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.

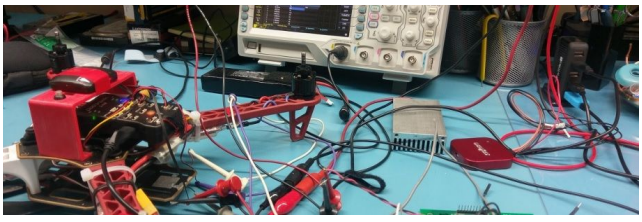


Fig. 2: SAFE Drone Prototype

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 incoming GPS coordinate packets. 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 temperature based heat mapping in a .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 is 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. Actual weights are tabulated in **Table 2**, below and though slightly over the estimate they are still well within demonstrated capability of motors.

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	10	
GPS Mounting bracket	100	
Battery - 11.1V 5200mAh 3S 30C	392	
Cables, connectors (est.)	150	
PCB, parts, antennae (to be determined)	40	
Total (estimated):	**1167**	**g**
Total (actual):	**1208**	**g**

Table 2: Actual vs Estimated All-up Payload

In addition we estimated 80g of additional of unaccounted weight being added for the printed circuit board components to include antennae. This brings the estimated all-up flight weight to ~1300g.

Based on manufacturer specifications, each SunnySky X2212 DC brushless motor 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 1400g (1300g + margin) 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.

Flight Hardware TESTING:

The MAVLink protocol features were implemented using custom C firmware and the MAVLink header libraries. This link is realized over an asynchronous serial connection. The first step was 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 microcontroller 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 LED and others were used throughout the project as diagnostic indicators.

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 microcontroller 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 microcontroller side, once a full packet is detected and the message ID matched to a known stream ID, the microcontroller software is then able to parse out the individual sensor data and use as needed. Using a 4x20 character LCD wired to the micro., we tested successful reception of GPS data by displaying these data on the LCD and comparing them to a cellular phone's GPS sensor as reference.

A longer term goal was to include an onboard energy monitoring feature, to 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 microcontroller 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 micro. 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 micro. LCD screen). After a few second countdown, the micro. 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.

Microcontroller firmware is described in more detail in section *G, Supervisory Microcontroller*.

The next stage of testing involved a flight-ready prototype PCB which pairs the received signal strength indicator (described in section *E. Signal Strength Detection*) with the microcontroller in order to test analog sampling while onboard the quadrotor copter.

C. Project PCB

Our project circuit board encompasses a wide variety of both analog and digital components, representing many disciplines

learned at UMASS (with the exception of circuit board design). This board features:

- Onboard 3.3V/5.0V 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
- USB UART bridge for connection to host PC
- Modular analog sampling daughterboard
- Integration into existing airframe

As mentioned the PCB features a modular sensor design which allows 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 reduced the financial impact risk if modifications to the sensing circuitry were required after testing.

The PCB 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 PCB design and final build is shown in Figure 3, below.

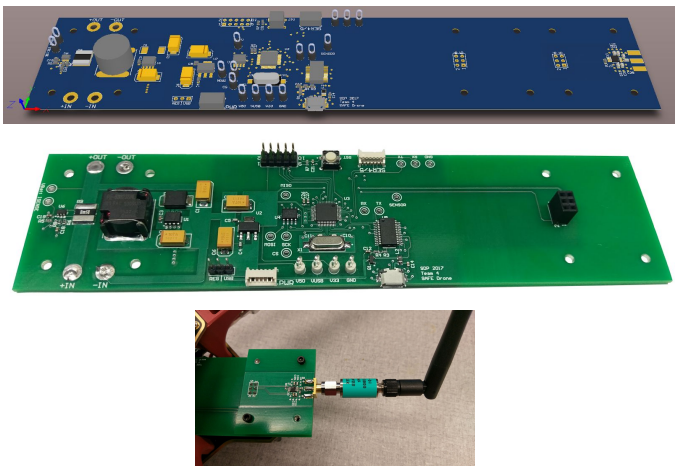


Fig. 3: Draft and Constructed PCB

The power topology of the system is as follows: battery power enters our PCB and immediately passes through a current sensing shunt. The power is then split into paths: a high current path leaves the PCB to power the rotor motor controller. The second path remains on the circuit board and powers a 5Vdc switching regulator. This 5V regulator then supplies a 3.3Vdc linear regulator. We then provide a connector on the board to supply power to the commercial flight controller (5V).

Once the power supply circuits were assembled, we load tested each to our anticipated current requirements. This was achieved using a prototyped DC electronic load which was constructed for our Electronics II course (in which we also needed a way to efficiently load test our switching regulator circuit), shown in Figure 4. We examined each power supply, especially the switching, for proper output filtering both loaded and unloaded. In all cases our supplies produced less than 20mV of ripple/switching noise.

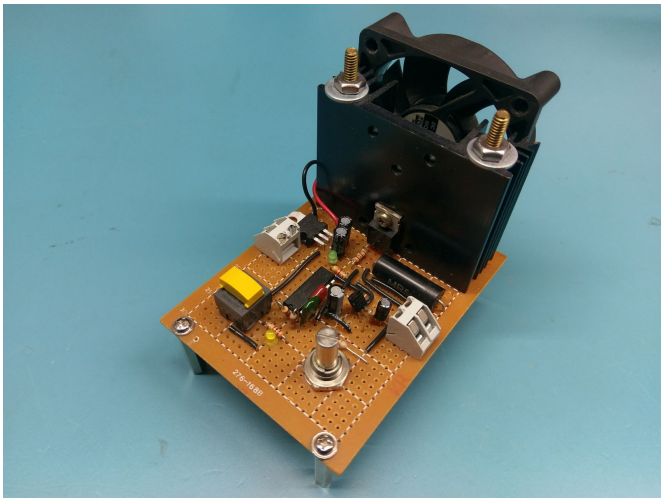


Fig. 4: Electronic DC Load

One challenge we faced related to the power circuitry was a lack of clarity in the flight controller specifications. Repeatedly the power specification for the flight controller was quoted as “5V Vcc”, with “Vcc” identified on the power input connector. However in testing we noted “Vcc Warning” messages in our flight controller logs, stating that our Vcc voltage was too low. After some investigation it was discovered that the flight controller allows for multiple Vcc supply points, for redundancy purposes. In order to achieve this without backfeeding other supplies, the flight controller implements diode OR-ing at each of these separate inputs. Thus our 5Vdc was actually being dropped by 0.3Vdc (the diode forward voltage drop) before reaching the flight controller logic. Unfortunately our board design utilized a fixed output switching regulator. To correct, we purchased the adjustable version, calculated the proper “programming” resistors, and modified the PCB by hand in order to accommodate these additional two resistors. This bumped our 5Vdc output to 5.3V, and successfully eliminated the “Vcc Warning” messages.

A second challenge that was not noted until very late in the project was injected noise onto our power supply rails. We have noticed that when the flight controller is powered from our PCB, the 5V power supply becomes noisy, with a 256kHz pulse of about 100mV being injected onto the rail from the flight controller. Attempts to filter this noise were unsuccessful, and being late in the project phase, was unable to be corrected. It is suspected that a lack of localized storage capacitance near a digital logic (the noise rate a power of two is too coincidental) device inside the flight controller is causing this perturbation in the Vcc source.

Lastly, the ADC (which is sampling the output of our signal strength indicator) accuracy was tested by 'injecting' DC test signals into the ADC line and monitoring the ADC results. This was done using an HP 6632A system power supply and reporting the ADC values constantly over serial port to a terminal. The ADC input signal was tested over the expected range of the MAX2015 signal strength chip: 0.5-1.8Vdc. In order to test for sampling noise, we also monitored this ADC output while rapidly increasing and decreasing all propeller motor speeds (with propellers off). Our results showed that our ADC was accurate to around 3mVdc, roughly one bit of ADC resolution.

D. 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 5** 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 5**, 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 had hoped to implement the terrain following feature after the main subsystems were finished, but the flight controller's automatic landing sequence proved it to be unnecessary.

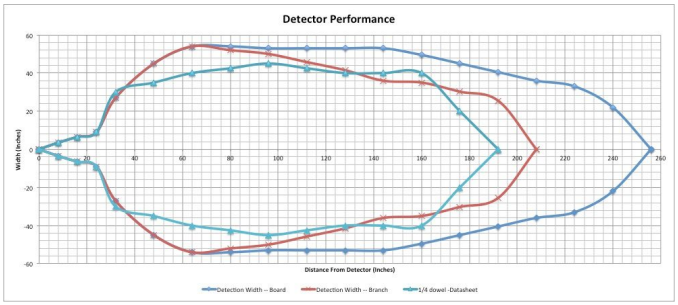


Fig. 5: Ultrasonic Distance and Tree Detection Experiment

E. 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 6**).

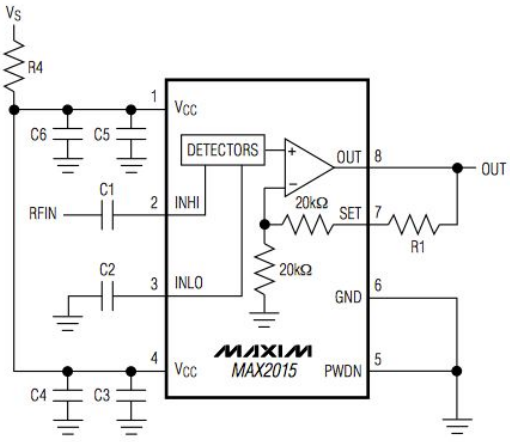


Fig. 6: RSSI Mode Detection circuit [8]

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. 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 designed 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 7**) showing 900MHz at about 0.2dB/m.

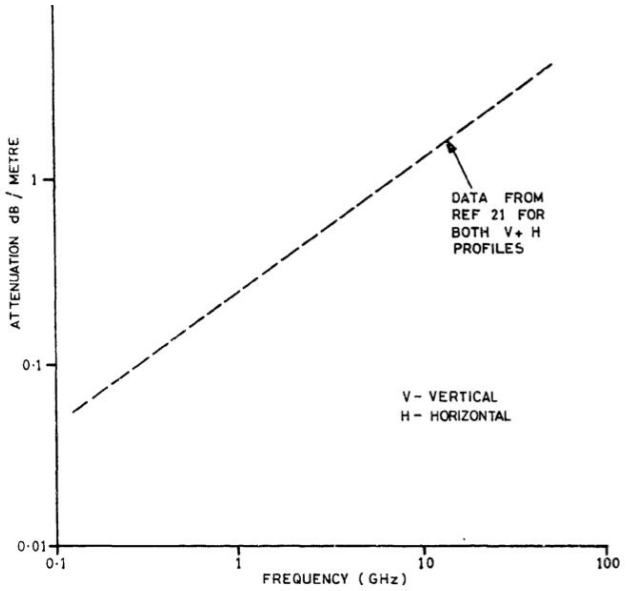


Fig. 7: Foliage Attenuation per meter [8]

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 our purposes, we will make the reasonable assumption that losses from the phone transmission at the forest floor to the drone directly above the canopy range from 1dB-20dB

For design purposes, we 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. 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 8**:

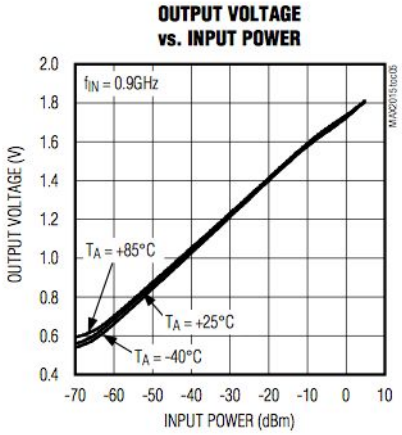


Fig. 8: MAX2015 Voltage Output vs. Input power [7]

In order to audibly test our circuit with various cell phone setups, we needed 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 detected with our final prototype. It has been tested up to 100' away and the circuit has successfully detected the cell phone. The comparator circuit is not used in the final prototype, as the output of the signal detection circuit is already properly scaled to that of the operating range of the microcontroller ADC (described in *G. Supervisory Microcontroller Block*).

The final PCB version of this circuit was tested with a phone that had no service. The results are shown in **Figure 9** where 0.54V is the ambient noise level with no cell phone on.

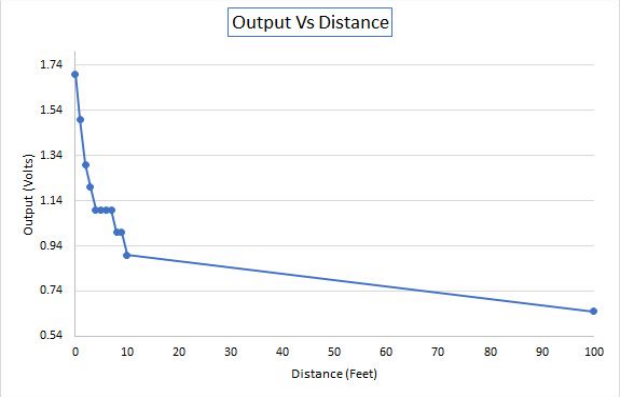


Fig. 9: PCB signal strength measurements vs distance from cell phone

As shown in **Figure 9**, the cell phone that had no service was successfully detected above the ambient level at 100 feet away.

F. 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 10**, below).

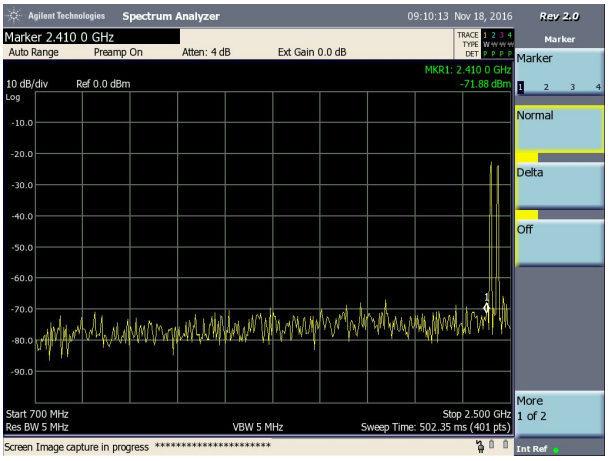


Fig. 10: Spectral analysis of remote control frequency without low-pass filter

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.

G. 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

The device is clocked by an external 16MHz oscillator, and has a 10 bit resolution ADC that samples at 384KSPS. Since the signal we are trying to capture is approximately 500us long and has a 1ms period the sample speed is well above the Nyquist rate. 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 is used to transfer the data to the onboard EEPROM at a rate of approximately one full sample packet every half second.

Also, it has multiple UARTs which are used to:
1. Read the GPS coordinate data from the flight controller via the MAVLINK protocol.
2. Send and receive data from the host PC.

To fully implement this we learned more about the precise software timing. It was found that each sample was being written to the EEPROM within 10ms but each GPS coordinate packet was being received at a rate of 1 packet every 500ms. In order to make up for the latency of the GPS packets we implemented an algorithm to sample the signals as fast as possible, but only save an intensity level if it is the highest out of all intensity levels sampled since the last GPS packet was received. We also implemented interrupt-based serial routines for the MAVLink packet reception, in order to assure no packets were missed. This allowed us to still sample all of the signals we needed to, and map them each to a GPS coordinate even though the GPS packets were much slower than the signal.

All of the data the drone collects is downloaded to the host PC and loaded into the SAFE Drone program. The program is used to create a heat map as shown in **Figure 11**.

Fig. 11: Example Heat Map

The overlay is temperature based, the higher the voltage level (aka dBm level) the hotter the signal. The temperature points are centered at the GPS coordinate points that the signal intensity is mapped to. In **Figure 11** the points where the centers have a lot of red or even a lot of white translates to the 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 12**. 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.

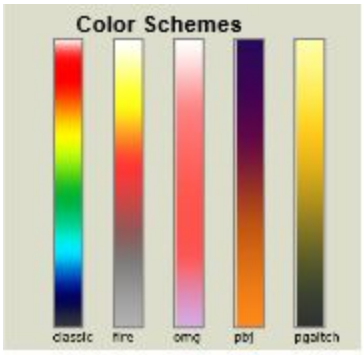


Fig. 12: Available Color Schemes

The higher the signal intensity mapped to a particular GPS coordinate, the further towards the top of the color scheme the interior of 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 13**.

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. 13: 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 temperature mapping for each signal intensity. One major caveat is that not all points taken are seen. The points seen are based on the average of all of the points taken. Using this formula:

$$s_x = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}$$

$n =$ The number of data points

$\bar{x} =$ The mean of the x_i

$x_i =$ Each of the values of the data

$$\text{Outliers} > \bar{x} + \left(\text{DF} * s_x\right)$$

The program creates the heat map from the outliers. This was done in order to keep granularity and thus increase the accuracy of the map. Using different deviation factors yields different results. Using higher deviation factors yields a more refined map but too high a deviation factor can result in no points being shown. Lower deviation factors result in saturated maps so it's best to first look at the standard deviation and make an educated guess off that.

Fig. 14: Deviation Factor = 0

Fig. 15: Deviation Factor = 1

We see from this example that by increasing the deviation factor we get a more refined heat map. **Figure 11** shows this same sample set with a deviation factor of 2 which is very refined and thus shows the location of the signal source.

Lastly, the program uses a python to html software adapter to create a KML file. This is a tool in the google maps API called [10]gheat and it creates 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.

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. This was done in case we needed to transmit on campus for testing purposes or for demo day. The HAM radio license test was successfully completed on February 8th, 2017.

An additional function that could be incorporated 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 18** below.

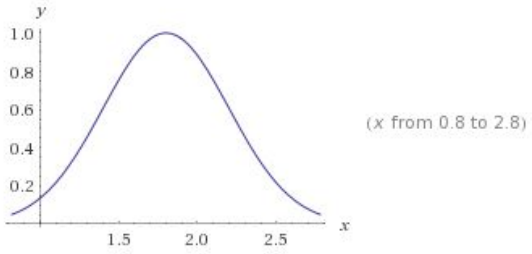


Fig. 18: Probability function assuming normal probability

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 18**:

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

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. This function was not implemented into our final system due to time constraints.

Another additional function that can be incorporated is an antenna reflector to turn the omni-directional antenna into a semi-directional. This may be wanted in order to increase the signal detection power of the antenna that is directed downwards (towards the cell phone). This will help reduce the effects of noise as well. In order to do this, the reflector has to be aerodynamic so the drone can still fly without hinderance. We chose to use a kind of metal mesh instead of a flat piece of metal or metal strips. This metal mesh has to have horizontal (relative to the antenna) wires at least one-sixteenth of a wavelength apart to appear as a solid. The wavelength for the 900MHz frequency is 33cm. This means the spacing of the mesh holes has to be 2.08cm or less. Testing was done on four different types of mesh. They included: a diamond shape wire mesh with 1mm holes, a steel and aluminum grease pan mesh that had less than 1mm holes, ¼" steel chicken wire mesh, and aluminum screen mesh that had less than ¼" holes. These pieces of mesh were larger than one-half of a wavelength wide (16.65cm) and were placed one-quarter of a wavelength (8.33cm) behind the antenna. A phone with no signal was tested with the detection circuit at various distances up to 100 feet apart. The results of these tests on the signal detection circuit are shown in **Figure 19**.

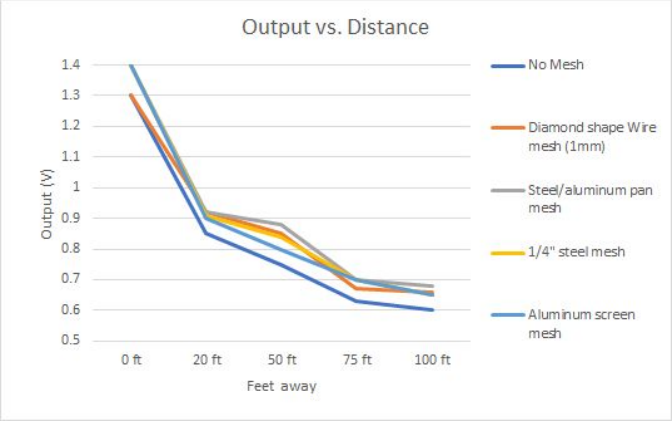


Fig. 19: Output of the signal strength detector vs. distance from cell phone

As shown in **Figure 19**, the steel/aluminum grease pan mesh has the greatest impact on the signal reflection at 100ft away, showing an increase of about 100mV when compared with no mesh or reflector.

An efficient search algorithm is also needed in order to deploy the SAFE drone and maximize the battery life for a single search. This study turned up three different efficient search algorithms, shown in **Figure 20**.

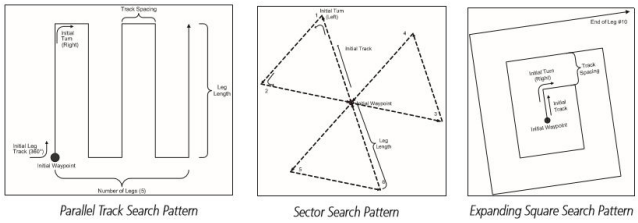


Fig. 20: Efficient Search Algorithms[11]

One of these search patterns will be the path that the search and rescue crew decides to take. The most efficient is determined by the search and rescue crew because it depends on the terrain that is being searched. In all of our flight tests, we use the parallel track search pattern.

We successfully completed this project within the $500 budget constraint as well as successfully met all the required specifications.

III. Project Management

Appendix B will show a detailed gantt chart. To date, we have accomplished all of our deliverables. Appendix B reveals a large amount of work behind 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 successfully completed 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, efficient search algorithm studies and various additional functions have been completed. 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. We have a completed and working PCB as well as a fully functioning prototype. We also have complete subsystem integration. The hardest part turned out to be deciphering the Atmel software framework for our particular microcontroller. We needed to use something professional grade to meet all of our specified criteria and so we had to get spun up on how to program the device very quickly. Overall, this was a successful project that met all the specifications and is able to detect a lost person in the wilderness based on their cell phone location.

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].

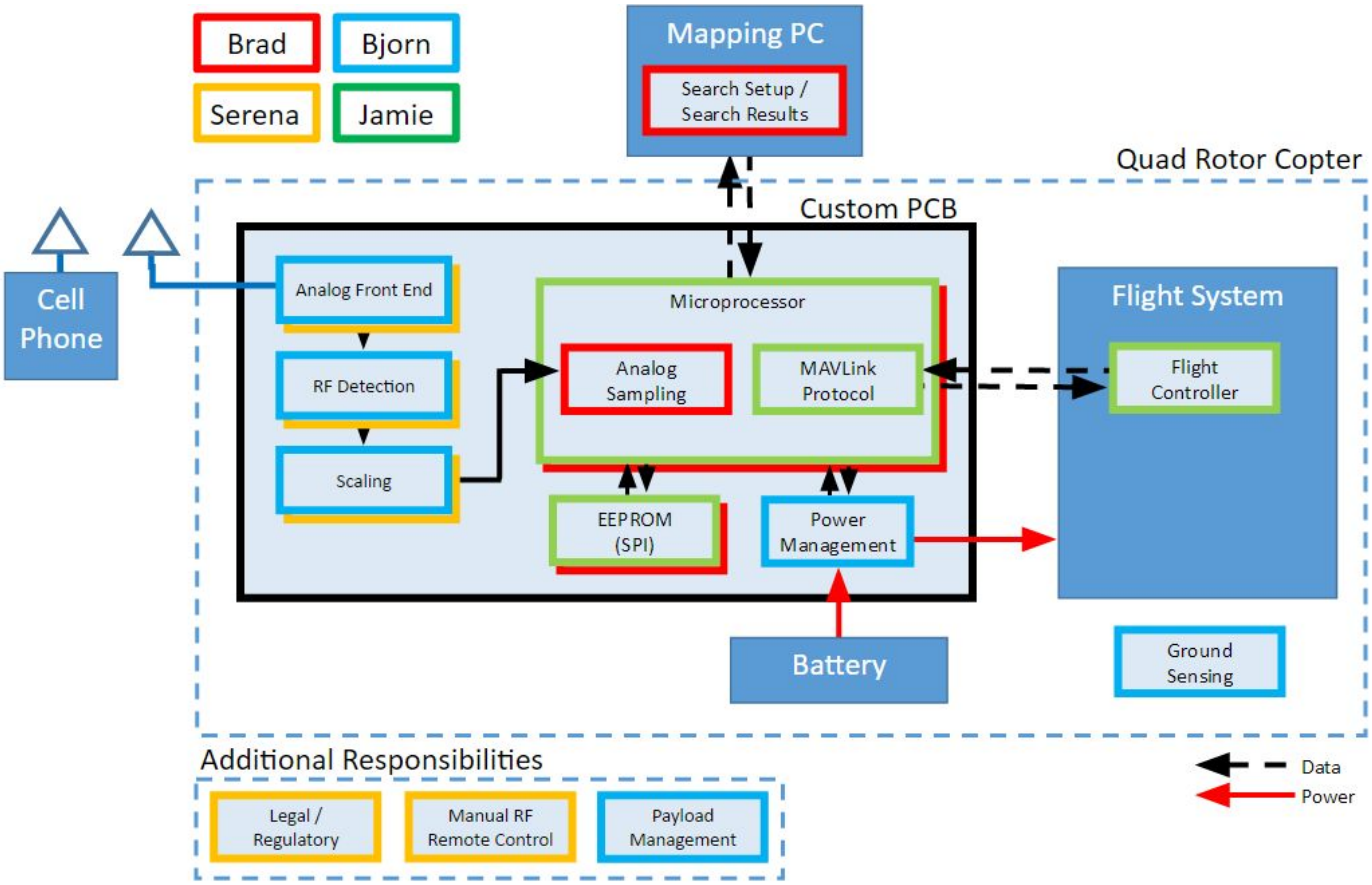
[8]"Effects of Trees and Foliage on the Propagation of UHF Satellite Signals", 2016. [Online]. Available: http://www.dtic.mil/dtic/tr/fulltext/u2/a048304.pdf. [Accessed: 23- Dec- 2016].

[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].

[11]*Integrated Flight Deck Pilots Guide*, 1st ed. Garmin, 2007, p. 9.

Appendix A: Block Diagram



Appendix B: Gantt Chart