D.R.O.P

Delivery Robot with "Otonomous" Parachute

Authors:
Lahari Chilamkurthy, Vikram Bhat, Daniel Cohen

Electrical and Computer Engineering, Carnegie Mellon University

I: Use Case/Application Area

Airdrops are a delivery mechanism for emergency supplies when ground access is not possible, for reasons such as lack of roads or political restrictions in an area. Current technology consists of guided and unguided drop systems. An unguided airdrop can be as simple as a toss from an airplane or helicopter. This type of airdrop can be imprecise, requiring a large cleared area for landing. A guided airdrop, like the U.S. Air Force's Joint Precision Airdrop System (JPADS), is an expensive technology that requires a human to operate via a remote control. With our project we hope to improve on these methods by building a self-guided airdrop system. The D.R.O.P. will be a precise, autonomous, and inexpensive way of airdropping small items for emergencies.

II: Quantitative Requirements

The solution has three requirements, lateral distance of the drop, payload weight, and accuracy. Lateral distance and accuracy requirements were determined in relation to current technology, while the payload weight is driven by the use case.

Lateral Drop Distance

If we consider the ground to be the xy-plane, lateral drop distance refers to the displacement in the xy-plane from the drop location to the target. The drop must be sufficiently displaced from the target in order to test the D.R.O.P.'s ability to correct its trajectory and land precisely. The goal for this metric is 3 meters, modelled after unguided airdrops in Syria. These airdrops required 300 meters of lateral clearance for a 1000 meter drop height. Our drop height capability is approximately 10 meters, the real-life scenario scaled down 100x. Similarly scaling down the 300 meters of clearance by 100x, we concluded that an unguided drop from our test height will yield a 3 meter offset from the target. Therefore our solution must be capable of correcting for at least 3 meters of displacement.

Payload weight

For the device to have any airdrop applications, it must be capable of functioning while also carrying a payload, to model the blood or medicine during real usage. Therefore, payload weight will be 450 grams; the weight of a standard blood bag used in airdrops.

Accuracy

Accuracy is measured in terms of a landing radius from a predetermined target. Our solution is meant to supplement GPS localization already used in guided airdrops, which is precise up to 2 meters. With a landing radius of 2 meters or less, our device will go a step beyond the accuracy of existing GPS airdrop technology.

III: Solution Approach

The solution is a small device that houses the deliverable (i.e. the blood or medicine) as well as a guidance system. The guidance system consists of two tasks, perception and propulsion. Perception is the detection of a target location, while propulsion refers to the movement in the direction of the target. This system overview is fairly common among autonomous devices. The guidance system in conjunction with the physical housing is responsible for delivering the blood/medicine safely from a high altitude to a target on the ground. The autonomous nature of the guidance system allows for a near-seamless "fire and forget" procedure for ease of operation.

IV: System Specification / Block Diagram

The remainder of the section describes each component in the block diagram and how it affects the self-guidance process.

Omnidirectional WiFi Emitter

This is a generic term for the cell phone, computer, or router that will emit a WiFi signal (2.4GHz - 5 GHz). During antenna pre-testing we successfully detected the signal from a cell phone hotspot. This proves that using a cell phone as a WiFi beacon is possible for detection.

Directional Antennas

An array of six directional antennas, arranged radially, receive signals within Wifi range. The choice of six is because of the 66 degree beam-width of the antenna. That is, with six antennas the device receives signals in 360° laterally.

ESP32 Boards

Six ESP32's, one per antenna, compute the RSSI of the respective antenna at a set frequency. The boards are programmable in the Arduino IDE, and the RSSI values measured during operation are sequentially passed to the Raspberry Pi Zero over the inter-integrated circuit (I2C) communication protocol.

Raspberry Pi Zero

Main computational hardware. Contains a program developed in-house that takes six sets of RSSI values as input and produces three PWM signals as output.

Digital Filter

A filtering technique is required to remove noise from a sequence of RSSI data. Following the filtering process, the WiFi RSSI data is usable for direction finding.

Direction Finding Algorithm

A program that first determines the antenna(s) that produce the strongest signal. Then, the direction and magnitude of the antenna's signal is vectorized in terms of the three component directions the device is capable of moving in. Finally, the vectorization is expressed as three PWM values to be used by the motor controllers.

V: Implementation Plan

Components to be designed:

Antenna perception system

This system includes six directional antennas, each connected to an ESP32 WiFi equipped board. A short piece of C code on each ESP32 will search for a specific WiFi SSID (of the target beacon) and acquire the RSSI of that signal. This value will then be transmitted to the filter software running on the Raspberry Pi via the I2C protocol.

Digital filter

This filter will run on the Raspberry Pi in Python and remove noise from incoming RSSI data. We are considering filtering techniques such as median filtering which attempts to remove erratic spikes using a sliding window technique, or signal smoothing with a moving average filter. The resulting data is then passed on to the vectorization algorithm in an array.

Vectorization algorithm

This python algorithm runs on the Raspberry Pi and takes in the RSSI values of each antenna (passed in an array from the filter), and computes a PWM duty cycle for each of the motors to be fed into the control system. More specifically, it will take into account all antenna RSSIs and solve for the path vector needed to decrease distance to the target. It will then compute PWM duty cycle values for each motor in order to achieve that vector based on the direction of each motor on the device, and the relative direction of the target.

Propulsion control system

This system involves the code on the Raspberry Pi that controls the speeds of the motors. This involves a Python script that generates a 50 Hz PWM signal to control the ESCs (Electronic Speed Controllers) which are connected to the motors. It will take in the output of the vectorization algorithm as a duty cycle value for each motor and output the signal on the GPIO for the corresponding motor.

Housing

The housing will be a hexagonal prism built from laser cut or hand cut material that will contain the payload, compute, propulsion, perception and attachments to the parachute. A hexagonal prism was used as we are using six antennas, so each face could hold an antenna.

Components to be bought off the shelf:

- 6 x Directional Antennas
- Brushless Motor Controllers
- Raspberry Pi Zero
- 3 x Brushless Motors
- 2 x Parachute
- 3S LiPo Battery

- Omnidirectional WiFi Emitter (Hotspot)
- Cables and connectors
- Logic level shifters and power converters

Pre-implementation testing:

Antenna Directionality Testing

In order to determine if using directional antennas to discern direction is possible, we first had to measure their sensitivity to different angles. For this test, we wrote a simple Arduino program that searched for a specific SSID of a personal phone hotspot, and continuously printed the RSSI of the signal in a loop. We walked around in 5, 10, 20 and 25 foot radii semi-circles around the antenna, pointing an iPhone WiFi hotspot at it, and measured the RSSI as we swept across the circumference of the semi-circle. We oriented the antenna such that the 16 degree beam-width was in the same plane as our horizontal semicircular sweep. Below is the testing setup.

Figure 1: Semi-circular sweep test

Figure 2: Antenna mount

Some results from this test can be seen below

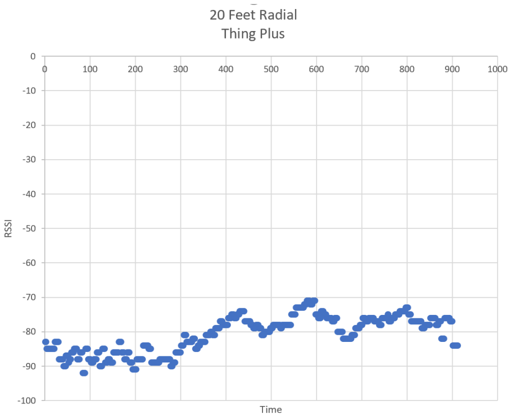


Figure 3: Results from a 20 foot radial sweep test

As can be seen in figure 3, the antennas have a pair of sensitivity peaks, one at each edge of an approximately 30 degree angle from the center of the antenna. Theoretically, the antenna would have a single peak right in the middle of the 180 degree sweep for 16 degrees. Clearly, upon inspecting the data, the 16 degree band of high-intensity readings cannot be reproduced. Additional testing is required in regards to vertical beam width and the feasibility of blindly finding a direction using these antennas. For example, we could rotate the antenna with a hotspot nearby and ask a team member to stop the rotation once they think they have found the hotspot direction, just from looking at the data. However, we foresee issues in this respect as looking at the data we have gathered, there is no distinguishing feature between the target being at the left peak's angle, or the right. Additionally, there is no difference between the signal strengths of the middle of the peaks (dead center of the antenna), and outside the two peaks, a region which theoretically should have the greatest signal strength.

Motor Thrust Testing:

The listed thrust of the motors using similar propellers to ours was 1000g. To test this, we designed a platform that we can attach our motor to and place on a scale to measure thrust. The housing is weighed down using weights, and the motor is turned on; the difference between the weight with the motor off, and the weight with the motor on is the measured thrust. Below is the test setup.

Figure 4: Motor thrust testing setup

The value we measured was 400g. Due to our makeshift thrust measurement setup, we theorize that a significant amount of thrust was lost due to the torque generated at the point of contact of the arm to the setup body. As the motor generates thrust upwards and lifts the setup, there is a force generated downwards due to the wooden slat acting as a moment arm. As a result, some of the thrust is cancelled out. Additional testing will be done to see if this thrust is enough to move our device at least 3 meters.

Drop Speed Testing

To find out how long our device would be in the air for, we dropped a plastic container filled with water attached to our two parachutes from a height of around 35 feet. The plastic container was filled with water to a total weight of 2.5 lbs, the maximum target weight of our device including the payload. The test setup can be seen below.

Figure 5: Dropped object

Figure 6: Drop test

We measured a total drop time of around 3.7 seconds. This gives us a better idea of how long we have in the air to perceive and compute a direction to travel to, as well as to actually propel the device all before hitting the ground.

Prototype Housing

We built a prototype housing made of Foamcore and plywood. This will act as a testing platform for us to see how our propulsion system is able to carry our entire device and payload; additionally, we will be able to attach an antenna to each of the faces to test the antenna system during a drop. There also is a lid with slits for the parachute that will attach to the body with a hinge. The inside will contain the compute, ESCs, battery and payload.

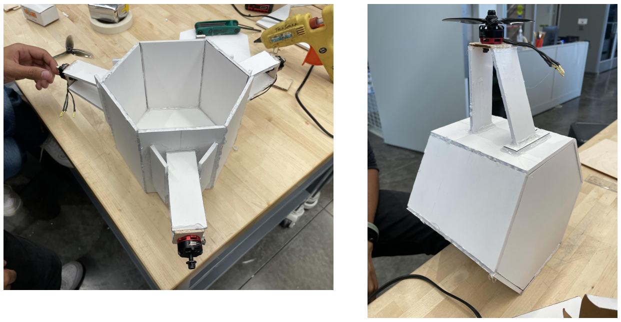


Figure 7 and 8: Prototype housing

VI: Trade Studies

There were a few different categories that we encountered choices for while deliberating on how to implement our solution. The main categories were methods of propulsion, perception, and compute method/component choice.

Propulsion

For propulsion, we looked into the various ways that we could create movement for our device, in order to guide itself to the correct location.

Compressed Gas

We found compressed gas as an option in order to cause reactionary movement from firing high pressure gas in certain directions. This idea requires storage of high-pressure gas and the means to control it’s release. Pneumatic components that enable this are high-density metal, and are generally very heavy in order to withstand the pressure of such gas, which is not feasible for our project because of our low weight requirement.

Moving Center-of-Mass

Steerable parachutes were used for the JPADS system developed by the US Military for guided airdrops that we took inspiration from. Similarly, we were thinking that using a set of moveable weights within the device in order to change its center of gravity could be used to control the direction of descent. The changing center of mass method is not able to create “instantaneous” motion, and requires the object to descend in order to move laterally. Because of our stringent drop height/time requirements, this option simply doesn't have the power needed to maneuver given the heights we want the device to operate at.

Motors and Propellers

Drones were immediately an inspiration for this project, so brushless DC motors designed for drones stood out as an option. One downside of the drone motor method is that motors require a larger battery, and one that is a higher voltage than what the rest of our electronic components require. This adds to our weight, which is detrimental, but we can help counteract this by using this same battery with a step-down converter to power the rest of our electronics, and minimizing the size of the battery because the motors will only being pulling current for a relatively short period of time because of our drop speed.

Perception

For perception or sensing, there were a couple different methods that we explored. Primarily, we looked at high-accuracy GPS (RTK), various cameras, and different RF signals for direction and location sensing.

High-Accuracy GPS (RTK)

Many of these options for sensing were not possible due to our use case, where minimal prior infrastructure can be used/assumed. This eliminates RTK because it requires static beacons placed in 20 kilometer increments from the target area, and in remote locations this may not be possible, and can't be assumed.

Cameras

For visible light cameras, because we want our system to be operable at night or on hazy days with limited visibility, it may be hard to use a standard camera by itself. An alternative to this is to use Infrared or thermal cameras, using infrared/thermal data as well as visible light data to detect some sort of marker on the ground. We are choosing this as a backup plan because it matches most of our use case needs. A downside with cameras however is that they require a relatively significant compute source in order to handle computer vision/detection. This means that our Raspberry Pi compute board might have to be upgraded from a low end Raspberry Pi Zero to a Raspberry Pi 4 in order to handle the higher amount of calculations needed for computer vision.

RF Signal Direction Finding

For the RF signal perception route, WiFi was chosen because of its ubiquity in modern phones, as well as the plethora of antenna choices available commercially to build our platform. Using directional antennas to sense relative direction to a WiFi source allows for fast percepts and low computational overhead. The issue with this system, however, is that the dimension of data in each perception is relatively low, and limited by the number of antennas present on the falling device. Because of this, our primary goal is to use antennas for perception, and if this proves to not be feasible because of the aforementioned downsides, we will switch to an IR/Thermal-camera based CV approach.

Computation

For compute component/design choice, our options were again limited by the weight requirement of our system.

Off-Board Compute

We had given some consideration to using an off-board compute station and using wireless signals to communicate back and forth between a heavier powerful ground station and a lightweight drop unit that relays sensor data. We deemed this method infeasible for two reasons: higher overall latency and the necessity for pre-built ground infrastructure, which impedes the ability of the system to be used in true "emergency" situations.

Single Board Computers (SBCs)

The most common methods of achieving onboard computation are SBCs, or Single Board Computers. Namely the Raspberry Pi series and the NVidia Jetson Nano. Microcontrollers are also available, which are advantageous because of their size and weight, however they are severely limited in computational ability, which affects our ability to do high-level control and techniques like advanced filtering or image processing, both of which might be necessary given our perception choices.

Within the realm of SBCs, the Jetson Nano poses an issue because of its weight with the included heatsink. Because of the Nano's immense computational power within its form factor, it consumes a significant amount of power, necessitating an appropriately sized heatsink. This aluminum mass causes the Jetson Nano to weigh in at 0.55 lbs, which eats a significant portion of our non-payload target mass, making the Nano not our first choice for onboard computation. This leaves the Raspberry Pi series boards, more specifically the Raspberry Pi Zero and the Raspberry Pi 4. These two boards represent the two ends of the spectrum, from miniscule and less powerful to slightly bigger and more powerful. Here, the Zero is our first choice because of its weight coming in at a mere 9 grams. Compared to the Pi 4, it isn't as powerful, but for basic filtering and control it is sufficient. If we decide in the future to go down the CV route, the Pi 4 is a good upgrade, allowing us to reuse code written for the Pi Zero and take advantage of the 4's more powerful processor for higher framerate CV processing.

Thus, we have explored numerous options for perception, propulsion, and computation, and analyzed each one carefully before narrowing down our decisions to a few components that best match the parameters of our project, while giving us the most flexibility possible within them.

VII: Bill of Materials

Item	**Quantity**	**Price per item**	**Total price**
RS2205 Motors (4-pack)	1	33.99	33.99
Propellers (16-pack)	1	15.99	15.99
Brushless ESC	4	16.49	65.96
Bullet Connectors	1	15.99	15.99
Antenna	6	18.99	113.94
ESP32 WiFi Board	6	13.22	79.32
LiPo Battery	1	17.00	17.00
Parachute (2-pack)	1	15.99	15.99
Housing material	-		40.00
		Total $	**398.18**

VIII: Tools

The process of fabricating our project will require many resources available at Tech Spark and the 18-500 lab. Namely, the housing was designed using the Fusion 360 software and will be manufactured using the facilities and materials at TechSpark. We have been using other resources in these two facilities like a soldering iron, wrenches, screwdrivers, nuts, bolts, and electrical tape. The WiFi boards are programmed with Arduino IDE and use Arduino Wifi library. The Raspberry Pi is programmed in Python using I2C and pigpio libraries.

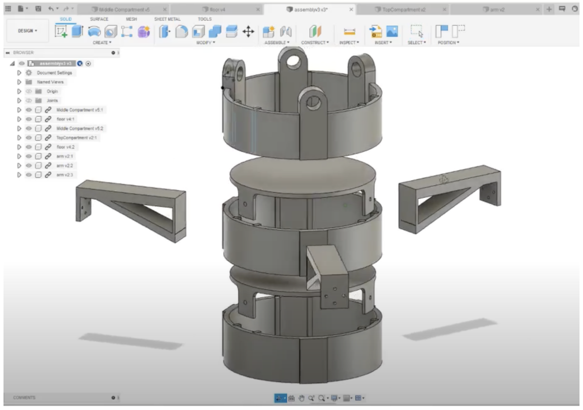


Figure 9: Housing CAD model

IX: Metrics and Validation

Drop height is one of our test inputs, it is about 12 meters and determined by the height of our available test site. The primary test input is the lateral drop distance, which is 3 meters as stated in the requirements. This will be a test of our device's ability to correct its trajectory.

Our testing site is going to be the Pausch Bridge, and our schedule allows for 2 or more weeks of testing. To vary the wind conditions during our trials, we will test on at least 10 of the 14 days. The tests can take up to 10 minutes each for setup, including a member of the team returning the device to the top of the bridge.

X: Risk mitigation

Our mitigation plan at this point is in relation to the use of directional antennas. We will continue the pre-testing to evaluate the antennas capabilities. An example of a successful test would be to hold the antenna(s) blindly and detect direction solely from the data. In the event this mode is not meeting our requirements in practice, we will have to change the mode of perception. The alternatives that we have lined up at this time are IR and thermal imaging. In other words, we will be resorting to an IR or thermal camera to replace the antennas, and the basis for our direction-finding algorithm will switch from RSSI data to computer vision. The target will change from a WiFi beacon to a marker or heat source, which also aligns with our use case in terms of user accessibility.

The camera will give more predictable data than the antennas, but we will face a tradeoff in both weight and compute time. The computer vision will likely require a Raspberry Pi 4 instead of Raspberry Pi Zero, a weight increase of 35 grams. Image processing is also a more intensive compute task because the new signal data is 2D (pixel values) rather than 1D (RSSI values). The latency involved with this new compute task will reduce the number of perceptions possible and has the potential to hurt our accuracy rate.

To briefly compare the two camera options, IR is better for the use case. With IR, the camera will detect a marker instead of a heat source. The marker is more reasonable for the user to provide than a heat gun (or to risk injury using their body as a heat source). Thermal imaging provides lower resolution which will result in lower compute times. There is also some risk of temperature conditions interfering with the thermal camera. CV with an IR camera is dependent on light conditions, whereas thermal imaging would work just as well under low light conditions.

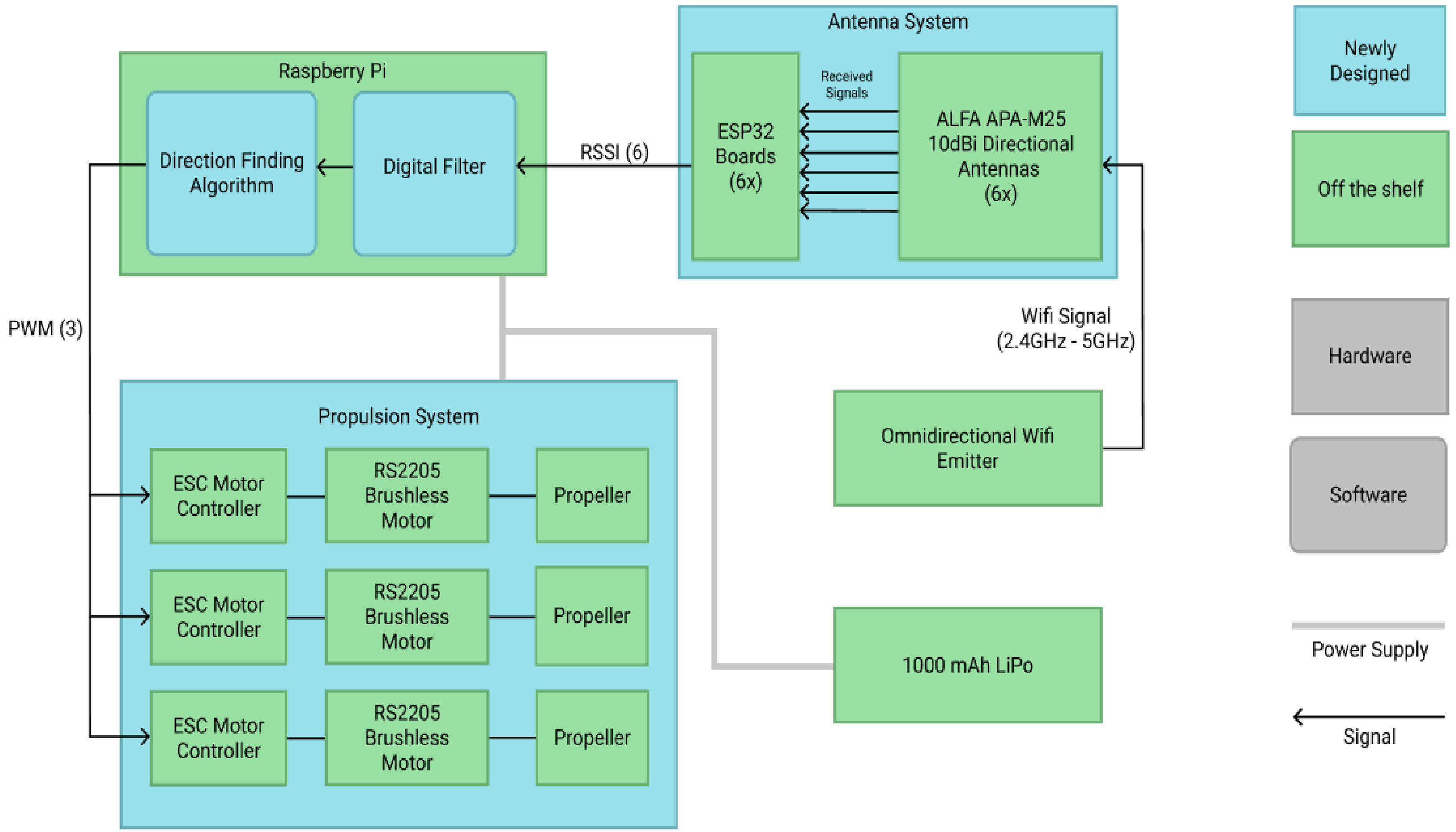
XI: Project Management

Our work distribution is presented in the schedule on the following page. The tasks are divided mainly by ECE area and any additional skills. Vikram is responsible for the mechanical side of the project, with tasks related to the housing and motor system. Daniel is responsible for the direction-finding algorithm and Lahari is responsible for signal data filtering and putting together the antenna system.

A small change has been made to the schedule we proposed two weeks ago. Namely, we will no longer be 3D printing the housing. Vikram completed a CAD model and attempted to 3D print it during the past two weeks. However, we found after printing just one small section of the design, that the weight of the full model would be too heavy. We quickly switched to fabricating the device housing out of Foamcore and plywood, in order to move forward with creating the MVP. We plan to replace the Foamcore and plywood with a more durable material during the integration phase, after we have completed more tests where we drop the device.

In our contingency plan, we switch from directional antennas to cameras. Vikram's tasks would remain approximately the same aside from integrating different hardware, like a Raspberry Pi 4 and camera. In this scenario, we would adjust the schedule to have Lahari working on the computer vision program to detect a marker. Meanwhile Daniel would have to adapt the direction finding algorithm to use relative position from an image instead of RSSI data.

XII: Block Diagram



XIII: Schedule

🟥 Lahari
🟩 Vikram
🟦 Daniel
🟧 All

	10/3 - 10/9	10/10 - 10/16	10/17-10/23	10/24 - 10/30	10/31 - 11/6	11/7 - 11/13	11/14 - 11/20	11/21 - 11/27
Pre-Tests								
-Drop Speed	🟧							
-Propellor Thrust	🟧							
Housing								
-CAD	🟩							
-3D Printing		🟩						
Antennas								
-Assemble Antenna Array			🟥					
Propulsion								
-Assemble Motors and Propellers		🟩	🟩					
Controller								
-RSSI Filter Design	🟥	🟥						
-Direction Finding from RSSI	🟦	🟦						
-Propulsion Vectorization		🟦	🟦					
Integration								
-Antennas w/ Raspberry Pi				🟧				
-Raspberry Pi w/ Propulsion				🟧	🟧			
-Assemble Parts in Housing					🟧	🟧		
Testing						🟧	🟧	🟧