Autonomous Search And Rescue Based On Cellular Beacon

Bjorn W. Galaske, EE; Jamie L. Kline, EE; Bradley M. Marszalkowski, EE; Serena L. Thomas, EE

Abstract—**Search and Rescue (SAR) in remote locations proves to be a difficult, and incredibly expensive task. Infrared detection with aircraft can be ineffective with dense tree cover, snow, and ice. Partially due to this terrain, the Canadian Armed Forces (CAF) coordinates roughly 10,000 rescues every year[1]. A single SAR mission can cost $1700/hr for a helicopter, and $7600/hr if a C-130 (Figure 1) is needed for remote/rugged terrain[2]. For under $500, SAR teams can have a single drone that will search a predefined remote area for cell phones that are powered on. The Search And Find Emergency Drone (SAFE Drone) records the signal strength and the GPS location. Upon return, that data is downloaded and converted to a signal-strength heat-map. The "hotter" portions of the map indicate the most likely location of the missing individual(s). The SAFE Drone system will assist SAR teams to make a more efficient, and cost effective search.**

I. Introduction

SAR teams continually deploy their efforts in part because of outdoor recreation, winter sports, and many other high risk activities. These activities include, but are not limited to hiking, mountain climbing/rappelling, rafting, snowboarding, skiing, hang gliding, hunting, camping, and even flying personal aircraft[3]. Most individuals that are capable of doing such high risk activities are young, and carry smartphones for pictures. Additionally, any cell phone in a remote area (no cell service) will continually attempt to connect with the closest cell tower. So, incase of an emergency due to any high risk activity, the individual essentially has an emergency beacon in their pocket. The cell phone will generally transmit with a maximum power of 2 watts[4] when searching for a tower. All of this is assuming, of course, that the cell phone is powered on. With this knowledge, there exists the opportunity to create an elegant engineering solution using a drone that searches for a cellular beacon in remote areas.

Current SAR drones use visible, and infrared imaging that provide rescuers with a visual means of searching remotely. These methods work great with most terrain. However, there are still remote areas that significantly inhibit these SAR efforts. For those remote areas, military grade aerial imaging such as the Forward Looking Infrared (FLIR) system[5] is necessary. However, this can still prove ineffective with dense tree cover.

Fig. 1: Special Operations MC-130H equipped with FLIR[6]

Anyone that has been camping knows that it is hard to make a good phone call in a forest, but if there is a nearby tower, you can actually have enough service to create a connection. This proves that it is feasible for the SAFE drone to detect a wireless cell phone through dense tree cover.

The SAFE Drone system has many other applications with great room for scalability. For instance, SAR teams could invest in a fleet of SAFE Drones to cover a vast area in a very short amount of time. Our drone is capable of travelling up to 30 miles per hour for 10 minutes, meaning it can traverse 5 linear miles. Depending on the sensitivity of the receiving system (due in part by seasonal tree foliage, sample rate, and height above the tree canopy), one single drone could cover a theoretical maximum of one square mile. This means ten drones could cover ten square miles. Further, the drone could recognize different frequencies in order to narrow down the search for a known frequency band. There can also be a phone application that will transmit an emergency signal to include GPS coordinates, and perhaps cell phone battery life. This emergency signal could be initiated with a quick touch of the screen. The drone would then relay the signal back to SAR teams to reveal a very quick response. With all of this scalability the drone could also transmit the gathered data wirelessly to get a real-time feed of the received signal(s). In short there are numerous possibilities for scaling this platform.

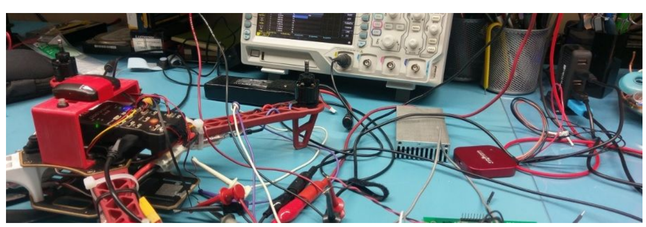


Fig. 2: SAFE Drone Prototype

II. Design

A. Overview

In order to find a cellular device the SAFE drone is first given a flight plan that covers a general area that the rescuer believes the cellular device is located in. It then samples signal intensity levels at a sample rate determined by the speed of the drone and the resolution desired. It then saves this data to the onboard memory. Once the drone completes the flight plan, it returns to the original launch point and makes both the signal intensity, and the GPS data available for review. The users download this data file onto a host pc and run it through the mapping software which then creates a weighted point temperature mapping .kml file ready to be overlaid onto a satellite map, thus allowing the users to pinpoint the location of the strongest cellular signals and determine the location of the cellular device. To do this the drone uses a flight controller, GPS module, supervisory microcontroller, power detector circuit, and host pc software together.

The flight controller performs all of the flight functions required as well as forwards GPS coordinates to the supervisory microcontroller. The power detector circuit converts the dBm power intensity of the incoming signal into an analog voltage which is fed into the microcontroller via the on-chip Analog to Digital Converter (ADC). The microcontroller samples both the GPS coordinates and the signal intensities and maps them to each other before saving the data to onboard EEPROM. When the drone returns to base, the microcontroller makes the data available via usb and the host pc creates the visual representation. Reference the block diagram in Appendix A for a full system flow diagram.

The first edition of this project detected a cellular signal and then would have attempted to decode the signal, retrieving the International Mobile Subscriber Identity (IMSI) from it in order to identify the individual user, and based on the receiving capability, a range assumption was made. However, this was deemed unfeasible due to time and budgetary constraints.

Specifications	Value
Max Flight Time	10 Minutes
Max Speed	30 MPH
Max Sample Speed	200 KSPS
Max Samples	5,300
Coordinate Accuracy	<1ft
Input Signal intensity Range	-70 dBm to 5 dBm
Frequency Range	835 MHz - 915 MHz
Assumed Distance to Freq. Source	<100ft

Table 1: System Specifications

B. Flight Hardware

To facilitate the areal scanning, a quad-rotor multi copter will be utilized. The frame, motors, motor speed controllers, and GPS module are reused from the ECE 2015 Senior Design Project (SDP), however a new flight controller was purchased. The 3D Robotics (3DR) PixHawk flight controllers offers a more powerful 32-bit processor and features a redundant processor for failsafe computing fallback. This open-source hardware runs Nuttx, a real-time operating system which enables predictable response times for critical flight functions. The flight controller is responsible for all of the motor control and dynamics of flight. Onboard magnetometers and gyroscopes provide stable flight with no user intervention after proper calibration. Connected to the flight controller is an external 2.4GHz spread-spectrum radio transceiver that provides both manual override control from a hand-held remote, as well as transmission of telemetry data to the ground for in-flight diagnostics. Most importantly an external GPS module allows the flight controller to navigate a predefined flight path. This path is setup on the host PC prior to flight, downloaded to the flight controller, and consists of a series of raster navigational waypoints that the multi rotor copter executes autonomously. A return-to-launch (RTL) feature brings the quad rotor copter to its launch site after completing the mission. Automatic landing is accomplished by using a barometer for relative (to launch site) altitude measurements.

Estimations for vehicle weight were made in order to assure that the flight hardware was capable of adequately lifting the required payload, tabulated in **Table 2**, below. Shown also is the actual weighed weight (to within ±1g).

Item	Weight (g)	
Bare frame, DJI Flamewheel	155	
Motors: Sunny X2212 (x4)	200	
Props (est) 9047 or 1047 (x4)	20	
Electronic Speed Controllers (x4)	128	
APM (PixHawk), Radio	13	
GPS - 3DR uBlok GPS And Compass	9.2	
GPS Mounting bracket	100	
Battery - 11.1V 5200mAh 3S 30C	392	
Cables, connectors (est.)	150	
PCB, parts, antennae (to be determined)	0	
Total (estimated):	**1167**	**g**
Total (actual):	**1197**	**g**

Table 2: Estimated All-up Payload

We anticipate approximately 200g of <u>additional</u> weight being added for the printed circuit board (PCB) and its associated components to include antennae. This brings the estimated all-up flight weight to ~1400g.

Based on manufacturer specifications, each SunnySky X2212 DC brushless motor can provide a range of 870-480 grams of thrust when paired with a 1047 prop (10" diameter, 4.7" pitch) at full throttle. This is achieved over a battery range of 11.1V (nominal, fully charged) to 7.4V (typical discharge level). The below equation is used to calculate total vehicle lift capability based on motor throttle percentage:

$$F_{tot} = (F_{\max\_mot})(\%_{Thr})(4)$$

In order to achieve proper vehicle throttle response it is common practice to target vehicle hover capability at 50% throttle. Hover at a much lower throttle percentage can result in underdamped (twitchy) stability response; hover requiring a much higher throttle can result in overdamped (sluggish) response. However for vehicles that do not require rapid maneuvering as is in our case, one may trade off responsiveness for higher payload capability. Taking all these factors into account the minimum throttle required to maintain flight of a 1400 g craft was calculated over the expected battery operating range. This data is shown in **Table 3**.

Battery (V)	Thrust /mot. (g)	Throttle (% min.)
11.1	870	40.2%
10.0	720	48.6%
8.0	520	67.3%
7.4	480	72.9%

Table 3: Required Throttle Over Battery Range

Numerous test flights were made including several fully autonomous missions, and the platform performed as expected.

Up until now we have only described getting the craft in the air and autonomously scanning a pre-defined area. However, we also need to take advantage of the sensor data (namely GPS) available from the flight controller for use in our mapping system. The flight controller features numerous universal asynchronous receiver-transmitter (UART) ports which implement a communication protocol called MAVLink (Micro Air Vehicle Link). This standardized protocol allows an external companion controller to read raw sensor data, modify navigation waypoints, change the current flight mode, override motor commands, etc.

TESTING:

In order to test the MAVLink protocol features a surrogate Atmel ATMega16 microcontroller was prototyped and programmed using custom C firmware and the MAVLink header libraries. The first step is detecting the "heartbeat" of the flight controller: this is a MAVLink message sent over the UART every second identifying the device (flight controller) node ID and its current status. On start-up, the ATMega was programmed to continuously check for bytes in the UART receive buffer, and proceeds to 'build' or assemble the packet sent from the flight controller. Once a full packet is successfully received, the message can be parsed depending on the message ID. Upon full receipt of each heartbeat message, we toggled an LED indicator providing visual feedback that the communication link was active. This will continue to be used throughout the project as a diagnostic indicator.

Once the UART signaling and protocol information was verified to be correct the next step is to request raw sensor (such as GPS, gyroscope, etc) data. In order to do so a request must be sent from the ATMega to the flight controller, requesting it to start a defined datastream. The flight controller in turn then periodically sends out data pertaining to the datastream type requested. On the ATMega side, once a full packet is detected and the message ID matched to a known stream ID, the ATMega software is then able to parse out the individual sensor data and use as needed. Using a 4x20 character LCD wired to the ATMega, we succeeded in displaying the requested GPS sensor data. These data were verified using a cellular phone's GPS sensor as reference (**Figure 3**).

At some point we would like to include an onboard energy monitoring feature, used to more accurately predict the amount of energy remaining in the battery, and more importantly, how much energy is required to return to launch (based on previous consumption data and current GPS coordinates). In order to implement this, we needed a way to command the craft to return to the launch site once it reaches a critical energy level. The flight controller can be setup to support multiple *flight modes*, which are groups of settings and parameters particular to a desired mode of operation. For example *Stabilize* mode allows the user to easily command the craft in a particular direction without having to worry about keeping the craft stable. *Loiter* mode utilizes the GPS sensor to maintain a fixed position in space when no control inputs are otherwise given. In our case we wanted the ability to have the ATMega switch the flight controller to an *RTL* or return-to-launch mode.

The setup for this test was similar, however to verify the correct switching of the modes, we also connected the flight controller via USB to a ground station software suite called Mission Planner, available open source. This software allows you to define the autonomous flight paths, set flight controller parameters, and also view various status indicators including the current flight mode. The ATMega was programmed to monitor the flight controller for its heartbeat signal, and after receiving three successive heartbeats, begin a countdown timer (also displayed on the ATMega LCD screen). After a few second countdown, the ATMega sent a MAVLink packet instructing the flight controller to change modes, and would continue to make this request until it verified (from the status received in the heartbeat message) that the mode was indeed changed. This mode change was successfully reflected on both the LCD and the ground control software.

Moving forward the MAVLink software routines will need to be ported to our target microprocessor platform, described in more detail in section *F, Supervisory Microcontroller*. While the development environment will remain the same (Atmel

Studio), the new processor is completely different than the 8-bit core Mega processors taught at UMASS, and so this will require learning and understanding an entirely new architecture. The logic of the MAVLink packet interpretation will also large remain the same, however the plan is to implement an interrupt-driven UART circular buffer. This will ensure that during flight, packet loss from the flight controller is minimized due to receive buffer overruns. In addition, the software routines will need to be more robust and able to recover predictably from transmission or receipt errors. Currently there is only rudimentary message verification (for instance, no bounds checking on GPS coordinates, no fallback for a non-responsive flight controller, etc). The implementation of UARTs and input bounds checking are topics covered in courses at UMASS.

The next stage of testing will involve a flight-ready prototype microprocessor which will pair the received signal strength indicator (described in section *D. Signal Strength Detection*) with the prototype microcontroller in order to test analog sampling while onboard the quadrotor copter. Several challenges exist for this: most notably finding a location without nearby cellular towers. Second we will need to set the transmitting phone in one particular place, and center our pre-defined search pattern around this location. Additionally we will want to make sure the flight height directly above the phone is the same as previous open-air phone / detector tests. What we want to see is similar signal strength levels as said test, without interference from the various high-current circuits onboard the craft. This brings us to the next challenge, which is assuring that our prototype, and eventually deliverable PCB, have proper ground and power routing to assure that any electromagnetic interference from said DC controllers does not affect our signal detection abilities or the stability of our microprocessor system. This will be accomplished by strategic ground plane segregation and stitching, proper star or single-point ground connections, and careful cable routing/management.

The next stage also involves the planning and routing of a first-spin PCB (described in section *F. Project PCB*).

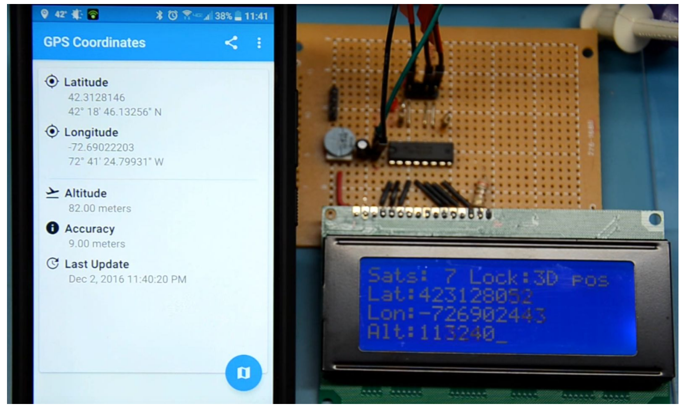


Fig. 3: Test of MAVLink Data Streams

C. *Distance Sensory*

We initially were unsure how accurate the flight controller's autonomous landing sequence would be, so we planned to complement it with either a LIDAR, or an ultrasonic distance detection system. Because there existed the possibility of implementing a terrain following subsystem, we chose the ultrasonic. This would allow for wide beam detection that can be adjusted with a receiving cone.

We chose the MaxBotix, LV, EZ series of sensors, which can be fed directly into the flight controller through an analog, PWM, or serial signal. It detects more than 20 feet of range and with a fabricated cone, much like a horn antenna, we can adjust the width of the detection range. This can help minimize ultrasonic noise, and follow the terrain better, while sacrificing some maximum range. See **Figure 4** to see the results of testing the sensor (without a cone) with both a tree branch, and a 10 inch board.

As you can see from **Figure 4**, the sensor accurately detects a tree branch that closely resembles a "charlie brown christmas tree with a long trunk." These results are very acceptable for terrain following, and we hope to implement the terrain following feature after the main subsystems are finished.

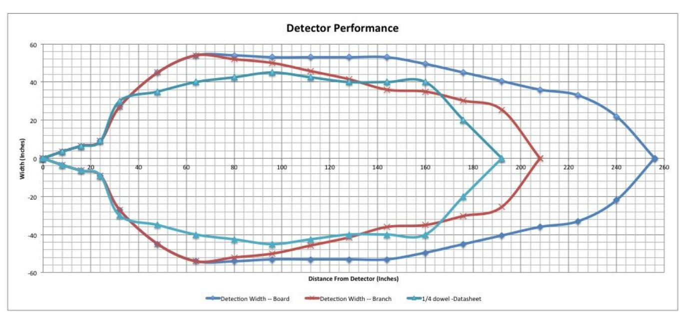


Fig. 4: Ultrasonic Distance and Tree Detection Experiment

D. Signal Strength Detection

For signal strength detection, we used the Maxim Integrated MAX2015 logarithmic detector in received signal strength indication (RSSI) mode. It accepts input signals from 100MHz-3GHz with an input power ranging from -65 dBm to 5 dBm [8]. The circuit used was based on the RSSI mode circuit from the datasheet of the MAX2015 (**Figure 5**).

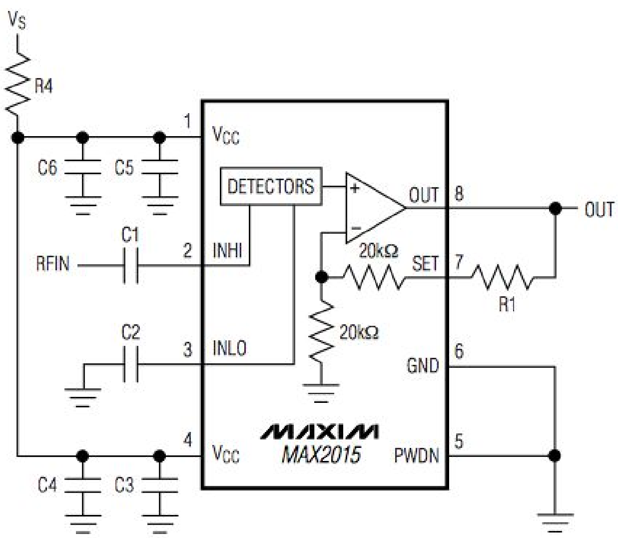


Fig. 5: RSSI Mode Detection circuit [8]

The antenna used was the W1900 penta-band antenna from Pulse Technitrol company. The datasheet for the return loss claims bandwidths that we were not able to duplicate with a vector network analyzer. Further testing is required, but it is currently capable of receiving the signal due to the simple fact that the cell phone transmits with such a high power in remote areas. For a proper design to work with the heat map (explained in block F), we should design the RF input so that the power is at its maximum when the drone is directly above the signal (100 ft away due to the tree canopy).

To accurately design this would mean performing a tree top experiment ourselves because the closest study that was relative to our application was conducted in 1977 titled "Effects of trees and foliage on the propagation of UHF satellite signals." [9]. This study revealed losses in a thick forest to be around 6dB. Unfortunately, this study was performed in the 200-300MHz band. A portion of this study did reveal a plot of attenuation per meter vs. frequency (**Figure 6**) showing 900MHz at about 0.2dB/m.

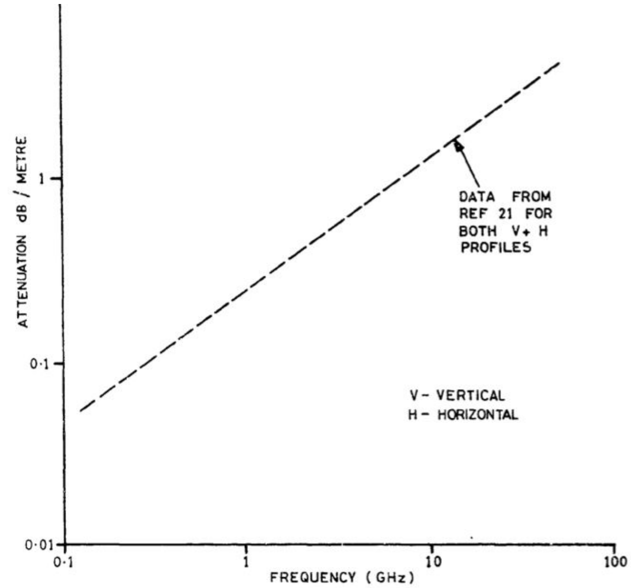


Fig. 6: Foliage Attenuation per meter [8]

Unfortunately, this experiment does not specify what kind of trees, how dense, how many trees, etc. Another study was performed at the 870MHz band but that doesn't involve a forest. This study [10] revealed a piecewise equation for the "slant path" empirical model claiming 26 dB of maximum loss depending on the angle of incidence. Of course all of these results conflict, and they are not the proper experiment for our purposes. A necessary experiment would take UHF signal strength measurements 100 ft away as the control, then take the same measurements at the top of a dense canopy in summer (fully developed leaves for deciduous trees). This experiment should be taken for every type of full grown tree. For the time being, we will make the reasonable assumption that losses from the phone transmission at the forest floor to the drone directly above the canopy to range from 1dB-20dB

For design purposes, we will need the 33dBm signal to be attenuated by 28dB from the cell phone to the input of the MAX2015 detector to prevent the detector from becoming saturated at 5dBm. The foliage attenuation combined with the loss through the PCB will leave the signal very close to what we want. This tuning will determine the error of the heat map. If we saturate the detector before the drone is directly over the cell phone, then the heat map will show a larger radius, therefore a larger margin of error. This will be a challenge to overcome in our final design simply because of changes in foliage attenuation across a forest. However, the forest works in our favor. The drone will be following in close proximity to the canopy. This means there will be an increasing foliage attenuation the further the drone is from the cell phone. This is directly due to the decreasing angle of incidence. The drone approaches the cell phone, the angle is increased, and the foliage attenuation decreases, therefore the signal strength increases which will then result in an accurate signal strength heat map.

Some impedance matching will lower the VSWR to provide a clean signal into the MAX2015, which outputs a nominal 0.5v-1.8v signal over its detection range. Using the graph from the datasheet we can then determine the cell phone's signal maximum from **Figure 7**:

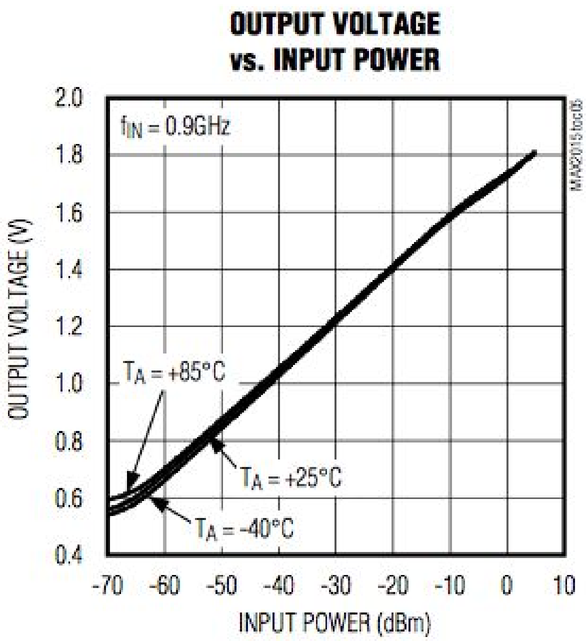


Fig. 7: MAX2015 Voltage Output vs. Input power [7]

In order to audibly test our circuit with various cell phone setups, we need a comparator stage at the output of the signal strength detection circuit. This includes a potentiometer that can adjust the threshold of the comparator and a piezo buzzer. This circuit will buzz whenever it detects a signal power that is higher than the desired noise threshold.

The circuit was tested using various cell phone setups. The first setup was using a cell phone in an area with service. The circuit buzzed when the cell phone made a call, indicating the detection of the phone in constant contact with the tower. The second setup included a cell phone first turning on in an area with service where it sends out a signal to the tower with the IMSI attached and the circuit buzzed once. The last setup included a cell phone in an area with no service. The beeping of the circuit indicated that the phone sent out 10 quick signal bursts to search for a potential tower, then 20 seconds later, the phone tries again to contact the tower with 10 quick signal bursts. This last setup is the one that we will be detecting with our final prototype. It has been tested up to 100' away and the circuit has successfully detected the cell phone. This comparator circuit will be replaced by a gain stage in order to properly scale the MAX2015 output signal to that of the operating range of the microcontroller ADC (described in *F. Supervisory Microcontroller Block*).

E. Manual Control

In order to successfully demonstrate a working prototype safely on campus, there needs to be a fallback available to manually control the drone. This means we first needed to determine whether the remote control for the drone interfered with the same frequency as the cell phone 900 MHz GSM band. Using a spectrum analyzer, our antenna, and our remote control, we find that the signal from the remote control only affects the 2.4GHz frequency and not the cell phone band we are targeting (see **Figure 8**, below).

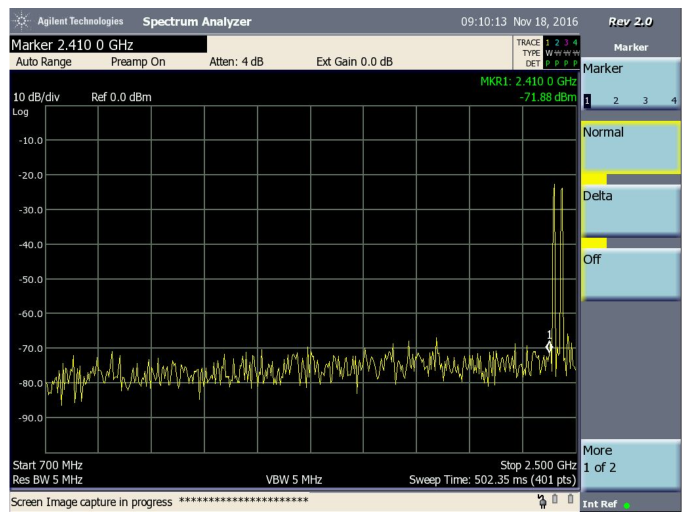


Fig. 8: Spectral analysis of remote control frequency without low-pass filter

Because the signal strength detector circuit will detect any frequency between 100MHz and 3000MHz, we need to filter out the remote control signal. To do this, we are using a low-pass filter. It is an SMA connected low pass filter that is in series with the antenna. To test this, we used the low-pass filter, the spectrum analyzer, our antenna, and the remote control. We found that the low-pass filter successfully filtered out the signal from the remote control.

F. Supervisory Microcontroller

The supervisory microcontroller is used to sample data, create files, and run communications for the device. The Microcontroller being used is the AT32UC3B1256. This device was chosen because of its high internal clock speed, high sample rate for the ADC, physical dimensions, and large number of peripheral communication capabilities. Using the programming skills learned in computer systems lab, we are able to program the device.

The device has an internal clock speed of 20MHZ and a 10 bit resolution ADC that samples at 384KSPS. This ADC gives a voltage resolution of 3mV per bit which translates to roughly 0.1861 dbm per bit. It has an Serial Peripheral Interface bus (SPI) interface which will be used to transfer the data to the onboard EEPROM at a rate determined by the sampling speed. Also, it has multiple UARTs which will be used to read the GPS coordinate data from the MAVLINK. Lastly, it has onboard USB interface support to potentially be used to transfer the data to the host PC.

In order to fully implement this we need to learn more about precise software timing. We want to sample at a specified rate based on the speed of the drone. To do this we need to take into account the clock speed, transmission delays, and buffering times. Once this information is collated we will then take the GPS module, power detector circuit and microcontroller out into a cellular dead spot to collect empirical data on how various sampling speeds will affect the accuracy of the device. In order to analyze the data we will need controlled data points as well in order to compare them to the empirical device data.

All of the data the drone collects is downloaded to the host pc and loaded into the result mapping software. It is used to create a weighted point temperature mapping as shown in **Figure 9**.

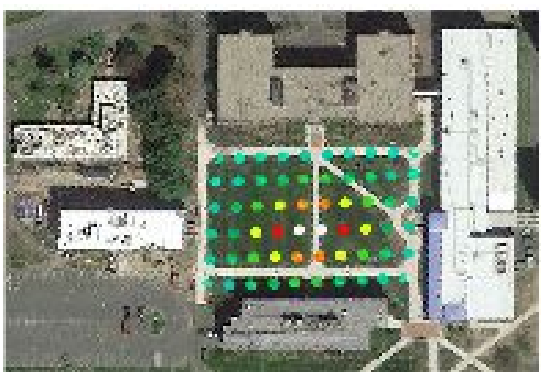


Fig. 9: Example Heat Map

The overlay is temperature based, the higher the voltage level the higher the point density. The temperature points are centered at the GPS coordinate points that the signal intensity is mapped to. In Figure 9 the blue points have the least point density and the white has the most. This translates to the white points having a higher signal intensity and thus identifying where the cellular device is.

The overlay is customizable via the user software. The point size, color scheme, and opacity can all be defined. The color schemes available are shown in **Figure 10**. This helps the accuracy because the backdrop of the overlay could be a matching color of the temperature point. Or, if for example there were two points that had the same intensity. By increasing the dot size, one could ascertain the most likely location of the cellular device by looking at the intersection of each of the dots as their size increased.

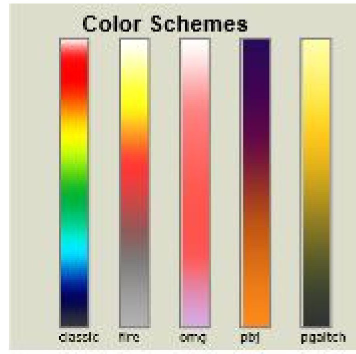


Fig. 10: Available Color Schemes

The higher the signal intensity mapped to a particular GPS coordinate, the further towards the top of the color scheme the point appears. Even with small variances between points it is possible to determine the strongest point because the software averages all of the points together before assigning a color. To create the overlay the software first takes a list of signal intensity, and GPS coordinate tuples as shown in **Figure 11**.

Example Data File

```
155,(42.393960,-72.528880)
175,(42.393930,-72.528962)
250,(42.393900,-72.529044)
318,(42.393870,-72.529126)
400,(42.393840,-72.529208)
558,(42.393810,-72.529290)
400,(42.393780,-72.529372)
318,(42.393750,-72.529454)
250,(42.393720,-72.529536)
175,(42.393690,-72.529618)
150,(42.393615,-72.529576)
```

Fig. 11: Example Data File

It then finds the 4 corner points in the list, the most northwest, northeast, southwest, and southeast points. It uses these in order to create a box that all of the other points are located within. It then averages out all of intensities in order to define the temperate mapping for each signal intensity. Lastly, it uses a python to html software adapter. It then uses a tool in the google maps API called [10]gheat to create the visualization in the form of a .kml file. This .kml file is then simply loaded into google earth and shows up as an overlaid image.

G. Project PCB

Our project circuit board will encompass a wide variety of both analog and digital components, representing many disciplines learned at UMASS (with the exception of circuit board design). This board will feature:

- Onboard 3.3V/5.0V/ADC reference regulators to power board in addition to flight controller and radio
- High current battery monitoring circuitry
- 32 bit microprocessor for fast analysis of received signal strengths and vector calculations
- EEPROM for local storage of sampled data
- UART connection to flight controller
- Wifi or USB UART bridge for connection to host PC
- Modular analog sampling daughterboard
- Integration into existing airframe

As mentioned the PCB will feature a modular sensor design which will allow the RF front end and detection circuit to be changed easily. The scope of this project is limited to the one sensing and detection technique; however by moving this circuitry to a daughterboard we reduce the financial impact risk if modifications to the sensing circuitry are required after testing.

At the time of writing, the overall mechanicals and footprint of the PCB have been established. This particular form factor allows the board to be secured inside the craft, with the battery and power leads conveniently protruding from one end, and allowing open-air exposure to the sensing board at the other. The majority of the effort for the PCB will occur shortly after MDR, as this board will be required to begin testing of the integrated system. The PCB outline is shown below in **Figure 12**.

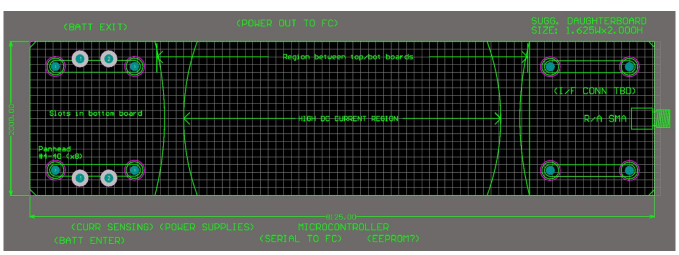


Fig. 12: Draft PCB Mechanical Layout

H. Miscellaneous

There were also a few legal issues that needed to be addressed for our design. The first legal issue was getting the drone registered with the FAA as a UAS (Unmanned Aircraft System). This was done before we were able to fly our drone. The second legal matter is getting a HAM radio technician class license in order to transmit on the 900MHz GSM cell phone band, which we may want to later if when we test the final prototype on demo day as well as add on the IMSI detection functionality if time permits.

We are currently on track and under budget. So far, we have used about $335 of our available budget; $165 remains. We anticipate we will spend an additional $70 on a PCB and about $70 on various other parts.

An additional function we may want to incorporate is a probability calculation. This will reveal the probability that a phone is at any particular point chosen (by SAR team member) on the resulting "heat" map. This calculation will assume a normal distribution of points. These sample points are given as power levels at certain coordinates on the "heat map." There is a higher probability that a person is at a point if there is a higher power level. When graphing all the power levels, they will be at one point in the normal distribution graph given in **Figure 13** below.

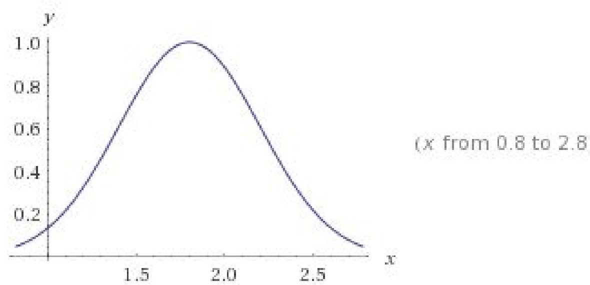


Fig. 13: Probability function assuming normal probability

The probability (out of 1.0) is given on the y-axis, with the voltage/power level given on the x-axis. In this model, the highest output voltage when the drone is directly above a cell phone is assumed to be 1.8V. This means that you can determine the probability of a phone being in an area by this formula of the distribution given by the graph in **Figure 13**:

$$f(x) = \frac{1}{\sigma}\sqrt{2\pi}\,\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

where f(x) is the probability, $\sigma = 1/\sqrt{2\pi}$, and μ is the highest output voltage when directly over a cellphone, assumed here to be 1.8V. So, when given a power level as x, the probability that a person is at those coordinates is f(x). This function can easily be inputted into a program to display the probability when a user clicks a point on the map.

III. Project Management

Appendix B will show a detailed gantt chart. To date, we have accomplished our MDR deliverables (dated 11/26 in appendix B), and parts of our future tasks. High in priority is the PCB design for the RF detection circuitry and the microprocessor. This will be needed so we can test before CDR. Appendix B reveals a large amount of work ahead of us, but we expected that having chosen such an ambitious project.

Our team has a very impressive working relationship. We are in contact through a group chat almost every day with any question or comments we might have. We are all respectful of each other's schedule, and everyone has been flexible with scheduling changes, or design changes.

Each member has a unique skill set and watching that unwind has been fascinating. Brad is a digital savage. He understands computing, and processing at a very core level. Serena has knowledge of everything. She is capable of any task, but she has become our communications/signal processing, probability, and math expert. She has also devoted much of her time to studying for the HAM radio test to ensure that *if* we need to, or decide to transmit, that we will be legal in doing so. Bjorn is the physics, and sensory guru with a general interest in hardware design and RF expertise. Jamie is the go-to guy for anything. He is incredibly knowledgeable with hardware, the software interface, and the realistic design of any project at hand.

The team has many overlapping skill sets which proves handy when there is an excess of work on another portion of the project. It is not always feasible to assign an entire subsystem to an individual, so there have been many occasions where we have had to divide the work accordingly. This task overlap is possible mainly because of our effective communication skills and individual drive.

IV. Conclusion

At this current point in time, we have completed a significant portion of the project. We have accomplished fully autonomous flight control of the drone as well as manual control. We also have the signal strength detector circuit working and detecting cell phone signals at 100' away. The "heat map" functionality has been programmed to overlay sample points over a map when given a set of GPS coordinated and power levels. Additionally, the distance sensor works and accurately detects distance. After PDR, we made a few changes in our detection approach, since we found that detecting the IMSI is not feasible within the time and budget constraints. The next step is to get the PCBs drafted and set to order. We will be working on subsystem integration where a prototype will be put on the drone to test. The hardest part will most likely be detecting the cell phone signal through a canopy of trees as well as learning how to detect the weak signal over the ambient noise level in conjunction with trying to fly at the right speed in order to accurately detect a signal burst from a cell phone. We will have to find the right search algorithm, speed of the drone, and distance calculations in order to successfully implement the detection circuit.

Acknowledgments

We would like to take this time to say thank you to Andrew and team Otto for allowing us to build upon their previous project's platform. We would also like to say thank you to our evaluators, Professors Vouvakis and Janaswamy, whose valuable insight has helped move our project forward. Lastly, we would like to thank our advisor, Professor Leonard, for keeping us on track and offering his knowledge and expertise.

References

[1]"Search & Rescue | Canada North America | National Defence | Canadian Armed Forces", *Forces.gc.ca*, 2016. [Online]. Available: http://www.forces.gc.ca/en/operations-canada-north-america-current/sar-canada.page. [Accessed: 22- Dec- 2016].

[2]"Breaking News, Analysis, Politics, Blogs, News Photos, Video, Tech Reviews - TIME.com", *TIME.com*, 2016. [Online]. Available: http://content.time.com/time/nation/article/0,8599,1892621,00.html. [Accessed: 22- Dec- 2016].

[3]"High Risk Activities Safety Briefing", 2016. [Online]. Available: http://www.lewis-mcchord.army.mil/safety/Publications/PFOR/Sports/Off-Duty_High_Risk_Sports_Briefings.pdf. [Accessed: 22- Dec- 2016].

[4]"DBm", *En.wikipedia.org*, 2016. [Online]. Available: https://en.wikipedia.org/wiki/DBm. [Accessed: 22- Dec- 2016].

[5]"Lockheed C-130 Hercules", *En.wikipedia.org*, 2016. [Online]. Available: https://en.wikipedia.org/wiki/Lockheed_C-130_Hercules. [Accessed: 22- Dec- 2016].

[6]"Dust Devils Deactivated", *Airforcesreview.com*, 2016. [Online]. Available: http://www.airforcesreview.com/reports/Dust-Devils-Deactivated/. [Accessed: 22- Dec- 2016].

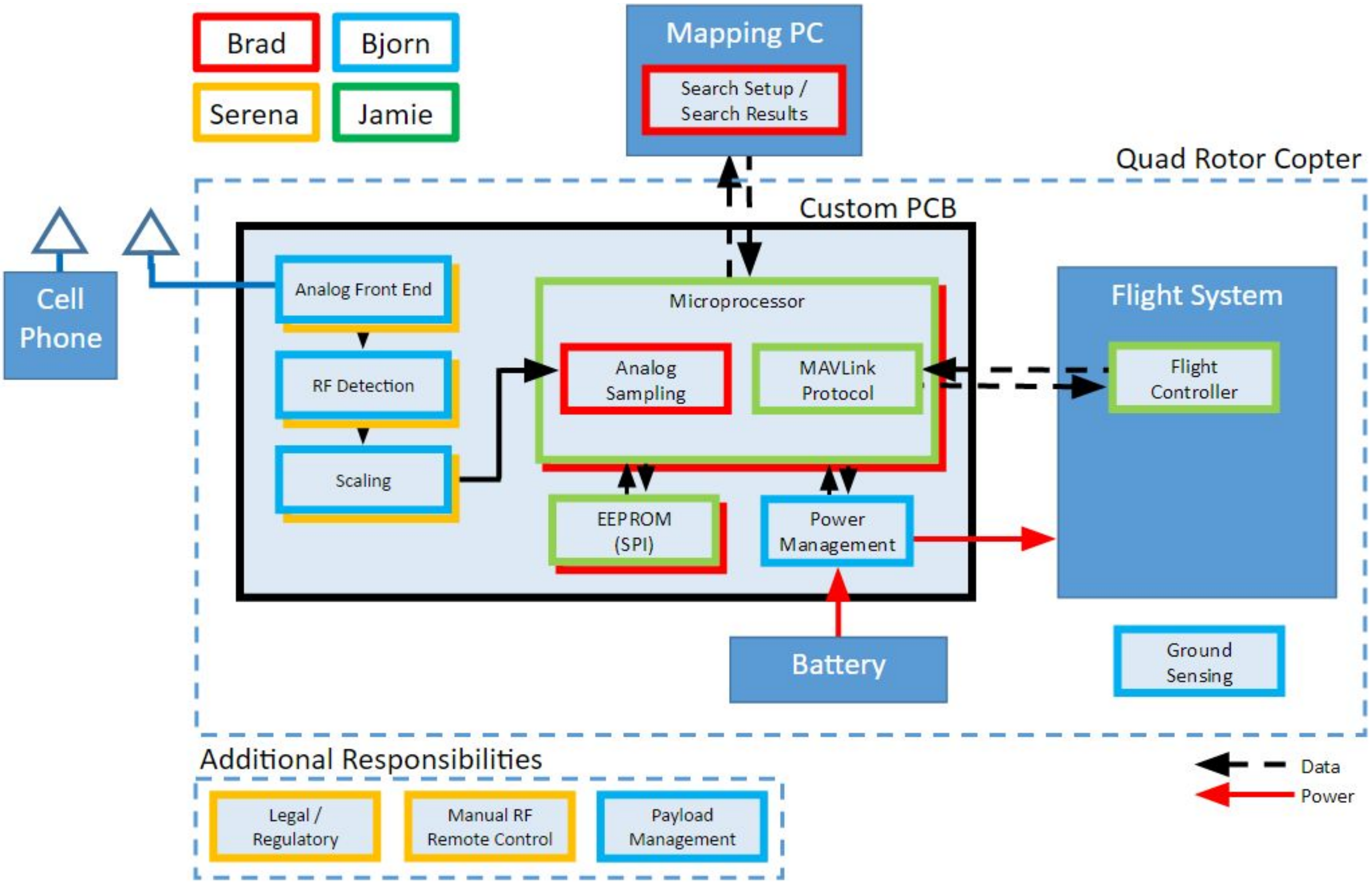
[7]"MAX2015 Logarithmic detector/controller", 2016. [Online]. Available: http://datasheets.maximintegrated.com/en/ds/MAX2015.pdf. [Accessed: 23- Dec- 2016].

[8]"Effects of Trees and Foliage on the Propagation of UHF Satellite Signals", 2016. [Online]. Available: http://www.dtic.mil/dtic/tr/fulltext/u2/a048304.pdf. [Accessed: 23- Dec- 2016].

[9]Y. Meng and Y. Lee, "INVESTIGATIONS OF FOLIAGE EFFECT ON MODERN WIRELESS COMMUNICATION SYSTEMS:", *Progress In Electromagnetics Research, Vol. 105, 313–332, 2010*, 2016. [Online]. Available: https://www3.ntu.edu.sg/home/eyhlee/Prof%20Lee/PIER%20foliage%20review%202010.pdf. [Accessed: 23- Dec- 2016].

[10]Google "gheat code"
Available: https://code.google.com/archive/p/gheat/
[Accessed: 23- Dec- 2016].

Appendix A: Block Diagram



APPENDIX B: GANTT CHART

SAFE DRONE Team 4: Serena Thomas, Jamie Kline, Bjorn Galaske, Brad Marszalkowski

START DATE	END DATE	DESCRIPTION	DURATION (days)
10/12/16	12/5/16	Distance Sensing (Bjorn)	53
10/12/16	12/5/16	Power Detector proto (Serena)	53
10/12/16	12/5/16	Fully autonomous flight/GPS (Jamie)	53
10/12/16	12/5/16	Heat Map (Brad)	53
12/5/16	12/10/16	Website (All)	5
12/5/16	12/10/16	Report(All)	5
12/5/16	1/10/17	Power Supplies/Current Sensing routed for PCB (Bjorn)	35
1/15/17	4/10/17	Terrain Following (Bjorn)	85
12/5/16	1/1/17	Order Last Parts & Daughter-Board PCB (Bjorn & Serena)	26
12/5/16	1/10/17	Detector Output to ADC (Serena)	35
12/5/16	3/10/17	System-wide power/ground distribution (Jamie)	95
12/5/16	3/10/17	Mother-Board PCB (Jamie)	95
10/12/16	1/10/17	Flt Controller Interface/Micro Controller (Jamie)	88
12/5/16	3/10/17	Sampling ADC/Write ADC to EEPROM (Brad)	95
12/5/16	3/10/17	EEPROM Data to PC (Brad)	95
12/5/16	3/10/17	uController Code/State machine function (Brad)	95
3/10/17	4/10/17	Microcontroller Integration (Brad)	30
3/10/17	4/10/17	GUI/Search Algorithm (Brad)	30
3/10/17	4/10/17	Power Detection within 100' (Serena)	30
3/10/17	4/10/17	Efficient Search Algorithms (Serena)	30
3/10/17	4/10/17	Power Algorithms (Bjorn)	30
3/10/17	4/10/17	Host PC MavLink Support (Jamie)	30
3/10/17	4/10/17	Quad Mounting Pwr/Mechanicals (Jamie)	30
3/10/17	4/10/17	Aesthetics (wiring, hardware, paint, etc.) (Jamie)	30