Senior Design 1 Initial Project Document

April 27, 2021

D.E.A.R Drone

Department of Electrical Engineering and Computer Engineering
University of Central Florida

Dr. Samuel Richie

Group B

Austin Perkins	Electrical Engineering
Raymond Chenoweth	Electrical Engineering
Ellie Lane	Computer Engineering
Dan Biller	Computer Engineering

Table of **Contents**

Table of Figures

Table of Tables

1. Executive Summary

This project was designed around the idea that each member would incorporate his or her own skill set and past into this project. The team members consist of two electrical engineers and two computer engineers, so it must incorporate aspects of electrical analog systems and also software design and the mix of these two skill sets is a computer-based system in a mechanical machine format. We then dived further to incorporate our past projects and ideas into the “flow” and give us new ideas and problems to work to fix. We took in Ellie’s past in machine learning, Dan’s past in drone manufacturing, Austin’s background in mechanical systems, and Raymond’s background in electrical analog systems to come to the idea of drones using machine-based learning. We incorporated the idea of accident avoidance, and the current environment in the world in response to Covid and how that limits trade and thought of a faster response to delivery in using drones and accident avoidance software to fly safely and quickly through urban environments that trucks find a hard time navigating though. This combines together to form the problem and solution of our drone project.

Now that we have the base of our idea, we now need to go through the difficulties and needs of doing something like this drone would need. We immediately started the discussion of how our drone would achieve the main component of this project, accident avoidance. We need a system that can think fast, incorporate multiple data points and parameters, and act in its best interest to the circumstance. This is where the different aspects of our disciplines will come into play. We must have a CPU that can handle these different parameters as well as reliably fly the drone as this will be an automated system instead of direct control like most commercial drones are. We will then need different sensors to incorporate different data points so the drone will know when it needs to avoid an obstacle or when it is safe to fly. We need to incorporate computer vision with a camera for even more accuracy as well for different environments that some sensors won't work well in. Then machine learning will be directed to the project through the use of simulations, thus “teaching” our drone before any need for physical testing, which could cause hundreds of dollars in damages to the product depending on what happens. And finally, we need to adjust our drone to our specifications of weight, lift, and space to incorporate all our different sensors and package space.

We have certain aspects of these designs that we need to attain for the drone to be viable for operation, and our objectives follow this route. Starting backward from the drone frame design, we need to limit the weight of the drone frame, sensors, brain, and all other aspects of the weight because of diminishing returns on lift compared to the motors used for lift. For safety and ease of operation, we’ve designated half of our total weight that our motors can lift to the actual drone, thus giving the second half to flight operation and the ability to avoid obstacles when needed. Next, this drone itself would be the prototype so there is an obvious limit to the size and weight of the package it would carry, but it would still need to be physically adjusted to reliably carry our package. Our drone needs to have safety incorporated into every feature of its design as well because this product has the potential to directly interact with customers when delivering or picking up packages. We need to incorporate such things to guard against accidents. That is half taken care of by our accident avoidance program, as that will avoid directly coming into contact with any humans or animals during operation. The other half must be taken care of through our physical modifications to our drone itself or safety manuals we supply for operation. We also need to have a long battery life so as to have enough energy for our drones to reach destinations that can be on other sides of cities, which when taking into account the growth of cities, can be long distances. The drone will operate completely independently of any control networks and will be capable of working even if it loses all networking signals. It is fully autonomous and all of the decision-making happens on the flight control board. The computer will use GPS for coarse path planning of the overall trip. The sensors and computer vision will be used for fine path adjustment. This means that no human intervention is required throughout the flight process. While the drone does have a network connection to the ground station, updates and alterations can be issued to the drone during flight. Also, the flight can be canceled and the drone can return back to its launch site. This means that the operator has the flexibility to control the drone if they want to, but it is not a requirement for a human to take any actions during a delivery flight.

With proper budgeting and realizing where and when need to acquire higher quality products versus bulk, we will come closer to a realistic budget and what we actually need to spend. Currently, we are in the agreement of a budget of $1000 with a deviance of plus or minus $200 depending on design changes down the line. Sticking to this budget will require compromises and decisions to be made, which are similar to what companies making a profit will have to make also. Our time frames are realistic and achievable and incorporate design and writing. We will follow our own timeframes so as to stay ahead of deadlines imposed by outside influences. Time management is an excellent skill to have in any workforce environment and we will be improving upon ours in order to maintain the quality of output for the project.

Research has shown us that while our project is being incorporated into many companies, it is the edge of research into its respective field, and any advances into the aspect of accident avoidance will provide leaps and bounds into logistics and advancement into the field of drones and the future of the supply lines in the world. This is already steadily looked into because of events in our world such as Covid, the blockage of the Suez Canal, and many others that have affected delivery times and cost the global market millions of dollars. We already see this in such things as Matternet, Zipline, and the top competitor Amazon. This is only taking into account drones as well. Accident avoidance can be incorporated into hundreds of technologies, such as autonomous driving, helping in medical fields for the disabled, and many more. The progress being made by drone projects will continue to perfect those efforts and see where it can be used more in day-to-day life while also providing extreme convenience for people needing deliveries. Imagine being able to order a project, medicine, food, and much more and have it delivered within the hour because the city in your area is able to package it up and send it on a drone to be delivered to your door.

2. Project Description and Motivation

With the rise of artificial intelligence in recent years, coupled with cheaply available multirotor platforms, autonomous drones are being used in more places. Furthermore, the jobs they perform are becoming increasingly complex. One of these roles is package delivery.

Figure 1: Obstacle Avoidance

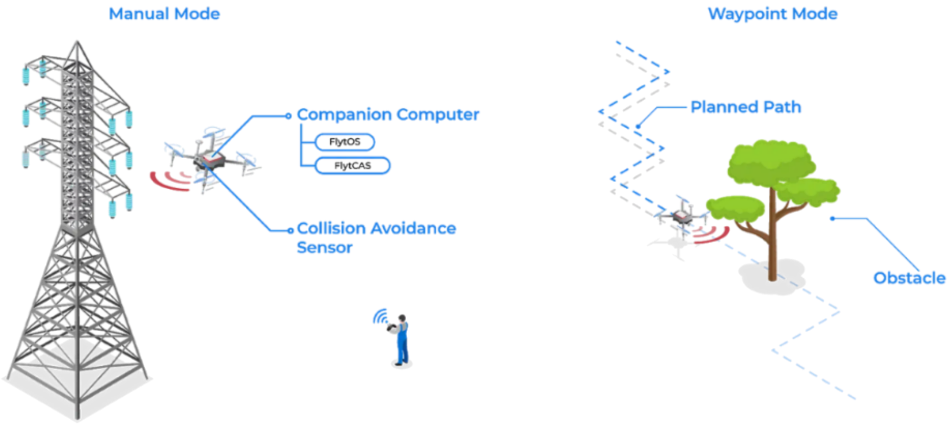
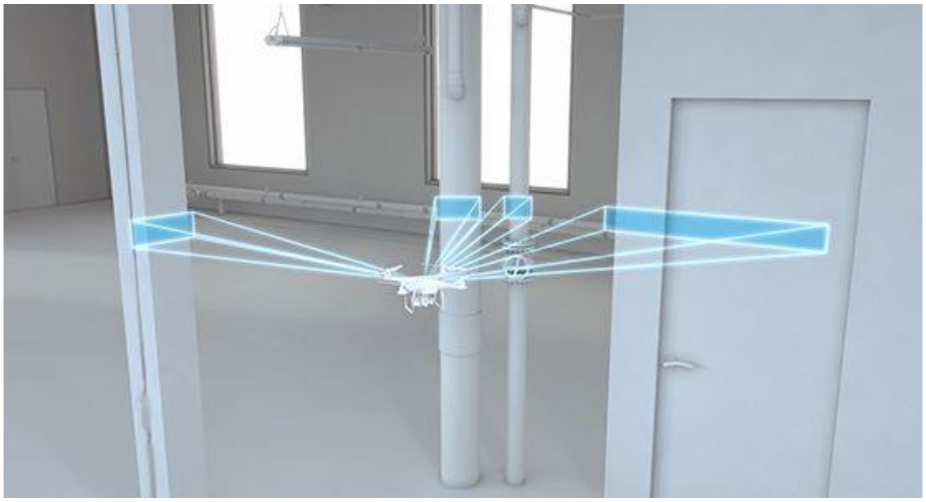


Figure 2: Proximity Detection



Online shopping has been trending upwards for years, and the recent pandemic has only accelerated this. This and other package shipping services require a cheap and scalable method for last-mile delivery: the delivery from local distributors to customer’s homes. Additionally, it can be problematic to deliver medical supplies and food to remote areas that are not accessible by vehicles.

Drones are the perfect platform to fill this role. They are cheap and scalable, as well as being able to travel quickly and navigate to non-accessible places. However, one problem with current drones is that they lack the ability to detect and avoid local obstacles. A general course can be planned and followed using GPS and terrain mapping data, but there is still a need to avoid things like buildings, trees, and other drones **Figure 1**.

The project we are proposing is a payload delivery drone with the ability to detect and avoid local obstacles **Figure 2**, enabling fully autonomous package delivery.

Our motivation comes from the last huge change in the distribution processing would have to be shipping containers. What shipping containers did was made a standard unit that could be transported easily on ships, trains, and almost every form of transportation that we currently use. The impact that had was astronomical on the global economy. Can be easily stated that shipping containers are the cornerstone for the world trading market currently. We believe our drones could potentially have the same level of impact but on a more local scale. When improving the distribution of packages and mail. You are effectively making the world a more closely net Community.

Regardless of if you are delivering postcards or Christmas gifts you are essentially bringing people who cannot be in direct contact closer together. That means family who live out of state, our brave soldiers who are serving this great country are more connected to their loved ones when you improve package delivery and distribution. That is our Motivation for building a drone that can deliver packages, ultimately, we believe we are bringing people closer together than ever before.

2.1 Goals

Our first goal is to use collision avoidance in combination with GPS. The sum total of all of our ideas is to make a drone that is capable of maneuvering itself and delivering a package. In hopes to revolutionize the distribution of our mailing systems of today. The goal is to make distributing mail quicker, cleaner, safer, and more effective than it has ever been before. In doing so we believe the best way to do that is with a drone that has an external camera, IR sensors, and GPS, that can function practically anywhere and deliver packages safely and effectively.

Another goal is to make sure our drone delivery systems are as diverse and capable to match the surroundings and environment. With the use of GPS and an external camera, we hope that our drone will be flexible enough to safely maneuver the number of objects and avoid them. While continuing its path to deliver the package.

To be able to create a drone that is capable of maneuvering in accomplishing such a goal, we need to apply knowledge of kinematics and robotics with a touch of aeronautics. We will create a test bed that will allow us to test the drone’s ability of object avoidance using machine learning before applying it to her actual drone. Once applied to the actual drone we will have the computations be run on the flight controller that will monitor the position and orientation of the drone autonomously.

Our final goal is to inspire the next generation of future minds to be interested in technology and how the world functions and what can still be improved upon and used to make the world a more effective and better place. These drones would be used and highly populated cities in towns and we hope that the youthful spectators will be inspired to look up our DIY instructions of how they could build their own drone and hopefully become drone hobbyists.

2.2 Specifications and Objectives

To guide us with what we will be accomplishing in our drone project then specific objectives and specifications need to be made. These need to include clear direction and numerical values of what we want to achieve. If these items are not made clear or numerical then there can be miscommunication and not a very guided direction to head in when it comes to making decisions about how to build.

2.2.1 Objectives

Our methods to achieve our goals for this project, is more of a divide and conquer approach to hopefully and successfully complete this building project.

The first objective of our drone build project is to acquire the necessary hardware such as the frame, the motors, propellers, ECS, and flight control. It is important that these materials have the proper specifications for our drone objective. The frame must be large enough to support carrying a package as well as giving us enough room to build on top of it. The motors and propellers must provide enough to lift tonight only carry the frame in the hardware but as well as a package and be efficient enough to last the duration of the trip. The flight controller needs to be robust enough to handle the inputs from all of our sensors to accurately maneuver the drone effectively to its respective coordinates.

The next Objective will be to make sure we have the proper batteries incorporated into our drone build. The importance of having the proper batteries is critical for multiple reasons. First being you don’t want too heavy of batteries for your needs because ultimately it will way down your drone and make it less efficient. Ultimately you would want a Goldilocks middle region of a drone that supplies the perfect amount of power for our flight time but it’s not too heavy to weigh down the frame.

Successful object detection, tracking, and collision avoidance is our most important objective as well as our top priority. This feature is the cornerstone of our project and its success will determine the outcome of this project. In order to Achieve this Objective, we plan on having an onboard flight controller that will be in charge of the orientation and stability of the drone while also doing the computations of the input from the GPS and the external camera. Rather than detecting individual objects, our camera system will create a three-dimensional map of its environment by using computer vision as well as the inertial measurement unit on the flight controller. Using this map, the navigation system will detect if the current planned path with colliding with any physical objects on the map. If so, evasive maneuvers will be taken to navigate around the object or structure. Once the obstacle has been avoided and there is a clear path for the drone to return to its planned course, it will do so and carry on as if there was no obstacle. This system will be implemented using a Simultaneous Localization and Mapping (SLAM) algorithm to create a point cloud. This point cloud will be fed to the navigation system to work out the most efficient path around the obstacle.

Our final Objective is to have a fully functioning package delivery drone that is user-friendly to where you can plug in the correct coordinates and it will find the best path from the lift-off point to the package's destination.

2.2.2 Specs

When deciding our specifications, we wanted to keep it reasonable and affordable. In **Table 1**, the physical specifications were majorly influenced by costs because the larger the quadcopter the more expensive materials will be, but with that, we needed a sweet spot of size in order to fit the modifications. **Table 2** has our goals in terms of electrical performance by trying to maximize energy efficiency and increasing performance. **Table 3** goes into the type of sensor specifications we will need to fly the quadcopter/drone to the destination and be able to perform robot vision for the object detection system.

Table 1: Physical Specifications

Physical Size of Drone	2 to 4 feet long
Number of Motors	4
Number of propellers	4
Number of Circuit Boards	3
Number of Controllers	1
Weight of Drone (Without Batteries)	5 - 8 pounds
Weight of Batteries	1 pound
Payload Weight	0.3 - 0.5 pound

Table 2: Electrical Specifications

RF of Controller and Drone	2 - 2.4 GHz
Voltage of Drone	13 - 15V
Battery Capacity	5000 - 6000 mAH

Time of Battery to Run	30 Minutes
Amount of Voltage in 1 cell of LiPo	3.6V
Controller Voltage and Current	7V and 1A

Table 3: Sensing Specifications

Field of View	60 degrees horizontal (+- 10 Vertical)
Range of View	0.5 - 10 Feet
Measuring Frequency	10Hz
Autopilot	GPS
Number of Sensors	4 (IR, Camera, GPS, Altitude/Pressure)

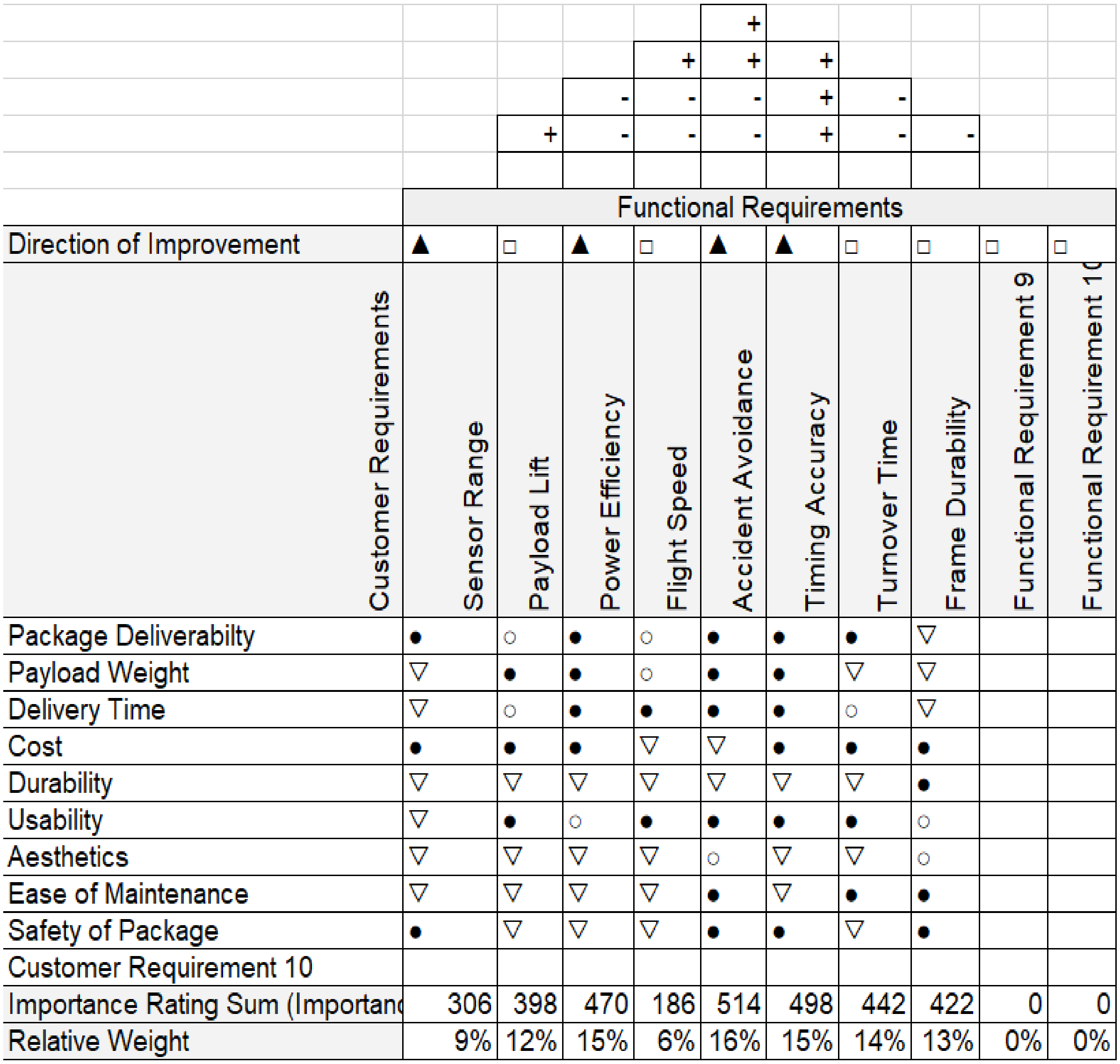
2.3 House of Quality

Here we have our House of Quality in **Table 4**, Where we show a correlation between the customer requirements and the engineering requirements. We see this using the legend given in **Table 5** to correspond what we are seeing to what the data means, which shows a strong correlation of certain subjects like Accident Avoidance in the engineering requirements to the customer requirement of actual Package Deliverability and Safety of Package. We see in the “hat” or triangle at the top of the HOQ the correlation between different engineering requirements and how they relate to one another. Such as Timing Accuracy, Turnover Time, and Accident Avoidance having strong “+” relationships.

House of quality is an important evaluation of what will be a priority to the customer and project and provide that bridge of understanding for the mission. This ensures proper communication between the engineers and the people interested in the product. To create this from the perspective of engineers it was highly brainstorming what a customer might want and be interested in for the product. We had to use empathy and put ourselves in the customer's shoes to see what we would want from the product without extremely considering the detailed design. From there we created the customer priorities on the left column. Then we thought about the technical parts of drones and the product we were building from an engineer and design point of view, which is where the titles in the top row were created. The middle consisted of thinking about the relationship between to two and visualizing the priorities ahead. These requirements were not always the same, but when the thought of together do have some correlation sometimes when considering making them happen together or how they interact.

In order to make this more accurate from a company standpoint, it would be in their best interest to research and survey customers to see their unique perspectives. There are teams typically placed on design projects like this to interact with people from outside the project to get that feedback and not create an echo chamber of ideas. These strategies will help make the product more applicable and the House of Quality is a great tool to summarize that information for a team working with designers and engineers. The benefits will trickle down into the project by cutting down costs. If you can see that one design factor is greatly prioritized and connected then the team knows to put a lot of time, though, and money into it. But for items not as greatly prioritized, it can be used to cut costs and not push the team so hard for a feature that is not as cared about as others.

Table 4: House of Quality



					+					
				+	+	+				
			-	-	-	+	-			
		+	-	-	-	+	-	-		
	Functional Requirements									
Direction of Improvement	▲	□	▲	□	▲	▲	□	□	□	□
Customer Requirements	Sensor Range	Payload Lift	Power Efficiency	Flight Speed	Accident Avoidance	Timing Accuracy	Turnover Time	Frame Durability	Functional Requirement 9	Functional Requirement 1
Package Deliverabilty	●	○	●	○	●	●	●	▽		
Payload Weight	▽	●	●	○	●	●	▽	▽		
Delivery Time	▽	○	●	●	●	●	○	▽		
Cost	●	●	●	▽	▽	●	●	●		
Durability	▽	▽	▽	▽	▽	▽	▽	●		
Usability	▽	●	○	●	●	●	●	○		
Aesthetics	▽	▽	▽	▽	○	▽	▽	○		
Ease of Maintenance	▽	▽	▽	▽	●	▽	●	●		
Safety of Package	●	▽	▽	▽	●	●	▽	●		
Customer Requirement 10										
Importance Rating Sum (Importanc	306	398	470	186	514	498	442	422	0	0
Relative Weight	9%	12%	15%	6%	16%	15%	14%	13%	0%	0%

Table 5: House of Quality Legend

Relationships		**Weight**
Strong	●	9
Medium	○	3
Weak	▽	1

Direction of Improvement		**Correlations**	**Correlations**
Maximize	▲	Positive	+
		Negative	-
Target	□	No Correlation	
Minimize	▼		

2.4 Block Diagram

Diagrams needed to represent this project would be a general flight diagram and then the software component. **Figure 3** represents the basic form of the path that the drone will fly through to deliver a package safely and accurately to a GPS point. Having this visual is a great start for what will need to go into accomplishing it. A huge project like this needs to be broken down into smaller missions or tasks that make up one product. That idea is Divide and Conquer. Completing one part of the diagram leads a path to the next. **Figure 4** shows a diagram of how the software will be performing.

The software part of a drone contains its own set of instructions and pathways to accomplish its goal. **Figure 4** goes into detail about how it will detect an object and process it in order to properly avoid it. This gets accomplished by working with the hardware of the drone, the sensors, the camera, and the program written. From there it will take in all the data and pass it through algorithms that process it to determine what needs to be done and sends those instructions of avoidance to the drone. We will be using specifically computer vision, which has the optical flow, filters the image, and gets smarter with each use. The ultrasonic sensor works to give data of where the object specifically is or how far away it would be from the drone, which helps determine how fast a decision needs to be made or how fast the program is able to make that decision.

Figure 3: Flight Path Diagram

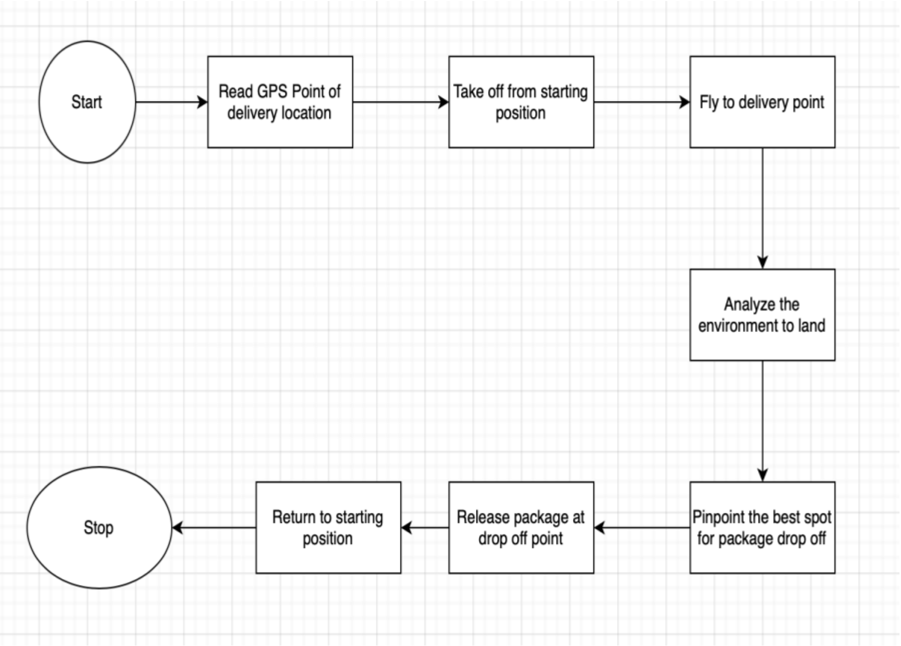
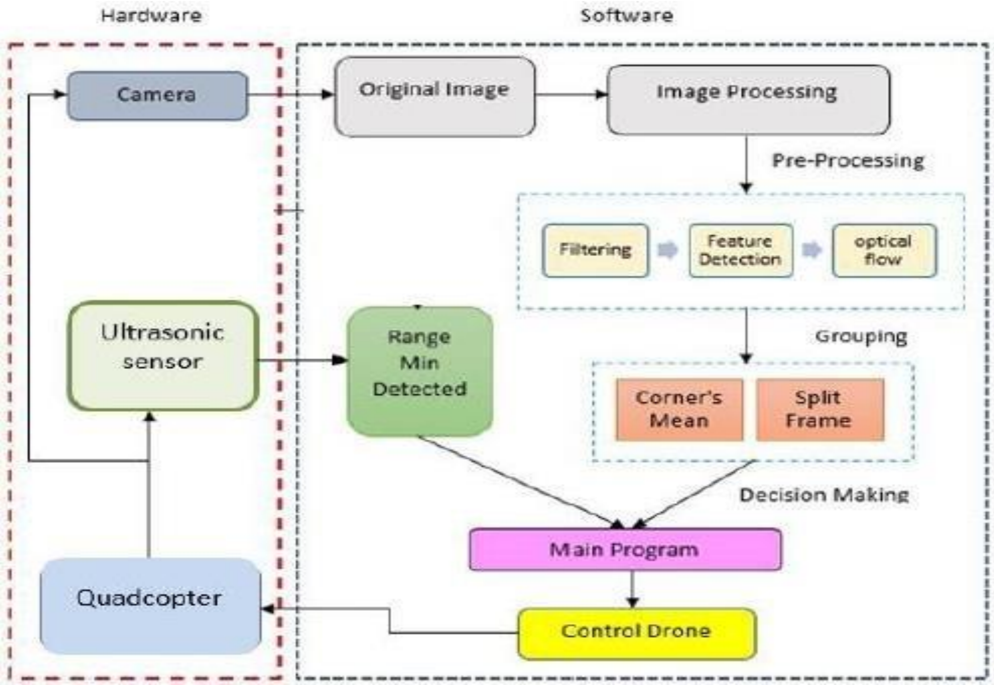


Figure 4: Software and Hardware Flow



These diagrams are a great guide into what needs to get accomplished and a visual of what is happening. There are many factors to a team besides just engineering and having things laid out in a visual guide those conversations of features and what is happening. An expanded team for a drone might have a sensor team that only works on sensors and might not need to know about the computer vision but just needs to know how it is interacting or the importance of where the data is going. Overall, very helpful for keeping the communication in line with the teams.

3. Technical Content

3.1 Existing Projects

When approaching an idea, it is important to see what has already been done and currently out there on the market. If research is not done, then repetitive creation could happen and after a bunch of hard work and hours put it then it could end up already being around. This section goes over existing projects that helped provide us with direction and get the ideas flowing for what is out there and what is still left to discover and create.

3.1.1 MatterNet: Drone Companies in the Medical Field

Figure 5:MatterNet's Drone Product

MatterNet is one of the few companies in the recent future that has stepped into the field of logistics and medicine with the help of drones. Their company started the climb to the top of their field by getting permission from the country of Sweden to provide their services of moving blood samples and other needed supplies between hospitals using drones, as shown in **Figure 5**, and increased their reputation through their innovative housing unit for the drone, which made the interaction between the drone and user simple and easy to understand. They then partnered with the US company UPS to supply logistics in the NC-based hospitals and to include themselves in US airspace. Their drone is intuitive and easy to use, with a quadcopter style-based drone with a center space designated for packages. It is continuously monitored by in-company drone pilots but is run on the autopilot and accident-avoidance system so as to remove human error and time frames. MatterNet's mission statement regularly takes their drone in city limits and flies in urban environments, showing the similarity in situations we envision our drone's mission statement to be.

3.1.2 Amazon Delivery Drone

Prime Air, a division of Amazon, is focusing on making 30-minute delivery a reality. While this requires many infrastructure changes, the key development they are using to make this possible is an autonomous delivery drone. The drone will be loaded with a package and then sent off to deliver it. The autonomous drone will be capable of taking off, navigating to the destination, selecting a safe drop-off zone, navigating back to the warehouse, and landing. It does this all without the supervision of a human.

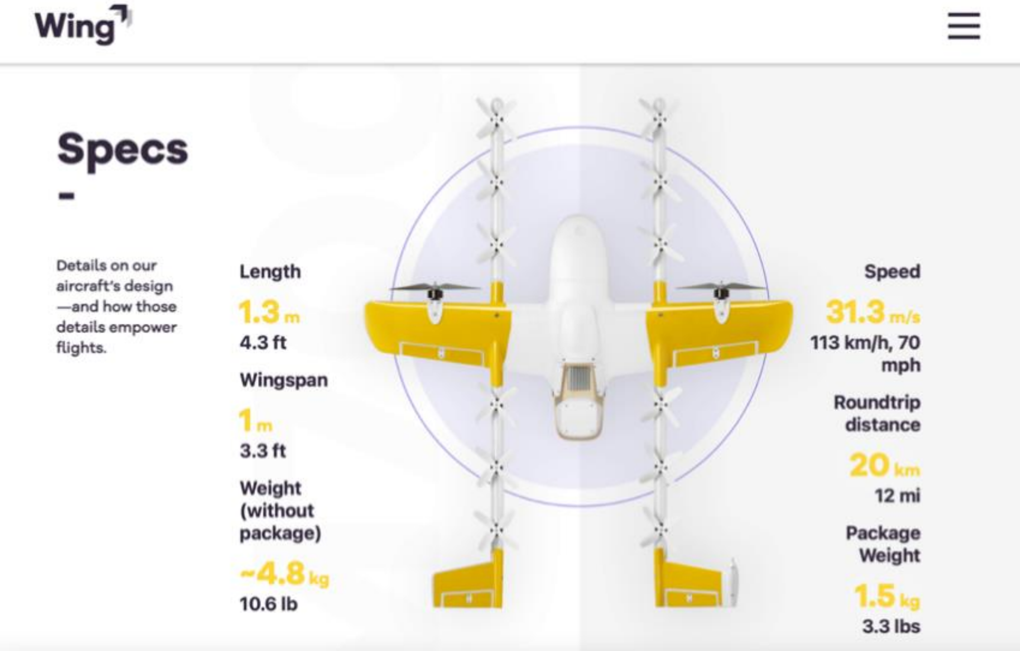
The specifications of Amazon's delivery drones are very similar to our own specifications. The drone can fly for up to 15 minutes and deliver up to 10 miles away, giving it a round trip range of 20 miles. These specifications are maintained while carrying a payload of up to 5 pounds. Additionally, the propellors are fully shrouded for safety. This shroud also acts as a fixed wing, increasing the efficiency and speed of the drone. This is a key factor in achieving the above specification requirements. It is also an area where our drone differs from Amazon's implementation. Their drone is controlled with six degrees of freedom as opposed to the standard four degrees of freedom. This helps the drone to remain stable in windy conditions. **Figure 6** shows the actual drone.

Prime Air's drone uses similar, but more advanced sensors to our own. They utilize sonar, thermal cameras, and depth cameras to sense in every direction around the drone. These sensors are utilized in the two most safety-critical stages of flight: transit and drop-off. During transit, the drone uses these sensors, combined with machine learning algorithms, to detect static and moving objects from any direction. During drop-off, the drone scans the area for hazards such as powerlines, wires, people, and animals, as well as uneven terrain. This allows the drones to make decisions in real-time to keep themselves and the people around them safe.

Figure 6: Amazon Delivery Drone

3.1.3 Google/Alphabet Wing X

Figure 7:Wing Prototype 1



Wing

Specs

Details on our aircraft's design—and how those details empower flights.

Length
1.3 m
4.3 ft

Wingspan
1 m
3.3 ft

Weight (without package)
~4.8 kg
10.6 lb

Speed
31.3 m/s
113 km/h, 70 mph

Roundtrip distance
20 km
12 mi

Package Weight
1.5 kg
3.3 lbs

Starting in 2019, Wing, a drone carrier project under Google and Google's parent company Alphabet, received Air Carrier Certification from the FAA. This drone has been testing and going through prototype models for a few years prior and doing test flights in Australia, Finland, and the U.S. **Figure 7** below summarizes a lot of the performance specs that come with the approved Wing carrier drone. Some of the design choices seen that make for interesting consideration are the multiple smaller propellers in parallel of each other that help it have more hover control. The drone also uses tether technology in order to hold packages [1]. Researching more of the technology that is included with the success of this drone, showed similarities to our design choice. Both our drone and the Wing include the use of LiPo batteries, Wi-Fi, GPS, and use of a camera. The LiPo batteries are the best choice for maximum energy and power output in the industry right now for projects of this caliber. Wi-Fi and GPS are used to communicate with the drone and pilot system to keep track of where the drone is, where it needs to go and alternative routes in the case of avoiding objects or things in the way. The camera is used for computer vision to give the drone the most accurate readings of the surroundings and its delivery spot and that is exactly what we will try to accomplish by using a camera on ours for its computer vision capabilities [2].

The drone did not always have this structure, if anything, it almost looked like the complete opposite. In **Figure 7**, the length is longer than the wingspan, but in earlier prototypes, it had a longer wingspan than the length size, this can be seen in **Figure 8**. This is an interesting observation and brings up curiosities on how this affected the power output, consumption, and accuracy of the drone. The drone also capitalizes on modern technology and investments to help make it a zero-emissions drone. In comparison, our drone structure will have more of a square shape overall and the body of the drone is built in the Z-axis of sorts. We are hoping this will give it more balance once components are being added to the drone and keep the center of gravity in one compacted area [1].

Figure 8: Wing Prototype 6 and 7

The technology included is not completely public but speculated to be a Python and C++-based back-end system for the traffic management system and then the front end includes JavaScript. This being a Google drone is able to implement their advanced GPS and mapping systems that are being used in much other software. Seeing this fully built and operational drone using the same software's that we will utilize is reassuring that we are headed in a great direction. Our drone will also be using Python and C++ in the firmware and software for controlling. These are dependable and widely available resources for computing drone projects with those languages. Python provides a wide variety of computer vision and machine learning libraries and resources that are applicable to drones. C++ is an embedded systems software to easily communicate with hardware, firmware, and software at once for the controllers and such [2].

3.1.4 Tesla's Autopilot

Tesla has one of the best and most competitive auto pilot cars on the market as of current within the automobile industry. They are currently capable of object avoidance as well as self-driving to the owner of the car through a parking lot to the most convenient location for them. These features are similar to what will be implemented on our drones such as collision avoidance and auto pilot to a destination.

The Tesla automobiles currently work through a Central control computer. This computer controls the acceleration of the car depending on the objects it detects. The object avoidance that Tesla implements are far more sophisticated than what we will have to implement in our project but how Tesla implement and anticipate what objects will do will be very useful to how we wish our drone to behave when confronted with the obstacles.

Tesla automobiles are equipped with numerous cameras that are capable of processing live images and interpreting hundreds of objects simultaneously. With this implemented in Tesla cars are capable of detecting traffic lights, road debris, and other hazards. We anticipate our drone to be capable of detecting a fewer number of objects but hopefully have the same outcome of obstacle avoidance the Tesla has accomplished [45].

We also wish our drone to be capable of finding its way to a certain location. Tesla has implemented this in their cars through their key fob. Drivers are able to click on the key fob and the car was able to go into auto pilot and drive to the most convenient location for the driver. With package delivery in mind for our drone, we hope to have a similar outcome [45].

Due to budget, time, and processing limitations, we are not expecting our drone to be able to detect all of the objects that Tesla cars are able to however we do expect to be able to detect a few objects simultaneously and accelerate and navigate accordingly. We see Tesla automobiles as inspiration on how it is currently revolutionizing how cars function in our society and how well designed its autopilot currently is.

3.2 Details Related to Project and Part Selection

3.2.1 Frame

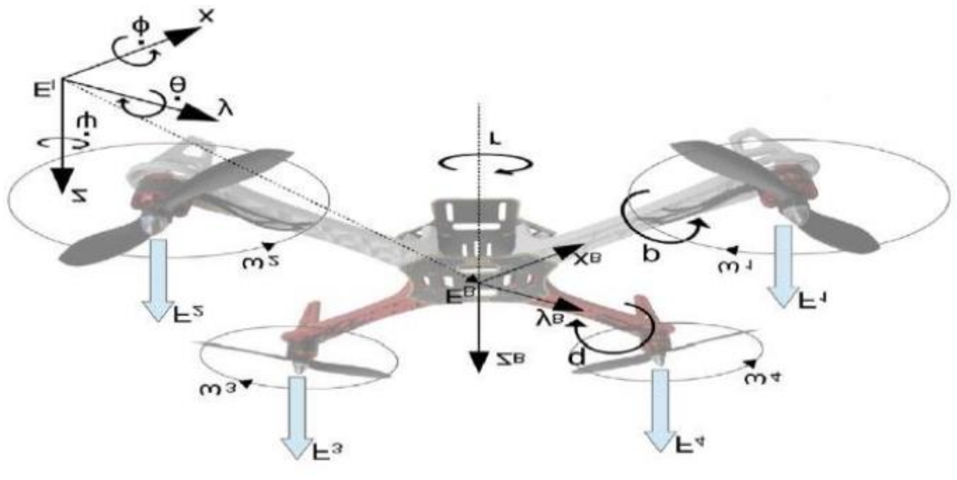
Figure 9: The Readytosky 149HD

The frame is the foundation of our drone project. So, making sure to pick a drone Frame that would be able to support all our needs is critical. We then narrowed it down to 4 drones to pick from. The first is going to be The Readytosky 149HD. Some pros and cons about The Readytosky 149HD as seen in **Figure 9**. The Readytosky 149HD is a great drone from its reviews and we liked the idea of having guards for the blades because safety is always a major concern. The cons being that we might not have enough building area on the drone to accomplish our goal because the drone is only 149 mm large. Also, with being that small we were also unsure if we would be able to get the correct amount of lift needed to deliver and carry our package.

The Readytosky S500 is the next drone we took at as seen in **Figure 11**. The pro of this drone is that it is 450mm and we believe it would be large enough for any design or configuration of sensors we need to put on it. Also, it has landing gear which we believe to be important for 2 reasons, the first being that it gives us area under the drone to build from and has room for our package. The second reason is that with landing and taking off the thing our drone is going to be hitting the most will be the ground so having landing should improve the longevity of the drone. The con to this drone is that we don't know how stable the landing gear would be on uneven terrain, as well as not having guards for the blades.

The FPVKing 500-X4 is the 3rd drone we looked at as seen in **Figure 12**. The FPVKing 500-X4 is the largest drone we looked at being 500mm in size. The pros to this drone are not only the size that would give us the building freedom we need but also the arms of it being able to fold down or come off which would be handy in transport as well as any un foresee maintenance that might need to be done to the drone during testing. The cons for this drone are similar to the ones found in The Readytosky S500 with the same concerns about the landing gear being unstable on uneven ground. If we look into **Figure 10**, we see the physics that we need to apply and look for when looking into frames.

Figure 10: Physics Looking into Drone Frame



Frames will be the foundation of our project because when trying to divide and conquer the decision of parts it was quickly seen that making a decision on one part led to another one. This put a stop to trying to make a lot of decisions at once and stopping to evaluate more the decisions being made. All of these frames were leading to what type of motor we would eventually pick and other components because a heavier frame means having a more powerful motor that can power it. It was important for our frames to have a landing gear attached to it whether we were going to have to attach our own landing gear or look for frames with it already attached. Some pros to having the landing gear attached means that we would not have to buy others or make the landing gear, then there is the issue of securely attaching the landing gear. This would be a con when approaching **Figure 8**, but then the cons with having landing gear is that if it were to break, we might have to replace the whole frame.

Replacing the whole frame would not be very cost efficient to the project and means more testing would have to happen in order to make sure the drone would not crash and break the landing gear. A delivery drone specifically also needs to be able to handle the weight of the components and the potential packages. Smaller frames might not be able to handle all of that and lead to looking at frames that match a certain size versus a certain shape.

Figure 11: The Readytosky S500

Figure 12: The FPVKing 500-X4

The YoungRC F450 is the last drone we looked at and is as seen in **Figure 12**. The YoungRC F450 is 450mm in size and so we believe it would give us the building freedom we need. We also think the landing gear on the YoungRC F450 would be better suited for multiple terrains while still giving the space underneath for creative freedom. The only cons for this drone are that it doesn't look to have the largest amount of space in the center as much as the other drones offer as well as not having blade guards.

In **Table 6** below, we compare the drones in a side-to-side manner and look at the price differences as well materials used in the frame. As we compared the drone’s side by side in the table above we quickly found that The YoungRC F450 is the most cost-effective drone. We then also come to the conclusion that we would like our drone to be made out of plastic rather than carbon fiber for two reasons, the first being that carbon fiber is heavier than plastic which will hold us back from the maximum amount of lift and weight of the package. The second reason being that is a part where to break or chip in testing there would be a very limited number of repairs, we would be able to do because when carbon fiber breaks it does break cleanly, it splinters out making it impossible to repair. So as to reiterate, this drone has the highest potential not only because of its size and performance in certain sectors that can affect our testing and eventual completion of our project, but also the direct cost to the project budget.

Figure 13: The YoungRC F450

From all this research we were able to pick The YoungRC F450 in **Figure 13** as our drone frame for the following reasons. First, it is 450mm which we believe is large enough to carry a package while giving us the freedom to place the sensors as we see them. Seconded is that it is made out of plastic and is lighter than the other drone frames and will be easier to repair if damages arise. The third is the landing gear being able to withstand a wider range of terrain while also giving us the needed space underneath to build as we need for our sensors, battery, and package placement. The fifth reason is the cost being more than half the cost of the other drone frames. This series of reasons are why we believe The YoungRC F450 will be the best drone frame and makes up for not having guards for the blades and will still be the foundation on which we build upon.

Table 6: Drones Frames

Names	Cost	Size	Materials	Landing gear
Readytosky 149HD	$46.00	149mm	Carbon Fiber	No
Readytosky S500	$47.00	450mm	Plastic/Carbon Fiber	Yes
FPVKing 500-X4	$75.00	500mm	Carbon Fiber	Yes

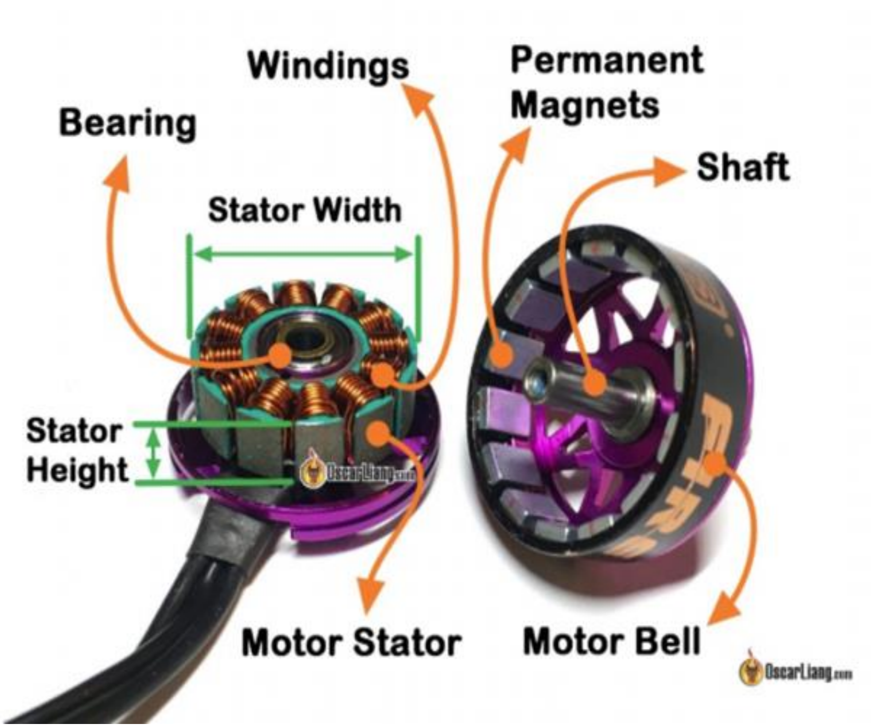
YoungRC F450	$21.00	450mm	Plastic	Yes

3.2.2 Motors and Propellers: Stators and their relation to Lift

Our motors are vital aspects of our drone and need to be calibrated to our specific needs. However, when researching motors, we found that it very much is a chicken and egg scenario where we need to also discover the size of our propellers and base that on stator size.

We know our frame size to be at least 450mm wide with wingspan included, so when we take that into account, we can begin to estimate what other properties of our drone will come to. With Guidance from **Table 7** from DIY makers of drones, and the physical picture we have of the inner working of a drone motor from **Figure 14** we can correspond to what we need for our motor needs and begin the tiresome process of achieving the right motor for the right kind of lift and mission we will need to acquire the motor for. Motors need to account for the numeric and specifications that come from the frame, which shows how much motor power is needed.

Figure 14: Inner Components of Motors



Motors will lead to more decisions being made because the ESCs correspond to the motor power also. The table goes into how when the frame size changes the motor power and such also increases or decreases according to size. So, picking the right one will be important to help the drone take off with enough burst power and stay steady. To show this we can look below at the many different instances and kinds of motors we will need to consider for our future drone and possibly any future adjustments that will need to be done to our motor's specifications because of future changes to actual environmental factors that we cannot perfectly see now and will not be able to account or until such a time that we have all other components that our motors will have to lift so as to perfectly know environmental factors.

Table 7: Propellor and Motor Size Comparison

Frame Size	Prop Size	Motor Size	KV
150mm or smaller	3" or smaller	1306 or smaller	3000KV or higher
180mm	4"	1806	23600KV
210mm	5"	2204-2206	2300KV-2600KV
250mm	6"	2204-2208	2000KV-2300KV
350mm	7"	2208	1600KV
450mm	8",9",10"	2212 or bigger	1000KV or lower

We see from this that since our frame size is at the very edge of the Graph (450mm) we will need to supply propellers of 8" or higher on to our motors, with a stator size of 2212 or higher. The number 2212 is the technical language used by most drone manufactures when discussing motor and stator size. The number 2212 is broken into two sections that are described with the first two digits being the stator width, while the last two digits are the stator height. The higher these two numbers are meaning the more torque one motor can supply. We decided after comparing the cost to the amount of material we were buying what motor to get, shown in comparison in **Table 8**:

We eventually picked the Qwinout motors as the proper motors to use, not just because of the cost for only the motors, but because of the package, we have where we can buy both the ECS, propellors and the frame, thus keeping cost down, even when the weight of the Qwinout motors is more than all other motors of the comparison.

Table 8: Comparison of Different Motors

Name	Current	Weight	Price
Powerday	4-10 Amps	47g	$32.69
Goolsky	4-10 Amps	48g	$49.99
Readystosky	4-10 Amps	190g	$36.99
Qwinout	4-10 Amps	249g	$26.88

3.2.3 Batteries and Power Distribution Boards:

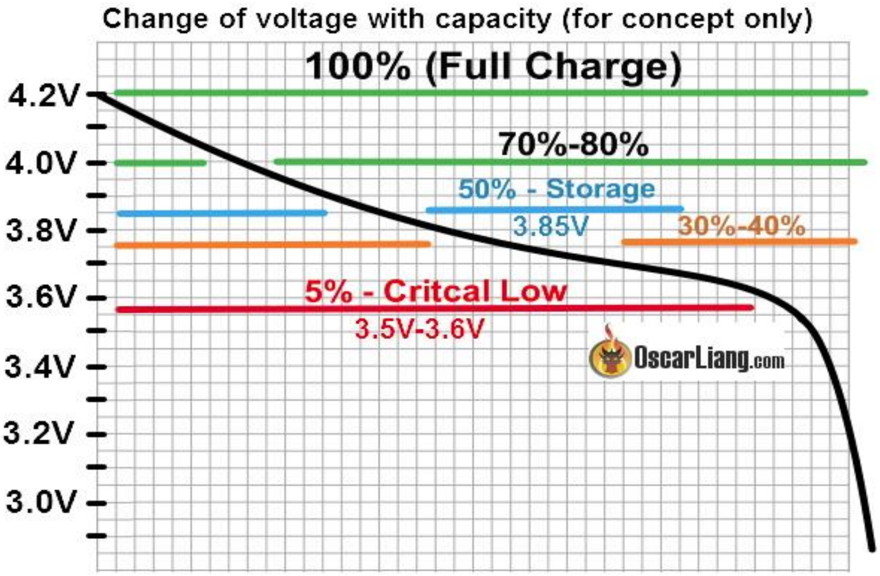
We know from our research into drones that almost all drones fly using the Lithium Polymer battery pack. Because it has a higher energy density and lighter weight than other battery types. However, we also need to decide on what type of battery level we will be used as the battery is again dependent on what types of components it needs to be able to power. Namely how much current the motor and ESC will be using during maximum operation. Each cell of a Lithium Polymer battery includes 3.7 Volts and is said to be 1S, with 2S being 7.4 Volts, etc... However, if we see in **Figure 15**, this is actually the average usually given for that specific cell with the top of the fully charged battery is around 4.2V. Now, if we chose an ECS that drew 15 Amps for operation, and know the amount of time we want our drone to run per battery pack, then we take the C rate and battery capacity and use this equation in **Table 9** to find the numbers needed:

Table 9: C-Rating Equation

Max Current Draw = Capacity * C-Rating

So, taking the current draw of 15 amps and times it by the amount of time required we get 15 times (10 divided by 60 to account for minutes in an hour) and that would give us 2.5 Amps per hour. This is the amount the battery needs to supply. Using this number, we can then find the C-Rating and Discover the battery that will fit our needs. Now to also go over the storage of this battery type, it must be more well maintained for the battery to be usable, as shown in **Figure 14**, the storage charge that most LiPo batteries must be at is around 50% to safely be stored for further use in the field or for travel. We also know that from our mission of avoiding obstacles and delivering a package to a destination, we need enough battery life for this initial model of the drone to last at a minimum of 10 minutes. Now if we take the max current draw of our ESC that we have decided upon, namely the 2-4S 30A Brushless ESC, and take our known timeline we need, ten minutes, and divide those ten minutes by 60 mins, we get 0.167 hours, which is the time we need. We're going to convert this is mAh, however, so we must take the max current draw, which is 40 amps when under “burst” conditions, which is when the drone is doing many different things and needs to move the motors fast, and then times that by our 0.167 Hours. We, of course, convert this to milliamps but afterward, we achieve our mAh we need to know. We see we achieve a mAh’s of 6680 mAh. Further, we can now find our needed C rating so we may find the battery we need for our timeline and drone type.

Figure 15: Concept of Charge Levels of Cells and Storage Levels



C rating can also be said to be the charge rate for the batteries, or how long it will take for them to discharge or charge, with 1C being equal to 1 hour or 60 mins. 2C being equal to 0.5 Hours, or 30 mins and so on and so forth. If we want a more efficient testing period, in other words, if we don't want to have to buy numerous batteries to also have them available for testing, we need to have at least a C rating of 1C, with a more robust route would be to have 2C for a 30 min charge rate as shown in **Table 10:**

Table 10: Difference in C-Rating

C Rate	Time
2C	30 mins
1C	1 hour
0.5C	2 hours
0.1C	10 hours

Now when comparing real-world batteries and discovering what we need and what can be affordable, we will use **Table 11** to compare different batteries and their prices.

Table 11: Different LiPo Batteries on Amazon

Brand Name	Voltage level	C-Rating	Capacity	Case Style	Weight	Price
SIGP x1	11.1V	25C	2250 mAh	Hard Case	180g	$12.60
Turnigy x1	11.1V	20C	5000 mAh	Hard Case	360g	$24.72
Zeee x1	11.1V	50C	5200 mAh	Soft Case	334.7g	$33.99
Zeee x2	7.4V	80C	6000 mAh	Hard Case	309g	$49.99
HRB x2	11.1V	55C	7000 mAh	Hard Case	449g	$103.99
Zeee x1	11.1V	100C	8000 mAh	Hard Case	466g	$68.99

We see from comparing our different categories that in between these 5 different batteries, the ones with our needs are only the 7000 mAh HRB and the 8000 mAh Zeee. We know that we do not need above 8000 mAh for our operation time however if the price range is acceptable, it is still within our market as this would just give us more operation time overall. However, we see that for the price given for the HRB we get 2 battery packs, which cannot be beaten by Zeee even in our price range, showing that we will be giving very much consideration to the pros and cons of higher mAh or cheaper price.

Now for PDB or Power Distribution Boards, they can range from the most basic where you have what amounts to copper plating that integrates the power from the battery to the components like the ESC or the flight controller. This will not work for us our sensor package will require a power module converter as shown in the basic PDB in **Figure 16**:

This shows how the basics of the PDB can be simplified immensely. To continue to our needs for this project, however, we will be including voltage regulators in the sections flowing to different outputs in the PDB so we may control how much voltage goes where, as the sensor package will need a certain voltage with much “noise” and the ECS might need the voltage regulators to quickly increase power intake when sudden accelerations or decelerations happen.

For when it came to designing or PDB otherwise known as power distribution board we had a couple of options in terms of software to design on. The software being Eagle or Kicad. We chose Eagle over kiCad for a few reasons. We believe Eagle to be more widely supported if we had to ask questions when finding complications in our design. We have also already taken a class with Eagle and have some level of familiarity. Eagle has an iOS version as well windows version so the program would be accessible to everyone in our group. While kiCad did seem easy and intuitive the fact that it was not assessable where IOS (Apple devices) made it unusable. The software we decided to design our power distribution board on is Eagle as you can see this information more clearly in the graph below. We must verify and be very insistent with the specifications of the PCB software we will need to use and implement in our design as this can affect all systems within our design, as power must run into every component and without efficient flow, we will have catastrophic effects such as crashing or missing data points that could lead to our drone missing instances if objects coming towards it. Which will lead to catastrophic crashes and maybe the destruction of the entire drone.

Figure 16: Basic PDB Module with Converter between Battery and PDB

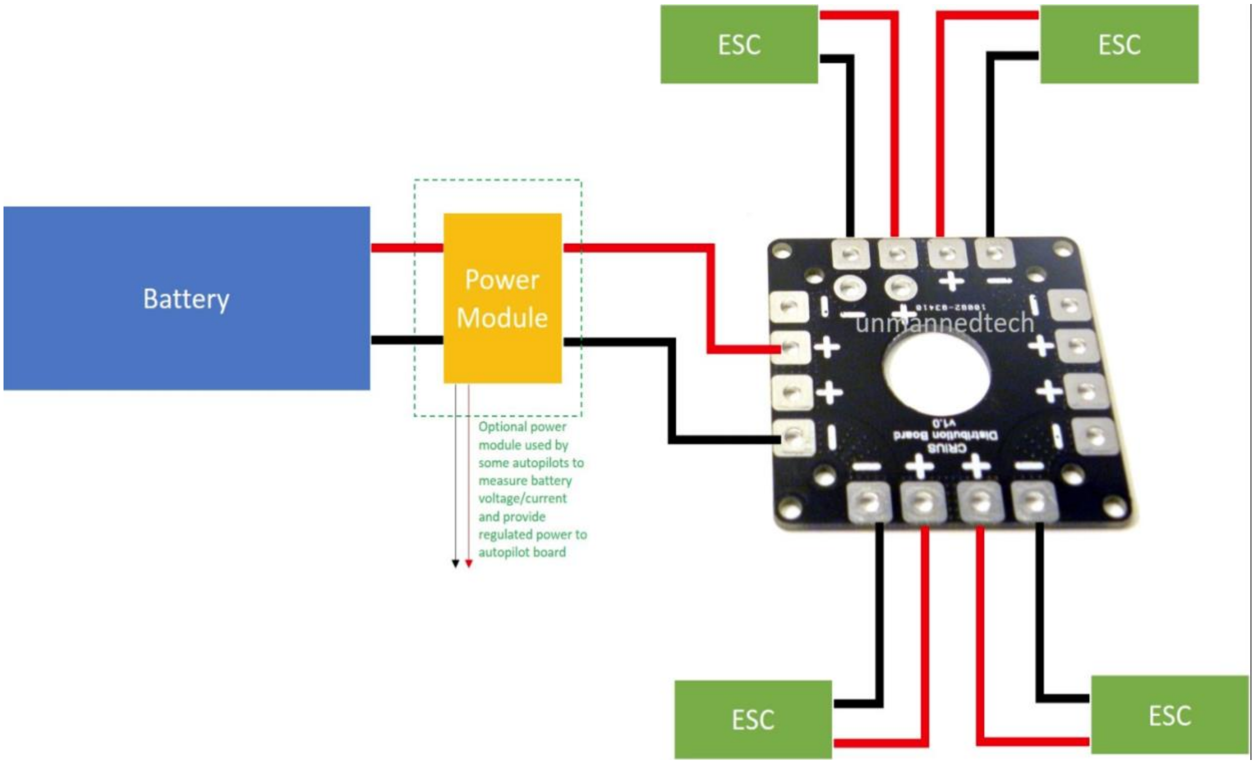


Table 12: PCB Software Design Comparison

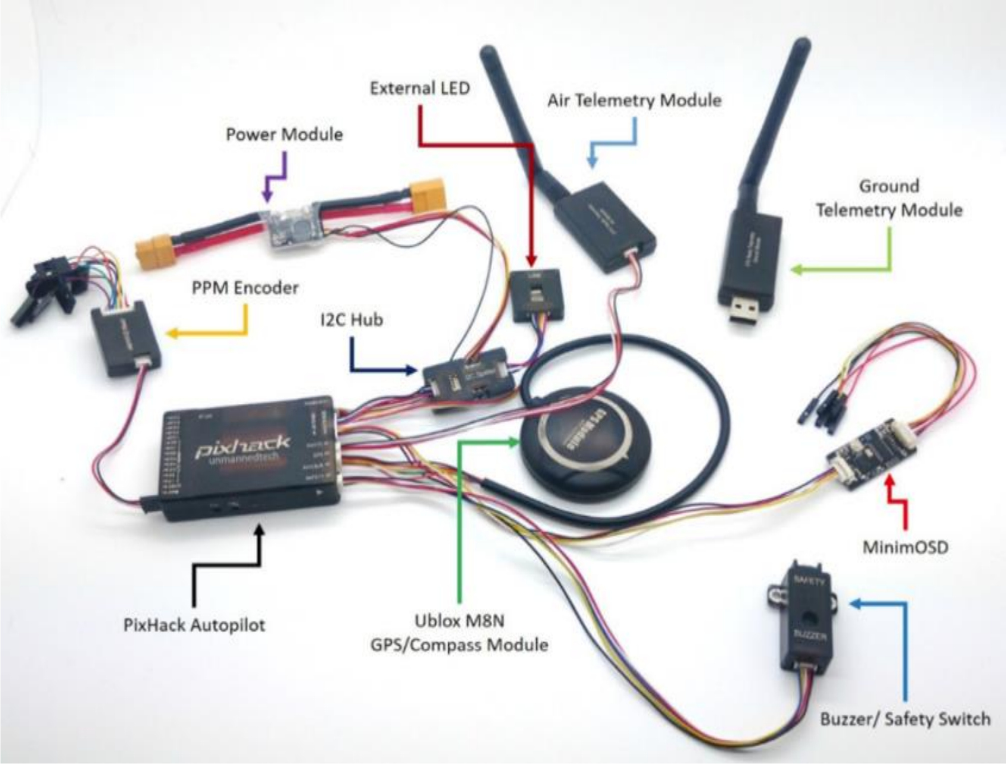
PCB Software			
	IOS	Window	User-friendly
KiCad	FALSE	TRUE	TRUE
Eagle	TRUE	TRUE	TRUE

Now the mission of the PDB is to distribute the power from the battery safely and effectively. The loads the battery will need to power directly will be our ECS and our flight controller. Our ECS being a 2-4S Brushless ESC and being rated toward 3 Amps. The flight controller is the other important Device the PDB will have to deliver power to safety and effectiveness. The flight controller is the brain of the drone and is the most expensive part. So, making sure we deliver the correct number of voltages needed is critical. The voltage needed to power the flight control is 5 volts. To make sure we do this, we first compare the two in **Table 12** we will use the TI web bench and upload the correct Library into Eagle that will allow the correct voltage output. While dealing with our limited space on the drone for placement and have the correct milled out hole to fasten our power distribution board to the drone frame.

The next part of our PDB design will be to have it manufactured and ship to the United States. There are many companies that will manufacture such a board and we don't expect this to have a strong impact on our budget. However, the limitations on global trade in this current climate might after shipping times and prices which could possibly affect where we purchase our manufactured board.

3.2.4 Autopilot systems and their relation to GPS systems:

Figure 17: Components of a Basic Autopilot System



For Autopilots, they included many components that we will be going over such as shown in **Figure 17**, to include the basic components like the telemetry system, the power module that regulates power for the autopilot system, and the compass module. This system is made to be in parallel with a manual system that all drones come with and the autopilot system itself is controlled from a "ground station" that is usually a computer with the ground telemetry system attached to it so the Drone and the Ground station can talk to each other. Now for our instance of use in the term for ground control, we will most likely be relying entirely upon a program that will act faster than anything we can possibly do with human error and reflexes.

The Autopilot runs on a processor that runs separate front the flight controller and is usually a 32-bit processor although there are 8-bit ones, we will be using Raspberry Pi and Navio for our purposes which we know will handle the processing requirements of our specific style of Autopilot. We will be using many different sensor systems that we will include in our drone such as the barometer to measure pressure, while drones usually come with accelerometers and gyroscopes, and the autopilot system itself will need a compass module and a GPS module that must be far enough away from the drone so as to not interfere with the electronics of the boards. Now, these sensors are known in drone manufacturing as DOF or "Degrees of Freedom". With more DOF you get more sensors included in your drone. While we will be buying drones that do not come with many sensors, we will be adding many sensors to artificially raise our drone's DOF. To use GPS modules as we will talk about later in the section, a drone needs at least 10 DOF. To continue onto our telemetry modules, these speak to each other through RF frequencies, and both the drone and ground modules must be the same MHz to work properly. From there we again talk about our power module converter, which is a module that is in between the lines from the battery to the Power distribution board so there will be less "noise" in the power, as we will need that for when we work with GPS signal and sensor modules. Most power module converters put out a stable 5V from the battery. This also lets the GPS take over for flight control when the battery gets low as with the ability to acquire more power with "clean" signal strength, the GPS can tell the ground station when it is low and will return to base. Finally, there will be instances of loss of signal for the GPS system as this is inevitable when going through the thick bush or other obstacles. This is where the optical flow mechanism will come in, where we use machine learning and computer vision that will be talked about later in this paper to "drive" the drone when needed.

3.2.5 Flight Control Boards

The flight control board is the single component that integrates all other individual components into a coherent system. Its overall purpose is to take in inputs from the user, a predefined waypoint mission, and various sensors in order to translate these signals into the proper motor speeds to achieve stable flight. This control board is heavily coupled with the flight control firmware. Together, these two elements determine the capabilities of the drone, which can vary wildly depending on the feature set of the board-firmware pair. On the low end, the control system only tries to hold the drone level and varies the motor speeds according to user input for steering. On the high end, the control system can autonomously fly the drone using GPS data and waypoint missions. Some can even perform computer vision processing. Because our application requires autonomous flight, the high-end flight control boards are more appropriate. Given that this is the keystone component of our system, it is crucial that we select the optimal control system for our application. **Table 13** compares flight control boards and displays the features we considered when deciding on one. We see the weight difference is extreme between different boards, but the firmware is incredibly similar systems to each other. Other things that are similar are Bus width, with many being a 32-bit variety because of the influence of cost that of course comes with boards that are trying to achieve the highest processing power with such small sizes.

Table 13: Flight Control Board General Features

Board	Price ($)	Weight (g)	Firmware	OS	Popularity	Bus Width
Pixhawk	80	60	ArduPilot, PX4	No	High	32 Bit
Pixhawk Cube	250	150	ArduPilot, PX4	No	High	32 Bit
Navio2 + Raspberry Pi 4	168 + 35/55/75	23	ArduPilot, PX4	Linux	Medium	64 Bit
BeagleBone Blue	80		ArduPilot, PX4	Linux	Low	32 Bit

Table 14: Flight Control Board Hardware Features

Board	Processor	RAM	Ports	Sensors	Networking
Pixhawk 1	180 MHz ARM® Cortex® M4 with single-precision FPU	256 KB	I2C, CAN, ADC, UART, CONSOLE, PWM, PWM/GPIO/PWM input, S.BUS/PPM/SPEKTRUM, S.BUS out, microUSB	gyroscope, accelerometer/ magnetometer, accelerometer/gyroscope, barometer	wifi, DSM-X
Pixhawk Cube	180 MHz ARM® Cortex®	256 KB	Same as Pixhawk 1 but more UART ports +	gyroscope, accelerometer/ magnetometer,	wifi, DSM-X

	M4 with single-precision FPU		UART HW flow control	accelerometer/gyroscope, barometer	
Navio2 + Raspberry Pi 4	Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz	2GB, 4GB or 8GB	UART, I2C, ADC, 12 PWM servo outputs, PPM/S.Bus input, HDMI, MIPI display port, MIPI camera port, microSD, USB3.0, USB 2.0	Accelerometers, gyroscopes, and magnetometers, Barometer	2.4GHz WiFi, Bluetooth, BLE
BeagleBone Blue	Octavo Systems OSD3358 1GHz ARM Cortex-A8	512 MB	4x UART, 2-cell LiPo, 2x SPI, I2C, 4x A/D converter, CAN bus (w/ PHY), 8x 6V servo motor, 4x DC motor, 4x quadrature encoder, USB 2.0 480Mbps Host/Client Port, USB 2.0 Host Port	9 axis IMU (access, gyros, magnetometer), barometer, thermometer	2.4GHz WiFi, Bluetooth, BLE

Our first key consideration is the available flight control software supported by a particular board. Because the control firmware and control board are so interconnected, this feature will have the most significant impact on the flight control board we decide to use. This firmware will be discussed in a later section, but for now, it is sufficient to know that we narrowed our options down to between PX4 and Ardupilot. The weight of these boards is very similar and only accounts for a small percentage of the overall drone, so that is not a big concern. All of the boards discussed in **Table 14** have autonomous waypoint mission capabilities, which is a requirement for us. Our next biggest consideration is the supported operating systems. Many of the options do not even run operating systems; the flight control firmware runs on bare metal. However, the BeagleBone Blue and Navio2 both support Linux. This can be extremely helpful because it would allow us to run external applications in addition to the control firmware such as the computer vision software. This would be beneficial but not strictly necessary. Another consideration is the popularity and community around each of these boards. Having a large, active community developing solutions on a particular board not only validates its merit but also provides resources for us to learn from and ask questions during the development process. Additionally, it is important to appraise the processor on the board. Given that the options we considered were all supported by the firmware we chose, we know they will be powerful enough to run that. However, having extra power could allow us to run other software like the computer vision algorithm or sensor fusion programs. A final consideration for us was the price. Keeping the overall cost of the system low is a key functional requirement for our project. This board is one of the single most expensive components and as such, price is important to consider.

3.2.5.1 Pixhawk

The Pixhawk is a very capable flight control board. It supports both PX4 and Ardupilot. This will give us options later on when deciding the flight control software. It does not run an OS which will force us to use an external processing board to run the computer vision algorithm and translate the information it gathers into inputs the Pixhawk can understand. With that being said, its popularity is very high, and it has an active community which could be very helpful if we run into problems with hardware integration. The Pixhawk has a relatively powerful ARM Cortex M4 processor and 256 KB. This computing power is useful in processing sensor data such as ultrasonic sensors and lidar. While we would need an external computer vision board as stated previously, splitting the sensor processing between them would be beneficial to balance the workload. As seen in **Figure 18,** the variety of ports on this board give us options for connecting to various external sensors as well as other processing boards. One key benefit of this board is in the networking capabilities. While all of the Flight Control Boards do have WiFi capabilities, this particular one supports DSM-X. This is a protocol that allows communication from a radio control transmitter on the ground to the drone itself. This is not strictly necessary because our drone will be flying autonomously, but it is a good safety precaution if the drone starts behaving unsafely and looks like it will crash. Having the ability for a human operator to be able to take over control could not only save the drone from damage but could also save team members from injury. In terms of price, this is one of the cheapest options.

Figure 18: Pixhawk 2.0

3.2.5.2 Pixhawk Cube

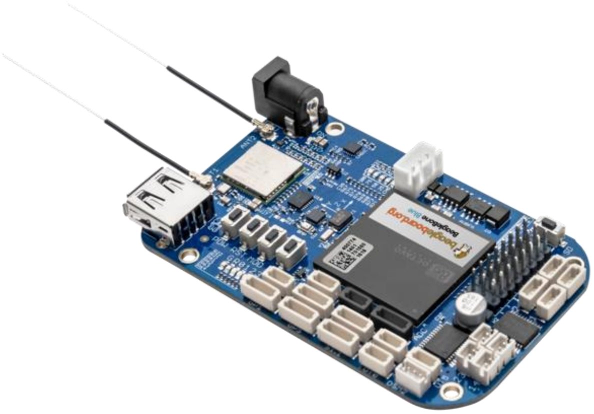
The Pixhawk Cube is extremely similar to the original Pixhawk. It also supports ArduPilot and PX4 for flight control firmware with no underlying OS. It has the exact same processor, memory, and integrated sensor combination. Again, the benefit of having DSM-X capabilities applies. One of the only differences in hardware is that it has more UART ports, each with higher bandwidth, as seen in **Figure 19**. The biggest difference between the prices. The Cube is $250 as opposed to $80 for the Pixhawk. The extra ports do not seem to be worth the huge price increase, so for that reason, we have decided that the Cube is not the best board for our use case.

Figure 19: Pixhawk Cube

3.2.5.3 BeagleBone Blue

The BeagleBone Blue is different from the previous two boards in that it runs a Linux-based operating system. This will allow us to run other programs concurrently with the flight control firmware. Both PX4 and Ardupilot are capable of running on top of the Linux kernel. The processors on this board are significantly more powerful than both versions of the Pixhawk. In addition to the single-core ARM Cortex A8, there are two Programmable Real-time Units (PRUs). These PRUs handle the I/O signals for interfacing with peripherals. This takes the load off of the main processor, freeing it up for user applications. While this processor is probably fast enough for sensor processing and light computer vision applications. However, it may limit us in terms of the number of sensors we could process and the speed at which the computer vision runs. This board has 512 MB of memory which will be useful when running heavier applications like the ones mentioned previously. This board has all of the sensors needed to achieve stable flight, including accelerometers and gyroscopes. Additionally, it has a magnetometer to maintain heading, a barometer to detect relative altitude, and a thermometer. The BeagleBone Blue supports Wifi, Bluetooth, and BLE. Unfortunately, it does not support DSM-X which means we would be unable to use a human-controlled transmitter to intervene in the case of an emergency. Bluetooth or WiFi could be used to send signals to the drone during flight, but it is not a good recovery option in the event of an imminent crash. As seen in **Figure 20,** this board has an extensive set of I/O ports such as 4 UART, 2 SPI, I2C, and many PWM outputs to control servos and motors. It also has USB 2.0 for connecting to the ground station or an external board. This may become useful if we determine that we need more processing power. At $80, this board is a very powerful option. It is the same price as the Pixhawk, but it runs Linux and has a significantly more powerful processor. The only thing that is missing is the DSM-X receiver, but this could be easily purchased and integrated separately from the control board.

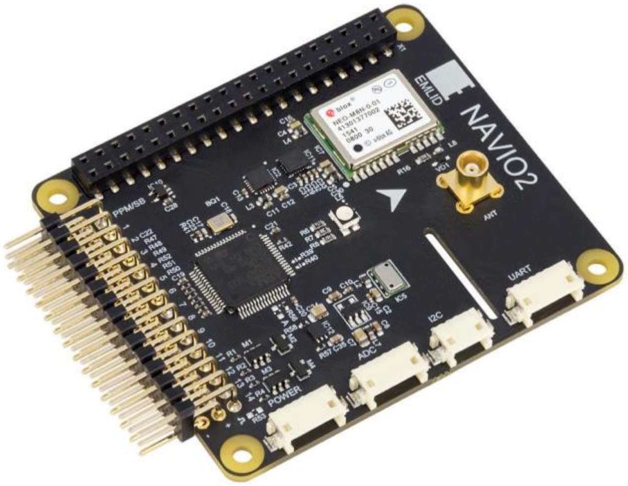
Figure 20: BeagleBone Blue



3.2.5.4 Navio2

As mentioned previously, more demanding programs such as sensor fusion and certain applications of computer vision may require extra processing power in the form of an external board. The processing board would be required to communicate with the flight control board in order to get sensor data as well as to send control inputs. The Navio2 is probably the most unique board in this respect because it is not a fully functioning standalone board. It is a daughterboard intended to be attached to a Raspberry Pi. This option drastically simplifies the connection between the control board and processing board. The board has connections that fit directly onto the GPIO pins of the Raspberry Pi, as seen on the top left side of **Figure 21.** This would also increase the capabilities of our drone. The Navio2 itself contains all of the features you would expect from a flight control board that you don't get from the Raspberry Pi itself. This includes the onboard sensors (accelerometers, gyroscopes, a magnetometer, and a barometer). It also has an M3 coprocessor used to process PWM signals from the sensors. As with the BeagleBone Blue, this frees up the main processor, in this case, the Raspberry Pi, for more computationally intensive tasks. The difference is that the Raspberry Pi's processor is significantly more powerful than the BeagleBone's. It is a quad-core rather than a single core, 64-bit versus 32-bit, and runs at a higher clock rate. Additionally, the Raspberry Pi can have between 2 GB and 8GB of memory which is particularly necessary for imaging processing applications like robot vision. In terms of I/O, this is one of the better options. Six of the Raspberry Pi's GPIO pins are left unused by the Navio2. This gives us an additional 6 UART, SPI, or I2C input on top of the 3 ports included on the Navio2. The Raspberry Pi also has four USB ports that can be adapted to any of the other protocols. This gives us tons of options in terms of connectivity which will allow us to add in more sensors or different sensors during the development phase. Just like the BeagleBone, this board combination supports WiFi, Bluetooth, and BLE. It does not implement DSM-X, but as mentioned previously, that could be added with an external receiver. Another benefit of this solution is the low power requirements of the Raspberry Pi 4. It typically only draws 3W versus the 10W which the Jetson Nano board consumes. Power consumption is something to consider when deciding if we will get an external processing board as this drone runs on a battery. As far as I can see, the only downside of this flight control board solution is the price. It seems like a much better version of the BeagleBone and the price reflects that. It is between $200 and $250 depending on the model of Raspberry Pi we choose. This is around $60 more than the BeagleBone Blue paired with an Nvidia Jetson Nano configured similarly. Because the flight control board and computation board are interdependent, we must consider them as a pair when making decisions. With that being said, the extra reliability and ease of communication between boards makes the Navio2 seem like the best option for a flight controller board.

Figure 21: Navio2



3.2.6 Data Processing Board

The purpose of the data processing board is to take in all sensor data, process it, and then output drone commands. As mentioned previously, flight control boards only have very weak coprocessors which are used to process onboard sensor data and run flight control algorithms such as stabilization. In order to integrate additional sensors such as ultrasonic or infrared rangefinders and process the data in order to achieve autonomous navigation, additional computational power is required. Since we plan to implement a robot vision-based navigation algorithm, our system will require a relatively high-powered data processing board.

Recently, powerful single-board computers have risen in popularity for robotics applications. Being powerful but small and light makes them perfect for our application. Just like the flight control board, this is one of the key components of our system, so it is important to be diligent in choosing the very best options for our application. One of the main considerations for us was computing power. The CPU and GPU processing capability is what determines the speed of our obstacle detection algorithm. Running slower could be the difference between avoiding an obstacle and crashing. For that reason, we only looked at units with high computing power. On a related note, our particular application of computer vision is a memory-constrained task. This means that more memory is a significant benefit that would realize serious performance benefits. We only considered data processing boards with a minimum of 2GB of memory, but more is better. Connectivity is another big concern for our board. We need to be able to connect every sensor onto the board, as well as connect it to the flight controller. This means that we need a certain number of GPIO pins in addition to the other I/O methods on the board. Certain solutions integrate with the flight control board much easier than others. This is something to consider as spending less time on the flight controller-data processor interface will allow us to spend more time on algorithm optimization and additional feature development. Being part of a flight platform that runs on batteries, this board needs to be lightweight as well as efficient. A data processing board that uses too much power will decrease the flight time which is a trade-off we have to consider. From a programming perspective, software support is something we should consider when choosing a board. The ability to program with high-level languages or even run an operating system is something that can drastically increase the ease and speed of the software development process. The ability to support open-source libraries is even better. On a related note, having a community around the processing board will allow us to adapt previous solutions from other contributors as well as ask questions to experts when we have trouble. The final consideration for our data processing board is cost. With our overall goal of keeping the drone inexpensive, this is important to consider.

As mentioned in the flight controllers' section, the data processing board and flight control board are very interdependent and thus should be considered together. The two options we have for integrating them into the system are a separate flight controller-data processor pair and an all-in-one solution. The option that we choose could determine the data processing board that we go with.

3.2.6.1 Nvidia Jetson Nano

The Nvidia Jetson Nano is a very small but very powerful single-board computer. As seen in **Table 15,** it has a 64-bit Quad-core ARM CPU, an NVIDIA Maxwell GPU, and either 2GB or 4GB of memory. These components make it a very powerful computer, especially for image processing and AI-related processing. As far as connectivity, the Jetson Nano has 40 GPIO pins, 4 USB 3.0 ports, and a MIPI CSI port for a camera. This is important because the Jetson Nano does not have the capability to be part of an all-in-one solution. We will be required to connect it to all of the sensors and to the flight control board on our own. Since all of the GPIO pins will be open, this will be plenty of connections for our use case.

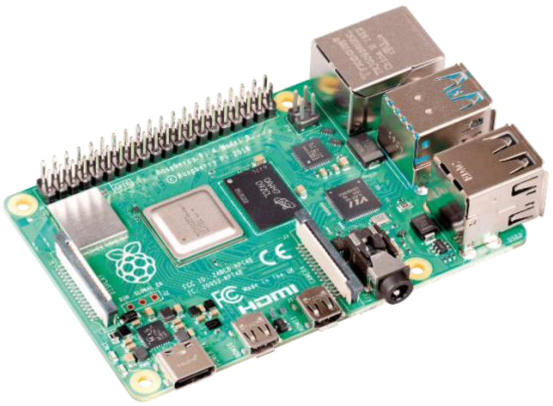
Figure 22: Nvidia Jetson Nano

At 250 grams, the Jetson Nano is relatively lightweight and would not make a significant impact on the final system. One of its biggest flaws, however, is its power consumption. This board can use between 7.5W and 15W under load. This has the potential to impact the efficiency of the drone and decrease the flight time. Additionally, this much power requires a relatively large and heavy heatsink, as seen in **Figure 22.** Being a full-fledged computer, supported software is not much of a concern. It can run Linux and any software on top of that. The Jetson Nano does have an active community, though it is smaller than the Raspberry Pi 4's. Thus, the topics are less diverse and may not be as applicable to our application. The price is around $60 for the 2GB version and $90 for the 4GB option. This is more expensive than the Raspberry Pi 4 on its own, but it is important to consider the price in conjunction with the corresponding flight controller. After doing this the system with the Jetson Nano could be between $140 and $170.

3.2.6.2 Raspberry Pi 4B

The Raspberry Pi is very popular for robotics projects, and for good reason. The latest version, the Raspberry Pi 4B, is extremely powerful and very efficient. It has a 64-bit Quad-core ARM processor, integrated Broadcom GPU, and either 2GB, 4GB, or 8GB of memory, seen in **Table 16.** These specifications allow it to tackle many of the computational tasks we would need it to. The board has 40 GPIO pins, 2 USB 3.0 ports, 2 USB 2.0 ports, and a MIPI CSI port for a camera. At this point, it is important to consider how we will connect the board to the flight controller. Like the Jetson Nano, the Raspberry Pi can run independently and communicate with the flight controller via communication standards like UART or I2C. However, we also have the option to connect it directly by using the Navio2. This would drastically simplify the connection between the flight controller and data processor, as well as make communication between them easier. The software for both units would be running on the Raspberry Pi 4B. While this would make integration easier, it would slow down computation as well as take up some of the GPIO pins. In fact, the Navio2 would take up 34 GPIO pins, leaving only 6 for the sensors. Many of these pins would be utilized by connecting the board to an external flight controller anyway, so this is not much of a drawback.

Figure 23: Raspberry Pi 4B



Also, it is important to note that with the 6 remaining pins and 3 additional I/O ports on the Navio2, it is possible to connect multiple sensors to each. This means that I/O availability is not too much of a concern. Where the Raspberry Pi 4B really shines is in the airframe-specific specifications. As seen in **Figure 23,** the Raspberry Pi is significantly smaller than the Jetson Nano. It weighs only 45g, which is significantly lighter than the Jetson Nano. It also only consumes between 2.5W and 5W of power. Together these two attributes would contribute to a very efficient system which could improve our power efficiency and flight time. As a single board computer, the Raspberry Pi can run Linux as well as any software that is compatible with Linux. This Board has a very active community, especially for related robotics projects. This could be very helpful in learning from other's experiences and making connections to get help when needed. Additionally, since the Raspberry Pi 4 is backward compatible with software for all previous versions of the Raspberry Pi, there is a huge backlog of projects that we could learn from. Even though it is a very capable board, it is still inexpensive. The 2GB version is only $35 and the 4GB version is $55. Being able to go all the way up to the 8GB version for an even lower cost than the 4GB Jetson Nano gives us even more flexibility when it comes to choosing our configuration. However, that is not the whole picture. Going with the discrete data processing board, this unit would be cheaper than the Jetson Nano, but not when implementing the all-in-one flight controller-data processor system. Combined with the Navio2, the price ranges between $200 and $250 which is more than any of the configurations with the Jetson Nano.

Table 15: Data Processing Board Specifications

	Raspberry Pi 4	NVIDIA Jetson Nano
CPU	Quad-core ARM Cortex-A72 64-bit @ 1.5 Ghz	Quad-Core ARM Cortex-A57 64-bit @ 1.42 Ghz
GPU	Broadcom VideoCore VI (32-bit) (Integrated)	NVIDIA Maxwell w/ 128 CUDA cores @ 921 Mhz
Memory	4 GB LPDDR4**	4 GB LPDDR4 (2 GB version also available)***
Networking	Gigabit Ethernet / Wifi 802.11ac	Gigabit Ethernet / M.2 Key E (for Wifi support) (2 GB version includes USB Wireless adapter)
Display	2x micro-HDMI (up to 4Kp60)	HDMI 2.0 and eDP 1.4
USB	2x USB 3.0, 2x USB 2.0	4x USB 3.0, USB 2.0 Micro-B
Other	40-pin GPIO	40-pin GPIO
Video Encode	H264(1080p30)	H.264/H.265 (4Kp30)
Video Decode	H.265(4Kp60), H.264(1080p60)	H.264/H.265 (4Kp60, 2x 4Kp30)
Camera	MIPI CSI port	MIPI CSI port
Storage	Micro-SD	Micro-SD
Price	$55 USD	$99 USD

Table 16: Data Processing Board Benchmarks

Test	Raspberry Pi 4 B 8GB	Jetson Nano 4GB	Notes
Tinymembenc Standard Memcpy	2526.8 MB/s	3501 MB/s	MB/s > Higher Is Better
TTSIOD 3D Renderer Phong Rendering With Soft-Shadow Mapping	32.3075 FPS	41.00 FPS	FPS > Higher Is Better
7-Zip Compression	3701 MIPS	3501 MIPS	MIPS > Higher Is Better

Compress Speed Test			
C-Ray Total Time – 4K, 16 Rays Per Pixel	609.431 Seconds	932 Seconds	Seconds < Lower Is Better
Primesieve 1e12 Prime Number Generation	519.238 Seconds	468 Seconds	Seconds < Lower Is Better
AOBench Size: 2048 x 2048	125.643 Seconds	187 Seconds	Seconds < Lower Is Better
FLAC Audio Encoding WAV To FLAC	85.805 Seconds	104.01 Seconds	Seconds < Lower Is Better
LAME MP3 Encoding WAV To MP3	124.952 Seconds	144.21 Seconds	Seconds < Lower Is Better
Perl Pod2html	0.58082059 Seconds	0.7114 Seconds	Seconds < Lower Is Better
Redis GET	491168.39 Requests Per Second	568431 Requests Per Second	Requests Per Second > Higher Is Better
PyBench Total For Average Test Times	5673 Milliseconds	7080 Milliseconds	Milliseconds < Lower Is Better

3.2.6.3 Comparison

In terms of performance, the Raspberry Pi 4B and NVIDIA Jetson Nano are very close. As seen in **Table 16,** the slightly faster CPU of the RBP4 gives it the edge in serial tasks like compression, while the discrete GPU of the Nano makes it noticeably faster in parallel tasks like rendering. It is also important to consider AI image processing applications, as our drone will be using computer vision to navigate. **Table 17** shows that the Jetson Nano is between three and ten times faster than the RBP4 when running Convolutional Neural Network algorithms.

However, this extra GPU performance of the Nano comes at the cost of nearly three times the power consumption and five times the weight compared to the RBP4. In embedded applications, having adequate compute power is important but having extra is not necessarily a good thing. It means that you sacrificed in some other aspect of the system such as cost, efficiency, size, or weight. Based on our research, we are confident that the Raspberry Pi 4B has the adequate performance to run all of the flight control and data processing algorithms. The Jetson Nano would allow us to run our algorithms faster, but we have determined that the tradeoff on power consumption, weight, and complexity is not worth it.

Additionally, if the RBP4 turns out to be underpowered for our application, we have a backup plan or adding a “Compute Stick” which is basically an external GPU. As seen in **Table 17**, the Intel Neural Stick 2 and the Google Coral USB both improved the RBP4’s performance to near the Jetson Nano’s, sometimes even beating it. This would be an additional $60 to $80 on top of the already more expensive RBP4 + Navio2 combination, but it is important to note that this is only a backup plan. Based on our research, we will not need to use these. We have determined that the performance of the Raspberry Pi 4B combined with the Navio2 is sufficient for our purposes, and the extra cost is worth the increased simplicity and additional capabilities it would allow.

Table 17: Data Processing Board CNN Benchmarks

Model	Framework	Raspberry TF-Lite	Raspberry Pi NCNN	Raspberry Pi Intel Neural Stick 2	Raspberry Pi Google Coral USB	Jetson Nano
Efficient Net-B0 224x224	TensorFlow	14.6 FPS (3) 25.8 FPS (4)	-	95 FPS (3) 180 FPS (4)	105 FPS (3) 200 FPS (4)	216 FPS
ResNet-50 (244x244)	TensorFlow	2.4 FPS (3) 4.3 FPS (4)	1.7 FPS (3) 3 FPS (4)	16 FPS (3) 60 FPS (4)	10 FPS (3) 18.8 FPS (4)	36 FPS
MobileNet-v2 (300x300)	TensorFlow	8.5 FPS (3) 15.3 FPS (4)	8 FPS (3) 8.9 FPS (4)	30 FPS (3)	46 FPS (3)	64 FPS
SSD Mobilenet-V2 (300-300)	TensorFlow	7.3 FPS (3) 13 FPS (4)	3.7 FPS (3) 5.8 FPS (4)	11 FPS (3) 41 FPS (4)	17 FPS (3) 55 FPS (4)	39 FPS
Binary model (300x300)	XNOR	6.8 FPS (3) 12.5 FPS (4)	-	-	-	-
Inception V4 (299x299)	PyTorch	-	-	-	3 FPS (3)	11 FPS
Tiny YOLO V3 (416x416)	Darknet	0.5 FPS (3) 1 FPS (4)	1.1 FPS (3) 1.9 FPS (4)	-	-	25 FPS

Open_Pose (256x256)	Caffe	4.3 FPS (3) 10.3 FPS (4)	-	5 FPS (3)	-	14 FPS
Super Resolution (481x321)	PyTorch	-	-	0.6 FPS (3)	-	15 FPS
VGG-19 (224x224)	MXNet	0.5 FPS (3) 1 FPS (4)	-	5 FPS	-	10 FPS
Unet (1x512x512)	Caffe	-	-	5 FPS	-	18 FPS
Unet (3x257x257)	TensorFlow	2.0 FPS (3) 3.6 FPS (4)	-	-	-	-

3.2.7 Mapping Algorithms

The mapping algorithm is the piece of software that makes our drone fully autonomous. It allows the drone to understand where it is within its environment and understand the obstacles it has to overcome, rather than simply telling it the distance to the closest obstacle. The rangefinders included in our drone will be a key part of the collision avoidance system which will stop the drone from bumping into things. However, our goal is for the drone to be able to navigate between obstacles and reach a destination. In order for the drone to plan a path for itself, it needs to know where it can go and where it cannot. From there we can run a simple pathfinding algorithm such as A* and the drone will navigate autonomously.

When selecting a mapping algorithm, we analyzed several characteristics. One of the key requirements was the computational expense. Being an embedded system, we only had limited resources to work with. Keeping in mind that there are other programs being run in addition to the mapping algorithm, such as the flight control firmware, the operating system, and sensor reading, it was important for our mapping algorithm to be efficient with its computational resource usage. Additionally, it is important for our algorithm to be fast. Being a drone, every second of flight time takes energy. Flying vehicles do not have the luxury that a ground rover has in terms of waiting idly while computations are performed. This also means that flying the drone slower to account for the slower mapping algorithm will reduce range. It is important for our mapping algorithm to keep up with our drones flying at a reasonably high speed. Another consideration is that our mapping algorithm should be based on monocular computer vision. One of the goals of our drone is to be inexpensive so that it can be deployed on a mass scale for shipping companies. For this reason, we chose camera-based computer vision over LIDAR mapping since LIDAR sensors are orders of magnitude more expensive than cameras in the current market. Similarly, we chose a single camera over stereo cameras because two cameras would increase cost and complexity. Monocular computer vision algorithms rely on motion parallax to calculate depth information rather than comparing the information between images of different perspectives as with stereopsis.

A key consideration for our project is the level of robustness of our algorithm. For this, it is important to understand the difference between Visual Odometry and SLAM (Simultaneous Localization and Mapping). Visual Odometry aims to create a locally consistent record of camera motion, which is effectively the same as drone motion for our use case. SLAM, on the other hand, aims to create a globally consistent map of the environment. Both algorithms will give us useable 3D maps of our environment, which we can then use to calculate a path for the drone to take. The difference is how they do it, which can be seen by comparing **Figure 24** and **Figure 25.** SLAM uses much more sophisticated calculations to make it more accurate over long distances and long timeframes, as well as making it robust over periods of bad visual information. Of course, the extra calculations also slow down the SLAM algorithm significantly. A comparison of the different algorithms we looked at, and their different approaches to solving the problem can be seen in **Table 17.** Since Visual Odometry suffers from inaccuracies in regions far in space and long ago in time, but remains accurate in local time and space, it should be sufficient for obstacle avoidance. The goal for us is not to create an accurate 3D map of the world, but just to get our drone past an obstacle. While the robustness features of SLAM would be nice to have, they are not strictly necessary and probably not worth the additional computation time. With these considerations in mind, we selected four potential mapping algorithms to evaluate.

Figure 24: Mapping Algorithm Flowchart

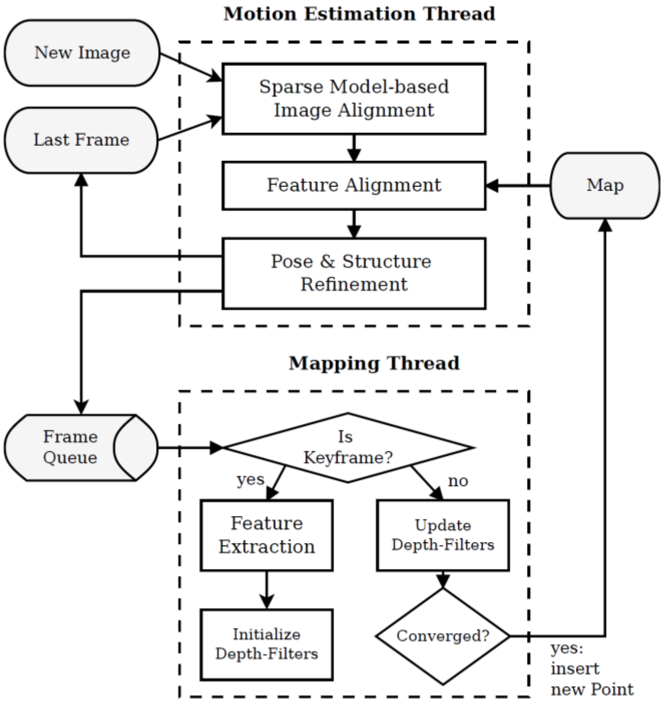
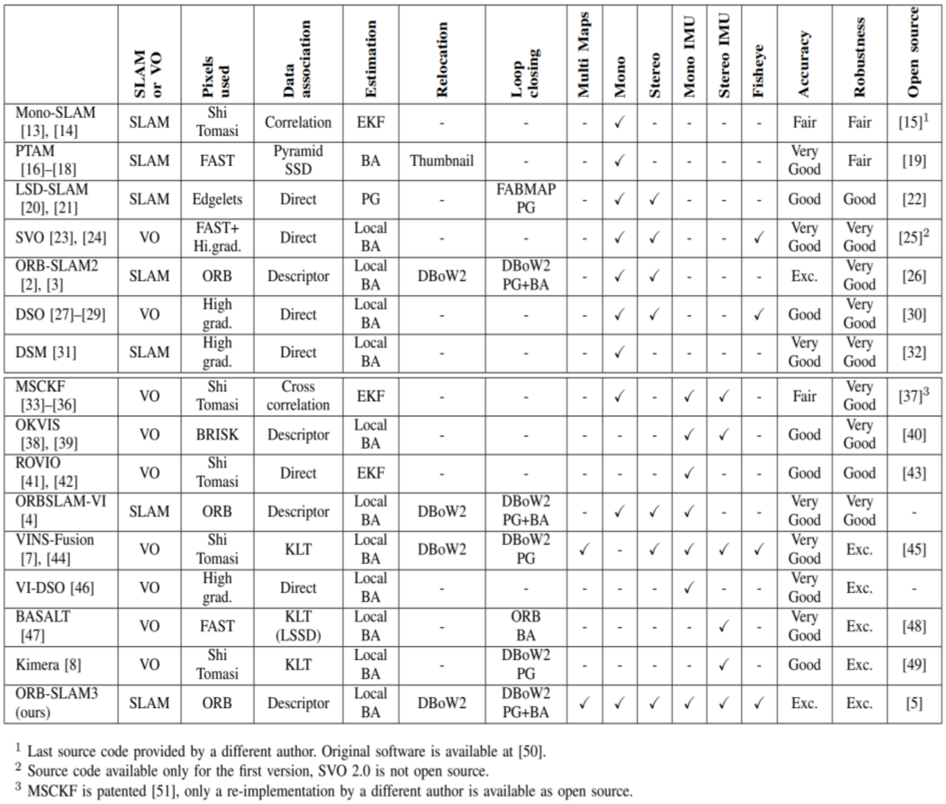


Figure 25: Mapping Algorithm Table



	SLAM or VO	Pixels used	Data association	Estimation	Relocation	Loop closing	Multi Maps	Mono	Stereo	Mono IMU	Stereo IMU	Fisheye	Accuracy	Robustness	Open source
Mono-SLAM [13], [14]	SLAM	Shi Tomasi	Correlation	EKF	-	-	-	✓	-	-	-	-	Fair	Fair	[15][1]
PTAM [16]–[18]	SLAM	FAST	Pyramid SSD	BA	Thumbnail	-	-	✓	-	-	-	-	Very Good	Fair	[19]
LSD-SLAM [20], [21]	SLAM	Edgelets	Direct	PG	-	FABMAP PG	-	✓	✓	-	-	-	Good	Good	[22]
SVO [23], [24]	VO	FAST+ Hi.grad.	Direct	Local BA	-	-	-	✓	✓	-	-	✓	Very Good	Very Good	[25][2]
ORB-SLAM2 [2], [3]	SLAM	ORB	Descriptor	Local BA	DBoW2	DBoW2 PG+BA	-	✓	✓	-	-	-	Exc.	Very Good	[26]
DSO [27]–[29]	VO	High grad.	Direct	Local BA	-	-	-	✓	✓	-	-	✓	Good	Very Good	[30]
DSM [31]	SLAM	High grad.	Direct	Local BA	-	-	-	✓	-	-	-	-	Very Good	Very Good	[32]
MSCKF [33]–[36]	VO	Shi Tomasi	Cross correlation	EKF	-	-	-	✓	-	✓	✓	-	Fair	Very Good	[37][3]
OKVIS [38], [39]	VO	BRISK	Descriptor	Local BA	-	-	-	-	-	✓	✓	-	Good	Very Good	[40]
ROVIO [41], [42]	VO	Shi Tomasi	Direct	EKF	-	-	-	-	-	✓	-	-	Good	Good	[43]
ORBSLAM-VI [4]	SLAM	ORB	Descriptor	Local BA	DBoW2	DBoW2 PG+BA	-	✓	✓	✓	-	-	Very Good	Very Good	-
VINS-Fusion [7], [44]	VO	Shi Tomasi	KLT	Local BA	DBoW2	DBoW2 PG	✓	-	✓	✓	✓	✓	Very Good	Exc.	[45]
VI-DSO [46]	VO	High grad.	Direct	Local BA	-	-	-	-	-	✓	-	-	Very Good	Exc.	-
BASALT [47]	VO	FAST	KLT (LSSD)	Local BA	-	ORB BA	-	-	-	-	✓	-	Very Good	Exc.	[48]
Kimera [8]	VO	Shi Tomasi	KLT	Local BA	-	DBoW2 PG	-	-	-	-	✓	-	Good	Exc.	[49]
ORB-SLAM3 (ours)	SLAM	ORB	Descriptor	Local BA	DBoW2	DBoW2 PG+BA	✓	✓	✓	✓	✓	✓	Exc.	Exc.	[5]

[1] Last source code provided by a different author. Original software is available at [50].

[2] Source code available only for the first version, SVO 2.0 is not open source.

[3] MSCKF is patented [51], only a re-implementation by a different author is available as open source.

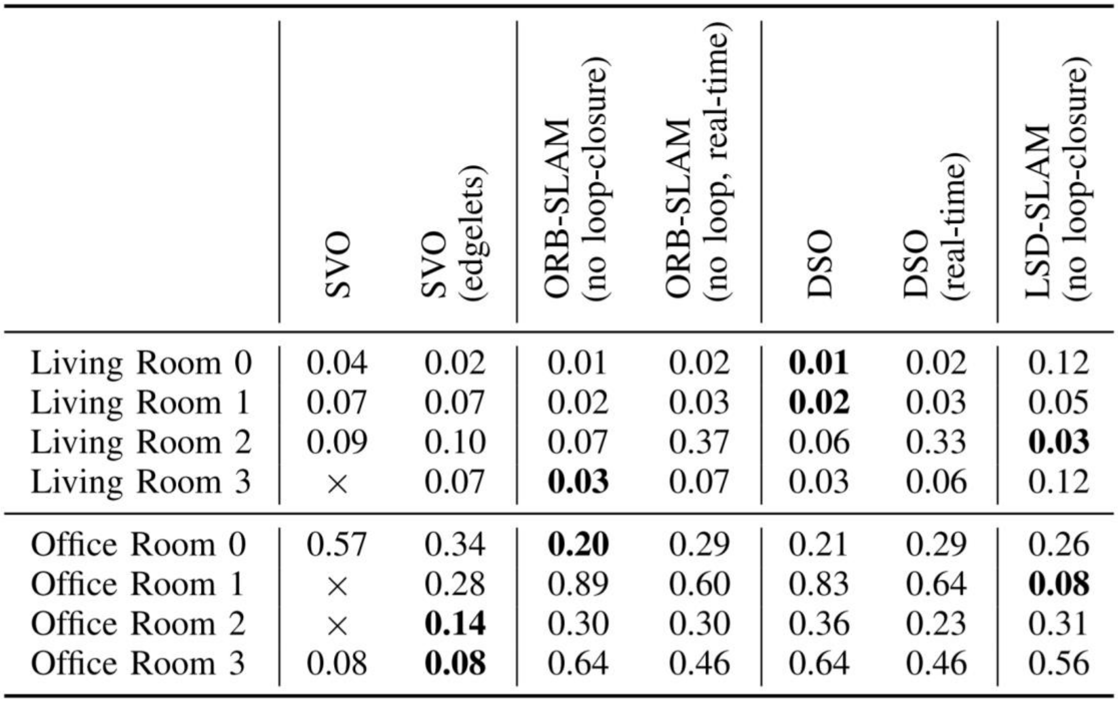
3.2.7.1 Semi-Direct Visual Odometry 2.0

Semi-Direct Visual Odometry was first implemented in 2014 and was later extended in 2017, making it one of the oldest algorithms being compared. This is not necessarily a bad thing, though, as it is still very effective. As the name suggests, SVO implements Visual Odometry and lacks the features of SLAM such as loop closing and key point reuse. However, it makes up for it by being around ten times faster than the SLAM algorithms, as seen in **Table 18.** Additionally, its CPU utilization is between three and five times lower than that of the SLAM algorithms. This will free up resources for other processes such as path planning.

Table 18: Mapping Algorithm Frame Time Comparison

	Frame Times (ms)		
Mapping Algorithm	Mean	St.D.	CPU@20fps
SVO Mono	2.53	0.42	55
ORB Mono SLAM	29.81	5.67	187
LSD Mono SLAM	23.23	5.87	236
DSO Mono	9	-	-

Table 19: Mapping Algorithm Accuracy

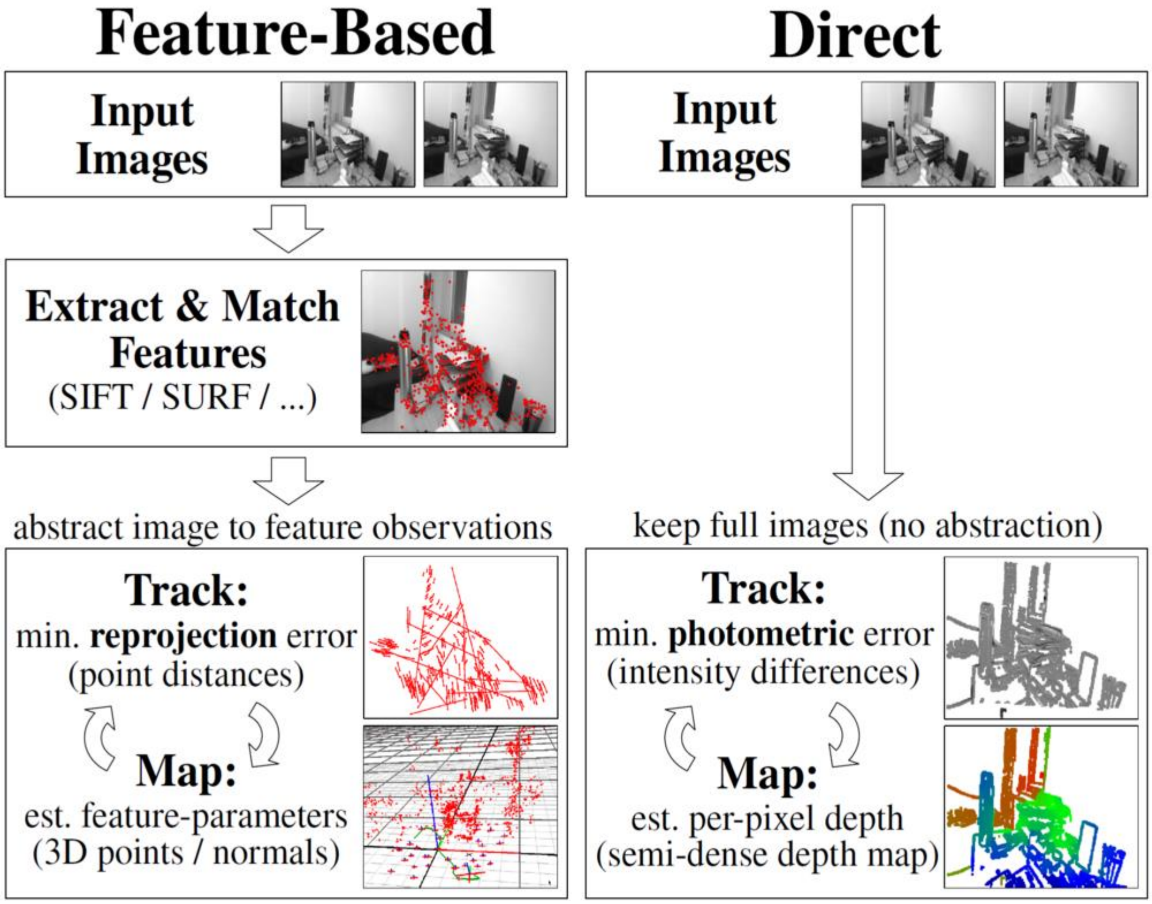


	SVO	SVO (edgelets)	ORB-SLAM (no loop-closure)	ORB-SLAM (no loop, real-time)	DSO	DSO (real-time)	LSD-SLAM (no loop-closure)
Living Room 0	0.04	0.02	0.01	0.02	**0.01**	0.02	0.12
Living Room 1	0.07	0.07	0.02	0.03	**0.02**	0.03	0.05
Living Room 2	0.09	0.10	0.07	0.37	0.06	0.33	**0.03**
Living Room 3	×	0.07	**0.03**	0.07	0.03	0.06	0.12
Office Room 0	0.57	0.34	**0.20**	0.29	0.21	0.29	0.26
Office Room 1	×	0.28	0.89	0.60	0.83	0.64	**0.08**
Office Room 2	×	**0.14**	0.30	0.30	0.36	0.23	0.31
Office Room 3	0.08	**0.08**	0.64	0.46	0.64	0.46	0.56

Another requirement we had was for our mapping algorithm to work well with a monocular camera setup, which SVO 2.0 does very well. Additionally, it is seen that SVO does very well compared to the other algorithms we looked at, in terms of accuracy, as seen in **Table 19.** Each algorithm has a dataset on which it performs best, but the fact that SVO is keeping up with the others while being so much faster is impressive. It is important to note that in the comparison **Table 19,** ORB-SLAM and LSD-SLAM both have loop closing turned off, which is one of the features that makes it more accurate but also more computationally expensive. Overall, SVO is a very effective mapping algorithm and seems like a good fit for our application. It would be nice to have some of the extra features of SLAM, but they are not strictly necessary. The extra computational power freed by using SVO 2.0 versus some of the other algorithms could be worth the slight decrease in inaccuracy.

3.2.7.2 LSD-SLAM

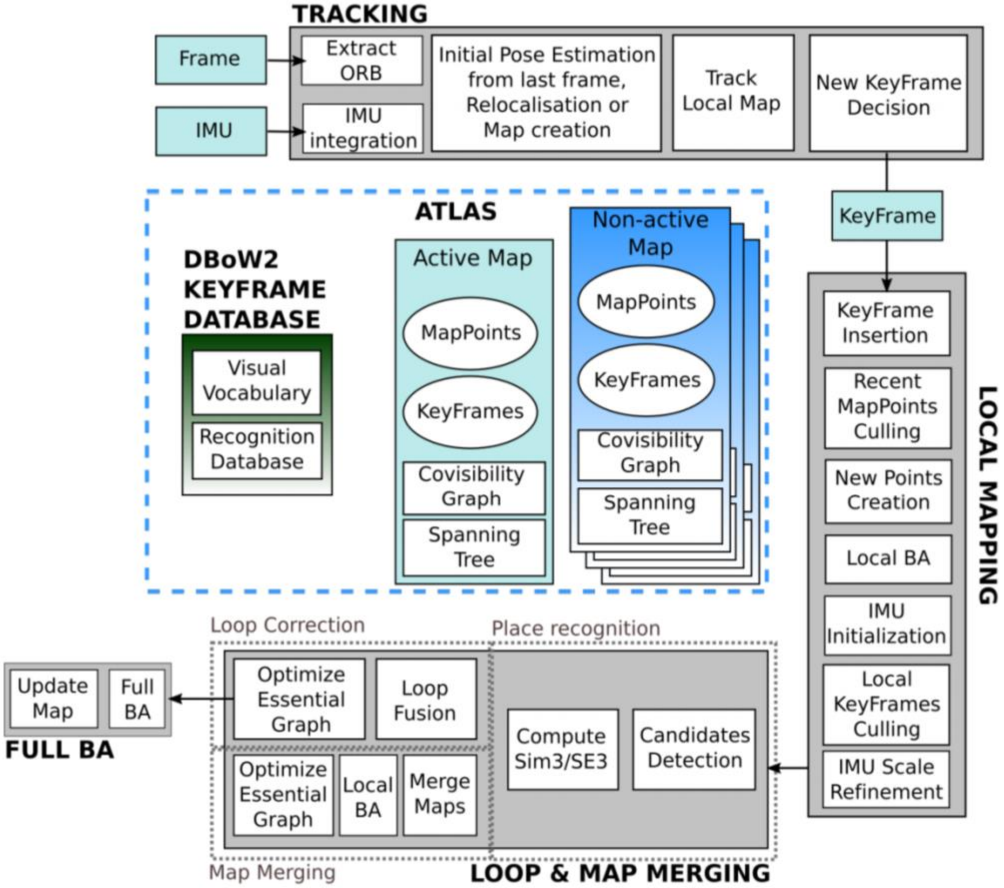
Figure 26: Feature-Based vs Direct Visual Information Mapping



LSD-SLAM or Large-Scale Direct SLAM was released in 2014 with no updates since then, making it the oldest algorithm being considered. It uses the direct method, shown in **Figure 26,** which increases accuracy and robustness in sparsely textured environments, as well as making the 3D map denser. Non-direct methods take the input images and extract features. This decreases the bandwidth of data being processed, but abstraction necessarily decreases information that can be processed. By contrast, the direct method skips feature extraction and uses the full images for tracking and mapping. This does not necessarily speed up the algorithm as processing happens on larger objects of data and could increase memory usage as these images must be stored for longer. In terms of speed, LSD-SLAM is relatively slow. In **Table 18,** it is shown to be slightly faster than ORB-SLAM, but it takes significantly more computational power. Additionally, ORB-SLAM has since been updated to increase both accuracy and speed, while LSD-SLAM has not. It does work well with a monocular camera setup, however. Being a SLAM algorithm, it is extremely accurate and robust. LSD-SLAM is a very effective method for mapping, but it is simply too slow and too expensive to run on an embedded system such as our drone. For that reason, this does not seem like a very good option for us.

ORB-SLAM3

Figure 27: ORB-SLAM3 Mapping Algorithm



The second SLAM algorithm we evaluated was ORB-SLAM3 which uses Oriented FAST and Rotated BRIEF (ORB) techniques for feature extraction. As this implies, ORB-SLAM is a feature-extraction-based method as seen in **Figure 27.** The newest version, ORB-SLAM3 was released in mid-2020, making it a very new algorithm. As expected, it is one of the best we have looked at. It is relatively fast for what it does, definitely faster than LSD-SLAM. It is also one of the most accurate options. It supports monocular setups like the other options we have discussed. It is also extremely robust against periods of poor visual information. An example of poor information would be pure rotation with no lateral motion. This is bad because it is possible for previously mapped regions to leave the visual frame even though the region entering the frame is unable to be mapped. This is because, without lateral motion, the system is not receiving depth information. Effectively, pure rotation forces the system to restart its map. ORB-SLAM3 is robust against this, but even if it does happen, the algorithm supports re-localization, combining the new map with the old map, as seen in **Figure 25.** It also supports map reuse over multiple runs, which could decrease computation around high traffic areas like the warehouse. This algorithm supports all of the other features expected of SLAM such as loop closing too. Overall, ORB-SLAM3 is the most fully-featured, robust, and accurate of all of our options. Our only concern is that it could be too slow or too computationally expensive.

3.2.7.3 Direct Sparse Odometry

Direct Sparse Odometry is the Successor to LSD-SLAM, made by the same research team. It was released in 2016 which leaves it average in terms of age. It is an implementation of Visual Odometry similar to SVO. However, it does have multiple extensions available, such as Visual-Inertial DSO which adds in measurements from accelerometers and gyroscopes to obtain more accurate ego-motion. It also has an extension with loop-closure and Sim(3) pose graph optimization. This gets the algorithm closer to full implementation of SLAM. As seen in **Table 18,** Direct Sparse Odometry is only slightly slower than SVO, and between two and three times faster than both SLAM implementations. Being Visual Odometry rather than SLAM, it is also less computationally expensive. It is designed to work with a monocular camera setup, so that should not be a problem. It is relatively accurate as seen in **Table 19.** Furthermore, it is robust to short periods of poor visual information. The robustness can be increased further with the previously mentioned extensions. Overall, Direct Sparse Odometry is a very good option for our system. It is fast enough while also not utilizing too much of the computation power. It also gives us the flexibility to pick the features we really need without including the ones we don’t which will help us optimize our computation usage.

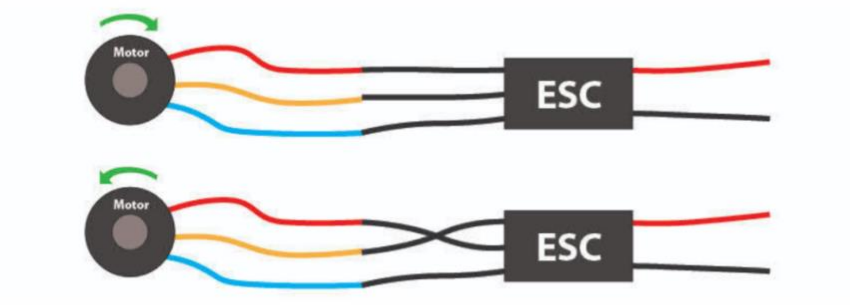
3.2.7.4 Comparison

It is impossible to determine the mapping algorithm for our application without performing our own testing. However, from our research, ORB-SLAM3 appears to be the best option for us. Being a full SLAM implementation, it is the most fully featured, the most accurate, and the most robust. For this reason, ORB-SLAM3 is our first choice. It is possible that it does not run fast enough on our processing board, or that it utilizes too much of the available compute resources. In that case, we will use Direct Sparse Odometry. This is our second choice because it is still very accurate and robust, while still being much faster than ORB-SLAM3. It will also give us the option to add in features such as loop closing if we have leftover compute power. Our final backup option if the previous two do not work is Semi-Direct Sparse Odometry. It is the fastest and least computationally expensive algorithm we looked at. It is slightly less accurate and robust than the others, but it is almost guaranteed to be able to run on even the lowest powered compute board. We are not further considering LSD-SLAM as an option. It is not a bad algorithm, but there are better options for what we are using it for. ORB-SLAM3 is just as full-featured while being faster and less computationally expensive. If ORB-SLAM3 does not work for us after testing, it can be assumed that LSD-SLAM will fail for the same reasons. For that reason, we will attempt to use ORB-SLAM3, with DSO as a backup and VSO as a worst-case scenario.

3.2.8 Electronic Speed Controllers

ESC stands for Electronic Speed Controller and from the name comes the goal of the device. The device's goal is to control the speed of the drone and this is being given by telling the motor to rotate and essentially acts as a translator from the commands being given through the pilot and what voltage/current needs to go through the motor to accomplish it. The ESC adds a lot of control to the motor starting with how the wires are connected. Depending on the user's choice for wire connection will determine which direction the motor rotates. As can be seen in **Figure 28** below, shows that connecting the wires very specifically will turn the motor in one direction and the opposite direction, which is important to set up a 4-propeller system with 2 of them going in the same direction and 2 going in the opposite [3].

Figure 28: ESC Wiring Options



From here there are a few factors that go into choosing an ECS. The factors can include weight, size, voltage, and current output, BECs, and 1 ESC vs 4in1. For all of these factors, ultimately affect the performance of the ESC on the drone. Weight and size are a consideration for all components on a drone because the heavier and larger the size the more power will be needed to fly and control the drone. With ESCs they have small and light sizes, but there is a balance to find based on the motor because if too small then the ECS will overheat and not perform as well. This is part of the reason why it is recommended to make an ESC decision after making the motor decision. Motors have a max output of power and some ESCs can handle more power than the motor is giving but at a higher cost. To save costs is it better to stick to the max power of the motor and not much reason to pick an ESC that handles more power than it is able to give off [3].

Figure 29: ESC 4 IN 1

Add-on components are optional when choosing an ESC. This includes selecting ESC with BEC and a 4 in 1 option versus an individual. A BEC is a battery elimination circuit that provides more control to the power given to other components. For drone projects, this is not a necessary option as the Power Distribution Board (PDB) takes care of this operation. Another optional component is purchasing individual ESC vs 4 in 1. The 4 in 1 is a board that has 4 ESCs already put together and ready to use, whereas individual ones need to be put together. With drone enthusiasts, it might be more cost-effective to purchase the 4 in 1 option, but for beginners, if one ESC is broken then the whole 4 in 1 board needs to be replaced. Hence, why for this project, we wanted to take the individual route, but through researching our desired frame and motor design, we found a kit that includes the ESCs needed to operate. A picture of the ESC is above in **Figure 29** and has a working current of 30A for the motor. While purchasing this separately would be more expensive, finding it with a kit helped with those costs [4].

3.2.9 Coding Language:

For this project, we will be incorporating the basis machine learning code to better direct the drone to the target and then building upon that with computer vision to accurately see objects and find the best path around. Deciding on the machine learning language and library was relatively easy. The biggest debate was choosing Python vs Java as they both can perform the same end goal. With Java, some advantages are the security of the compiler and runtime environment, and because of how the language is structured it helps the performance of machine learning projects perform faster. Some disadvantages would be that it requires a more complex project set up and can be harder to understand when working in a group and can take up more memory over time [5]. For Python, the language is known for its user learnability and easy-to-read syntax, which having two members not extremely familiar with coding, this will be easy for them to follow along and learn. Python’s ML libraries are also very well-known and have a large audience of users online and practiced a lot in the industry, so if we come across bumps with the understanding, we have resources to look at. The main library we are looking to utilize is Tensorflow, which has an abundance of documentation and learning material for us to deep dive into learning how to adapt it to our project. Some disadvantages to Python that might arise is the run time errors because it is a dynamically typed language and known for being slightly slower than other high-level languages. **Table 20** below goes into outlining some of the factors that are looked at when comparing Java vs Python. Noting the easier to code once for Java comment, it has been known that it is easier to maintain and set up a project that can be updated, but with Python, it might come with more hoops to jump through [6] [7].

The next software choice for our project included the firmware being used for the flight controller and so on. The two main options after research for this were Ardupilot and PX4. They both are based in C++ so when deciding which to go with we tried looking to the one that worked best with the flight controller. After further researching the flight controllers, it became apparent that the technology was very popular and the ones we narrowed down could handle both. This opened the doors back open for either one. As students not knowing an extreme amount about the subject, we looked into which software had the more prevalent communities and help forums to guide us to get familiar. With Ardupilot, some of the cons when researching was that a lot of the information was dated back to 2017 and not very kept up and up to date. PX4 had a very nice user interface that made it easy to navigate and find the information easily. There is clear documentation and clearly stated or divided into subject matters for drones. Because of the clear directions and strong forum community we decided to go with PX4 for the time being. The PX4 software will be based mainly on C++ when changing things to interact with the firmware and hardware, and then Python for the software and guidance systems.

Table 20: Python vs. Java

Python	Java
Easier to learn	Easier to code once
Dynamically Typed	Statically Typed
Slightly more popular	Faster than Python
TensorFlow	Weka, Mallet, Deeplearning4j, MOA
Platform dependent	Platform independent

3.2.10 Ground Control Station Software

Ground control software is the program needed for a user to interface with the drone during flight planning and during the flight. There are many different competitive options including QGroundControl and MissionPlanner. We selected QGroundControl for multiple reasons. Firstly, it is supported by the Dronecode Foundation, the same foundation in charge of PX4 autopilot. and MAVLink. Given that, it is very well integrated with the flight control firmware we are using, as well as the messaging protocol. It seems best to stay within the ecosystem rather than trying to integrate a separate ground control software with the rest of the system. It is also highly rated and known to have a clean intuitive user interface, which is another reason for us to choose this option. It is also the only officially supported option for PX4 firmware, so the decision was made for us.

QGroundControl supports all necessary options for setting up and configuring the drone. It allows for flashing the PX4 firmware and establishing the configuration parameters. After the drone is set up, the ground control software can be used to plan the mission, setting waypoints and home coordinates. The software is also used to interface with the drone during flight. It supports the MAVLink communication protocol, which is a way to send commands to the drone. On top of controlling, it with MAVLink commands, QGroundControl supports virtual joysticks to fly the drone. This means that we do not need a dedicated transmitter. During the flight, the software can display telemetry data from the drone such as altitude, flight speed, angle, battery level, and other information. It also supports video streaming which will be interesting because we will be able to see exactly what the drone sees and how it navigates. The ground control software displays the drone on a real-time flight map, including satellite data, topographic data, restricted flight zones, and other geographic data. Additionally, the software supports data logging and flight playback. This will allow us to see all of the data, including the camera input, and analyze it if something goes wrong. QGroundControl is available on every major operating system, including Windows, Linux, OS X, iOS, and Android. This will give us the flexibility to set up and transfer our ground station to any environment.

3.2.11 Drone Control Communication Protocol

3.2.11.1 MAVLink

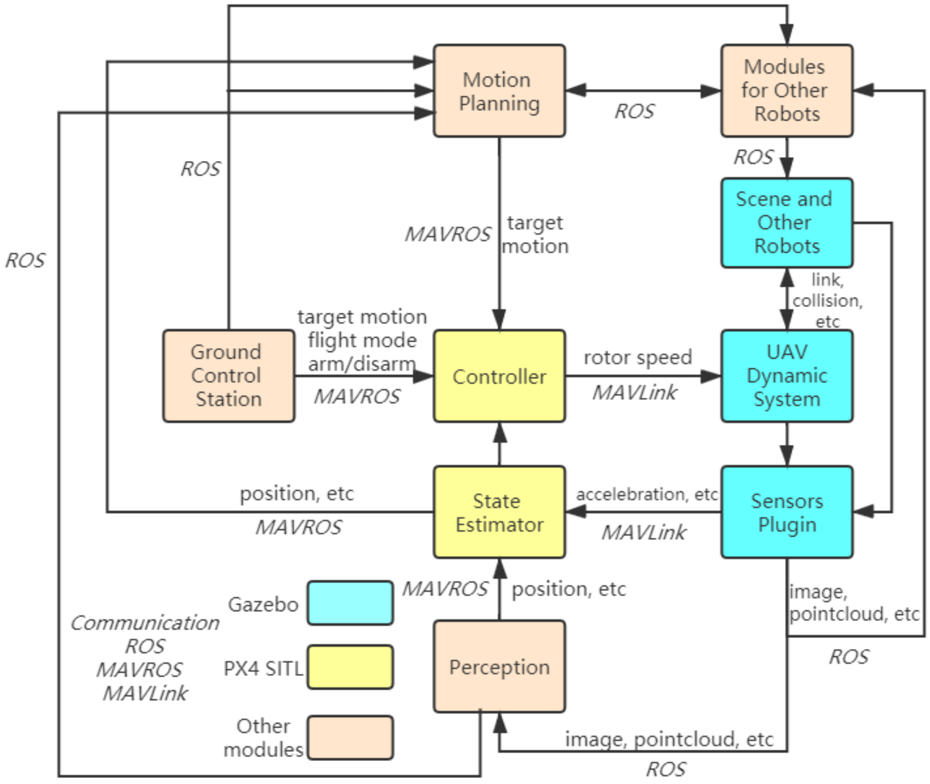
MAVLink is a lightweight messaging protocol designed for drone communication. It is useful for both drone-ground station communication and interprocess communication on the drone itself. It utilizes a hybrid architecture for efficient messaging. Data streams are sent using the publish-subscribe design pattern. On the other hand, Configuration sub-protocols are sent using a point-to-point architecture with retransmission. This combined with the efficient error detection and lack of framing makes the protocol very efficient with low bandwidth overhead. Additionally, the protocol is very reliable. MAVLink supports packet drop detection, corruption detection, and packet authentication. Another key feature is that the MAVLink command set is extensible. The commands are defined as XML files and are easily able to be changed and added to. This protocol also allows us to use basic and programming language and operating systems. It supports C, C++, Java, and Python as well as Windows, Linux, OS X, iOS, and Android. Since this protocol is supported and maintained by the Dronecode Foundation, the same foundation that makes QGroundControl and PX4, it is already integrated with this software. This makes it an easy choice for our communication protocol. It also has many extensions and developer APIs like DroneKit, which will make development even easier.

3.2.11.2 DroneKit

DroneKit is a developer API built on top of MAVLink. It is designed to make the development of drone software using MAVLink even easier through high level abstraction. DroneKit enables the creation of Python applications that communicate with vehicles using MAVLink. It has functions that make it easy to connect and access the vehicles information. Some of the accessible information is vehicle telemetry, vehicle state, and parameter information. In addition to this, it enables easy mission management such as changing waypoints and direct drone control. It's intended use is to be run on a companion computer separate from the flight controller. However, it should not be too hard to set up a ROS node on the same board as the flight control software and have them both running concurrently. We expect DroneKit to be a very valuable asset because it has a very active forum of developers using it for similar projects to ours. It is mainly used for high level cognitive functions such as our SLAM algorithm. These forums should be helpful in our learning how to integrate the different processes together and get everything working. Additionally, it is very well documented with examples for each of the features like the visual odometry package.

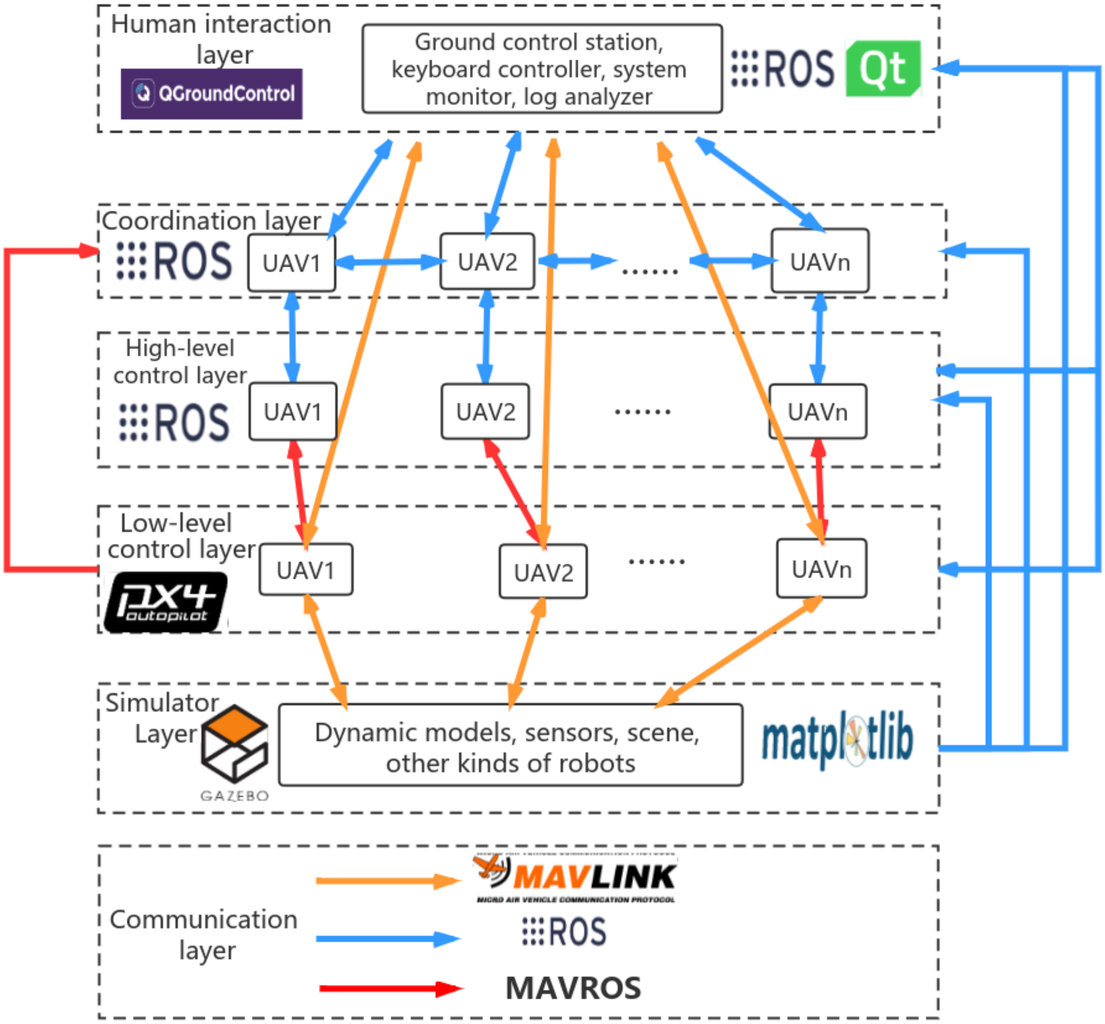
3.2.12 System Stack

Figure 30: System Architecture



Our overall platform will be implemented in two separate systems. The first is a simulator and the second is the physical flight control board. The two systems will by functionally the same but used for different purposes. The simulator will be used for testing our high level cognitive functions such as the SLAM algorithm, the collision avoidance software, and interface between them. Once we successfully implement the system in software, we are safe to move it to hardware. From there we will install and integrate all of the firmware and software on the Raspberry Pi and Navio2. Whether it is the simulation program or the physical flight control board, the software stack is the same. The lowest level software would be operating system and the flight control firmware. The Navio2, our flight control daughter board, comes with a preconfigured operating system image. It is a version of Raspbian with a preemptive Linux kernel. This means that it will do a better job at meeting the real-time scheduling constraints of a safety critical system. It also has all of the Raspberry Pi GPIO pins preconfigured to interface with the Navio2. This overall system stack can be seen in the flowchart in **Figure 30.**

Figure 31: System Stack



Furthermore, it has ArduPilot, ROS, DroneKit, and GStreamer preinstalled. Since we are using PX4 instead of ArduPilot, we will just have to swap the two software's out. ROS is a collection of tools and libraries that help to simplify the process of programming robotic systems. This will be helpful with in creating a unified system between the simulation, the drone, and the ground control system. It also implements useful libraries like OpenCV (computer vision) and MoveIt (path planning). DroneKit, as discussed in a previous section, will be useful in integrating our SLAM algorithm with the flight control software. GStreamer is video streaming software that will allow us to stream the drone's camera view to our ground control station in real time, allowing us to see what it sees. The software program flow as well as inter-process communication is shown in **Figure 31.** It illustrates the way that each layer of software builds on the layers before it, and how each of them communicate and send data between the layers.

3.2.13 Software Simulator

When creating a complicated system, it is important to test each component individually before integrating it into the whole system. The high-level functions in particular, such as the collision avoidance decision making and the path planning algorithms, need to be tested thoroughly as there are many ways that it can give incorrect results. Before testing the cognitive functions on the physical drone, which could easily be damaged in a crash, it is helpful to test using a simulator. The program needs to run the flight control firmware as well as the 3D mapping algorithm exactly the same as it would on the physical drone. However, instead of sending the MAVLink commands to physical hardware like the ESCs, the software sends the commands to the simulation software which then interoperates them and simulates what a real drone would be doing. For us it is important that our simulator is relatively accurate so that we can be sure it is transferable to our physical drone. Another requirement we have for the simulation software is that it works well for testing computer vision. The flight control software has already been vetted and know to be stable. Also, the communication protocol can be tested without flying the drone or putting it in danger. The only part of our system that really needs to be tested before applying it to the drone is the collision avoidance and decision making. For that reason, it is important that the simulation supports computer vision testing and also provides an accurate output for it. The last consideration we had for the simulation software is that it works we our existing stack. We do not want to have to change a particular aspect of our system to make it compatible with the simulation software.

3.2.13.1 Gazebo

Gazebo is one of the most highly regarded simulation software's for drones and autonomous robots. It is designed to be particularly suitable for testing object-avoidance and computer vision programs. It supports both software in the loop and hardware in the loop simulation. Software in the loop, or SITL, means that the flight control software runs on the same computer as the simulation, which is useful because it does not require us to actually have a working drone to test on. On the other hand, Hardware in the loop, or HITL, means that the flight control software and computer vision program run on the flight control board, exactly how they would during a real flight. The only difference between HITL and a real flight is that the flight control software does not output the signals for the motors or other components to active. It is purely running the software on the board as if it was another computer. This is useful because different computers, especially when it comes to embedded computing, can run code differently and have completely different outputs or bugs. One reason could be the kernel scheduler could change the order that threads run. For this reason, testing on the real board that will be running the software is the most accurate way to simulate our system.

Another reason that Gazebo is a good simulation environment for us is that it supports our entire stack. It is made to be run on Linux which makes it compatible with our ground control software. It is also designed to work well with PX4 flight control firmware. As far as high-level functions go, Gazebo supports ROS, which we will be using for several of the computer vision related tasks. The simulation environment is set up to be easy to use and allows the developer to focus on the high-level functionality rather than the physics simulation and aircraft setup. It has several pre-configured vehicles including many different multirotor, fixed wing aircraft, land vehicles, and sea vehicles. This will let us pick the one that matches our aircraft the best rather than having to design our own aircraft. Gazebo also has pre-loaded environments to test obstacle detection and collision avoidance. Another nice feature is verbose debugging; We expect that our software may have some bugs that could cause it to crash. If this happens during flight or on the drone at any time, it will be almost impossible to understand why it happens. On the other hand, using the simulation software with verbose debugging will tell us exactly what happened and why the software crashed. In addition to the previous features discussed, Gazebo is open source and extensible, which means that, in the event that we run into a situation where we need a feature that does not currently exist, we will be able to create it. This gives a lot more flexibility because it means that we are not particularly constrained by the software's current feature set. Overall, Gazebo seems like a great option for our drone software testing.

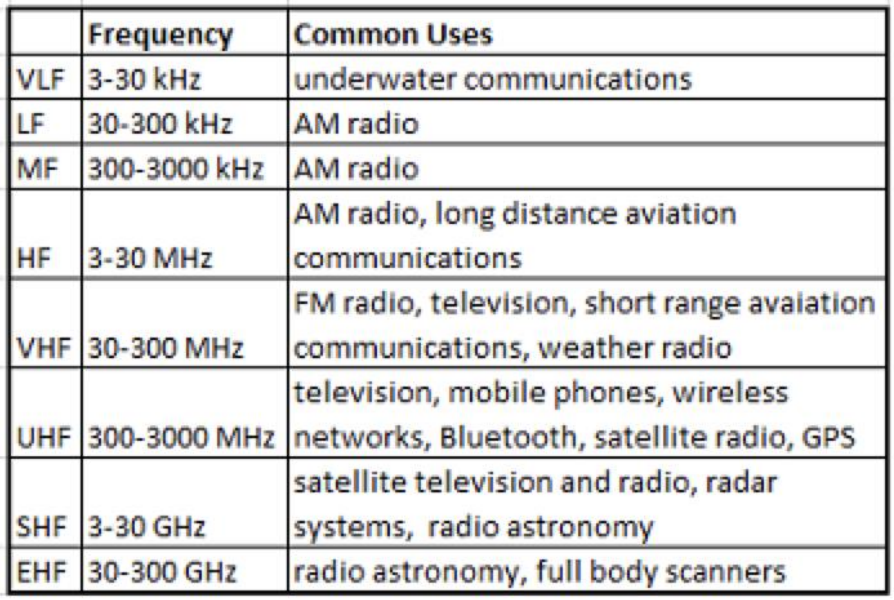
3.2.13.2 XTDrone

Even with all of the features that Gazebo has, XTDrone seems like an even better option for our simulation purposes. As mentioned in the previous section, Gazebo is open source and extensible. XTDrone is one of these extensions. It seems like a better option for us because it is particularly targeted at our feature set. It was created as another student groups senior computer science project. XTDrone is a computer vision simulation software based on ROS, PX4, and Gazebo. It is conveniently already set up to work well for what we will be testing, which is the SLAM algorithm and the collision avoidance algorithm. Like Gazebo, this software is designed to be accurate and applicable to the real hardware that we will deploy it on. One of the main reasons we are selecting this software over the standard Gazebo is because of documentation. Surprisingly, XTDrone has significantly more extensive and more detailed documentation than the official Gazebo release. It is also more relevant to what we are doing. It also has pre-configured computer vision algorithms which are much more advanced than the scenarios in standard Gazebo. For these reasons, XTDrone will make it much easier for our group to test our actual drone software, rather than fiddle with the simulation itself.

3.2.14 Communication Types:

Choosing the form of transmission that is used to communicate to the drone comes with a couple options. The common ones are Wi-Fi, Bluetooth and regular radio frequency. For Wi-Fi, this option was our number one pick because of the commonality and built-in securities with setting up an RF network. Doing just straight RF set up can leave the channel open for others to overtake the drones, but essentially all these options are ways of communicating over RF. The next idea would be Bluetooth as it has an ease of access with connecting to multiple controls, but Bluetooth typically has a range maximum on communicating to the drone, which can become dangerous if it needs to leave that boundary. Wi-Fi covered most of the checkboxes with being able to set up a network for the drone, cover a decent amount of area and available when purchasing the controllers instead of having to install the Bluetooth setup or spend more for the feature. We will also be using GPS to autopilot the drone to the destination, which operates on the Ultra High Frequency (UHF) levels with Wi-Fi and Bluetooth. Our flight controlling being the Navio 2 will have Wi-Fi installed already and made it the easier choice to go with. This will allow us to set up our own network when controlling the drone and giving it instructions. The **Figure 32** below shows that this will operate on ultra-high frequency.

Figure 32: Different Types of Communication



	Frequency	Common Uses
VLF	3-30 kHz	underwater communications
LF	30-300 kHz	AM radio
MF	300-3000 kHz	AM radio
HF	3-30 MHz	AM radio, long distance aviation communications
VHF	30-300 MHz	FM radio, television, short range avaiation communications, weather radio
UHF	300-3000 MHz	television, mobile phones, wireless networks, Bluetooth, satellite radio, GPS
SHF	3-30 GHz	satellite television and radio, radar systems, radio astronomy
EHF	30-300 GHz	radio astronomy, full body scanners

3.2.15 Sensors:

Sensors are going to be a core part of the drone as it will assist in communicating to the drone that it is on the right path and also what objects will be in the way. After research the list of sensor options can include LiDar, ultrasonic, infrared (IR), GPS sensor, and time of flight. A common sensor used in Unmanned aerial vehicles (UAV) is ultrasonic. Ultrasonic uses high frequency sound waves to bounce off objects and detect the distance and size and so on. This helps will try to accurately determine the size of the object and the time it takes for the waves to bounce back determine how close it is. They help detect objects because they are not a camera and do not have to identify color or transparency so works well in any lighting. They are restricted to a certain range depending on the one purchased and do not determine objects well with significant textures. Infrared (IR) sensors are another that are not effective by the amount of light in the area the drone is going through or detecting. This sensor emits an infrared light and then with a photodetector can determine where the object is positioned and how far away. Some advantages to IR is that they are smaller in size, which helps with not adding weight and size to the drone, and it can detect textures in objects better than ultrasonic. Again, it is limited to a certain range and can be affected by the environment and dense objects. LiDar comes in with having some advantages to the range and accuracy by being able to detect farther and can detect objects with texture and more 3D structure. Also, able to process the object faster as it uses smaller wavelengths and not affected by day or night. Disadvantages to the other sensors is the high cost and can be dangerous to use without proper eye protection. Time of flight sensors are another IR structured sensor but use an IR LED light to emit and calculate the object. Time of flight sensors do have a higher cost like LiDar, but do not get affected by weather conditions, can be used for 3D mapping and provides long range of sensing. In **Table 21** below, it summarizes some of the things to consider when choosing sensors. For our project, we will be utilizing multiple sensors in order to make accurate decision making for the drone. Those sensors are still to be determined as we need to evaluate which ones are better suited for the goals we have. Our goal is mainly to avoid objects that come in the way while also fight the best path of avoiding in order to keep it going to the same destination [7].

Table 21: Comparison of Sensors

	Long Range	Cost	Can handle External Conditions	3D Imaging
Ultrasonic	No	Low	No	No
IR	No	Low	Yes	No
LiDar	Yes	High	Yes	Yes
Time of Flight	Yes	Moderate	Yes	Yes

When comparing the sensors in **Table 21** and trying to decide what will be need for our project, Ultrasonic was the best for the job. Most IR would be influenced by ambient light and with LiDar and Time of Flight being a little expensive, then ultrasonic was best for the job. Ultrasonic can detect objects and be very low cost to the project. Ideally, if there was a larger budget then LiDar is the next buy to make it more accurate for the job. Because of the choosing the Navio and the Raspberry pi then it included the Wi-Fi, GPS and altitude sensors and eliminated having to search for other options. We will be using the Wi-Fi for security and communicating with the drone, GPS will be used for navigation and location tracking, and altitude sensors will detect how low to an object or ground the drone is at.

4. Standards and Realistic Design Constraints

Drones are strictly monitored and have many associations that come with them in order to ensure the safety and standards of the machines. With this comes a set of standards and constraints to follow based off our goals. In order, to create an efficient drone many constraints are considered to have it maximize its ability, which goes into further details below. We also research standards associated with certain aspects of the drone to make sure it is not outdated and will not cause harm to the environment and people around.

4.1 Standards

When mass producing any product in any industry comes a set of standards set by companies and organizations to uphold the quality and safety of the materials being used and the product being produced. Even a certain screw being used might have a certain standard it needs to uphold. In this section, we will talk about the standards identified with certain aspects of the drones and where those standards are coming from.

For the following standards they will be coming from sources of organizations. A big organization that creates engineering standards is IEEE. IEEE stands for Institute of Electrical, and Electronics Engineers and the history of the organization’s roots go back to 1884. From then to now they have expanded exponentially and broken up into many boards and committees that do the brute research into what the standards should look like and have a process to get them approved.

Another set of standards to look at is for coding. Having standards for coding will help keep the life span and ability to update easier. A popular standard that will be used for coding is the PEP8 standard. This stands for Python Enhancement Proposal, which comes from Guido van Rossum, Barry Warsaw, and Nick Coglhan, dating back to 2001. (INSET C++ STANDARD)

4.1.1 Wi-fi

Wi-Fi standards have been observed by the IEEE organization since the 1990s. We see this in **Table 22**. The first of the Wi-Fi standard series starts with the standard 802.11 that was released in 1997. From this point on the naming convention adds letters to the end of that standard number/name. The first in the series was 802.11a, which utilized the 5GHz band, but came at an expensive cost. After 802.11a was made in 1999, soon came 802.11b which was on the 2.4GHz band and lowered the cost greatly. This naming conventions continues down the alphabet and all indicate different requirements that come with it and have different purposes depending on the use. For example, 802.11ah has the ability to operate on frequency bands below 1GHz. The goal with creating this standard was to allow Wi-Fi to be extended for wider range and lower energy use, and this standard was made official through publication in 2017. Another very specific standard for Wi-Fi is the 802.11ad. This standard was created to provide extremely efficient speeds under the 60Ghz frequency, but whereas the example before had longer range, this standard has extremely short distance and only able to be communicated up to 11 feet from access point [8].

When it comes to the common Wi-Fi networks and what will be used for this project, we will look at more recent standards. Standard 802.11n is capable of using the 2.4GHz and 5GHz and goes back to being created in 2009. This standard is what more modern routers use in reference to dual band capabilities. More recent and modern one is 802.11ac, which uses the 5GHz band, but also can access the 2.4GHz band by using the 802.11n. Both these common standards take advantage of MIMO, which stands for Multiple Input Multiple Output. This describes how on routers will have multiple antennas in order to be able to handle the amount of data and traffic that is happening through the channels [8].

For our drone, the Raspberry Pi 4b will have the necessary equipment to use Wi-Fi with our software and the drone. The Raspberry Pi model being used is suited to be used with the IEEE standard Wi-Fi of 802.ac/n, which were the two described above. This will allow for decent speeds and long ranges of communication with the drone as it is flying.

Table 22: 802.11n versus 802.11ac

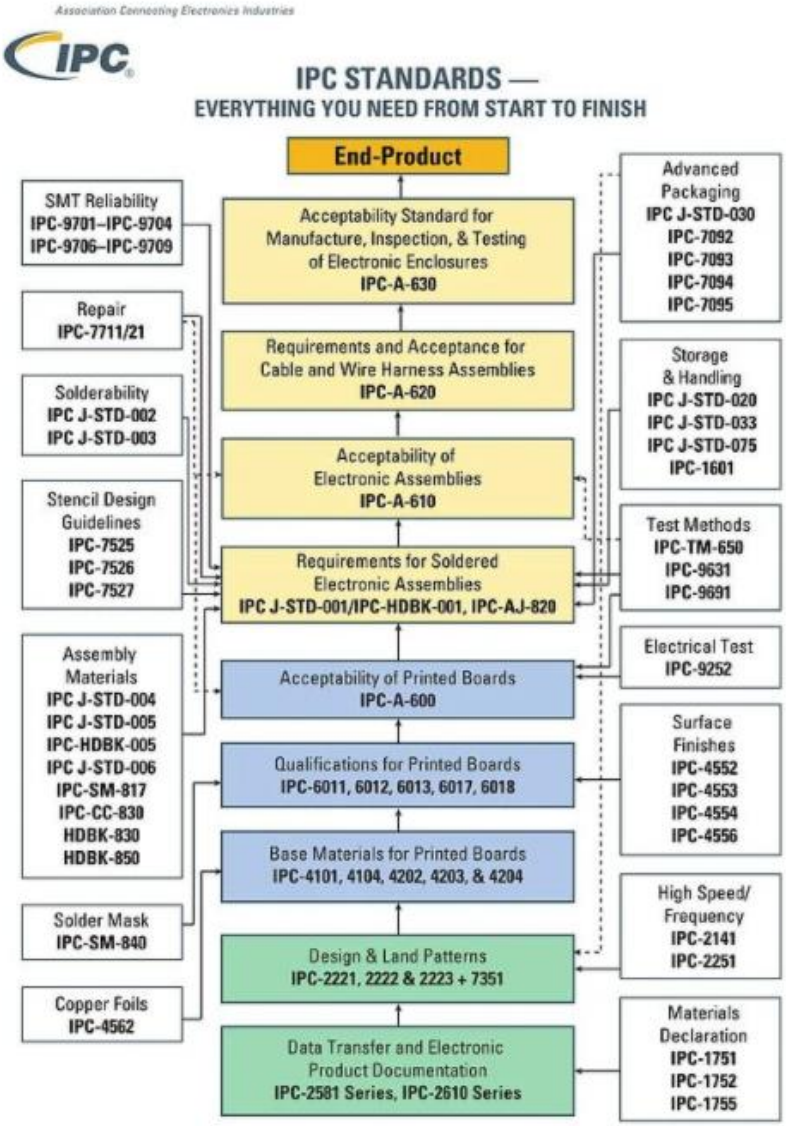
802.11n versus 802.11ac Features		
	802.11n	802.11ac
Band	2.4 GHz & 5 GHz	5GHz
MIMO	Single User	Multi-User
Channel Width	20 or 40 MHz	20,40,80,80-80,160 MHz
Modulation	64 QAM	256 QAM
Spatial Streams	4	3-4

4.1.2 PCB Standards

For our PCB or Printed Circuit Board, we will be using this style of circuitry to make our own Power Distribution Board or PDB. We will use the standard given by Association Connecting Electronics. This company, just like another company talked about later in the paper, was made to produce and deduce standards for individual parts and methods for producing printed circuit boards.

We see from looking into the standards provided by the IPC that it covers many aspects of PCB's and many such aspects we ourselves will need to use such as enclosures, cables ad their thickness, different types of soldering and which tools are used for such things, and finally different styles of documentation that we will need to submit for our PCB. We can see that every aspect of our PCB that we are making for our own uses will have to be up to standard given by the IPC.

Figure 33: Standard IPC Roadmap for PCB



For our actual Board we will pay more attention to certain standards that pertain more to us then others, such as 2221, which gives everyone the standard from which the general form of PCB's are designed. The next, which when taking these into account of what their standards are for, are more of the regular laws that must be looked at when designing any PCB's, are 2615, and 6012. Going over them quickly, 2615 is the generic standard for how thick and wide PCBs are allowed to be as well as the tolerances for how well these PCB’s need to perform under pressure and other conditions so that the impossible is not asked of a PCB. 6012 is the standard test that PCB’s need to be able to reach to be ready for consumer production. These things include performance and other qualification requirements.

Finally, when talking about the IPC and their standards for PCB we have to include their classifications for different uses of PCB and what that means for what kind of qualifications they have to pass to be allowed into circulation and use. We are using ours for the power distribution of a drone’s power system, which means many things, but the most important thing is that our system is an aviation system when applied into the fullness of its operation, but when applied to this project only meets the needs of only a regular PCB system. That means that we will be putting effort to reach the third level classification of PCB’s even when we need to only reach the first level for use in our project. We see through all of this the different standard documentation and needs our project will have to reach, with the total standard roadmap shown in **Figure 33**.

4.1.3 Motor Design Standards

One of our key components of our delivery drone is our motor system. Without the proper motors are drone project we literally and figuratively would it make it off the ground. Due to the critical roles that the motors play we have done research on the Motors available for consumers well also looking into the standards in which motors are held to, to determine the motor specifications that are best for our drone delivery project. The motors that we judge are based off of the specification details provided by the manufacturer and with that information we compare the motors to make the best decision for our delivery drone project.

Some of the important factors to notice on the specification PDFs provided by the manufacturer about the motors are electrical rating for voltages, power, and current. Other mechanical limitations for the motors to also pay attention to when comparing the specifications provided by the manufacturer are torque and RPMs variation. The specification details ascertained from the manufacturer are critical to making educated decisions on which Motors are best suited four our delivery drone project. The specifications help us to elicit the best motors for the best outcome of our project.

There are plenty of regulations placed on motors, but it is the manufacturers obligation to meet these regulations. These regulations also known as standards are determined by industry and varying third-party associations like IEEE/ NEMA. Those third-party associations are IEC which stands for International electrotechnical commissions. The International electrotechnical commissions (IEC) has taken up the due diligence a fabricating standard practice technical restraint for electrical components internationally. Such electrical components that IEC have helped create standards for are Motors. The standards that IEC have created are to ensure that motors are being tested in rated accurately. The IEEE also helps check in regulate standards that are applicable to motors as well in regard to their testing, performance, and ratings. **[2]**

NEMA, also known as National electrical manufacturer association, has a very unique role in motor standards. For example, NEMA also design standards specifically to certain environments the motor is intended to operate within. Such as but not limited to certain enclosures to protect against like water and dust so the Motors can run effectively in these environments. **[7]**

4.1.4 Flying standards

Drone flying has gone though many changes in recent years and will continue to change for years to come. Drones have been used in so many aspects of life that the FAA have passed many regulations and standards that operators need to meet so that no laws are trespassed upon during operation of a drone. Prior to 2020, drones that exceeded a certain weight and size would require a pilot license that must be earned through classes and money. This, at the time, even engrossed upon some sizes that commercial drones were able to achieve, so that many drone camera men and woman were hindered from my hobbies and job opportunities. Now that 2020 has rolled past, the FAA has enabled many drones that now are more mass produced in today’s age and used for different aspects to be used without this pilot license, thus opening the doors for more different commercial drone companies, and our project itself, to take off without interference.

4.1.5 Programming standards

For our project we will be using Python and C++ for the most part to accomplish our goals. The Python will largely be used for the machine learning, autopilot and computer vision aspect of the drone. C++ will be used for the code that communicates more closely with the hardware of the drone.

Python specifically has its own set of coding standards called PEP8 as mentioned in earlier section. This standard has been around since 2001 and helps Python projects be more consistent in nature since Python has an extreme amount of flexibility with how to type out code. This was one of the cons when choosing a programming language because with Java at its base level makes you follow certain requirements to perform different jobs needed. We knew we would have to have a strict standard to abide to in order to have it be very understandable by reviewers and other teammates. PEP8 standard helps to outline how to space, naming conventions, the layout, even when to not follow PEP8. These guidelines and standards could be compared to how there are writing standards for papers in college and so on.

While these guidelines/standards can go into extreme detail, **Table 23** shows a few of the rules that will be followed or recommended. By looking at the details of the naming conventions below it shows how easier it would be to distinguish elements like class, package, or functions. Benefiting the reader that has not touched it will provide ease of access for other members or future people observing. A technique we will use to make sure this is getting accomplished is having a teammate who does not code often try to read it over and follow what is happening so in the worst-case scenario that someone needs to come in and take over, they can do that with confidence. These provide a nice, neat guideline to easily scan and read and know where things are located because everything have a spacing convention. All the elements will be clearly labeled and laid out so there is no confusion on what might be happening.

Table 23: PEP8 Standards

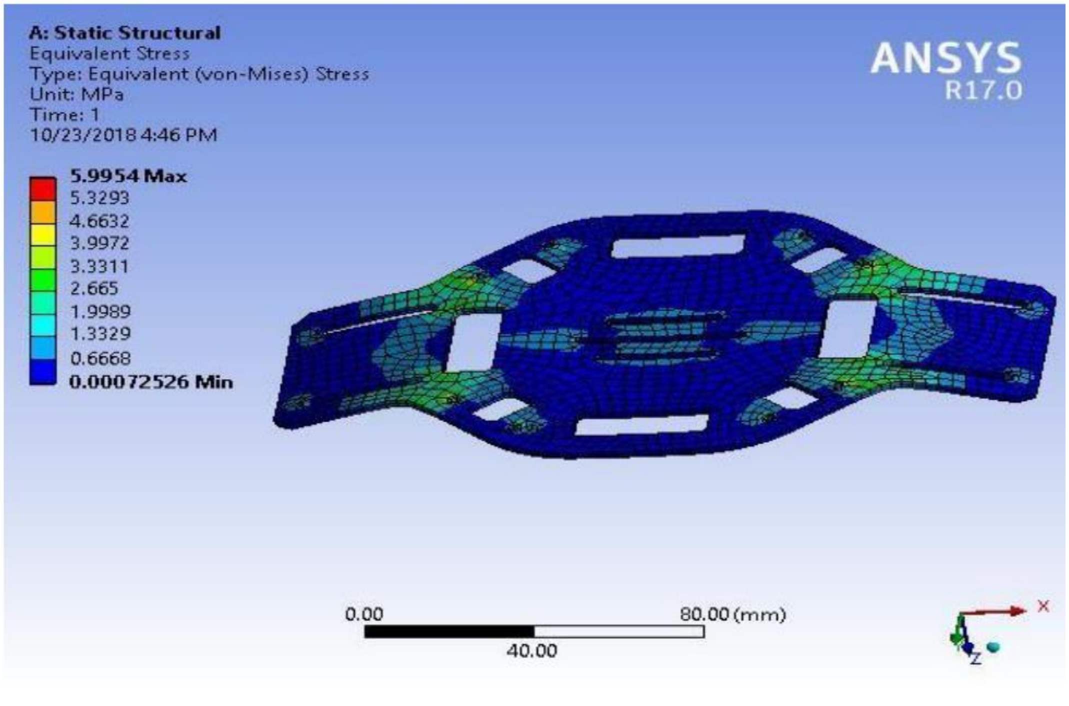
Item	Standard for PEP8
Function/Variable	Lower case, separate by underscores
Class	Start the word with capital letter, separate by underscores
Package	Short, lowercase word(s), DO NOT separate by underscores
Blank Lines	Top-level functions and classes with 2 blank lines; method definitions inside classes with 1 blank line; all other times – use sparingly
Maximum Lune Length	79 characters or less
Indentation	Use 4 consecutive spaces; use spaces over tabs
Line Break w/ Binary Operators	Recommended to break line before the operator, but can do after as long as all the code is consistent
Imports	Each import has separate line
Comments	Help with readability and be complete sentences
Source File Encoding	Use UTF-8

4.1.6 Battery Standards

Battery Standards for Lithium Polymer cells come from organizations such as IEEE and TSA, and even the US Postal Service. For the TSA we see standards based around their transportation and storing use when in planes. The TSA requires that batteries that have equal or less than 100-watt hours in their cells when transporting or storing so as to not reach a state where the battery could have enough charge within the cell to hurt passengers if damaged. Over 100-watt hours means one needs special permission to fly with these types. The US Postal service has the same kind of restrictions but for mailing and distribution of these types of batteries, with a further restriction saying these types of batteries cannot be mailed over air traffic but only upon land-based transportation. While the standard cell voltage of 3.7 volts is a “standard”, it is more of a physical standard than an organizational imposed one. The Lithium Polymer battery has more range than a set voltage. The 3.7 volts are the optimal and also usual voltage of battery cells in a LiPo. When a cell is near a complete discharge rate it is more in the range of 2.0 volts, and when fully charged it reaches 4.2 volts usually. However, both of these states can cause damage to the actual cell, so the standard is to hold it at the 3.7 volts mark so as to provide the longest-lasting product. We can state that the size and outlook of the outer shell of a battery is not a standard that is upkept as there are many different styles of batteries appearing, as well as many different styles of cell counts to different batteries.

4.1.6 Frame Standards

Figure 34: Tensile Strength of Baseplate under 20N of force



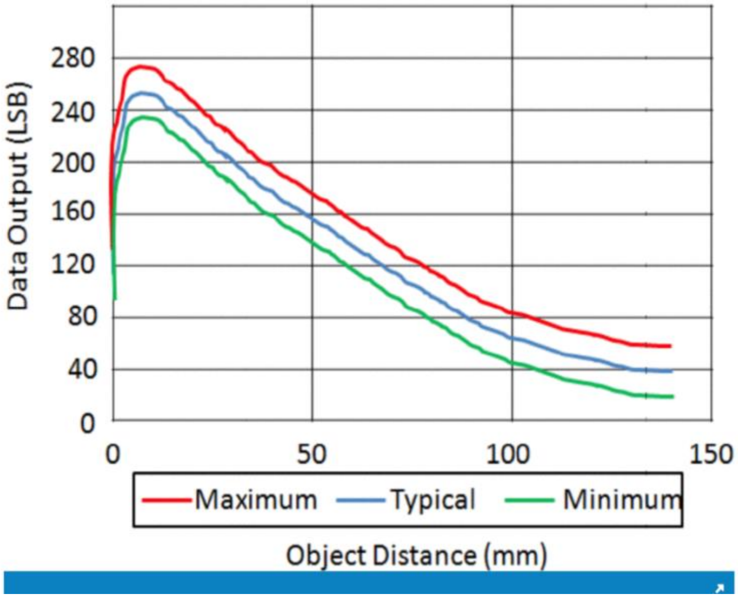
The frames we see in the different sections of this paper all have a generic outlook to them that shows there must be a type of standard to it. When looking deeper into the history and why this is so, we find the current frame is based around research done into the many different forces that would act upon commercial drones when in flight, such as gravitation force, deformation and stress upon the materials, and the weight of the drone and components attached led to the current design we see in many drones today. This is not a standard per se as better frames could come to the market, and we see there are slight variations to the frames themselves even in the frames we looked at in this paper, but through research done using a program known as ANSYS, which a simulation program used to stress-test different materials in differing ways, this frame style is the one that stood up the best. The branching out of the arms lets the mounted motors have more space and keeps the ratio of motor weight to propellor size to a more achievable quantity while giving room for components that need to be added into the drone for stable flight. We can see the tensile strength in **Figure 34** of the base plate of a standard drone frame base plate when put under 20 Newtons of force. This shows the generic structural analysis and how the interaction of the base plate and the arms is well within what the materials and shape can handle.

4.1.7 Sensor Standards

IEEE has been able to provide standards for years, but for some of their standards, they are not made publicly available and required to be purchased or subscribe to a subscription. They have many boards and committees that work on creating these papers and researching what would be the best to standardized and makes sense that they would need a way to capitalize on their product to keep the quality of what they are outputting and giving to people and companies.

One of the latest standards on sensors is IEEE 2700 which was released around 2014, but not released to the public. The description and overview of this requirement go over standards for sensors and how the performance should be, and these help with integrating with systems, like how we will be connecting them to a drone. The standard covers sensors including “accelerometers, magnetometers, gyro meters/gyroscopes, barometers/pressure sensors, hygrometers/humidity sensors, temperature sensors, ambient light sensors, and proximity sensors.”

Figure 35: IEEE 2700 Standard, Proximity Sensor



From researching the standard as much as possible it seems like it gives guides to how the sensors on machines would perform in certain circumstances and when set up a certain way will output stats like the below image. **Figure 35** uses the analysis of proximity sensors to show how at a certain distance it will give a max output of data, but then slowly start to decrease as the distance becomes larger. We will be using an ultrasonic sensor as one of our proximity measurements to detect objects in addition to a camera. Having these standards implement if possible, would be beneficial to know what restrictions we would be dealing with. For example, if the below figure demonstrated an ultra-sonic sensor then we know that the quality of data and detection after about 40mm starts to extremely decrease.

4.1.8 Transportation Standards

When it comes to standards on transportation in reference to drones, they are two major organizations. One of the organizations to help set the standards on transportation in the United States is the department of transportation otherwise known as DOT. The second organization that set standards on transportation for our project due to the fact that this is a drone delivery project is the FAA otherwise known as the Federal aviation administration. Now the FAA and DOT I'm currently in the process of beefing up standards around drone transportation because it is a fairly new revelation in the industry. Due to the fact that drone delivery is so new to the industry standards is still being created around it.

While the FAA does help create standards for drones, they are more worried about the airspace the drones take up rather than the transportation factor. So, the FAA is more concerned about making sure you as a pilot and the drone is also known as on manned aircraft systems or UAS for short are taking up the correct about of air space. The list of regulations that the FAA tries to enforce are as follows Do not fly over 400ft, or flying above groups of people, and above all else and the most important is to be registered and have passed the part 107 exam. **[15]**

The department of transportation (DOT) deals more specifically with the transportation side of our drone delivery project. One of the new standards the department of transportation has recently pass on drones is to have the drones have a form of ID attached to the physical drone. The reason that they wish to attach an ID to the physical part of the drone is that so if there are any damages or accidents it is easier to find who is liable for those damages that may or may not be caused. **[15]**

It is also considered to mention the department of transportation they only such regulations but also tries to push for certain agenda. One of which is artificial intelligence usage in drones. They incentivize artificial intelligence usage and drones by offering to fund and making the information easy and accessible.

4.2 Realistic Design Constraints

With building a new project and idea it is easy to get carried away with what features and items to include with it, but that cannot always be the mindset. When creating a product there are certain constraints to consider and think about how to product is impacted by the following factors in order to make it helpful for society, environmentally sustainable and maximize its features. For this section, we have broken down the different categories of constraint topics that were considered when designing this.

4.2.1 Economic

We are developing a product that is the best use will be bought in bulk and our target audience are distribution companies. While our current drone design budget is $1000, we expected to retail a single drone for $3000, and as you purchase more in bulk it becomes cheaper to produce, thus making the price of the drones fluctuate to become cheaper.

Table 24: Economic Budget

Item	Quantity	Cost
Drone Frame	~2	~$50
Battery’s	~2	~$20
Flight Controller	~1	~$200
Camera	~1	~$65
Blades	4	~$10
Motors	4	~$30
ECS	4	~$20
PDB	~1	~$30

Total		**~$650-$700**

In regard to our drone, we have capped our budget at $1000. With our budget cost entailing the drone frame, Batteries, Flight Controller, Camera, Blades, Motors, ECS, and PDB as seen in **Table 24**. With a relatively high budget, we will be sure to produce a high-quality drone with high-quality parts. With the goal of this product to be mass-produced, we believe the budget being $1000 will give us some wiggle room with other surprise costs that will come with such things as manufacturing and selling of the drone otherwise known as postproduction.

For post-production, we would like to invest in a sales team to consult with a distribution company to sell them on all the benefits our drones would be able to bring to their business model. With the estimated retail cost being $3000, we would want about $500-$600 of the total cost to go towards funding the sales team and to maintain good standing with our clientele.

Even with starting local, we still do have the vision to bring our drones to a global market. The potential for drone delivery in the highly dense community is limitless, as well as our product be trained and tested at this point, the retail cost in foreign markets would increase. This higher rate of $3,700 would also accommodate for the cost of shipping, international tariffs, the personal cost of going into a new market, and the design accommodation that may be needed for the new locations. Such as changing the user manuals and programs to use different languages specific to that country.

4.2.2 Environmental

When introducing any new technology, it is important to investigate its environmental impact. Especially technologies which are meant to be deployed at a large scale. Understanding the impact of our product will enable us to focus on the aspects that benefit the environment while simultaneously minimizing the negative impacts.

When analyzing the environmental impact of our drone, it is important to compare it to competing solutions. Our product does not exist in a vacuum. In our particular case, this means that packages will be delivered whether we deploy our drone or not. Last-mile deliveries are typically carried out by large trucks which run on diesel. In order to be considered a net positive for the environment, our product simply has to be less harmful than delivery trucks. As it turns out, this is not a clear-cut scenario.

In a study from the University of Washington, it was shown that delivery drones emitted less carbon dioxide than delivery trucks on the same route, even though the drones traveled a greater distance. This is due to the lighter design and more efficient electric motors. However, this effect was only seen on shorter routes with light packages. The opposite was shown when large packages were delivered over a longer distance. A similar study reported by the LA Times showed the same results. They explained that this is because drones can only carry single packages to a single location, whereas trucks can carry many more. This increases the efficiency per package for trucks. They showed that trucks are also more efficient in densely populated areas whereas drones are better for sparser areas.

A third study by the Smithsonian reports similar findings in terms of the effects of geography on environmental impact. They also concluded that the method of power generation for an area's power grid had an effect. Because drones rely on batteries that need to be charged, the power grid's efficiency was a major factor in the overall efficiency of drone delivery. In some areas with less harmful power generation, the drones had a 54% decrease in greenhouse gas emissions over delivery trucks per one kilogram of payload. However, in more harmful areas that use coal to generate power, the drones only produced 23% fewer emissions. This means that the environmental impact of delivery drones is dependent on where they are deployed. It also means that as our power grids become greener, the delivery drones will as well. This is good compared to delivery trucks that rely on diesel engine innovation to pollute less, which has remained relatively stagnant.

4.2.3 Social

Thinking about the social implications of delivery drones it can be thought of in pros and cons. The pros to having delivery drones are faster packaging and more efficient delivery from company to customer. There are cons that surround drones which provide constraints when approaching the design and style to them. Within the US specifically, many people value their right to privacy hence why drones are limited by the government and have extremely strict rules when it comes to flying in certain areas. Because of these constraints if we wanted to deliver in certain areas or places then we would have to research those social constraints otherwise the idea could fall flat.

From a customer standpoint, if people are to trust the product and trust that the drone's purpose is for delivering packages and not for spying then in our designs, we would want to keep it as secure as possible. Having an extremely secure drone will help customers feel like they can trust the product and company to do what is it intending and nothing more.

Thinking about the constraints with the software design, it will have to be held to a high standard of accuracy and correctness. These drones have the potential to crash into objects or people and potentially start fires if crashing in the wrong place at the wrong time. Because of these protocols will need to be considered for taking the best decision. Worst case scenario it might even need to make the best worst decision that causes the least damage.

These drones are meant to provide ease to everyday delivery and would be flying around everywhere. With constantly being around people a social constraint to consider is the noise effects. A lot of environments and social happiness comes from peace and quiet, but if the drone is obnoxiously loud then it will disturb the peace of some people. We will be trying our best to have a quiet drone.

4.2.4 Political

Now it wouldn't be very American if the government didn't try to legislate and tax every aspect of our lives, and drones are no different. To be able to fly these drones someone will have to be certified with an FAA part 107 exam which costs about $150 per exam. Also, because laws around drones are still new and not many tired in court, we and the company we sell to would want a lawyer.

The reason why you would want a lawyer is because of drones flying in residential areas and over people and roads. The chances of something unpredictable happening is high. They the law is currently written is that whoever the owner of the drone, is liable for the damages. There are also some areas that flying over is illegal such as theme parks, military facilities, and airports to paying attention to geographical location and path routes are important.

Even those these drones would be open to many liabilities we would take plenty of preventable measures to avoid litigation and to make sure we produce the safest drone feasible. Another concern would be that legislation hasn't completely caught up with the drone technology of today so the laws we face today could not be the laws we face tomorrow.

4.2.5 Ethical

Ethical livelihood is an important concept in the engineering world because of its implications when disregarded. Ethics are a set of "rules" or social cues we as a people are taught or inherently understand through living within society. For engineers this is slightly the same, however, there are certain aspects related to research and development and how we should ethically do these things as engineers that is known to a set style of rules. These include quality insurance so that when doing research and finding results your findings and techniques can be reviewed and recreated and thus prove to the wider engineering community in your truthfulness. Also, pro quo because of this outlook is the ability to have the moral "high ground" when doing research, thus providing reliability to your research and work. This is done by not testing on Humans, and only testing on animals when absolutely needed, hence when this invention or research directly affects human's health and lifestyle, thus such a level of research is needed and even then, this must be done as humanly as possible. And when all other routes have been investigated and human trials are the only route, agreements must be made between the researcher and the person to be researched about what is and what is not allowed. This is handled through paperwork and a transparent outlook on what is happening.

Ethics yet entitle not just the aspect of human health and safety, but economic and social responsibilities as well. Engineers, specifical ones within government agencies and places that have given access to civilian tax dollars have an ethical responsibility to not waste money and supply reasoning for why they are spending money are certain things versus others. This is to keep engineers accountable and not buy their way out of problems that crop up in research and development that could be solved using inventing and innovating.

Engineers need to post their findings in public discourse and with empirical evidence to show their honesty and freedom from bias. This format they submit their findings in must be to standard and show meticulous attention to detail so as to be completely repeatable and easily understood or as it is known, “dumbed down” to be better understood when given the opportunity. This format will also be completely transparent in the way this information was helped along by outside sources such as fellow researchers or graduates that helped bring this project to fruition so as to properly show gratitude were due. Citations from sites, links, and acceptance from sources to use their research must be given to be ethical.

In the terms of our drone and its references to the ethical outlook and need for research, there is an aspect of research into the guidance and accident-avoidance system that may need to be researched in the future with regards to how it interacts with human heat signatures and outlines, and such that is, this drone must be properly documented in the right format and transparently shown as to be easily taken up by future researchers that need to look into future aspects of this project. To further look into our drone and its potential ethical impact, we use many sensors to provide guidance and telemetry as well as use these sensors as the backbone of our accident-avoidance program, and these sensors could be used for non-ethical purposes if designed that way. To avoid this and continue ethically with the project, we will not be recording anything intercepted by the drone for future reference, so as to keep personal privacy to our topmost concern.

4.2.6 Health and Safety

Health and safety are paramount to any aspect of research and innovation. As such, products are always included with health and safety warnings whenever these two aspects can be affected detrimentally to a human or animal. These warnings include such things as drawbacks that come with using such products. These can be seen in commercials when pharmaceutical companies provide a list of side effects that can be incurred from using their product, no matter how minuscule the chance of happening, so as to be transparent. This same instance can be inferred through materials used such as anything radioactive, chemical, or potentially hurtful when touched or inhaled. Such things will apply to our drones and products like it. The materials we use to make our product can have adverse effects on livelihood, such as when manufacturing the completed product, or when taken to more in-depth analysis, the mining and acquiring of materials used can have adverse effects as well.

To show more of how health should be shown on product placement and usage, the aspect of using products can have adverse effects as well, such as the output of CO2 when using diesel or gas engines. For our product, this is not an issue as we will be using drones run by electric batteries and motors. This does not mean drones do not have impacts on the environment when in use, as the electricity to power this drone is needed from other resources on the power grid itself, but we can see that when taken to the extreme case we can label anything and everything as detrimental to the environment and users in general. So, for a professional outlook, we keep to closely related subjects that affect personal health directly in the use of our drone product.

Now when showing the positive increase in health-related aspects of our drone, we can turn to statistics and materials used. The drone product is much smaller and more easily produced than delivery trucks used by most companies, thus keeping more materials out of the environment instead of needing them to fulfill higher material needs for bigger products. Continuing with aspects that are positive improvements, through statistics we see a correlation to a decrease in suicides when taking the idea that there will be fewer retirements for delivery workers when there are fewer smaller delivery routes (as this would be the demographic used for our drone) thus decreasing need to be away from home and increase the mental health of delivery workers in response. **[2]**

Safety is an aspect of engineering that we as producers of products must take into steep consideration because it is our products and thus our actions that can affect the livelihood and safety of other people. This is shown in the way we process and package and generally affect our products. We must take into account the safety of not only the person, but the environment, property, and investments. We do this by applying warning labels and color-coded codes that show certain aspects of danger. Such as the radiological sign for radiation leaking, or the skull and crossbones for chemical aspects that could hurt individuals. We must also give out safety manuals to show the correct way to operate our products and instructions on if the events that if safety is compromised, the user will be able to help others and or themselves.

Our Drone will have an autopilot feature that will enable the drone to fly from pick up point to drop off point with little human interaction, and while that means little chance of human error, that means that there is a possibility for machine error. This is taken care of with our accident-avoidance software that lets the Drone avoid and dodge upcoming buildings and other obstacles in its way. This program will limit property damage, and the height level of flight will make it safer against human accidents, with the only chance being when the drone is taking off and landing. These instances will be shown in the provided safety manual that is our responsibility. This will provide the safety we must insure for property and human health. Regulations and more safety warnings are what will protect the environment, thus completing our aspect on safety.

4.2.7 Manufacturability

Manufacturability is an important consideration in any engineering project. This includes our ability to acquire resources, tools, and supplies needed in the manufacturing process. It also involves the speed at which parts can be made and the ease of assembly. This is important to consider because it ensures that our product can be feasibly used to accomplish our goal. If we cannot manufacture our delivery drone fast enough or reliably enough, then it will not be able to serve its intended purpose, even if each individual drone does meet all of our requirements and specifications.

For this project, many of the mechanical components are standard and can be acquired off the shelf. This includes the frame, motors, ESCs, and batteries. Some other components like the flight controller board are more specialized. For our project, we will be using commercially available solutions. However, at scale, it would be beneficial to create a custom flight control board. This would reduce unit costs and allow for more flexibility. Our drone also uses off-the-shelf sensors such as ultrasonic sensors and cameras. These sensors are connected via standard communication protocols like UART which allows for ease of integration as well as the flexibility to swap them for other sensors.

Choosing the Navio2 and Raspberry Pi as our flight control board drastically increased the manufacturability of our product. Having an all-in-one solution, as opposed to a flight control board and a computation board, allows us to integrate that piece into the drone much easier. It also makes debugging simpler. Overall, our delivery drone is relatively easy to manufacture. It has no specialty parts, and the parts are designed to work together well.

4.2.8 Sustainability

Drones have been proven to be very sustainable on paper. Because of the amount of technology available and the emphasis on the environment, if drones are not kept to that environmental standard, then they will not be sustained. When designing these drones keeping that constraint of the environment will help lead to more sustainability. These drones are predicted to be the future and help the environment as they leave extremely small amounts of carbon footprints behind. Making sure to have a clean charge source will be a constraint to carry on the sustainability because using renewable energy as the charge source helps with the longevity of the drone.

In order to keep it sustainable, it also needs to be built as stable as can be so that it would last as long and also one thing known with drones is crashing so making that to not impact the area it is in will help keep it flying as long as possible. Having as many components as possible with similar lifespans will help the lifespan of the drone overall because if enough parts fail at once then mass production companies might see it more cost-effective to throw away instead of salvage. This constraint is one that contributes to the design process of the drone and considering making parts that need repairs, easily accessible for quick fixes, and lower the carbon footprint of broken or dead parts.

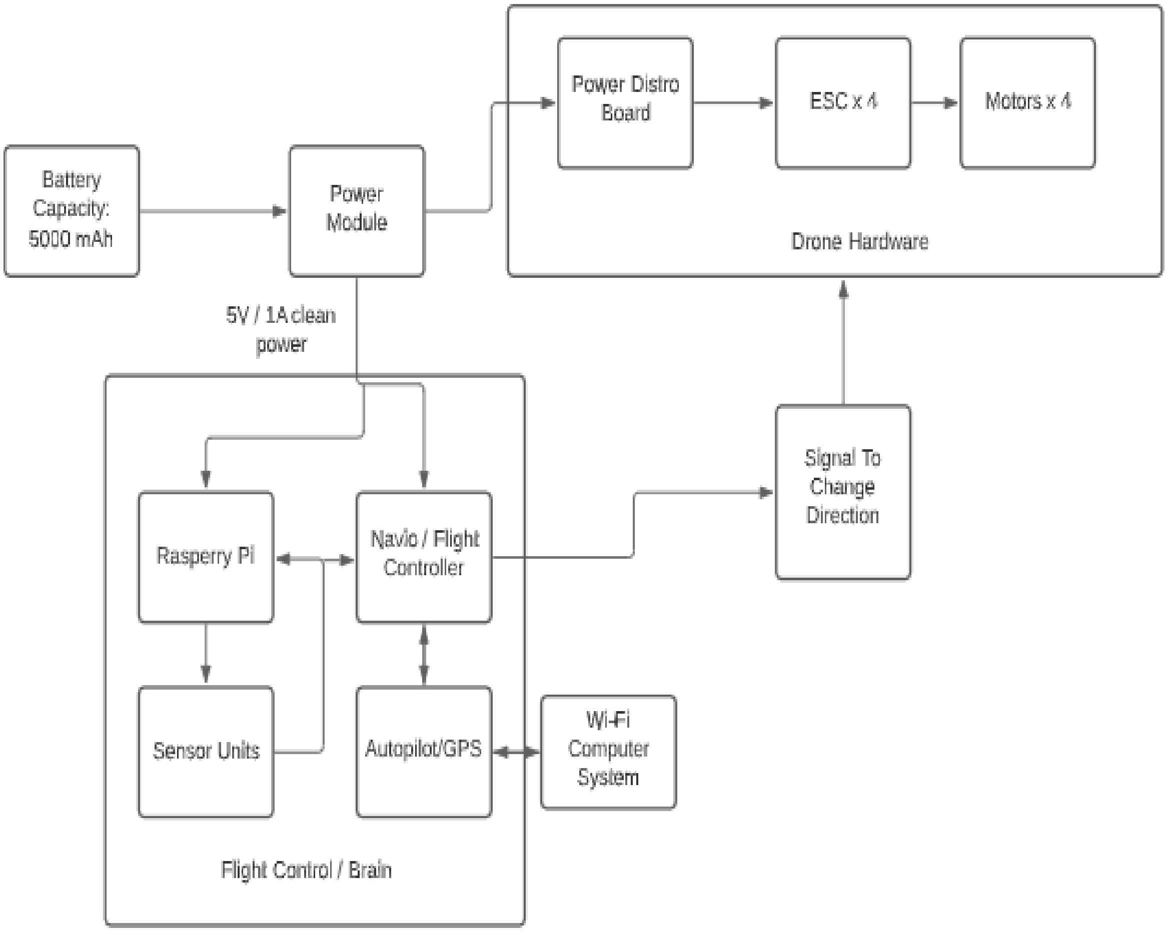
The software setup will require a lot more effort to ensure sustainability. Typically, Java is more known for having a better foundation of software set up for projects, which leads to less maintenance down the road. Despite this, we still went with the decision to use Python because of its ease of use and readability. We are hoping these factors will help make it more sustainable in the long run as long as there is time and effort put into setting it up the best way possible. Knowing our goal is to make this software have a long lifespan and easy update feature, it will require us to have detailed documentation within the code and not just writing it to work.

5. Project Hardware and Software Design

After diving into what goes into a drone and the components made up of it, there can be the start of design and decisions. This section goes into what decisions we made and how we want to design our drone in order to make it an efficient automated delivery drone. The hardware designs will go over how we will be connecting and securing everything into place and why we are using the parts we chose. The software will be outlined how we are going to be setting up connections and using the sensors with the software. These designs will be important to the records of the project as to see what we would need to do if starting from scratch. Designing is where the main decisions are happening with the unique choices we are trying to implement and explain how to recreate the idea we are trying to portray.

5.1 Hardware Flowchart

Figure 36: Hardware Flowchart

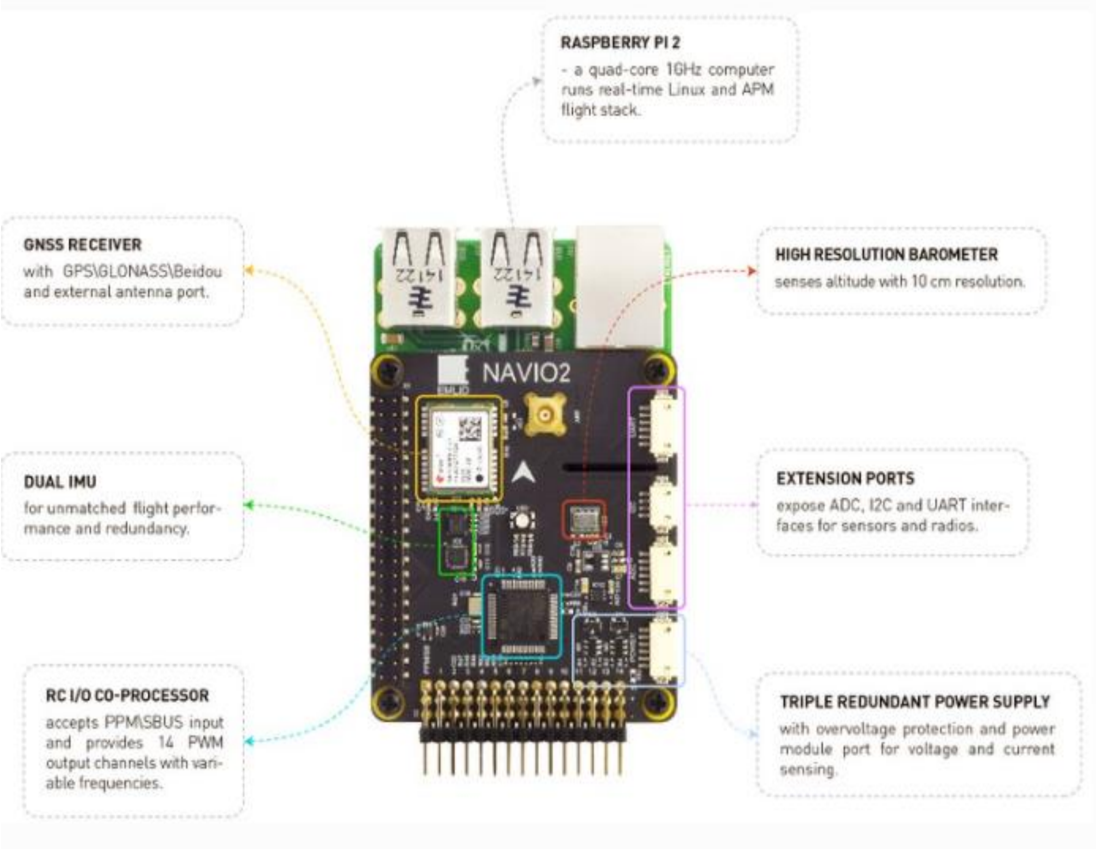


When we take into account multiple routes for data and power to flow throughout our system, we need to input a flowchart to simplify this measure and make it much easier for the users to understand where and how our thinking was when designing such a project. Here we see in **Figure 36** how we went about making our flowchart, including things like power lines and where certain currents and voltages will flow, to data lines and how our flight controllers and sensor readers will accept and send data to the rest of the system.

5.2 Controller Design

The flight controller will be set up with the raspberry pi 4 b and be able to communicate with the software that has machine learning and computer vision for object detection. The design will follow what is recommended in terms of what needs to be connected for proper communication. We will also be using the Wi-Fi setup as a way to communicate instead of Bluetooth or other options.

Figure 37: More Detailed Navio Functions



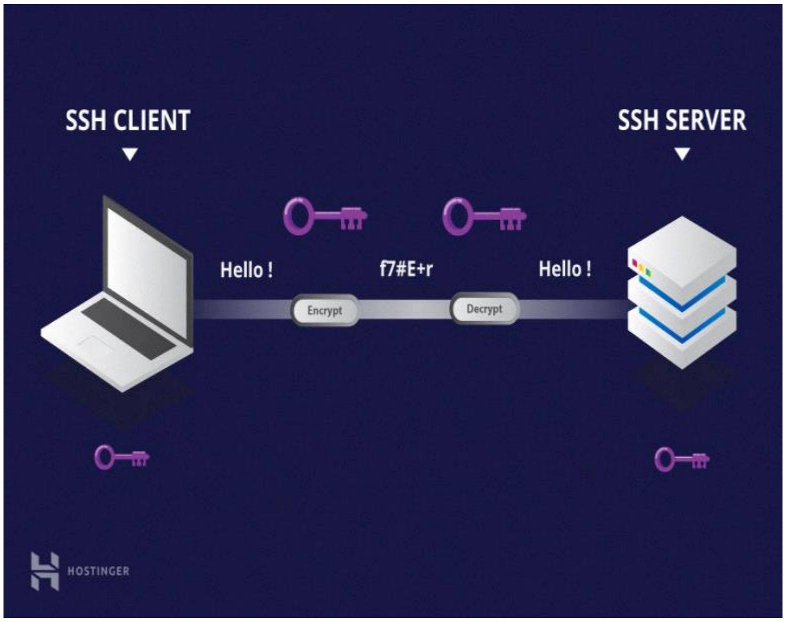
We have in the pin layout, that the total pin layout for our designated flight controller, the Navio 2, is very encompassing and can handle many different outputs as we can see. It also shows in the pin layouts to the left of the board as seen in the board itself that this board is easily integrated into the Raspberry Pi and will have no problems with having to share data between boards in heavy activity such as when camera vision is occurring, and the flight controller needs input to drive the drone. We also see the ease of understanding what is on the board, because of how under the board we have an easy-to-understand guide to what each pin is mostly used for. We have ease of power input with the USBs and see that our battery will need a power module that can convert from the leads given by the battery to USB format for ease of use and gives us future aspects of the board that we need to think about. We can see that 5 Volt pins are spread through the Raspberry Pi integration section of the Navio 2 board as well as 3.3 Volts, showing these two levels of voltage to be the most commonly used for this board and what we will need to ensure our power module that will be converting power form our battery to our board is attuned too. Further on, we can see the receiving and transmitting pins, which lets us make an easy estimate for where our pin layout needs to be and what pins are providing data to the flight controller when needed, thus giving us an easy to understand troubleshooting test for when or if we have problems providing and receiving data in between the boards.

We see in **Figure 37** that the Navio 2 is a very high-class flight controller that includes many different sensors included into the board such as a barometer and other such similar and more simple sensors, leaving the ports for more useful sensors that we will include. Our power USB ports also show that the Navio and Raspberry can share power input because of a designated port from the Navio 2 to the Raspberry Pi. This will simplify how many lines we need to run from the designated power module between the battery and the PDB and to the flight controller and the data processing board. Further, the built GPS receiver will let our autopilot system that will take input from our accident-avoidance system be very intuitive and more of a simplified process. We will not need to input our own system and be able to simplify our data processing and coding problems.

5.3 Wi-Fi Communications Design and Flowchart

Through hooking up the Wi-Fi component with ethernet and further downloads it will have its own Wi-Fi network connection, which will be used to download controls and other parts to it. Both our ground control laptop and our flight control board (Raspberry Pi 4 B) are equipped with a Wifi module. This normally this could be used to connect to an external Wifi network. However, we will be flying our drone in fields outdoors, where it is likely to bad poor, or no signal. This is not a problem, though, because the Wifi modules can be used in another way. They can be set up to actually transmit a Wifi network similar to how a hotspot works on a smartphone. We will set up a hotspot on the ground control station laptop and then have the drone’s computer connect to that wifi network. This means that even though we do not have a connection to the internet, the two computers can talk to each other wirelessly. After the two computers are connected on the same wifi network, we can remotely log into the drone via Secure Shell Protocol. This will allow us to securely access and issue commands to the drone’s computer without the risk of being hacked and taken over. The architecture of the secure shell protocol is shown in **Figure 38**. It works by encrypting traffic between two clients on the wifi network, so that any other computer cannot read the messages, even if they are intercepted. The secure shell protocol exchange will be used to send and receive MAVLink and ROS messages between the drone and the ground control station. This will allow us to control the drone manually as well as send it updated mission requirements. It will also allow us to receive telemetry data about the drone such as the battery voltage, altitude, pitch, and speed.

Figure 38: Secure Shell Protocol Architecture



5.4 Limitations

Limitations are something that engineering projects will always try to minimize but will never succeed in completely removing. There is a factor of any good product and can cause growth and ingenuity in the design of products. Here we work within our limitations for the future of our drone project and show that we can overcome such limitations with careful planning and ingenuity.

5.4.1 Low Power Mode

Power is not infinite and must always be regulated to ensure that proper coordination of our limited power supply is directed to its needed target and that we as engineers perform a style of triage to our own systems so as to know where and when our power is needed most for continued operations.

Now our drone will of course in the lifetime of its operation come to a point of battery life that it will enter such a state as low power. This cannot be avoided and is even encouraged to some degree because of our batteries style of design, and most others as well, that show that depleting a battery to the limit of its capacity and then charging it to fullness extends battery life to a figure worthy of this procedure is taken into consideration.

During operation, our drone will enter a low power state sometime during operation, and it will send a signal to ground control with its last known location as well as the notice of low power. It will not continue operating as it will objectively weigh itself a higher priority than the timely delivery of its package and will turn around and return to the known location of ground control. This is a standard procedure and will be followed and implemented numerous times during flight operations to ensure the program is in full working order.

5.4.2 Data Limitations

Bluetooth is a very low bandwidth style of data transfer. We will never be able to send large amounts of data through our ground control to our software system, namely our flight controller, at any reasonably fast speeds. That is a limitation that is being overcome by our drone's reliance on itself most of all. By giving our drone the ability to achieve decisions on its own and react in time to a fast-paced environment we are setting up our project for success within our limitations. When we need to supply our drone with direct data input from ground control, our drone will be a closer vicinity to ground control so as to limit data travel times and thus increase productivity.

5.4.3 Range Limitations

When talking about range limitations, we need to talk about not just Bluetooth and its limiting bandwidth, but the physical range limitations of our drone and how long it can run. We know from Bluetooth's own specifications that the 4.0 variation has a range of roughly 300 feet. This is quite a distance achieved and shows limitations that our specific ground control has to adhere to. We know then that our ground control can only achieve direct control up to 300 feet. Continuing past that range will have to be directly enacted by the autopilot system.

5.4.4 Background Interference Limitations

Technology needs to consider the traffic going on around it constantly. Even when using home Wi-Fi networks, it can have interference from things like the walls, other computers, or even the weather. That same concept applies to the technology and components being used on the drone. We are using a Wi-Fi network for our flight controller, which needs to be set up in a secure way to get rid of the interference from other people trying to connect or take over. This is especially important for the future of drones because, with package delivery, we need to prevent packages from being stolen, and with that comes securing the drone from background interference that might make it vulnerable to people taking over.

Other background interference the drone could face is weather conditions from the environment and extreme data traffic. Data traffic could include cell phone towers, data being sent through other internet signals, or different wavelengths in the atmosphere. We will have to make sure the drone is wired correctly to have a strong connection to itself and the computer attached to it.

5.5 Sensor Chassis Design

When it comes to the sensor design, we will be placing them to have max exposure to what it will be capturing. In order to securely do this, we will need to hold it down by having a chassis design, duct tape, or zip ties. To begin imagining the setup of the sensors we will go with temporarily securing them with duct tape or zip ties and if seeing it needs a more permanent home then using the resources available to 3D print a chassis that would screw into the frame.

With the drones it will be quickly moving at times to avoid objects and needs to be able to endure that jerkiness or even withstand the object it is trying to avoid. Luckily with the technology now a days it is more accessible to have 3D printers through colleagues and fellow students that would be able to print a design that can be securely attached to the drones. **Figure 39** shows an example of a bracket/chassis that is able to securely hold an ultrasonic sensor, which is what we would be using it for also. As it is able to see, there are holes for screws to attach it to a frame item and for the sensor itself, which will make sure it is held steady while trying to perform measurements and distance of objects.

We believe coming up with a chassis for a drone to be critical. The reason why it is so important for a drone to have some protection from the elements are as follows. If we do not have a secure design that would protect the inner workings of our drone the chances of a component being damaged during testing skyrocket. The reason it becomes more liable to not have a chassis is due to the fact if the drone is to crash, then come into contact with dirt and water other debris it could have a chance of ruining A component or two and then delaying progress in our project. That is the reason why adding a casing over the main components of a drone is so important.

Now when it comes to designing the chassis for how the components fit there are quite a few requirements we would like it to have. We do intend on 3-D printing our chassis. When 3-D printing a Chassis that will cover the components of our drone, our chassis needs to be durable enough to survive the crash but light enough not to cost too much drag and weigh down the drone. We intend to design our 3-D CAD of the chassis with 40% Fill. A 40% fill will do is save plastic, add structural integrity, while also not adding too much weight. What 40% Fill means, is that only 40% of the interior walls of the chassis will be implemented, just leaving 60% of the interior wall’s to be empty.

The next important decision that our group made about our chassis is what filament to print our chassis with. Now when it comes to 3-D printing filaments there are hundreds to choose from. The two most common are PLA and ABS. ABS filament takes a higher temperature for the extruder to properly lay down the ABS filament, but it also is more durable. The higher extruder temperature causes there to be more plastic fumes, harder to print with, and overall, less user-friendly. While PLA is more user-friendly, requires less ventilation and less harmful plastic vapors emitted, and is overall more user-friendly. This is an important comparison and needs to be carefully looked into. You can see these compared in **Table 25** below.

Table 25: Filaments

Filaments		
	PLA	**ABS**

Printing Temperature (Celsius)	190-220	240-270
Harmful Fumes	No	Yes
Price per Kg	$19	$21
User Friendly	Yes	No
Durability	Fair	Very

So, in conclusion our group believes a chassis will be important. The reason why will be important is to protect it from water dirt and other debris if it is to crash as well as protecting the longevity of the components inside the chassis. Our chassis will be made up of PLA filament printed at a 40% fill to make sure our chassis is light but also durable.

Figure 39: Ultrasonic Sensor Bracket

5.6 PCB Design

The PCB Design for our drone delivery project will play a critical role. The critical role in which we will design a personal circuit board will be to design a power distribution board.

The software that we plan to design our personal circuit board on will be the eagle auto desk. With the eagle auto desk, we will be able to upload libraries from Webbench. Webbench is a website that has already planned out schematics that we will be able to use for our drone delivery project. Our power distribution board will need to be able to take the power delivered buy the battery and distribute it to 4 different ECS evenly and safely. The way we intend to do this is by using voltage regulators schematics from Web bench and implementing them in a functional Design on Eagle.

Once we have a functional cad design on the Eagle of our four-voltage regulators from the web bench we will then proceed with manufacturing our personalize circuit board. When it comes to manufacturing or personalize circuit boards there are a few options for us to choose from. Our values on which we make our decision efficiency, cost of shipping, and how long it may take to ship. As you can see in **Table 26** below the companies we might potentially pick from.

Table 26: PCB design

Company	ETA arrival	Cost	Shipping	Total	Location
PCB Unlimited	4 Days	$23.64	$34.53	$58.17	China / Taiwan
PCB Way	4 Days	$38.00	$19.00	$57.00	China
JLCPCB	3 Days	$23.50	$32.30	$53.80	China

As you can see in **Table 26** above. The companies above are the ones in which we are comparing to see which would be best for our personal circuit board. The companies in question are PCB Unlimited, PCB way, and JLCPCB. PCB Unlimited has the option of manufacturing in China and or Taiwan with the total coming out to be estimated at about $58.17. The total for PCB unlimited comes from a shipping cost of $34.53 and a manufacturing cost of $23.64. PCB Way is manufactured in China with a total for the personalized circuit board being $57. The total for the personalized circuit board from PCB Way comes from a shipping cost of $19 and a manufacturing cost of $38. JLCPCB has its manufacturing done in China and the total for our Personalize circuit board would be $53.8. The total for our personalized circuit board from JLCPCB comes from a shipping cost of $32.3 and a manufacturing cost of $23.5.

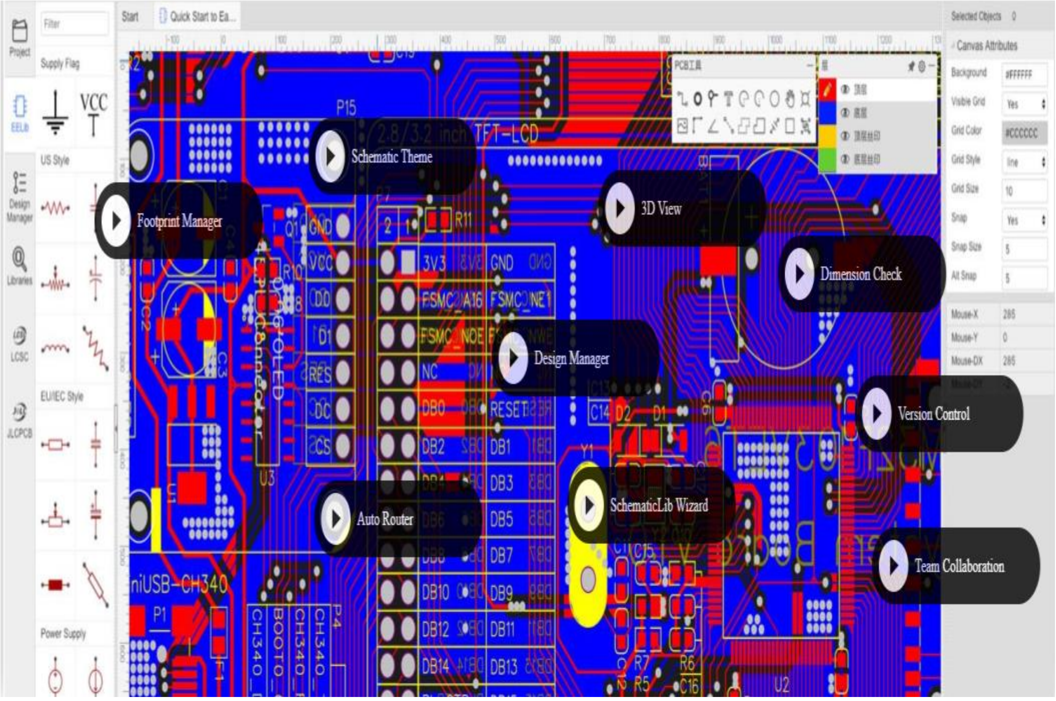
Our group believes the company that would be best suited for producing our PCB would be JBCPCB. The reason why we believe JBCPCB to be the best company is for a variety of reasons. First of which they have quite competitive prices. The second reason is if they have quite an extensive list of components that would make designing our personal circuit board quite convenient. And lastly, the expected ship dates for a personal circuit board aren't excessively long. Our group also understands that in the day and age of COVID-19 that manufacturing dates and ship shipping dates may be delayed.

5.7 Manufacturing Design

For our manufacturing, we have needs for our project that need to be met for us to choose a certain company over any other. After looking through the companies that we go over in our PCB section we have come to realize that our project has certain requirements that other engineering projects wouldn't have. Such as limiting a budget that's smaller than even some of the smaller engineering projects that some companies do. Another is our time frame. This is more similar to other professional projects but is still tighter than usual, where we have much less time for our parts and needed materials to come in so we can start testing and debugging. We need a company that can handle our timeframe while keeping to our budget.

We talked earlier on how our company we were more favoring was JBCPCB, and that because of the short shipping times as well as the ability of their ordering program being well reinforced by their own design program called EasyEDA as shown in **Figure 40** that saves on time because when we design our PCB the program will cross-reference parts and make sure we don't order parts that aren't actually available, which is a very annoying problem many engineers face. Luckily our PCB is a very simple concept for our needs as power distribution boards are simple in design as they only need to route power to certain areas and may be lower or increase the voltage as needed. We do this with capacitors and resistors and the simplest parts, so our PCB will be a very simple design easily implemented. Our board will be most likely the standard 1.6-inch-thick board and have copper power lines interlaced with voltage regulators to keep voltages where they need to be for certain parts and keep the design within scope.

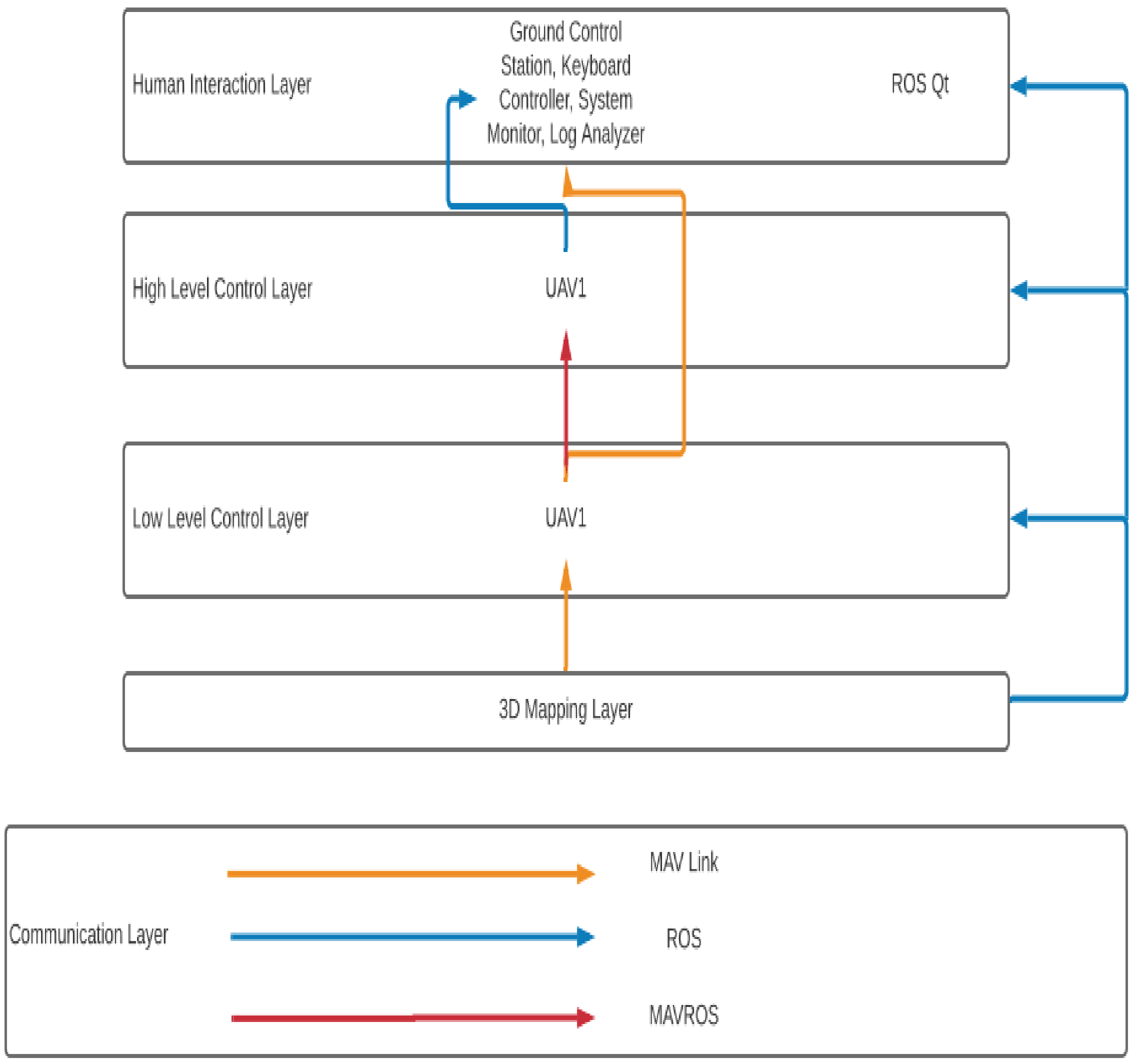
Figure 40: EasyEDA Program for JLCPCB Manufacturing



Selected Objects 0
Canvas Attributes
Background #FFFFFF
Visible Grid Yes
Grid Color #CCCCCC
Grid Style line
Grid Size 10
Snap Yes
Snap Size 5
Alt Snap 5
Mouse-X 285
Mouse-Y 0
Mouse-DX 285

5.7 Software Flowchart

Figure 41: Software Flowchart



Our entire software stack is built on top of the MAVLink, ROS, and MAVROS communication protocols, with each one being used for different areas of the communication, shown in **Figure 41**. Our low-level control layer is running PX4, which takes in information about the drone such as orientation, rotation, and forces on the drone, and uses that to keep the drone in stable flight. Additionally, the low-level control layer is in charge of taking in GPS data and waypoint mission information and calculating the adjustments to motor speed in order to navigate to the correct location. From there, the next level of abstraction is the high-level control layer. This is a ROS node that basically acts as a router, taking in information from the flight control software, the 3D Mapping Layer, and the Human Interaction Layer, and then routing it to the correct destination. The low-level control layer sends telemetry data to the high-level layer. The Human Interaction

Layer sends user commands and mission data to the ROS node. Lastly, the 3D Mapping Layer sends the point-cloud information to the ROS layer. From there the ROS layer routes it to the correct software that will be using that information.

5.8 Manual Pilot Controls

Since our drone is intended to be fully autonomous, we will not have very much of a need for manual controls. However, during testing, it may be useful to be able to pilot the drone for certain things like positioning at the beginning of a test and returning home after a test. With such minimal usefulness, we do not feel that it warrants spending the extra money on a transmitter and receiver. Fortunately, our ground control station implements a virtual controller, so we will not miss out on any functionality. We have two options for this. The first is using the virtual joysticks as seen in **Figure 42.** This will require a touchscreen on our ground control station software.

Figure 42: QGroundControl Virtual Joysticks

Luckily, QGroundControl supports iOS and Android, so we will be able to use any of our phones. However, by doing this, we would be giving up useful features such as telemetry logging. We do have access to a laptop with a touchscreen, however. The other option would be for users to plug in a USB Joystick or Gamepad. This could be achieved with hardware that many of us already have, such as an Xbox Controller. By using either of these two options, we would be able to circumvent the need to buy expensive hardware that has a limited effect on our end project. However, it is important to note that these controls are not as responsive as using an actual radio transmitter. This is because the commands are sent over MAVLink, which shares the bandwidth with the rest of the commands and telemetry data. Additionally, MAVLink commands are processed by the flight control software, which takes processing time. On the other hand, the manual RC transmitter would be routed directly to the motors, avoiding the processing delay, though it is small.

5.9 Final Parts Decisions

The final parts decision section of this paper well in detail, explains how we came to the final decision for the parts. The parts discussed in this section will ultimately be purchased for our project. We believe these parts will produce the highest performing project we are capable of producing. As well as detailed explanations as to why these are the best parts to achieve the best possible performance out of our drone delivery project. Some of these decisions came from the decision of other ones because the parts interact with each other for proper performance. This is further explained in the decision of the parts.

5.9.1 Propeller Decision

When it comes to choosing propellers, it is no small or easy task and is very calculated. Propellers are a critical component of the drone because they are directionally correlated with the amount of lift you can achieve with your drone. There are a lot of reasons why you might pick a certain propeller over another.

The first reason why we choose the propellers that we chose is due to the size of the frame. Once we had chosen the size of our frame, we knew we needed propellers that would provide a large enough lift for not only our frame but also our package. Now with a larger frame, you are able to install larger propellers. If we had chosen a smaller frame and larger propellers the radius of the propellers would intersect, and it would be impossible to get the drone off the ground. With the drone frame we have decided upon being 450 mm but for easier to understand metrics can be converted easily to inches with the end results being 17.7 inches. With a drone frame of 17.7 inches in length, we have some flexibility and how long the length of our propellers we want. The standard size of propellers for a frame of our size is somewhere in the range between 8 inches and 10 inches.

There are drawbacks to either having larger propellers or smaller propellers. With smaller propellers, we will have greater rotations per minute than our larger propeller counterparts. Now there are quite a considerable number of pros for having larger propellers. The first pro of having larger propellers is that we are capable of generating a larger amount of lift. A potential con of having larger propellers is that you have lower rotations per minute because you are rotating a larger mass when you have larger propellers. Lastly, it is believed that larger propellers are more beneficial to a drone’s battery life than its smaller propellers counterparts. With all of this information, our group has decided that we would much rather have larger propellers than smaller propellers. The length of propellers that we believe would be the best fit for our 17.7-inch drone frame would be 10-inch propellers. 10-inch propellers would have a 5-inch radius leaving 7.7 inches in the center of the frame to build upon.

Once you have the size of your propellers decided there is a final decision to make. The next decision to make about propellers is what material you would like your propellers to be made out of. The two most common materials propellers are made out of are plastic and carbon fiber. Carbon fiber is a denser material. With a denser material like carbon fiber, you will have less flex as the propellers rotate. Also, with carbon fiber being denser it will add more weight to your overall weight of the drone. Plastic propellers on the other hand would be lighter but they would flex a little more when rotating. When doing a price comparison of carbon fiber versus plastic propellers, the carbon fiber propellers will end up costing more. Our group has decided that plastic propellers would be the best fit for our project due to the fact that they cost less while producing the same amount of left and generally speaking being lighter.

In the end, the result for a journey to find the best propeller for our drone delivery project. My group has decided that the best propellers for the best possible outcome would be 10-inch plastic propellers. The 10-inch propellors will provide the necessary lift and stability for the drone. These factors were considered important to the project because of the environments the drone would be flying through. These drones could even be flying next to each other and have the potential to knock into each other and cause fires, dropped packages, and broken drones. These 10-inch propellers will alleviate that risk and help it fly correctly. Sizes differences in propellers can affect how hard the drone needs to give energy off of the batteries and going too big could be unnecessary weight and cost of materials. We will be having different colored propellers on the drone, red and black. The different colors will help distinguish the set-up of propellers because certain ones need to turn one way and the others will turn the other, so having the two different colors will distinguish which direction they are rotating.

5.9.2 Frame Decision

We came to a decision about our frame through many looks into different aspects that would our continued efforts into the project, namely things like frame size for if the frame could even withstand the weight and flight style we were going to put it under, landing gear type because this style of the project at best would be considered “pre-alpha” for many industry-standard professionals and we do not have a designated location with prime landing and take-off spots so we will need a robust landing system to supply us with good potential take-off and landing success. Also, the frame itself has many holes integrated with the frame that let it be easily modified to fit our needs, for this we would most likely use them when having to input our sensor units and keep them stable and safe to our drone during flight, so we would need these holes and cabling to secure our vital components to the frame.

To continue, the frame also needs to compensate for its own weight that will take away from the lift, so anything the frame can get rid of and keep structural integrity is well-liked. Our drone does this by making the arms of the drone hollowed and instead of solid reinforced plastic, is honeycombed to keep structural integrity while reducing weight. We mainly investigated the frame size when looking for different drone frames when initially looking into what frame we would take. We did this by understanding that our drone would not be like other consumer drones that only have a certain weight they need to lift and only have so many features. Ours would be heavier than any consumer drone when we include all the other components like sensors, PDB's, Batteries, and other components, but mostly things that would not be on a regular drone system like our second brain system. Thus, when looking into frames we decided the only possible frame that could hold our system would be 450mm and above, thus thinning the search for frames. And with all these qualifications, we came to the final decision of the QWinOut F450 drone frame with landing skid gear, as shown in **Figure 43**.

The frame chosen has a 4-legged landing gear that sits a few inches off the ground so the mainframe of the drone is not immediately coming in contact with the ground, which is useful for when there might be a puddle or something on the ground and protects the equipment from the elements. It also has two distinguishing colors of 2 red platform wings and 2 white platform wings which will help tell which are the back ones or the side ones depending on how we decide for them. There is also a square-shaped middle part that extends down in order to store the equipment and make it compact. This was a pro when looking because it keeps the weight of the equipment in the center of the drone to try and not throw off the center of gravity. Overall, this frame is very simple and looks like a great frame to start with as a beginner drone group builder. The frame chosen will provide stability and a good layout of options to put the equipment being attached, the wings also have holes throughout so if needing to strap or hook it down to the frame helps for that security. The platform it comes with has a distinct area for the connectors of the motors, ESCs, and battery which help with the organization of the components to secure them into place and not be crossing wires and getting confused with what connects where.

Figure 43: Drone Frame of Project

5.9.3 Motor Decision

When looking and deciding finally on what motor we would be using for our project, we are looking into many specifications that could affect performance, and the motor is one of the key components of this drone that cannot be sacrificed in terms of trying to find cheaper yet worse designs. We had to have a motor that was powerful enough to supply our needed lift requirements and also be able to fit on our frame. Luckily, most frames come with a motor specification attached to them so finding the right classification of the motor was as easy as reading which motor our frame would need. Taking this to find our motor, was more of a challenge because different motors did the same thing but were less cost-effective because of outside factors that were out of strictly engineering outlooks such as shipping and other things. When we did eventually decide upon a motor it was not just because they met our specifications but because the materials came in a bundle deal to save on shipping cost. We bought a bundle that included not just the motor, but the frame, ESC, and all the plugs needed to operate these materials. We can see the exact motor we bought here in **Figure 44**.

The motor that came with the frame kit is an A2212 1000KV Brushless Motor with a current capacity of 12A/the 60s, 2-3 Li-Poly cells, and KV of 1000. These specs correctly matched our frame requirements and what would be needed to operate. It came with 4 orange and silver motors displaying the company name, QWinOut, and have the connectors with them. Having the motor be in a frame kit helped to alleviate the costs greatly because typically these are more commonly sold separately and quickly become expensive when having to purchase 4 of them. The frame kick comes with the 4 that are going to be needed for the frames and propellers. Also, has the correct connections ready to hook up and organize correctly to get the items to work together and start customizing how we will be communicating with the motors through the ESCs and software and such. We will be using the max and mins of the motor to test accuracy and use them as boundaries for the code controlling the motor.

Figure 44: QWinOut A2212 Motor

5.9.4 ESC Decision

Our ESC, just like our frame and motor, came in a bundle to ease the expense, however, this was one of the hardest components that needed the most rigorous inspection to make sure we were providing our drone with an ESC that could handle the amp and voltage output needed to give our motors the power needed to achieve proper lift. The ESCs are the things that will directly be controlling the motors speed and RPM as the flight controller itself just sends a signal to these components and then they send the actually needed input voltage into the motors. To further protect our system, this ESC comes with its own software, taking away our need to bring it up to operation and saves on precious time. It is protected from outside signals so that it can’t be interfered with by other RF sources, which will be paramount in the urban environment that we envision our project taking place. **Figure 45** shows the ESC that we have decided upon.

Our ESC came with the frame kit ordered through eBay and does not need any soldering to put together. It even comes in a protective black casing and the wires attached to it are encased in protective covering. It is a 30A brushless ESC with a continuous current of 25A and a burst current of 40A for up to 10s. These were the exact specs we were looking for based on the frame and motor decision, which was great to see it was included with the kit and it was going to fit the goals. This helps consolidate the work needed to create ESCs. All the wires are different colors to distinguish what they are connecting to and the purpose. Without those colors, we would have to improvise and probably find a way to label them. This will be used to control the motors and change direction and much more. It will be possible these might have to be replaced if damaged, but luckily, they are not a huge expense and common to replace. The ESCs also have all the information needed to find replacements for it with having the brand and specs of the ESC on the front of it. The biggest thing to note with the ESC is it is not necessary to go all out on the specs. The ESCs work according to the motor and if an ESC can handle up to 50A, but the motor only goes up to a maximum of 20A then that is all that will be necessary from the ESC. So, while it is important to get good quality products, staying within what is being used is important and will save a lot of money in the end also.

Figure 45: 2-4S 30A ESC

5.9.6 Battery Decision

For our drone delivery project, it is critical to get the appropriate battery for our design. The reasons why it is so important to obtain the correct battery for our drone delivery project are Weight, and the correct milliamp-hours (mAh) otherwise understand how long the battery can produce a current.

Now battery cannot be chosen at random they need to be very balanced towards your Objective. Due to the delicate nature in which batteries need to be balanced they often cannot be decided upon until other parts have been decided upon. So, to properly decide the correct battery you first need to decide what motors and ECS you will be implementing in your drone project. Since we have now made those decisions, we were able to narrow down our options for battery dramatically.

Now has been talked about previously the size and weight of a battery is very important. The size and weight of a battery for a drone delivery project are critical because if we were to hook a car battery up to our drone first would apply too much voltage and would have a weight in pounds so much that the drone wouldn’t be useful. So, if a battery is too heavy it will create too much drag for the drone and will have diminishing benefits because as the drone struggles with drag it will use more electricity in the long run. That's why the Weight and size of a battery are so important. The main way to gauge a battery's size is through the categorization of 1s-6s. With 1s being the smallest and 6s being the largest.

The most important feature and batteries are their milliamp hours. The reason why milliamp-hours are so important is due to the fact that more milliamp-hours are equal to more flight time. The longer the flight time the more you were able to do with your drone in between charging the batteries. Now the amperage draw also plays an important part in flight time. The equation that shows their relationship is as given:

(Milliamp hours/ avg amp draw) *60= amount of flight time in minutes.

This equation played a critical role in why we choose the battery that we chose. For the amperage draw variable we mainly looked at our ECS's and What was the max amount of amperage they could take /handle and calculated a milliamp-hour range that we would need to use in the battery that we chose.

These are the factors that lead us to choose the Turnigy nanotech 6000-mAh 3s Lipo pack as seen in **Figure 46** to be the Best option for are the drone delivery project. We believe the 3s to be the perfect size range for what we need with our battery and it not being too big or too small. Then we Calculated our milliamp-hour range to be around 6000.

Figure 46: Final Battery Decision

With the Turnigy nanotech 6000-mAh 3s Lipo pack battery chosen. We Believe this battery will supply our drone with the perfect amount of average to ensure steady and efficient flight. This battery will also not be too big to way down the drone. With these factors maximized we project our drone to have a much more sustainable flight time and to be more effective.

The battery is a rectangle shape that is enclosed in a blue rubber casing with the front having all the information of the battery. There is also a black, red, blue, and white cased wire that will help distinguish which wire is being used. The battery will provide the power needed for all the components and be rechargeable. We will be having to place these in a way that does not affect the balance of the drone either because of its shape with the frame in order to keep the drone fly steadily and efficiently.

5.9.7 Battery Leads Decision

When connecting our battery to our drone we have two methods to implement. The methods in which you can connect your battery to your drone are you either Sauter them together or you use connection leads. There are pros and cons for both, but my group has decided for a few reasons the battery leads would be better than soldering together.

The reason why we believe connection leads would be better is for multiple reasons. The first reason why we believe the battery leads would be superior to the alternative of soldering is due to the fact that you can plug and unplug the battery from the rest of the drone. The ability to plug in and unplug the battery from the rest of the drone is superior to having it constantly soldered. With the ability to unplug the battery you were now certain that there's no electricity flowing through any of the circuits or to the motors thus making it safer to handle. The safety benefit, as well as an easier-to-manage battery supply, is what led our group to decide to use battery leads instead of soldering.

The connectors we went with are orange in color and have a hard casing for the structure. This will make it durable with try to use the batteries and possibly connecting and disconnecting them. Having that sturdy case will keep the battery wires protected from elements around it and even the case itself as it has space between where it is plugged in and the hard shell of the connector case. Having the connectors provide ease and convenience to the team when trying to construct what is basically a prototype drone. In the future, the decision to solder might be more appealing if it helps to secure the battery better and we know that it will be perfectly secured in that manner for production. Solder does help to eliminate weight as the casings and connectors add some fraction of weight on, but for this project there cannot be much harm to do it and the for the right cost. Shown in **Figure 47**.

Figure 47: Final Leads Decision

5.9.8 Flight Controller Decision

The drone not only has mechanical hardware that it needs to operate correctly but needs proper processing power. We decide on a dual aspect of this component to save processing power and to give our drone the best chance of succeeding with all of the sensor inputs it will receive as well as when it needs to direct the ESCs to drive the drone. For the two-part system, only one of the boards will be used as the designated “flight controller” and that was decided to be the Navio 2. This board has all the simplistic design aspects that a young product could need such as simple wire design and ease of input of pins. This board is made to be in tandem used with the Raspberry Pi 4, however, that will use for a different aspect of control for the drone. The Navio itself will be used to directly control the ESCs during flight and intake the sensor inputs from the more simplified sensors such as the IR and accelerometer sensors. As these will just be parameters that will be read and interpreted to be either in the allowable values or outside them. In **Figure 48** we see the board we will be buying for our flight controller and the easy pin configurations that are shown on the board.

Figure 48: Flight Controller Board Navio 2

5.9.9 Data Processing Board Decision

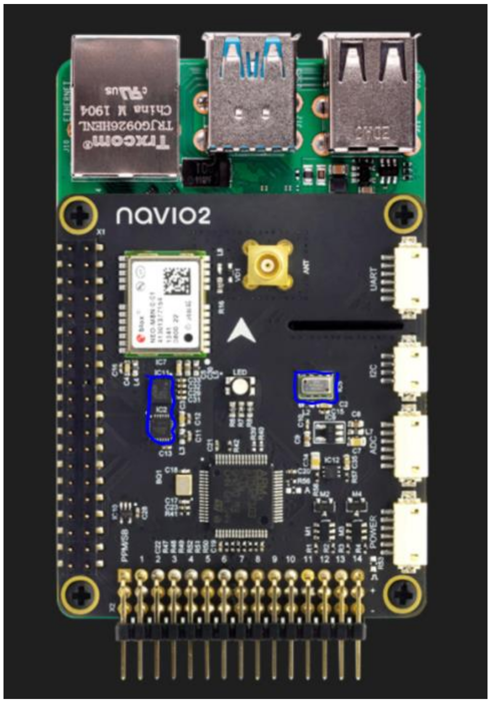
Figure 49: Data Processing Board Raspberry Pi 4b

As talked about in section 5.8.4, we acquired two distinct boards for our processing needs because of the intensity of our sensor inputs and decision-making needs. The flight controller was decided to be the Navio 2. However, the Navio 2 was made to be used in tandem with the Raspberry Pi 4. We decide on this configuration of Raspberry 4 so as to let the Raspberry Pi be used as our intense Data Processing Board. That is the designated title, however, what it really means is that our Raspberry will be used strictly for the data input and use of our camera sensor unit and the use of robot vision that using this camera effectively will entitle. This is a very processing-intensive program so will need almost the entire processing power of the Raspberry Pi which is why the Navio will be used for everything else. We see the Raspberry 4 and its ease of pin configuration and how it easily ties into the Navio 2 in **Figure 49**.

5.9.10 Sensor Unit Decision

The Sensor unit is actually many different sensors all packed together into one packet, but many are taken care of by being included in the processing boards we will be using for our drone, namely the Navio 2 with its accelerometers, barometers, gyroscopes, and finally an included magnetometer which we can see circled in blue in **Figure 50**. The sensors on the Navio are used to tell the drone how it is moving in space. The accelerometer, gyroscopes, and magnetometer collectively make up the Inertial Measurement Unit or IMU. This sensor unit gives the drone orientation, angular velocity, and specific force. Also, on the Navio there is a barometer that can detect a change in air pressure, telling us the current elevation relative to the launch zone. However, all of these sensors only tell us what the drone itself is doing in space. They tell us nothing of the environment it is in, which is fundamentally important to autonomous navigation.

Figure 50***: Blue Circled Sensor Packages on Navio 2***



For this reason, the Navio sensors are not the only sensors we need nor will be using. On the Navio is many of the basic sensors that are included on flight controllers, but our needs go beyond the usual as we will need accident-avoidance aspects as well. We do this through the use of external sensors. For collision avoidance, we will be using ultrasonic sensors facing in every direction around the drone. This provides us with data on the distance to the nearest obstacle in every direction. The way that these sensors provide the drone with distance information is shown in **Figure 51.** The transducer sends out an ultrasonic pulse.

After some time, the receiver receives the reflected signal. The time of travel is measured to calculate the distance to the object. We will use this to ensure that we do not get too close to an obstacle. This still only supplies us with a relatively simple view of the world. For the actual navigation, we will be using a SLAM algorithm to create a three-dimensional map of the world. Using a camera we will record images of the environment and use computer vision to process them. This will provide us with a point cloud as seen in **Figure 52.** This is simply just a collection of points in three-dimensional space where an object was detected. With enough of them, the map starts to look like the real environment. This map will tell us what areas are safe to navigate through and what areas are not.

Figure 51: Ultrasonic Rangefinder

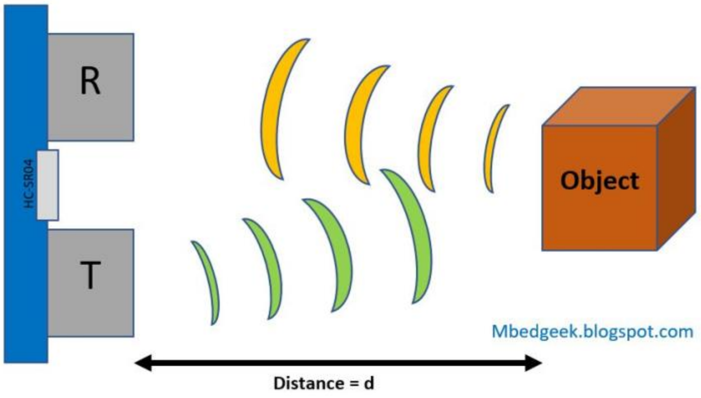
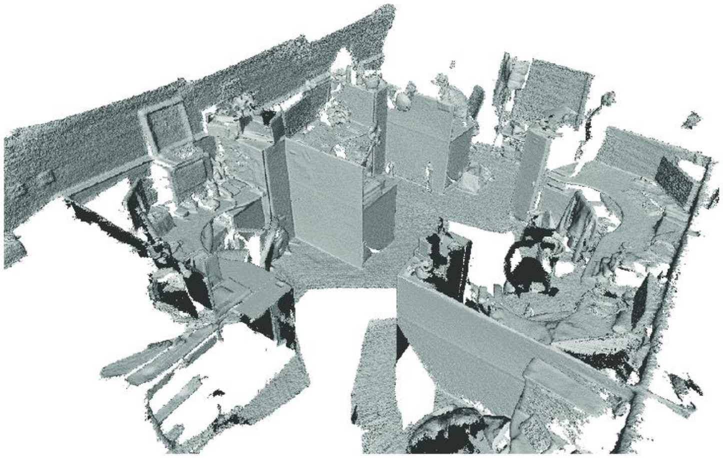


Figure 52*: Mapping Algorithm Output*



5.9.11 Coding Language Decision

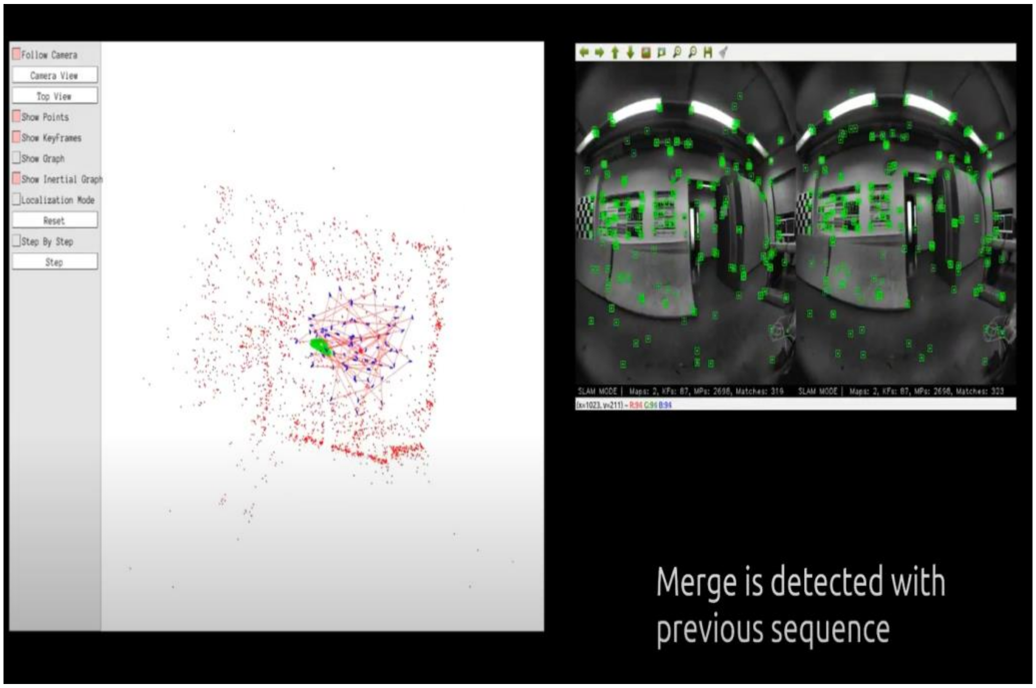
The coding languages being utilized for the drone will be Python and C++. Python will be utilized for the main software of the computer vision and controlling the drone. Python provides a style standard to follow and lots of documentation for completing projects of this magnitude. It is very popular in the machine learning and computer vision community for its feasibility of understanding and being extremely open-source.

C++ will be utilized for a lot of the code between the hardware and firmware and lower language items. This will communicate how to set the hardware to certain numbers of performance. C++ is an algorithm-based language with object-orientated programming. This allows us to have a lot of flexibility in terms of things we can do with it because it will adapt to our software programs or with communicating to hardware.

5.9.12 Mapping Algorithm Decision

After carefully reviewing the pros and cons of each of the mapping software's in terms of features, speed, computational demand, and accuracy, we have ranked them from best to worst for our particular application. Unfortunately, because our computational resources are limited and it is impossible to estimate how much power we can afford to allocate for the mapping program, we cannot make a concrete decision on which one we will be using. We are required to implement our setup and test it with various different mapping algorithms to see what gives us a good balance of speed and accuracy/features.

Figure 53: ORB-SLAM3 during operation

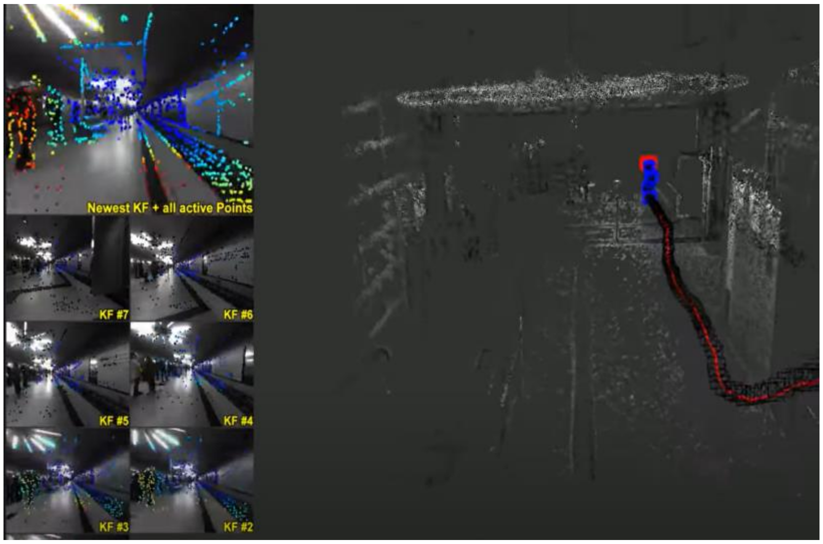


Merge is detected with previous sequence

Our top choice for mapping algorithms is ORB-SLAM3. This application is the most feature-rich of the algorithms we looked at. It is a full SLAM implementation, which means it includes features such as loop closing and key point reuse. These help to ensure that our measurements do not drift over time, making our map more accurate. It also includes multi-run map reuse which would reduce the computational demand in high traffic areas such as around the loading warehouse, as the map has already been calculated for that area. This implementation is also more robust to periods of poor visual information. Whereas another algorithm could become lost if something like this happens, ORB-SLAM3 would allow us to pick up where it got lost and continue on with the mission. While these features are nice, they are not strictly necessary, and they do come at a cost. ORB-SLAM3 is significantly slower than some of our other options, which could limit the speed at which the drone can fly. For these reasons, we will attempt to use ORB-SLAM3 as our mapping software. In the event that it does not run fast enough on our setup, we will downgrade to the next best option. We see ORB-SLAM3 in **Figure 53** in a working capacity.

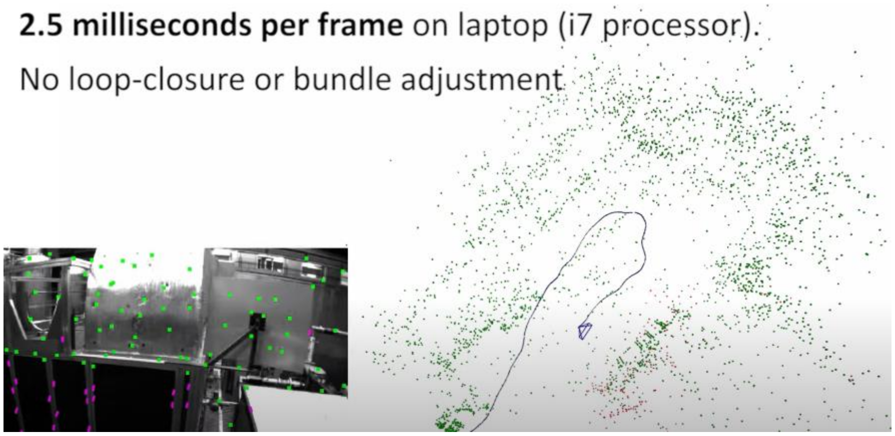
Our second-choice mapping algorithm is Direct Sparse Odometry. The base level of the algorithm is an implementation of Visual Odometry, which is not as fully featured as SLAM. It does not share some of the abilities such as loop closing and key point reuse. However, it is significantly faster than the ORB-SLAM3. Even though it is not a full SLAM algorithm, it is still more accurate and robust than the other options we looked at. The reason that this is our second choice is that it also comes with extensions such as IMU integration and loop closing. This will allow us to customize the feature set to achieve the perfect balance between efficacy and speed. If a certain feature does not appear to be worth the performance decrease, we can simply take it out. This flexibility will be very helpful in achieving a working final implementation. We see in **Figure 54** Direct Sparse Odometry in a working capacity.

Figure 54*: Direct Sparse Odometry during operation*



Our last resort option is Semi-Direct Sparse Odometry. Again, this is an implementation of Visual Odometry, not SLAM. While it is not as robust or accurate as of the previously discussed options, it is significantly faster. It is an order of magnitude faster than ORB-SLAM and two to three times faster than the base implementation of DSO. This is our last resort because it is not as good as any of the other options, but it is faster. If we absolutely cannot run anything else, this algorithm will be better than nothing. LSD-SLAM on the other hand was completely rejected as an option. It did not work as well as the other SLAM options in terms of accuracy or robustness, yet it was even slower than both of the visual odometry implementations. For this reason, this is the order that we will attempt to integrate the mapping algorithms. If a particular algorithm is too slow to run, then we will simply move on to the next best option. We see in **Figure 55** the Semi-Direct Sparse Odometry in a working capacity.

Figure 55: Semi-Direct Sparse Odometry during operation



6. Testing Detail for Hardware and Software

Testing is imperative to product development because of how it prevents failures. Sometimes failures can cause a whole machine to be destroyed but testing the software and hardware components can show the flaws before interacting with the whole system. Section 6 goes into the details about we will be testing the components for functionality and making sure the items are performing to our objectives and specifications before doing full launch tests. Then we will be going over how the final tests will be completed.

6.1 Hardware Testing

Our group believes in doing thorough hardware testing to ensure each part performs as it should. The reason we believe thorough hardware testing is necessary is for safety and to make sure we don’t accidentally ruin any of our parts unnecessarily. Our group believes in measuring twice before testing once. What measuring twice before testing one entails is as shown below through calibration testing start-up sequencing testing and field testing in multiple testing in between to make sure nothing is out of order or unsafe for us to continue before we eventually test our parts.

6.1.1 Calibration Testing

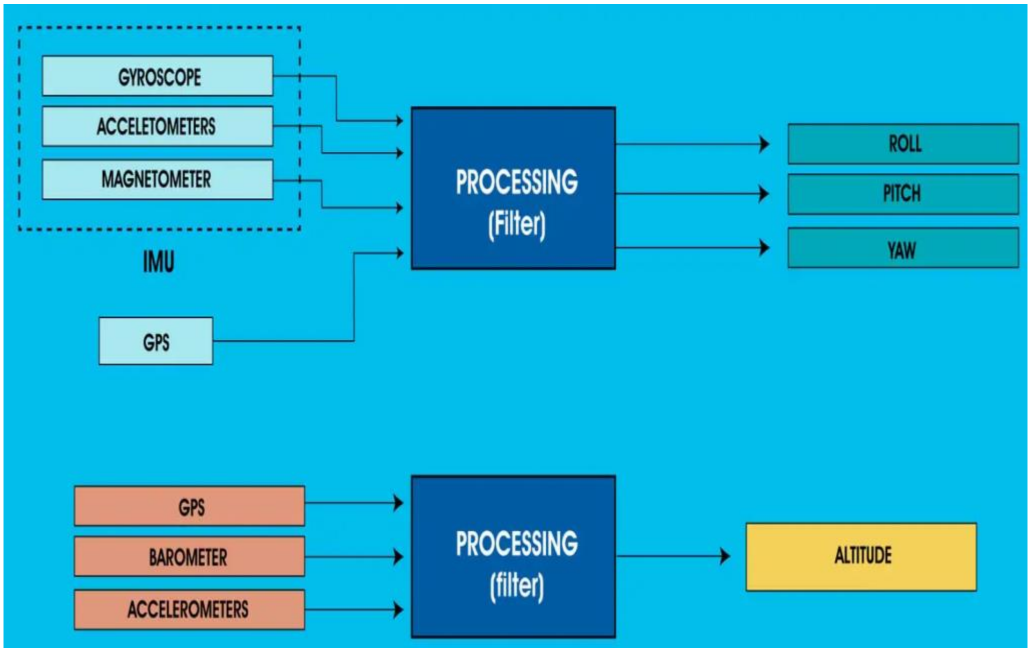
Calibration testing on a drone is critical. The reason in which calibration testing is critical for a drone because without the correct calibration your drone will not be able to level itself. As well as not being able to orientate itself in the correct direction if it is not calibrated correctly. From my hardware ‘s perspective, there are three points of failure to make sure are calibrated correctly. Those three points are the power distribution board, ECS, and flight controller.

For the power distribution board, it is important to calibrate correctly for a few reasons. First of which is that all the components on our drone are voltage sensitive. If our power distribution board were to be distributing the incorrect number of voltages to individual compartments they would not be powered correctly. As well as be at risk of overheating and frying themselves. For these reasons, it is important that our power distribution board be calibrated correctly.

The next piece of hardware to calibrate correctly are the ECS. The ECS Job is to take input from the flight controller and to distribute the correct amount of current to the Motors respectively so that the drone can orientate itself correctly. So, if the ECS were not calibrated correctly the drone would be unable to be balanced and hover as well as maneuver at all. To ensure that the ECS is calibrated accordingly we will be sure to test them to make sure we are getting the same input out of all four ECS.

The last piece of hardware is the flight controller. The flight controller's calibration is critical. The flight controller uses a gyroscope to make sure that the drone is level. Not only that but the flight controller is also responsible for being the main form of communication to the ECS and the motors to make sure that the drone can maneuver properly.

Figure 56: Calibration Stylized Testing



For all the reasons listed above is why calibration is important in the way that we plan on testing it is through trial and error. As well as taking careful measurements before officially testing it to making sure that each part of our drone is calibrated correctly and doing its job properly before pressing the start button. We believe this method will help our drone overall be more calibrated and balanced and ensure quality testing.

We see in **Figure 56** that calibration is needed for the system to understand certain variables and immensely helps other programs such as our flight controller, where if we didn't fully calibrate our sensors such as the gyroscope and the accelerometer and magnetometer, our flight controller would have much more difficulty flying using such parameters as roll pitch and yaw which can lead to very harsh flying conditions and even to crashing if the parameters are too inaccurate.

6.1.2 Start Up Sequence Testing

Our group is a firm believer in measuring twice before testing once. What this entails for the startup sequencing, is that we plan on testing all the hardware individually as well as setting up to make sure all resistances, capacitances, and voltages are as they should be before turning it on. For example, when the ECS arrives we will take a multimeter and an oscilloscope to all parts of the board to make sure we are getting the correct inputs and outputs throughout it before our formal start. This procedure will also be used for the battery, power distribution board, ECS, and Motor.

There are a few reasons why we would want to be this thorough with designing our drone project. One of such reasons is that when we go step-by-step and check each part individually, we gain a better understanding of what that component is doing and processing and how it should function properly. With that information, if an error or a mishap were to happen, we would then be able to diagnose the problem sooner and more accurately.

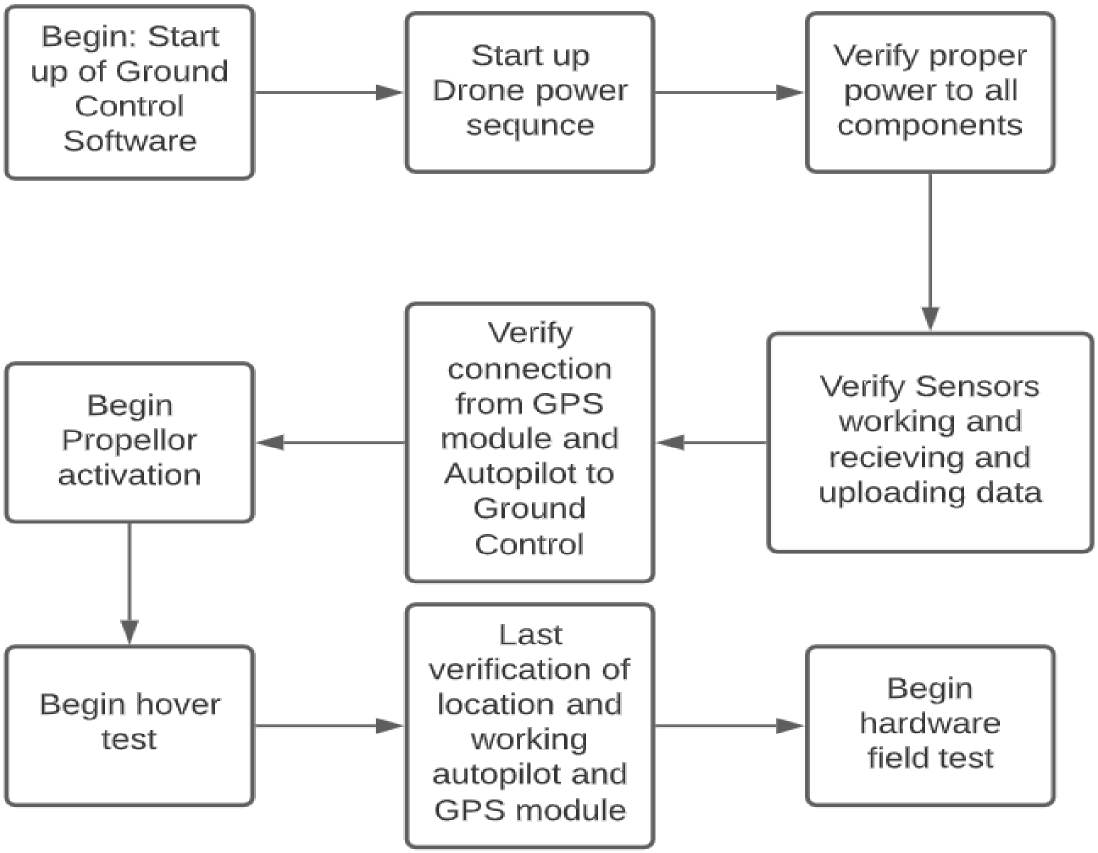
Once each part of our hardware has been tested to make sure is operating properly, we will then hook them up together in the order in which they are intended. Then test again to make sure we are getting voltages, resistance, and capacitance through each step of the drone and that everything is running smoothly as it should. Then capacitance the results to make sure they line up with our original result. After this procedure has been completed and we have tested the hardware twice. Once individually and once as a collective we believe it will be safe to turn on the drone.

With these extra steps in place, we believe the first time we set up the drone we will have a successful start. On the off chance we do not have a successful start with all of our testing beforehand we hope to now have a better idea of where to go for the next step to get our drone operational.

We see in **Figure 57** that our flowchart shows the intended sequence we will be following for further testing when performing field tests and we will be applying this style of testing for Senior Design 2 where we will have a working drone and sensor configuration. This will enable us to have a stable and routine start-up sequence while applying safety to all field tests so as not to damage the drone and hardware components on the drone itself, while also verifying that all data inputs are actually receiving data from surroundings and not facilitating a failed and useless testing environment. We can see from our flowchart that our start-up sequence will go in the order of Ground Control activation, namely our QGroundControl Program that will be facilitating our "talking" to the drone and all communication for route control and verification of completion of objectives such as testing or dropping off a package.

Moving on, we have the activation of the drone’s power system, or in other words the activation of the hardware of our project, with a further step of verifying that all components of our drone are receiving power and there are no broken or damaged lines in our power distribution system. This is needed for any further testing or processes in our start-up sequence because if even one of our sensors isn’t working it invalidates all the testing that can be done because of loss of data streams that could enable and proper test.

Figure 57: Start Up Sequence Flowchart



After we have verified all components are working and receiving power, we verify all sensors are actually attuned and ready for testing, such as the ultra-sonic sensor is reading distances properly and the camera vision is properly on and detecting certain images we will prepare such as a picture of a raccoon other objects to verify that it is reacting and reading properly.

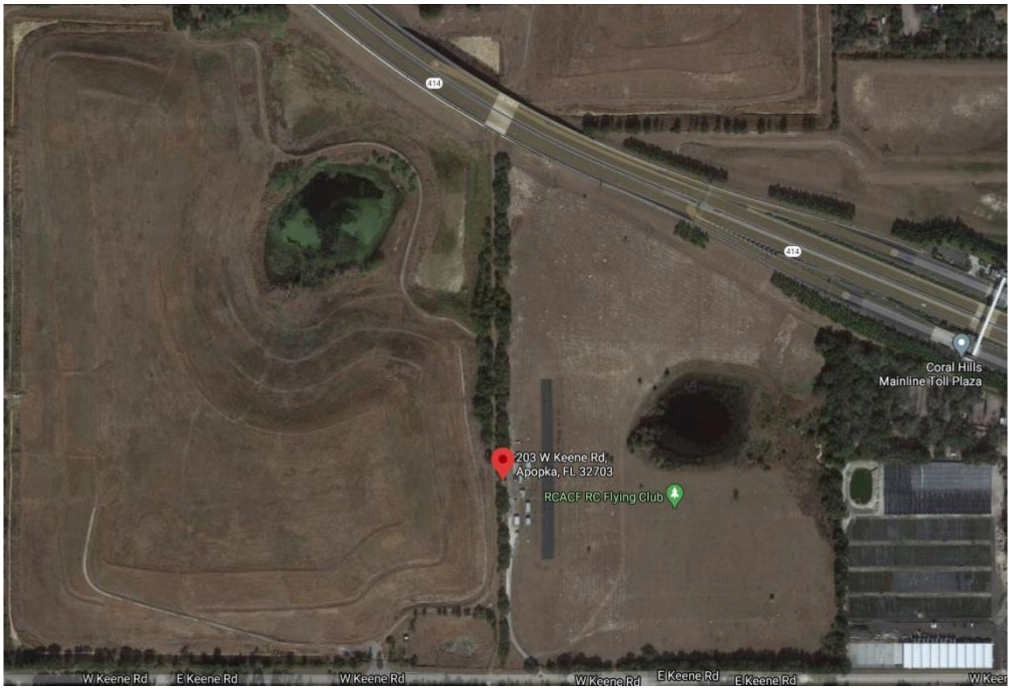
To continue with the start-up, we have the verification of the GPS system letting the drone and us operators wherein the location it is currently and verifying this is right and accurate to the level the GPS module can achieve. This will help and begin the verification of the autopilot system. Showing that our autopilot has properly uploaded the correct route to the location we have routed for it and it reads the correct location via coordinates for the drop-off point.

Finally, we begin the final processes that can be done before a proper test can be fulfilled. We start up the propellors and verify all systems are still working and GPS is still receiving data and sending data even during the vibrations of the propellors are causing on the frame. We then achieve lift-off and enter our true testing phase. We verify the third and last time that the GPS and autopilot systems are interacting properly with the Ground Control system and that all sensors are working properly and finally begin a true test of our drone accident-avoidance system.

6.1.3 Field Testing

During our testing, we will eventually reach the stage of quality in data that the only way to extrapolate more will be to do a field test and bring our drone to operate in a real environment, or as much as we can make our drone to be in while keeping to the budget and constraints of the law. For our project, we came to the idea of showcasing our drone's ability to dodge buildings and other environmental stimuli and showcase our accident-avoidance software and how well it works when put under pressure while in tandem with how well our drone's sensor package can pull in data and compare parameters in real-time.

Figure 58: Field Testing Site Tangerine Field



Now to showcase these two things we need for there to be an adequate location, which in this case means that it needs a wide grass or open terrain and no other major outcroppings hanging over it and to be outside of downtown and campus limits as those places require that we would need to acquire a drone pilot license for our operation, which if it can be avoided will reduce down the cost of our project and make it more viable. And we need adequate stimuli so as to fully showcase and extrapolate a good amount of data from the drone’s interaction with the course. We are then deciding to do our field test in a wide-open field, preferably a grass plot, so that means our best guess to be a park or other open space such as that, and we found through research into where other drone pilots that fly in the city go to that there is a perfect location in Apopka Florida known as Tangerine Field. It is open land with plots of grassland and even a runway, cover for when thunderstorms hint so we can be in accordance with safety guidelines when operating in the open air and is close enough to our home locations to be viable for this project needs. We see this in **Figure 58** that showcases the field as a viable option for testing.

Figure 59: PVC framework of Buildings for Field Test

Figure 60: Box that will form the Skin of City

This means we now have an adequate location for testing, now we need adequate stimuli for our drone and data gathering needs. We do this by recreating the most likely environment our drone will be encountering during operation as a product in consumer's hands, or in other words, an urban environment. Now because of budget and site restrictions, this means in no way are we able to emulate an actual cityscape. We will be recreating the tight spaces and restricted airways that come with cities instead. To do this we will make the framework of a building or in this case multiple buildings out of PVC pipes so as to give our site some sturdiness in case of high winds and other factors of the flight-testing site. We will then provide the “skin” of the buildings with cardboard to showcase the more rectangular aspect of downtown buildings, with variations in the different structures to provide a good difference in data input so that any field testing brings back multiple stimuli injection into our data network for our drones “learning” software system. We give the basics of this idea in **Figure 59** and **Figure 60** to show the PVC framework and cardboard skin of our made-up cityscape.

6.2 Software Testing

For testing the software there will be tools to be able to measure output and then also testing scripts used to simulate the runs and also collect data. With drones, there is simulation software out there that is able to practice machine learning and object detection without having to use the drone yet. One simulation software we will be utilizing is Gazebo. Gazebo has been highly recommended by PX4, which is one of the design choices and able to set up connections and simulate because PX4 uses SITL. SITL stands for Software In The Loop. This means that is able to run the software flight stack and use a simulator like Gazebo to create values and see how the data would be outputted. In order to do these types of simulations, a MAVLink API needs to be set up. **Figure 61** is an overview of how the data and telemetry come out of the simulation. The API allows sensor and message data to go through and then output telemetry based on our needs.

Figure 61: MAVLink API

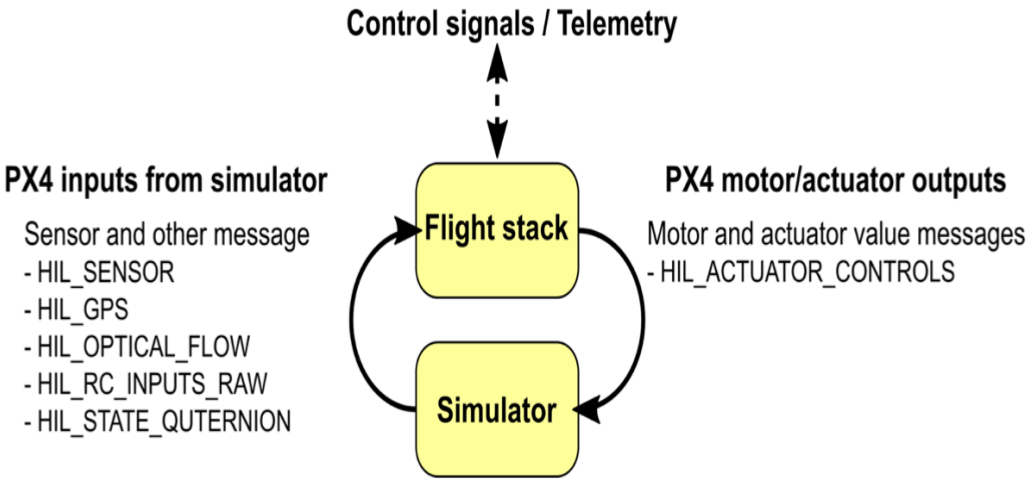
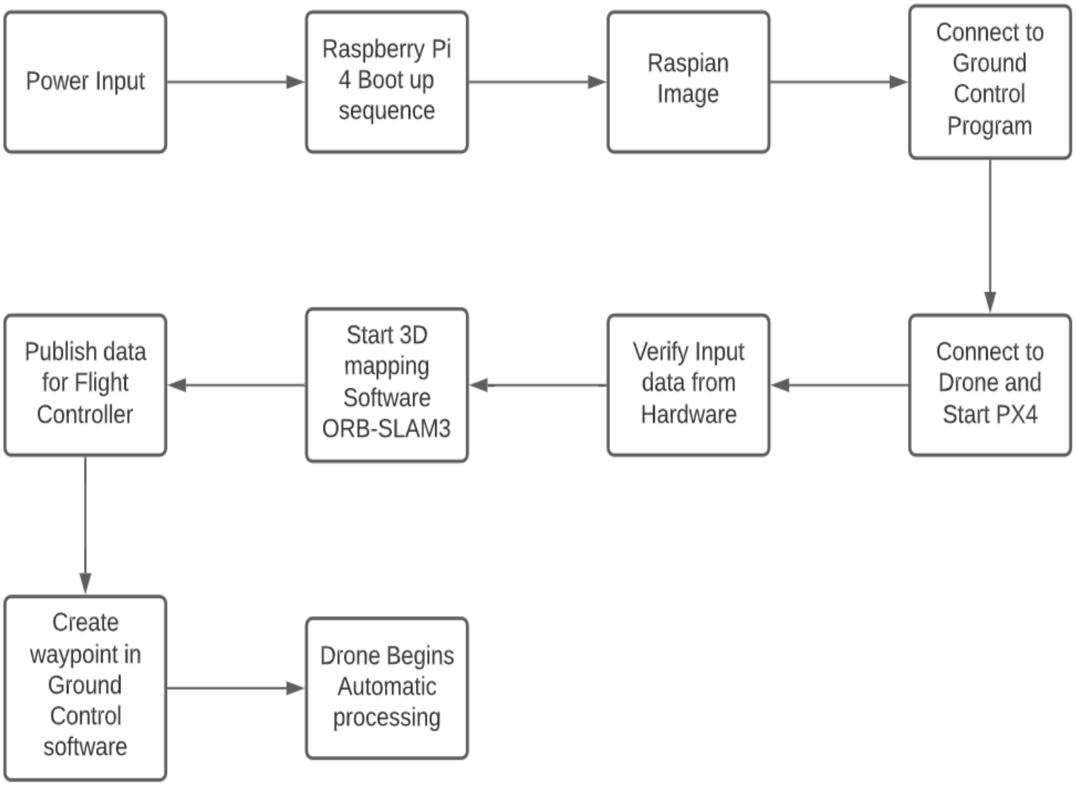


Figure 61 outlines the messages that would be used to access certain parts of the drone connections. HIL_SENSOR is used for IMU readings, HIL_GPS is used to access GPS RAW values. These can be specified and used for debugging because the HIL_STATE_QUTERNION gives telemetry of the altitude, speed, locations and so on which can be analyzed more with what needs to happen and what would happen. We will be able to plug up our flight controller board of the Navio2 setup and then test it with our software to see if it would be working properly. It will also allow us to ensure that all of our individual software are working independently, and also that they are communicating effectively. Testing is extremely important to the design process of projects because, with projects just as drones and material items, hundreds of thousands of dollars can be put in these things, and if not tested because a demo could result in damage to the environment

and the item at hand. All that time and money spent would be destroyed and have to start over again. Having software like Gazebo also allows us to be making progress on the software part of the project without having to work with all the hardware parts or have the drone flying yet.

6.2.1 Software Startup Sequence

Figure 62: Software Start Up Sequence

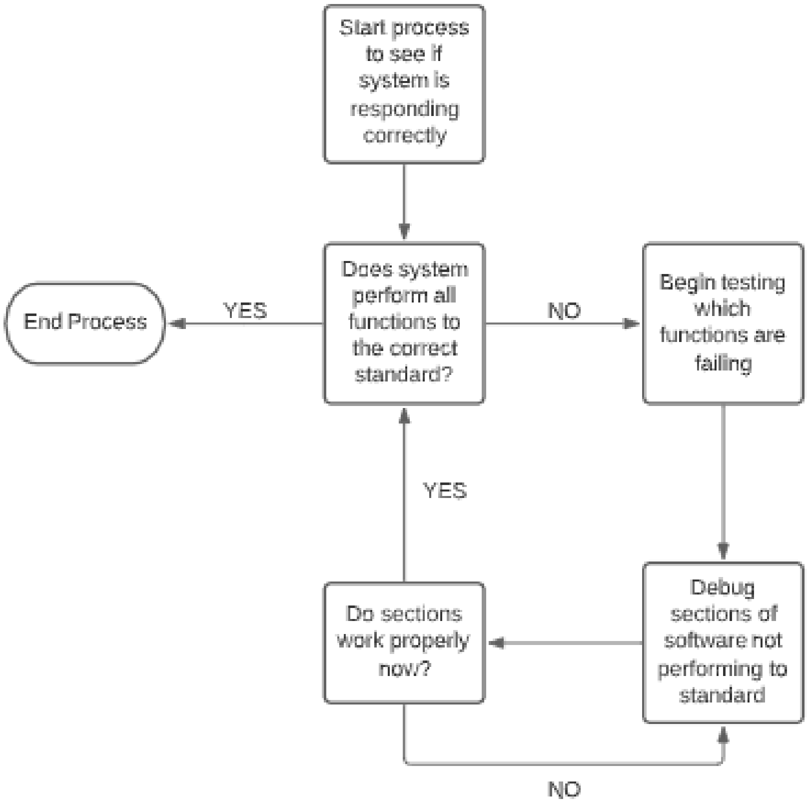


When the battery is connected and the drone receives power, the Raspberry Pi will automatically boot the installed Raspbian image. After the board is fully booted, it will connect to our ground control station laptop via a WiFi hotspot. Once connected, we are able to SSH into the drone. We will use this to start PX4 on the drone. When this software is up and running, we will perform a pre-flight check to ensure all of the hardware is working correctly and all of the telemetry data is being passed to our ground control station, we will start running the 3D mapping algorithm. This will immediately start publishing the point-cloud data for the pathfinding algorithm to use. From there we will create a waypoint mission and set the drone into launch mode. The rest of the behavior of the drone will be handled fully autonomously. We see this process sectioned out in **Figure 62**.

6.2.2 Software Failure Testing

We will need to prepare for all possible outcomes during the intense operation of our drone's standard operating procedures. We will have fail-safes and guidelines that we will follow in case of failures that can cause catastrophic events to the drone. This section will be going over the software style of failures and how we will be approaching them. We see this in **Figure 63** on how we will take such action in the future.

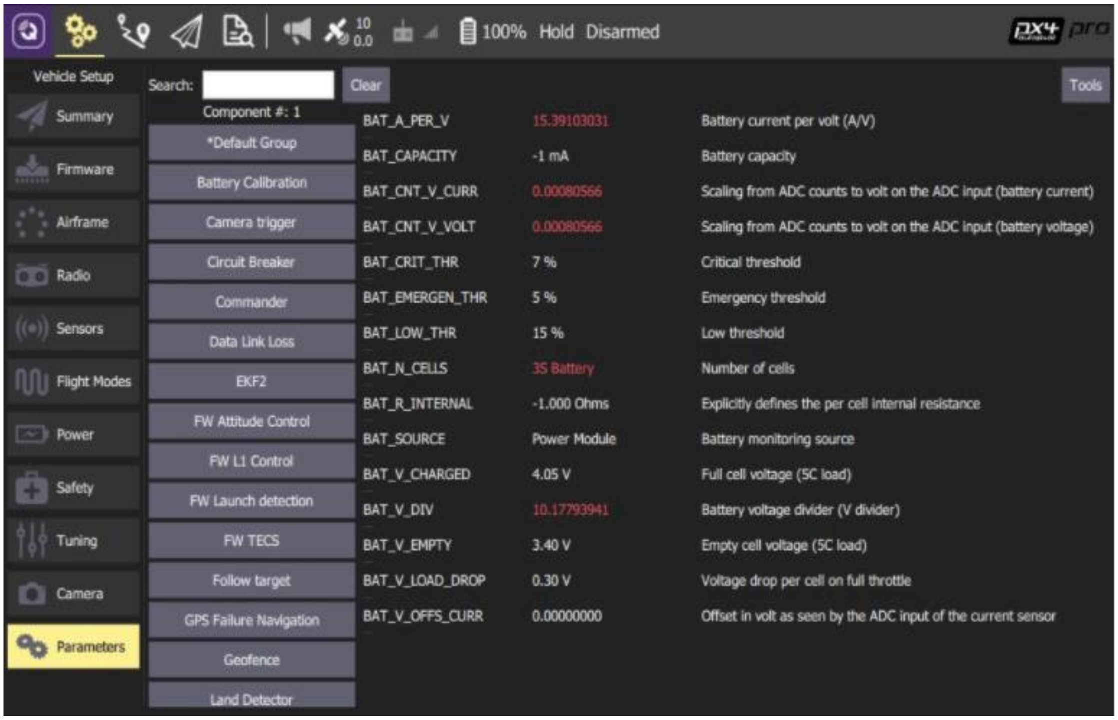
Figure 63: Software Fail Procedures



6.2.3 Parameter Testing

Our parameters are paramount to our successful implementation of an accident avoidance system. These need to be fine-tuned to have any chance of a successful dodge. We are using a ground control program known as QGroundControl that has an easy-to-understand user interface. This program lets us control the parameters of our flight controller so as to provide a tighter detection range for our sensor. When entering the GUI for QGroundControl under the Parameters tap we see we can change all kinds of systems within our flight controller's umbrella of control, as shown in **Figure 64**.

Figure 64: Parameter Screen on QGroundControl

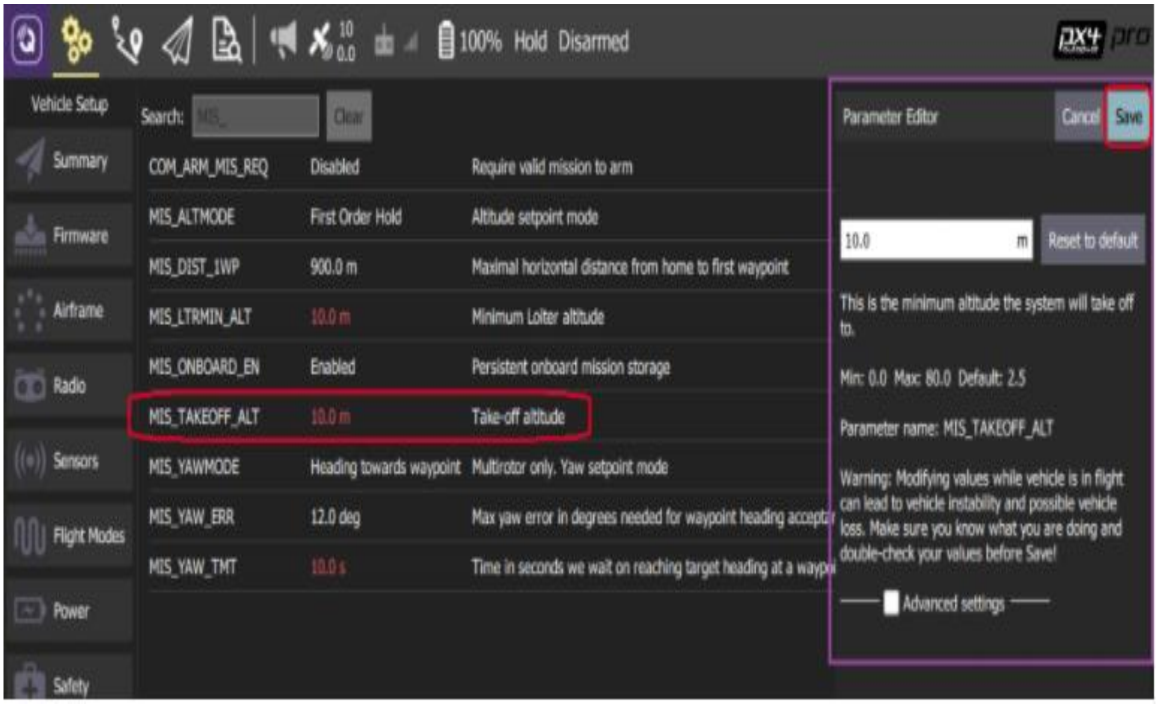


BAT_A_PER_V	15.39103031	Battery current per volt (A/V)
BAT_CAPACITY	-1 mA	Battery capacity
BAT_CNT_V_CURR	0.00080566	Scaling from ADC counts to volt on the ADC input (battery current)
BAT_CNT_V_VOLT	0.00080566	Scaling from ADC counts to volt on the ADC input (battery voltage)
BAT_CRIT_THR	7 %	Critical threshold
BAT_EMERGEN_THR	5 %	Emergency threshold
BAT_LOW_THR	15 %	Low threshold
BAT_N_CELLS	3S Battery	Number of cells
BAT_R_INTERNAL	-1.000 Ohms	Explicitly defines the per cell internal resistance
BAT_SOURCE	Power Module	Battery monitoring source
BAT_V_CHARGED	4.05 V	Full cell voltage (5C load)
BAT_V_DIV	10.17793941	Battery voltage divider (V divider)
BAT_V_EMPTY	3.40 V	Empty cell voltage (5C load)
BAT_V_LOAD_DROP	0.30 V	Voltage drop per cell on full throttle
BAT_V_OFFS_CURR	0.00000000	Offset in volt as seen by the ADC input of the current sensor

We can further search specific parameters we will need to truly implement our system using a given search bar, which will be useful in our case because of the drastic number of sensors we will be inputting into our drone's flight controller and data processing board. There could be instances of names being not perfectly cloned over to the GUI of QGroundControl so the ability to look up specific parameters will be a useful tool in our project. To continue further with actually changing parameters, this GUI has a helpful system for the user when one must be changed; namely when clicking and changing a parameter a sidebar menu will populate with the already given parameters and what they actually mean and how changing certain ones will change the outcome of the system in total, as seen in **Figure 65**. This figure demonstrates what it would look like when trying to operate the parameters and search function making it a breeze to use.

This will ease our inputs so we can slowly and steadily raise and lower our parameters where they are needed so as to provide our accident-avoidance system with the adequate data inputs it needs to perform to the best of its abilities. We will be able to implement test cases that change parameters rapidly or drastically in order to test out some of the extreme cases and see how the drone will react to them. It is important to utilize these sources instead of using real objects that in the case of a test failure would make the drone crash and become damaged.

Figure 65: GUI Parameter Help Interface



COM_ARM_MIS_REQ	Disabled	Require valid mission to arm
MIS_ALTMODE	First Order Hold	Altitude setpoint mode
MIS_DIST_1WP	900.0 m	Maximal horizontal distance from home to first waypoint
MIS_LTRMIN_ALT	10.0 m	Minimum Loiter altitude
MIS_ONBOARD_EN	Enabled	Persistent onboard mission storage
MIS_TAKEOFF_ALT	10.0 m	Take-off altitude
MIS_YAWMODE	Heading towards waypoint	Multirotor only. Yaw setpoint mode
MIS_YAW_ERR	12.0 deg	Max yaw error in degrees needed for waypoint heading accepta
MIS_YAW_TMT	10.0 s	Time in seconds we wait on reaching target heading at a waypoi

10.0 m

This is the minimum altitude the system will take off to.

Min: 0.0 Max: 80.0 Default: 2.5

Parameter name: MIS_TAKEOFF_ALT

Warning: Modifying values while vehicle is in flight can lead to vehicle instability and possible vehicle loss. Make sure you know what you are doing and double-check your values before Save!

6.2.4 Sensor Input/Output Testing

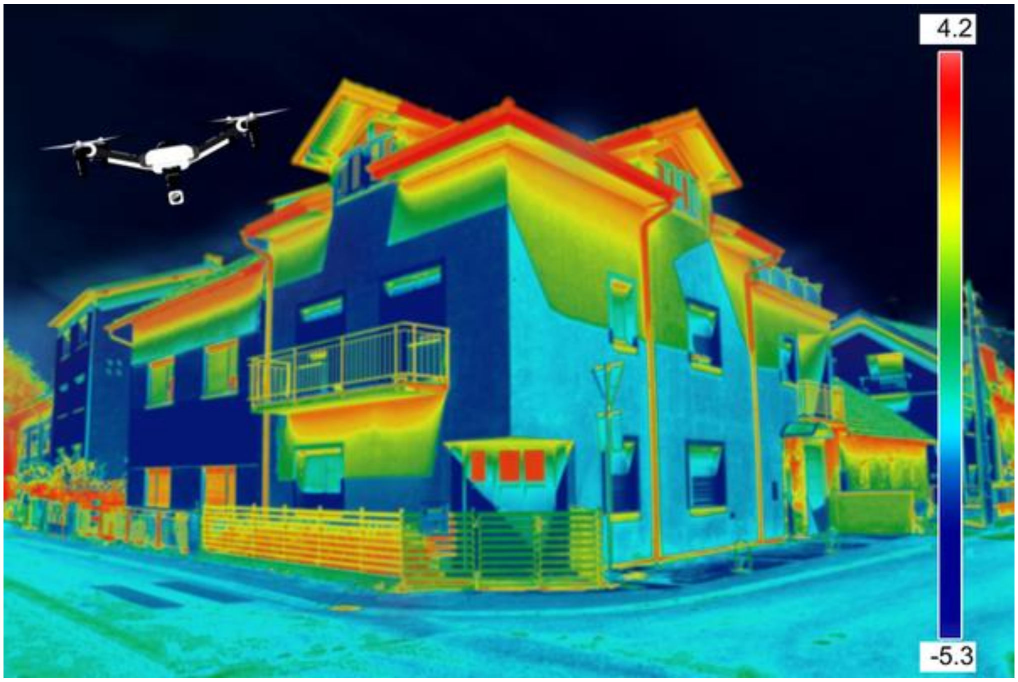
Sensor testing will require using different environments of “test cases” that they could come in contact with. Some sensors will need to be tested in different lighting conditions, while others will be speed and height changes to see what data it will be giving us.

To start off with this system we will use simulation software that can connect to the sensors and give it certain conditions to test the functionality of the sensors. Next, we will attach to the drone and do some test flying to see what data it will be giving us. For example, the accelerometer will be attached to the drone while we test going in straight lines and having a speed detector on us to compare that to the data the sensor gives. So, if we see the speed detector showing us 10 mph and the sensor showing 60 then we will need to work on calibration or see if the sensor is faulty.

The barometer will be used to help test the position of the drone in addition to the GPS. We will be testing these items to make sure they can give data back to a device that will translate it into coordinates. We then will have these on the drone and place them on different objects with different heights and test what coordinates it will give. Testing on steady objects first is the very important starting point for these sensors because we are able to hand measure it ourselves, whereas once it starts flying will be more complicated to accurately measure and will need to have confidence in the sensors that they can complete the tasks and checks.

Ultrasonic sensor testing will consist of having it on a steady object or table in a clean room and putting objects in front of it at different lengths. With everything standing still, we can measure out how far to put the objects to test the max detection rate and size of objects, if it makes it easier or harder. Then based on that feedback it will upgrade to testing it while hovering and slightly moving. The goal of testing the ultrasonic sensor is to gain confidence that while it is moving at a set speed it will be able to detect the object in front of it, otherwise, this will result in a crash of the drone.

Figure 66: Heat Signatures with Numbered Parameters



Our IR sensors are very instrumental in anything we will have to avoid that holds such a heat signature that it is hotter than the surrounding area, and to the surprise of most this includes many different things that could affect our drone, not just alive things that are moving in tandem with our drone and could potentially hit it such as a bird but even air gust and spots of rising heated air that could destabilize our drone in flight if the drone ran into such a thing unexpectedly. Now IR is not the best sensor for daytime as this is wildly known as only the best IR sensors are sensitive enough to distinguish things from the UV heat of the sun during the day, but for nighttime flight, we can be assured our drone will be very good at it. For as this can open the market for 24-hour operations and help to deliver companies everywhere with this specific problem. Now for our specific testing parameters, we would "teach" our drones flight controller system when to distinguish certain heat signatures from others and teach it what to classify as background versus what not to with parameters we will implement and change within our ground control program and fine-tune during actual testing, using numbers such as shown in **Figure 66.**

6.2.5 Camera Tracking

Camera testing will happen once when we get the camera and will wire it up to a device and test the functionality. We will be looking for it to properly show an image with the quality it has described in the description, if this is not achieved, we will be sending it back and finding an alternate camera to use that will be up to standards. We then will be testing it further by using our object detection software and set up an environment that will easily detect objects to see if it will put a box around the image to identify. When this test is done, we are testing how well the camera communicates with the software for it to accurately detect objects.

Figure 67: Image Testing for Camera Tracking learning program

For our actual testing of the project and camera specifically, we will be in a field. The field will have a route created from objects we have and a cardboard living situation to test if it can travel through a fake city or neighborhood area to deliver the package. The camera will have to detect the area it will land, trees, poles, and other common objects that we will have to represent what it might come across in the real world. We have examples of other testing environments that we will take ideas of testing from, such as shown in **Figure 67**, where they used images of people and started classifying them by parameters to steadily step by step teach their program what was a person and what was not.

6.2.6 Stress Testing Drone in Flight Operations

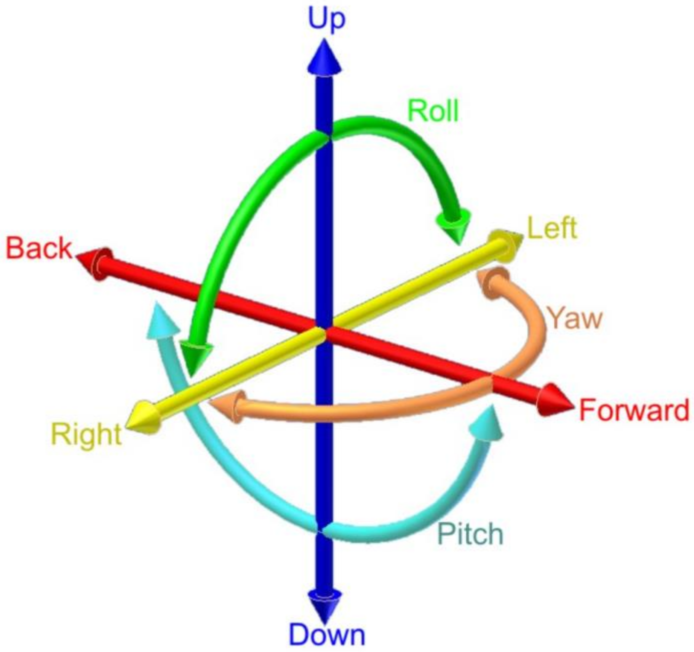
Now, during the testing and operation of our drone product, there will be times when during nominal and normal flight conditions, there will be unexpected factors that could interfere with the flight path. This does not mean gust of wind or movement of the package within its own casing and whatnot, as those will have to be compensated for within the flight controller’s normal processes and should be accordingly easy to account for. What these stress tests are for are the abnormal cases that are not taken into account by the flight controller's original programming. Such as actual objects interfering with the flight path or crashing into the drone during operation, which can cause the drone to unexpectedly have to overcorrect itself in regular flight which can lead to crashing. To combat this, we use our accident-avoidance software to give the drone an additional set of “instincts” that will take over and perform the correct procedures when such situations occur. This can be very stressful to the flight system and its ability to fly if perchance, the drone is not in a situation where it is in the way of something such as a building or tree that it can easily read and move out of the way from, the drone will need to take extreme evasive maneuvers to have a chance of dodging something that is going at any reasonable speed that it can catch up to a drone mid-flight. So of course, during these corrective course adjustments we are going to have more extreme G forces and higher variances in pitch and yaw when flying, thus leading to a much higher potential in crashing anyway if we do not find the parameters that perform admirably during such maneuvers. As we see in **Figure 68**, crashes can be devastating for the future continuation of our project, and the need to test parameters that can induce such crashes to need to highly regulated in simulation and out of it so as to give us ample data to correct any instance this devasting impact to our project could have.

Figure 68: Drone Crash During Operation

We see earlier in this paper that our ground control program, QGroundControl, has a robust and easily manageable parameter system. It also lets us easily see what our changes to the parameters will do to our drone in summaries given for each different parameter that is easily understood by the programming. Things that are from sensors that aren’t part of the flight controller programming aren’t as intuitive but can still be found from other sources, such as the input of an infra-red sensor. So, the action of dodging may be carried out by parameters that are easily described and changed to control how any actual action that our accident-avoidance program would take can be easily fine-tuned to be more easily achievable for the best results.

For the best description of how our accident-avoidance program would achieve favorable results would be to describe a pilot might actually dodge something in a regular helicopter, as that is the closest comparison in-flight standards that can be shown. In **Figure 69** we see a graph showing how each variable of flight influences direction and can rapidly affect which direction a plane or helicopter can take.

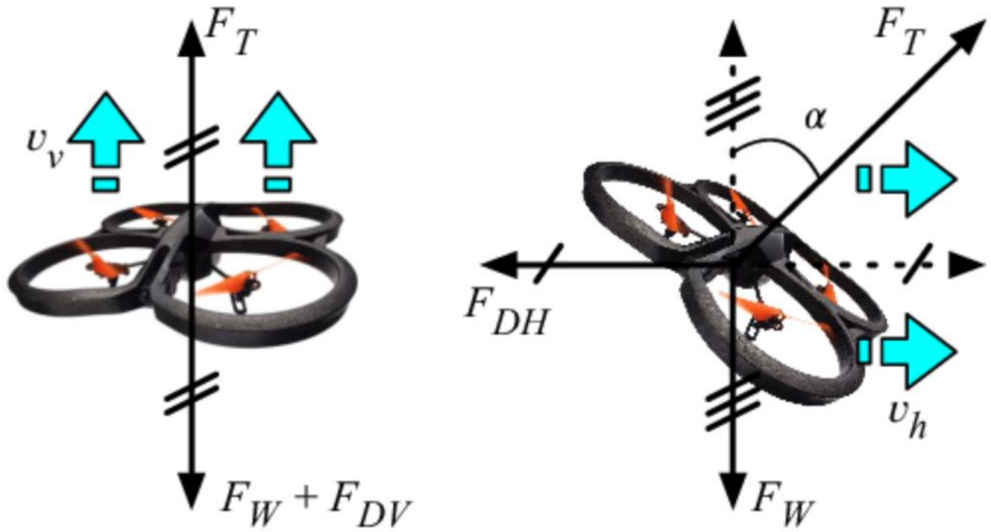
Figure 69: Pitch, Yaw, and Role



We know from other instances of crashes and successful dodges in such situations that needed such fast-paced movement that it was the pitch, yaw, and roll to be precise that affected the plane or helicopter’s chance of actually dodging such objects that could interfere with their flight path. The pitch is the action that our drone will take when moving the front of the drone up or down, thus affecting the speed with which it is traveling forward or slowing down. So, when we come into a situation where the accident-avoidance software activates and takes temporary control of the flight controller, this will be the first action that such programing will take. Slowing down by decreasing the drone's pitch will increase the time the drone has to react to the incoming object that will interfere with the flight path. Decreasing the pitch will lift the front of the drone so as to have more downward force from the propellors acting in the opposing direction of our previous forward movement. This action can cause stress upon the motors and possibly the package that the drone will be carrying at the time, so we need to test parameter adjustments for this action just as any other action. This action however is less intense than others as it does not require intense side-to-side movement that could affect the drone’s ability to even have a stable hover. The reason we are testing this action in a testing environment is because of the other aspects of the drone's continued operation. Such as the motor, sensors, propellors, and even the controllers and data processing board that can be negatively affected by G forces that they are not designed to take. If these tests are done properly the components could be stressed to their breaking points. The G forces could have effects on those items that we will not understand until testing them. We need to carefully test and perfect the act of slowing down with pitch as so to not hamper the drone's continued ability to fly to the objective.

Continuing with the two parameters that will need to stress test our drone with is yaw. This is the action showed in **Figure 69** that enables the drone to fly left or right and permits the drone to change the direction it is heading when moving forward. As we will be slowing down using pitch during any instances of the accident programming activating, the drone will need to adjust its yaw so as to slightly adjust any forward motion it is still achieving during the incident, as the pitch will drastically decrease forward movement, but it will not be able to completely erase it during operation in such a fast-paced instance. So, to make sure our drone is using the left-over forward motion to its own advantage we will increase the yaw to the left or right depending on location and other factors that the drone will have to take into account in that moment of movement such as buildings or trees or objects around it to achieve an effective “dodge”. Parameters will need to adjust in the program during stress testing to make sure that again, during evasive maneuvers our sensitive equipment will not be so negatively affected as to not be able to continue with the designated mission and become less than effectively operable, as we will still need our sensors for future missions. These testing measures are imperative for making sure everything is ready to go because they will prevent causing damage to unnecessary equipment. Keeping the equipment as safe as possible adds to the longevity of the drone and lifecycle of equipment, which aligns with our standards and realistic design constraints of the project.

Figure 70: G Forces Acting Upon Drone



And finally, after we have tested both our pitch and yaw, so as to be perfectly adjusted for evasive maneuvers such as slowing down with pitch, and turning left or right with yaw, we need to adjust and test our roll parameter, which we see in **Figure 69** will change the angle with which the drone is approaching its forward motion. The roll function can almost be considered our most important parameter to adjust and test as this action is the thing that can decrease our system G force and actual stress on our system during the accident-avoidance action. By decreasing or increasing our roll we achieve an angle that will have the G forces acting upon our drone during such instances to be less or more depending on the other two parameters at the time. We can see this in **Figure 70** as the G forces working a drone during flight and when the angle of the drone changes the direct direction of the acting G forces change as well. Keeping these factors in mind when creating the project is a part of the attention to detail needed for the accuracy that will lead to the drone performing how we want it to. It can be easy to misinterpret the variables or accidentally assign the wrong one to them and then the controls will be completely backwards.

When we have adjusted and tested the exact stress factors that our drone and all its components can handle in a series of stress tests in simulation and in real environments, can we assuredly perform accident avoidances in the correct and effective outcome our drone will need in such instances where time is of the essence? We see the correct and hopefully, future success of our drone in **Figure 71** where a drone has successfully adjusted all its pitch, yaw, and roll parameters to perfectly dodge out of the way of incoming objects.

Figure 71: Drone Correctly Dodging Incoming Object

7. Guidelines and Procedures

There is nothing more important to our project into ourselves than our own personal safety. We model our safety guidelines set by OSHA and common-sense policies and PPE experienced through members of our group who have a military background and Industry experience. In following our safety protocol, we will ensure no one in our group will be injured during the testing of our drone.

7.1 Safety and Procedures while Testing

Our group intends I am having a very strict start-up sequence before testing our drone. We believe it is important to have a strict start-up sequence before testing the drone for safety reasons. Drones can be quite dangerous.

Some of the dangers or risks our group undertakes when testing a drone are as follows. Drones' propellers have very high rotations per minute and can easily cut or slice. One of the worst possible outcomes for a drone, the propeller would be to take a finger of which is not unheard of in the drone hobbyist community. Other concerns when testing our drone would be the drone accidentally running into us or falling out of the sky onto us. Our group intends on taking measures to minimize all these risks when testing its delivery drone project.

To start off the list of ways in which we plan to mitigate these risks is to follow suit in the ways that gun ranges minimize risks when changing targets. First, we will have open communication on when we are about to start the drone and that no one is in front of the person controlling the drone. While standing a safe distance away from the drone. As well as making sure everyone is aware when the drone is being placed or being handled to not start the drone and if the drone is not to be started that the battery to disconnect to further ensure safety when handling.

During the testing of the drone hardhats and safety, goggles will be worn by all members of our group. The reason safety goggles and hard hats will be worn is due to the unlikely event there if anyone is hit or ran into by a drone that the damages would be medicated by our safety gear.

After the test when the drone has landed, and we have no further intentions of flying the drone. One member of our group will approach the drone with leather gloves on before disconnecting the battery. Once the battery is disconnected from the drone, we then deem the drone to be safe to be handled without gloves on. Once the drone is deemed safe to be handled without gloves on, we are then able to make any adjustments to the drone before our next test.

Table 27: Safety While testing

Step	Task	PPE
1	Communicate that the drone will be tested	Face Mask
2	Connect Drone to battery supply	Gloves
3	Step a safe distance from the drone	Safety Goggles/ helmet

4	approaching drone to pickup/repair	Gloves/ Safety Goggles
5	Unplug the drone from the battery	Gloves

So, the sequence in which our group will start up before testing the drone is as follows. First, ensure open communication that the drone is being handled and placed. While wearing gloves whenever the drone is connected to the battery power supply. Then before starting the drone that no one is in front of the person controlling the drone. While wearing hard hats and safety goggles. Lastly when approaching the drone after it has landed in preparations to pick the drone up out-group will wear gloves before disconnecting it from the battery power supply. While making sure everyone in our group understands that the drone is now safe to be handled before testing again and is disconnected from the battery. As seen in **Table 27** above.

7.2 Tool Use and Procedure

Our drone delivery project will require us to use very specific tools to accomplish our goal of having a drone be capable as well as delivering a package while detecting objects. The tools in which used to assemble our drone delivery project or just as important as the parts we use. Without the proper tools we will never be able to assemble, test, and modify our drone to the correct specifications. The tools that we will be used to assemble our drone and to test various parts are as followed.

First, for the assembly portion of our drone, a screwdriver will be critical in attaching all of the arms and legs to the body of the frame as well as securing the Motors and the propellers in the correct orientation of our drone. Secondly, we will use wire cutters to attach the correct male or female end of connection leads accordingly to match the assembly of our drone. Third, we will use a soldering iron to Sauter any parts that we may need to add to our PCB design or other boards as they need it. Lastly, we will be using a multi-meter to check tire distribution throughout the board to make sure that we are getting the correct voltages throughout.

In terms of software, there’s also a tool that we will have to use. The tool that we will be using for software will be a laptop. The laptop will not only run the software we need to code the proper procedures. We will also be using the laptop as a controller for the drone so that we will be manually able to maneuver the drone from the laptop.

So, list the tools that we will be using in this project to build a package delivery drone. The tools are a screwdriver, wire cutters, soldering iron, Multimeter, and a laptop. As you can see in **Table 28** shown below, the tools that we use are a screwdriver, wire cutters, soldering iron, multimeter, and a laptop and what those tools are and what their functions will be for our drone delivery project.

Table 28: Tools

Tools and Functions		
#	Tool	Functions
1	Screwdriver	Assembly of drone frame
2	Wire Cutters	Spicing for battery leads
3	Soldering Iron	Hardware repair
4	Multi Meter	Testing Voltages
5	Laptop	Drone controller and Software testing

7.3 Parts Management

Parts management is a critical role for any project that wishes to meet deadlines. Without proper park management, your ability to obtain the parts you need and to make sure they are available when you need them is a delicate process. Our group has put in a lot of research to make sure the parts we use for this project will be ready and ample for us to use at our leisure.

The procedure in which our group has called about part management is to minimize any bit of struggle to obtain a part. First, we looked online to see how many available products we're listed as well as shipping time for those parts. The parts that had limited supply and longer shipping times were prioritized first. To prioritize them first we ordered them with ample time and ordered a few at once just in case anything got damaged in shipping or would become damaged through our testing. Second, we purchased the parts that were not as scarce and had quicker ship times. For all parts, we messaged manufacturers to make sure they had no intentions of ending the lifecycle of any of their parts.

In terms of shipping, our group understands the climate in which the world sits with how Covid 19 is hurting the estimated time of arrival. As well as some international waterways being blocked can cause shipping delays. Our way of handling these and foreseeable challenges is to order the parts early and keep a count of how long they actually take to be shipped. And if the parts take longer than expected to show up, we will plan accordingly to order more of those parts so that they show up before they are possibly needed for a project.

In terms of storage and where to ship the parts we order. Our group has decided that it should be sent to one of our places of living to whoever is in the most centralized location between the four of us. As well as who is closest to campus in most convenient to reach. Also, to whoever has the more available space to work on the project if need be. This keeps the items organized and in one spot for us to work from and easier when contacting companies for replacements or such.

So, the ways in which we ensure proper parts management is clear as you can see explained in **Table 29** below. We prioritize the parts that are less available and have a longer shipping date. Second, we keep track of how long it actually takes the parts to arrive at us and come to the conclusion if we need to order parts again now before it's too late. Lastly, we store the parts in the most centralized location, so it is convenient for our project. With our parts managed in this orderly fashion, we believe we should run into no hiccups that would suck back our deadlines for a drone delivery project.

Table 29: Parts Management Steps

Parts Management	
step	**Goal**
1	Order the parts we believe might go out of stock soon and have the longest shipping dates
2	Note how long the parts actually took to ship in order accordingly if the part might break under testing
3	Keep parts in centralized locations

7.4 Finances

For any project to be successful budgeting and financing are critical. The way our group has decided to budget and finance our project is it carefully thought-out system. We are a self-funded project so in essence the money put into this project is our own and we do not have sponsorship.

To make sure not one of our members was left holding the foot of the bell at the end of our project. We decided to implement a ledger for the person who bought the part tells us the cost after shipping and then divides that cost evenly amongst all members of our group. Screenshots of the receipt the ship dates and the cost are all required to make sure we have an itemized list of where the cost is coming from. We also created a Google Docs form to fill out to help create an itemized budget at the end of our project. We believe having everyone pay 1/4 the cost of every product we buy is the most efficient way of keeping a balanced budget without having any overlapping money exchanges.

We believe with following this procedure of financing that no one in our group will be left footing the bill. In the end, we will have a very organized and detailed Ledger of how and where the cost of our project came from. As seen in **Table 30** below the steps we take her as followed. First, purchase the part. Second screenshot of the receipt and shipping date. Third, fill out the Google forms with the receipt of the total cost and shipping dates. Lastly, receive 1/4 of the cost of your part from all project members.

Table 30: Finance Process

Finance Procedure	
Steps	**Action**
1	Purchase part
2	Screenshot part ordered and shipping date

3	Fill out Google forms
4	Receive payment

8. Administrative Content

8.1 Project Milestones

For our project, we created our milestones by starting with the hard deadlines given by the course and then dividing that up into smaller personal group deadlines that will keep us on track. **Table 31** contains our deadlines for the first half of the project that concentrates on research and structure, then **Table 32** contains how we are splitting up the planning, decision making, and research that will be going into creating the main paper and helping with ideas. **Table 33** contains a vaguer outline for how implementing and creating will go in the second half. Our group consists of two computer engineers that concentrated on a lot of the software design choices and then two electrical students that concentrate on the power and hardware choices. We then took this knowledge and have meetings to get an understanding of what everyone was working on for the case someone needs to take over. We do not want to be too reliant on one person knowing something so specific.

Table 28 shows how we broke up accomplishing the paper deadlines and then in the behind the scenes we would conduct weekly meetings that had checkpoints for people to present their findings. We would also end meetings with objectives for the next time, which came in handy for the decision-making process because we would have a decision needing to be made and then take the week to all research and that following meeting present our findings and decide as a group which we would go with for the project.

Table 31: Main Timeline

#	Task	Start	End	Status	Responsible
1	Ideas	1/11/21	1/29/21	Completed	Group B16
2	Project Selection and role Assignment	1/25/21	1/29/21	Completed	Group B16
3	Divide & Conquer 1	1/25/21	1/29/21	Completed	Group B16

4	Divide & Conquer 2	2/1/21	Feb 12	In Progress	Group B16
5	Table of Contents	2/1/21	4/27/21	In Progress	Group B16
6	20 Page Submission	2/1/21	2/16/21	In Progress	Group B16
7	40 Page Submission	2/16/21	3/9/21	In Progress	Group B16
8	60 Page submission	3/9/21	4/1/21	In Progress	Group B16
9	80 Page Submission	3/9/21	4/5/21	In Progress	Group B16
10	100 Page Submission	4/5/21	4/15/21	In Progress	Group B16
11	First Draft	2/1/21	4/20/21	In Progress	Group B16
12	Final Document	2/1/21	4/27/21	In Progress	Group B16

Table 32 : Research, Documentation, and Design

Research, Documentation, & Design					
14	Frame design	2/1/21	4/2/21	Researching	Raymond & Austin
15	Sensors	2/1/21	4/2/21	Researching	Raymond & Austin
16	Motors and blades	2/1/21	4/2/21	Researching	Raymond & Austin
17	Controller	2/1/21	4/2/21	Researching	Raymond & Austin

18	PCB Layout	2/1/21	4/2/21	Researching	Raymond & Austin
19	Power Supply	2/1/21	4/2/21	Researching	Raymond & Austin
20	Computer Vision options	2/1/21	4/2/21	Researching	Ellie & Dan
21	Algorithm code outlines	2/1/21	4/2/21	Researching	Ellie & Dan
22	Languages to code in	2/1/21	4/2/21	Researching	Ellie & Dan
23	platforms for the code	2/1/21	4/2/21	Researching	Ellie & Dan
24	Set up Github repository	2/1/21	4/2/21	Researching	Ellie & Dan

Table 33: Senior Design 2

Senior Design 2					
25	Build Prototype	TBA	TBA	TBA	Group 16
26	Testing & Redesign	TBA	TBA	TBA	Group 16
27	Finalize Prototype	TBA	TBA	TBA	Group 16
28	Peer Presentation	TBA	TBA	TBA	Group 16

29	Final Report	TBA	TBA	TBA	Group 16
30	Final Presentation	TBA	TBA	TBA	Group 16

8.2 Project Budget

For our budget we are preparing for the worst so that if something goes wrong, we will have wiggle room to buy more, we understand things will break or be defective, so we made sure to plan accordingly. As seen in **Table 34** we have allowed ourselves to have multiple drones' propellers and batteries in case things go south or break. In **Table 35** we allow ourselves to have multiple IR Sensors due to our research showing they don't always work effectively, and it is cheaper to buy in bulk. Additionally, we will use numerous cameras and altitude sensors to make sure we can fly safely and effectively. For **Table 36** we are planning to have a radio controller to help with returning the drone and with troubleshooting to make sure our drone runs as we want it to. The controller having wifi will have the communication standards that we need to properly tell the hardware and software what to do. Overall, we are staying optimistic about what we need but prepared to need to give a little more as with drones, crashes are to be expected and results in breaks that mean buying new items.

Table 34: Physical Components

Physical Components	
Drone Final x 1	$150 - 200
Testing Drones x 2	$100
Propellers x 20	$20
Housing Compartment x 1	$50
Batteries (Drone and Sensors) x 3	$200

Table 35: Sensing Components

Sensor Components	
Altitude Sensor x 2	$30

IR Sensor x 9	$20
Camera x 3	$200

Table 36: Electrical Components

Electrical Components	
RF Transmitter x 2	$14
Controller	$130
Charger	$50

9. Conclusion

The D.E.A.R. Drone will be able to show how object avoidance can be accomplished for autonomous drones used for package delivery. By optimizing the use of power and sensors we will create a drone that can properly deliver a package while also avoiding objects. LiPo batteries are used as the power source to keep it optimized for efficiency of delivery. Ultrasonic sensor and a camera are used for the detection of objects and being able to see and find the correct path. There are also sensors to detect the location of the drone and how fast it will be going to provide accurate travel time. This project will incorporate machine learning to optimize how the drone will travel and deliver while avoiding objects. On the electrical side, it will work on optimizing the batteries and controls with sensors. Embedded systems will be used to help communicate the system holistically and perform the mission. We will also be trying to maximize on a smaller budget than typical companies, which will hopefully lead to showing this advanced technology is affordable and reliable because whenever there are new breaks in technology it can be intimidating to the population and community.

We were able to create this idea by combining the skill set of everyone on the team as it is made up of 2 electrical engineer students and 2 computer engineering students. Our backgrounds all came together to help guide us in a great direction with some people having backgrounds with machine learning, machine learning, electrical analog systems, or some past drone manufacturing experience. All of these subjects contributed to the decisions being made and trying to make the best choices for a quality delivery drone. We had also quickly realized that there are many many parts that go into creating a drone and while we could try and make one person decide the motor, another person decides the frame, and others decide the flight controllers, it was seen quickly that they are all connected and sometimes one decision could not be made without another. So, this help organized our plan and decision-making and created a domino effect. We started with picking out the frame, which led to the motor decision, which led to the ESC decision, and others because of how they all interact as a system.

The drone will be incorporating a self-guided flying system that is able to go from point A to point B without someone using a controller but also uses a camera and computer vision in order to avoid objects and make new paths depending on the circumstance. From the software side, the project includes using C++ for the embedded systems and hardware portions and then using Python for the main software coding sections that also have computer vision. The computer vision implemented Python because it was clear how easy and widely used it is in the machine learning and computer vision community. There are libraries of TensorFlow and more that will be used to accomplish the goals for the project.

When deciding the parts, sitting down and discussing what is the most important aspect of the drone is important because if not having a priority, then the project will become expensive and some decisions need sacrifices in order to accomplish the right thing. We started with our frame because that would decide the size of things going further and would decide how much battery and motor power we would need.

Automated drone delivery is slowly becoming normal and trying to enter the world to be more common. Through research, we found that many delivery companies like Amazon and such are already starting to test drones in the US and other countries to test how accurate and applicable they are to the population. We are hoping to discover what goes into these machines and the improvements that could be made to them in order to perform more properly for the area the drones are traveling in. After going through this past year of Covid and living through a pandemic it was seen everywhere places shutting down to try and protect people. Delivery drivers and people apart of the shipping process were put in harm's way constantly. Having improvements with delivery drones could have helped immensely with taking the people out of harm's way even if it was made accessible for them to use drones in the process.

Our budget going into this was hoping to spend around $1000 with about $200 wiggle room to go above if needed for accidentals. We were able to research the measurements and types of parts we needed then once we knew what we were looking for specifically, used our resources of Google, Amazon, and eBay to get the best prices for it. Sometimes this led to great savings because, for our frame, we were able to find a frame kit with the exact dimensions and accessories needed to put it together. This saved a tremendous amount of money because buying it all together saved from buying things individually. The next biggest expense seen was with the flight controller and Raspberry Pi. The computer brains of the drone can get extremely pricey as the technology that goes into these tiny minicomputers is expensive to create, but luckily this should be a one-time purchase and not be a burden on us later. It is also not necessary to immediately start working with this because there is simulation software that makes it easy to initially test code and what will be happening to drone before uploading stuff onto the raspberry pi and flight controller.

Appendix.

Bibliography

[1] "Wing," [Online]. Available: https://x.company/projects/wing/.

[2] N. Heath, "Tech Republic," 4 April 2018. [Online]. Available: https://www.techrepublic.com/article/project-wing-a-cheat-sheet/. [Accessed 2 February 2021].

[3] "Drone Nodes," [Online]. Available: https://dronenodes.com/drone-esc-electronic-speed-controller/. [Accessed 1 February 2021].

[4] "eBay," [Online]. Available: https://www.ebay.com/itm/Qwinout-F450-4-Axis-Airframe-450mm-Drone-Frame-Kit-w-30A-ESC-A2212-1000KV-Motor/284005585210?hash=item42200d513a:g:4-IAAOSwENVfWIUH. [Accessed 1 February 2021].

[5] "PLCGURUS.NET," [Online]. Available: https://www.plcgurus.net/drone-programming/. [Accessed 10 February 2021].

[6] Y. Nader, "hackr.io," 08 January 2021. [Online]. Available: https://hackr.io/blog/python-vs-java. [Accessed 10 February 2021].

[7] Shawn, "SEED STUDIO," 2020. [Online]. Available: https://www.seeedstudio.com/blog/2019/12/23/distance-sensors-types-and-selection-guide/. [Accessed 5 February 2021].

[8] "electronicsnotes," [Online]. Available: https://www.electronics-notes.com/articles/connectivity/wifi-ieee-802-11/standards.php. [Accessed 20 February 2021].

[9] J. Reagan, "dronegenuity," [Online]. Available: https://www.dronegenuity.com/lipo-drone-batteries-users-guide/.

[10] O. Liang, "Oscar Liang," Feburary 2017. [Online]. Available: https://oscarliang.com/lipo-battery-guide/. [Accessed 20 02 2021].

[11] Alex, "Drone Trest," 2 September 2015. [Online]. Available: https://www.dronetrest.com/t/power-distribution-boards-how-to-choose-the-right-one/1259. [Accessed 15 March 2021].

[12] NA, "IEC," March 2021. [Online]. Available: https://www.iec.ch/government-regulators/electric-motors.

[13] QWinOut, "Ebay," Ebay, 14 April 2021. [Online]. Available: https://www.ebay.com/itm/Qwinout-F450-4-Axis-Airframe-450mm-Drone-Frame-Kit-w-30A-ESC-A2212-1000KV-Motor/284005585210?hash=item42200d513a:g:4-IAAOSwENVfWIUH#shpCntId. [Accessed 14 April 2021].

[14] NA, "IP and NEMA standard," March 2021. [Online]. Available: https://www.delvallebox.com/news/en/ip-and-nema-standards.

[15] NA, March 2021. [Online]. Available: https://www.delvallebox.com/news/en/ip-and-nema-standards.

[16] NA, "BatteriesInAFlash.com," NA, Janurary 2015. [Online]. Available: http://www.batteriesinaflash.com/c-rating-calculator. [Accessed 14 April 2021].

[17] A. Jacobs, "QuadQuestions.com," Janurary 2015. [Online]. Available: https://quadquestions.com/blog/2014/12/27/qav250-c-rating-explained/. [Accessed 15 March 2021].

[18] Turnigy, "HobbyKing.com," Hobby King, January 2020. [Online]. Available: https://hobbyking.com/en_us/turnigy-5000mah-3s-20c-lipo-pack-xt-90.html?queryID=&objectID=69450&indexName=hbk_live_magento_en_us_products_hbk_price_stock_2_group_0_asc. [Accessed 1 April 2021].

[19] FAA, "faa.gov," Federal Aviation Admin, 1 January 2021. [Online]. Available: https://www.faa.gov/uas/recreational_fliers/. [Accessed 10 April 2021].

[20] NA, "DroneQuadCopterX.blogspot.com," NA, 2 January 2016. [Online]. Available: https://dronequadcopterx.blogspot.com/2019/01/quadcopter-power-distribution-board.html. [Accessed 10 April 2021].

[21] "U.S Department of Transportation," March 2021. [Online]. Available: https://www.transportation.gov/tags/drones.

[22] NA, "mttr.net," Matternet, 17 November 2020. [Online]. Available: https://mttr.net/. [Accessed 10 April 2021].

[23] NA, "uavcoach.com," NA, January 2015. [Online]. Available: https://uavcoach.com/where-to-fly-drone/orlando/. [Accessed 10 April 2021].

[24] March 2021. [Online]. Available: https://www.tesla.com/autopilot.

[25] M. Russia, "instructables.com," NA, January 2017. [Online]. Available: https://www.instructables.com/PVC-Drone-Obstacle-Course/. [Accessed 14 April 2021].

[26] R. Harris, "bls.gov," BLS, December 2016. [Online]. Available: https://www.bls.gov/opub/mlr/2016/article/pdf/suicide-in-the-workplace.pdf. [Accessed 14 April 2021].

[27] NA, "docs.qgroundcontrol.com," 1 January 2020. [Online]. Available: https://docs.qgroundcontrol.com/master/en/SetupView/Parameters.html.

[28] I. School, "iacschool," 13 Febuary 2017. [Online]. Available: https://icaschool.com/2017/02/13/drone-thermal-imaging-next-wave-uav-inspection-services/. [Accessed 15 April 2021].

[29] NA, "easyeda.com," EasyEDA, 15 April 2021. [Online]. Available: easyeda.com. [Accessed 15 April 2021].

[30] I. Y. S. L. J. P. Jaehoon Jung, "Object Detection and Tracking-Based Camera Calibration for Normalized Human Height Estimation," *NA,* vol. 1, no. 1, p. 10, 2016.

[31] NA, "embention.com," Embention, 21 Febuary 2020. [Online]. Available: https://www.embention.com/news/sensor-calibration-in-drones/. [Accessed 21 April 2021].

[32] M. Irving, "New Atlas," 19 March 2020. [Online]. Available: https://newatlas.com/drones/drone-dodgeball-obstacle-detection-system/. [Accessed 25 April 2021].

[33] NA, "Emissary Drones," NA, 25 Sep 2017. [Online]. Available: https://emissarydrones.com/what-is-roll-pitch-and-yaw#:~:text=PITCH%20%E2%80%93%20This%20moves%20the%20Quadcopter,which%20way%20it%20is%20tilted.&text=YAW%20%E2%80%93%20This%20moves

%20the%20Quadcopter,stays%20level%20to%20the%20ground.. [Accessed 25 April 2021].

[34] Y. B. D. B. A. M. E. P. M. Chen, "A Case for a Battery-Aware Model of Drone Energy Consumption," *research gate,* vol. 1, no. 1, pp. 1-8, 2018.