive functions, as well as its battery life in both roles, are among the most important characteristics for its mission, and have been weighted accordingly. We want the system to operate independently for long periods of time to collect data, and be able to pick up as much detail in the bands of interest as possible. A UAV with a long battery life/range can allow us to move locations more easily.

Table 2.4: Hardware Non-Functional Requirements Table

#	Non-Functional Requirements	Ideal Value	Limit Value	Weight
1	Sample Rate	1 MHz	1 Hz	25%
2	Maximum Frequency	2 GHz	1.750 GHz	20%
3	Minimum Frequency	10 kHz	24 MHz	15%
4	ADC Resolution	12 bits	8 bits	20%
5	Max Bandwidth	10 MHz	6 MHz	20%

Hardware Non-Functional Requirements Description

The software defined radio that we use will perform better if it has certain specifications. The sampling rate of the SDR should be at at the nyquist rate of the signal in order to accurately record it. However, to just get a sense of if there is a signal transmitting on the frequency, a very low sampling rate will be sufficient. Since the group will acquire a technician’s license, they will have access to VHF and UHF frequencies. Therefore the the SDR must be capable of analyzing the specific frequencies available to us. Having a high maximum bandwidth will allow the system so see more information at one time and therefore gives us more data to be able to analyze. A high ADC resolution will allow data to be recorded at a more precise level.

Software Requirements

Table 2.5: Software Functional Requirements Table

#	Functional Requirement	Weight
1	Display current information on the bands being monitored on a custom GUI	20%
2	Record signal whitespace data and weather information locally	40%
3	Utilize historical data along with collision detection to determine where to transmit	40%

Software Functional Requirements Description

The software for the transmission system must be able to scan the available frequency bands and record their received signal strength in order to gather historical data for each frequency. From this recorded data, the software will be able to determine which frequency band is ideal for transmission. Current information of the area being monitored will be displayed to the user and all data will be stored locally. Weather data will also be collected so trends and timings can be analyzed for later use. This software will be compatible to run on a microcontroller.

Table 2.6: Software Non-Functional Requirements Table

#	Non-Functional Requirement	Weight
1	GUI displays best candidate frequencies	45%
2	GUI allows user to start and stop recording data	10%
3	GUI displays view of monitored frequencies	45%

Software Non-Functional Requirements Description

The Graphical User Interface (GUI) should be able to display important information as well as give control of the system to the user. The GUI should be able to display statistics about the usage of the frequency environment. Visual graphics will be displayed to succinctly describe this and display transmit information. By recording whitespace information of multiple bands the GUI should determine which of the bands are optimal to transmit upon.

Constraints and Assumptions

Design and Implementation Constraints

The design of each radio unit will have multiple constraints required to be easily deployable and all weatherproof. For a weatherproof RF design, there will be limitations on the material used for the enclosure so that water stays out and RF can be transmitted. The weight of each unit will have to minimized to allow a UAV to carry and deploy them. Furthermore each radio unit will need to easily grasped by the UAV, this forces a geometric constraint on the packaging of the unit. The unit will also have to have sufficient cooling to ensure the integrated electronics do not overheat. The unit must also be large enough to house a battery that can power it for the required amount of time. Due to the limitations on UAV usage and RF transmission regulations, only simple demonstrations may be completed of the final product.

Assumptions and Dependencies

The radio unit is intended to be deployed in areas that are generally inaccessible to humans for concern over safety. This issue would best be resolved by a UAV controlled a distance away from the mission objective location. This deployment system is necessary for the overall system to successfully operate, and must maintain a transmission signal far enough and hold a battery charge long enough for the static system to be deployed and for the UAV to return to the home position. There must also be active RF signals being transmitted in the area of deployment for the static system to be able to receive and transmit data. Most importantly, white spaces have to be found within the RF signals. Historical data will also need to have been recorded before the static system can begin receiving and transmitting data.

Applicable Codes/Standards/Regulations

The National Telecommunications and Information Administration (NTIA) and the Federal Communications Commission (FCC) are in charge of regulating radio communications and have many regulations that the device must abide by when receiving and transmitting signals. Additionally, since the project uses a UAV for transportation, the system must abide by any rules and regulations set by the Federal Aviation Administration (FAA).

It is illegal for anyone to transmit a signal without a technician class license. Therefore, at least 1 member of the team will take the required exam in order to become licensed. This license gives access to all VHF and UHF Amateur bands which can be found in Appendix A1.

To make sure the group stays within the rules set by the FAA, the group must keep line of sight while controlling the UAV, the UAV cannot be flown directly overhead anyone that is not controlling the UAV, the UAV can only be operated during daylight hours, and the UAV must stay below a maximum height of 400 feet above ground level.

Concept Development and Selection

Since the team is still determining the scope of their project, there has been limited progress on the concept development and selection. With that said, the team has come up with three prospective designs after their communication with their stakeholders and their advisor. In addition to the three preliminary concept designs, the team developed a concept selection matrix and a QFD. All concepts followed a general system architecture for the overall flow as shown in Figure 2.2 The team also divided up the two different, mechanical and electrical, elements of this system into three concepts each.

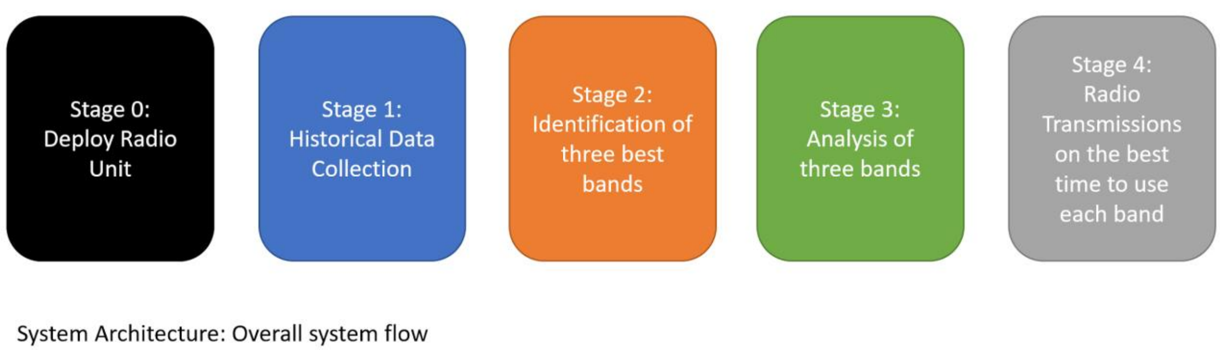


Figure 2.2: System Architecture

Mechanical Concepts:

Concept 1: UAV Landing - In this concept, the UAV will be responsible for the delivery of the electromagnetic spectrum mapping system (EMS system). The UAV will have to takeoff and fly a certain distance with the EMS system and will eventually make a soft landing to deploy the EMS system. From there, the EMS system will detach from the UAV and return to the dispatcher to pick up and deploy another payload. It is important to note that in this system as well as the system for the UAV Drop, will have the same EMS system. Once the system is deployed, it will map the surrounding EM spectrum and transmit that data to an end user station. This concept is shown below in Figure 2.3.

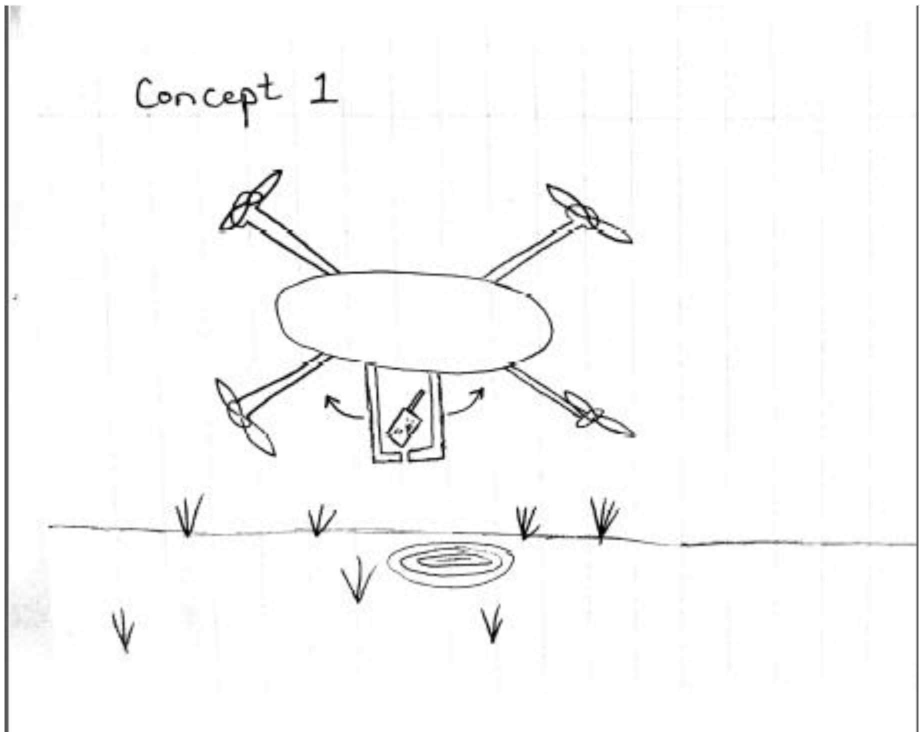


Figure 2.3: Concept 1

Concept 2: UAV Drop - In this concept, which specializes in a quick, stealth deployment will feature an EMS system box that is dropped from a UAV and makes a parachute landing. This system, which is ideal for military applications, is made to discreetly drop an EMS system into hostile territory. After the UAV deploys its payload, it will fly away back to the base to pick up another system. The payload will then free fall and deploy a parachute making a soft landing into the desired position. From there the EMS system will function as indicated above - transmitting data to the end user station. This concept is depicted below in Figure 2.4.

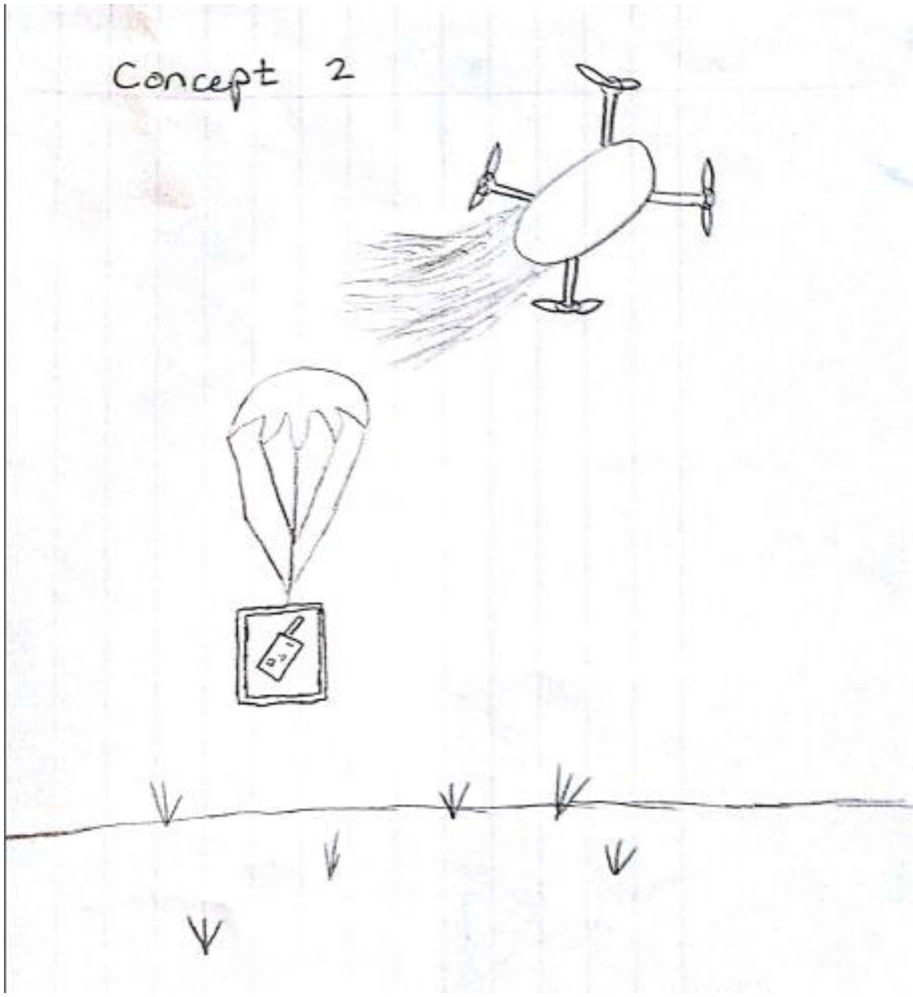


Figure 2.4: Concept 2

Concept 3: UAV Range extension - In this concept, the UAV will have a different functionality than in the two listed above. A user will deploy the UAV by some indiscriminate mean (such as a car or by simply placing down the unit). From there, the UAV will fly around with a range extender that will be utilized to extend the range of the EMS system. In addition to one UAV, multiple UAVs can be deployed to areas of interest in order to further extend the range of the EMS system. This concept is depicted below in Figure 2.5.

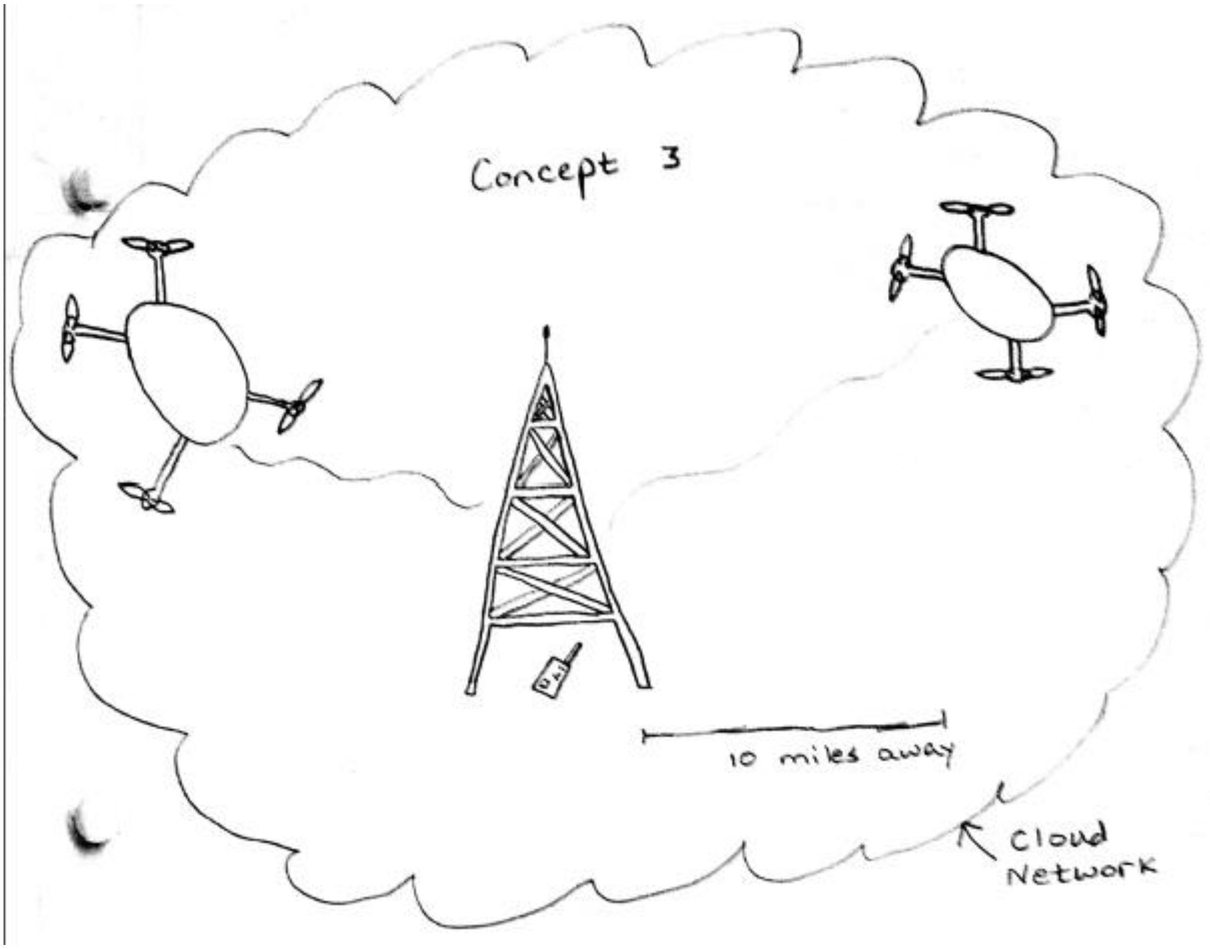


Figure 2.5: Concept 3

In order to quantitatively measure which choice is the optimal, the team created a pugh matrix to weight each option. This is depicted below in Table 2.7.

Table 2.7: Mechanical Deployment Pugh Matrix

Concepts							
Design		UAV Landing		UAV Drop		UAV Range Extension	
Selection Criteria	Weight	Rating	Weighted Score	Rating	Weighted Score	Rating	Weighted Score
Battery Life	10%	5	0.5	7	0.7	2	0.2
Product Enclosure Rating	15%	8	1.2	5	0.75	2	0.3
Transmission Success Rate	5%	6	0.3	6	0.3	8	0.4
UAV Range	10%	6	0.6	8	0.8	3	0.3
Compliant with Regulations	15%	8	1.2	2	0.3	6	0.9
Minimal Deployment and Setup Time	15%	10	1.5	6	0.9	6	0.9
Low Maintenance	20%	7	1.4	1	0.2	4	0.8
Low Cost	10%	5	0.5	4	0.4	3	0.3
Total Score	100%	7.2		4.35		4.1	
Rank		1		2		3	

For more information on each of the selection criteria options, please view the requirements section. In addition to the Pugh Matrix, the team utilized a QFD to help better determine which mechanical model to use. The QFD is shown below in Figure 2.6.

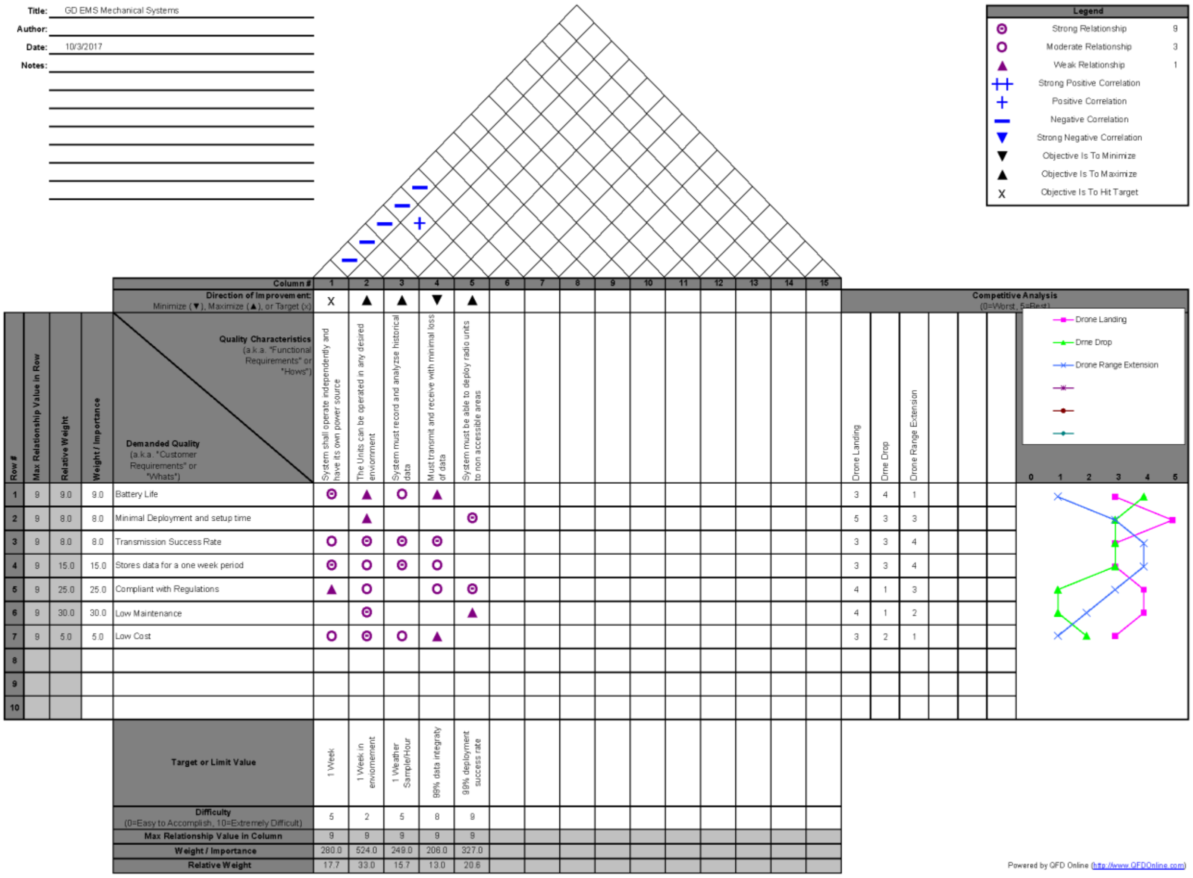


Figure 2.6: Mechanical QFD Matrix

Electrical Concepts:

Concept 1: In this concept, the team determined that one of our requirements is to have the system monitor the EM spectrum and simultaneously monitor several weather variables. The different concepts focus around how the weather data will be obtained. In the first concept, the team will install several sensors within their EMS system that will record various aspects of the weather. The team will install the following sensors: a barometer, Humidity sensor, Thermometer, and Rain Sensor

Concept 2: In this concept, the team will pull weather data, through the use of an API, from a local weather app. This concept will have no on board sensors and will gather data solely from a weather app.

Concept 3: In this concept, similar to the concept above, the team will pull data from a historical database instead of a live updating database. To clarify, the team will pull data from an established, historical database.

The concept development for the mapping unit is depicted below in the Pugh Matrix Table 2.8.

Table 2.8: mapping unit Pugh Matrix

Concepts							
Design		Sensors		Live Data		Historical Data	
Selection Criteria	Weight	Rating	Weighted Score	Rating	Weighted Score	Rating	Weighted Score
Effective Transmitting Range	20%	6	1.2	5	1	5	1
Record whitespace and weather data	30%	9	2.7	7	2.1	7	2.1
Display info on GUI	10%	6	0.6	6	0.6	8	0.8
GUI allows user to record data	20%	6	1.2	6	1.2	8	1.6
Battery Efficiency	20%	2	0.4	6	1.2	8	1.6
Total Score	100%	5.7		4.9		5.5	
Rank		2		3		1	

Preliminary and Detailed Design: Geometry, Materials, and Manufacturing Alternatives, Software/Hardware decomposition, System Integration Plans

System Architecture

Overview of System Architecture

Figure 2.7: Above shows a high level overview of the system architecture.

The total system is best described in multiple phases as shown above. The whole process begins when a UAV carrying the mapping unit is flown to the desired location. Once there, the mapping unit will scan the area and store data about the local RF environment. After this, the data will be analyzed to determine the best bands per variable (time, weather, day of week), and use this information to better transmit messages. The system requires multiple subsystems involving hardware and software that are described in depth in the following sections.

Hardware Architecture

The hardware architecture encompases two sub-systems, the mapping unit, and the deployment unit. The mapping unit consists of a software defined radio, a microcontroller, an antenna and a battery. The deployment unit consists of an UAV along with its transmitter, receiver and batteries. Figure 2.8 below shows the architecture with the components in detail. The statistics will be completed using a laptop running python. The statistics may later be ported to the microcontroller depending on complexity of the analysis.

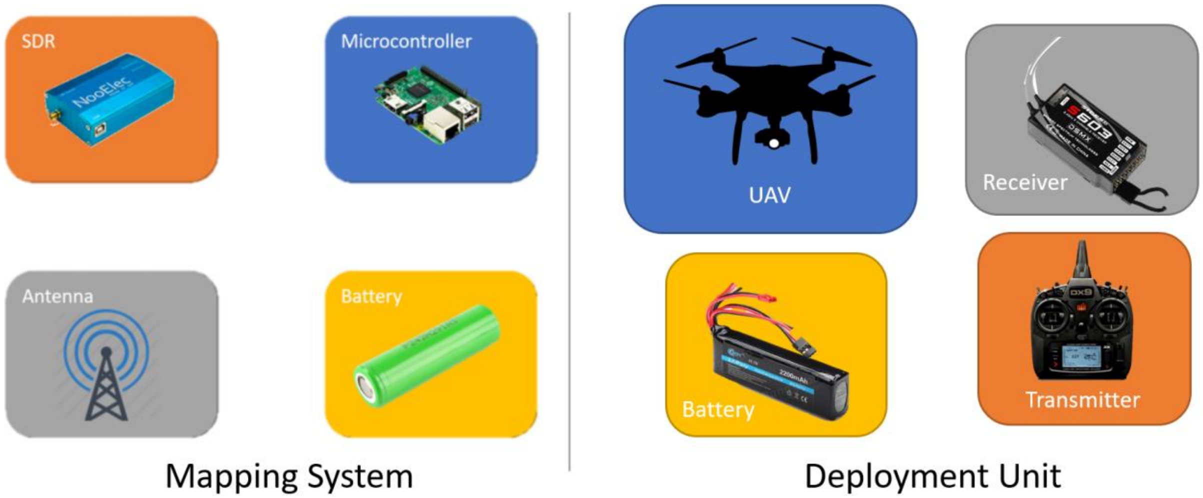


Figure 2.8: The above figure describes the hardware architecture of the system.

Software Architecture

The software is split up into multiple phases based on what the system is currently doing. There is an initial data collection stage, a data analysis stage, and a transmission stage. The overall architecture flow begins with data collection where a microcontroller will interact with a software defined radio to scan through various frequencies and store the relevant data. Next the system will enter the analysis stage where the data collected from the mapping unit will determine the optimal frequencies to transmit. This information will be stored in a model that will be accessed by a transmission capable computer in the transmission stage. At this point the computer and SDR will transmit preset data to the optimal frequency at that moment.

Communication Architecture

Due to the FCC, it is necessary that the system does not transmit on any frequency that it is not licensed to transmit on. However, a technician class frequency license is easily attained, and will permit users to transmit on certain unlicensed bands. The technician class frequency license gives access to the 50, 144, 222, 420, 902, and 1240 MHz bands.

After data collection and historical analysis, the system will need to utilize the detected white spaces. In order to do this, there will be a Transmission unit and a Receiver unit. The transmission unit will select the optimal frequency from the statistical analysis and transmit to the Receiver unit. The Receiver unit will always listen for the Transmission units data on the current optimal frequency in the hopes of receiving the correct signal. Frequency Modulation will be used for communication between the units.

System Design

System Behavior Description

The system consists of two subsystems. The first subsystem is the deployment unit, which is a UAV designed to lift a three kilogram payload. The second subsystem is the mapping unit. This mapping unit is the designated payload of the deployment unit. Both subsystems will be rigidly secured to each other. The entire system will follow the behavior shown in Figure 2.9 below. There will be 3 stages of behavior for the entire system. The first stage, Historical Data Collection, will occur with both subsystems. The second stage, Data Analysis, will occur on a user laptop or directly on the device. The last stage, Transmission, will occur on two laptops, one utilizing a HackRF and an RTL-SDR, and the other utilizing any SDR. The system including both laptops and all SDRs will be referred to as the transmission station. This transmission station will utilize a transmission capable SDR to transmit on bands and frequencies determined by the data collected and analyzed in the first and second stages.The first stage will last approximately 36 hours. The second stage will be completed in a few hours. The final stage will be run as required.

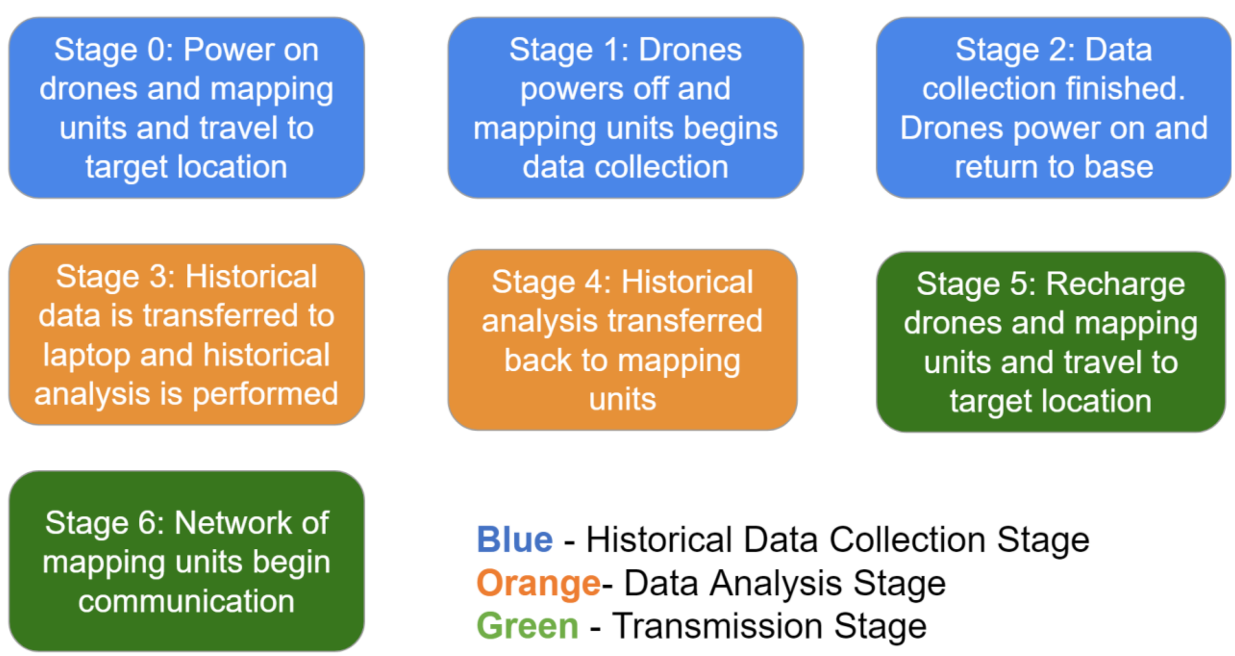


Figure 2.9: The above figure shows a closer look into each section that is displayed in the high level overview of the system architecture

Databases/Data Storage

The historical data will be stored locally on an SD card connected to the microcontroller. The unit will be able to store data for a period that is determined by the selected battery. Rather than storing all the data that the SDR is gathering (signal power in dB), it will only store relevant header information and a single character to represent whether or not there is a signal present on that frequency. All data will initially be packed into a binary file format that drastically reduces the space required to store the data on the collection unit, which has relatively little storage. The output will be written to different files based upon a user setting to avoid total data corruption in case of errors. This will ensure that a write fail will not destroy all the collected data. Furthermore, splitting the files this way will make them easier to analyze during the data analysis stage. Along with the binary file there are also weather files that store formatted info from the Weather Underground API[9] and the settings file generated by the user interface. Once collection stage has finished, the binary file can then be downloaded to a computer and converted into a CSV to be used in the statistical analysis, with frequency, time, and weather being extrapolated from the binary file, config file, and weather files.

Data Conversion

The data gathered by the SDR is complex value that contains a magnitude and phase value of the signal. This is passed through a 1024-point FFT. N samples of the FFT are taken at a given sample frequency, with the default being 75. These 75 samples are then averaged together to reduce noise and get an image of the spectrum around the desired frequency. The mean (μ) and standard deviation (σ) of the averaged FFT are taken, and each bin in the FFT is marked with a 1 or a 0. 1 if the power at that bin is above ($\mu+.3\sigma$), and 0 if it is below ($\mu+.3\sigma$). This provides a reliable way to detect signals with signal-to-noise ratios above unity. Figure 2.10 below is a test showing successful detection of known FM radio stations between 99 and 101 MHz. The red plot shows a single, raw FFT frame from the SDR. The orange plot shows the average of 512 raw samples, severely cutting down the noise and making the channels very clearly visible. The green plot shows the results of the decision for whether or not there is an active channel in that bin, with a sensitivity of 0.5.

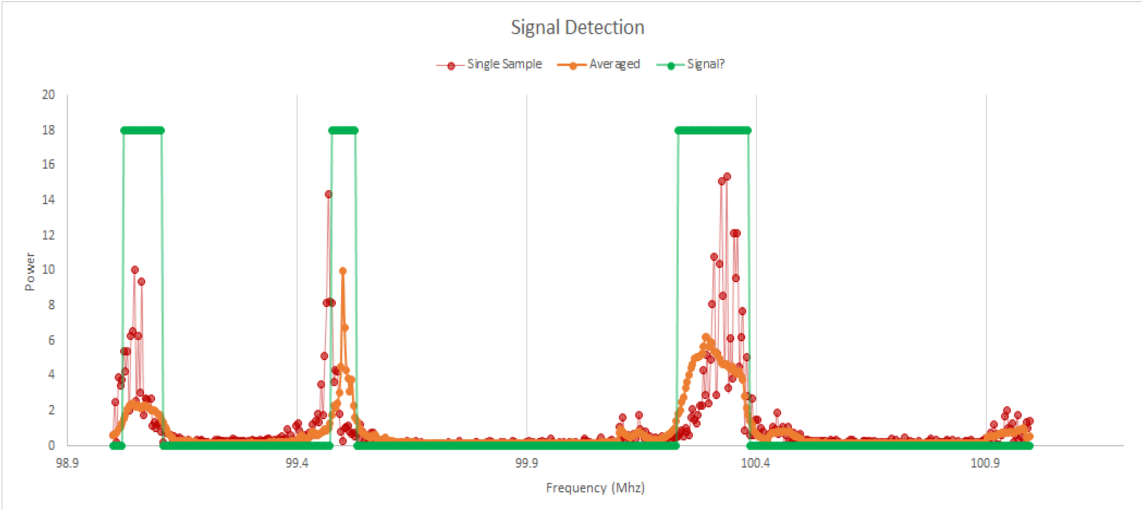


Figure 2.10

Sub-System Interfaces

Only one interface exists between the two subsystems. This interfaces is implemented through a single digital connection. This connection will allow for the mapping unit to be notified of when the user has landed the UAV. The user will notify the unit by using a switch on the transmitter. Then the microcontroller will power off of the deployment unit using a relay housed on a separate circuit with just a a relay running between the UAV and the its power source. The microcontroller can then re-enable the UAV, when it is done historical collection stage is complete. This exchange is portrayed in Figure 2.11 below. For the scope of this project an off the shelf unit will be purchased for the relay. If such a unit cannot be found, the interface will be forgone for the prototype.

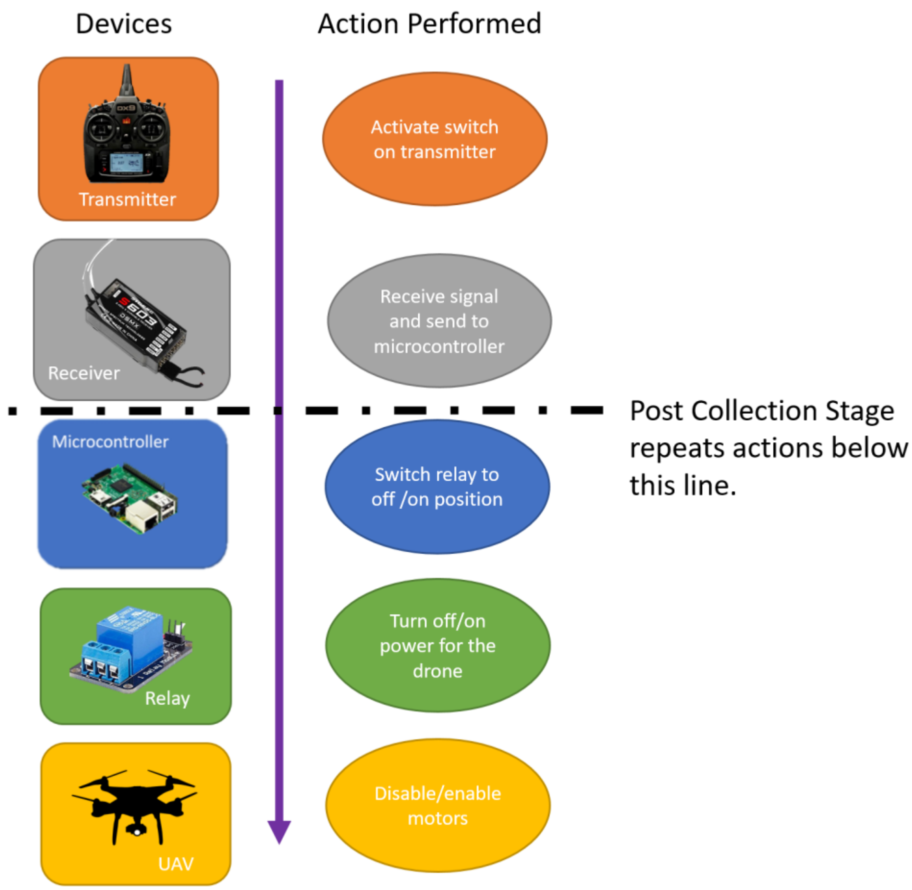


Figure 2.11: Subsystem interaction

User Interface Design

The system will interface with the user in multiple ways depending on which stage, referring to figure 2.11, the system is on. The UAV will be controlled with an eight channel transmitter by the user. Six channels will be used to control the UAV's flight while one channel will be used to set the flight characteristics (sport, regular) and one channel to interface the UAV with the mapping unit.

The microcontroller will begin data collection as soon as the UAV is launched. The travel time of the UAV will be subtracted from the data.. The microcontroller will have the data collection program set to run at startime and will begin recording data according to the parameters laid out in the config file created by the user. The data collection script has a number of input parameters to govern its operation. The user can select between a continuous or non-continuous mode, time before changing the write files, sensitivity threshold coefficient, number of iterations to collect for (governed by file write change time), and which frequencies to monitor. In continuous mode, the unit will scan between the minimum and maximum set frequencies through constant increments. Non-continuous mode only monitors the center frequencies provided. These can be set prior to sending out the UAV for greater ease of use but can be changed as necessary and created with the GUI.

Once powered on, the mapping unit will scan through the frequencies of interest and output all relevant data to binary files. This unit is able to operate independently until the user decides it has received an accurate record of the RF environment of the selected area. If the unit is in an area with internet access it can also be controlled through SSH. This can be used to pull files that have finished writing. The user will then import the information to a laptop running the python analysis module. A stretch goal for the project would involve porting the statistics model onto the microcontroller. This is not a starting point as these calculations are computationally expensive and would reduce the amount of samples recorded. A graphical user interface has been developed using TKinter to serve as a hub of operations during this process. From here the user can convert binary files into csvs, download files from the mapping unit through SCP, run analysis, export settings files to the Pi, interface with the team's data server, initiate collection of data through SSH, perform signal detection, and transmission. This GUI reduces the necessary knowledge of the end user and limits the need to work via command line interfaces.

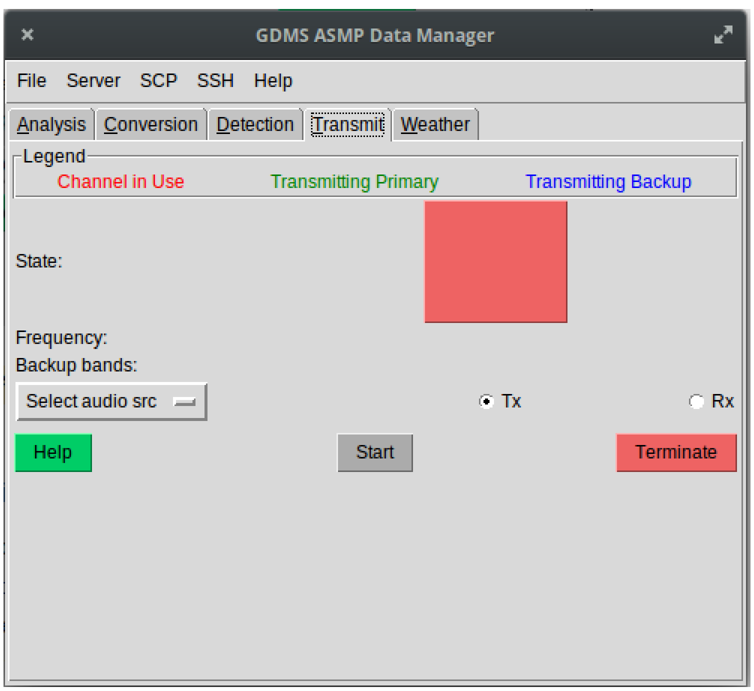


Figure 2.12 Screenshot of GUI. User is currently on the transmission screen.

Mechanical Design

The mechanical design for this project will span two major sections. The first is an encasement unit that will enclose the mapping unit from the outside elements and prevent any water or dust damage to the electronic components by maintaining an IP67 rating. This enclosure will house the BeagleBone Black, the RTL-SDR RTL2832U, and 4 - 8 D Cell Batteries. The second stage of the mechanical design will involve design and fabrication of the UAV. There will need to be a number of power calculations performed in order to determine the rating of the motors that will produce enough lift to take the UAV airborne. There will also need to be calculations to determine type and size of batteries that will be used to power the UAV long enough to reach its destination while supporting its payload. The enclosure would have to be mounted onto the UAV in order to transport the mapping unit to the target location. The encasement will have to be secured so as to prevent damage to the microcontroller and SDR, and remain secured for the duration of the flight.

Components and Sub-Assemblies - UAV Design

The UAV lift calculations were completed using a basic static equation modified with empirical data. This equation was developed by Gabriel Staples[7]. The equation uses geometry of the propellers to calculate the total lift of the UAV. When comparing to empirical data it was shown to be within 20% of the actual thrust. The key characteristics of a proller is the length and pitch of the airfoil. On the motor, a kv rating can be used to find the rpm of the motor at a certain voltage according to the equation below:

$$RPM = kv * Battery\,Voltage$$

This equation is best used for static thrust calculations. The comparisons of data between the equation and actual measurements show that it is not as accurate for dynamic thrust. Thus, for this project just static thrust was considered and Vo was set to zero. The equation used is shown in figure 2.13 in its expanded and simplified form, F is given in Newtons.

Dynamic Thrust Equation
F = thrust (N), d = prop diam. (in.), RPM = prop rotations/min., pitch = prop pitch (in.), V_0 = propeller forward airspeed (m/s)

Expanded Form:

$$F = 1.225\frac{\pi(0.0254 \cdot d)^2}{4}\left[\left(RPM_{prop} \cdot 0.0254 \cdot pitch \cdot \frac{1min}{60sec}\right)^2 - \left(RPM_{prop} \cdot 0.0254 \cdot pitch \cdot \frac{1min}{60sec}\right)V_0\right]\left(\frac{d}{3.29546 \cdot pitch}\right)^{1.5}$$

Simplified Form:

$$F = 4.392399x10^{-8} \cdot RPM\frac{d^{3.5}}{\sqrt{pitch}}(4.23333x10^{-4} \cdot RPM \cdot pitch - V_0)$$

Gabriel Staples, 2013. http://electricrcaircraftguy.blogspot.com/

Figure 2.13: Static Thrust Equation.

To use this equation multiple motor, prop, and chassis options were used to run an excel optimization for maximizing thrust and minimizing weight. The data for this is shown in figure 2.14 below. This was completed leveraging the vlookup function in excel. The final calculations indicate that the UAV will be lift 3 kg of payload weight. Of Course, this is only an approximate calculation. This will only be within 20% of the final thrust. The final parts used in the build were at similar spec to the items presented in this analysis. This was due to purchasing requirements, and availability.

Brushless Motor			
Item Index	Description	KV Rating	Weight (g)
1	DL2306	2700	134
2	Genuine Gemfan 1104	4000	18.8
3	DL2205	2300	112
4	LHI	920	300
5	D2826	1400	272
Propeller			
Item Index	Description	Diameter(in)	Pitch(in)
1	RayCorp 6045	6	4.5
2	RayCorp 1045	10	4.5
3	RayCorp 5030	5	3
4	XOAR PJM-E	14	7
5	Genuine Gemfan'	8	4.5
Battery			
Item Index	Description	Weight (g)	Cell
1	Multi Star 5200mAh	453	4
2	HRB 5000mAh	405	3
Chasis			
Item Index	Description	Weight(g)	
1	Tarot FY690S	600	
2	Hobbypower S550	550	
OPTIMIZATION			
motor	5	1,5	
prop	2	1,5	
batt	1	1,2	
FIXED			
Batt Voltage	14.8	V	
RPM	20720		
Prop Dia	10	in	
Prop Pitch	4.5	in	
Thrust	53.55125674	N	With atmoshperic density being 'standard day' at 1.225 lg/m^3
Thrust (g)	5458.843704	g	
Total Propl. W.	1778	g	
Payload Weight	3680.843704	g	

Figure 2.14: UAV power and weight analysis

Components and Sub-Assemblies - Bill of Materials

Bill of Materials						
BOM Number	Sub Assembly	Model Number	Model/Maker Name	Part Name	Quantity	Unit of Measurement
1	UAV	775	KDE	Brushless Motor	6	Each
2	UAV	N/A	Genuine Gemfan	Propellor	1	Each
3	UAV	Battery	Multi Star 5200mAh	Battery	4	Each
4	UAV	GFY690S	Tarot	Chasis	1	Each
5	UAV	B01HEMK81C	Turnigy	Plush 60amp Speed Controller w/5A BEC	6	Each
6	Enclosure	P430	Stratasys	ABS+ Plastic	18.41	Cubic Inches
7	Enclosure	Black	Raspberry Pi	Microcontroller	1	Each
9	Enclosure	RTL2832U	RTL-SDR	Software Defined Radio	2	Each
10	Enclosure	M5	NooElec RaTLSnake	Antennas	2	Each
11	Enclosure	Solo 5	Rosmoss	PowerBank	1	10000mAh

Geometric Specification - CAD Models/Drawings

The structural design for the encasement unit will be modeled in SolidWorks. The final CAD models for the prototyped mapping unit enclosure may be seen below in Figures 2.15 and 2.16.

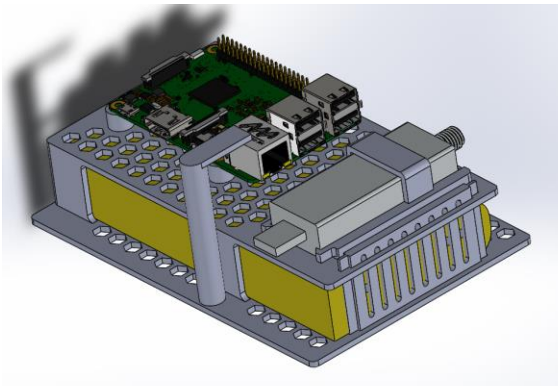


Figure 2.15: Internal Frame

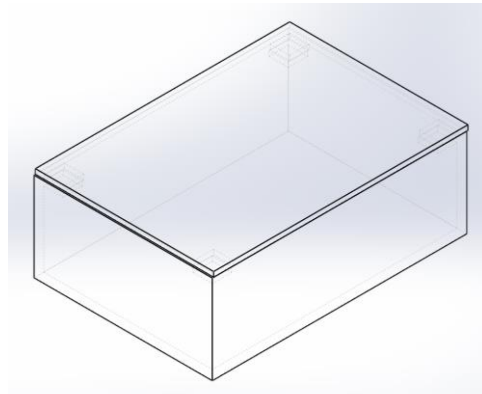


Figure 2.16: External Enclosure

The team’s current design concepts for the prototyped mapping unit enclosure involved two separate parts. The first was an internal skeleton that will house each of the electronic components that make up the mapping unit. This skeleton will hold the components in place so as to survive any vibrations brought on by the UAV flight, and may be seen in the above Figure 2.15. The tolerancing for this design must be extremely precise, as the electronic components are modelled so as to remain secure from the friction forces of the skeleton. Without these precise tolerances, the components would be prone to dislodging from the unit. The lower level will house the battery which will supply power to the software defined radio and Raspberry Pi which are secured in the second level. The microcontroller unit was placed on standoffs that better mitigate overheating from the units. The same can be said about the skeleton’s cellular structure, which takes on a honeycomb pattern to better manage the heat flow from the electronics, as well as provide for an all around lighter part weight. The battery is held in place via a detachable chamfered door which slides through a slot toward the front of the design. The SDR unit is housed in place by a combination of strategically placed raised surfaces and self-sustaining snap fits. There is also a handle that will allow the team to relocate this internal frame from the external encasement so as to easily remove the SDR and microcontroller for data extraction upon completion of sampling, and remove the battery for charging. This way, the internal frame may be transferred in and out of the secure metal shielding whenever the team may find it necessary.

The external enclosure may be seen in Figure 2.16. There is a need for precise tolerancing in this design as well, as the skeleton must be secured by the friction forces of the external enclosure in order to mitigate damage brought on from UAV vibration or other transportation issues. The external enclosure will require fastening screws in each of the four corners which will lock the top cover into place with the lower housing and provide for a water resistant finish. This will be further enforced by the addition of a submersible cord grip which will be inserted along one of the side walls in order to provide access for the SDR's antenna to the outside world without sacrificing the enclosure's resistance to the elements.

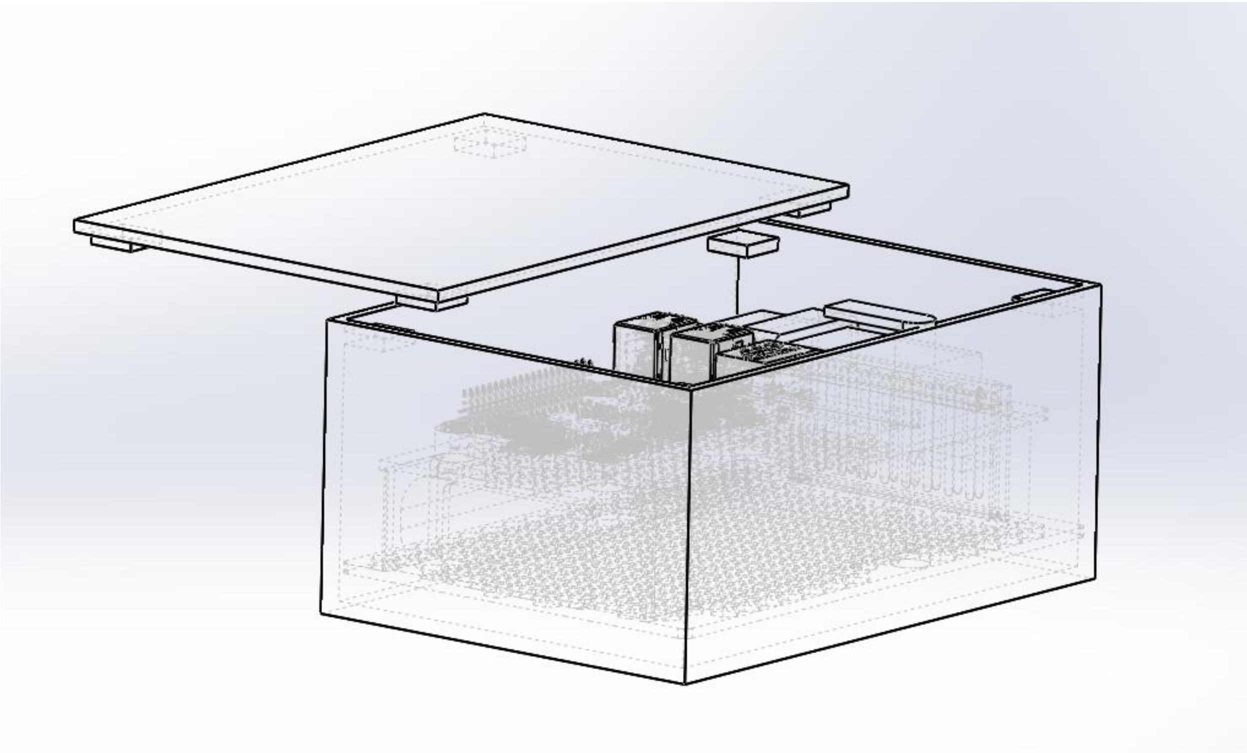


Figure 2.17: Early Prototype Assembly of Enclosure for Mapping Unit

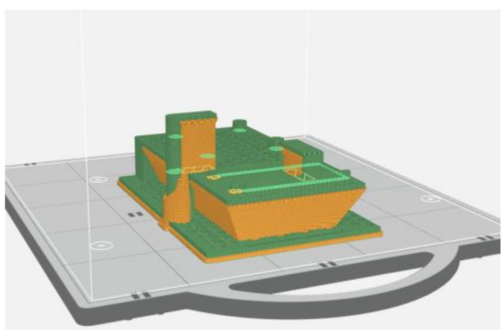
The final prototype assembly for the mapping unit enclosure may be seen above in Figure 2.17, in which the external enclosure houses the internal frame that secures the mapping unit's electronic components.

Materials Selected

The primary materials needed for the mechanical design consist of ABSplus plastic which will make up the internal frame, and aluminum alloy 6061 T6 to make up the external enclosure. The aluminum will provide shielding for the software defined radio unit against the electromagnetic radiation emitting from the UAV as well as any other nearby EM wave interference. Additionally, this alloy has good machinability, excellent formability and weldability, and excellent corrosion resistance. An all plastic design would not protect the unit from the elements as well as the aluminum, and could not provide the same level of EM shielding for the SDR to gather accurate data.

Manufacturing Process Selected

The team decided that the best method to fabricate the final plastic enclosure would be done via additive manufacturing. This would be most practical in the sense that it would produce durable parts in a timely fashion at a low cost. This type of manufacturing would be performed through the Stevens PROOF lab using a Stratasys Dimension SST 1200es printer. This is the ideal printer for manufacturing this part, as it is an FDM printer capable of printing in ABSplus material of precise layer resolution 0.01” by 0.013”. The support material for this printer is P400SR which is soluble in a hydroxide solution. Soluble support material will maintain the part’s surface finish and overall material strength. Two parts would need printing: the frame and the battery door. The estimated print time, material used, and support structure orientation based on the Stratasys Dimension SST 1200es printer specifications are displayed below for the internal frame and battery door in Figure 2.18.



Tray Estimations		
Printable Frame	Print Time	11h 45m
	Model Material (in^3)	4.464
	Support Material (in^3)	4.413

Tray Estimations		
Front Door	Print Time	34m
	Model Material (in^3)	0.385
	Support Material (in^3)	0.161

Figure 2.18: Estimated print properties for internal frame (left) and Battery Door (right)

In terms of design for manufacturability, a fully 3D printable box would be infeasible and does not make sense to proceed with additive manufacturing for the entire design. That would be a waste of material and cost, and defeat the purpose of additive manufacturing, which is to form more complex geometries. Thus, the team selected a more durable metal external enclosure and would need to machine it. The aluminum enclosure would need to be machined out of the aluminum alloy. This would be done by first bending the aluminum sheet so as to outline the general shape, and then cutting that sheet down to size to fit into the confines of the plastic enclosure. There would be two parts here, both of which would be fastened together by machine screws. For prototyping purposes, the team decided to first emulate the aluminum enclosure with bandsaw-cut and belt sanded acrylic panels assembled to meet the same geometry and fastened together via Weld-On solution. The acrylic material was considered waste material and served no additional cost to the team.

The UAV’s fabrication mainly involves simple assembly of components such as motor, esc, flight controller, and prop attachments to the frame.

Process Description or Optimization Notes

Moving forward, it would be beneficial for the team to pursue other means of manufacturing such as injection molding and larger scale machining. This could allow for faster production rates and increase the amount of batches produced per unit time. There would also exist stronger mechanical properties after being fabricated by way of more professional methods.

Optimization studies will need to be run on the enclosure unit in order to design for the least amount of mass and volume possible while maintaining structural integrity.

Electronics/Electro-Mechanical Design

Circuits

This project will only use a single custom circuit. This circuit will be used to disable the UAV power system via the microcontroller and regulate power to the microcontroller. The schematic for this circuit is shown below in Figure 2.18. There are two primary purposes of this circuit. First, by disabling the power to the UAV, the mapping unit will ensure that it won't accidentally receive any interference from the UAV electronics. Second, due to the long duration of the historical data collection stage, the microcontroller will require a stable power source. The board will contain a relay to be able to turn off the UAV once a signal to do so is required. A receiver port will be used to signal to the microcontroller to disable the UAV. The relay used in this circuit is not the final relay integrated circuit to be used. The final integrated circuit will include a relay controller that will work with a double digital input that will not require any additional power from the microcontroller to keep the relay active.

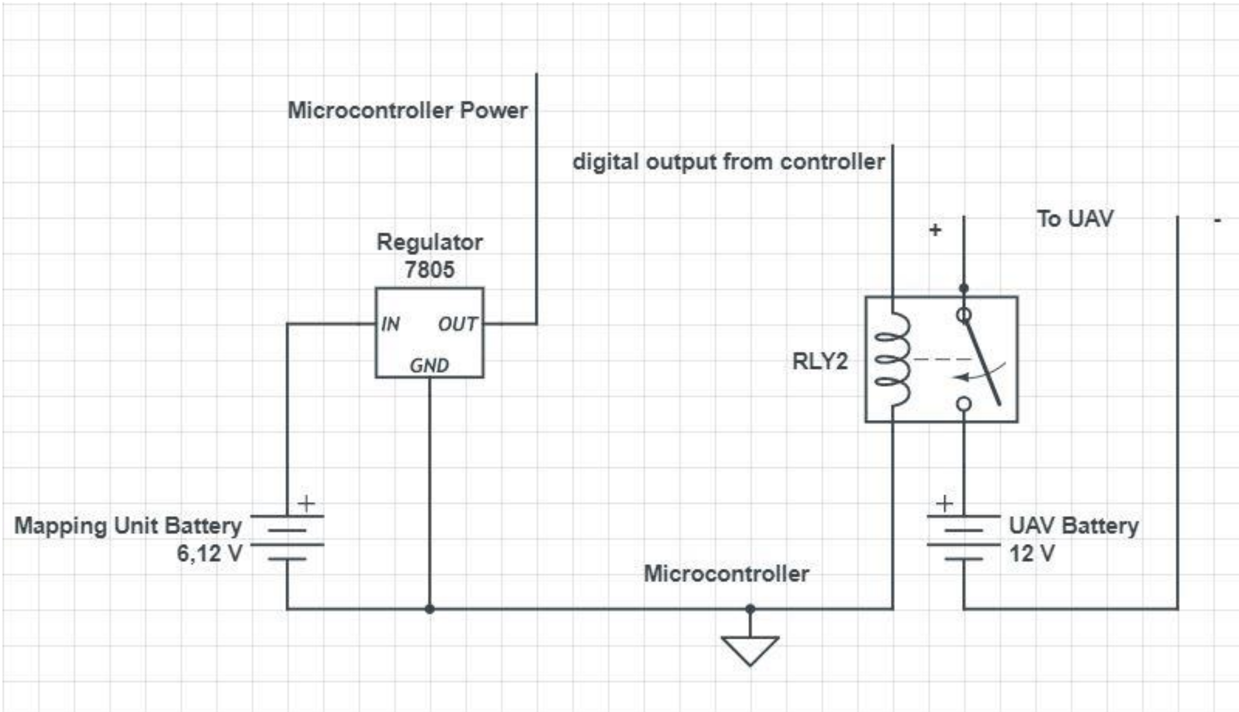


Figure 2.19: System circuit diagram with single digital input that will be used to regulate power to microcontroller and disabling the UAV

The drone wiring was completed using the schematics shown in figure 2.19. The primary components were the drone autopilot system (DJI NAZA V2) and the power system using 6 Motors and 6 ESCs (Electronic speed controllers). The BTM is the bluetooth module used to tune the autopilot using a phone app. The RX is the receiver for the transmitter. The LED is a module used by the autopilot to indicate its current status. The power is distributed to the escs before being transferred to the autopilot which then distributes it to the modules in blue.

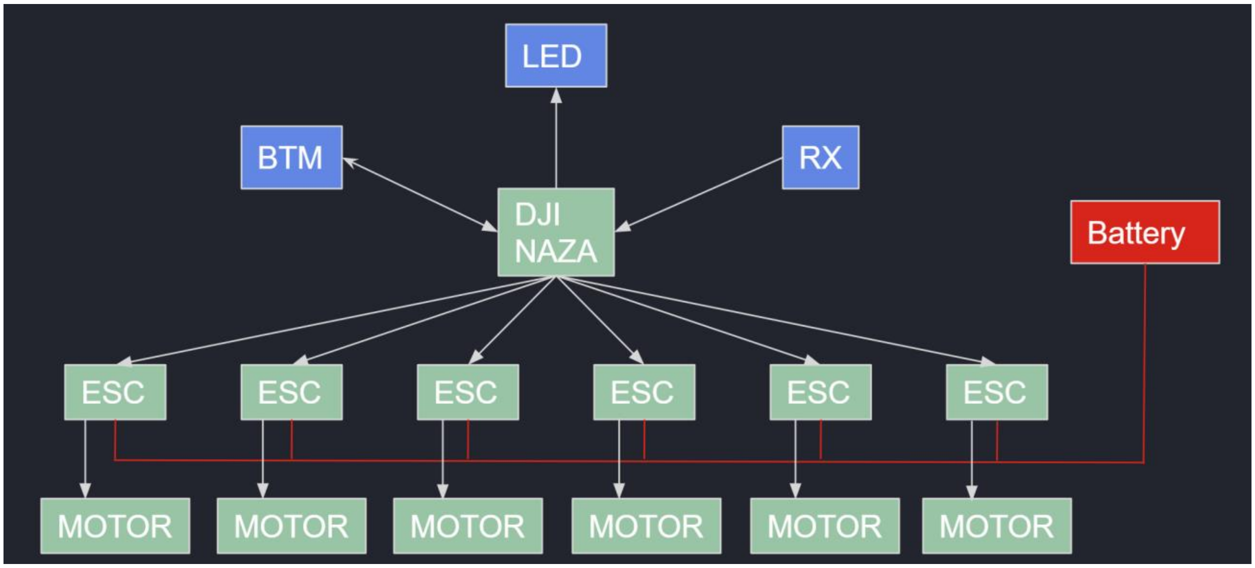


Figure 2.20: Prototype drone wiring

Power System

The power system for the mapping unit was analyzed by starting at the power usage of the electronic components that make up the unit. The total amperage required to drive both devices was found to be 350mA with 5V required for the microcontroller (RB). The microcontroller will then feed the SDR the appropriate voltage. The breakdown is shown in figure 2.21 below.

Power supply					
Item	Amperage (mA)	Voltage (V)	Power (W)	Weight (g)	Links
Micro Controller	300	5	1.5	45	
SDR	50	3.3	0.165	141	
Fan	300	5	1.5	40	
Total	650	13.3	3.165	226	-

Figure 2.21: Total power consumption for mapping unit

Next, multiple possible batteries were investigated. The batteries were put in series until they provided the required 5V for the microcontroller. Then the capacity of each battery was investigated along with its weight and volume. A LiOn and a D Cell were found to have the best capacity to weight, and capacity to volume ratios respectively. The calculations are shown in figure 2.22 below.

Type	Series Qty	Voltage (V)	Capacity (mAh)	Time (hr)	Weight (g)	C/W	Single Volume (in^3	Tot Volume	C/V
AA	4	6	2000	5.71	92	21.74	0.48	1.90	1052.26
D Cell	4	6	15000	42.86	540	27.78	3.21	12.85	**1167.45**
3.7 LiOn	2	7.4	2000	5.71	70	**28.57**	1.01	2.02	992.06
3.7 LiOn	2	7.4	6600	18.86	310	21.29	4.03	8.05	819.73
11.1v LiPo	1	11.1	3000	8.57	272	11.03	28.67	28.67	104.63

Figure 2.22: Battery Comparison 1

Finally, a minimum on time was determined from functional requirements. This was then translated to the quantity of each battery series needed. The D cell with four cells to a series was decided to have the best capacity and volume.Figure 2.23 presents this information below. However, during alpha prototype implementation it was found that the D cell batteries had extreme drop of cycles. Therefore they could only be used for a portion of the calculated times. Therefore the 3.7v Li-Ion was instead planned to be used. Once again this was changed to a 5v power bank for ease of design. This powerbank ensured that the team did not have to design their own battery charging circuits. The powerbank chosen had a total capacity of 20,000 mAh. This was the closest to the required 12,600 mAh, but below it.

Time Limiting	36	Hours	* Intermittent			
Type	Series Qty	Total Cap	Over Cap	Total Hours	Total Weight (g)	Total Volume (in^3)
AA	7	14000	1400	40	644	13.30464489
D Cell	**1**	**15000**	**2400**	**42.85714286**	**540**	**12.84848563**
3.7 LiOn	**7**	**14000**	**1400**	**40**	**490**	**14.112**
3.7 LiOn	2	13200	600	37.71428571	620	16.1028
11.1v LiPo	5	15000	2400	42.85714286	1360	143.36

Figure 2.23: Battery Comparison 2

Software Components Design

Algorithms

The algorithm for transmission will use a learning model created by the statistical analysis of the long-term data. The inputs for this model will be the current time and weather condition, and the output will be the frequency to transmit on. Once a frequency is received, the algorithm will check to see if the band is currently occupied. If it is not, it will transmit for one second. Otherwise, it will wait for the next transmission slot and repeat the process.

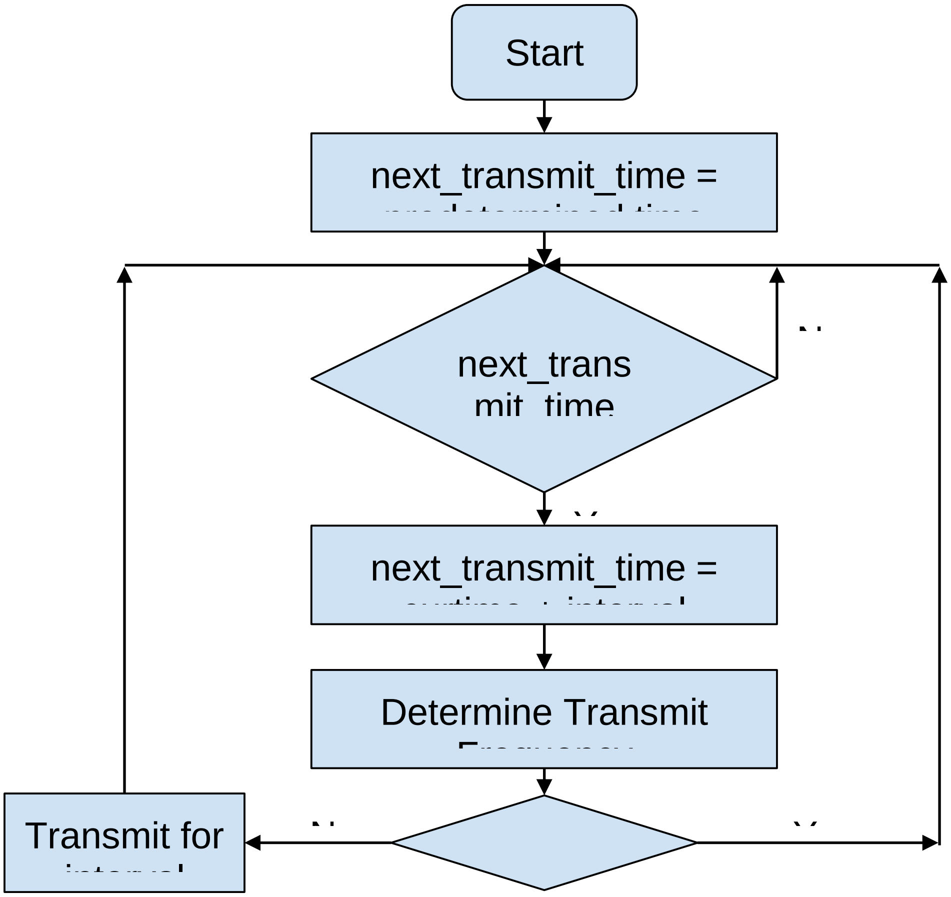


Figure 2.24: Flow Diagram for Transmission unit

The algorithm for the receiver during transmission will use the same model as the Transmitter. At the start of each time increment it will determine the band to listen to. For the duration of the time increment, it will record what it receives along with the band it was on.

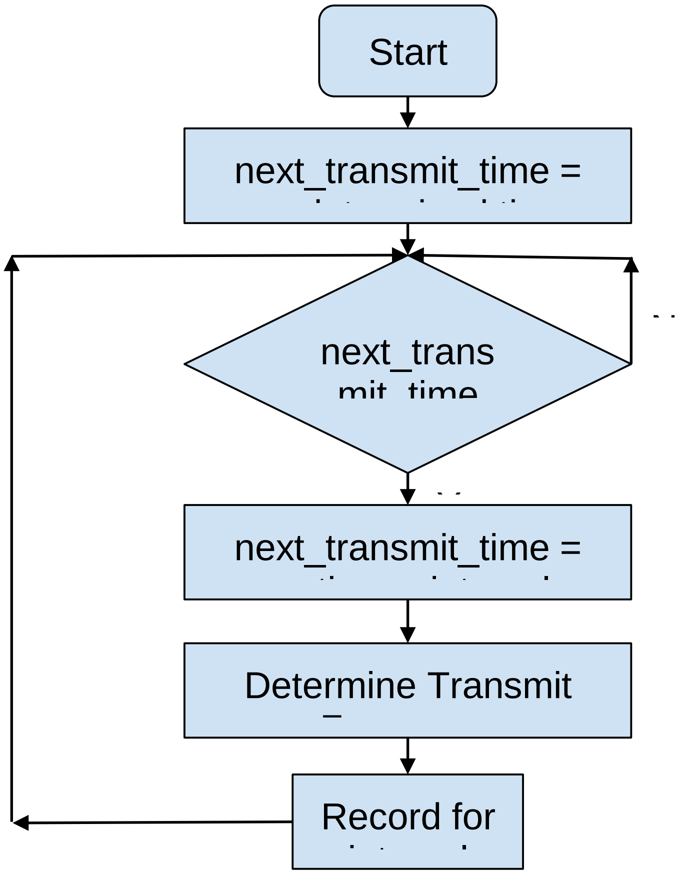


Figure 2.25: Flow Diagram for Receiver unit

Modules/Classes/Methods

The overall system can be best explained by breaking it down into multiple sections. There is software needed for the data collection/storage, analysis of the historic data, and the transmission sending protocol. The next paragraphs of this section will be split to cover each of these topics respectively.

Data Collection

The first step in the system architecture involves the collection of data that succinctly describes the RF environment of the area. The information gathered here will be used later on to govern transmission protocols. These decisions will be made based upon the activity on different frequencies. To obtain this a mapping unit will scan the frequencies of interest and record relevant metadata. The mapping units will run on a Raspberry Pi and utilize a software defined radio. Using the GNU Radio library, the system will determine if a signal is present through the use of a dynamic threshold and will record the data in a binary file. The thresholding process is discussed above in the data conversion section.

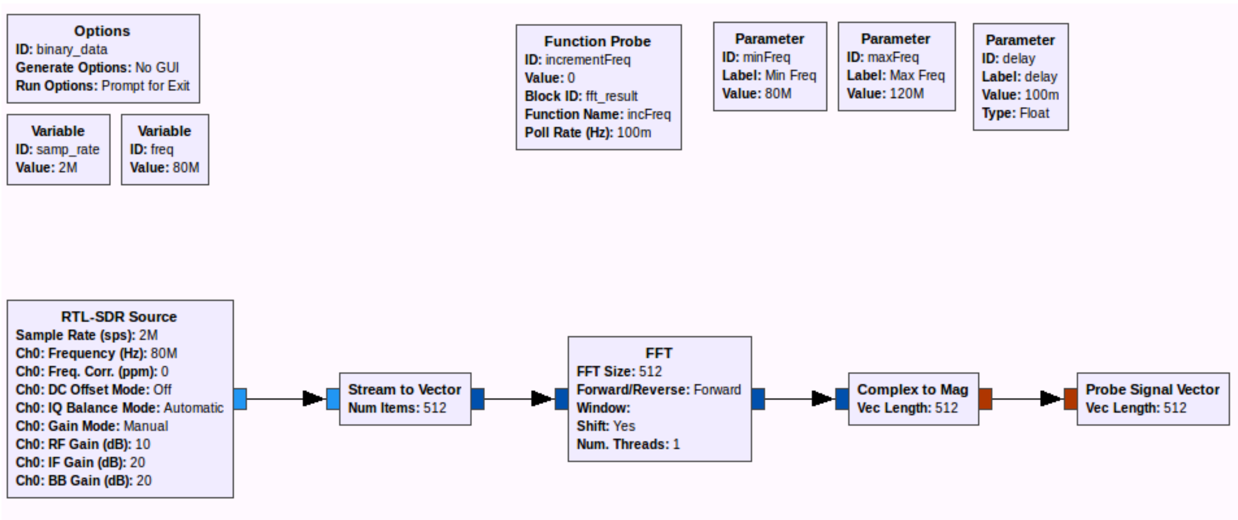


Figure 2.26 Current flowgraph for the data collection in GNU Radio Companion.[4]

In order to get an accurate understanding of this value in the next software module, a timestamp is recorded using the datetime library.[1] The timestamp, center frequency, and whether or not a signal was detected are recorded in a binary file. The binary files themselves only actually record a single byte per data point. The only information recorded is the thresholding result in the form of a character. The frequency information and timestamp are extrapolated from other information included in the header of the binary file. This method has resulted in 99% reduced file sizes as compared to CSV. Figure 2.27 shows the format of the binary files. Each one has 76 bytes of header information. The rest of the data is extrapolated using this information in the unpacking script that was developed. The unpacking script requires the binary file of data, config file for that collection, and the weather file in order to graft together a CSV with the proper information. The script converts the raw binary into readable CSV files that are able to be understand by most programs. Once this data has been collected, the collection stage comes to an end and the data is ready to be analyzed.

Binary file format		
Data	Type	Size
Start timestamp	String	30 bytes
Start frequency	Unsigned int	4 bytes
Sample Rate	Double	8 bytes
Data points	Char	1 byte per
Close frequency	Unsigned int	4 bytes
Close timestamp	String	30 bytes
EOF		

Figure 2.27: Binary file format structure. Uses 76 bytes of header info per file for extrapolation purposes. These files are written to for a set period of time.

Frequency	Signal?	Precipitation?	Relative Humidity	Cloud Level	Hour	Minute	Second	DOW
469000000	0	0	60	2	6	52	2	6
469001953.125	0	0	60	2	6	52	2	6
469003906.25	0	0	60	2	6	52	2	6
469005859.375	0	0	60	2	6	52	2	6
469007812.5	0	0	60	2	6	52	2	6
469009765.625	0	0	60	2	6	52	2	6

Figure 2.28: Sample output following the data collection stage. This record displays combined information from the binary file, config file, and weather file.

Statistics

The historical data collected will be statistically analyzed to create a prediction model. The collected data will include the state of the band along with the current time. The current time will be converted to: day of the week and time (Ex. Monday - 11:00:01) and just time (11:00:01). The time will be kept using the 24 hour method to simplify design. The independent variables will be, day of week, hour, minute, second, current cloud coverage, and precipitation. The dependent variable will be the center frequency of the band most likely to be free at the time. Data will be collected for a period of time between two weeks for initial study to one month for the final model.

The model will be built using a stochastic gradient descent model mimicking a support vector machine (SVM)This is the most suitable method as the dependent variable is discrete, and the it can hold more than two distinct values while allowing for online learning. The model will built using the scikit python library. To further quantify the quality of each variable, a classification test will be run with a forest of decision trees to rank each independent variable. This will directly output the best and worst features. The final model will be built using cross validation principles with approximately 120 hours of data, with one six hour being left out.

Transmission Protocol

After the historical analysis has concluded, the transmission station must test using the data to send signals. The channels that the unit will transmit and receive on will be determined by the historical analysis .csv file that was created from the long-term data collection. The transmission unit of the transmission station will consist of a laptop with one transmission-capable SDR, and any other SDR for receiving. The receiver unit of the transmission station will consist of a laptop with any SDR Both units will be using the historical data from the Analysis stage. The transmitter will sample the optimal frequency in order to determine if the frequency is occupied. If the transmission unit finds the channel unoccupied, it will transmit data using frequency modulation over the selected frequency. If the transmission unit finds that the channel is currently occupied, it will fall back to the next best frequency. If all frequencies are in use, the transmission unit will transmit nothing and attempt the first channel again, repeating the process. The Receiver unit will listen on the same frequencies as the Transmitter unit at the same times, and record the received data. Once the transmission test is complete, the data will be reviewed to determine the success and collision rates.

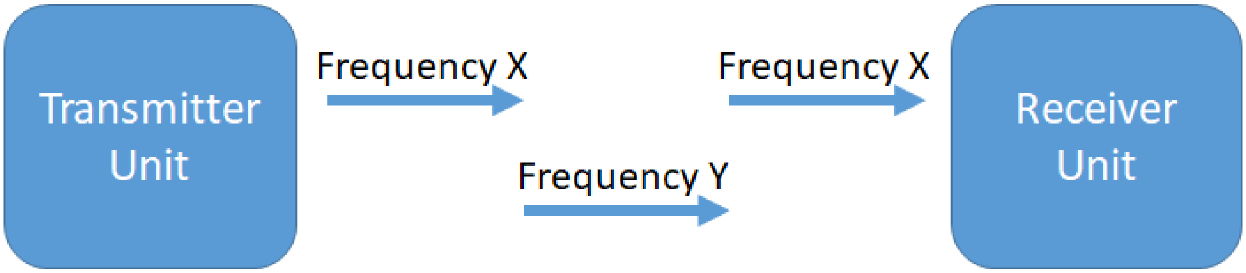


Figure 2.29: The above figure displays how the Transmitter unit will switch between frequencies when the statistical analysis determines that there is a better frequency to transmit on.

Coding/Implementation Details

The primary software package being used for the project is GNU Radio and the associated Companion.[4] This library and associated software enables communication with the software defined radio. The companion provides an interface to manipulate blocks that perform various signal processing functions. Examples of these include filtering, resampling, and throttling. The key with the project is that these functions are implemented in software rather than hardware which enables dynamic control from the mapping units and cheaper cost at the expense of performance. However, the nature of the project does not require maximum performance as there are no hard real time requirements. These blocks are able to be run from the companion and are converted into associated Python or C++ executables. For this project the group will be using Python due to the ease of implementation.

The nature of the project requires methods that are not included as default GNU Radio Companion blocks, so it is necessary to directly manipulate the Python files being generated.[4] The companion makes this achievable through the use of function probe blocks, which enable Python functions to be run on the associated blocks. The blocks have been manipulated in order to record timestamp information, change the center frequency, and perform signal detection using power detection. Other resources needed to achieve this are a number of Python modules which are listed in the appendix.[1, 3] Implementation within the Python is done in an object oriented fashion where the flow graph itself is an object. All of the blocks, variables, and parameters are set in the constructor. The object has getter/setter methods for all variables which were used in order to manipulate the center frequency with a function probe to enable scanning. The connections are defined using a connect method. The program works by continuously running the flow graph at the fastest rate possible unless it is throttled or guided by a sample rate variable. This Python program can be run from terminal and operate on the collection device.

This program will be placed on the microcontroller and will run at startup. The device itself is operating using headless variant of Raspbian. This microcontroller will interface with the storage, software defined radio, and regulation circuit mentioned above. The transmission program will also be written using GNU Radio Companion and Python.[4] It will use similar features to the data collection stage with the addition of transmission and getting information from a predictor module. It will read from the model produced from the analysis to determine the best bands and set its parameters. GNU Radio can inherently handle different types of transmission (wide/narrow band FM, AM, qpsk, ofdm).[4]

The statistical analysis will be completed using python in combination with the scikit library. A stretch goal for the project could involve performing these operations on the microcontroller. This is not an immediate priority as this would impact the amount of samples collected as these calculations would likely be computationally expensive and drastically reduce battery life.

Design Evaluation Methods

Analysis Cases and Results

One of the major constraints the team had to contend with was the weight of the UAV and the amount of payload the UAV could carry. Initially, the team considered purchasing a Matrice 100 UAV which weighed 2431 grams - leaving only 443 grams for the enclosure, after the battery and electronic components were factored in. In addition, the team determined that they could create a more efficient UAV with less money. Using this philosophy, the team began determining the specifications of their own UAV. The teams spoke to their stakeholders who determined that the payload weight should be at least 3 kg. Accordingly, the team determined that the UAV batteries would produce a total of 14.8 volts in order to spin 5 propellers, creating a thrust of 53.55 newton's. This gave the team the ability to carry, after the weight of the UAV was subtracted, 3,680.84. This analysis was completed using excel optimizer. Several of the variables are shown below in Figure 2.30

OPTIMIZATION			
motor	5	1,5	
prop	2	1,5	
batt	1	1,2	
FIXED			
Batt Voltage	14.8	V	
RPM	20720		
Prop Dia	10	in	
Prop Pitch	4.5	in	
Thrust	53.55125674	N	With atmoshperic density being 'standard day' at 1.225 lg/m^3
Thrust (g)	5458.843704	g	
Total Propl. W.	1778	g	
Payload Weight	3680.843704	g	

Figure 2.30: Weight optimization

In addition to the payload that had to be determined, the team needed to ensure that the center of mass of the enclosure lined up with the center of the UAV in order to ensure stability of the system. Shown in Figure 2.30, is a printout of the mass properties from the teams solidwork model, which specifies the center of mass. The mass is slightly shifted to the side which is why the team determined that they would redesign the enclosure as shown in Figure 2.16: Final design sketch for mapping unit enclosure.

```
Mass properties of Static Unit encasement
    Configuration: Default
    Coordinate system: -- default --

Mass = 0.67 pounds

Volume = 18.41 cubic inches

Surface area = 210.53 square inches

Center of mass: ( inches )
     X = 2.65
     Y = 1.89
     Z = 3.17

Principal axes of inertia and principal moments of inertia: ( pounds * square inches )
Taken at the center of mass.
     Ix = (1.00,  0.00,  0.01)      Px = 0.94
     Iy = (0.00,  0.99, -0.11)      Py = 3.95
     Iz = (-0.01, 0.11,  0.99)      Pz = 4.22

Moments of inertia: ( pounds * square inches )
Taken at the center of mass and aligned with the output coordinate system.
     Lxx = 0.94          Lxy = -0.01          Lxz = 0.02
     Lyx = -0.01         Lyy = 3.96           Lyz = -0.03
     Lzx = 0.02          Lzy = -0.03          Lzz = 4.22

Moments of inertia: ( pounds * square inches )
Taken at the output coordinate system.
     Ixx = 9.99          Ixy = 3.32           Ixz = 5.59
     Iyx = 3.32          Iyy = 15.28          Iyz = 3.96
     Izx = 5.59          Izy = 3.96           Izz = 11.26
```

Figure 2.31: Mass Properties of Concept 2 Model

The historical analysis will determine what frequencies are the most likely to contain white space at a given time and weather conditions. The optimal frequency to transmit on will be statistically determined from the large amounts of sample data that the mapping units will be gathering. The purpose of the project is to determine if there is a relationship between time of day, day of week, and weather characteristics to the signals being transmitted on selected frequencies. Historical data and transmission/reception testing are still required before it is possible to determine how strong the relationship of the parameters to the frequencies are.

Impact on System Design

The analysis cases for software determine the bands of interest and what signals to search for. The project at the moment is primarily focused on FM transmission in the unlicensed band. A current goal involves sampling each frequency once per second which determines the speed at which the program needs to run. This rate can be calculated based on the band size being monitored and reflected within the Python. Analysis of the cases has also lead to the formation of the transmission protocol. The power requirements have also lead the group to performing the statistical analysis using excel/libreoffice calc rather than performing it directly on the microcontroller using a Python library. Having this done on the mapping unit will be computationally expensive considering the vast amount of records and the low amount of RAM.

The analysis of the power requirements determined the need for a microcontroller with the option for analog inputs, so that the remaining battery level for the D-Cell power pack can be determined.

Prototyping: Plan and Budget

In order to create an alpha prototype for the mapping units, a Raspberry Pi 3 B will be used in conjunction with a RTL-SDR, and they will both be mounted on a custom built UAV. The mapping unit will have a 3D printed custom enclosure to provide protection from weather and keep all components secure during flight. The enclosure shall be mounted on top of the drone, with the drone's batteries underneath, so that the antenna can extend upwards above the drone. The enclosure shall be constructed and mounted so that it evenly distributes the weight of all components, so that it does not hinder they UAV's ability to fly.

The Raspberry Pi 3 B will operate with Raspbian and utilize GNU Radio and certain necessary Python modules. In order to do this it was necessary to install all of the necessary files for running GNU Radio. Additionally, all of the drivers needed to run the RTL-SDR needed to be installed to the device as well. The project is managed with Git and the Pi is set up as well for ease with change control and an easy solution to extracting recorded data files.

The software on the prototype will be run from a Raspberry Pi and a user's laptop. In order to implement this the current programs need to be transferred onto the microcontroller. The microcontroller also needs to be configured to interface with the software defined radio, user, and the enclosure unit. This requires an operating system, necessary libraries such as GNU Radio, a programming environment, and the necessary hardware drivers. Communication with the mapping unit will be achieved using SSH or ethernet, and controlled through terminal until a GUI is developed. Statistical data analysis will be performed on the laptop in order to save battery power. Data collection will begin and will run for multiple weeks in order to gather enough data for early statistical analysis. The results of the binary file format will be converted into a csv file and a statistical analysis will be run to highlight variables of importance (hour of day, minute of hour, weather, etc).

The enclosure of the collection unit is subject to change in the final device, but for the prototype device, it is necessary to allow enough space for the software defined radio, battery power bank, and the Raspberry Pi. It is also necessary to provide enough room to allow for necessary cooling of the devices to prevent overheating. The initial enclosure will attach to the UAV via four mounting screws that will prevent the mapping unit from detaching from the aerial vehicle. A frequency analysis will be performed in order to stabilize the enclosure and prevent any damage to the mapping unit. For the alpha prototype version of the enclosure, ABSplus thermoplastic will be used to fabricate the design via additive manufacturing. In order to waterproof the design, a series of compact plastic submersible cord grips will be utilized in the enclosure fabrication in order to allow wire access from the SDR to the antenna, which will remain outside of the enclosure due to space requirements. Additionally, there will be a separate battery bank that will house the one 10,000 mAh Romoss Solo 5 battery powering the microcontroller. This will also be constructed out of ABSplus plastic in a similar manner to that of the enclosure for the purpose of the prototype build. Further compact plastic submersible cord grips will be used here in order to connect the batteries to the raspberry pi while still maintaining a waterproofed design. In the future the team may replace the plastic with aluminum for a more robust and longer lasting design.

For the transmission stage, two HackRFs have been purchased and will be used in conjunction with a laptop through GNU Radio, as the Raspberry Pi does not have the necessary processing power to utilize the HackRFs effectively. The transmitters will attempt to share information on bands that were previously analyzed, without stepping over any other transmissions that might occur on the band.

Testing and Evaluation: Plan

Collection Testing

Testing of the mapping unit will be done by confirming that known signals are detected during a given interval. The unit will scan well-known bands such as common FM radio stations and try to identify the locations of each station. The results will be checked against a list of known radio stations on those bands.

For further testing with lower-powered signals, a common off the shelf two-way radio such as a walkie talkie will be used to transmit a voice message to another radio. The collection unit will be tuned to watch the known frequency of the walkie talkie and record data. The channel will be used at predefined times, and the results of the recording will be checked against the testing plan to see how many transmissions were detected. The alpha prototype should be able to determine all times for channel use and idle. Using an RTL-SDR as the receiver with the Raspberry Pi as the controller, a maximum sample rate of 2 observation windows per second is achievable. By default, each observation window is 2MHz wide and subdivided into 1024 on/off ‘bins’, allowing multiple signals to be detected. Each window is observed for 75 samples, and analysis is done on the averaged result.

Transmission Testing

The results of the collection and data analysis stages are critical to the transmission stage, and must be refined throughout the testing process. After bulk data collection and analysis it will be necessary to determine if the model is accurate and producing viable results. Due to the size of the data collected, observational validation on the data is impossible. Therefore, validation of the statistical analysis will be conducted by running the transmission and identifying throughput, as well as the number of collisions.

Testing the software component of the project will involve using the data generated during the collection stage to identify optimal transmit bands and times. A stationary transmission station consisting of two laptops, one with a HackRF and RTL-SDR and the other with an RTL-SDR will be given the results of the data analysis stage. With this information, these devices will know where to transmit and receive at any given time. The transmission unit will send data on these optimal frequencies and the receiver unit will record all transmitted data. Since the transmission unit can not overwrite or interfere with other transmissions on a given frequency, it must check the frequency that it is attempting to transmit on for other transmissions. In the event that it determines that the frequency is in use, the transmission unit will fall back to to a backup band when necessary. In the event that all of the frequencies available to the transmission unit are in use, then nothing will be transmitted by the unit.The receiver unit will show all bands in operation, and it will be possible to visually confirm that the transmission unit is avoiding artificial interference caused by handheld radios during the test.

Software Testing

It is necessary to test the alpha prototypes software to determine that the transmission unit will transmit on the correct band at the correct time. The statistical analysis will determine the band that is most likely to be free for transmission. However, it is still necessary to check the band through signal thresholding to make sure that there are no other transmissions before the transmission unit is able to transmit. In the event that the band is found to have outside traffic, the transmission unit will fall back to the next best band determined by the statistical analysis.

This test will be done using 3 common off the shelf walkie talkies. One will be used to provide outside traffic on a band. To test that the transmission unit will not transmit over other users, a walkie talkie will transmit continuously over the highest rated channel from the statistical analysis. When examining the data afterwards, the group will be able to determine if the transmission unit had fallen back to a different channel. This test will be run with 1, 2, and 3 walkie talkies. On the test with 3 walkie talkies, the transmission unit should not transmit at all since it only has three channels to choose from.

Battery Power Testing

The alpha prototyping testing for the battery will be done by running the collection unit until depletion. The battery was selected for having a minimum duration of 20 hours. Total battery capacity for the selected battery is 10,000 mAh. The total estimated duration is approximately 25 hours. These estimates are a worst case scenario, so the results of the prototype may show that the collection unit will be able to run for much longer. In this case of longer lasting batteries, the statistical model that will be made from the collected environmental data will be much stronger. If the battery fails to last 36 hours during data collection, an additional battery pack can be added. Adding additional batteries will require an enclosure redesign.

Enclosure Testing

In order to test the enclosure, a test to measure how much moisture will be let into the enclosure will be conducted. This will be accomplished by stuffing paper into the enclosure and pouring water onto the unit to simulate rainfall. All electronics inside the enclosure will be removed. The paper will then be removed from the enclosure and will be analyzed for moisture. In the event that the enclosure keeps all of the water out of the system, the paper towels will be found to be completely dry. This will result in a pass for water resistance. If the paper towels are found to be wet, the test is a failure and redesigns will be performed as necessary. Due to scheduling constraints with the fabrication process, the team will be simulating the external aluminum shell with band saw cut acrylic sheets that presume a similar geometry.

Drone Testing

The drone will be tested by conducting two simple hover maneuvers. The motors and the frame will be observed for vibrations. The flight controller and other electronics will checked prior to flight to ensure proper connections and setup. The flight controller will be tested for stability control during the first flight. The second flight will be used to attach the enclosure to the drone and test the hover again. This will allow the team to check for any weight imbalances and vibrations induced from the enclosure connections to the drone. All issues will be noted and the unit shall be redesigned accordingly. The alpha prototype will only test the first maneuver.

Testing and Evaluation: Results

Collection Testing

Initial testing has been done to detect known FM radio stations. A sample result is shown below in Figure 2.32. The green line is the output of our detection algorithm, scaled to make results more apparent. The algorithm is discussed in the Data Conversion subsection of System Design. 0 means no signal is detected in a given band, 18 means a signal is present. This test was done with a sensitivity threshold of 0.5 and an averaging size of 512.

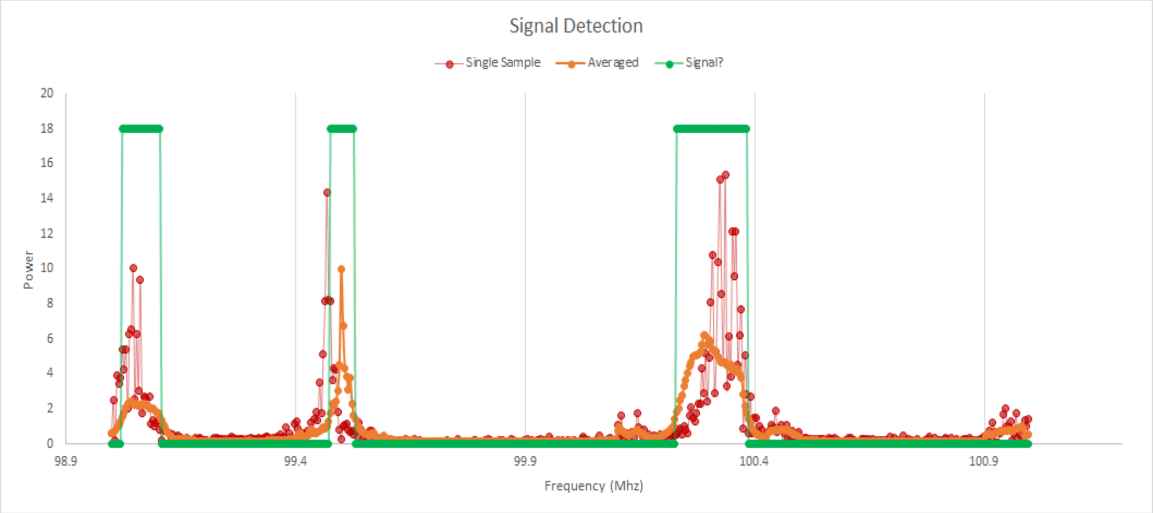


Figure 2.32

The frequency of interest was between 99 and 101 MHz, with signals detected at 99.1, 99.5, and 100.3, corresponding to WAWZ out of Zarepath, WBAI, and WHTZ (Z100) from the top of the Empire State Building.

Drone Testing

The drone was tested by applying power and spooling the motors. Once there was no indication of any loose connections, the throttle was increased until the drone was slowly hovered. The flight controller was tuned to ensure the drone did not drift in any different directions. The center of gravity was also tested by using the bluetooth module to analyze motor inputs and were ensured that they were all roughly equal. During the second hover the drone tipped over possibly due to balancing issues. This caused one the screw holding one of the arms in to shear off moving the arm. This was fixed by adding longer screws that fully threaded the nut.

Beta Prototyping: Design Revision and Optimization

Mechanical Design

After the alpha prototype, it was found that the original design had issues with heat dispersion for the Raspberry Pi. Although the enclosure was made to circulate air for passive cooling, the design was still having problems. After roughly 4-12 hours of running the collection script, it would often crash due to thermal problems. The Pi would be running upwards of 85 degrees. To counter this problem, the group will be adding a iUniker Raspberry Pi Heatsink Fan Dual Fan to the design to implement active cooling. The fan should hopefully alleviate most thermal problems that the group has ran into at the cost of battery run time.

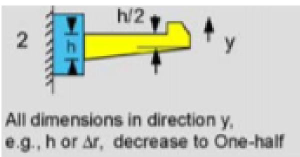
The team was also able to make significant progress on the drone they had been working on. The self tightening counter treads, which would ensure that the propellers could safely rotate without coming undone from the motor. The team was also able to perform several, proof of concept, tests which showed the drone was capable of hovering. Unfortunately, due to Stevens private policy, the team is not allowed to perform extensive testing that is necessary to fully test the drones flight capabilities. The team plans to test the drone flight more extensively at a private location away from Stevens.

From the limited testing the team was able to perform, the team learned several things about their analysis. First, that the current battery capacity is not great enough to support the drone for an extended period of time. Secondly, that the amperage of the battery is not producing optimal results, the total thrust the team was able to get out of the drone was lower than our analysis predicted. To remedy this, the team ordered a higher capacity battery that should be arriving for the next milestone presentation. Overall the team was pleased that the drone was able to achieve a hover but, with a larger battery, the results could be greater.

Modifications to the enclosure were made for the beta prototype as well. A second version of the internal frame was designed to better fit all of the electronics at tighter tolerances in order to reduce movement mid flight. This was achieved by examining the fitting of the version one frame and adjusting the dimensions from there. Having the version one was a great test to determine the accuracy of the printer, and allowed for easy modifications to be made. Optimization of the version 2 frame included several major additions. The most substantial was the better design for the snap-fits which house the SDR in place. This was achieved through combination of beam calculation design as well as a static analysis in SolidWorks Simulation. The equation for dimensioning the cantilever snap joint may be seen below in Equation 2.33, and these designated symbols in respect to the beam and cross section in Figure 2.34. It became apparent that a trapezoidal cross section would provide the optimal support for this design.

$$y = 1.64 * \frac{a+b}{2a+b} * \frac{\varepsilon * l^2}{h}$$

Equation 2.33: Dimensioning Trapezoidal Cantilevers with Permissible Deflection



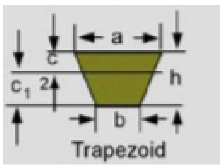


Figure 2.34: Design for trapezoidal beam deflection (type of planar view design on left and shape of cross section on right)

The static analysis was a great way to check if the material plasticity would be capable of withstanding the force of the user's force to press open the snap fit, which in this case was assumed to be a conservative 20 N of force in both directions. This analysis can be seen below in Figure 2.35, in which the maximum von mises stress was found to be 2.556 * 10^7, while the yield strength for ABS plastic is around 30 Megapascal, indicating no buckling will occur.

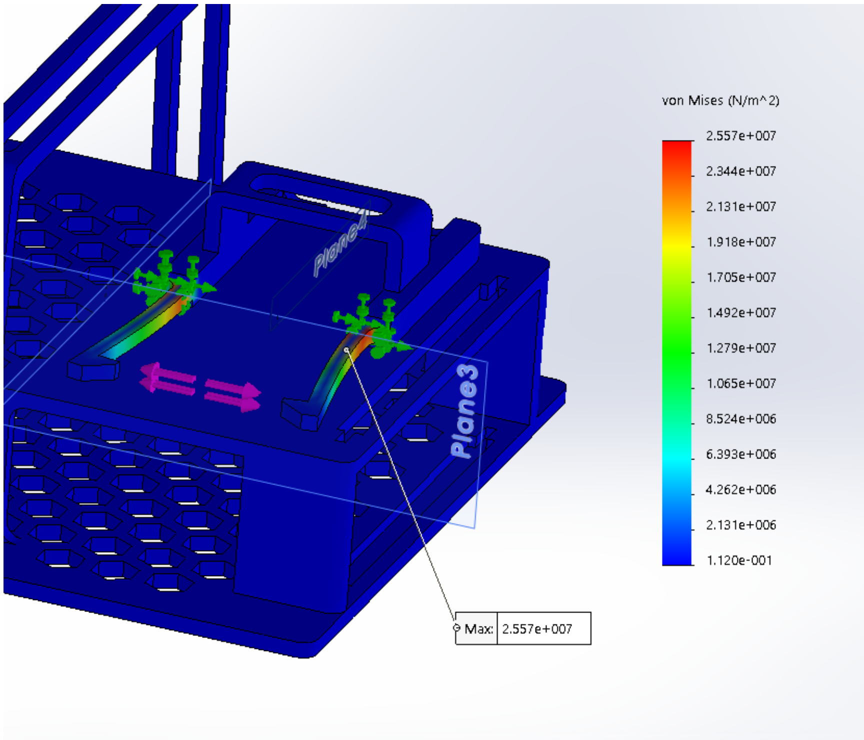


Figure 2.35: Static analysis for the snap-fit design

Optimization of the entire part was conducted in a design study using SolidWorks Simulation. Mass was set as the goal to be reduced based on the dimensioning of several holes dispersed throughout the part. The study optimized the design to reduce the overall mass from 101.23 grams down to 83.63 grams by increasing the size and placement of the holes. The most recent model may be seen below in Figure 2.36. Also important to note was the reduction of the standoff diameters for the raspberry pi to reduce contact between the board and the standoff. The back of the lower level of the frame was also extended to accommodate for any USB output to the battery. The handle was also refined to make contact with both sides in order to provide for sturdier support.

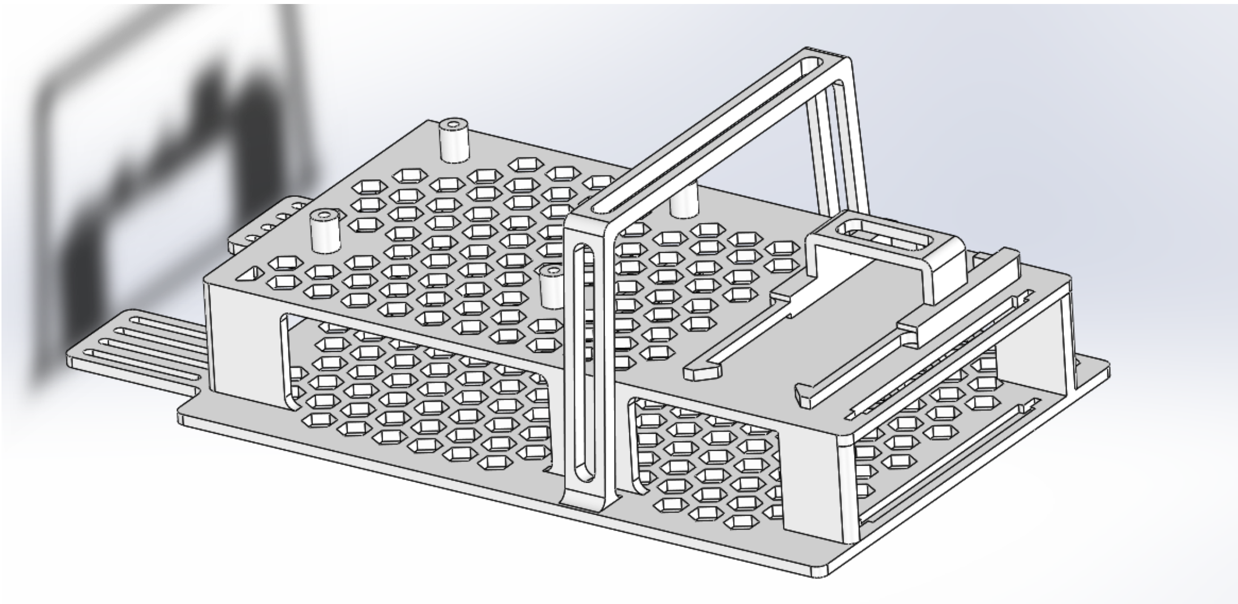


Figure 2.36: Printable Internal Frame Model V2

The external enclosure was decided to be constructed of 3D printed plastic and printed to fit the internal frame at a very tight tolerance so as to prevent movement of the frame and any damage to the electronics during flight. The design for this was mostly sketched out by hand, and designed with the intention of cutting, bending, and welding the sheet metal into the desired shape. This will be further discussed in the fabrication, assembly, and manufacturing section. Due to complications with the welder, though, the team decided to have a laser cut acrylic alternative enclosure as a backup if further complications arise.

Electronics/Electro-Mechanical Design

The addition of the fan will have a large effect on the battery run time due to it's additional .3A current draw from the Raspberry Pi's battery. Due to the increase in current draw, the batteries projected run time has been lowered by 57 to 30 hours. This is due to the extra 85% more power draw for the fan. Another change was the removal of the interface between the Raspberry Pi and the drone for the beta prototype. This is because there were no relays on the market that could control the 120 amp 12 volt output given the 5v input. A custom solution would have to be engineered to accomplish this, which for now is out of the scope of the project. Instead the Raspberry Pi will be manually triggered using a button or switch while the drone is switched off.

Software Design

The overall software architecture and planning has changed significantly since the onset of the project. The alpha prototype testing led to changes in the implementation of the signal detection method, user interface, and general data collection. The data collection now incorporates weather information which is received using the Weather Underground API. A script was written that pulls weather information for the desired location, parses out the fields of interest, formats appropriately, and outputs to a csv file. This script does not require polling and is run at set increments from the GUI. The weather fields of interest include the humidity, whether or not it is raining, and the timestamp that the data was collected. This timestamp is used in the unpacking script to graft together the output .csv file. Another new feature for the beta prototype is the use of configuration files for user parameters rather than command line flags. The configuration files are generated using the GUI and are read by the collection script which is much cleaner than using flags. These configuration files are then used by the unpacking script to collect variables rather than that data being written as header information on the binary data.

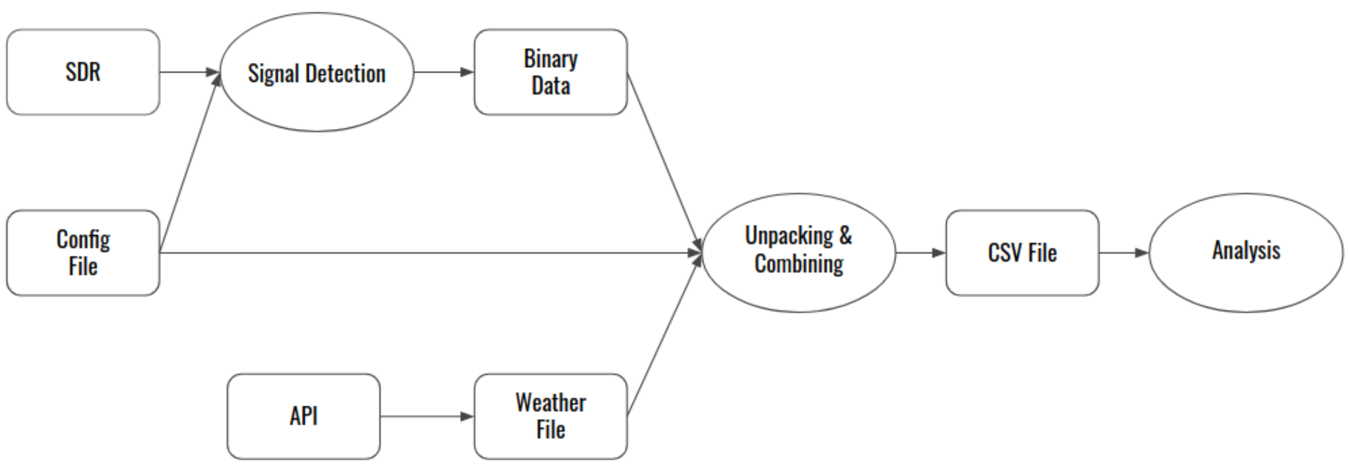


Figure 2.37: Revised software flow including weather info and configuration files.

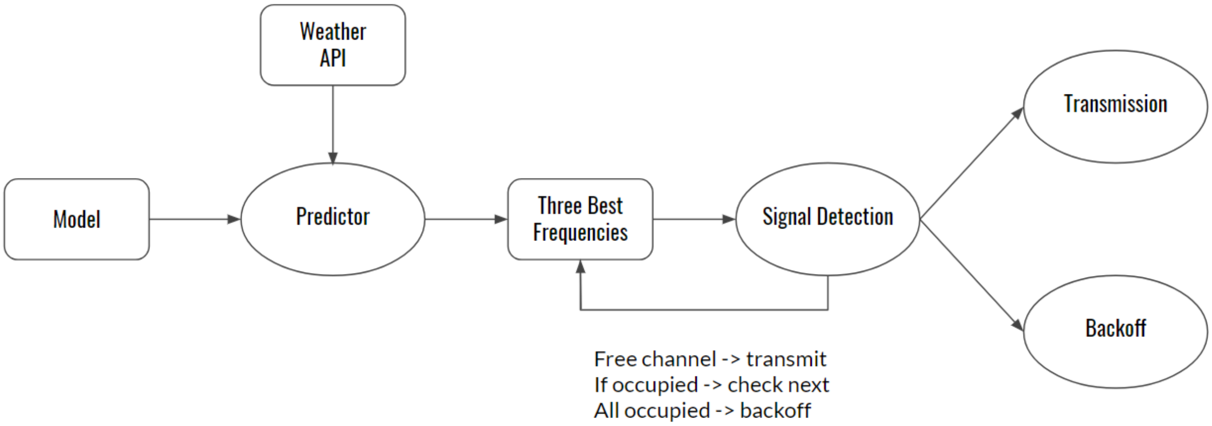


Figure 2.38: Revised software flow continued.

The user interface has also received a major overhaul and new features. In terms of functionality the GUI can now handle uploads/downloads to an SFTP server hosted by the IDEAS program, creation of configuration files, weather collection, data conversion, and a live demo of the signal detection method. These functions are accessed through either the menu bar or notebook interface. The UI features were developed with the TKinter python module. The GUI has also become more robust in order to avoid blocking the UI thread. This involves the use of multithreading, subprocesses, and using a queue to read the standard out of other various scripts. The live demo is the most critical new addition which highlights the detection method in real time. This is displayed using three squares that are colored according to whether or not there is a signal on the selected frequency. The user selects three bands based upon the channels from a walkie talkie and starts the detection subprocess. The three squares then change from red to green based upon if there is a signal on the channel.

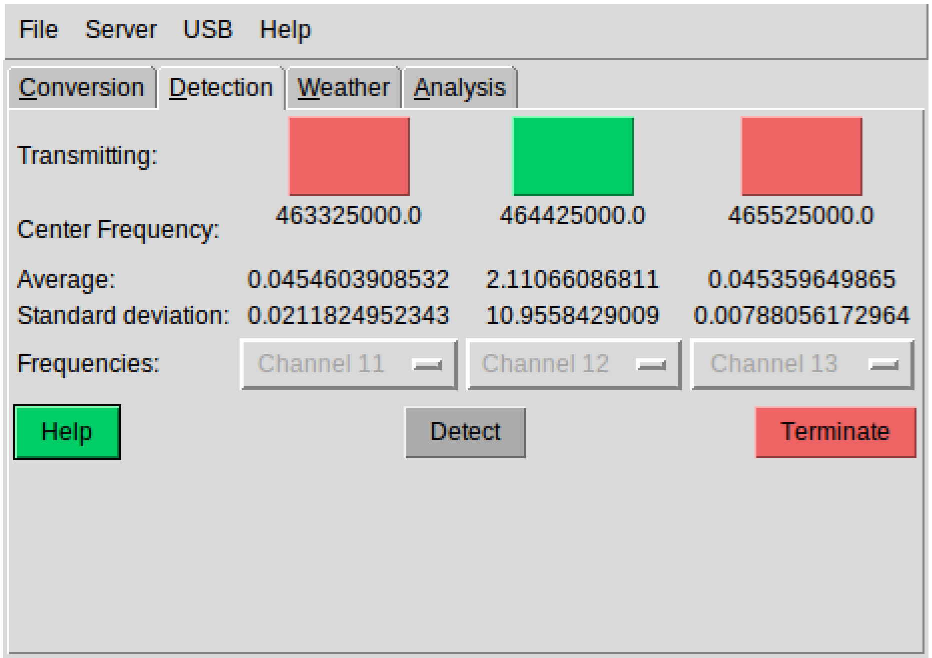


Figure 2.39: GUI detection test sample image. A signal is present on Channel 12. Note: in newer editions of the GUI the color scheme has flipped where red = occupied.

The HackRF will be used for the transmission portion of the final version of the project. The HackRF will use GNUradio to transmit a WAV file using the Narrow Band Frequency Modulation (NBFM) transmit block. The statistical model will be used to determine what channels are the most likely to be free at a given time. Before transmitting, the script will check the most likely to be free channel to see if it actually free to transmit on. In the case that the channel that was chosen is not available, it will fall back to the next best channel from the statistical analysis and check the band again. In the case that the frequency is available, the HackRF will transmit the WAV file on the selected frequency. To test that this method is working, the group will listen in on the different frequencies using other SDRs. If the listeners see that the WAV file is moving around between the selected channels based on their availability, the solution is working as intended.

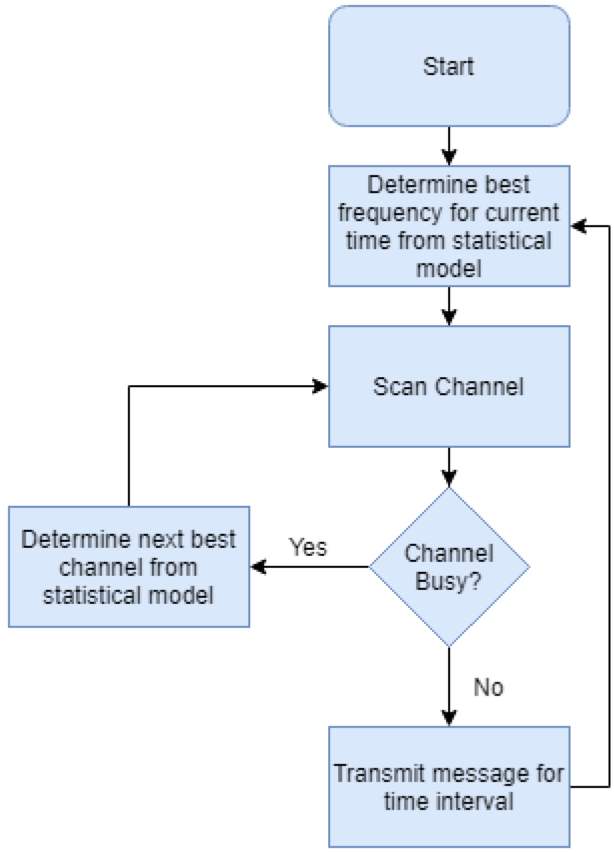


Figure 2.40: Transmission flow graph

Statistical analysis was completed in python using a few python libraries. The libraries can be found in Appendix D. The recorded data was kept in files using the CSV format. This data was first loaded into a single table. In this table each row was a frequency sampled at a certain time stamp. The columns in this table represented the weather conditions, and the current time split into hours minutes and seconds. After loading the information the data was split into multiple tables, each table represented the data for a single data point. In these tables the predictor values were precipitation (0 or 1), weather conditions (see Appendix D), humidity %, cloud cover, hour, minutes, seconds, and day of the week. The predicted value was if there was a signal on the chanel.

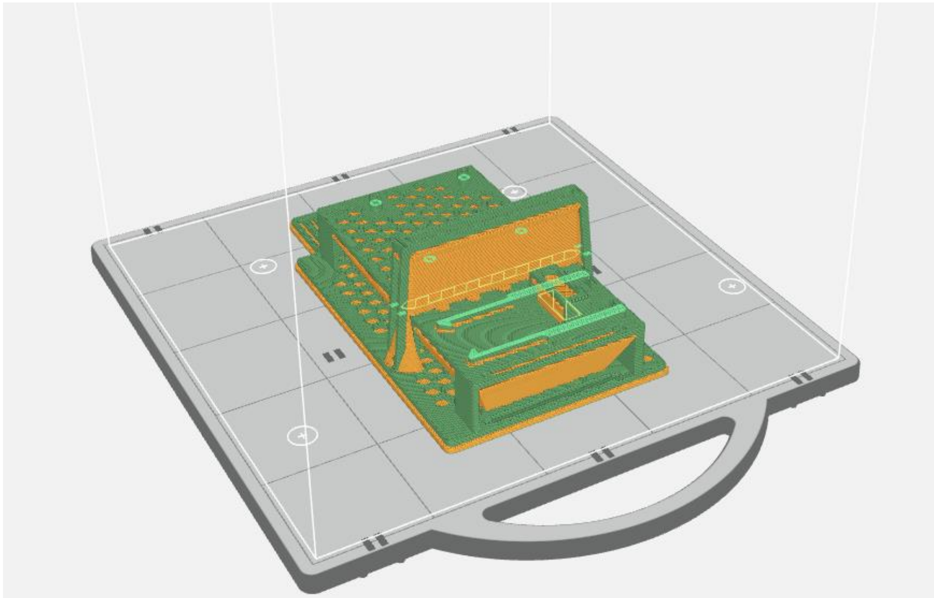
Each frequencies data was then split into training and testing data. The testing data is on only used for model validation, the final prototype will use all data for training. Once the data was split for testing and training, it was split again into predictor variables and predicted variable. Next a logistic regression model was built for each frequency. This type of model was used because it allows for classification of data using a binary classifier, in our case, this classifier is the status of the singal on the frequency. Each model is trained with the training data that was created early. The models are then validated using the testing data. This reports the accuracy of each model.

Finally, by using the area under the ROC (Receiver Operating Characteristics) curve, which gives an estimation of the predictability and sensitivity of the model, we can rate each model. This reports the top three frequencies with the best models that have the highest chance of predicting the appropriate time to broadcast. The final models can be used to predict the frequency to use by looking at the current weather and time and identify which frequencies will empty and returning the one with the highest predictability.

Fabrication, Assembly, and Manufacturing Processes

Construction of the drone was mainly performed with hand and power tools. The frame was the first assembly process and involved screwing in the connecting arms, motor housings, and inner levels together with screws and snapfits when in transport. Upon completion of the frame, the electronics were then assembled, and included soldering of the ESC's to the respective motors, and wiring the flight controller to the connecting wires from each motor. The GPS unit was also connected to the flight controller and mounted on top of the frame. Maintaining the wires neatly between the two inner frame levels was another aspect of the build so that there were nothing outlying during flight. Connecting the propellers and mounting the battery underneath the frame were the final steps that took place during construction.

Fabrication of the internal frame for the enclosure was conducted via additive manufacturing. A Stratasys Dimension 1200 SST was intended to print the second version frame in ABSPlus plastic. Some of the Stratasys printers were unresponsive in the Stevens PROOF lab due to the high demand of parts being printed at the same time, and so the team's design file was sent out to the manufacturer Stratasys in order to have them print and ship the file back to the lab. Print time and the sliced object with supports may be seen below in Figure 2.41. This was determined using GrabCAD Print software.



Tray Estimations		
Printable Frame v2	Print Time	12h 51m
	Model Material (in^3)	4.896
	Support Material (in^3)	4.619

Tray Estimations		
Front Door	Print Time	34m
	Model Material (in^3)	0.385
	Support Material (in^3)	0.161

Figure 2.41: Sliced object with print times for printable frame V2 (left) and battery door (right)

Assembly of the outer aluminum enclosure has also been underway. Manufacturing was modeled around the notion that a spot welder would be utilized to fuse the walls together as this was the only available source of welding to the team. With that in mind, there needed to be folded panels made 90 degrees to the main walls. This was because spot welding requires two parallel surfaces making contact in order to fuse pieces together. This made the bending challenging, as the walls were not flush with the bottom surface due to the bending of the folds. The sheet metal was first measured out to the dimensions of the internal frame, and then cut to the desired surface area in two dimensions. From here, four cuts were made to separate the four walls from where the folds were located. At this point, the folds were bent inward, followed by the two adjacent walls. All folds and walls were clamped together, and the spot weld was attempted but unsuccessful due to the material and thickness of the material. There was too much voltage running through the spot welder’s prongs into the thin strip of metal in order to adequately fuse together. The current status of the aluminum enclosure is documented below in Figure 2.42.

Figure 2.42: Aluminum enclosure prior to welding

At the time the progress toward fabricating the enclosure leaned toward laser cutting acrylic sheets and then binding together with an adhesive to form the internal fit of the printable frame. The EM shielding will be maintained through aluminum sheets which will deposit alongside the acrylic enclosure’s inner walls. The top panel will be fastened with a 3D-printed L-bracket that will allow screws entry into both the top and side panels through drilled out holes. The submersible cord grip will still exist here to feed the antenna to the outside. Where issues may arise is in the teams continued issue when dealing with the heat flow brought about by the raspberry pi, as the acrylic enclosure maintains poor heat conductivity and may be lacking in heat mitigation. After this the enclosure was mounted atop the UAV.

High-Performance Testing and Evaluation

Thresholding and Detection Testing

From the testing the group did with the walkie talkies, it was found that the SDR could listen for and respond to only nearby radios, with some modification to the collection unit code. The SDR was programmed to work in conjunction with a GUI, such that it lights up indicators whenever it detects one of our walkie talkies transmitting nearby on an appropriate band, and ignores more distant signals on the same band.

One irregularity observed in the walkie-talkie test was a delay in a detected signal going silent, sometimes causing other channels to show false-positives. It was determined that not enough time was being given for the vector stream from the SDR to replace all the samples from the previous channel after switching to a new channel, resulting in contamination between channels when trying to detect signals. This issue was initially fixed by adding a static delay after switching channels, then by forcing the program to wait and count incoming samples after switching channels, ensuring that the entire vector stream was loaded with fresh data.

The test shown in Figure 2.43 below was run to show the effects of averaging size, essentially the number of times each observation window is observed before switching to the next. From left to right, the amplitude at each frequency bin is shown from 461.925 to 463.925 MHz. From front to back, each colored line indicates a run with a different number of observations: 1,5,10,15, 20, 30, 40, 50, 75, 100, and 200. The dark red line in the front shows the result of only one sample, a very noisy plot where some of the low-power signals are completely drowned out by the noise in the sample. The orange line directly behind it shows the result of 5 samples, with the noise still very visible but greatly reduced. Skipping to the first yellow line, which shows the result of 20 samples, the overall noise level is greatly reduced and the “empty” space between detected signals looks quite flat.The trend continues as more samples are taken, with unused bandwidth becoming flatter and detected signals standing out further. On the graph, this is most immediately apparent looking at the rightmost section of the graph that becomes flatter and flatter moving from front to back. For the purposes of detection, 75 samples was chosen as a good default value to reduce most of the noise. The original default value was 512 samples. Due to the fact that every sample requires a new FFT to be calculated and averaged, there is a direct relationship between the averaging size and the time it takes to observe a channel. Reducing the averaging size from 512 to 75 means that channels can be cycled through roughly seven times faster than in the original Alpha prototype.

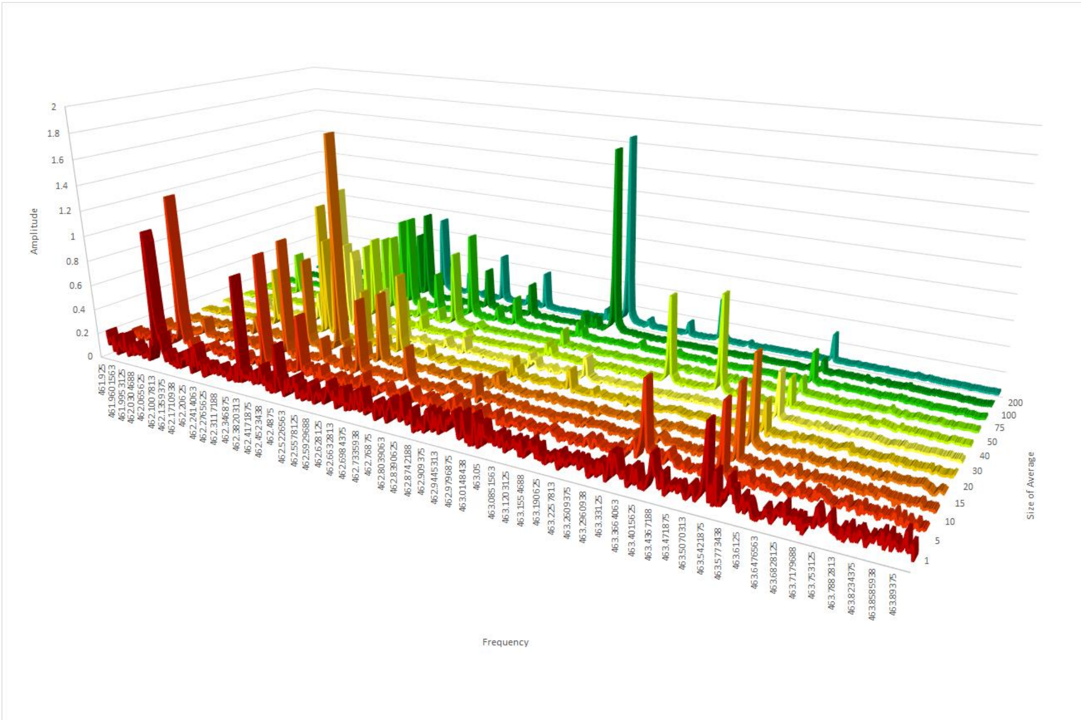


Figure 2.43: Effects of Averaging on Spectrum View

Figure 2.44 below is a zoomed-in view of the above graph examining a persistent signal on 462.925MHz, demonstrating how the signal becomes easier to distinguish from the surrounding noise as more samples are added.

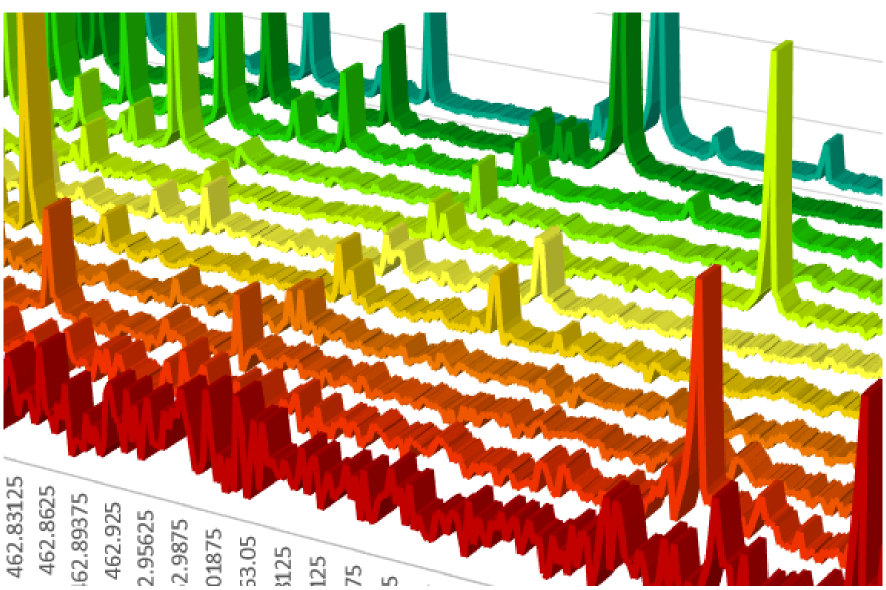
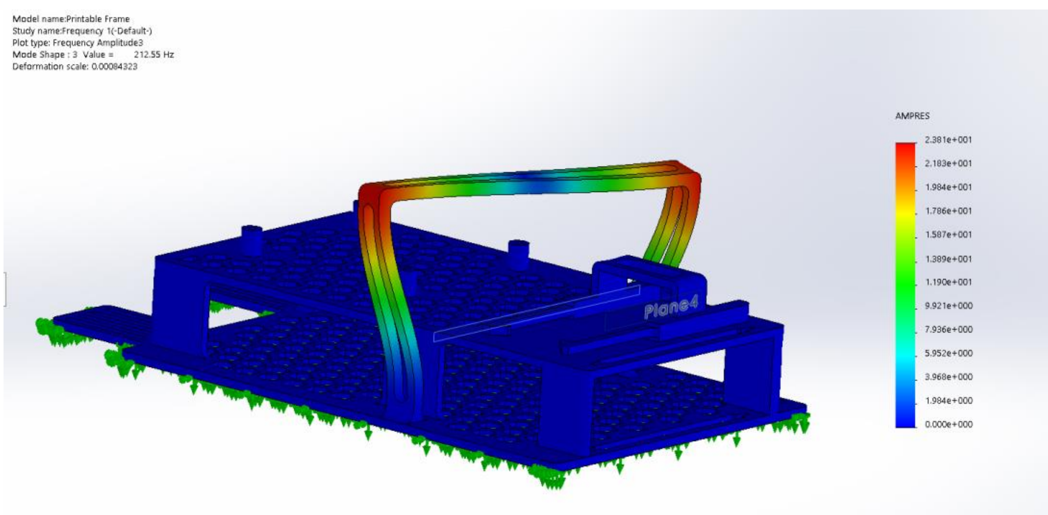


Figure 2.44: Section View on 462.925MHz

Mechanical Design Testing

The majority of the mechanical design testing comprised vibrational analysis of the enclosure, as the objective was to create a system in which the mapping unit's components were securely packaged and capable of being transported in harsh environmental conditions. Maintaining zero internal movement within the enclosure design was key in establishing a means of transport for the mapping unit. Due to a lack of labspace and equipment that could test the vibrational accuracy of said system, the team performed several frequency analyses of the enclosure to simulate the effects of the UAV motors and in-flight trajectories on the internal enclosure and electronics. The results for this simulation may be seen below in Figure 2.45, in which there was visible deformation present at operating frequencies of 212.55 Hz and 351.58 Hz, indicating a need to reinforce those areas incapable of withstanding the tested forces.



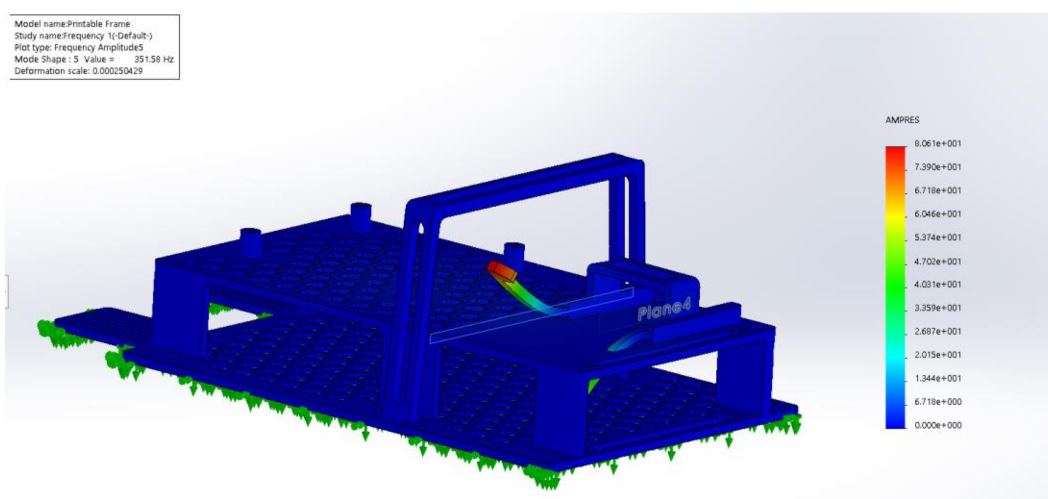


Figure 2.45: Frequency Analysis of enclosure performed at 212.55 Hz (upper) and 351.58 Hz (lower)

Analysis Testing

The statistical analysis and model was validated using data collected over two days. Only two days of data was able to be collected due to hardware limitations that caused the raspberry pi to overheat when collecting data using the SDR. Still the data was analyzed and the analysis report is shown below in figure 2.46. The two days were collected in 7 files with a total of 31.1 million data points. Two center bands were recorded with 1024 side bands for each center band frequency for a total of 2048 unique frequencies. The models could not be trained for frequencies that were only empty or always filled, therefore these frequencies were discarded. In total there were 1395 discarded frequencies, and only 653 frequencies that passed. The average training size was approximately 10,650 points for each frequencies. The mean testing score was .94 with a best possible value of 1.

This at first seems like a very good result but this is only because most channels were mostly just either on or off with the model predicting either on or off the entire time. Still, the models were rated using the testing (30%) data collected, some of these could not be rated as the data was rated perfect because there was nothing but ones or zeros. The frequencies with the best predictability were reported. Arguably the most important metrics are the average correctly identified positive rate and the correctly identified negative rate. These have the values of 37.90% and 64.38% respectively. These metrics give a good representation of how poor these models are due to the lack of data. For the final project submission the analysis will be run again with a much larger data set.

```
Analysis starting with 7 files.
Total of 31182848 data points.
2048 unique frequencies to analyze.
Spliting frequencies, 0 left.
Frequencies split.
Spliting test/train, 0 left.
Testing and training data split.
Training models, 0 left with 1395 discarded...
653 models trained with average training score of 0.94.
Average training size was: 10657.51 for the models.
Mean testing score: 0.94.
Mean correctly identified positive %: 37.90%.
Mean correctly identified negative %: 64.38%.
Rating freq with 122 skipped.
1. Freq 53734375.0 with roc score 0.97
2. Freq 53738281.25 with roc score 0.97
3. Freq 52076171.875 with roc score 0.97
```

Figure: 2.46: Statistic analysis validation results

Transmission Testing

The team conducting two tests with regards to transmission. The first was to see if the units can successfully detect signals produced by the purchased walkie-talkies and the second involved transmitting on empty frequencies. To test the signal detection the group created a detection script which was a slightly modified collection script. The file writing was removed and it purely output whether or not there was a signal on three provided frequencies. This script was then tied into the GUI which would update based upon the results. The team tested by causing interference and viewing the GUI results. When a signal was detected the GUI would display that information along with power information and standard deviation.

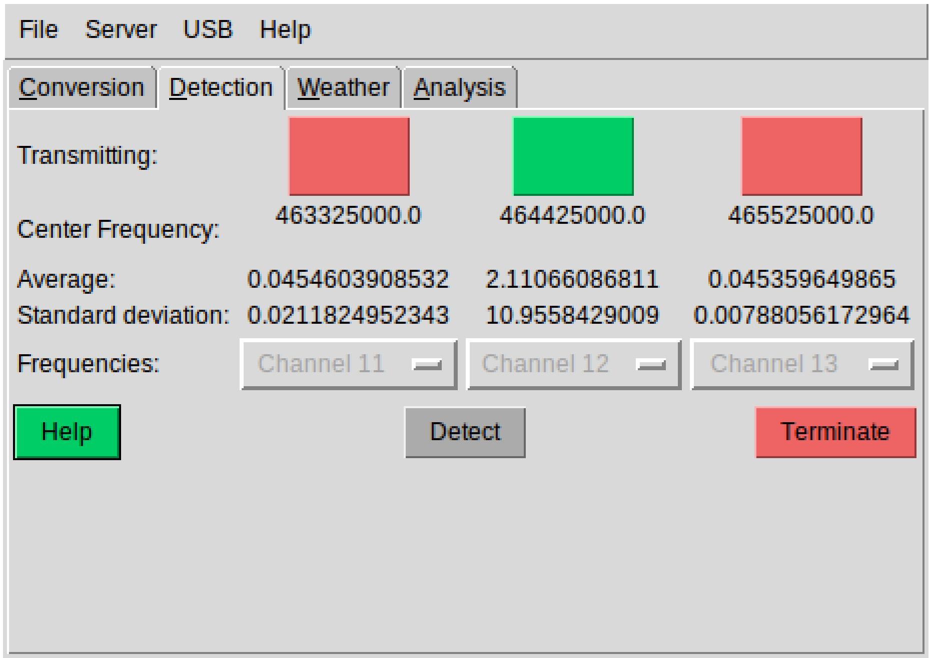


Figure 2.47: Screenshot of GUI during signal detection test. A signal is present on 464 MHz.

Transmission was tested by having the Transmission Unit play an audio file over the optimal bands as returned by the statistical model. Interference would be broadcast on the primary, secondary, and tertiary channels while the transmission was ongoing. A second computer running a visualization of the active bands would be observed to confirm that the audiofile transmission jumped to back-up bands whenever the primary band was in use, and ceased transmission when all bands were in use.

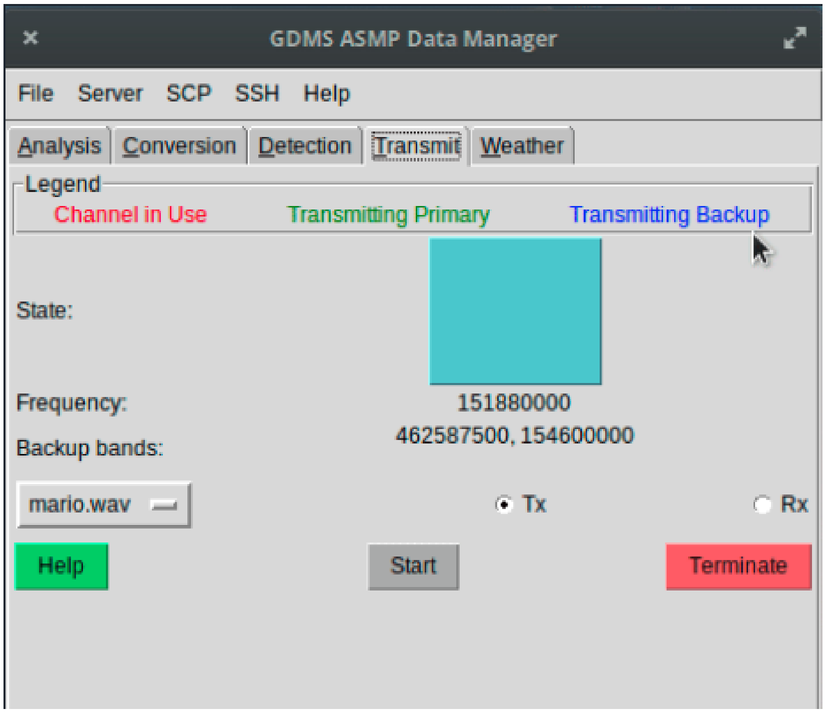


Figure 2.48: Screenshot of GUI during transmission testing. The HackRF is now transmitting on a secondary band due to interference on the primary.

Final Design Specification

Overall System

The overall system was able to meet 90% of functional requirements and 45% of non-functional requirements. The final specification can be split up into two subsystems, hardware and software. These two systems comprise of a UAV delivery system that is capable of transporting a mapping unit. This UAV has been custom built by the team in order to lift the mapping unit. Although small, the mapping unit is approximately 1.5 kg and consists of an enclosure to hold a Raspberry Pi, battery bank, and software defined radio. Once delivered this unit is capable of recording relevant whitespace information in the area. In terms of functional requirements all have been met with the exception of being able to operate in any desired environment. This is discussed below in the hardware specification.

The team was only able to meet 45% of non-functional requirements. This low score is the result of missing two key operations: requiring no user input after unit deployment, and having the ability to collect data for a week. The lack of a subsystem interface and mapping unit battery life are discussed in the hardware section below.

Software

The final design specification for software encompasses a number of components that will be explained separately. There are the data collection scripts, analysis scripts, transmission scripts, and the graphical user interface for use on a user laptop. These programs were designed to meet the requirements listed out in Milestone 1. The team was able to meet 100% of all functional requirements and 90% of the nonfunctional requirements. The missing 10% is considered open work and is described below.

The data collection scripts were designed to be run on a Raspberry Pi using python and interface with a software defined radio. They are able to read in data from the SDR, determine if there is a signal present using a static threshold based on power analysis, and compress the results into a binary file. The script can be controlled through configuration files that control parameters such as the frequencies to monitor, file size properties, and variables that control the sensitivity of the threshold. The second factor of data collection was weather information. The group was unable to access a weather API that included historical data due to budgeting concerns, so the data was instead recorded in real time. The information was gathered from the Weather Underground API and was polled once every hour. Data of interest included rain information, humidity, current time, and a weather status. The end result of data collection involved grafting together the weather and binary files to produce a csv to be used in analysis.

The statistical analysis completed in the analysis stage was run using an online model. This model sits in the working directory of the data manager python program. The model is based on stochastic gradient descent, and aims to mimic a support vector machine. The model was initialized to the default parameters offered by scikit learn. During every analysis run the model will be loaded and taught more samples. After the learning is complete the model will be saved back to the disc to be used during the transmission stage and the next analysis. Due to the nature of the data collected, the model was subset of models each applied to its on frequency. The features for each model were hours, minute, second, day of the week, cloud coverage, humidity, and weather condition.

After the analysis stage is completed the system is then able to transmit using the predictor described above to transmit on the optimal frequencies. The transmission is accomplished through the use of a HackRF connected to a laptop. The initial project scope involved transmitting from the Raspberry Pis, but this proved infeasible due to the limited processing power. The transmission script is able to determine the three optimal frequencies based on the current conditions, perform signal detection, and transmit for a brief window on empty frequencies. This runs in a loop and continuously gets the best frequencies and finds brief windows to transmit on.

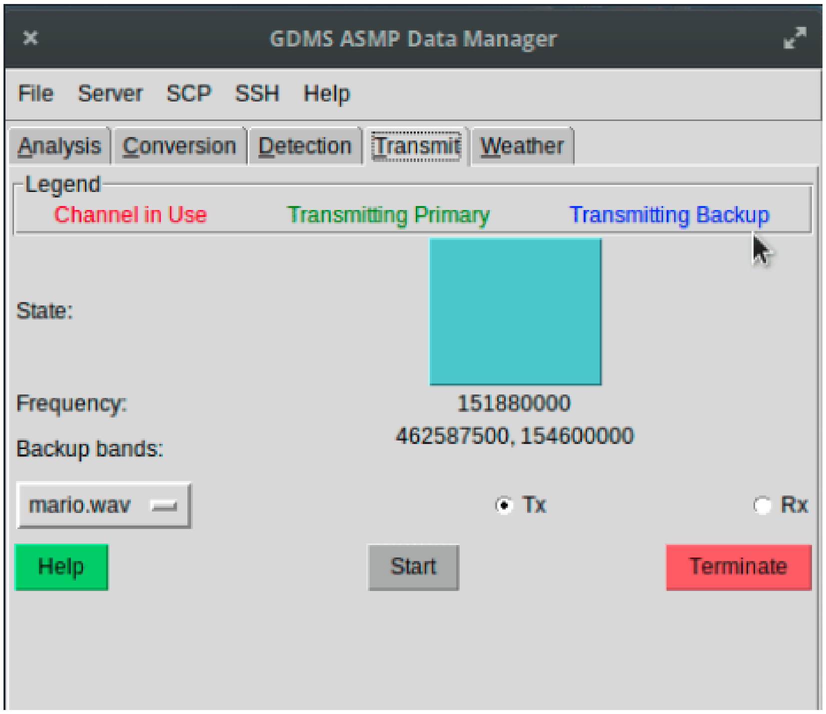


Figure 2.49: Screenshot of GUI during transmission stage. The primary band is occupied in this sample so the transmission is occurring on one of the backup frequencies.

The connecting piece to all of the software is the graphical user interface. Written using the TKinter library this module serves as a central hub for the scripts described above. Using the GUI a user is able to complete the following: perform signal detection and display information of desired bands, transmit and view information on the three optimal bands, convert collected binary files into csv files using weather data, run analysis on collected data, view resulting graphs from analysis, file management including uploads and downloads, interfacing with the Raspberry Pi using SSH to initiate collection, export config files to collection unit, and import data files from collection unit. Using a GUI to perform these actions reduces the overhead in running such a system and simplifies the complexity of the software module for the project.

While the group was able to meet nearly all of the software requirements, there still remains open work. The only incomplete requirement for software is the following: GUI allows user to start and stop recording data. The group did not meet this requirement as a focus was placed on avoiding interaction with the collection unit as much as possible. To preserve storage space and improve battery life it was decided that the Raspberry Pi should operate without a desktop environment. Due to the long periods of collection and speed of collection the team decided that displaying this information is not critical to the performance of the system. In terms of the requirement the group is able to initiate collection through SSH in the GUI but it does not get updates nor control over the process. This control was forgone due to the way that SSH commands are handled. Running scripts through SSH is tied to that terminal session and therefore terminates upon closure of SSH. For this reason the collection is started through a shell script and is placed in a tmux[8] session that can be restored later. The group assumed that a user would likely not have the GUI nor SSH session open for the lengths of time that collection entails.

In the future, other groups can make improvements to the thresholding method used to determine if a signal on a channel. Currently, the thresholding method being used is not able to identify any form of direct sequence spread spectrum due to how the thresholding uses the noise floor as a baseline to compare everything else to. If another group was able to find better ways to detect if a signal is on a channel by means of artificial intelligence or other methods, it would greatly increase the usability of this project in the real world.

Hardware

Based on the requirements laid out at the onset of the project, the team was only able to meet 15% of functional requirements and 80% of nonfunctional. These missing requirements are mainly power requirements such as a battery life of a week for the mapping unit, transmission range, and UAV range. The mapping unit battery life was unrealistic due to the increased consumption rate of the microcontroller cpu and fan during sampling. Also the UAV range ideal value exceeds the FAA's current regulations on line of sight piloting. The limit value can easily be reached with the transmitter and receiver of the drone. The drone specifications towards lift have been theoretically met with the rated motors which are all factory tested for their lift ratings. Combined with the propellers, the chosen 3kg payload should be no problem. A picture of the completed drone is included in picture 2.50.

Figure: 2.50: Completely assembled drone

With regard to the non-functional requirements of the radio hardware, 80% of the goals were met at least to their limit specifications. Only maximum bandwidth (2MHz) did not meet the limit specification (6MHz), due to the max sampling rates of the hardware available. It was not feasible to purchase SDRs with higher sampling rates, as they were generally larger and would struggle to run on a Raspberry Pi. The prototype exceeded the limit value for sample rate by 2x (2Hz sample rate per frame compared to the minimum of 1Hz). The minimum and maximum frequencies supported by the RTL-SDRs met the set minimum (24MHz) and maximum (1.75GHz). The RTL-SDR’s ADC meets the 8-bit minimum requirement.

The largest final specification drawback for the mechanical and electrical systems is the lack of sub-system interface. This interface was dropped due to it being deemed not in the scope of the project. This was because of the custom engineering that would be required to develop a relay system with a 5 volt input and 14v 240 amp output. Otherwise the delivery unit was developed to the specification planned. A change to the final specification was to use a similar capacity but higher discharge rate and voltage battery to provide more power for lifting heavier payloads.

The enclosure was also not weatherproof in the final design. This was due to a lack of time and hardware for weatherproofing the seams between the acrylic sheets. Although the glue should keep most moisture out the enclosure is not 100% weatherproof. Another large drawback for weather proofing is the hole for the antenna cable. This hole was to be covered with a weatherproof cord-grip, however due to difficulty of customizing the antenna wire, this was forgone. Future editions of this project can easily weatherproof the enclosure by using epoxy on all surfaces and a larger cord grip. An image of the final enclosure is shown in figure 2.51.

Figure: 2.51: Final Enclosure

The team's decision to fabricate the external enclosure via laser cut acrylic was the result of a lack of manufacturing capability to fuse the aluminum sheets together. Despite several attempts to weld a working aluminum enclosure, the team ultimately decided that the acrylic shell would suffice. The team decided that the enclosure's ideal mounting location on the UAV was atop the main base plate. This was due to a need to maintain system portability, as mounting on the bottom wouldn't allow for complete UAV landing gear foldability. Additionally, the enclosure would need to be more securely mounted as there would be no plate below it to keep it from falling to the ground. This would complicate the mounting design, but is easily attainable in future modifications. There were also issues in balancing the enclosure so that an equal distribution of weight was present throughout the design. Mounting the enclosure on the bottom may mitigate the issue, but tests should be run in order to locate the center of gravity and properly place the enclosure.

Section - 3: Entrepreneurship and Business Development

Business Objectives and Risks

The scope of this project is to construct a proof-of-concept device that will be able to be transported into a dynamic environment via UAV. Once the device has been delivered, it will begin to take samples of different frequencies that have been pre-selected and gather data on parameters such as received signal strength, time of day, location, and weather. Once a large amount of data has been recorded, the device will use software to determine the ambient noise levels at the individual frequencies, and from these observations the system will be able to recognize if a signal is being transmitted at that frequency.

Competitive Intelligence: Market Analysis

A general theme between many of the companies that the group researched was the need to increase bandwidth while keeping costs down. Being able to tap into the unused space in unlicensed bands would allow customers to use more frequencies while not having to purchase cost prohibitive frequency licenses. Other common requests from companies was to ensure that fast and secure communications were available for the communication between mapping units, and that the new technology would be easily integrated into existing technology.

Financial Analysis

Our group is looking to sell 30% of our company for $5,550,000. This amount was found after a careful financial analysis of this area of research. After averaging the price to earning multiples of other competing companies researching similar technologies (BAE, Raytheon, and AT&T), we found that our company's price to earning multiple will be around 20. Each mapping unit will be sold at a 70% markup at $5,040 meaning the company will make a profit of $2,075 for each unit. Additionally, many consumers will purchase the subscription plan for a higher throughput, and ease of operation. The subscription plan will cost $150/month. Assuming that the company sells 15 units and 10 subscriptions a month, the company will be making of

$500,000 its first year, $875,000 in the second year, and $925,000 in our third year By multiplying our first year profit by the price to earning multiple, our companies valuation is found to be about $18,500,000. Once we obtain our capital, the majority of the money will go toward acquiring facilities for our office, and paying our workers salaries. The manufacturing will be handled overseas.

Innovation Expo (Pitch Presentations, Innovation Showcase Demonstrations)

The link to the team's video is provided below:
https://www.youtube.com/watch?v=TaAquSZwfQ8&feature=youtu.be
The link to the team website is provided below:
https://senior.arunaruljothi.com
The link to the code repository is provided below:
https://bitbucket.org/mosch19/gdms

Appendix A

Team Organizational Chart

The organizational breakdown is presented below in Table A.1 below. The team member backgrounds are presented in figures A.1 and A.2.

Table A.1: Team work breakdown

Electrical/Computer Engineering	Mechanical Engineering
• SDR programming ◦ Mike Moschetti ◦ John Anticev ◦ John Grotke • GUI ◦ Mike Moschetti • Statistical Analysis of data ◦ Arun Aruljothi • Energy requirement analysis ◦ Arun Aruljothi ◦ Greg McNeil ◦ John Anticev	• Drone ◦ Arun Aruljothi ◦ John Grotke ◦ Justin Holdridge ◦ Mike Moschetti ◦ John Anticev ◦ Greg McNeil • Weight analysis ◦ Justin Holdridge ◦ Arun Aruljothi • Enclosure design ◦ Justin Holdridge

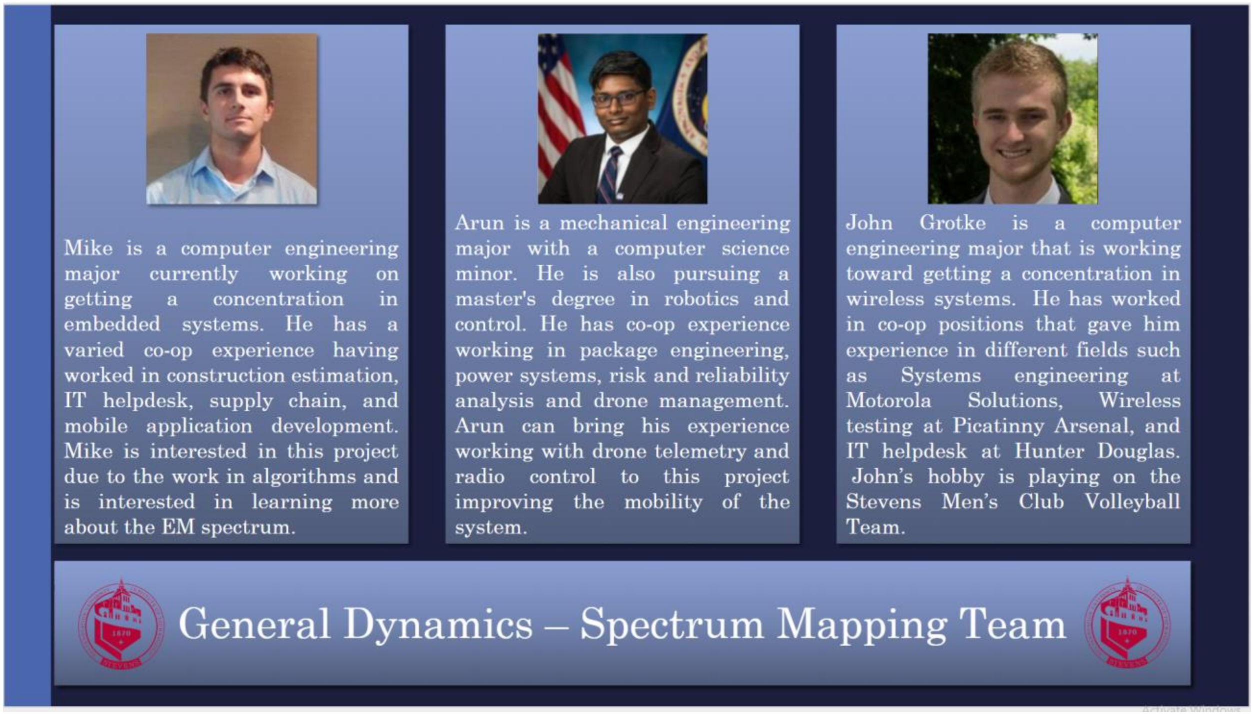


Figure A.2: Team background - Mike Moschetti, Arun Aruljothi, John Grotke

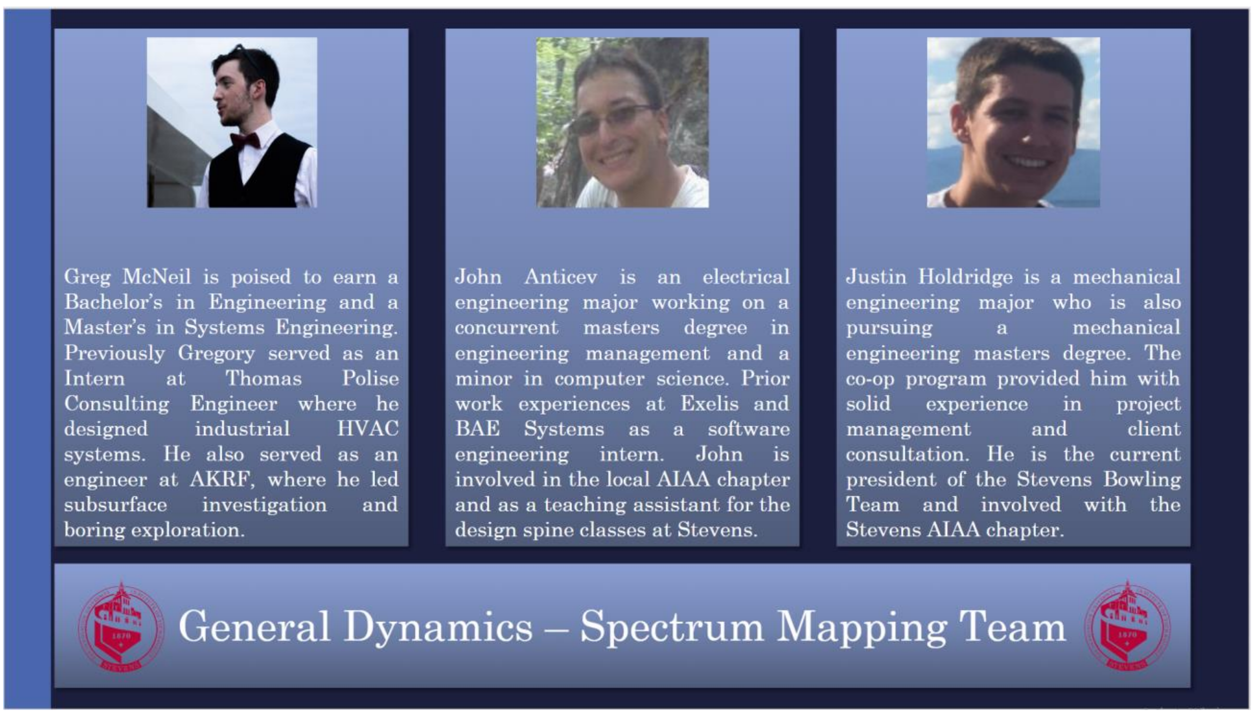
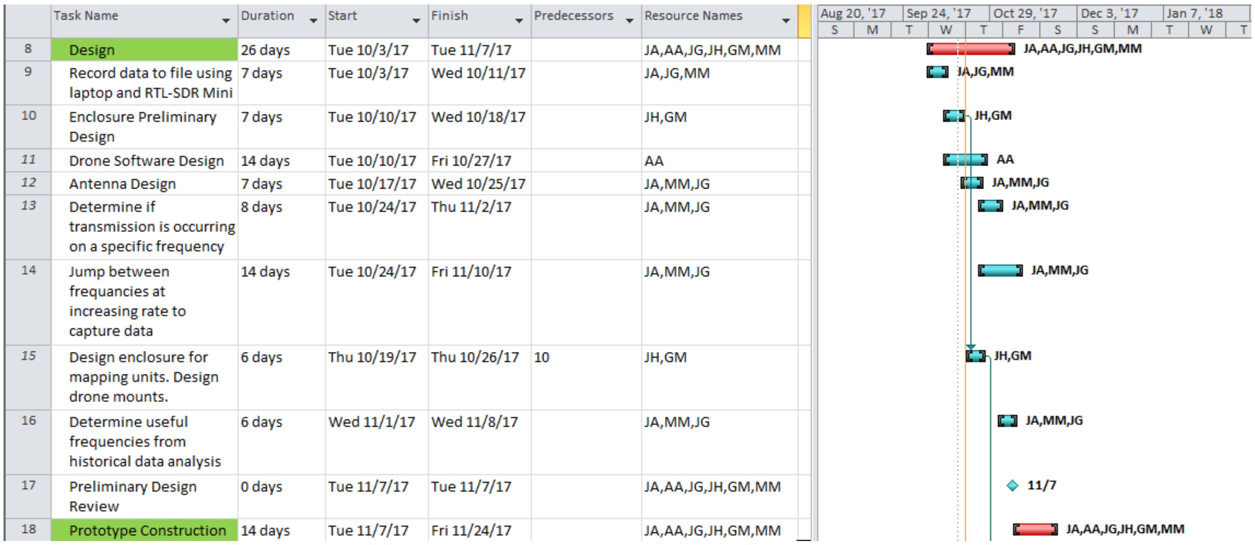


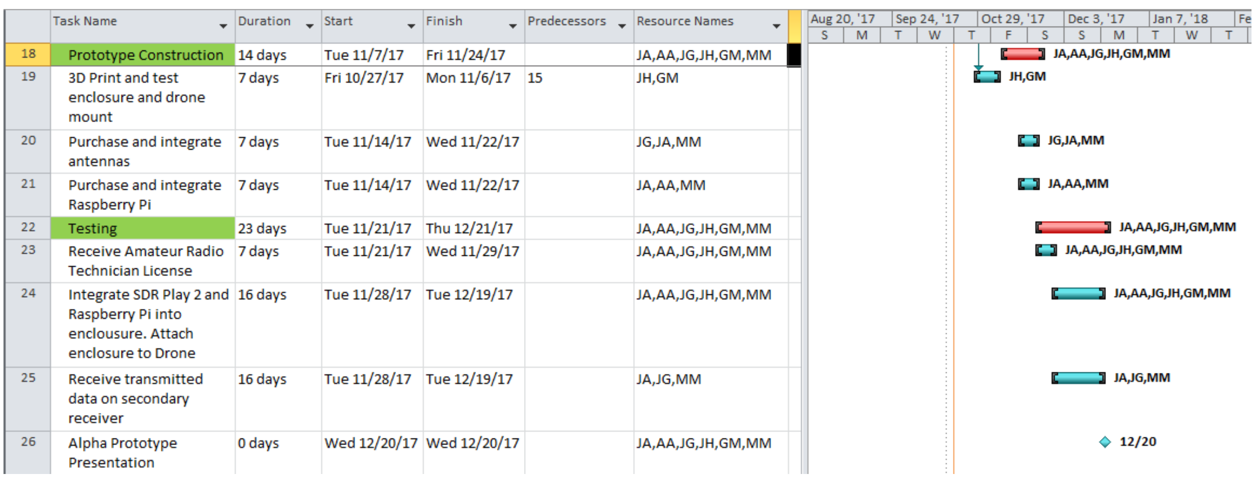
Figure A.3: Team background - Greg McNeil, John Anticev, Justin Holdridge

Appendix B



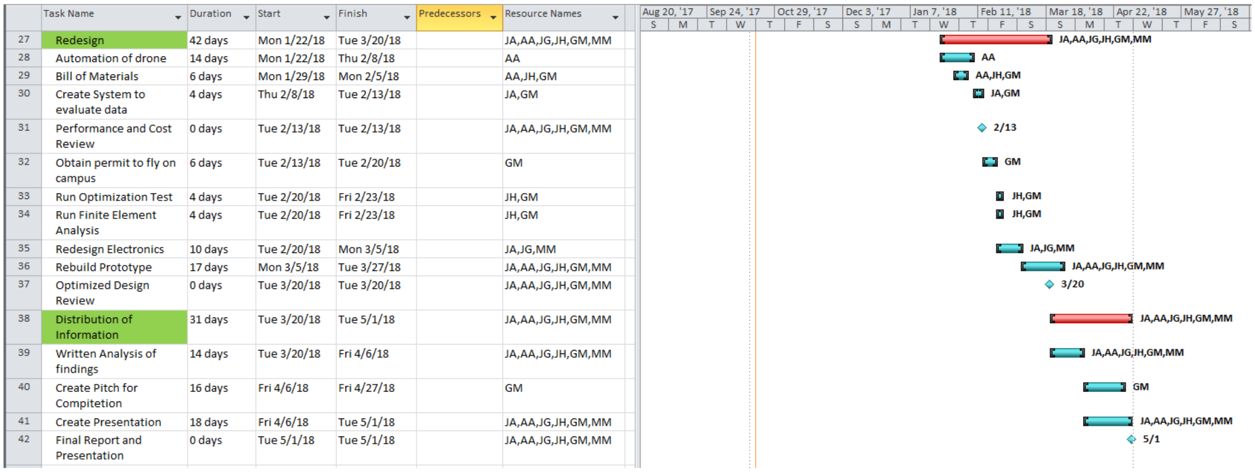
	Task Name	Duration	Start	Finish	Predecessors	Resource Names
8	Design	26 days	Tue 10/3/17	Tue 11/7/17		JA,AA,JG,JH,GM,MM
9	Record data to file using laptop and RTL-SDR Mini	7 days	Tue 10/3/17	Wed 10/11/17		JA,JG,MM
10	Enclosure Preliminary Design	7 days	Tue 10/10/17	Wed 10/18/17		JH,GM
11	Drone Software Design	14 days	Tue 10/10/17	Fri 10/27/17		AA
12	Antenna Design	7 days	Tue 10/17/17	Wed 10/25/17		JA,MM,JG
13	Determine if transmission is occurring on a specific frequency	8 days	Tue 10/24/17	Thu 11/2/17		JA,MM,JG
14	Jump between frequancies at increasing rate to capture data	14 days	Tue 10/24/17	Fri 11/10/17		JA,MM,JG
15	Design enclosure for mapping units. Design drone mounts.	6 days	Thu 10/19/17	Thu 10/26/17	10	JH,GM
16	Determine useful frequencies from historical data analysis	6 days	Wed 11/1/17	Wed 11/8/17		JA,MM,JG
17	Preliminary Design Review	0 days	Tue 11/7/17	Tue 11/7/17		JA,AA,JG,JH,GM,MM
18	Prototype Construction	14 days	Tue 11/7/17	Fri 11/24/17		JA,AA,JG,JH,GM,MM

Figure B.1: The figure above displays a gantt chart that shows important milestones for the first half of the first semester. The left section of the figure lists the important milestones and their deadlines, while the information is displayed graphically on the right.



	Task Name	Duration	Start	Finish	Predecessors	Resource Names
18	Prototype Construction	14 days	Tue 11/7/17	Fri 11/24/17		JA,AA,JG,JH,GM,MM
19	3D Print and test enclosure and drone mount	7 days	Fri 10/27/17	Mon 11/6/17	15	JH,GM
20	Purchase and integrate antennas	7 days	Tue 11/14/17	Wed 11/22/17		JG,JA,MM
21	Purchase and integrate Raspberry Pi	7 days	Tue 11/14/17	Wed 11/22/17		JA,AA,MM
22	Testing	23 days	Tue 11/21/17	Thu 12/21/17		JA,AA,JG,JH,GM,MM
23	Receive Amateur Radio Technician License	7 days	Tue 11/21/17	Wed 11/29/17		JA,AA,JG,JH,GM,MM
24	Integrate SDR Play 2 and Raspberry Pi into enclousure. Attach enclosure to Drone	16 days	Tue 11/28/17	Tue 12/19/17		JA,AA,JG,JH,GM,MM
25	Receive transmitted data on secondary receiver	16 days	Tue 11/28/17	Tue 12/19/17		JA,JG,MM
26	Alpha Prototype Presentation	0 days	Wed 12/20/17	Wed 12/20/17		JA,AA,JG,JH,GM,MM

Figure B.2: The figure above displays the gantt chart for the second half of the first semester.



	Task Name	Duration	Start	Finish	Predecessors	Resource Names
27	Redesign	42 days	Mon 1/22/18	Tue 3/20/18		JA,AA,JG,JH,GM,MM
28	Automation of drone	14 days	Mon 1/22/18	Thu 2/8/18		AA
29	Bill of Materials	6 days	Mon 1/29/18	Mon 2/5/18		AA,JH,GM
30	Create System to evaluate data	4 days	Thu 2/8/18	Tue 2/13/18		JA,GM
31	Performance and Cost Review	0 days	Tue 2/13/18	Tue 2/13/18		JA,AA,JG,JH,GM,MM
32	Obtain permit to fly on campus	6 days	Tue 2/13/18	Tue 2/20/18		GM
33	Run Optimization Test	4 days	Tue 2/20/18	Fri 2/23/18		JH,GM
34	Run Finite Element Analysis	4 days	Tue 2/20/18	Fri 2/23/18		JH,GM
35	Redesign Electronics	10 days	Tue 2/20/18	Mon 3/5/18		JA,JG,MM
36	Rebuild Prototype	17 days	Mon 3/5/18	Tue 3/27/18		JA,AA,JG,JH,GM,MM
37	Optimized Design Review	0 days	Tue 3/20/18	Tue 3/20/18		JA,AA,JG,JH,GM,MM
38	Distribution of Information	31 days	Tue 3/20/18	Tue 5/1/18		JA,AA,JG,JH,GM,MM
39	Written Analysis of findings	14 days	Tue 3/20/18	Fri 4/6/18		JA,AA,JG,JH,GM,MM
40	Create Pitch for Compitition	16 days	Fri 4/6/18	Fri 4/27/18		GM
41	Create Presentation	18 days	Fri 4/6/18	Tue 5/1/18		JA,AA,JG,JH,GM,MM
42	Final Report and Presentation	0 days	Tue 5/1/18	Tue 5/1/18		JA,AA,JG,JH,GM,MM

Figure B.3: The figure above displays a gantt chart that displays important milestone information for the second semester.

Appendix C

Alpha Prototyping Budget

The total alpha prototyping budget came out to under $3,142.70. The budget for all parts used in the alpha prototype is presented in the picture below. Items in green have been ordered and received, while items in yellow are on their way. As you can see in figure C.1 below, all items are green meaning everything has been received. The budget also breaks the parts into their respective sub assemblies, either the drone or the enclosure. The most expensive item were the SDRs used to transmit the signals from the computer. As of the writing of this report, the team has spent about 63% of their budget.

BOM Number	Sub Assembly	Model Number	Model/Maker Name	Part Name	Quantity	Unit of Measurement	Cost (single)	Cost Total	Notes
1	Drone	775	KDE	Brushless Motor	6	Each	$71.95	$431.70	
2	Drone	1260	MREL	Propellor	6	Each	$13.99	$83.94	
3	Drone	Battery	Multi Star 5200mAh	Battery	4	Each	$35.21	$140.84	
4	Drone	GFY690S	Tarot	Chasis	1	Each	$119.00	$119.00	
5	Drone	Turnigy	Plush	Electrionic Speed Controller (ESC)	4	Each	$35.00	$140.00	Only ordered 2 (sold out)
6	Drone	Naza M V2	DJI	Flight Controller	1	Each	$159.00	$159.00	
7	Drone	Frsky	ACCST	Transmitter	1	Each	$132.00	$132.00	
8	Drone	Frsky	X4RSB	Receiver	1	Each	$27.96	$27.96	
8	Drone	B6AC	Genuine SKYRC	Charger	1	Each	$62.99	$62.99	
9	Drone	RioRand	C32312	Battery Measurement	1	Each	$4.98	$4.98	
10	Drone	Firware	XT60	Power Plugs	1	Each	$8.99	$8.99	
11	Drone	CAN	DJI	Flight Controller Accessory	1	Each	$58.00	$58.00	
12	Drone	CP.NZ.000017	DJI	Bluetooth Module	1	Each	$48.60	$48.60	
13	Enclosure	N/A	N/A	ABS+ Plastic	18.41	Cubic Inches	N/A	$61.39	Dont buy off amazon
14	Enclosure	Model B	Raspberry Pi	Microcontroller	1	Each	$34.70	$34.70	
16	Enclosure	LIon Battery	adafruit	LIon Battery 2500mah	12	Each	$13.46	$161.52	
17	Enclosure	One	HackRF	SDR	2	Each	$317.95	$635.90	
18	Enclosure	ANT500	HackRF	Antenna	1	Each	$0.00	$0.00	Comes with SDR
19	Enclosure	SDXC U3	Amplim	SD Card	1	Each	$40.92	$40.92	
20	Enclosure	Startech	USB 2.0	USB Cables	2	Each	$2.79	$5.58	
21	Enclosure	6061	RMP	Aluminum	2	Each	$17.31	$34.62	
22	ALL	TR88302	TR	zipties	1	Each	$5.77	$5.77	
23	ALL	N/A	Tarvol	electric tape	1	Each	$8.99	$8.99	
24	Enclosure	Charger	Adafruit	micro lipo charger	6	Each	$5.95	$35.70	
25	Enclosure	Lipo SHIM	Adafruit	lipo power supply	1	Each	$9.95	$9.95	
26	Enclosure	JST Female	Adafruit	jst female for custom boards	12	Each	$1.50	$18.00	
27	Enclouse	LIon Battery	Adafruit	Lion Battery 2500mah	2	Each	$13.46	$26.92	
28	Drone	Turnigy	Plush	Electrionic Speed Controller (ESC)	4	Each	$35.00	$140.00	
30	Enclosure	BR & TD	Sandisk	SD Card	1	Each	$22.99	$22.99	
31	Enclosure	RTL-SDR	RTL-SDR Blog	SDR + Antenna	1	Each	$25.95	25.95	
32	Testing	UV-5RE	KDE	Radio	3	Each	$16.00	$48.00	
33	Drone	Turnigy	Plush	Electrionic Speed Controller (ESC)	6	Each	$40.00	$240.00	
34	Drone	KDEFX	BAOFENG	Motor Accessory	3	Each	$25.00	$75.00	
35	Drone	6S 50C XT90	YoWoo	Battery	1	Each	$80.00	$80.00	
36	Enclosure	Model B	iUniker	Heat Sink	1	Each	$12.86	$12.86	
37	Enclosure			Aluminum Plate	2			$30.00	
	compact plastic	mcmaster carr					Total	$3,142.76	
							Percent of Budget used	62.86%	

Figure C.1: Final budget for testing and prototyping

Appendix D

D1: Necessary Python Modules

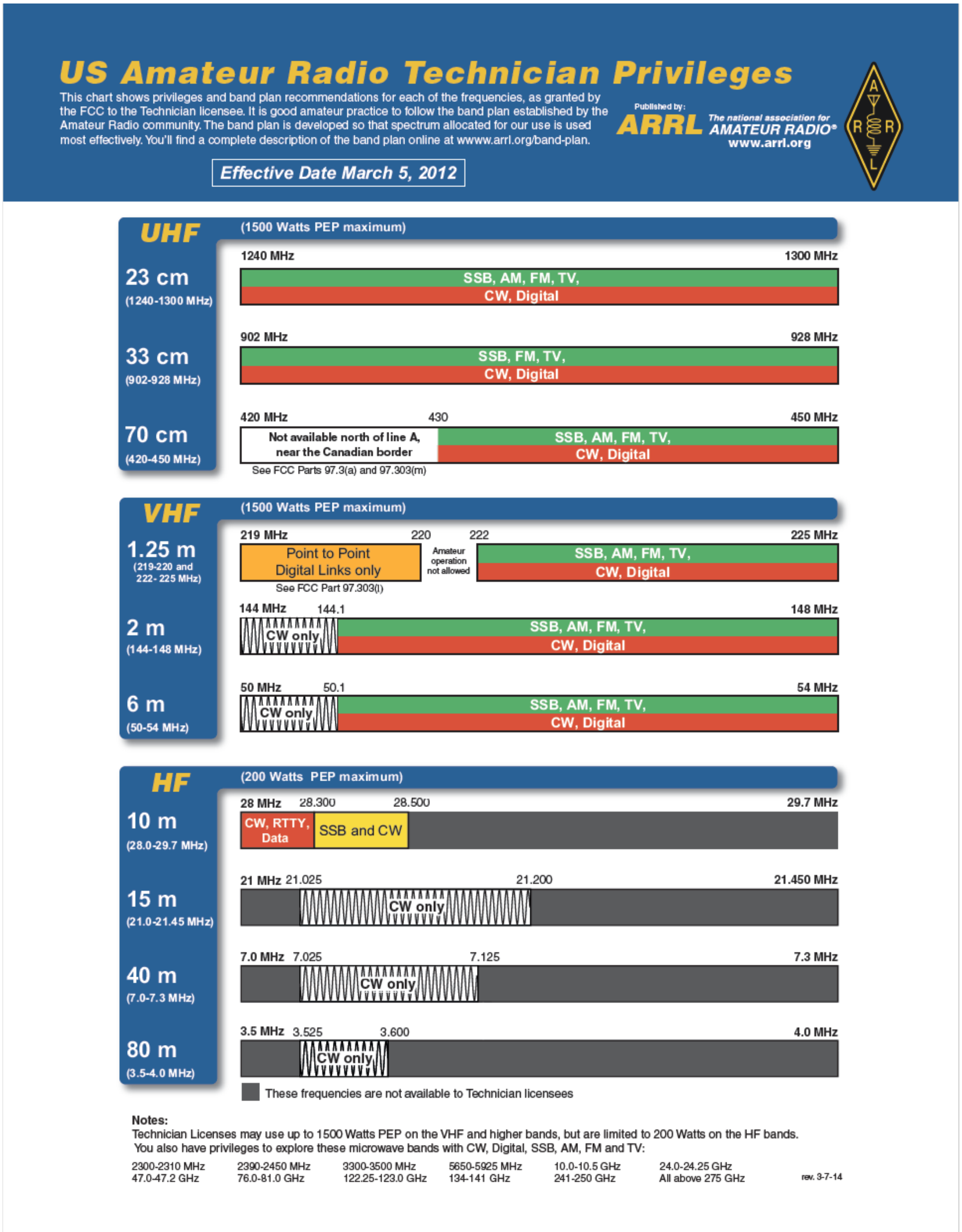
- Argparse
- Collections
- Cv2
- Datetime
- Getopt
- GNU Radio
- JSON
- Math
- Numpy
- Operator
- Optparse
- Os
- Osmosdr
- Pandas
- Paramiko
- Python Imaging Library
- Pickle
- Pprint
- Pysftp
- Scikit
- Shutil
- Struct
- Subprocess
- Sys
- Threading
- Time
- TKinter
- Ttk
- Urllib2
- Queue
- Webbrows

D2: Weather Condition Codes

fctcode	Forecast
1	Clear
2	Partly Cloudy
3	Mostly Cloudy
4	Cloudy
5	Hazy
6	Foggy
7	Very Hot
8	Very Cold
9	Blowing Snow

10	Chance of Showers
11	Showers
12	Chance of Rain
13	Rain
14	Chance of a Thunderstorm
15	Thunderstorm
16	Flurries
17	OMITTED
18	Chance of Snow Showers
19	Snow Showers
20	Chance of Snow
21	Snow
22	Chace of Ice Pellets
23	Ice Pellets
24	Blizzard

D3: US Amateur Radio Technician Privileges from the ARRL[2]



Works Cited

1. Basic date and time types. Python Foundation. docs.python.org/2/library/datetime.html#
2. Chen, Brian X. "Companies Try to Create Room on Radio Spectrum." *The New York Times*, The New York Times, 6 June 2012, www.nytimes.com/2012/06/07/technology/as-wireless-spectrum-is-squeezed-sharing-is-seen-as-solution.html?mcubz=0.
3. Mathematical functions. Python Foundation. docs.python.org/2/library/math.html
4. The GNU Radio Foundation. www.gnuradio.org
5. "Updated ARRL "US Amateur Radio Bands" Charts Now Available." *www.arrl.org*. ARRL, 16 May 2016. Web.
6. wxWidgets. www.wxwidgets.org
7. Staples, Gabriel. "RC, Arduino, Programming, & Electronics." ElectricRCAircraftGuy.com--RC, Arduino, Programming, & Electronics, 1 Jan. 1970, www.electricrcaircraftguy.com/.
8. "Tmux website". https://github.com/tmux/tmux/wiki
9. "Weather Underground API". The Weather Company. https://www.wunderground.com/weather/api/