University of Central Florida

College of Engineering and Computer Science

MediDrone

Final Design

Senior Design II

Group 28 Members:

Joey Hodson - *EE*
Seth Horowitz - *EE*
Ryan Grant - *EE*
Seba Villalobos - *CpE*
Varadha Anandakumar - *ME*
Lior Arin Barak - *AE*

Project Coordinators:

Dr. Samuel Richie
Dr. Lei Wei

Table of Contents

1. Executive Summary

The MediDrone is a project conceived toward autonomous transportation of medical supplies. The MediDrone team will design, test, and build a safe, autonomous, and functional medical utility drone. MediDrone consists of six engineering students of interdisciplinary studies including electrical, computer, aerospace, and mechanical engineering students. Each student is individually contributing along with collaborating with the MediDrone team to allow delivery of this project by April, 2023.

This report includes goals, methods, constraints, also includes but does not limit all possible outcomes for each method described. Goals of this project are described in section **2.2)**, however the Specification embodies all goals **3)**. Methods **4-7)** include Mechanical Design Details, Simulation, Analysis, Cooling Mechanisms, Payload Mounting, Payload Testing, Hardware Design Details, Power Distribution, and Financial methodology.

There is no one correct way to improve upon current state of the art drone technology. Through examination of patents, innovation, and expansion of utility; MediDrone will overcome technical obstacles in order to complete a drone of medical utility. Autonomous drones, Unmanned Aerial Vehicles, in the x-copter (such as quad-copter, hex-copter) form have been proven to be a developing technology which has already provided returns in the industries including but not limited to power utilities, water utilities, scientific research, warehouse distribution, and surveying; typically utilizing a camera or sensor as a payload. The MediDrone will be equipped to transport objects as a payload, such as insulin, epi-pens, and vaccinations.

The MediDrone team will test various different software toolkits, hardware packages, and communications protocols in order to ensure the highest level of flight success for the MediDrone in all technical aspects. A member of the MediDrone team will seek a drone license from the Federal Aviation Administration as well as possible utilization of amateur radio bands for communication by an FCC Extra Licensed MediDrone team member. PCB design will be a very large part in developing the hardware of the MediDrone. Constraints do exist such as price, supply chain availability, and location of goods; however MediDrone is confident in delivering the scope outlined throughout this document on time, successful, and functional.

2. Project Description

2.1) Project Motivation

In the past decade drone delivery systems have taken a significant role in reducing delivery time and costs. Companies such as Amazon are working on developing drones to deliver store items to customers. Restaurants are exploring the option of delivery drones for making food deliveries. The purpose of this project is to extend the use of autonomous delivery drones for making deliveries of medical items such as vaccines and organs.

The intended use of the medical drone delivery system is to accelerate deliveries between hospitals and to places where land vehicles are slow to reach. Hospitals may sometimes be limited in supply of a resource, such as a vaccine or potentially something like an organ in urgent situations. In the case of an emergency, the time it takes to deliver, say an organ, from another hospital or to deliver to the patient elsewhere may be too long. Drones offer fast delivery due to the lack of obstacles in their path such as cars or traffic, low-rise buildings, or even natural environmental obstacles. This medical drone delivery system will tackle a unique challenge of preserving medical items during the drone delivery process. Medical items like vaccines and organs often need to be stored at very low temperatures to keep preservation. As a result we will incorporate a cooling system while solving the challenges of designing a lightweight, fast, low-cost, and accurate drone delivery system. The greatest challenge of the development of the drone is the autonomy factor in all of this. Autonomy makes the deliveries even more efficient by freeing up one more individual while also allowing greater scalability for the drone. In other words, since an autonomous drone can travel greater distances without the use of an operator and a binding controller, a larger medical drone may be able to travel at even greater distances in the future.

2.1.1) Basic Operational Diagram

Below, **Figure 1**, is a rough concept of what the mission path will look like. The first if-block's condition is deterministic on whether a cooling system will be developed and integrated in the drone's delivery system. If the subsystem is available, the deliverable will additionally be denoted as cooled or uncooled. The system will need to be aware of this at some point around takeoff. If it is cooled, the carrying box will need to be at the appropriate temperature prior to loading the medical object.

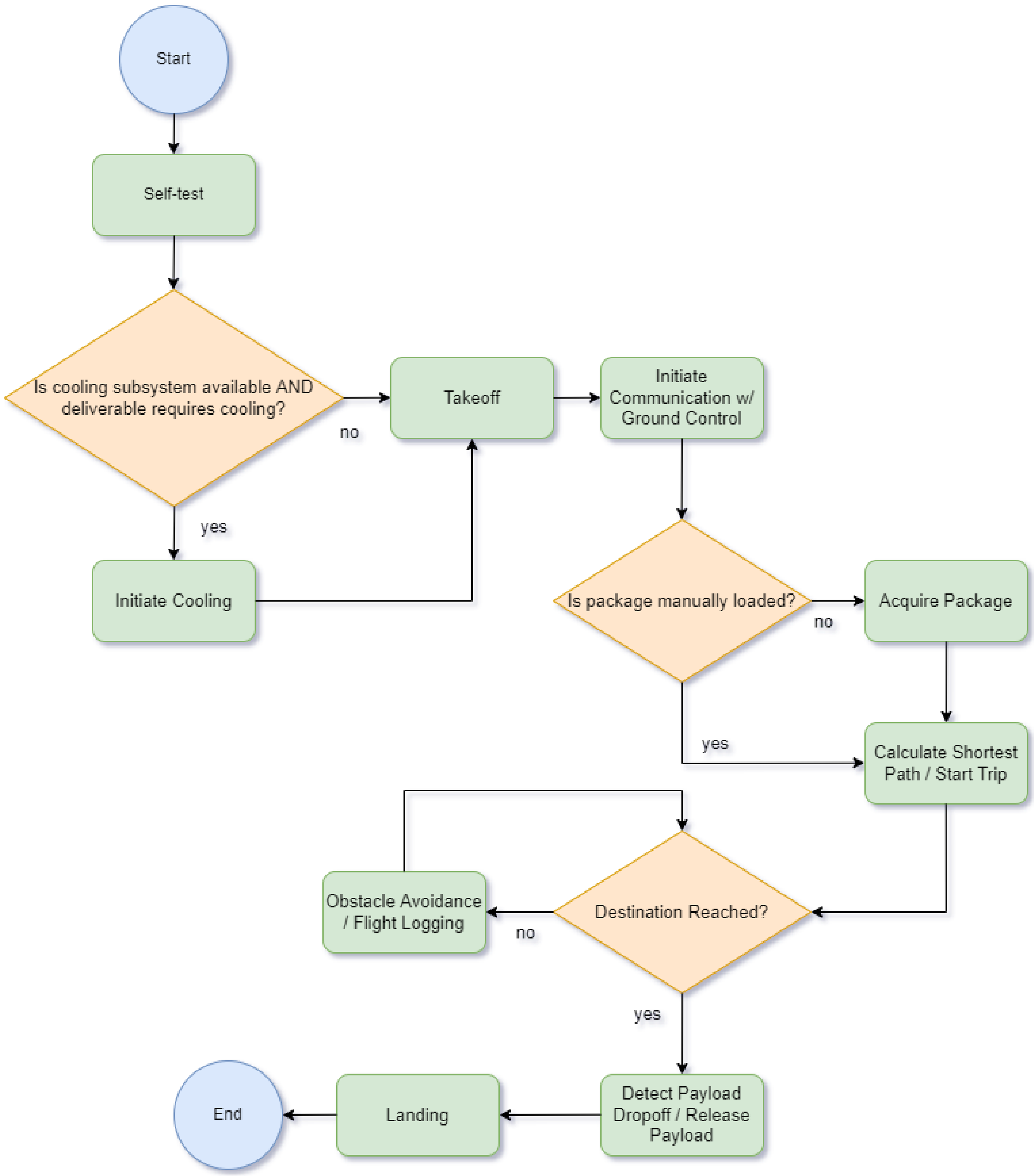


Figure 1

Final Update: Revise Block

2.2) Goals

The main goal of this project is to provide a cheaper and more flexible medical supply delivery system compared to those currently in use. Some systems currently in use include the following. Ground transportation, which is the most dominant form of transportation, is prone to delays due to traffic and does not follow the shortest path. Pneumatic Tube delivery, which allows quick transportation between buildings across a city, addresses the problem of timely delivery, but is expensive to build and maintain. Certain places may only be reachable in a very expensive fashion such as helicopter delivery. Moreover, due to disasters, natural debris and ruins may make delivery through air the only solution.

To solve this problem, we will be designing, building, and programming a drone that is capable of autonomously delivering a medical item from a starting point to a desired location. The drone will contain a box that can be loaded with supplies. This payload box will also implement a cooling mechanism or insulation methodology to ensure that the products remain preserved during delivery, keeping the temperature below a set threshold for different items. Upon receiving a signal, the drone will take off and begin moving towards a targeted location. To allow precise delivery, the delivery location location will be designated by a floor mat that is placed by the person requesting the delivery, the drone will use computer vision or RF communications to detect and maneuver itself to land on or near the floor mat. The floor mat dedicated to RF communications may operate based off angle of arrival of the drone communicated to the mat relative to the mat's communicated location. MediDrone strives not only to employ or hypothesize with state of the art technology alongside developing new technologies including but not limited to new methods in the field of medical class drones.

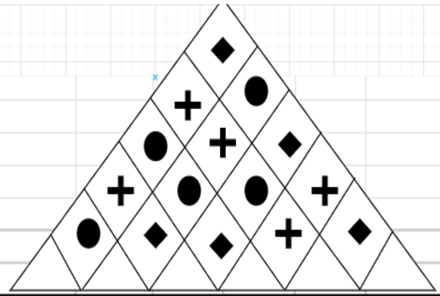
Safety is the overarching goal of the MediDrone, to complete a medical class drone but also guarantee developer and operator safety. While many hazards exist, the goal of safety can easily be met, and perpetuated through proper electrical, mechanical, and manufacturing safety alongside with process safety, allowing the payload to be secured to the drone and detached from the drone without injuring the operator or the drone itself.

2.2.1) Stretch Goals

The stretch goals for this project primarily include increasing the range, speed, and cost efficiency of delivery. Another stretch goal is to implement an actual refrigerated system. This is a harder goal to meet as the potential weight for this subsystem would need to be minimized and not interfere with the drone's main functionality of getting from location A to B. Another goal for MediDrone that may seem a bit ambitious is to create multiple landing pad locations. The vision here is that each landing pad will implement a QF charging station. From a larger scope, there would be a network of landing pads, maybe across a city or even a wider region, that would be able to charge every drone that lands on it. Each drone would then be given a payload to pick up and a set of coordinates to travel to. For the early development stages, creating two landing pads, each with wireless charging, would be a larger goal to meet.

2.3) House of Quality

In **Table 1**, below, we are able to make correlations between our functional requirements and the customer requirements. It's apparent there should be an emphasis on autonomy and range when it comes to the drone's design. Below you can see the correlation between the customer needs, the design requirements, and market competitors.



Project:	MediDrone
Date:	10/25/2022

			Functional Requirements					
		Direction of Improvement	□	▼	▲	▼	▲	□
Relative Weight	Customer Importance	Customer Requirements	Flight Speed	Payload Cooling	Range	Durability	Autonomy	Sensing
~24%	9	Urgent / On-Call Delivery	•	▽	•	▽	•	▽
~22%	8	Fast Delivery	•	o	•	o	•	o
~24%	9	Deliverable Cooled	o	•	o	▽	▽	•
~14%	5	Lower Expense	▽	▽	o	o	o	o
~16%	6	Ease of Use	▽	▽	o	•	•	o
		Importance Rating Sum (Importance x Relationship)	191	125	213	111	231	147
		Relative Weight	~19%	~12%	~21%	~11%	23%	~14%

Correlation Matrix	
Strong	•
Average	+
Little	◆

Relationships		Weight
Strong	•	9
Medium	o	3
Weak	▽	1

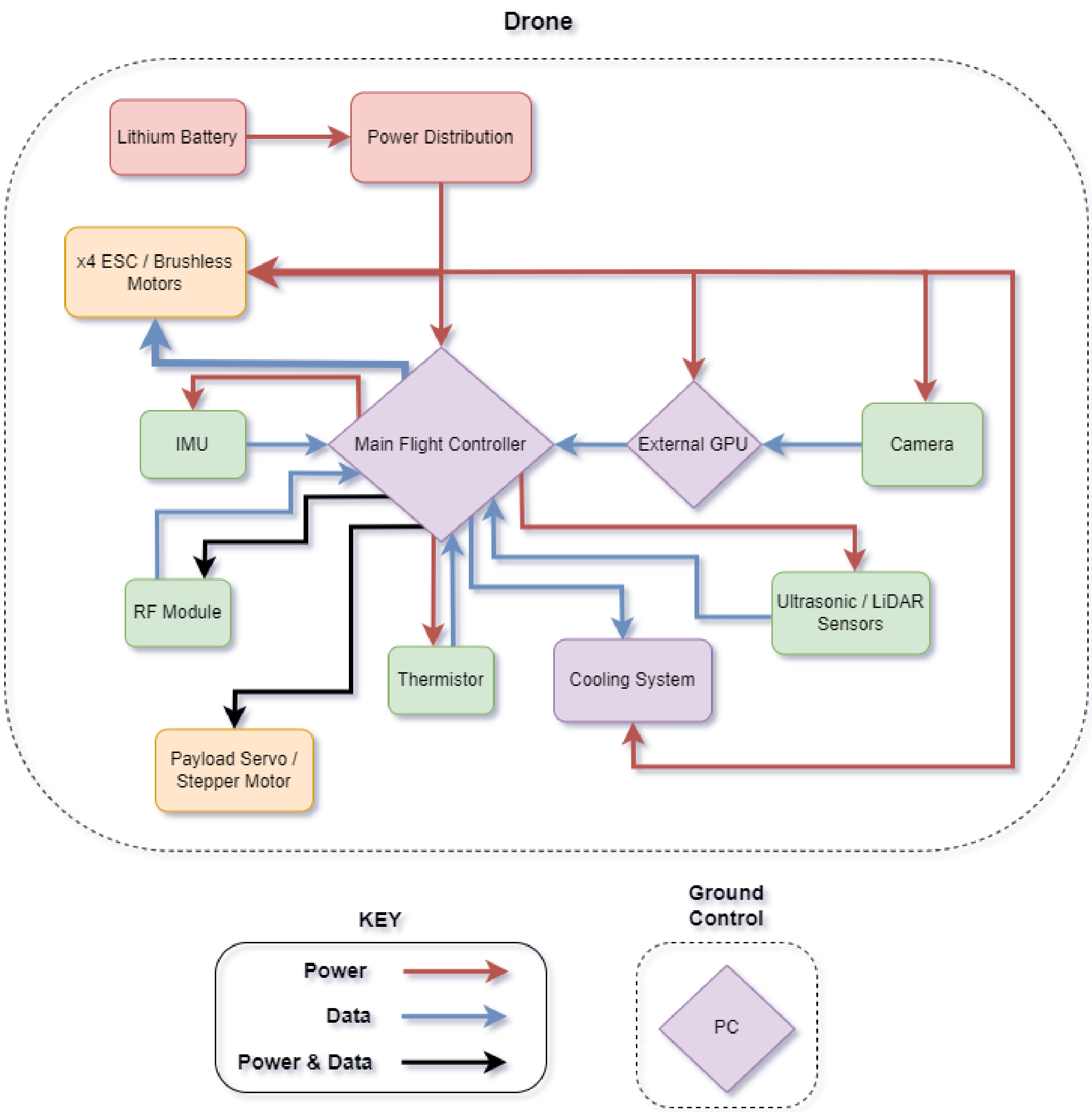
Direction of Improvement	
Maximize	▲
Target	□

Table 1

Revise House Table

2.4) Initial Design Overview

The initial design can be seen in **Figure 2** below. There have already been somewhat significant modifications to the different subsystems within MediDrone. This section will only cover a few of the more significant changes to the previous design. To begin, the flight controller previously had no hard design. In other words, it was left up in the air whether we wanted to design it ourselves, use an existing development board, or a combination of both. Also, the initial design didn't consider utilizing a GPS module which is non-negotiable for autonomy. Another larger change deals with the hardware layout as a whole. Previously, the thought was that the drone would focus around a main controller and simply read/write to different peripherals around the drone. With careful consideration, a new concept came about that better fits the drone's needs and overall should produce a more stable, quality, hardware network. One of the last major changes involves the landing system protocol the drone would undergo when arriving at its desired destination. Before, the idea was that the drone would operate under a machine learning model to pick up an image and decipher an exact landing position. This used extra hardware like a GPU and a camera. It also involved a greater software complexity than what will be implemented now. More details on these changes are found under the hardware and software design sections. Overall, the drone could be found functional with the initial design; again however, the team decided on different implementations to provide for what is thought to be a higher quality system.



Revision to Block Diagram: Cooling System in SD2 was Chapie double walled insulation, no active cooling

Figure 2

3. Standards, Specifications, and Constraints

3.1) Standards

The development of MediDrone follows many standards across the mechanical, hardware, and software designs. This section highlights many of the main practices used during the construction of the drone.

3.1.1) American National Standards Institute (ANSI)

1. Controller Area Network (CAN)
2. Unmanned Aircraft Systems (UAS)

3.1.2) Institute of Electrical and Electronics Engineers (IEEE)

1. Bluetooth
2. IEEE 1936.1-2021 IEEE Standard for Drone Applications Framework

3.2) Specifications

1. FAA requirements
2. Performance Requirements
    - 2.1. Flight
        - 2.1.1. Be able to carry a payload of 3lb
        - 2.1.2. Range of 1 mile
        - 2.1.3. Maintain distance of at least 5ft from any object
        - 2.1.4. Minimum maximum speed of 10mph
        - 2.1.5. Flight path should be within 10% of the length of the shortest path
        - 2.1.6. Drone should require no human input beyond initiating lift off to complete the delivery
        - 2.1.7. Initiate safe emergency landing when battery is at 5%
    - 2.2. Loading and Takeoff
        - 2.2.1. System startup should take no more than 30 seconds
        - 2.2.2. Loading box with supplies should be possible in less than 1 minute

2.3. Landing and Unloading
  2.3.1. Land within 5 feet of the target mat
  2.3.2. Unloading box with supplies should be possible in less than 1 minute
3. System Requirements
  3.1. Frame
    3.1.1. Should exhibit elastic behavior
  3.2. Propellers
    3.2.1. Be able to withstand RPM needed to gain altitude at a minimum rate of 1 m/s
  3.3. Motor
    3.3.1. Be able to power the propellers to the RPM needed to gain altitude at a minimum rate of 1m/s
  3.4. Battery
    3.4.1. Battery weight should not exceed 2lbs
    3.4.2. Battery should be rechargeable
    3.4.3. Battery should contain roughly 4Ah-6Ah of charge
    3.4.4. Battery is maintained and monitored by a battery management system.
  3.5. Camera
    3.5.1. Produce high enough quality imagery for classification algorithm to work
  3.6. Communication
    3.6.1. Communication should be possible within the flight range of the drone
    3.6.2. Due to a substantially long range from ground control, communication should be done over an RF signal.
      3.6.2.1. Drone's main flight control will be autonomous, but will log flight status data to ground control.
    3.6.3. Will require communication for manual overriding flight control.
  3.7. Sensors
    3.7.1. Sense temperature of deliverable to ensure temperature threshold [3.8] is kept: Not Included SD2
      3.7.1.1. Thermistor will be utilized due to it's high accuracy
    3.7.2. Camera [3.5] will be the main obstacle avoidance driver during forward flight.
    3.7.3. Flight landing, and any moving direction not in camera's range of view, will be handled by a combination of ultrasonic and/or LiDAR sensors

3.7.4. For simplicity, an IMU should be integrated into flight control
3.7.4.1. IMU must include: gyroscope, accelerometer, and inclinometer
3.7.5. All sensors should come digital.
3.7.5.1. Otherwise, all sensors should utilize the same ADC module to keep simplicity.
3.8. Cooling: SD2=Chapie
3.8.1. As it stands to date, the drone should keep load within +- 5 degrees fahrenheit of desired temperature.
3.8.1.1. This may vary per sensitivity of medical deliverable
4. Software Requirements
4.1. Determine initial flight direction

3.3) Constraints

The constraints discussed in this section directly affect many of the specifications and goals noted in the previous two sections.

3.3.1) Financial Constraints

The main constraint we face when designing the drone is financing. Drone's are notoriously expensive to construct and therefore will require us to optimize cost, quality, and functionality together. Our budgeting constraint can be broken down into a few areas of the drone. Firstly, the drone's range and payload will be impacted. The range for this prototype will be lesser than that of a commercial drone. The total size of the frame and motors will be decreased to support our budget; thus, decreasing the total drone's weight as well as the payload. With the drone's weight being reduced, the battery size must be reduced, and therefore will cut our total range down. Even with our smaller budget, it is still possible to optimize range with payload weight.

3.3.2) Payload Constraints

The drone will be able to carry smaller medical objects longer distances and carry heavier objects shorter distances. Regardless, the payload will be cut down due to our smaller drone size. The range may also be cut down if our stretch goal of developing a refrigeration system isn't completed. The item within the payload-medical box must maintain a temperature as according to the specifications above [3.8]. Without this system, the range may be decreased or the time of flight will need to be decreased to meet this temperature criteria. Also tapping into our budget, final drone construction tests will potentially be minimal to ensure damage to the hardware is reduced.

In Senior Design II the MediDrone could lift the payload, however flight time was sub 1 minute.

3.3.3) Drone Regulation/Licensing Constraints

On top of that, testing can only be done in certain regulated areas. Currently, we have been in brief contact with the UCF drone organization/club, First Person View Knights. FPV Knights is a student-run organization that gives UCF students the opportunity to learn about, design, and fly drones. FPV Knights have provided us information on not only how to get the design process underway, but also the requirements, regulations, and clearances that will have to be adhered to inorder to lawfully test our drone according to Federal Aviation Administration (FAA) rules and to Orlando and Orange County drone ordinances.

In order to become a drone pilot, we will have to complete a course to obtain a Remote Pilot Certificate from the FAA. For eligibility, we must:

- Be at least 16 years old
- Be able to read, speak, write, and understand English
- Be in physical and mental condition to safely fly a drone
- Pass the initial aeronautical knowledge exam: "Unmanned Aircraft General – Small (UAG)"

All members of the MediDrone team assumingly pass the eligibility requirements. Before the UAG exam, we will have to obtain an FAA Tracking Number (FTN) for our drone, by creating an Integrated Airman Certification and Rating Application (IACRA) profile while registering for the exam. For the UAG exam, we will have to study and be knowledgeable of the topic areas including:

- Applicable regulations relating to small unmanned aircraft system rating privileges, limitations, and flight operation
- Airspace classification and operating requirements, and flight restrictions affecting small unmanned aircraft operation
- Aviation weather sources and effects of weather on small unmanned aircraft performance

- Small unmanned aircraft loading and performance
- Emergency procedures
- Crew resource management
- Radio communication procedures
- Determining the performance of small unmanned aircraft
- Physiological effects of drugs and alcohol
- Aeronautical decision-making and judgment
- Airport operations
- Maintenance and preflight inspection procedures
- Operation at night

In order to keep our Remote Pilot Certificate, the FAA requires:

- Must be easily accessible by the remote pilot during all UAS operations
- Certificate holders must complete an online recurrent training every 24 calendar months to maintain aeronautical knowledge recency

Orlando adheres to the FAA's ordinances as well as having their own laws as to where drones are allowed to be operated. The FPV Knights fly their drones at Bill Frederick Park at Turkey Lake. We plan to test our drone flights at Bill Frederick solo as well as alongside the FPV Knights, so we can better practice drone flight operation.

Medidrone operated pursuant to all relevant laws during SD2 flight

3.3.4) Drone Safety Constraints

Testing can also lead to safety issues. A flying object carrying lithium batteries, which have a significant charge density, can lead to implications such as fire and/or explosions. Regulating our testing to meet certain safety criteria will be prioritized but also will potentially limit the total testing done on the drone and reduce the total drone's quality potential. Since we will be testing this drone at the Bill Frederick Park, we will have to be wary of where our drone is flying over, namely brush and wildlife. We will be testing our drone in the winter. This is arguably the time of year to be most wary of what our drone is flying over because it is the dry season. Because of this, we will be following along under the drone with the necessary means to extinguish flame if the battery explodes, namely a carbon dioxide fire extinguisher. By using a CO2 fire extinguisher we can ensure primarily that the fire will be put out and secondarily that any electronics may be salvaged from the damage. CO2 fire extinguishers are proven to be effective at suffocating fires at a minimal damage to electronics.

MediDrone batteries were kept in safe conditions, in non-flammable bags, charged, and operated properly.

4. Mechanical Design Details

4.1) Drone

The mechanical design of the drone takes a lot of consideration. First and foremost, this is not an average-sized drone. The drone will be greater in size than most drones seen flying around as it contains a payload.

4.1.1) Quadcopter vs. Hexacopter vs. Octocopter

Picking the type of drone will have a major impact on this project. We have taken into consideration designing a quadcopter, hexacopter, or octocopter. Each of these have different advantages and disadvantages.

TypeSD2 Quadcopter	Pros	Cons
Quadcopter 	• Compact design - allows for lesser use of materials in the frame, results in less complexity, cost, and weight • Less power consumption, we can use smaller and cheaper batteries to achieve the range requirement • Can use a smaller and weaker motor	• Cannot carry as much weight • If one propeller fails the drone cannot continue operating
Hexacopter	• Can continue operating if a propeller fails.	• Requires battery with

	● Can land safely if two propellers fail. ● Stronger lift allows for delivery of heavier objects. ● Larger design allows delivery of larger objects.	larger capacity to achieve range requirement ● Requires a stronger motor system ● Complex design
Octocopter 	● Can continue operating with up to three damaged propellers ● Allows delivery of much heavier and larger objects ● Can remain stable in various weather conditions	● Very high power consumption ● High cost of design

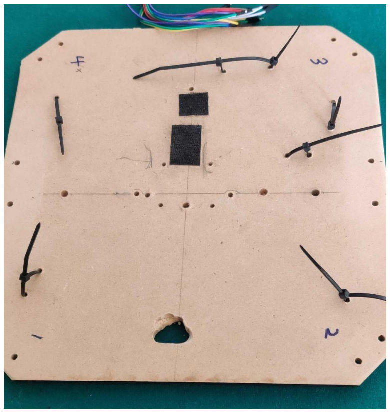
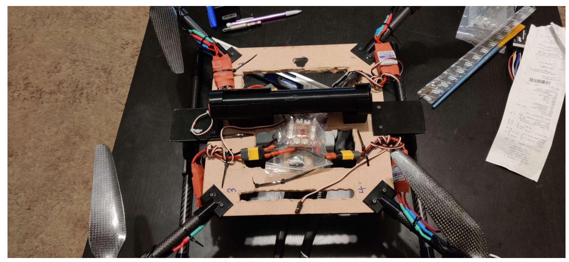
Table 2

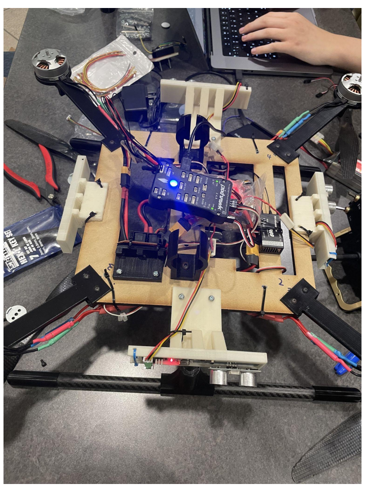
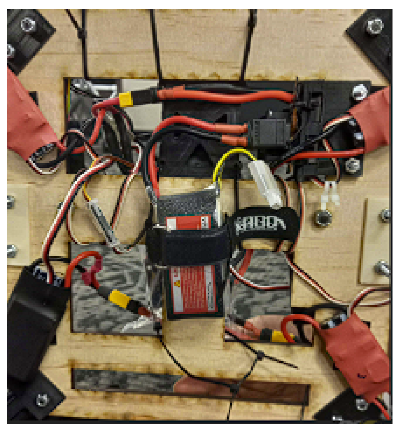
Given the project requirements, budget, and timeline, the quadcopter is a strong match for our needs. The lower complexity allows us to design our own frame, having freedom to accommodate the payload compartment, including any sensors and insulation mechanism we decide on. Furthermore, lower power consumption allows us to use a cheaper motor and battery while meeting the flight requirement.

4.1.2) Quadcopter frames

There are many components that go into a quadcopter but choosing the design for the frame is like choosing the foundation of a building. For the frame of a quadcopter there are multiple areas to consider before designing.

SD2 Medidrone utilized in house designs, carbon fiber legs, and wooden base. The payload was loaded on top of the wooden frame. The frame itself went through multiple iterations from a dense piece of MDF to a MDF base plate with large holes for weight loss. Our final version was a plywood base plate to help reduce the weight even more. The placement of the electronics and payload on the baseplate was essential to weight distribution. The payload was placed in the center and the electronics on the opposite sides of the payload to even the weight distributed across the baseplate.


→



→


4.1.2.1) Budget

A large budget doesn't mean a superior quadcopter. For every budget there are different options that'll benefit different projects. based on existing products such as Prime Air and matternet the budget would have to be quite large, almost around $5000. We’re not going to have $5,000 to work with so we will be using only what is necessary to meet our goals. Most options for frames found on Amazon and other websites range from$20.00- $250. Anything over $250 accounted for all of the components from motors to controllers for FPV(First Person View) drones.

Name	Cost	size	Materials	Landing gear
YoungRC F450	$20.99	9.96”L x 5.98”W x 2.17”H	plastic/brass	yes
Readytosky S500	$47.89	11.42"L x 7.09"W x 2.36"H	Carbon Fiber/Plastic	Yes
FPVKing 500-X4	$74.99	7.87”L x 0.63”W x 0.55”H	Carbon Fiber	Yes
Readytosky ZD550	$109.99	13.43”L x 6.65”W x 2.32”H	Carbon Fiber	Yes

Table 3

Material	Cost	quantity	Total	link
PLA spool	$20	1	$20.00	https://www.amazon.com/dp/B07PGZNM34?psc=1&ref=ppx_yo2ov_dt_b_product_details
Plywood	Free(used scraps)	1	$0.00	
Motors	$159.30	1 pack of 4	$159.30	https://www.amazon.com/dp/B01G3KIHS2?psc=1&ref=ppx_yo2ov_dt_b_product_details
PDB	$11.99	1 pack of 2	$11.99	https://www.amazon.com/dp/B0B2PF6YPQ?psc=1&ref=ppx_yo2ov_dt_b_product_details
Flight Controller	$189.99	1	$279.99	https://www.amazon.com/Controller-Wireless-Telemetry-Quadcopter-Multirotor/dp/B07NRMFTXL/ref=sr_1_5?crid=2R4SYUP3HIE54&keywords=pixhawk&qid=1682364719&sprefix=%2Caps%2C74&sr=8-5&ufe=app_do%3Aamzn1.f

				os.c3015c4a-46bb-44b9-81a4-dc28e6d374b3
Batteries				
ESCs	Free(from friend)	4		
Miscellaneous (screws,glue,tools, etc.)	$82.99	-	$82.99	
			$554.27	

Table: 4

4.1.2.2) Frame Size

Frame size is important because we want to be able to fit all the components on the frame and ensure that it's not going to hinder the flight capabilities of the quadcopter. An important term to know is the wheelbase. Wheelbase is the exact distance between motors, and this determines the propeller size. **Table 3,** below, simplifies the wheelbase, prop size, and suitable motors.

Frame Size(Wheelbase)	Prop Size	Motor	kilovolt(kV)
100mm	2 inches	1102 - 1104	6000+
120mm	3 inches	1104 - 1106	4000+
150mm - 180mm	4 inches	1306 - 1408	3000+
200mm - 220mm	5 inches	2204 - 2306	2100-2800
235mm - 280mm	6 inches	2205 - 2308	1600 - 2500
330mm - 350mm	7 - 8 inches	2208 - 2212	1500 - 1600
450mm - 500mm	9 - 10 inches	2212 -2216	800 -1000

Table 4

Final frame size

Frame Size(Wheelbase)	Prop Size	Motor	kilovolt(kV)
600mm	13 inches	4108	380Kv

Table:5

4.1.2.3) Frame weight

The weight of the frame needs to be considered because if the frame is too heavy then when we add the rest of the components to it. it won't be able to fly. It'll also drain more power from the battery to work causing the drone to only be able to fly for half or even less than the amount necessary.

The Final weight of the drone was around 5-7 lbs depending on the battery used, the material of the base plate, and after using velcro and zip ties instead of screws and nuts.

4.1.2.4) Frame type

There are 2 types of frames: freestyle and racing. Freestyle is flying the drone in a way that suits the pilot. It could mean doing acrobatic stunts or smooth flight paths for recording something using a mounted camera. Racing drones is a fast-growing sport where they shave off as much weight as possible without losing the crash resistance. It's all about the performance of the drone in racing which isn't what we are looking for.

The type stayed the same. Medidrone still used the freestyle frame concept.

4.1.2.5) Frame Layout

There are many layouts that have been tested for different purposes. The two layouts we're researching are the H and HX frames. A true H frame (**Figure 3**) is a frame in the shape of an H. It provides more space and is easier to build on. This gives us plenty of room to set our container on top and have space for a battery, electronics, camera, etc. and the HX frame (**Figure 4**) is similar to the H frame and has the H shape for the central body, but the four arms create an X layout. Some of the frames that we took into consideration include stretch plus frames and Z frames. There were too many cons for both and it would have led to more difficulties down the road. The stretch plus frame resolves the field of view problems with the plus frames, but it's prone to breaking from crashes. A z-frame is a stepped configuration where one base plate sits above the other. This reduces turbulent air to the rear motors while flying forward. But unfortunately, it isn't efficient enough for our needs.

The final version was a custom layout that follows the design shown below in **figure 5 .** This design was incorporated to ensure there was space for electronics and the payload.

Figure 3

Figure 4

4.1.2.5.1) Materials

The materials that we are really considering for this project is carbon fiber. A strong, durable, yet light material that we can get custom parts made if necessary. For the prototypes we'll be using wood, most likely MDF(medium density fiberwood), due to the cost of carbon fiber and the need to test our design multiple times to ensure we design the frame well.

Figure 5

4.1.3) Motors

A DC motor takes electrical power through direct current, and converts this energy into torque. The torque causes the component to move. It’s what makes the motoring action. The interaction of the magnetic and electric fields creates an electromechanical force.

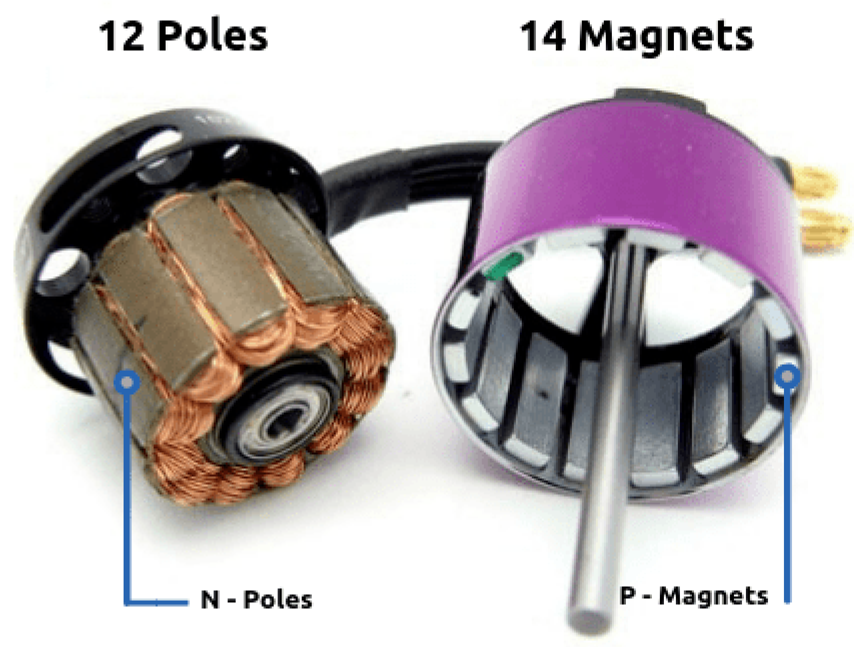


Figure 6

When choosing motors we need to have at least a rough idea of the size and weight of the drone we are building. The rough estimate of the weight will be around 10 lbs including the payload and 7 lbs without the payload. And the size will be around 7.5 x 4.5 x 5.2 inches. Our first decision is to figure out whether we're using brushless or brushed motors.

4.1.3.1) Brushless vs. Brushed Motors

Brushless DC Motors

A brushless motor is made up of two parts: the rotor and the stator. There is no point of contact between the two parts. The brushless motor uses electromagnetic induction created by the current that flows through its housing, surrounding the electric motor. The major benefit of brushless is high efficiency.

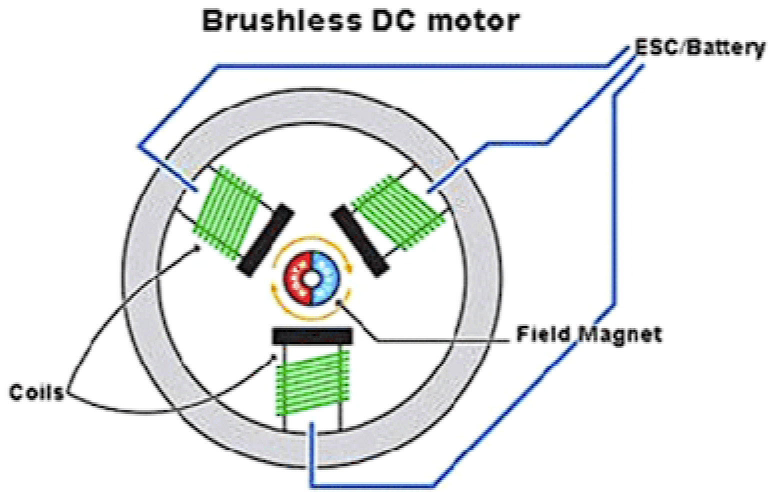


Figure 7

Brushed DC Motors

A Brushed motor consists of a rotor, the brushes connected to the rotor, and the magnets. It uses the brushes to mechanically control the direction of the voltage. The carbon brushes create a contact between the rotor and the magnets. An important thing to take note of is that a brushed motor is simple and is naturally the cheaper option. The major benefit of brushed motors is that it delivers greater torque.

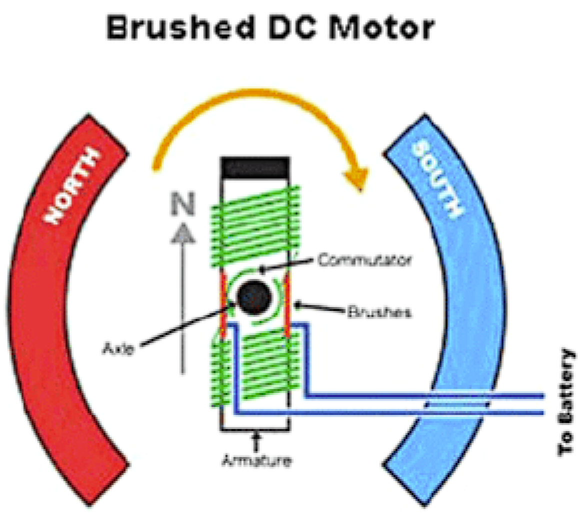


Figure 8

Here are some major features of brushless and brushed motors that are essential to us picking motors. Based on the information in the table below we decided to go with Brushless motors for their longer lifetime with minimal maintenance and because of their high speeds with no friction to cause any loss of efficiency.

Features	Brushless	Brushed
Commutation	Electronic communication based on Hall position sensors	Brushed commutation
Maintenance	Little to no maintenance	Periodic maintenance
Life	longer	shorter
Speed / Torque	Enable operations on all speeds with rated load	At higher speeds, brush friction increases and reduces torque
Efficiency	High	Moderate
Speed Range	Higher - no limitation due to mechanical contact	Lower - Mechanical limitations due to brushes
Electric Noise Generation	Low - because it has permanent magnets on the rotor, improves dynamic response	Arcs in the brushes will generate electric noise

Table 5

Once we've figured out we're using brushless out-runner motors.We refer back to **Table 4: Wheelbase** and based on the information there we take into account the size of the frame, the size of the propellers, the kilovolts necessary all so we can determine the motors compatible with the frame. Here are some of the options based on our research. Once we've decided on the motors we can look into our ESC controllers.

4.1.3.2) Motor Decision

In **Table 5,** below, has some motor terminology to better understand what we elaborate on and what we looked for when we were deciding on the motor. These terms and definitions were all directly taken from Astro Flight(https://www.astroflight.com/explanation-of-motor-terminology.html).

Term	Explanation
Kv - RPM Constant	The higher the voltage, the faster a motor spins. The Kv rating designates the RPM a motor will spin at full throttle when unloaded given an input voltage. To estimate the RPM of a motor, simply multiply the Kv value by the battery voltage. Kv is in units of RPM/Volt.
Kt - Torque Constant	The more torque a motor needs to produce, the more current it will draw. The Kt rating designates the amount of torque a motor will produce given a current. To estimate the torque of a motor, simply multiply the Kt value by current. Kt is in units of Inch*Ounce/Amp.
Rw - Winding Resistance	Winding resistance has to do with the electrical resistance of the motor. In the case of 3-phase brushless motors such as the Astro Flight 32xx series, the motors have three wires coming out of the housing, one for each phase. The amount of electrical resistance across any two of these three phases is the winding resistance. This parameter is very important for calculating the efficiency of a motor. The less winding resistance, the better your efficiency will be. Rw is in units of Ohms.
Copper Loss	There are two main ways for a motor to lose efficiency (losing efficiency means using battery power, but not contributing to the mechanical output). Copper loss is one of them. Copper loss refers to the heat generated when a wire conducts current. Copper loss is in units of Watts.
I_0 - No Load Current	I_0 (pronounced "i-naught") current has to do with the amount of current needed to run the motor when there is no load attached. Even if a motor is not powering anything, it still requires a small amount of current just to spin. I_0 is in units of Amps.
Iron Loss	Iron loss is the other main way a motor loses efficiency. Iron loss has to do with the power lost due to fluctuating

	magnetic fields in the motor and is directly related to I0 and the voltage applied to the motor. Iron loss is in units of Watts.
Efficiency	The efficiency of a motor is very important for a variety of reasons. First, motors with higher efficiency will be able to run longer on a given battery. Second, higher efficiency motors don't produce as much heat. This allows them to be pushed harder to deliver more power in continuous and burst applications. This also allows the designer to use a smaller and lighter motor. • Efficiency equals the amount of output power divided by the input power… • Efficiency = (Power Out) / (Power In) • Power Out is equal to the power that is left over after you subtract the copper and iron losses… • Efficiency = (Power In - Copper Loss - Iron Loss) / (Power In) • Finally, we can substitute our motor parameters into the equation to get… **Efficiency = (V*I - I 2*Rw - V*I0) / (V*I)** This is a very basic equation that can be used to get a rough estimate of the efficiency. There are more complex equations that will yield more accurate results. But this will give you a rough approximation of the efficiency given Input Voltage (V), Input Current (I), Winding Resistance (Rw), and No Load Current (I0). Efficiency is unitless.
Turn Count	The more turns a motor has, the slower it will spin. Turn count is how many times the wire is wound around the tooth of a stator. It is inversely proportional to the Kv of the motor. For instance, a 3210 4T has a Kv of 339 RPM/V. From this, it can be determined that the Kv of a 3210 8T is 165 RPM/V (339 RPM/V * 4T / 8T = 165 RPM/V). Further, the turn count is also directly proportional to the Kt of the motor. Using the same example, a 3210 4T has a Kt of 4 Inch*Ounce/Amp,

	which means the 3210 8T has a Kt of 8 Inch*Ounce/Amp. There is a common misconception that increasing the turn count is the same as gearing down. It is true that by doubling the turn count while maintaining the same input voltage and current yields half the RPM and double the torque, which is the same result as using a 2:1 gear ratio. However, spinning the motor at half the RPM will also half the maximum output power of the motor. This is because the maximum torque of all motors of a specific motor size (eg. 3210) is the same and unaffected by turn count (the reason they're all the same is because there is the same amount of copper loss). For example, the 3210 4T has a Kv of 339 RPM/V. We have a 50V power supply to use and would expect to get about 17,000 RPM (339 x 50 = 16,950). However, 17,000 RPM is too fast for our application. We want to run at half that speed, 8,500 RPM. To achieve this, we have 3 options: we can use a power supply of half the voltage (25V), we can apply a 2:1 gear reduction, or we can use a 3210 8T. We opt for the 3210 8T to avoid having to buy a new power supply and to avoid having to add a reduction system, but after looking at the application specs, we realize that we need 500 Inch*Ounce of torque for the product to work correctly. The 3210 series, however, can only supply 375 Inch*Ounces (any more and the excessive copper loss would cause the motor to overheat). The only option that will fit our needs is to use a 2:1 reduction on the 3210 4T. By doing so, we've achieved our desired RPM and also can afford a bit of overhead in torque (the reduction allows us to have a maximum torque of 750 Inch*Ounces).

Table 6

Everything shown in the table above is important, but some of the terms mentioned above are to just make it easier to understand what is going on and won't have that big of an impact on what motor we choose.

There are many features that are involved with picking the right motor for our drone. First of all we need to decide what motor size to go with. This is broken down to what is the size of the drone and the weight of the drone. In our case we currently don’t have a physical representation of our drone because we are still in the design and research phase. For the purpose of this project and based on educated guesses on what everything will weigh we are stating that the weight of the drone will be around 12 lbs. This includes the frame, batteries, wires, flight controller, motors, payload, and more. Now that we have our weight we can use the measurements from our solidworks model to give an accurate size of what the drone will be.

The drone is currently 25.89” x 21.89”. Even though this is the prototype, the final version will be around the same size due to the size of our payload. Using the size of the drone we can determine the right propellor size. Based on **Table 6: Wheelbase** we have various options for the propellor size and motor choices. For this size of drone we’ll convert the size of the drone to mm(inch to mm is 1:25.4). Therefore the size is 657.606mm x 556.006mm which according to the table means the propeller size will be over 10” long. The next step is to figure out the thrust to weight ratio.

Thrust to weight Ratio	Drone Application
2:1	Aerial photography
4:1	First-person view videos
5:1	Drone Races
7:1	Extreme acrobatic flying

Table 7

Based on the table above and this equation:

(thrust = thrust-to-weight-ratio x total drone weight),

We decided to go with a 2:1 ratio because it will be stable enough to carry vials and other items without causing them to shake around too much. We want our drone to be stable during travel and to have the speed necessary to travel from point A to point B in a reasonable time. After this, we determined the thrust needed from each motor using:

Thrust per motor = thrust/number of motors.

So, the values we’ve gotten for our drone:

- Thrust-to-weight ratio: 2:1
- Total Drone weight: 12 lbs

- Number of motors: 4 motors

This will show equations and values that we get from plugging in our numbers

Thrust = 2 * 12 lbs
Thrust =24 lbs

Thrust per motor = 24 lbs / 4 motors
Thrust per motor = 6 lbs / motor

Most drones are measured in grams. So, we are converting from lbs to grams to make sure there are no errors in calculations.

1 Lb = 453.6 Grams
Thrust per motor = 6 * 453.6 g
Thrust per motor = 2721.6 grams

This means each motor needs to be rated to do 2800-2900 grams of thrust so we don't force the motor to constantly push at its max.

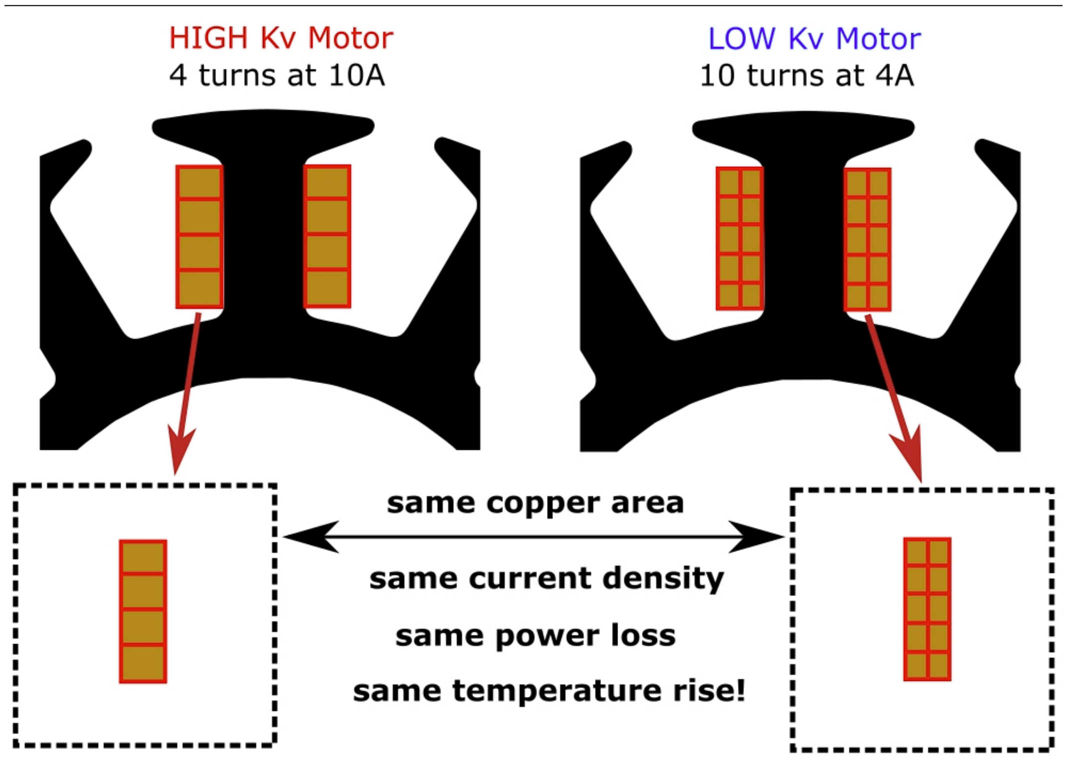
Now that we know how much thrust we need from the motors and the size of the propellers. We Have to decide whether we want motors with high Kv or low Kv. What is Kv? Kv stands for Kilovolts. The Kv rating designates the RPM a motor spins at full throttle when given an input voltage. The formula mostly used to estimate RPM of a motor is by multiplying the Kv value by the battery voltage. Kv is in units of RPM/Volt.

Formula: RPM = Kv * Voltage

High Kv	A high Kv motor has fewer winds of thicker wire that carry more amps at fewer volts and spin a smaller propeller at higher revolutions.
Low Kv	A lower Kv motor has more winds of thinner wire. It will carry more volts at fewer amps making it more efficient. It also produces higher torque and swings a bigger prop.

Table 8

Based on the table above and the image below It is clear that with a low Kv motor and the right amount of voltage. We can easily get the thrust we are looking for. The only thing to ensure is that we have a battery that is capable of sending enough voltage to the 4 motors to last for the time required.



https://things-in-motion.blogspot.com/p/blog-page.html

Figure 9

Name	Kv	Cost per motor	Power	Thrust	Current	Voltage
Racerstar BR4114 racing edition	400Kv	$22.99	566 W	2920 g With 17 x 4 prop	25.5 A	22.2 V 4 - 6s
BrotherHobby Avenger 2810	860Kv	$32.99	1202 W	2983 g With 8 x 4.5 prop	43.4 A	27.7 V 7S

BrotherHobby Avenger 2816	1050Kv	$42.99	1294 W	2921 g With 8 x 4.5 prop	54.6 A	23.7 V 6S
T-Motor Cine66	925 Kv	$45.90	1238 W	2992 g With 9 x 5 prop	50.5 A	24.5 V 5-6S

Table 9

The different motors shown above were all taken from BrushlessMotorsParade(https://brushlessmotorsparade.com/thrust/2901-3000). This site gave us the most insight into the motors and the details needed to choose which one we will be using. The site even includes dimensions for the motors and the prop sizes they are compatible with and more. After all this research and looking at great options. The motor that we will be testing first is the 5005 IPE 440Kv MAD COMPONENTS brushless Motor. This motor is $59.00 which is around the average price range for a motor of this size and power. We also noticed that other large drones were using this motor while having large payloads.

Figure 10

Motor	Prop	Throttle [%]	Voltage [v]	Current [A]	Torque [NxM]	RPM	Thrust [gf]
MAD 5005 IPE 440K v V2.0	Fluxer-pro 17 x 5.8in Carbon Fiber	30	14.53	0.83	0.047	1628	205
		35	14.52	1.24	0.064	1902	289
		40	14.51	1.69	0.08	2121	365
		45	14.51	2.18	0.094	2338	441
		50	14.5	2.88	0.116	2579	553
		55	14.48	4.04	0.148	2901	708
		60	14.45	5.1	0.173	3148	837
		65	14.44	6.14	0.194	3370	948
		70	14.42	7.26	0.215	3571	1064
		75	14.4	8.5	0.239	3781	1181
		80	14.38	9.77	0.259	3972	1294
		85	14.35	11.42	0.292	4135	1451
		90	14.32	12.87	0.311	4318	1564
		95	14.29	14.53	0.335	4498	1689
		100	14.25	16.98	0.368	4747	1819

Table 10

The values in the table above were calculated using the various formulas and variables explained earlier.

The motors used for the final iteration: Tarot TL68P07 6S 380KV 4108 Multi Rotor Disc Brushless Motor.

This motor checked off all the requirements necessary for the drone. The only restraint it had was that it was a 6s motor so the battery had to be changed from 4s to 6s.

Figure 11

4.1.3.3) Propeller Decision

The propeller is an important component to choose. There are 5 main variables that need to be considered: Size, Pitch, Blade Configuration, Material, and design. The first thing to know about propellers is that everything is linked and it's all one big compromise.

Size - This is the first thing to consider because it is directly linked to thrust, responsiveness, and the amount of grip the drone will have in the air. A larger prop will sweep more air but this also means the propeller will take more energy to get spinning. A larger propeller will take longer to respond to inputs from the motors and just consume more power. The benefit of having larger props means more thrust, a better grip, and is easier to control in the air. A smaller prop will respond faster to inputs because they sweep less air so they respond faster to inputs and use less power. If we used the smaller props for our drone it wouldn't be as effective because there wouldn't be much lift and the drone would likely not become airborne. A good example is wheels on a car. Like each car has its wheels that are a certain size. If the wheels on a car are too big or too small then the car is most likely going to run into other problems down the road. The efficiency would fall down so fast and it could make the car push harder just to achieve the bare minimum.

Pitch - This refers to the angle of the blade on propellers. The pitch of a propeller determines how much the propeller will move in one revolution. The best way to describe this is the gearing of a bike. A bike can have different sized gears on it and each gear is used for certain situations. Like if you are biking on flat land and you keep it on a low gear the gear is smaller and needs less force to rotate the wheel but it won't move as far. With a high gear the gear is larger and needs more force to turn and takes longer to do one rotation but because of this it moves a lot farther. The same thing happens with pitch. A high pitch propeller will result in more overall thrust and top end speed. A low pitch propeller will result in more low end torque and increased responsiveness. It will use less power and only be efficient at low speeds. For the purpose of this project we will be using a high pitch propeller because of the need to carry a large payload and have smooth control in the air. There shouldn't be a need for a lot of directional changes and it won't be flying in a confined area.

Blade Configuration - this refers to the number of individual blades on a propeller. The most efficient option would be a single blade but that is not practical due to its imbalance. So we will be using two blades because it is more efficient for its size, and there will only be two blades creating drag in the air. According to physics and fluid dynamics, increasing the number of blades is not as efficient as increasing the size of the blade. Increasing the number of blades will increase the thrust and grip in the air but this will come at the cost of responsiveness and more power consumption.

Material - there are many different options at a variety of price points. From cheap plastic to carbon fiber covered in fiberglass. For our props we want them to be rigid and durable with a minimal flexibility to ensure that there is some give and a point of failure in a location where if the prop breaks we can just replace the prop instead of the full drone. So, we have decided to use carbon fiber props because they are strong, light and will flex lightly.

Propeller design - The design of the propellers isn't just so it looks cool. The design can have a drastic effect on the thrust and flight time. With a bulky design there would be more energy needed to keep the drone flying which leads to shorter flight times. The design of the prop needs to be efficient because we are already going for a larger payload and we want the drone to have enough flight time as well as move effectively in the air without more drag or power consumption than needed. The image below shows a few examples of different designs with varying pitches and materials.

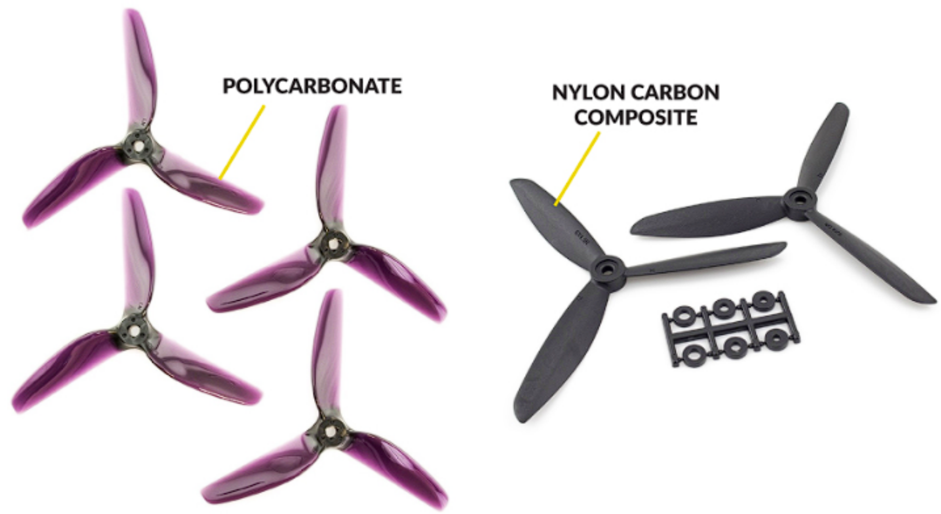


Figure 11

We’ve decided to go with the FLUXER-pro 13 x 4.4in Carbon Fiber propeller. This is the 1st propeller we are testing because even though the motor stated at the end of the last section uses larger propellers. We should be able to use these propellers to have enough thrust and not drain the batteries too fast.

Figure 12

4.1.4) Motor Mounts

In the case of our project, we will be designing our own motor mounts to ensure that we can adjust to whichever motor we choose to work with. Whether the mount covers the motor entirely, or is just a flat piece that we can screw the motor onto. A good amount of the designs I've found online are similar to the one in the **figure** below where the motor mount is mounted to a circular rod, most likely a carbon fiber rod that is supported by the main frame. The actual mount doesn't encase the motor but is just the flat piece on top that the motor is screwed down to.

Figure 13

The design we would like to go with will elevate this more for safety, wire management, and overall structural integrity. The design in the **figure** below shows something closer to what we would like to add to the quadcopter and we would incorporate how the mount is attached to the carbon fiber rod using the design shown in **figure above**. Our ability to 3D-print in PLA and test these parts will be essential because we need to test where the mounts are the weakest and have a chance of fracturing or failure. Once tested we can look into getting it done in a stronger material like carbon fiber and fiberglass the sections that might fail the first. Another thing we are taking into account for this design is if the worst happens and our drone crashes. We will leave a single point of failure on purpose to ensure that the drone's motors don't get destroyed but the mount will fail first if the situation arises. Creating a point of failure was a great idea we thought of because we would rather have certain parts break that we can replace with ease versus a motor or a controller that would cost us more than necessary.

Medidrone employs in house designed motor mounts. The motor mounts went through many iterations, from the 1st ideas based off of mounts found online to the current and final design. The final motor mount entails both the mount for the motor and the arm connected to the base plate. This was designed this way to ensure that the motor would not rotate like how it did on the previous version.

Figure 14

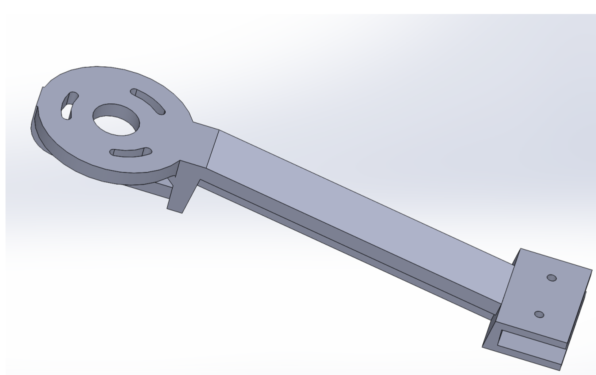


Figure 15: final Motor Mount

4.1.5) Sketches

This section is where we take all this research and our requirements into consideration and are brainstorming the designs for different components. Each part has some custom 3D printable component that we are focusing on making to adjust to the components we need the most.

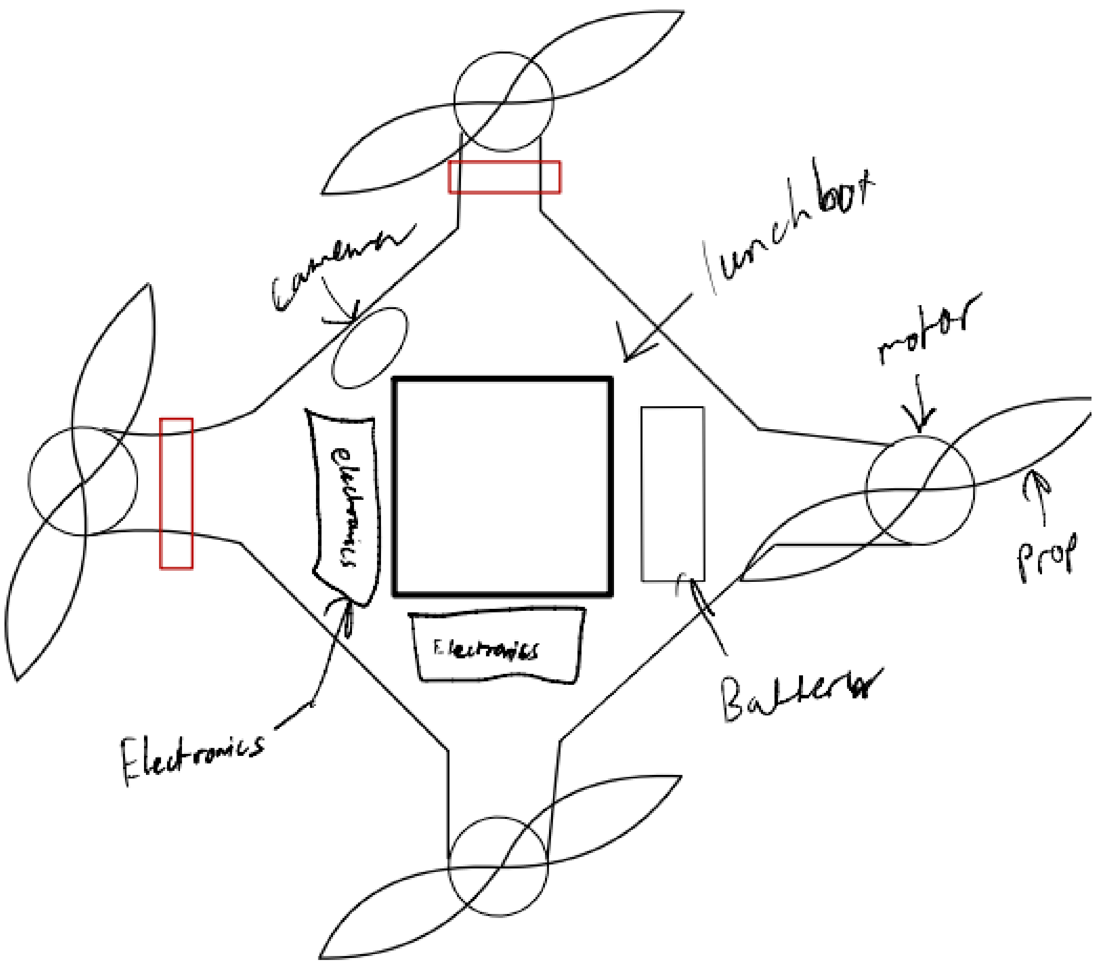


Figure 15

SD2 Medidrone Employed the design of Figure 15

This sketch is the 1st sketch we made based off of the basic components we thought were necessary. It includes a visual for the motors, propellers, frame, electronics, batteries, payload, and camera. When we finalize the design there will be one sketch with the components and another with a wiring diagram to explain how everything will be wired.

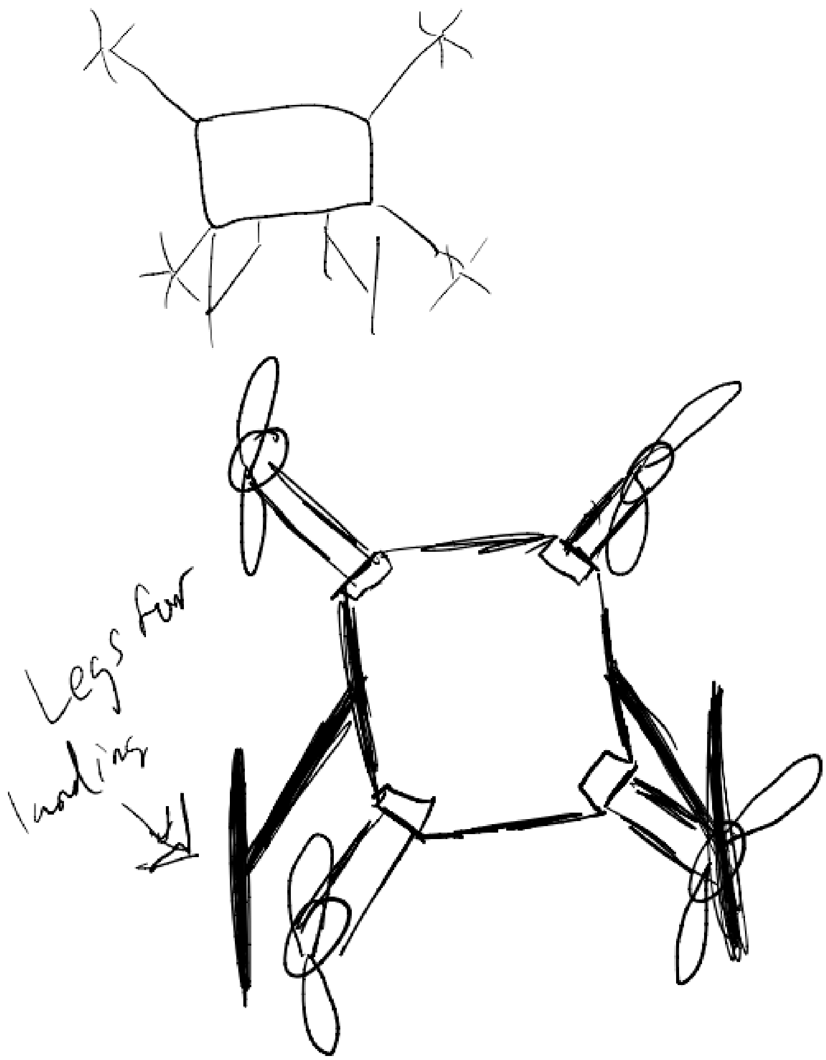


Figure 16

This sketch depicts our idea of using carbon fiber rods to create the X layout as well as use them for the landing gear. The purpose of doing this is to decrease the weight of the drone as well as make the landing gear structurally sound and a little flexible.

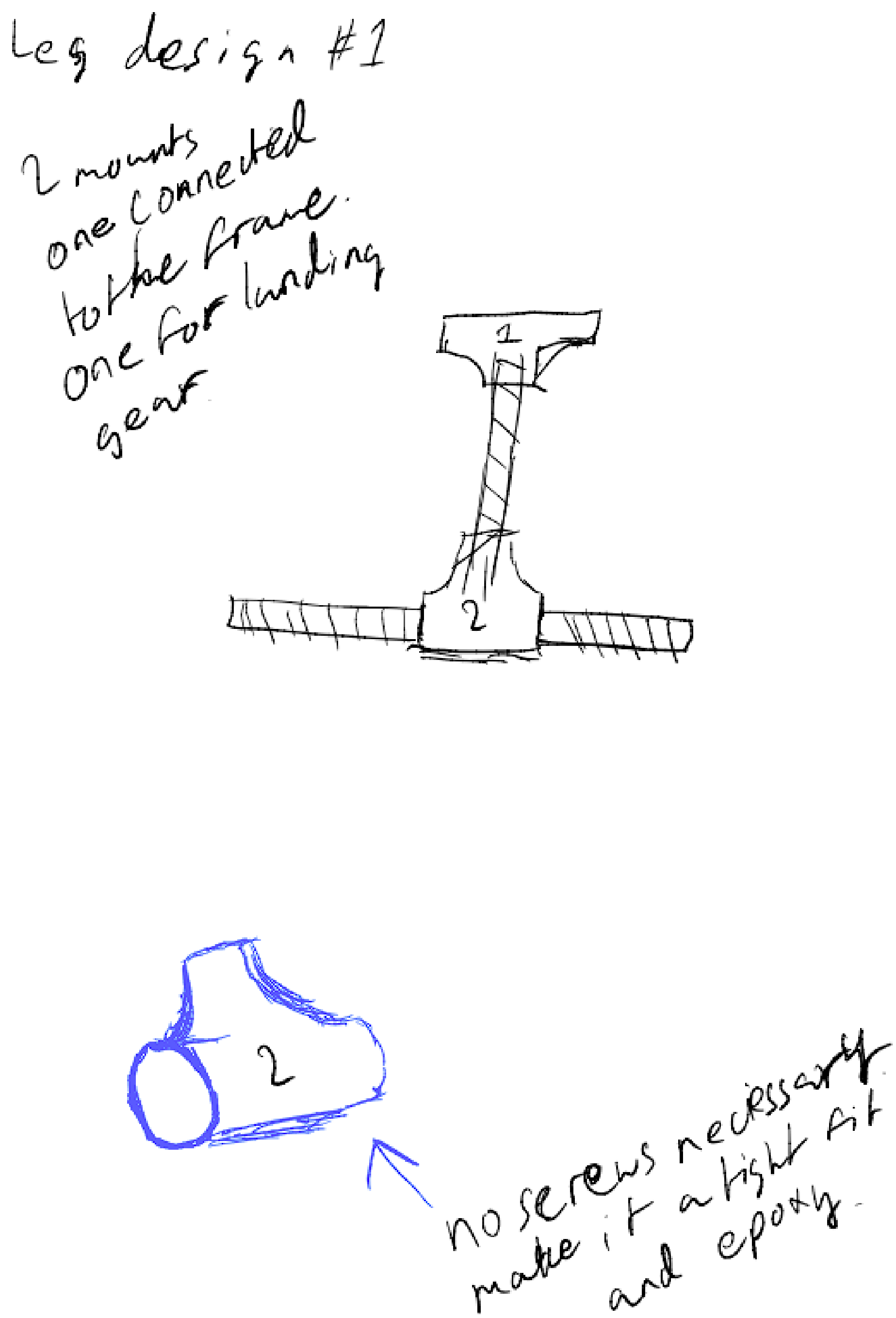


Figure 17

The top comment - “2 mounts, one connected to the frame. One for landing gear.” The bottom comment - “No screws necessary, make it a tight fit and epoxy.” Both mounts will be 3D printed out of PLA during the prototyping phase and then we will either increase the infill to make the parts stronger or change the material to one that has better properties for the structural integrity.

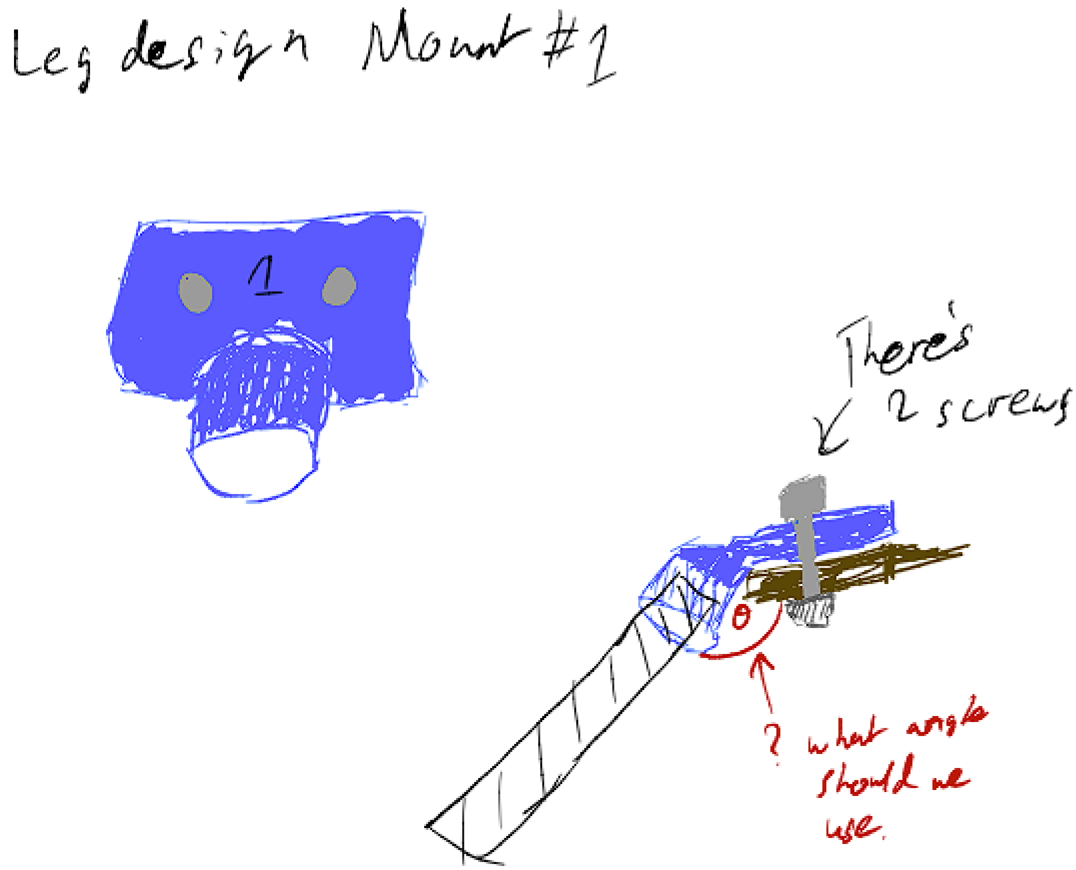


Figure 18

Here is another visualization of how the carbon fiber rod will be attached to the main baseplate. We weren't sure what angle we would be using at this stage in the process because we didn't know whether we were going to be adding the payload on top of the baseplate or below it. After some research and design tests we decided it would be best to put the payload underneath the baseplate and use the top of the baseplate for the electronics, batteries, and more. The components on top of the baseplate will be used to help the drone be aware of its surroundings as well as give us a clear signal. If we were to place the electronic components underneath the baseplate they would be closer to the carbon fiber rods which have a tendency to mess with the radio waves that we are trying to use. This would also make our drone top heavy because we are expecting the payload to be most of the weight in the drone. Therefore if it is below it'll be easier to control the drone while in the testing stage. Once we change it to autonomous, this will also allow the drone more time to react if the drone leans in a certain direction due to the distribution of weight.

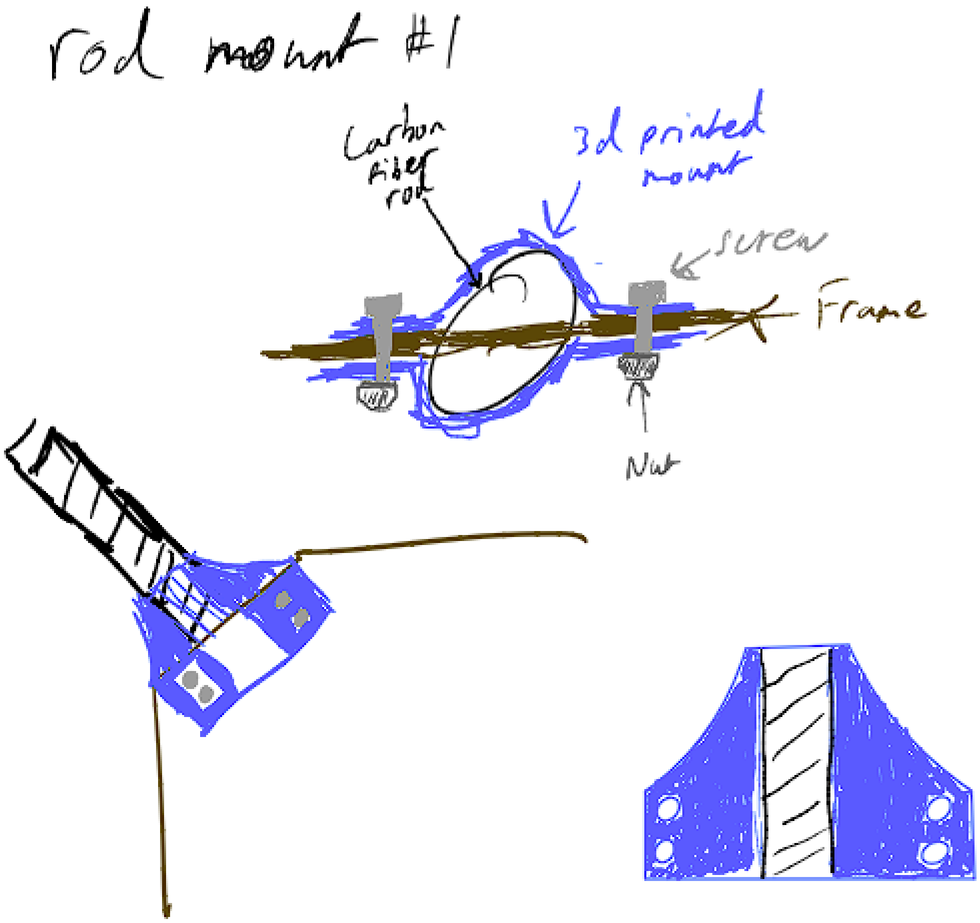


Figure 19

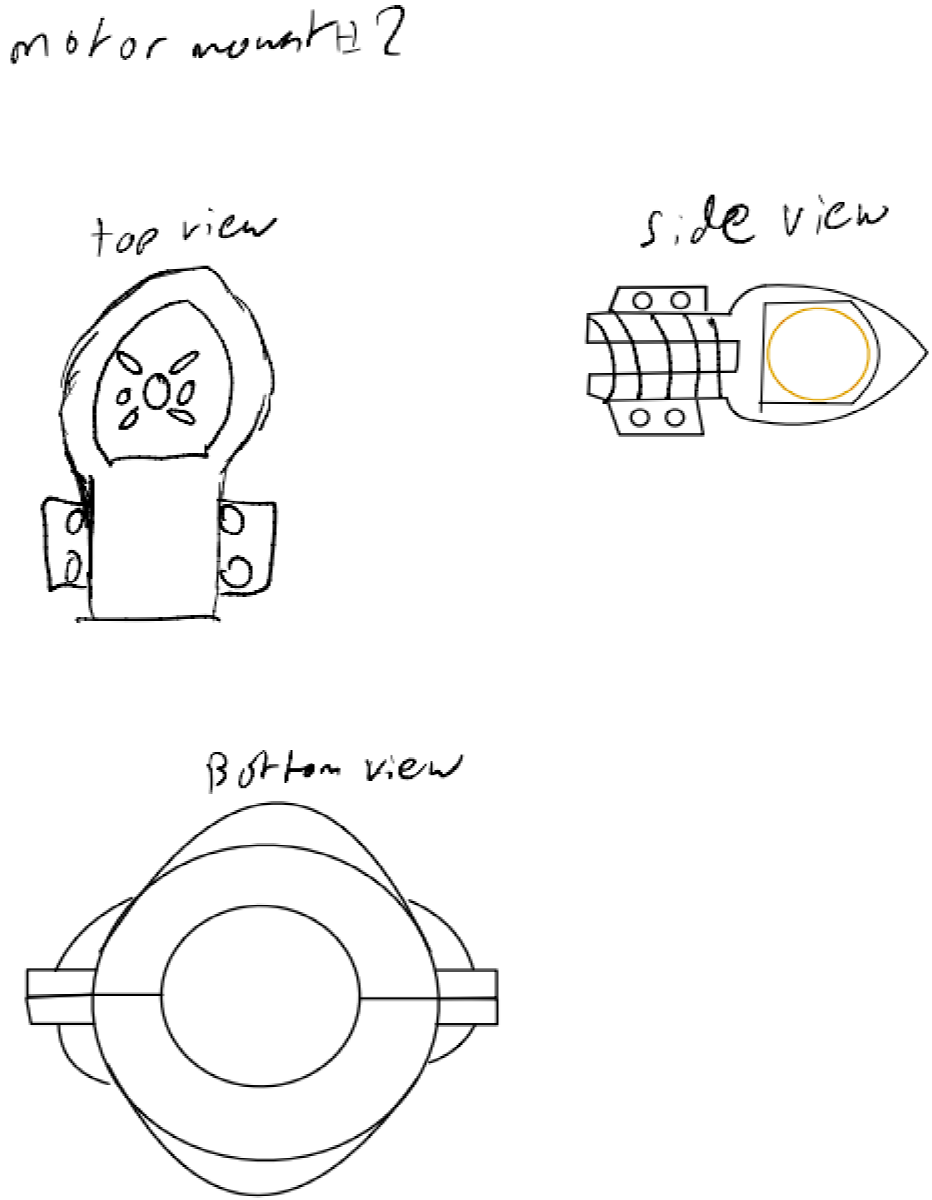


Figure 20

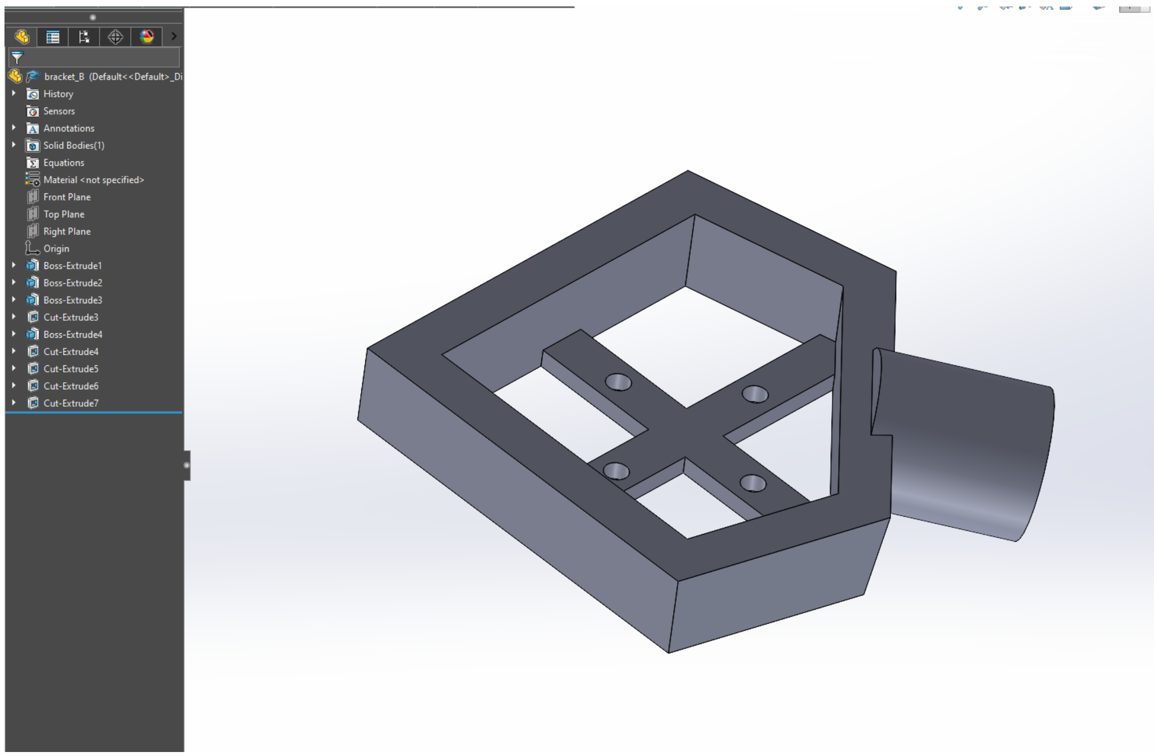


Figure 21

The image above contains the design for our 1st motor mount. This design is built to create a housing for the motor to attach to the base but still have undisturbed airflow. The bracket is set at a 10 degree angle to counter the initial angled bracket from the baseplate. This is to ensure that the propellers will be out of the way of the electronics and allow for there to be a housing to protect the motors, wiring, as well as the electronic components. In **IMAGE x** the motor mount is 2 components that clasp around carbon fiber rods using screws. We decided to just make a tight fit and use epoxy to ensure that it stays where it should. The next improvement we were thinking of adding is a notch and a screw to ensure that when we set the motor mount into place, there is a guide to follow. This will also make sure that it is facing upright and not at any angles.

4.2) Design

Computer-Aided Design (CAD) is imperative for the success of the project. By giving us the flexibility to specify exact dimensions and shapes for the components we design, CAD modeling allows us to customize our design such that every component of the medical drone fits. Furthermore, CAD modeling allows us to specify the material of each part in our design. Having the flexibility to create a 1:1 design of what the real world medical drone would look like allows us to perform realistic static and dynamic analysis and simulation. As such, there will be no surprises to how the drone performs in the real world.

4.2.1) Autodesk Inventor vs. SolidWorks vs. Creo Parametric

This section highlights the various differences among the following three programs. All three provide insight on the drone's mechanical structure.

4.2.1.1) Inventor

Inventor is a CAD modeling software created by Autodesk. It was created as an alternative to SolidWorks and has functionality of various editing forms. It is compatible with the rest of the Autodesk software, as well as has a drawing format for the models.

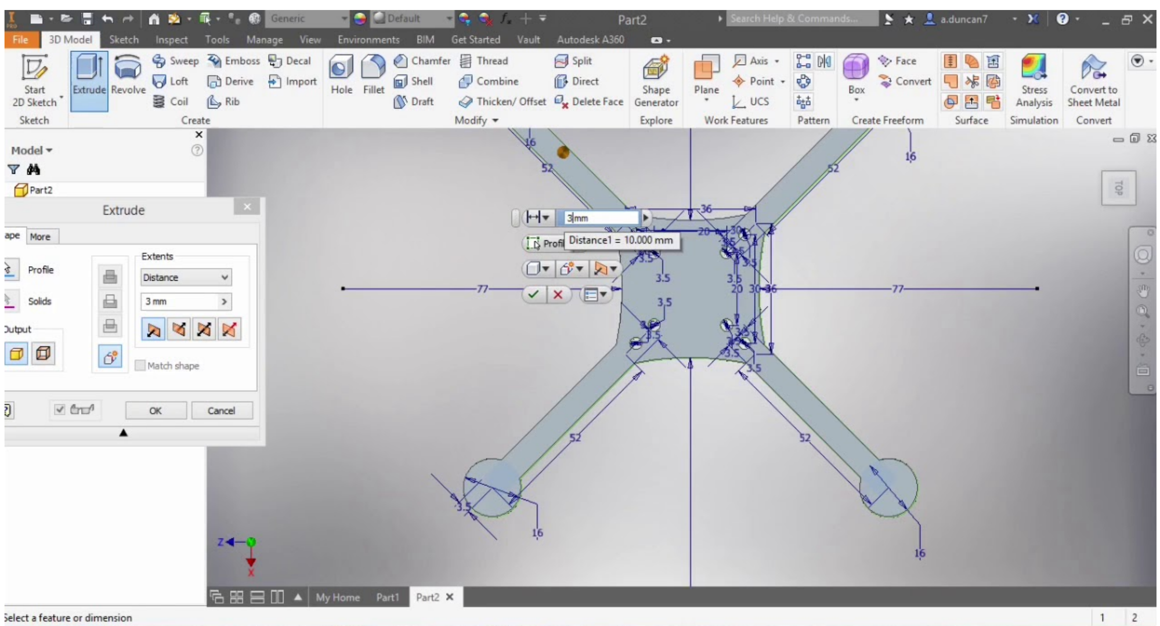


Figure 22

4.2.1.2) SolidWorks

Solidworks is a software that is often used to make CAD models. It uses a parameter-feature based approach in which the parameters are simply constraint values that define the geometry of the shape, the features are the blocks of the part. Below is a sample drone modeled in SolidWorks showcasing the ability to create real objects in a virtual space.

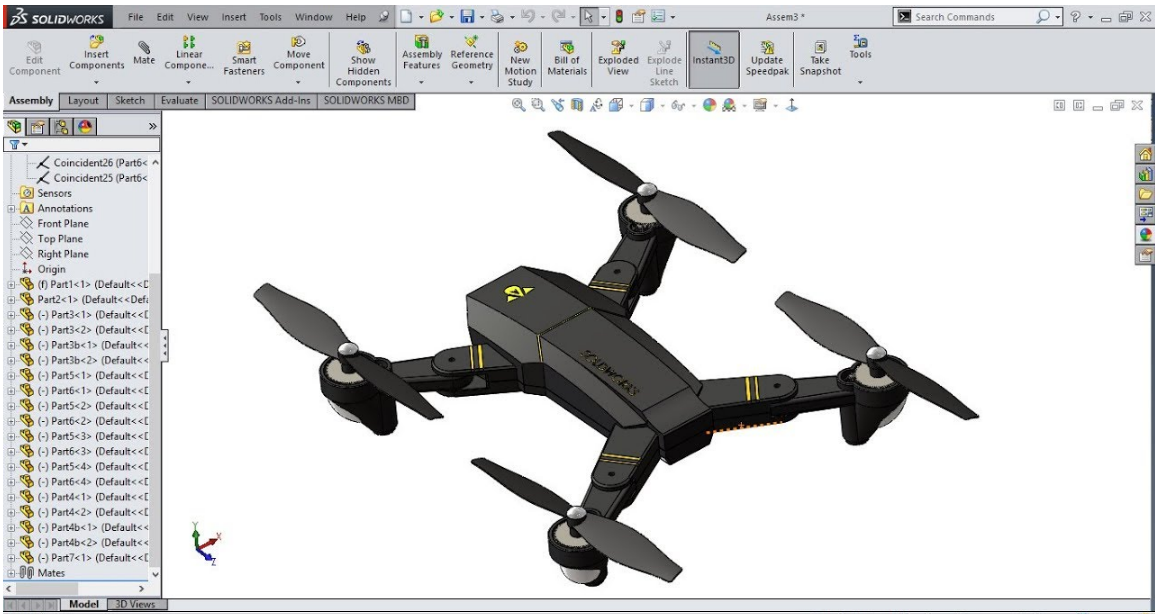


Figure 23

4.2.1.3) Creo Parametric

Creo Parametric is a widely used CAD modeling software that is well known for its low computational requirements. It provides parametric feature solid modeling, as well as Finite Element Analysis and simulation.

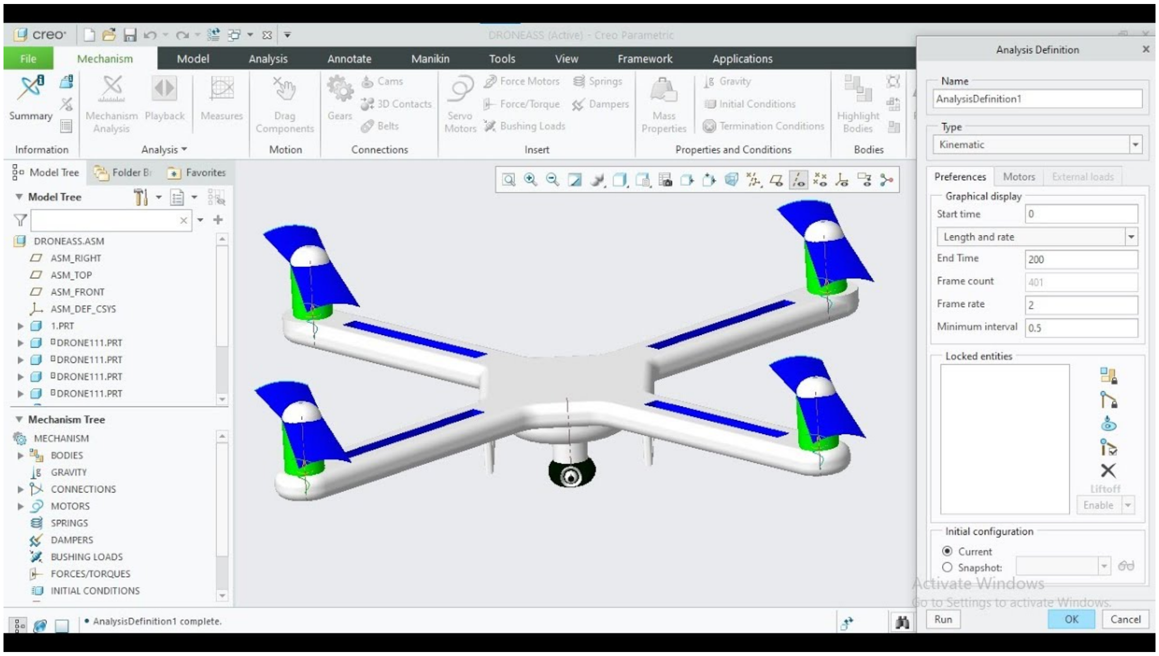


Figure 24

4.2.1.4) Pros & Cons

	Inventor	Solidworks	Creo Parametric
Pros	• Ability to seamlessly interact with the rest of Autodesk's platforms • Collaborative Work View • Assemblies work by setting constraints, joints, and parameters • Animations and simulating movement	• History based modeling allows the designer to easily make edits • It is possible to model multiple solids in a single part file • It is possible to make configurations for the same part file • Solidwork assemblies can be easily	• Reliable, as it is widely used in industry by companies such as Lockheed Martin and NASA • Very low computational requirements, as low as 4 GB RAM and 2GB disk space. • Concept Design allows for design of freeform shapes easily.

	can easily be done inside the program itself	exported to various file types, including IGS and STEP, which can be used for analysis in Ansys	• Supports exporting to IGS and STEP file types.
Cons	• Heavy computation power needed to run	• Errors are often not detailed and lead to a difficult learning experience • Finite Elements Analysis accuracy is not reputable	• Cannot make changes to a child part in assembly process • User interface is difficult to understand

Table 11

4.2.1.5) Choice of SolidWorks

In order to create CAD models of our design, the group chose to work with the SolidWorks software. Solidworks allows the group to quickly iterate on designs. The cad models are compatible and can easily be exported to other software, for uses such as 3D printing, finite element analysis (FEA), and physics simulations. SolidWorks has a free student license, which makes it a viable option to proceed with. Prior experience of the group is a plus in selecting SolidWorks as more time can be spent on optimizing the design rather than learning new software.

4.2.2) Drone Designs

Drone Design 1:

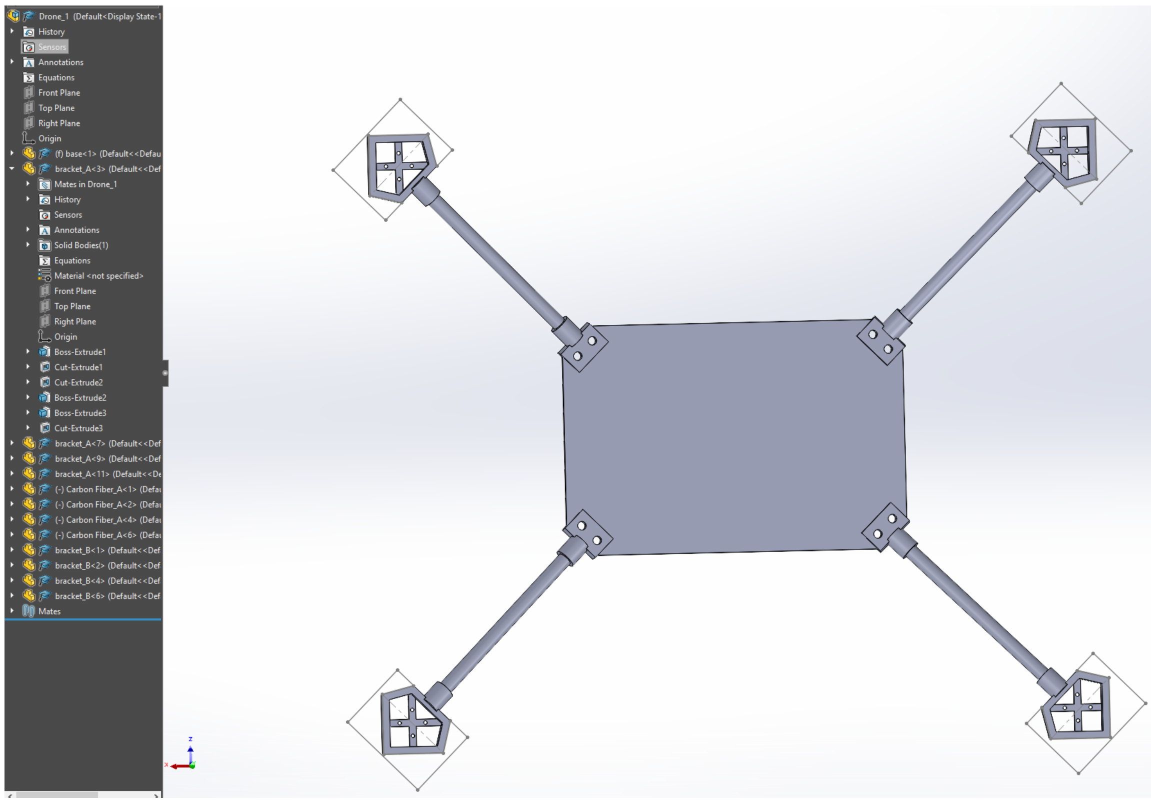


Figure 25

After we researched the information we needed. We decided to go for a larger frame to make sure that we have space for all the components.The **image** above is the bare bones of the 1st design which includes the base of the drone, the arms, brackets, and motor mounts. There are currently no fasteners as this is the 1st edition and I need to add the landing gear and adjust the design to add the electronics and other components.

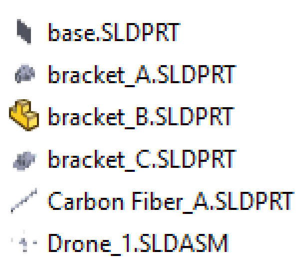


Figure 26

The way that we are naming our major components will be the original name like frame, drone, motor, etc. with a number attached to it to symbolize which generation it is. For example:Drone_1. For similar components like brackets that look similar but are used to hold different components will be differentiated using a letter after the name. For example: bracket_A.

Drone Design Landing Gear:

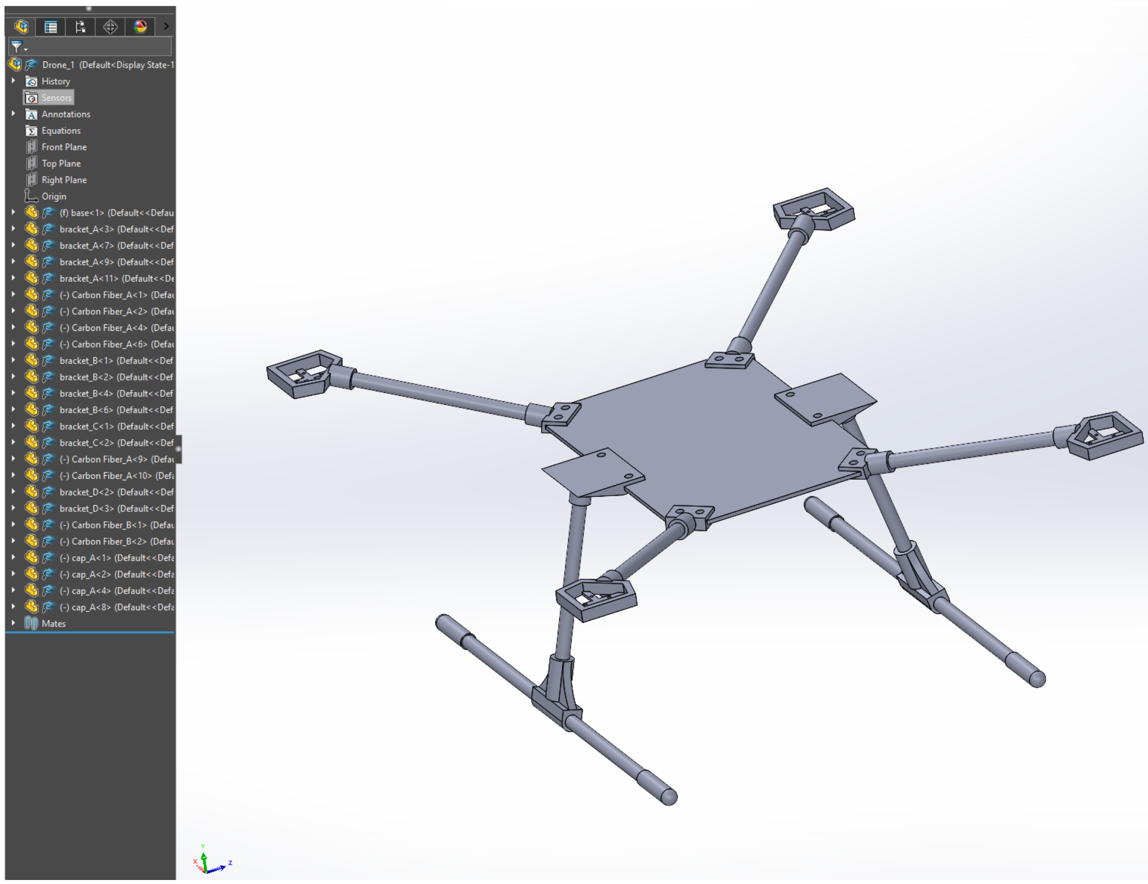


Figure 27

The image above showcases the initial version of the landing gear. This landing gear is inspired from the other drone projects we have seen in the past as well as the drones we've found online. Based on the prior designs we decided to mount the containment unit underneath the platform in order to ensure that the top of the baseplate can be used for electronics and other components. This will also help with cable management and if we end up putting a body over the drone for aerodynamic purposes then it would be best to keep the unit on the bottom so that it is easy to access for the operator. Our idea is to base the unit off a drawer with drawer slides or at least something similar to it.

Drone Containment unit: SD2 Did not use payload concealed design

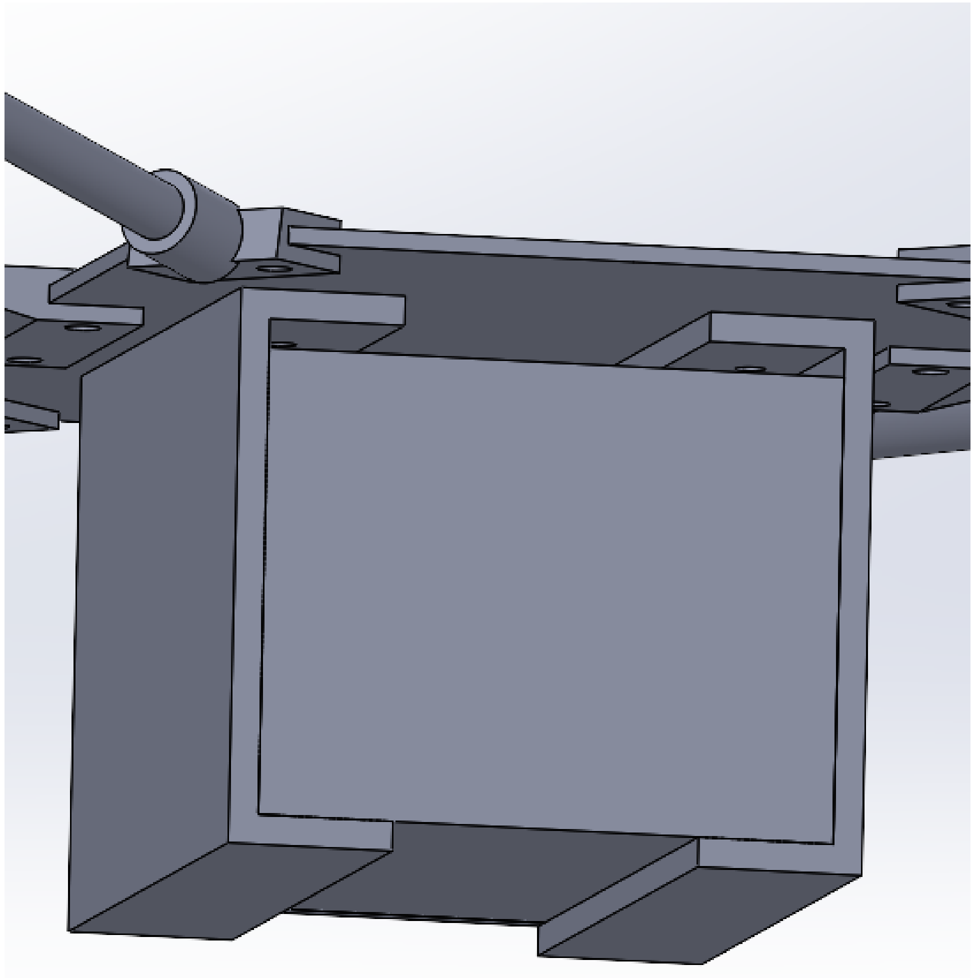


Figure 28

This is the current version of the containment unit. This containment unit is based on a cabinet. One side of the cabinet is open and the other side is closed off. This is almost as if it was a bookshelf that needs a door or a latch. The side that is open will be the front of the drone because the drone will be angled downwards but the velocity will keep the load within the drone as long as there are no external forces that affect the direction of the drone. This will also prevent the drone from putting extraneous loads onto the latch or mechanism that will prevent the load from falling. We will be adding some way to secure the load easily with either a latch, lever, or some other mechanism that will be easy to release but will also have the structural integrity to prevent the load from leaving the containment unit. In the event the drone makes a sudden change in direction that causes the unit to slide up against the release mechanism.

The final version of the containment unit was redesigned to sit on top of the drone and to be easily accessible. The new design was inspired by a bottle of pills. Since Chapie is able to keep the temperature of the payload consistent. The drone no longer needed to store a temperature controlled environment. So the model was downsized to fit 2 of the Chapies.

Figure: Payload container

Drone Lock Mechanism:

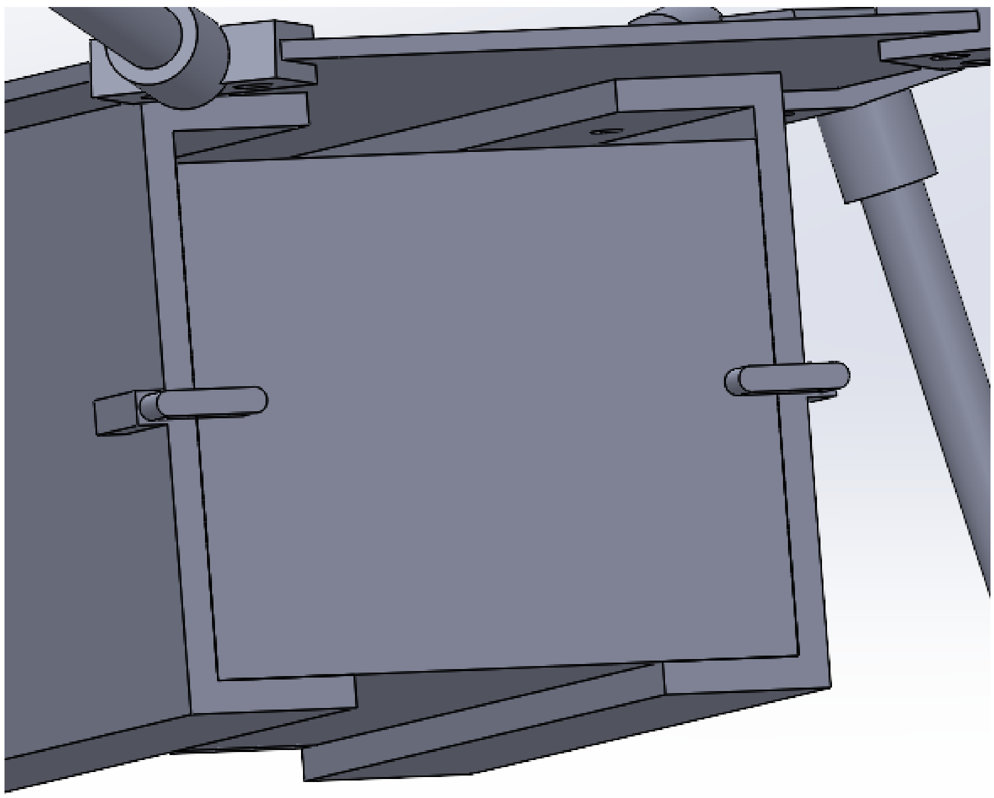
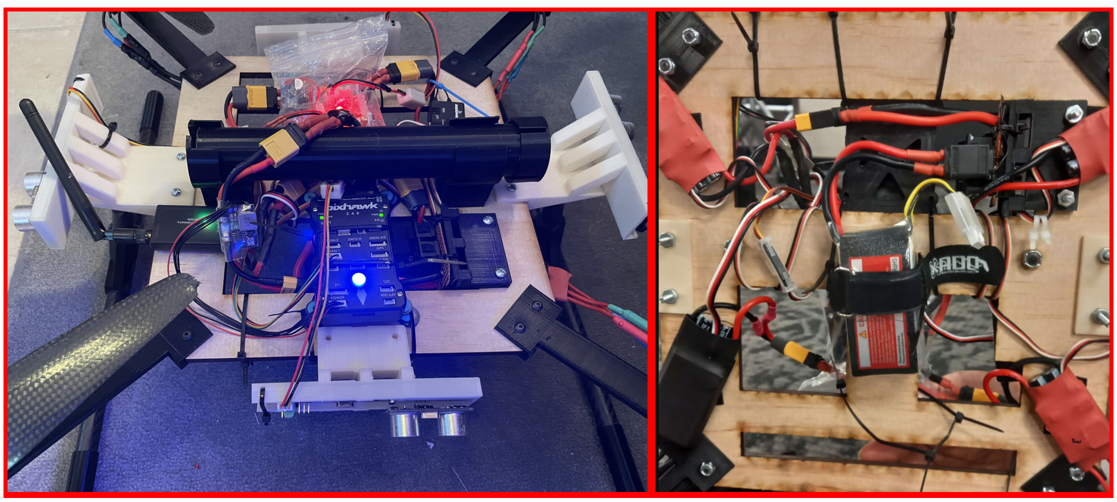


Figure 29

This is one of the lock mechanisms that we are testing. It is as simple as it looks. The goal of the lock mechanism is to make it easy for someone to understand at first sight and to also ensure the drone doesn't come out of the containment unit. The lock mechanism is held in with friction. Inside the rectangular portion that was added to the containment unit to support the lock mechanism. The idea behind this comes from a simple robotics project one of the members did in highschool. They would use a simple lever to prevent a phone from coming out of the robot while in the middle of an intense match. The purpose of the lever was to keep the phone stationary but also give the student the ability to release the lever quickly and take the phone out. We have noticed that this design is a great starting point but can't be the final version. The final version will take into account a simple but also a stronger method of locking the unit into account for various sized units and limit how much each unit could move within the container. This is also to ensure there are no loose items on the drone that could make it lose its balance and lead to problems.

Electronic Mounts
Additional mounts were required in order to include some of the main electronic components, including the radio receiver, flight controller, and a power switch. Compartments to hold these devices have been designed around the final baseplate.

Battery compartment was initially planned to be a standalone piece that would be attached to the bottom of the baseplate. By going through design iterations and finding ways to optimize weight and size, we developed a velcro solution that also keeps the battery attached and easily accessible.



4.3) Analysis

4.3.1) ANSYS vs. SolidWorks

In order to determine the efficacy of the design prior to committing to it wholefully by building in the real world, we must first analyze the predicted outcome. This can be done by performing FEA to determine the structural integrity of the design of the drone. This includes when the drone is at rest, during peak acceleration, and while in flight through air. Possible expansions to the analysis include determining the drone's resistance to damage in the event of collision. There are three types of analysis we must perform: structural analysis, fluid flow analysis, and a simulation of flight in the expected environment.

To perform the structural and fluid analysis, we will export the designed CAD model into the analysis software. Although Solidworks, which we are creating our CAD in, has a FEA tool, it has been suggested by advisors to avoid using it due to the difficulty of producing accurate results. As a result, we will be using ANSYS Discovery Student to perform our FEA. ANSYS has the ability to perform structural and fluid analysis. It provides the flexibility to alter the conditions of the analysis to achieve very precise predictive results in the desired projected real world environment. As such, ANSYS will prove very useful in determining how our drone will perform in our expected conditions of operation. Furthermore, ANSYS has a free student license, and comprehensive introductory material to ensure that the user is able to acclimate to the software and unlock its full potential.

4.3.2) Computational Fluid Dynamics (CFD)

The below image shows a sample analysis captured from the tutorial, visualizing the flow of water through a pipe. Generated through the application of (CFD) At the top, we have the option to change the type of analysis we perform, including structural, fluid, and thermal. On the left, we are given the flexibility to specify the material of the object, the type of fluid it interacts with, the temperature, and more.

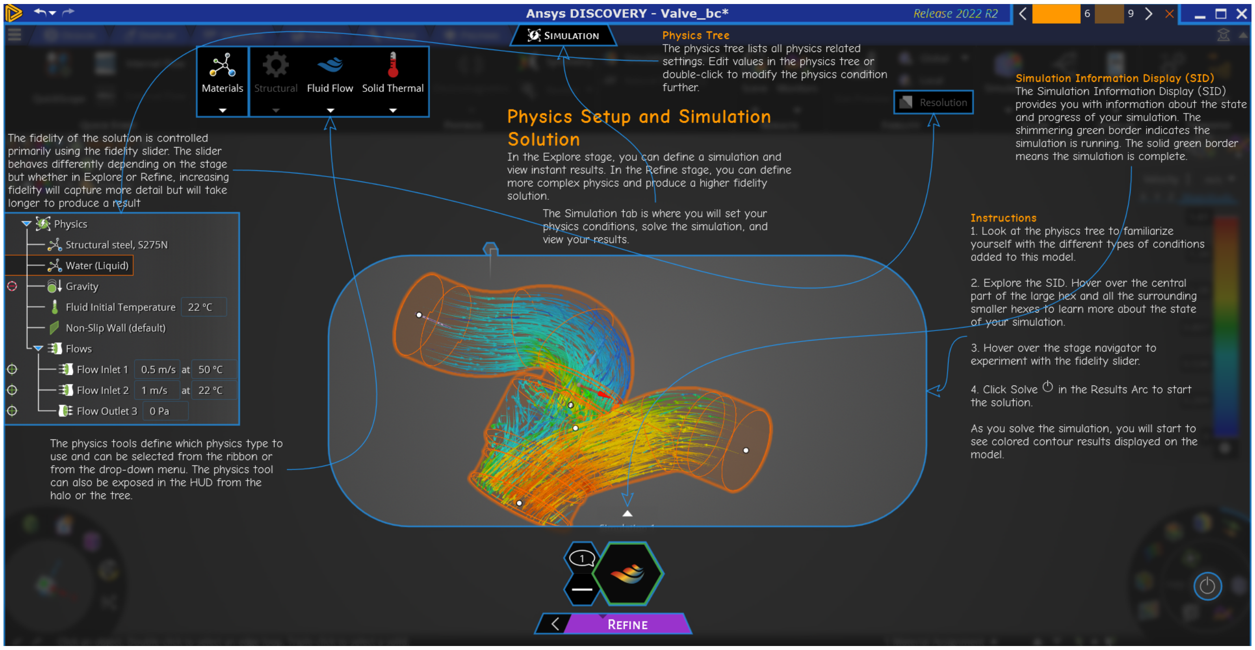


Figure 29

4.3.3) Structural Analysis

Structural Analysis is a form of testing the design by performing calculations on the structure, and is often a prerequisite to physical testing of a product. Its purpose is to study the behavior of the drone components under the action of forces. Below is a sample simply supported beam to illustrate the mesh generation in Ansys, and deformation heat map.

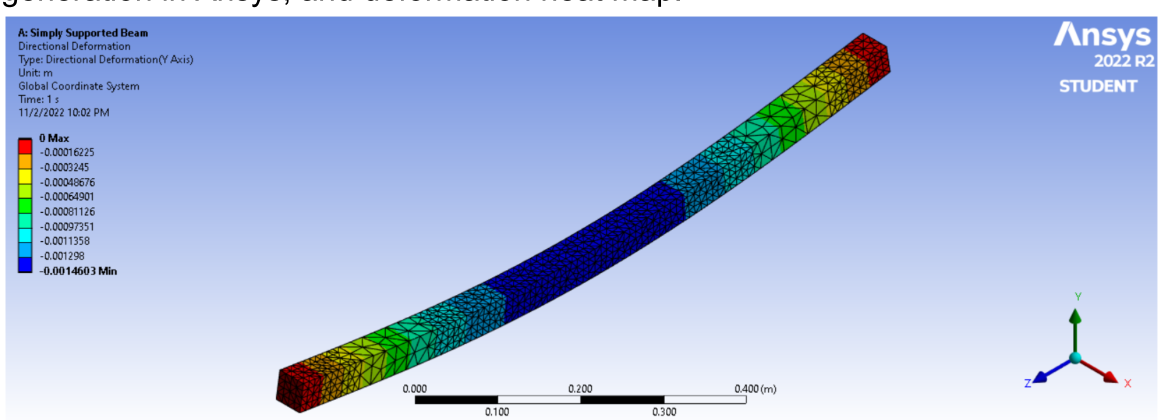


Figure 30

Our drone must be able to withstand the loads it will experience during takeoff, flight, and landing. Structural analysis will allow us to predict the drone's success by computing stress, strain, support reactions, accelerations, and stability throughout the drone structure.

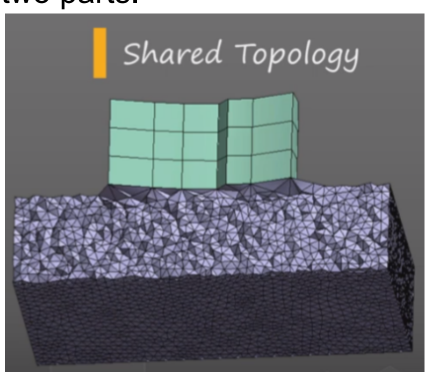
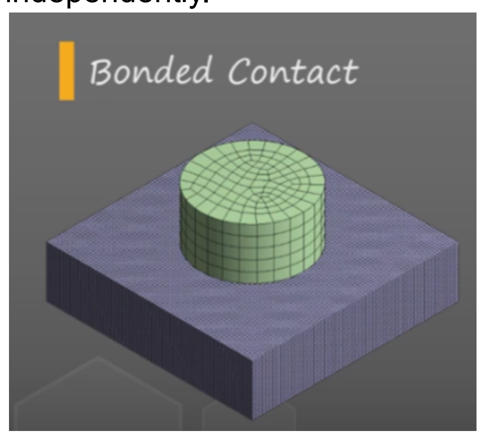
The primary loads we are concerned with are the loads applied by the propellers. Using Ansys, we will analyze how the application of forces normal to the propeller affect the structural deformation of the drone and movement. The key characteristics affecting our results include the material, geometry, and force values applied on the drone. The material and geometry will be as uploaded from the CAD model, the force values can be adjusted so that we can run the analysis in various different conditions. One condition may be applying an equal in magnitude normal force on all 4 propellers, another would be varying the magnitude of the force throughout the propellers, as we would with the motor power distribution system in the case of turning, moving, both of which can also occur while rising or decreasing in elevation.

Ansys has tools that allow us to visualize the results as a heat map along the contour of the drone. It allows us to see exact pressure values at certain locations as well. Knowing the material type we selected, we can also determine if the material at that location will experience one of the following: fracture, elastic deformation, or inelastic deformation. Furthermore, Ansys has a tool for visualizing the deformation of the structure under the provided loads as an animation.

Having performed this analysis in Ansys, it will be easy to predict if the physical prototype will perform well. This will minimize the amount of investment we put into an unsuccessful design, and will allow us to revise the material and geometry of our design. Moreover, it will provide us some insight on the limitations of the load that we can successfully deliver, as well as the maximum power we can afford to allocate to the motors in performing maneuvers such as takeoff, landing, horizontal flight, and rotation.

Ansys FTE is performed using a meshing algorithm that divides each face of a part into some triangulation. The analysis is then performed on each piece of the triangulation. Smaller faces, or key joints, are meshed by using many but small triangles. Larger faces may be triangulated using fewer but larger triangles. This algorithm allows for high accuracy at the expense of higher object count and therefore simulation runtime in small but key sections, while also maintaining accuracy for larger sections.

One major challenge using Ansys is that when importing an assembly, the interaction between any two components, even if they are immediately adjacent, is not assumed by the program; therefore, their interactions are defined incorrectly. The interaction mechanism is something that the user must specify. In assemblies that have hundreds of pieces, the specification protocol may take very long, and be very tedious. However, Ansys does provide some tools to make the process easier. Below are two of the integrated solutions Ansys provides that can set the rules for automatic mesh generation and interaction of parts.

	Shared Topology	Bonded Contact
Description	Shared topology sets the interaction between adjacent components by setting a continuous mesh along the two parts. 	Contact setting that bonds adjacent interfaces, establishing a rule that meshes them independently. 
Pros	• Continuous degree of freedom, accurate to how we expect the drone to behave.	• Independent meshes give the possibility of performing a more rigorous analysis on key components, and less rigorous on the parts that are not of interest. • Can specify distance tolerance to be considered “in contact.” • Has contact types that

		allow for separation, friction, or frictionless behavior.
Cons	• Does not allow sliding, or separation • High complexity may make high quality meshes difficult to produce and solve, especially at connections between fine and coarse components.	• May produce inaccurate results in regions immediately around the contact.

Table 12

4.4) Simulation

In SD2, the Medidrone team opted to not spend time on performing computer aided dynamics analysis before attempting to fly the prototype, as the prototype was durable, and could be tested in safe conditions in the real world.

4.4.1) ROS vs. PX4 autopilot vs. Webots

In addition to FTE and CFD, one of the best ways to ensure that the model we design performs up to the standards we set is by testing the complete drone in a simulated environment, including its movement, sensors, and logic. In order to be able to do so, we must program the drone using software that is compatible with robot simulation platforms. We researched several such software:

4.4.1.1) Robot Operating System (ROS)

ROS is a widely used set of software frameworks where the user can select the tools they need for their specific projects. Has been used for many kinds of robots including cars, drones, and more. Below is a sample image from a drone programming tutorial on ROS. Objects such as the rotors are easily defined and used in code.

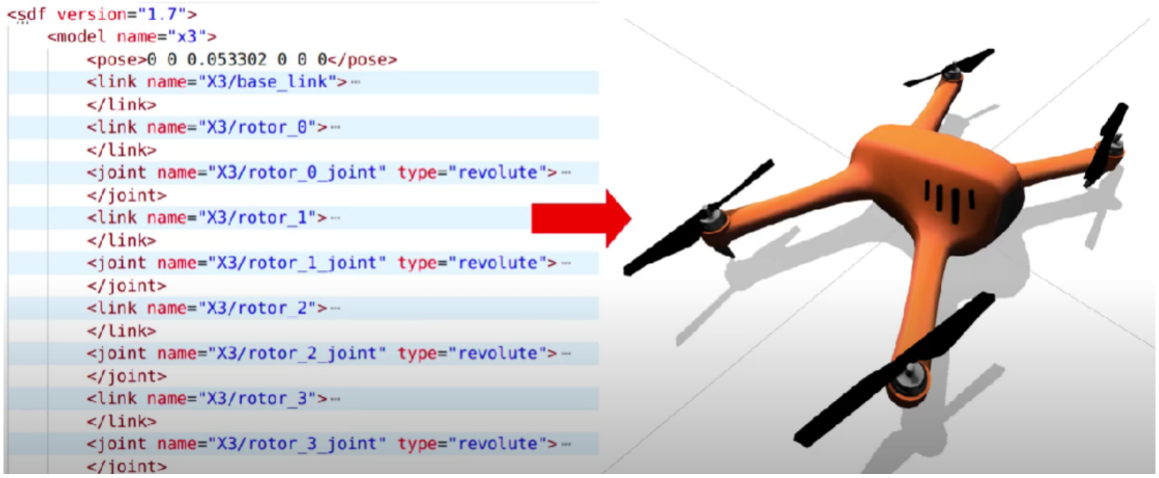


Figure 31

4.4.1.2) PX4 autopilot

PX4 autopilot is a tool for the development of autonomous flying units. It has capability of programming drones and simulating their behavior in simulated real world environments using Google Earth.

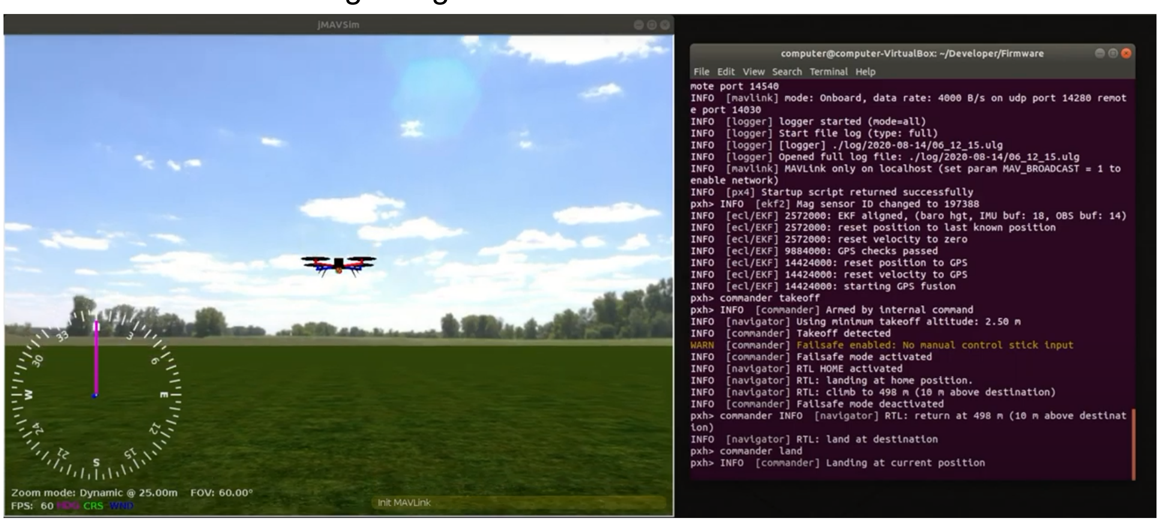


Figure 32

4.4.1.3) Webots

Webots is an open source robot simulation platform that allows modeling, programming, and simulating the robot. It is widely used in industry, research, and education. Below is a sample of programming a drone in Webots, set in a virtual environment.

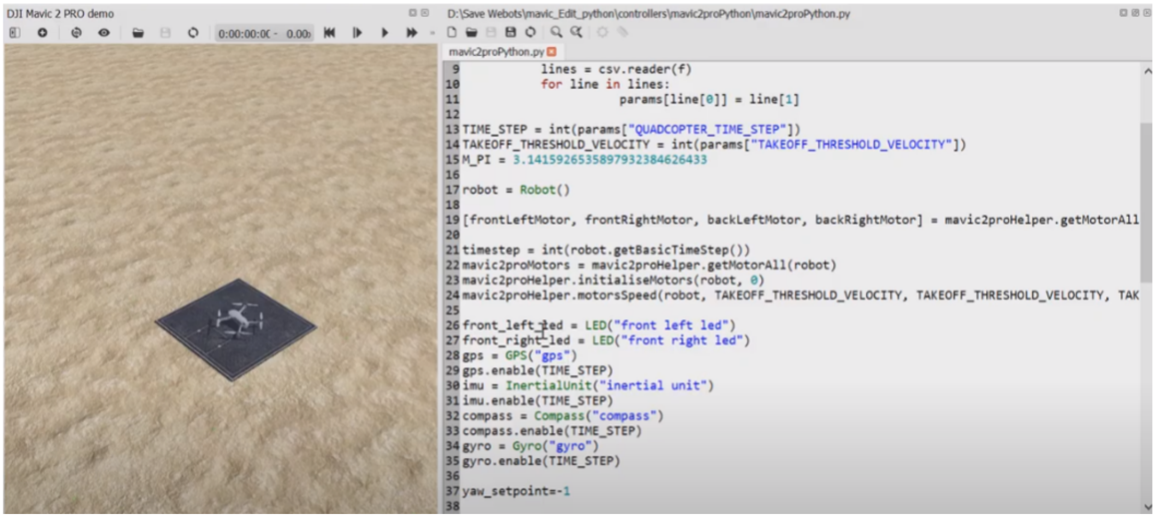


Figure 33

4.4.1.4) Pros & Cons

There are advantages and disadvantages among all of them as outlined below:

	ROS	PX4 autopilot	Webots
Pros	• Interacts well with simulation platforms such as Gazebo • Open source • Well tested • Compatible with a multitude of sensors, communication systems, and hardware components.	• Interacts well with simulation platforms such as Gazebo • Hardware and software are open sourced • Dedicated software	• Works with importing existing CAD models. • Can do the full simulation in Webots including flying in air. • Can generate indoor and outdoor environments • Open source • Supports a

		for drones ● Contains aerial maps generated with Google Earth.	multitude of programming languages ● Runs on a reliable physics engine (ODE) ● Supports multitude of sensors
Cons	● Not well supported for operating systems other than linux	● Relies heavily on other libraries.	● Complex programming required to deliver logic

Table 13

We will likely proceed with ROS due to its extensive usage, tutorials and resources available, as well as its easier-to-use nature. In this case, we need to incorporate a platform to complete the simulation testing phase. There are various platforms that are compatible with ROS and some of the other aforementioned programming software, and will allow us to perform testing of the complete drone. The two main ones we considered are Project Airsim, and GazeboSim.

4.4.2) Project Airsim vs. GazeboSim

4.4.2.1) Project Airsim

Project Airsim is a platform made by Microsoft for training and testing drones. The software supports integration of various drone sensor equipment, artificial intelligence models, real world terrain and features.

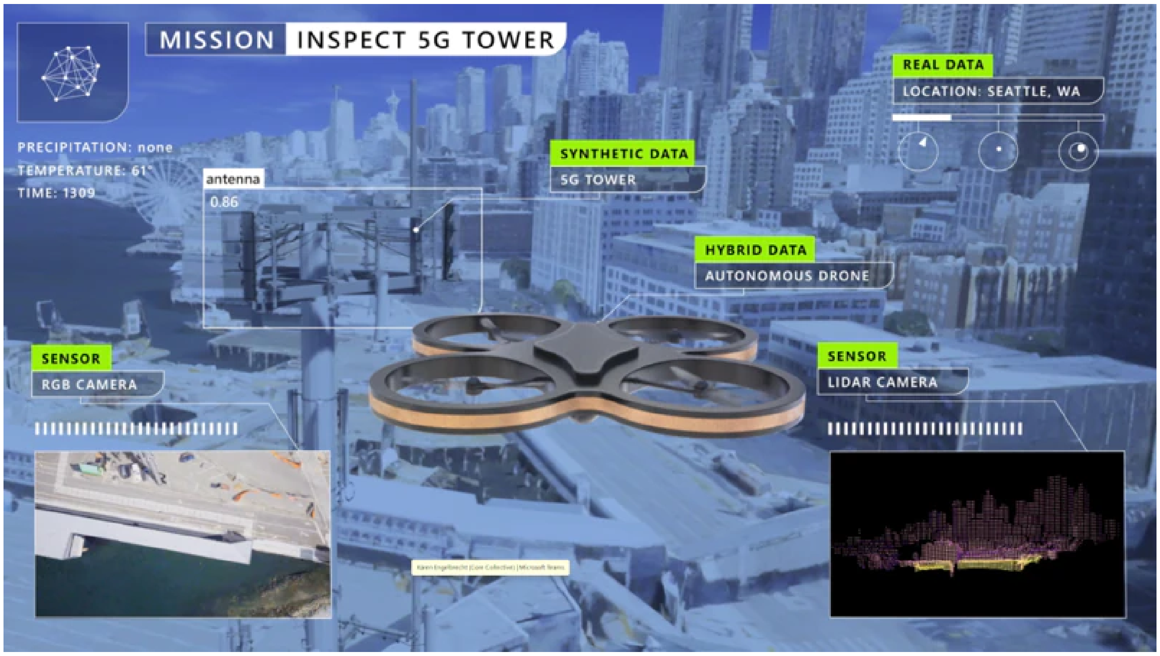


Figure 34

4.4.2.2) GazeboSim

GazeboSim is an open source robotics simulator. It provides the possibility of rendering an environment including various features such as different lighting, 3d terrain, and textures. Gazebo runs on the Open Dynamic Engine (ODE) physics engine. Gazebo can work with the Robot Operating System (ROS), which we plan to use to send commands for our drone. All these features make Gazeobism a strong testing ground for the drone’s maneuverability and sensors.

Below is an example of a drone that is trained to detect the black and white mat as a landing pad. It is being tested in Gazebosim.

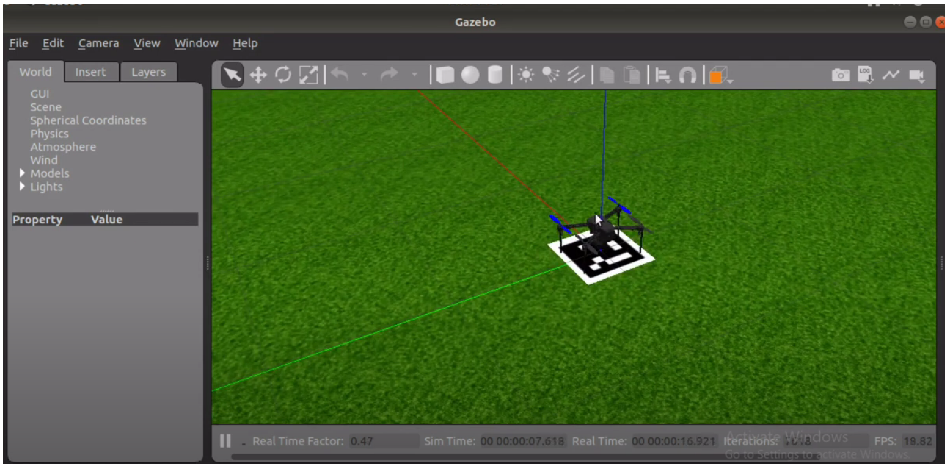


Figure 35

4.4.2.3) Pros & Cons

	Project Airsim	GazeboSim
Pros	• Contains pre-trained AI models • Virtual real world testing environment • Collision detection testing • Interaction with other flying objects during the test	• Allows testing of movement, and sensors • Uses reputable physics engine and environment rendering. • Instant availability for free
Cons	• Limited preview available • Not a lot of information from other sources on the program	• Virtual environments may not be 100% representative of a practical environment.

Table 14

Project Airsim offers a favorable testing environment for the drone. Microsoft is delivering a very robust and approachable program for drone and AI testing. However, their program's official release is sometime in the following year. For the time being, they have a limited preview available for select customers, which we have requested access to.

Most likely, we will proceed with Gazebosim. It is a robotics simulation tool that has been extensively used including in some national competitions such as the NASA Space Robotics Challenge. Gazebosim provides us with the tools we need to make sure that the sensors and controls of the drone are successful.

4.5) Virtual Testing

Moving forward, we have decided to use ANSYS for any FEA or CFD. There is a little bit of a learning curve to learning ANSYS; as such, prior to performing any virtual testing that could potentially be wrong and lead us astray in reviewing the design, we pulled a drone design from GrabCAD to learn the ropes of testing in ANSYS by performing it.

By following material from tutorials on youtube and an FEA course on edx.org, we were able to learn how to use the ANSYS workbench. The workbench is the project management platform that communicates model data across various types of ANSYS software including:

- ANSYS design modeler, where the geometry can be verified and edited
- ANSYS Mechanical, where static structural analysis can be performed
- ANSYS Fluent, where fluid analysis can be performed

All of these tools can communicate smoothly with each other using the ANSYS workbench to allow for an accurate transmission of data.

Furthermore, ANSYS workbench provides easy access to the Engineering Data notebook as shown in the image below, where mathematical variables can be specified as well as the material of the model components.

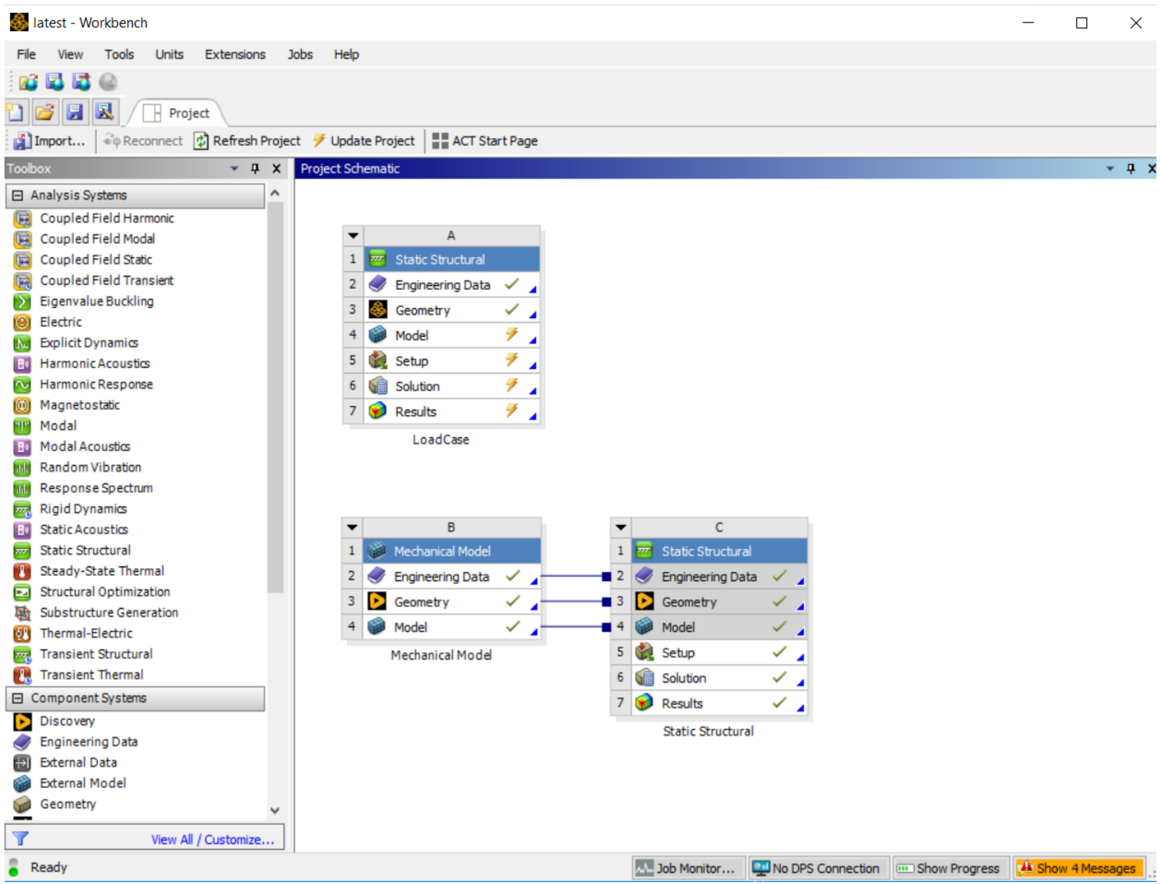


Figure 36

To be able to perform static structural and dynamic testing, one must become familiar with how the mathematical modeling of the physical problem is specified in ANSYS mechanical. We utilized the GrabCAD model to explore things such as importing geometry and developing a mesh. Following that, we were ready to begin understanding static structural testing.

4.5.1) Static Structural Testing

4.5.1.1) Sample Drone

To perform static structural testing, ANSYS mechanical requires for the system to be constrained, meaning it must have some supports with respect to which the solution will be computed. Furthermore, there must be at least one force which generates the stress throughout the model. As shown in the image below, we first tested by applying fixed supports on the landing gear of the drone, and an exaggerated vertical force on the frame of one of the propellers. The force is shown as a red arrow pointing up, the fixed supports are highlighted in blue.

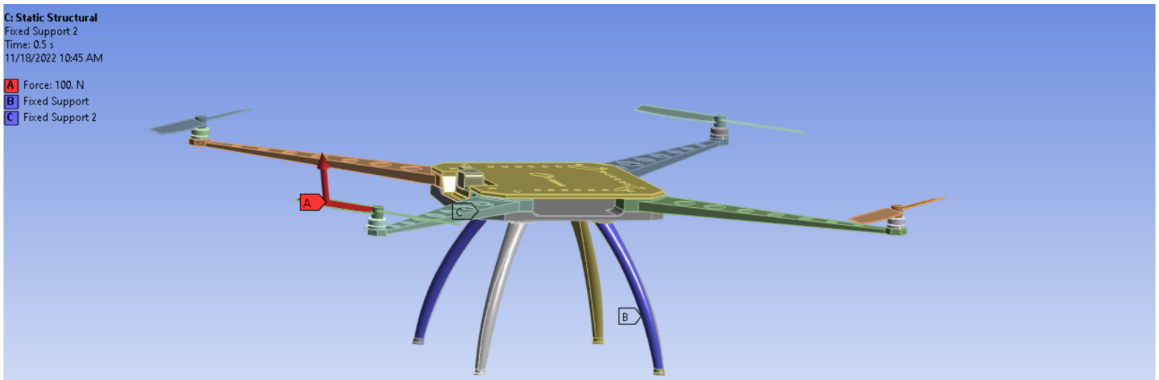


Figure 37

To gain insight towards the impact of applying the supports and load in such a manner, we specify for the program to compute an equivalent (von-Mises) stress solution, and a directional deformation solution.

Stress:

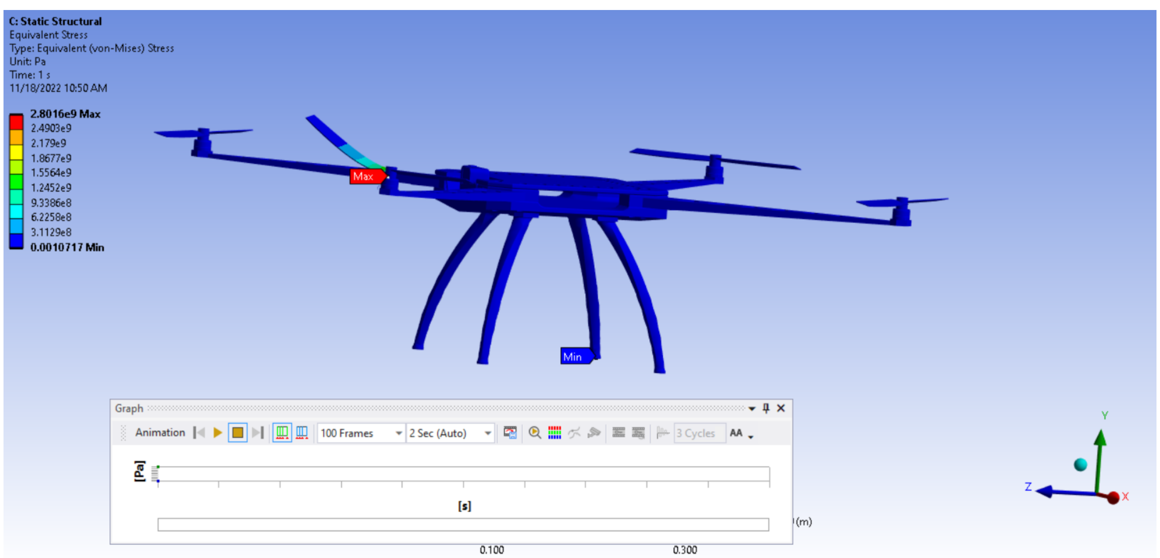


Figure 38

Directional Deformation:

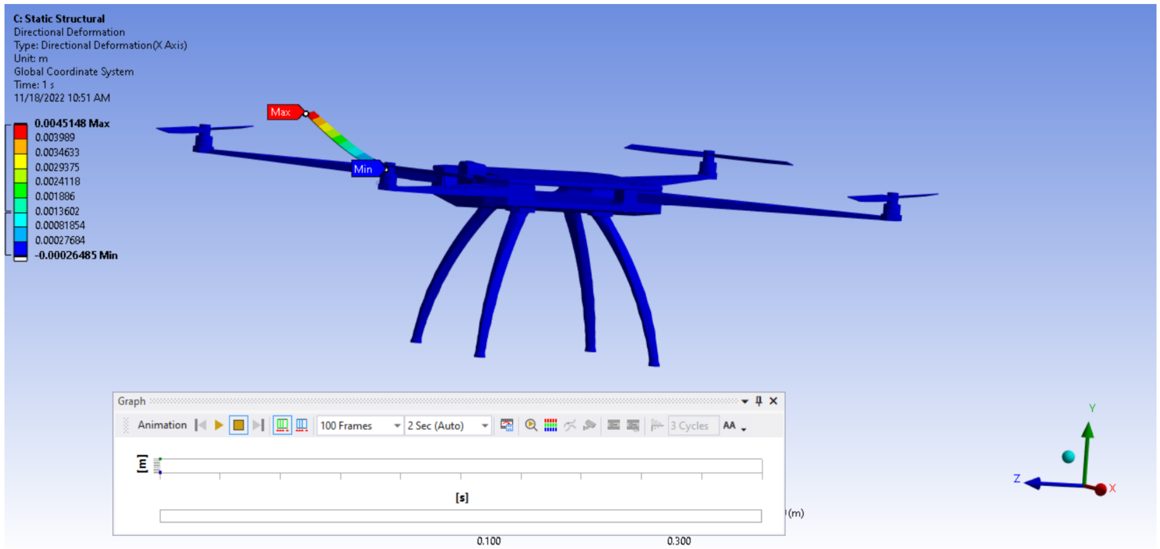


Figure 39

We also learned that ANSYS has a tool for visualizing the solution overtime. The line graph shown at the bottom of the images can be used to display the results of the structural test at any point in time up to the specified limit. This can be helpful for learning whether the damage from some application of load occurs over a long period of time, or if it is sudden. This will allow us to know if there is damage that we can react to before it causes fracture or inelastic deformation.

4.5.1.2) MediDrone

We are now ready to perform analysis on the first iteration of our drone design. This design includes the base, arms, brackets, and motor mounts, as well as the landing gear. We will be performing static structural, dynamic, and fluid analysis on our drone.

The first type of analysis performed will be the static structural analysis, following the procedure learned in the previous example. This time, we specify four vertical loads applied to the motor mounts, and two fixed supports in the landing gear as shown in the image below. This simulates the lift generated by the propellers and the load supplied by the delivered package.

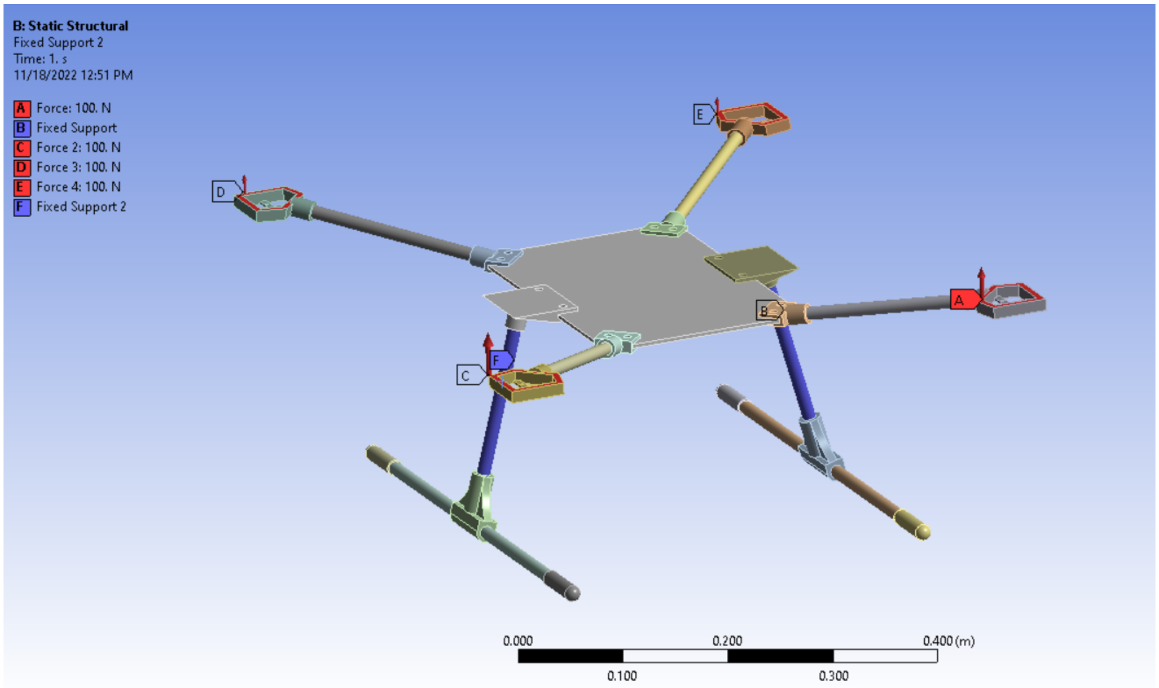


Figure 39

This will simulate the drone's response to generating lift through the propellers while being held down in the legs either by a person, or by a weight that surpasses the generated lift. In solving this solution, the mesh needs to be generated upon the drone. The image below showcases the generated mesh

.

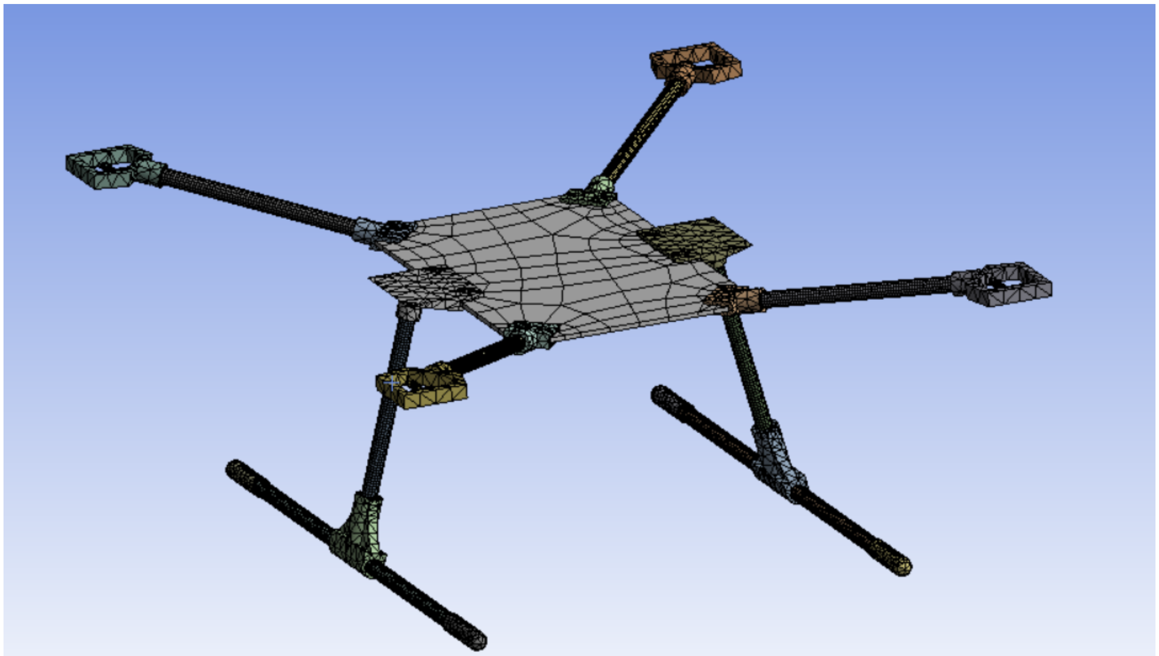


Figure 40

Upon running the solution, ANSYS throws an error as the number of nodes in the mesh surpasses the number of nodes allowed with the student licenses (32,000)

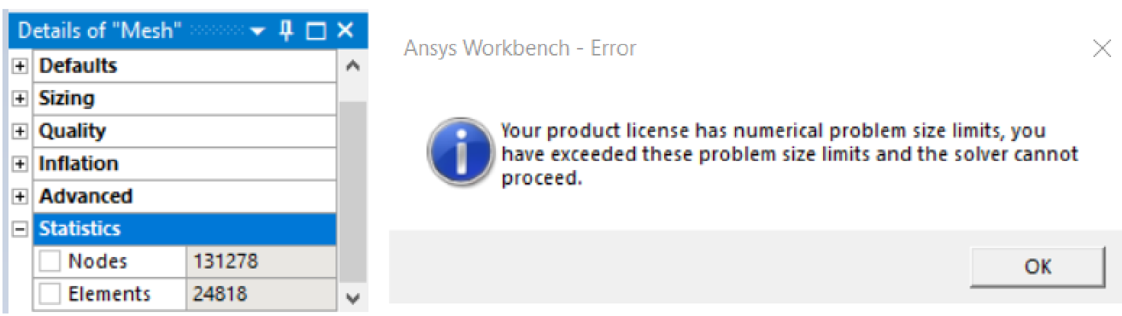


Figure 41

To reduce the number of nodes in the mesh, we open the geometry of our drone in ANSYS Discovery and make optimizations that correct the boundaries between faces, fix gaps between curves, and remove small faces and curves. This reduced the nodes in the mesh such that we can now run the solution.

Pivoting back and running the static structural simulation produces the following deformation.

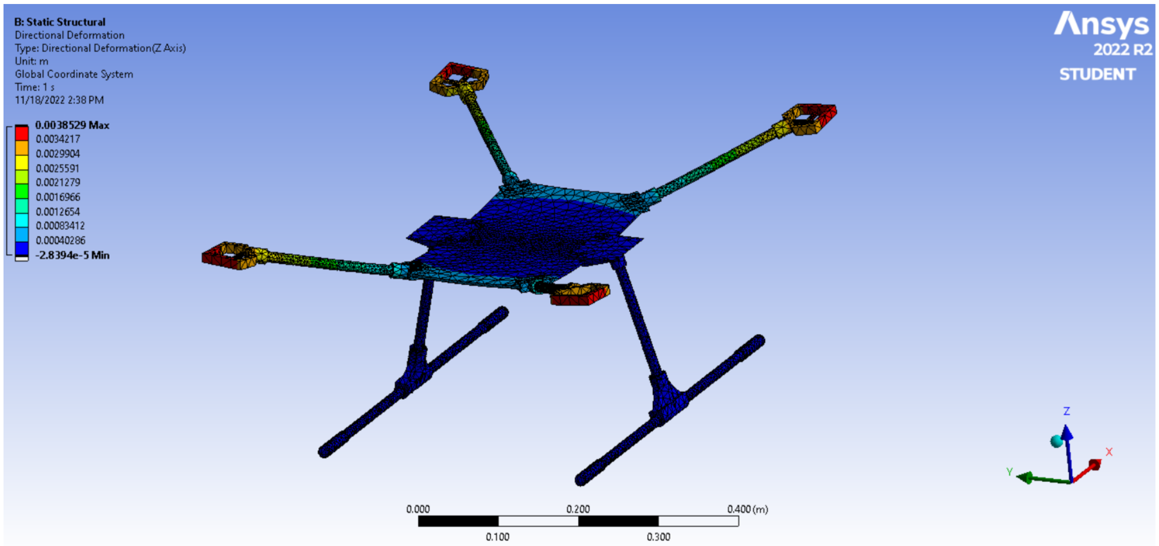


Figure 42

As is shown in the image, the highest deformation experienced is in the motor mounts. This is the expected behavior as the beams separation from the frame allows for a large moment arm where the load is applied, therefore bending the mounts upwards.

4.5.2) Fluid Analysis

4.5.2.1) Mesh Generation

The next type of analysis we perform is fluid analysis. Fluid analysis through computational fluid dynamics will allow us to visualize the air current as it travels around the contour of the drone. This can be done by instead of simulating the movement of the drone through air, placing the drone through a wind tunnel where the direction and speed of the wind is simulated to approximate the real world situation.

This analysis will be performed through ANSYS Fluent. A part of the ANSYS student package that is available for free. It was decided to use ANSYS Fluent instead of ANSYS Discovery due to Discovery's greater technical specification requirements. Furthermore, ANSYS Fluent easily connects with the design geometry and allows for other potential connections through ANSYS Workbench.

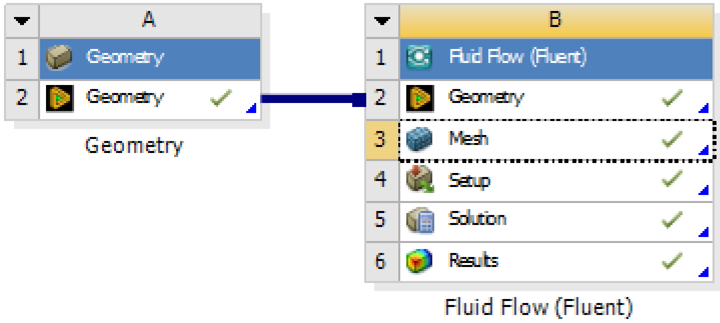


Figure 43

We begin the fluid analysis process by creating an enclosure around the drone geometry, as shown in the below figure.

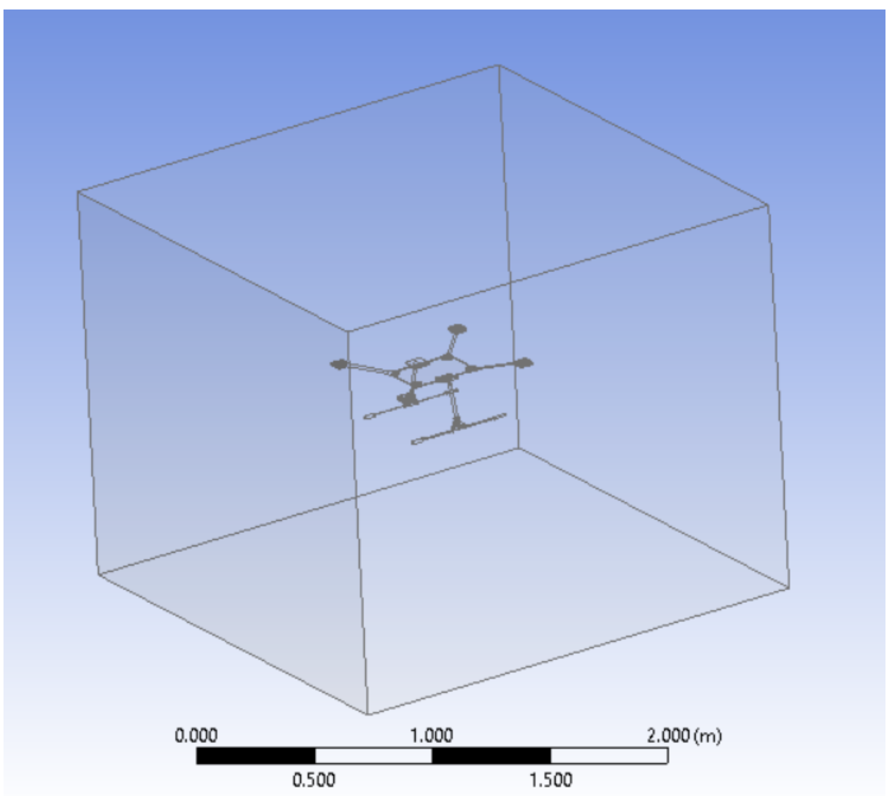


Figure 44

The enclosure is a fluid air type and serves as the wind tunnel. The next step taken is to subtract the drone from the enclosure. This is done through a boolean object that negates the drone geometry from the enclosure.

In fluid dynamics, it is typical to only solve for the fluids. As such, we do not need to keep the solid drone object, but rather simply the air that is the fluid domain. This allows for a simplification of the generated mesh, as the mesh only needs to be generated for one contiguous part that is that enclosure.

Upon generating the mesh, we once again run into an error of having more nodes/elements than are allowed for the student version. To get around this, more optimizations regarding the mesh need to be performed. One optimization that can be done is using the symmetric nature of the drone's design. The solution needs only be computed for one half of the drone, and that can be extrapolated to the other half.

The first step to doing this is identifying the plane by which the design is symmetric. In this case, we use the Y-Z plane. Then, we use the symmetry tool in Design Modeler to "cut" the drone in half as is shown in the figure below.

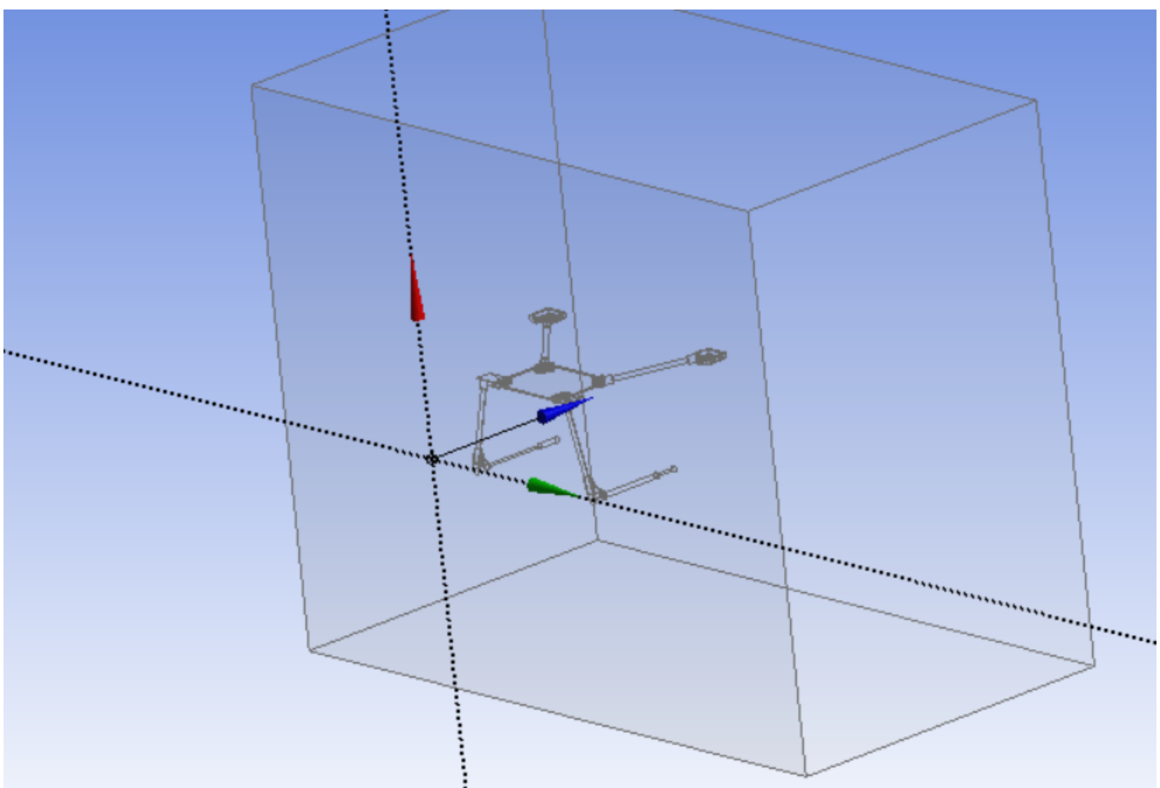


Figure 45

Following the above modification, we are able to fully mesh the geometry. As shown in the figure below, the enclosure itself does not require too much detailed meshing as it is a large and contiguous simple shape; however, the drone contains small pieces that make it rather complex to mesh. This is a testament to the importance of simplifying the problem when meshing, and finding opportunities to do so by defining symmetries. We can now proceed with the setup of the fluid solution.

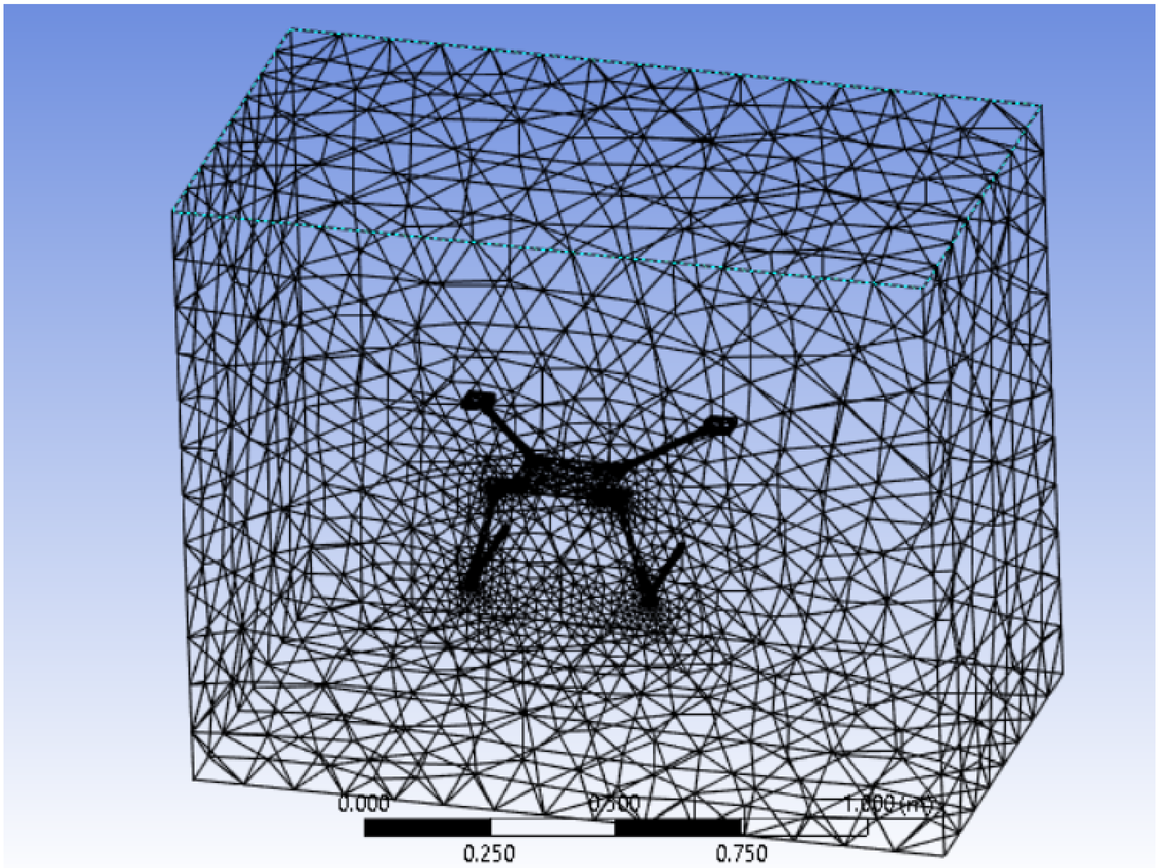


Figure 46

4.5.2.2) Solution Setup

Default specifications are used in setup excluding the following:

Variable	Value

Viscous Model	Turbulent K-epsilon (2 eqn.)
Fluid Type	Air
Inlet Velocity	3m/s
Outlet Type	Pressure outlet
Method	Coupled. Second order upwind turbulent kinetic energy. Second order upwind turbulent dissipation rate.
Runtime parameters	Number of iterations: 100 Convergence criteria: disabled

Table 15

4.5.2.3) Visualization and Discussion

Upon running the solution with the provided mesh and parameters, the following residual chart is produced at each iteration

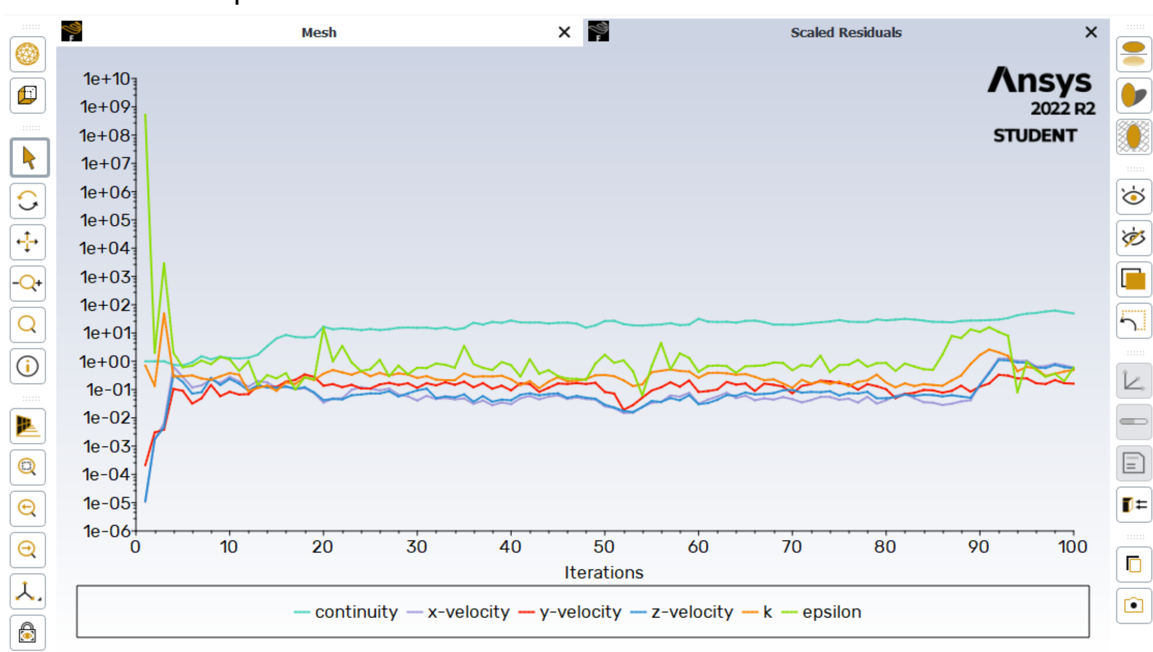


Figure 47

The residuals develop in an unpredictable pattern as the solution progresses through iterations. It is likely that the program is diverging in solving this fluid dynamics problem as was provided, and the flow visualization will be unrepresentative of the real world. Below is a visualization of the streamlines as the flow travels from inlet to outlet, encountering the drone.

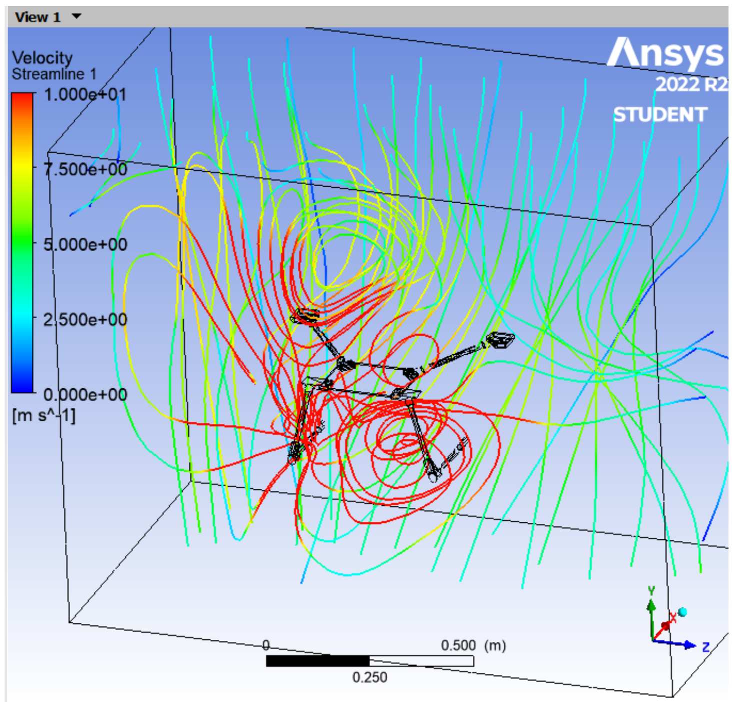


Figure 48

The viscuous model k-epsilon appears to be weak in solving this specified problem. Following a change from k-epsilon to k-omega the solution appears to converge much more readily.

Below is a figure to depict the new residuals per iteration. Closing in on residuals of 1e-03 shows that the model is able to converge on a fluid solution.

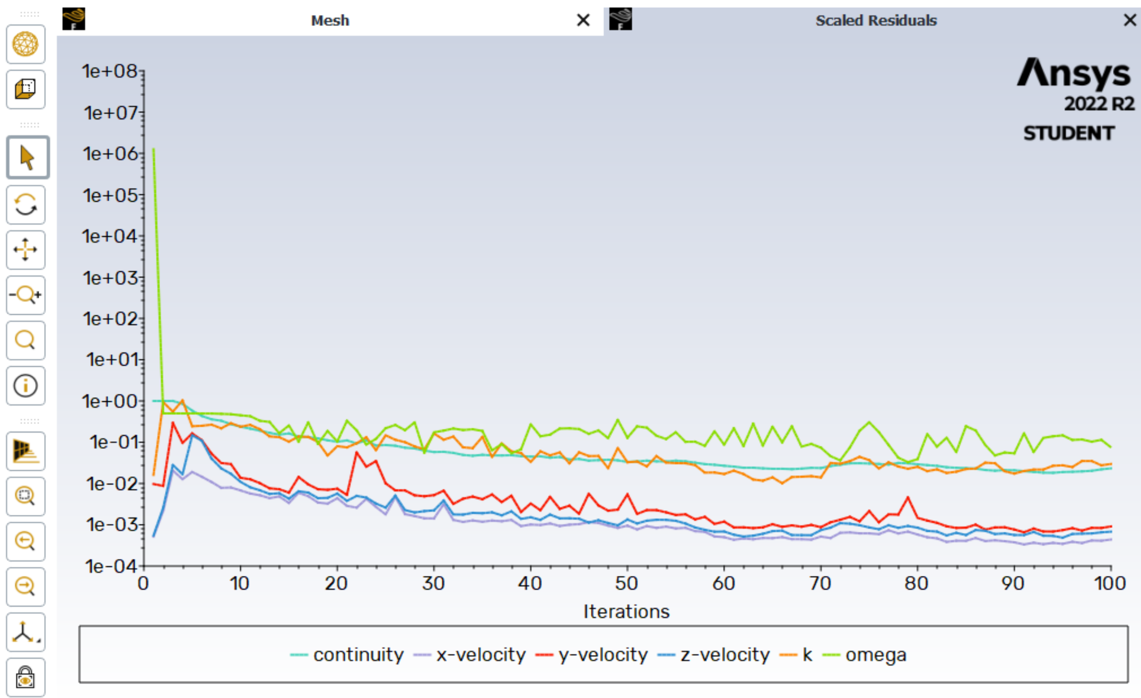


Figure 49

Below is a figure to show the new calculated streamlines within the enclosure as the air flows around the mesh. The airflow inlet is the topmost face of the bounding box, the outlet is the bottommost face.

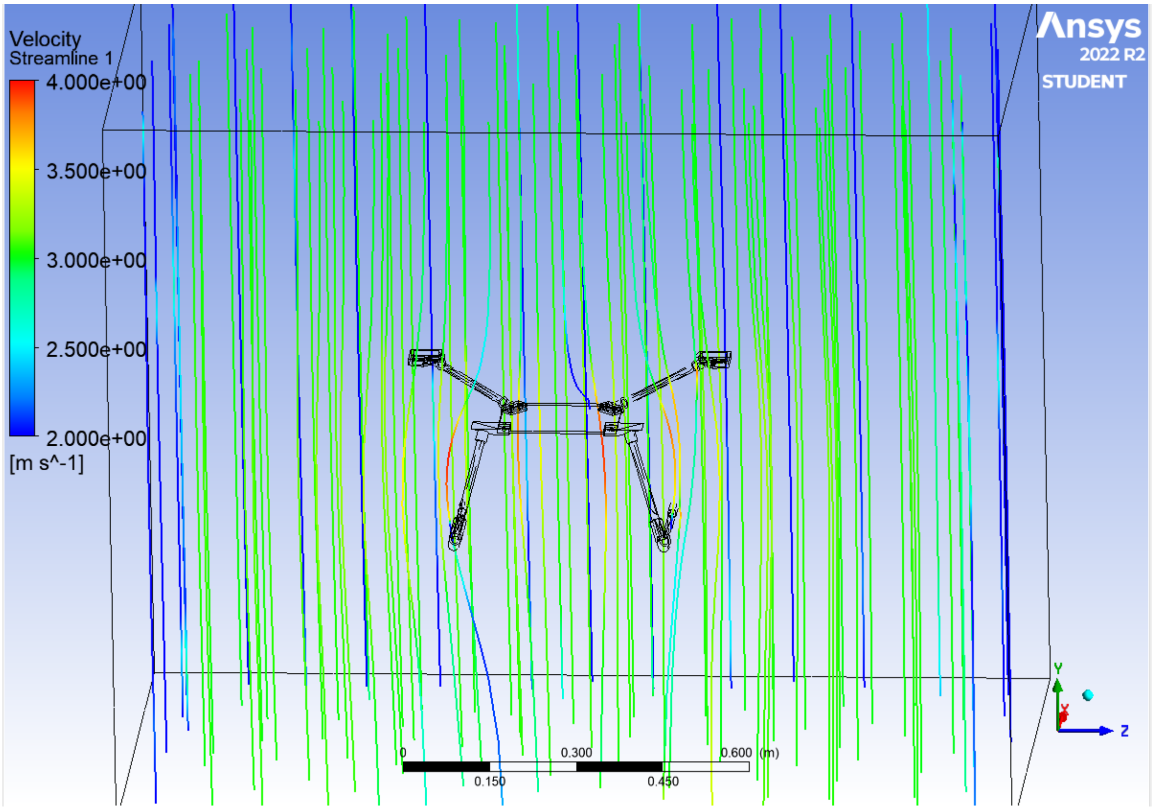


Figure 50

The streamlines allow us to observe the behavior of air as it flows around the drone while the drone is flying. In this simulation, we simulate the flow of air at 3 m/s downwards towards the drone. That is as the drone is gaining elevation at 3m/s while downward wind speed is zero. The consequence of this change in elevation is shown by the streamlines, as certain locations around the drone experience faster or slower air speed. This causes differences in pressure. In the figure below, we illustrate the pressure experienced in different locations around the drone.

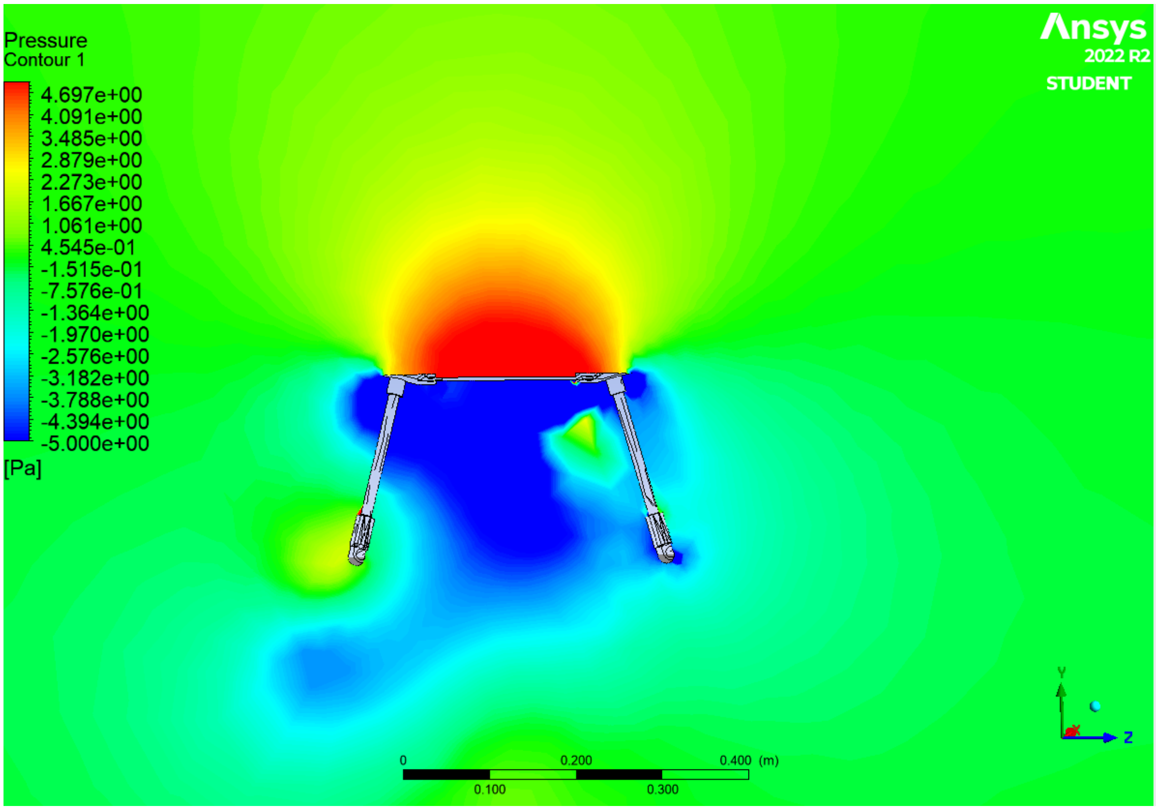


Figure 51

As can be seen, the top of the frame experiences the bulk of the pressure. Due it it's lack of aerodynamic form, the pressure is distributed across a large area and can slow down the drone. Furthermore, as the air travels around the top of the frame downwards it creates a large suction area underneath the frame where the pressure is significantly lower.

4.5.3) Dynamic Analysis

As mentioned above, this was not done in favor of spending more time testing and optimizing the physical build of the drone

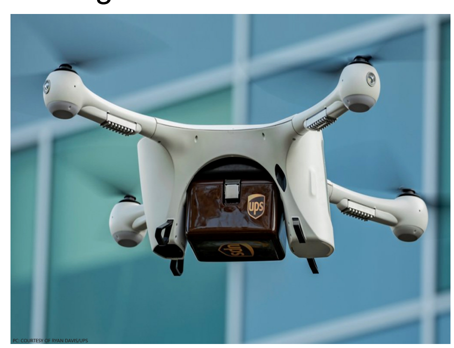
The final step in the analysis is to perform a dynamics analysis. This analysis is important to visualize how the drone will act when it is not constrained. The drone will generally be in an unconstrained state, whether it is ready to take off with a reasonable load, or during flight.

4.5.3.1) Gazebo Installation and Setup

This analysis will be performed using Gazebo in a simulated virtual environment on the complete design. To begin working in Gazebo, it must first be installed. Since we are working through a windows machine, it is easiest to install Gazebo by using a Windows Subsystem for Linux (WSL). After installing WSL, we can now use the terminal to follow nearly the same instructions to install Gazebo as one would through linux. We continue to follow a set of instructions from ROS Noetic that will allow us to download dependencies and a demo to run on Gazebo.

5. Payload Design

5.1) Current Payload Systems

<table>
<tr><th colspan="3">Commercial & Hobby Drone Applications</th></tr>
<tr><th></th><th>Non-Medical</th><th>Medical</th></tr>
<tr><td rowspan="2">Commercial
Many commercial systems do not need to make deliveries, rather utilize a camera or sensors only as a payload as observed in commercial power systems inspection drones and warehouse inventory drones. Agricultural</td><td>Defense:
</td><td rowspan="2">UPS blood delivery drone utilizing Matternet

See Matternet

Blood must be stored warm, whereas payloads within scope of MediDrone are to be kept thermally cold to maintain their medical utility. The Matternet UPS drone serves as a great example of innovation and poses the challenge to work on a drone</td></tr>
<tr><td>Power Systems Inspection:
</td></tr>
</table>

drones may utilize a payload containing pesticides or other liquids, without temperature regulation in scope.	Delivery to consumer:  Internal inventory counting:  Agriculture: 	capable of keeping a payload thermally cool rather than thermally warm. Patents Granted and Filed from Matternet include [Patents Assigned to Matternet, Inc. - Justia Patents Search] **METHODS AND SYSTEMS FOR TRANSPORTATION USING UNMANNED AERIAL VEHICLES** **Publication number:** 20220073204 **GROUND STATION FOR UNMANNED AERIAL VEHICLES** **Publication number:** 20210276735 **Unmanned aerial vehicle** **Patent number:** D776569 & **Ground station** **Patent number:** D938339
Hobby: Hobby drones may utilize a camera as a payload or be able to pickup objects, there is no scope including	Camera as payload 	

keeping objects at a specific temp or items from damage in these systems.	Payload Dropper Drone: 	

Table 16

5.2) 3D Printed Payload

A Pelican type container may be utilized as the Drone's payload, and may be designed or bought to specification. The functional block diagram shows the different options for purchase or design of Pelican type payload container.

Choosing the correct 3D printing filament is paramount for a successful design and functional print. Printed objects must stay durable while maintaining a lightweight composure for the scope of this project. Medidone team has assembled a comparison figure of 3D Printing Filiments.

SD2 Medidrone featured a 3D printed payload containment system mounted atop the wooden base of the drone

Filament	**Pro**	**Con**
PLA **SD2 Used PLA**	Cost Effective, Compatible with mainstream printers. Biodegrabable	Brittle, Mass
ABS	High Strength	Mass Hazardous
PETG	Durable	Too soft for scope Scratch prone Slick when wet

TPE	X	Difficulty/Not reliable in printing process. Slow print times
TPU	X	Difficulty/Not reliable in printing process. Slow print times
TPC	X	Difficulty/Not reliable in printing process. Slow print times
PA	High durability and strength	High temperature print, complex printing methods
PC	Impact resistant strength	High temperature print

Table 17

5.2.1) Pelican Operational Diagram

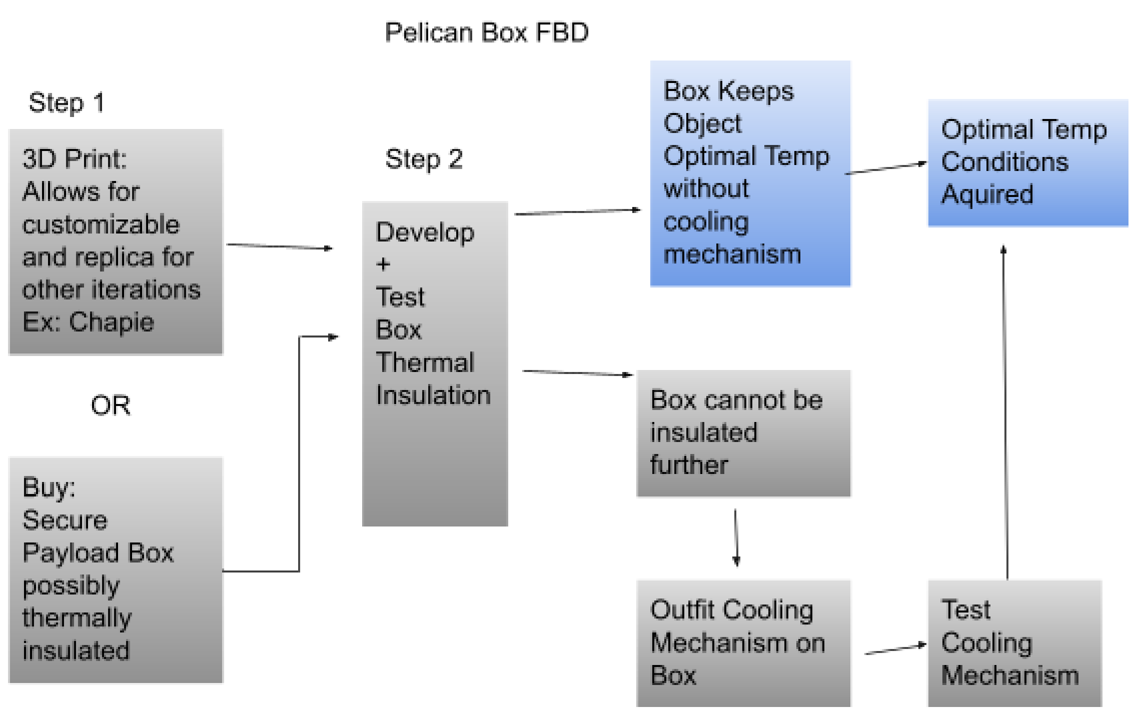


Figure 52

SD2 used Chapie as well as 3D printed an enclosure to contain the chapie and mount to the drone. No active cooling mechanism was needed for Medidrone.

5.2.2)	**3D Print Pelican Type Box:** **SD2 Medidrone 3D printed a cylindrical enclosure, not in the shape of a Pelican type box.**
Pro:	**Con:**
Choose best filament for the job Ex: PLA vs. ABS vs. PETG. See above figure	The "best filament' may be a professional injection mold, materials which cannot be printed using a 3-axis "personal" or "professional" printer
Design at home ● Low Risk if prototypes fail Open Source Payload Protective Box's ● Thingiverse	May not have the quality or strength; professional models may be better
Low Cost for PLA filament	3D print may be unstandardized
Members experienced in .STL design and properly printing projects from experience	3D printer may malfunction due to internal conditions (hard print) or external conditions (power outage)

Table 18

Buy "Pelican" Type Box:	
Pro:	**Con:**
Variable Weight	a purchase is not a renewable resource per each replacement
Strength	Potential conflict designing a mount

	for a private companies product
Will Keep Contents Safe	$$$

Table 19

5.2.4) Development

Prototype in house box, beginning with open source models (ex: sourced from Thingiverse), progressing to in house design of a safe and secure box. Research into *Cooling Mechanisms* and *Thermal Insulation* will allow MediDrone Team to develop either an entirely thermally insulated box capable of keeping the payload at optimal temperature regardless of flight and environmental conditions or will develop a thermally insulated box capable of enabling an electronically driven cooling mechanism utilizing methods such as Peltier cooling *see Peltier in Cooling Mechanisms*.

SD2 Medidrone employed thermal insulation rather than Cooling Mechanisms

5.3) Cooling Mechanism Development

A critical aspect of this project is ensuring the deliverable arrives safely, securely, and serviceable. Part of this process is the ensurement of the medical deliverable being preserved from the point of shipping to destination. With temperature monitoring and control, we can ensure that the deliverable is preserved by the time it reaches its destination. In order for us to figure out how to ensure temperature control whilst the package is towed onto the drone, we have to understand how medical supplies, such as vials and prefilled syringes, are typically packaged.

SD2 Medidrone employed thermal insulation rather than Cooling Mechanisms

5.3.1) Cooling Systems: Passive & Active

Cooling is defined as the process involving the removal of thermal energy from a system. As thermal energy levels in a system rises, the temperature of that system will also rise. In order to reduce the buildup of thermal energy from a system. A cooling system is a system designed to maintain a temperature threshold in order to preserve the components of the system. They are classified as being passive or as active systems, with the capability of having aspects of both.

Passive Cooling Systems SD2 Utilized

Passive cooling is the process in which cooling of the system is attained through the basic types of heat transfer: conduction, convection, and radiation without the need for energy to be added to the system. This would mean that cooling would be attained without the need for any current from the power supply, thus allowing for the drone to dedicate more current for flight control, communications, etc. Examples of this are: ice in a cooler, air ventilation, and nonconducting fluid submersion.

5.3.2) Passive Cooling Implementation

Icebox Passive

The first method of passive cooling we are going to analyze is an icebox implementation. This method involves the use of ice or refrigerant gel to preserve the deliverable inside the box. By having ice/gel in the box, the deliverable could remain at its threshold temperature by being inside of a chilled enclosure by means of conduction, or contact between the ice/gel and the deliverable. This method is seemingly quite achievable and sustainable, as this method is a common everyday practice that can be applied to anything from vials and medical supplies to beverages in a cooler; however, this method may not exactly be feasible as it would be adding more weight to the payload. Now we'll weigh out the pros and cons of implementing this method:

5.3.2.1) Icebox Cooling SD2 Icebox not employed	
Pros:	**Cons:**
Ice / Ice Pack easily obtainable	Adds weight to payload
Doesn't require power drawn from drone	If melts, package will become wet
Can be reused	Package could become too cold

Table 20

5.3.2.2) Comparison of Icebox Cooling Materials Icebox Not Used SD2

With the option of implementing an icebox in our payload, we also have the option of either using wet ice, dry ice, or a gel pack. Wet ice, being the frozen state of water, has a temperature of 32F (0C) or below. Dry ice, being the frozen state of carbon dioxide, has a temperature of -109F (-78C). A gel pack, or an ice pack, is a refrigerant gel that can be cooled down to a temperature that will keep the deliverable's environment to ideal temperature.

Icebox with Wet Ice https://tinyurl.com/Stock-Img-Wet-Ice-Cooler  Characteristics: Shape of Ice can be determined for payload Weight: 0.919 g/cm3 Temperature: 32F (0C)	Easily Preparable (freeze water) Melts (solid-to-liquid) If fully recollected, can be refrozen (sustainable) Will have loss when exposed to heat Could get deliverable wet
Icebox with Dry Ice	

https://hospitalcmq.com/blog/pfizer-coronavirus-vaccine/  Characteristics: Shape of Ice can be determined for payload Weight: 1.6 g/cm3 Temperature: -109F (-78C)	Order-in (delivered) Cost is generally $1 to $3 per pound Sublimates (solid-to-gas) Will have loss when the package is opened but solid state can still be preserved
Icebox with Ice Pack (Refrigerant Gel) https://www.uline.com/BL_2158/Cold-Packs 	Easily Preparable (chill to ideal temperature) Leak-Proof (so long as pack hasn't been compromised)

<u>Characteristics:</u> Flat Pack Weight: 3oz (85g, 0.1875lbs) Price: $0.27 per pack Temperature: 30F to 72F	Can be re-chilled (sustainable and reusable) Temperature gain but is essentially lossless because of its reusability

Table 21

5.3.2.3) Separation Barrier SD2 utilizes Chapie+ Payload Enclosure System

Ideally, the temperature for storing vials and injectables of medications is typically between 36-46F. Because the ice options have temperature ranges that could drop below our ideal temperature, it runs the risk of the deliverable becoming too cold. When medicines become too cold, it will become compromised and begin to degrade in quality. Thus, for either of these icebox options, we would have to implement a separation barrier between the deliverable and the ice/gel. Depending on our choice of which type of ice to use, the barrier’s material and thickness will be adjusted for ideal thermal energy maintenance.

Figure 52

5.3.3) Active Cooling Systems

Active cooling is the process in which these types of heat transfer are attained by means of an electrical-device or mechanism. This would mean cooling is attained through inducing current to one or more electrical components that would maintain the temperature of our deliverable by transferring thermal energy away from the system. Examples of this are: Peltier cooling, liquid cooling, fans, and refrigeration.

Passive Cooling Used in SD2

5.3.3.1) Payload Active Cooling System

An active cooling system can be implemented into MediDrone's payload by establishing a connection between the microcontroller and one of the following systems within the payload. In order to have a connection between the two, we will have to breach the layer separating the payload's contents and the microcontroller. This will result in the loss of the payload's insulation. With a loss of insulation, the deliverable is more susceptible to thermal energy increases as well as atmospheric pressure increases at flight altitude. We'll now analyze methods of active cooling systems to see how they can be implemented into MediDrone's payload design.

Not Used SD2

5.3.3.2) Peltier Cooling

Peltier cooling, uses the Peltier effect to make a one-way flow of heat between two materials. The Peltier effect is made possible through a device that takes an applied voltage, then produces a temperature difference on both sides of the device. Using a Peltier device, we can create this effect inside of the payload on our drone by directing a current into the device from the power supply and having the cooling side of the device facing our deliverable, whilst the heating side is attached to heatsink.

With an active cooling process, we can activate the Peltier cooling mid-flight. We will have a temperature sensor installed inside the package to monitor the temperature of the deliverable for quality assurance. If the temperature of the package drops below ideal, then we can program the microcontroller to direct current to the Peltier device once the sensor flags the microcontroller. Depending on the specs of our power supply, current draws, etc., we will determine if/what features may be disabled during emergency cooling.

The Peltier device requires a 8V / 3A minimum input. The device can reach down to 40F below ambient temperature. So, if the temperature of the compartment gets up to 76F, we can decrease it down to 36F; however, with our temperature threshold we will only need the device to be on very briefly to keep it within the ideal 36-46F. Peltier cooling is an ideal and practical method of cooling items in typically stationary scenarios. By comparing real applications of Peltier cooling systems, we can hope to gain insight as to how we can apply this technology to the preservation of MediDrone's deliverables.

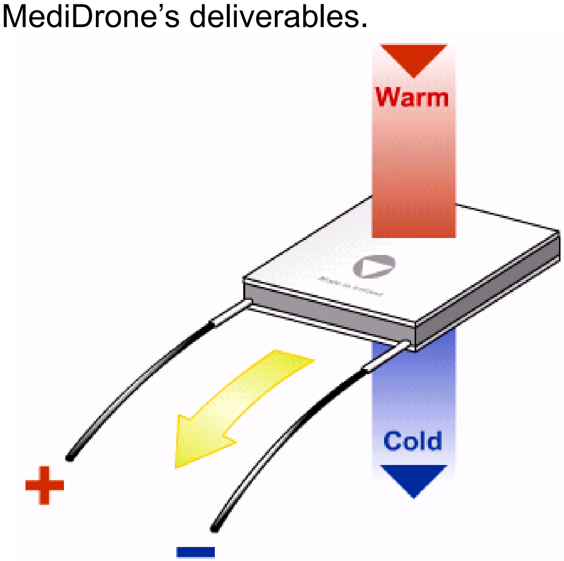


Figure 53 *Peltier Thermoelectric "block" module Peltier Conceptual Image*

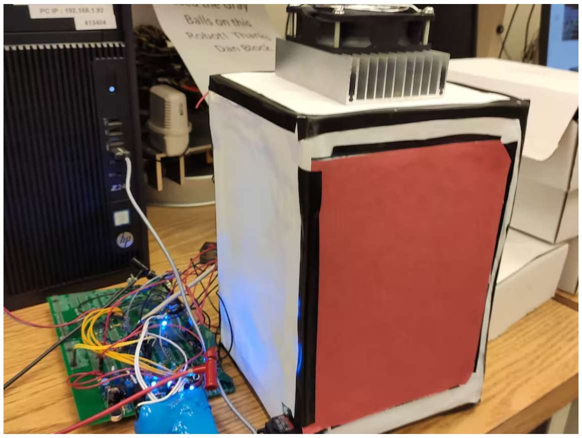
Figure 54 *Amazon Peltier Kit including heatsink and fan.*

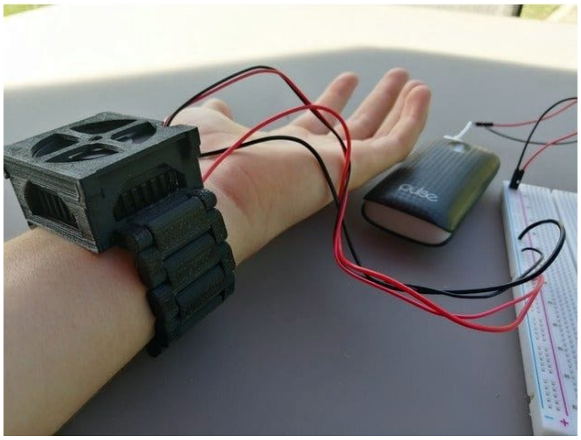
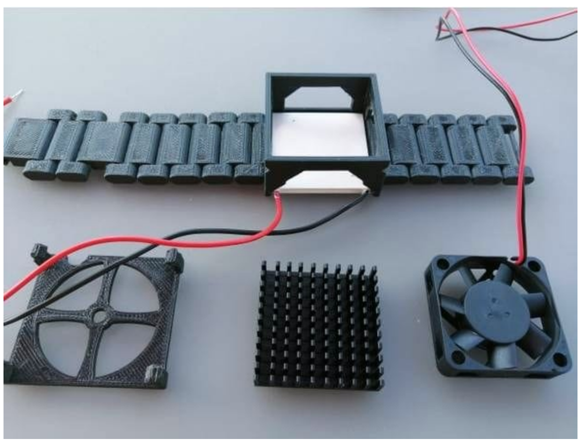
5.3.3.3) Peltier Cooling Not Used SD2	
Pros:	**Cons:**
Doesn't need to be refilled	Draws current away from drone (up to 5A)
Sustainable/Reusable	Can weigh upwards of 2 lbs with just minimum required components (Peltier device, fan, heat sink)
Can be activated during flight	Takes up package space
	Requires non-insulated layer to vent out heat

Table 22

5.3.3.4) Peltier Applications Not Used SD2

Peltier Mini-Cooler https://www.hackster.io/314404/peltier-cooler-94a048	

 <u>Components:</u> Peltier Device Heat Sink Fan Temperature Sensor	Peltier unit outside of package Bulky unit Wired Temperature Sensor Sensor reads when Mini-Cooler goes 0.5C above or below ideal temperature and will turn on or off, respectively
<u>Personal Peltier Cooler [By: Jarno V, Thingiverse]</u> https://www.fabbaloo.com/2019/07/design-of-the-week-personal-peltier-cooler	Non-enclosed design 3D printed components

<table>
<tr>
<td>

<u>Components:</u>

Peltier Device

Heat Sink

Fan</td>
<td>Compact and lightweight

Non-metallic material

In case of damage due to drone failure, can be reprinted

Heatsinks hot side facing arm

Blows cold air from cold side

Would not be applicable for MediDrone because of its unenclosed design</td>
</tr>
<tr>
<td><u>Ice Maker using Peltier Effect [By: DroneBotWorkshop</u>

https://dronebotworkshop.com/peltier-effect/</td>
<td></td>
</tr>
</table>

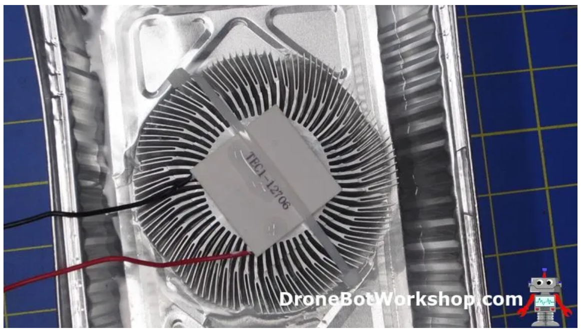
  <u>Components:</u> <u>Peltier Device</u> <u>Heat Sink</u> <u>Fan</u>	Non-enclosed design Hot side is heat-sinked Cold side is submerged in water that is then frozen after 12V input is supplied Would not be applicable for MediDrone because of its unenclosed design

Table 23

5.3.3.5) Peltier Block Diagram Not Used SD2

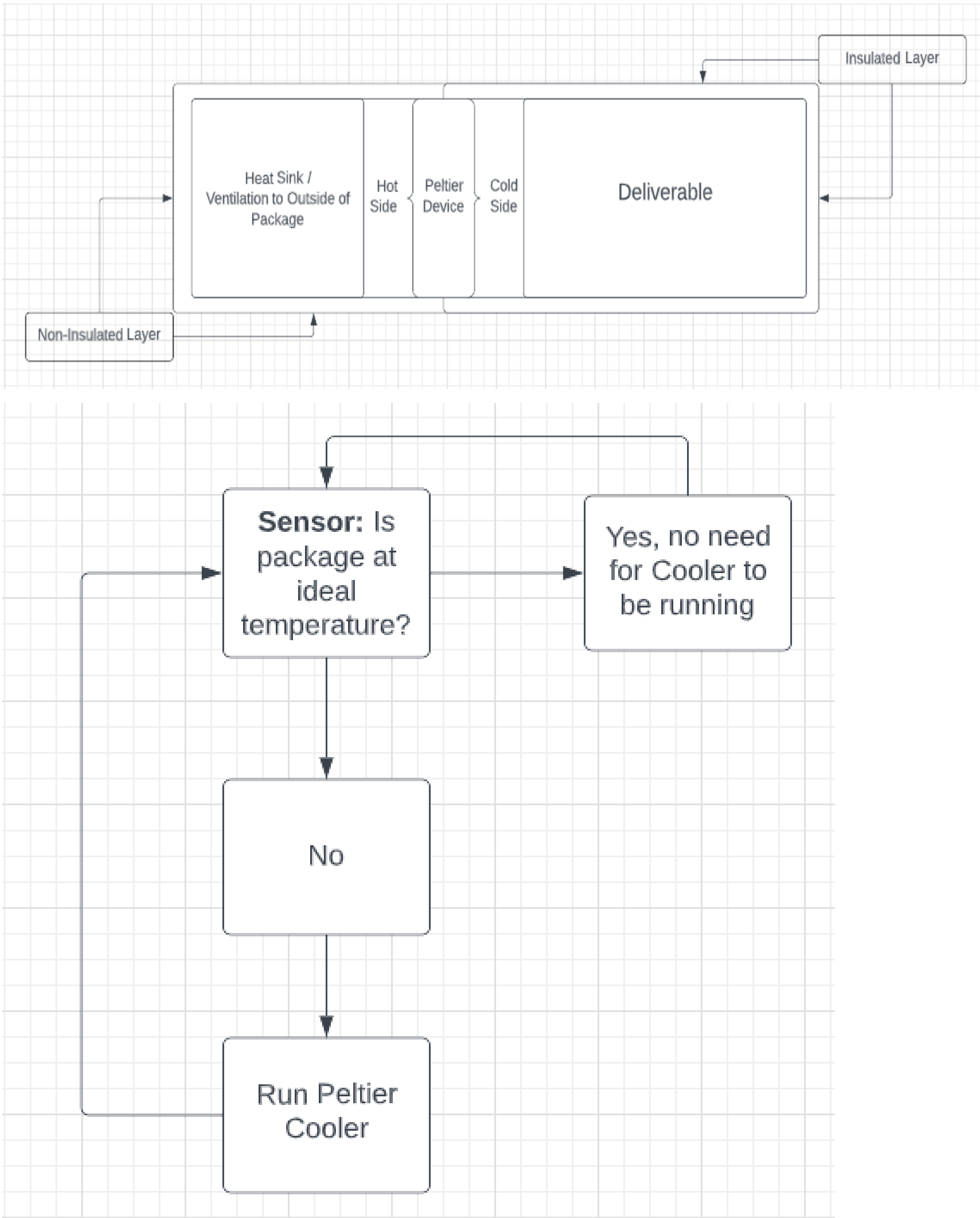


Figure 55

5.3.5.6) Peltier Application for MediDrone

In order to implement a Peltier Device to the payload of our drone, we would have to forgo an insulated container for the medical supplies. In the event of failure of the active cooling system, our payload would be left uninsulated and could perish in quality before reaching the destination.

5.3.5.7) Testing and Characteristics

Insulated only box	**Box with cooling mechanism Not Used SD2**
Will need RF type communication link from sensor+mcu inside box to receiver outside of box to keep proper insulation **Not Used SD2** Electronically produced thermal energy will have to be deemed negligible for a good test result Testing Constraint: Wire from thermistor inside box to outside of box can lower the insulation	The Peltier module itself will undergo testing in non-insulated conditions prior to being designed into & installed into the payload box. Testing Constraint: The test setup adds weight to a payload already cp
Compare Iterations of: Store Bought Insulated only, Designed insulated only , Store Bought+Cooling, Designed Box+Cooling SD2 uses insulated only, “store bought” Chapie with in house design cylindrical payload enclosure unit.	

Table 24

<u>Vacuum Insulated Container</u> *Dam Vacuum Insulated Cooler* https://www.damcoolers.com/product/dam-cooler-35/  <u>Characteristics:</u> External Dimensions (L X W X H): 15.3" x 14.5" x 21.15" Internal Dimensions (L X W X H): 10.7" x 11.5" x 16.8"	**Not Used SD2** Designed to hold food and beverages 1.3" Thin-Wall Design of vacuum insulation between double walled exterior and interior layer Heat can only be introduced into container when opened Vacuum insulation provides a low weight and reliable solution to packaging, as it has the most effective form of insulation and does not add any weight

<u>Thermal Foam Insulated Hard Container</u> *Yeti Tundra® 35 Hard Cooler* https://www.yeti.com/coolers/hard-coolers/tundra/10035020000.html  <u>Specs:</u> Internal Dimensions (W x D x H): 13 7/8” × 9 3/8” × 10 5/8” External Dimensions (W x D x H): 21 3/8” × 16 1/8” × 15 3/4” 3-inches perimeter of Polyurethane foam insulation on all sides	**Not Used SD2** Designed to hold ice and beverages Heat can permeate through container as thermal foam absorbs it (for our flight plans, it would insulate for our ideal flight times Polyurethane foam density can range from 2 to 60 pounds per cubic foot Not ideal for adding more weight to payload Heat can be introduced into container when opened Hard walls provide solid structure for contents inside The combination of a hard container and a weighted insulation material does not seem ideal for drone payload

<u>Thermal Foam Insulated Soft Container</u> *MIER Portable Thermal Insulated Cooler Bag* https://tinyurl.com/Soft-Thermal-Container  <u>Characteristics:</u> External Dimensions (L x W x H): 9.5" x 6" x 3"	**Not Used SD2** Lightweight Interior lined with thin layer of thermal foam between of aluminum and polyester Soft walls provide loose structure for contents inside Non-ideal packaging for deliverable, without sturdy structure and loose insulation this method of packaging does not seem reliable

Table 25

After comparing these types of container insulation, it is apparent that a vacuum insulated container will be our most effective method for deliverable preservation.

5.4) Chapie Sponsorship

Figure 56

5.4.1) About SD2 MediDrone used chapie successfully

What is Chapie?
Chapie offers vacuum insulated products, championing designs scoped to keep chapstick, lipstick, and other temperature sensitive products, in an unmelted physical state, in conditions where the contents would be ruined due to melting such as in a hot car or a trip to the beach.

Why Test Chapie for a Medical Delivery Drone?
Chapie has undergone rigorous testing for it’s intended scope, however, vacuum insulation may offer vast medical utility, bringing about a crossroads between MediDrone and Chapie for collaboration.

Who is Chapie?
Chapie was founded by Jacob Kirstein, a UCF Aerospace Engineering graduate. Seth and Joey have had prior engineering collaborations with Jacob. Chapie is inspired through experience, and entrepreneurial spirit. MediDrone is excited to work with Jacob as well as Jacob is to mentor MediDrone. Both organizations believe in uplifting each others technical ideas and further iterations of both MediDrone and Chapie.

From the Chapie website: MyChapie.com

Chapie was created by former SpaceX and NASA Aerospace Engineer, Jacob Kirstein. One day in 2020, during the height of the COVID-19 pandemic, Jacob's girlfriend asked if there was a way to prevent Chapstick from melting on hot days. Being from South Florida, this was a challenge they were both very familiar with. Jacob got to work and 3D printed a container that would become the first Chapie prototype. The prototype container was posted on social media and went viral overnight. Jacob got to work finding manufacturers to create the world's smallest vacuum-insulated flask. It wasn't all smooth sailing, Covid delays, hackers, and manufacturing methods plagued the production of the first Chapie presale campaign.

Figure 57

After 3 failed attempts to mass produce Chapie, while running out of time and money, Jacob gave it one more shot to get the product off the ground. Finally, a heroic effort by an advanced manufacturer proposed a proprietary process to create what is now known as Chapie. Since the first presale campaign, Chapie has been an instant success. The home-run product is continuously selling out online and in stores. As a company, we are focused on delivering a 5-star product accompanied by 5-star customer service. We are constantly innovating to create more advanced versions of our products and are in the middle of expanding our product lines to serve even more customers.

MediDrone considers Chapie as an ideal partner in the engineering process, completing senior design, but also contributing to industry other than UCF's design competition while participating in senior design. The opportunity to lend a hand in research, testing, while maintaining an active and negotiated sponsorship will help MediDrone complete a mission of utilizing a drone for medical utility while helping Chapie realize which ways their products contain medical utility.

5.4.2) Contingent “Contract” with Chapie

Medi-Drone X Chapie

Implementation of Chapie Plus

Can contain the "vaccine", Safe for transport, Will keep product contained & at temp

Requires a payload containing "box"

Implementation of Chapie Pro

Can serve as the payload entirely

Design mounting for Chapie Pro beneath drone

Benefit for Medi-Drone:

Use of Chapie, already optimized for high thermal conditions & keeping a delicate product resistant to those conditions

High Thermal Conditions referring to outside temperature during drone flight, similar to conditions of a summer day at the beach

Lightweight payload, no extras required

Benefit for Chapie:

Research & Test : Testing for alternate and medical utility

Process/Package Engineered: Further than the current iterations exploring all utilities

Proprietary Technology: Designed for use with "Chapie Payload" mounting to drone

Chapie "Sponsored" Medi-drone

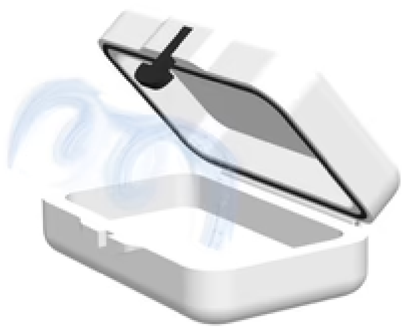
Content Creation: deliver unharmed ChapStick to the pool or beach of your choosing

Requests from Chapie:

Chapie Pro and Chapie Plus units for testing

Current Research on Chapie Products range of utility (Will sign an NDA)

5.4.4) Chapie Products and Specifications Pro/Con

5.4.4.a) Chapie Pro as Payload https://www.mychapie.com/products/chapie-pro  Internal Dimensions: 6.25" Long X 3.55" Wide X 1.63" Tall Weight: 11oz **Not Used SD2**	**Pro**: Strategic Partnership with Inventor Already endured test engineering for keeping products sensitive from melting in hot environment Vacuum Insulated Developing a mount will be a partnership with a local inventor, proper IP followed **Con**: No Supply until February Cannot modify, double walls vacuum sealed…. No cooling mechanism… insulation must acquire optimal temperature regulation outside of previously experiments (Chapie’s internal experiments for outside/hot car conditions) “Chapie Pro is a vacuum-insulated “ container that prevents deodorant, gum, chapstick, and lipstick from melting. It can also be used as an insulated small lunch box that keeps food cold or warm

5.4.4.b) Chapie Plus to Serve as Medical Device Thermally Insulated Container **In Self-Designed or Bought Payload** https://www.mychapie.com/products/chapie-plus-singles  Internal Dimensions: 0.84" Wide X 3.25" Tall Used in SD2	**Pro:** The medical product itself can be thermally insulated in a safe container, Box Cooling mechanism insulation is different than medical product insulation **Con:** No Supply until November/December More extreme temperatures to maintain than established research&testing "Chapie Plus is a vacuum-insulated " container that prevents chapstick from melting. Chapie Plus fits most large chapsticks and small chapsticks https://www.mychapie.com/products/chapie-plus-singles

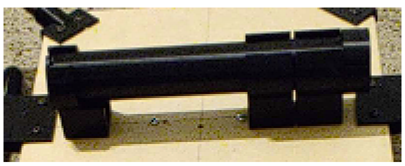
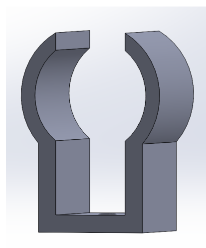
Table 26

5.5) Mounting Design

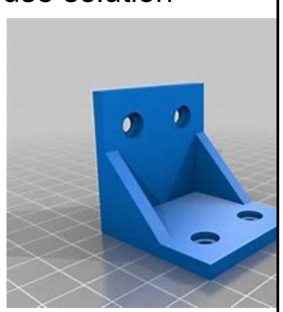
Must be able to support 3 lbs. ***Standard 2.1.1***

Below is a table describing three methodologies, each with a different specification for completing the task for mounting the payload to the drone.

Utilized Cylindrical Mounting System

5.5.1) Comparison of Techniques

Least Professional	Moderate	Stretch
Duct Tape Adhesive Any “Single Use” Solution 	Repurpose brackets or Open Source 3D print Design which also includes single use solution 	Fully Design and 3D print, wood cut, or other manufacturing process in-house Patentable Design  Suction: Stretch as an integration Design or Buy

		 *Example pictures	
Pro: The Payload can be deemed secured to go from point A to B	**Pro**: The Payload can be deemed secured to go from points A to B Low material cost	**Pro:** The Payload can be deemed secured to go from point A to B Technological innovation & problem solving	**Suction Pro:** Can purchase
Con: Little Design aspects Single Use Can ruin exterior of payload box Adhesive may be too strong to unbond from payload box without harming drone	**Con:** Time to design, test, implement In-House design and materials may not be accurate or precise	**Con:** Time Management	**Suction Con:** May lose grip Higher risk Integration Weight

Table 27

While not described in the table above, a mounting system could operate as a drawer if the payload is not entirely enclosed by the quadcopter frame. If the model where feet are used on two sides under the drone's frame, a drawer system can be implemented to slide out beneath the frame on either side not containing supporting feet. Constraints to this model include less stability for a parked drone. However, once the drone is parked, the drawer model may prove to be friendliest to the loading and receiving operators.

The mounting system is not payload concealed under the drone. The SD2 MediDrone utilizes mounting of Chapi's in the payload enclosure unit located atop the wooden frame of Medidrone.

5.6) Testing & Safety of Payload Box & Mounting:

Goals: Safe on the ground, during liftoff continuing through aviation and landing, No Safety Compromise: An unsecured payload box on a flying drone presents many safety hazards.
Safety of payload cylinder is crucial as SD2 Medidrone features the payload enclosure on top of the frame rather than concealed beneath the frame for proper drop mitigation.

5.6.1) Pre and During Flight Testing

Pre-Flight Testing:
- Mechanical/Aerospace Analysis : *See Section*
- Thermal Insulation and Cooling Mechanism Testing *See Section*
- Mounting of payload to drone
- Is the payload interfering with the drone operation?
- How does the drone fly with no mounting or payload?
- Is the payload container accomplishing temperature goals?

Flight Testing
- Is the flight Safe?
  -What are the safety goals
  -Safety with respect to can be thought of as Y(x) Where x==[Drone Systems, Operator, Payload, Contents]
- Flight without mounting and payload versus with payload
  - Complete flight without payload safely----> complete flight with payload----> complete flight with payload and contents
- Must prepare for the unexpected, diagnose problems, find safe solutions

5.6.2) Safety Regarding Drone Scope of Payload Development

	Machine Safety	**Human Safety**
Drone during development	Must be careful and concise with all components and verify all work done through internal control	Proper electrical safety including PPE, being aware if motors are on or off (especially if testing with propellers)
Drone no payload	Will flight make it from A--->B Safely?	Awareness if drone fails in flight Giving information to others at airfield of experimental flight
Drone With Mounting no Payload	Are the mountings secured? Can the mountings break and negatively affect the drone?	Make sure mountings do not fall off or move during flight… Mountings 3D printed may break…
Drone with payload, no contents	How does payload box affect drone performance of accomplishing task A--->B Safely	Mounting must be able to maintain the box affixed to the drone to avoid payload dropping during flight, endangering nearby human cranium
Drone with payload, full contents	Will the contents be delivered safely? (given satisfactory results in the two cells above) Can the drone successfully support a loaded payload container?	Mounting must support the payload container, and it’s contents

Table 28

5.7) Design Drone Payload Concealed Not Used SD2

Figure 58

Payload concealed drones are popular in the commercial drone space, pictured by the United Parcel Service (UPS) drone designed by Matternet below. The payload is concealed under the drone, and is dimensionally in the confines of the drone's body rather than a payload strapped to the bottom of the drone, extending further than the drone's body. Many of the payload concealed style drones are designed to fully stand up with the payload concealed under the body, where the body of the drone extends around the payload, allowing the drone to stand with the payload concealed safely. This allows the drone to take off with the payload rather than pulling up the load after the drone has taken off. This method lowers the torque and jerk on the drone during takeoff, decreasing risk with torque and jerk of the payload suspended from compromising takeoff.

Possible methods of a concealed payload designed drone may include the lid of the payload box securing the payload while also securing into the drone. Design includes the attachment of the payload to the drone, but must also include the separation of the payload from the drone once the delivery has been made. This separation must not compromise the drone, the payload, or the safety of the individual handling the drone. Allow payload mounting to refer to systems on the payload box which attach to the drone and drone mounting refer to the design of the drone to be compatible and mount the payload box. Methods of designing the payload mounting and drone mounting systems can be prototyped using an STL design application and prototyped using 3D printing.

Constraints of a payload concealed drone are limited but do exist. The payload must physically be smaller, leading toward the payload loading less contents than a non concealed payload. The payload must also have proper security concealed, which may be more difficult to design and test. Failures of any kind during flight can be terrible, not only toward the payload and its contents, but to any human or animal below the drone. The concealed payload must extensively be tested for flight-like conditions before flight, before high altitude flight, and before a long duration flight. Payload concealed drone may not be best for the loading and receiving operators. This development comes as a drawer model of payload mounting can be the most user friendly.

5.7.1) Testing of Concealed PayloadNot Used SD2

Given these conditions:
First: Not in flight
Second: short low altitude flight
Third: short high altitude flight
Fourth: long low altitude flight
Fifth: a long, high altitude flight

After success in these procedures, we will run multiple trials, and allow the drone to operate in a space with willing and safety aware humans (possibly wearing hardhats)

5.7.2) Concealed Payload Pro/Con

Pro	Con
• When successful, allows for better aerodynamics • Less risk in air • Safer if designed concealed	• More extensive design on the Drone specific to the mounting system as well as the mounting system toward the drone

mounting is 100% secure	● Must be operator friendly

Table 29

The concealed payload most likely will not be used in the final MediDrone product. Design constraints are very prevalent, as shown in mechanical design, a payload semi concealed by drone legs is much less weight than being totally concealed. The drawer payload method, if used, would not be a concealed payload where the concealing features act at the drone's legs; however the payload will be secured properly and allow for the best in-house design by MediDrone's mechanical team.

6. Hardware Design Details

6.1) Flight Hardware

There are a multitude of ways to design a flight controller, each design having its own benefits and consequences. Given the wide variety, this section will focus mainly on MediDrone's implementation and will weigh out the different options of the subcomponents within. We have decided to go with a Raspberry Pi operating as the central unit, while it reads and writes to a controller area network (CAN) bus. This serial communication protocol is used mainly in noisy-environment systems and will help minimize signal distortion from node-to-node communication within the drone. There will be a segment below overviewing this protocol. This section will also break down the different sensors, wireless communications, and positioning system into subsections, each highlighting the pros, cons, and why they fit into MediDrone's design. The testing and prototyping process of the controller and which out-of-box modules can be used can be found in the testing section of the paper further on, some mentioned throughout here as well.

6.1.1) Flight Controller

The drone's flight controller will be a designed, custom, printed circuit board mainly featuring a Raspberry Pi, a microcontroller, an external Bluetooth IC, an Inertial Measurement Unit (IMU), and a CAN transceiver to bridge with the hardware network in place. The sections below allow greater insight into each component. The PCB specifics are noted later in the paper. The flight controller, conceptually, sends signals to the electric speed controllers in order to adjust the flow of current to each motor. These instructions are used to essentially control the direction of the drone and its activities. A flight controller resilient to external conditions, with highly functional communication will be essential toward a functioning MediDrone.

6.1.1.1) Raspberry Pi

First and foremost, the Raspberry Pi was chosen over another single board computer (SBC) like the Beaglebone due to the team's familiarity and experience with the hardware and operating system. The Pi's usage was mainly a consideration for the software complexity. The Pi will help simplify and organize the software's architecture to meet certain criteria the team has decided to put in place. For more information refer to the software section of the paper. We have decided to use the Raspberry Pi Zero 2 W as our SBC due to its smaller size and non-excessive GPIO/port count. When comparing this to the Zero's predecessor, the Zero W, the CPU upgrade was the main standout decision in choosing the Zero 2 W, for only a slight price increase. The Zero 2 W utilizes a quad-core Cortex-A53, with a max clock frequency of 1.2GHz while the Zero W is only a single-core Arm11 processor having a max clock frequency of only 1 GHz. Both feature similar RAM with the same size of 512MB. Another main selling point for the Zero 2 W compared to the Zero W is the upgraded Bluetooth version from 4.1 to 4.2, however, the flight controller will utilize an external Bluetooth 5.1 chip for a greater range of communication which will be discussed later in this section. Without diving too deeply into the software, the Pi's focus is to send abstract flight commands to a microcontroller where the microcontroller then places broken-down commands, into segments, onto the CAN bus. More details on the microcontroller on the main flight board are located in the same section below. Being on the same board, with minimal distance between, the Pi and microcontroller's bidirectional communication will utilize UART or another serial communication as the Zero 2 W is capable of any of SPI, I2C, and UART connections from its 40 GPIO pins.

6.1.1.2) GPS

The GPS receiver is a crucial component for an autonomous vehicle. The coordinates provided by the module will allow the drone to get within a close proximity of the landing pad set at the desired location. The landing pad will then be a key factor to safely land the drone precisely within a small range of only a few centimeters. More details about this are found in the landing pad section. The GPS module will be placed on the main flight board.

6.1.1.2.1) NEO-6 Series

U-blox's NEO-6 GPS modules offer a cost-effective solution to reading GPS coordinates through their U-blox 6 positioning engine while maintaining a smaller chip package. This engine is capable of providing a Time-To-First-Fix (TTFF) of less than 1 second (from a hot start). A TTFF is a common GPS measurement based on performance that is essentially the latency of a GPS device acquiring satellite signals and navigation data and then calculating a position (referred to as a fix). This relatively low-to-standard latency will offer better performance from the MediDrone. All the NEO-6 modules are capable of communicating with a device with UART, USB, SPI, and I2C leaving room for adjustment when interfacing from the Pi. Another important specification to consider when choosing this module was its horizontal position accuracy. This module is capable of getting within a horizontal vector's magnitude of 2.5 meters. The IMU, discussed later, will provide detail on how the drone will know its current height in the air. With this knowledge, the precise landing implementation for the drone was constrained to have a minimum of 10 feet of detection, see Precision Landing section.

6.1.1.2.2) Antenna

According to the datasheet provided by U-blox for the NEO-6 GPS modules, the module is capable of supporting either passive or active antennas. Active and passive antennas are the same in physical structure yet active antennas include a built-in, generally low noise, amplifier for the incoming signal. An in-house design and build of an antenna is a possible ordeal, utilizing a proper spectrum analyzer and vector network analyzer. MediDrone will seek to purchase an antenna, however, will be outfitted to build an antenna module for transmission and reception in the case a bought antenna falters. An amateur radio license allows a civilian to build, test, and operate RF equipment including antennas, transmitters, and receivers, allowing the MediDrone team (with a single licensed member) to properly design, assemble, and test a home brewed antenna module. The datasheet provides exact specifications on recommended gain distributions for active antennas. For MediDrone's application, we have decided to use the GPSMOD2515 antenna. This active antenna meets the 15dB minimum gain U-blox has in place for the NEO-6 modules by providing a 16dB low noise amplifier. There is an adhesive tape on the bottom of the PCB that will be able to fit onto the flight controller. Additionally, the GPSMOD2515 has exposed solder pads suitable for a coaxial cable connection.

6.1.1.3) Direct-Flight Communication

It will be important for a ground engineer to be in contact with the drone while it is in flight. This will deal with monitoring the status of the drone and the drone's payload. For instance, if the point-of-destination changes mid flight the engineer must send the new coordinates to the drone. Also, if the medical deliverable's temperature rises above a temperature in which it could become perishable, the shortest path to saving the load may be to turn around. Realistically, MediDrone would have many repeaters, ground stations, and landing pads built as a network. Thus, requiring only 0.5 miles of communication range could be sufficient. There are various other reasons why direct communication and mid flight monitoring may be beneficial. With this said, reliable, low-cost, long-range communication is a feature the team felt was imperative to add.

6.1.1.3.1) Bluetooth 5

There are a lot of misconceptions in place about Bluetooth and that it's a standard for only short-range wireless communications. Bluetooth 5 has the capability of sending signals up to 1000m in an open field, with a reduced bandwidth. Not only that, Bluetooth 5.1 introduced Angle of Arrival (AoA) which will be discussed later on in the precision landing section. Nordic Semiconductor offers the nRF52840 system-on-a-chip (SoC) which supports Bluetooth 5.3. The company also produces a small breakout board as a dongle with the SoC making it easy for programmers to start developing immediately. In the dongle form, once programmed, the USB A port from the board can directly connect and communicate with the Pi on the flight controller. Thus, allowing the drone to potentially utilize the long-range capabilities of Bluetooth 5 and potentially AoA for precision landing. Using this SoC would allow easy scalability to add AoA landing pads later on if the drone initially utilized a different form of precision landing.

6.1.1.3.2) nRF24 Series

Another RF transceiver series from Nordic Semiconductor, the nRF24 series utilizes the Enhanced ShockBurst protocol (ESB) to transmit signals across the Industrial, Scientific, and Medical (ISM) 2.4Ghz band while still utilizing Gaussian frequency-shift keying (GSFK). The most commonly used, with vast user documentation, is the nRF24L01+ chip. This IC has many reported distances all depending on the environment, antenna, and surrounding schematic. This could optimistically give the drone upwards of 500m of distance to communicate with even though there have been much greater distances recorded. This could be a better suitor for the drone as it is more cost-effective and simpler to integrate within the flight controller. Additionally, there are various supporting online tools mentioning the implementation of this IC within various systems, unlike the newer, more complex nRF52840 SoC.

6.1.1.4) Central Microcontroller

The role of the microcontroller on the flight controller is mainly to segment and compile down higher-level commands sent to it by the Pi. For example, the Pi may send an abstract command to the controller like (turn to 265°, altitude + 1), where the microcontroller then will need to break this down further into commands for the motor electronic speed controllers (ESC). On top of this abstraction difference, an important role for this microcontroller is to also relay this information to all other nodes in the hardware network, in which the ESC/Motor nodes will be connected to. To do this, the controller will simply feed the CAN transceiver the simplified commands. More information on CAN and the exact receivers will be noted later in the Hardware Network section.

6.1.1.4.1) MSP430

To keep the entire system cohesive, each CAN controller will be the same microcontroller. Including this main microcontroller on the flight controller board. For this, we have decided to use the Texas Instrument’s MSP430 due to the team’s experience and knowledge with this controller. Additionally, this controller is relatively a general-purpose microcontroller and can vary in functionality to meet every network node’s need. The last important aspect of this controller is simply to make it so if there’s anything that can’t generally hookup to the Pi, whether that’s a lack of GPIO pins or there’s just not an interface that the Pi supports but the controller does, the controller can just send information from this module/circuit back to the Pi. The two should work cohesively to make the software abstraction a bit greater, offering a more fluid architecture.

6.1.1.5) Inertial Measurement Unit

An inertial measurement unit, or IMU for short, is a crucial part to any aerial vehicle, especially a drone when considering the different angles it may be oriented at (pitch, roll, and yaw), its height within the air, its current acceleration in one direction, and more. An IMU is generally composed of multiple different sensors like an accelerometer, gyroscope, magnetometer (compass), barometer, and more. Given the complexity of these units, during the testing phase of the flight controller, it may be a good idea to choose a few different IMUs to accumulate data to then see what unit lends the best results for MediDrone specifically. It should also be noted that the unit's orientation with respect to the controller board should be as flat/precise as possible to provide the most accurate inputs possible. This will be discussed later in the PCB design section. When picking out an IMU it is also important to keep in mind the software packages that are available for the specific piece of hardware. Sensor Fusion is commonly used when providing an output from all of the sensor inputs within an IMU, this can provide more accurate results than if providing the sensor inputs separately. However, this then leads to the point of ensuring a software library is available for the specific unit. A few potential options for these units can be seen below, all within a reasonable price point.

6.1.1.5.1) FIS1100

This surface mount IMU uses a 3-axis accelerometer and a 3-axis gyroscope, allowing 6 degrees of freedom (6DOF) sensor fusion. The package has two pins that can connect to a 3-axis magnetometer through I2C offering the ability to produce 9DOF sensor fusion. The IC's datasheet offers the solution to pair the AK8975C compass through I2C to complete the capability of 9DOF. This is surely one negative about this chip is that it will require the controller to incorporate both ICs to get a complete 9DOF. However, the AK8975C comes in a relatively small package and is a non-expensive component. But there may be a small, undesirable, latency for complete sensor fusion when having one of the inputs fed over I2C continuously. The main pros for this IMU is that it's inexpensive, it has a user-friendly datasheet for configurations, and that a development board

(FEBFIS1100MEMS-IMU6D3X-GEVK) is also offered for early testing. The XKF3 software development kit (SDK) also appears to contain all of the sensor fusion libraries needed for interfacing and programming the IMU.

6.1.1.5.2) LSM6DSLTR

This IMU appears to have a few differences when compared to the FIS1100. Firstly, it doesn’t appear an external magnetometer can be integrated with the other sensor inputs. It also doesn’t appear that this unit performs sensor fusion. Instead, the data can be read, directly and continuously, when programmed from registers within the IC. The unit is also 6-axis (3-axis accelerometer and 3-axis gyroscope). This IMU also has an embedded temperature sensor within, this isn’t a necessary function for an IMU but is just another feature that may lead to some use and allow more integrity of the system later on when developing and testing. This IC can communicate with I2C and SPI, unlike the FIS1100 that can only talk via I2C. This IMU is additionally a surface mount IC.

6.1.1.6) Controller Diagram

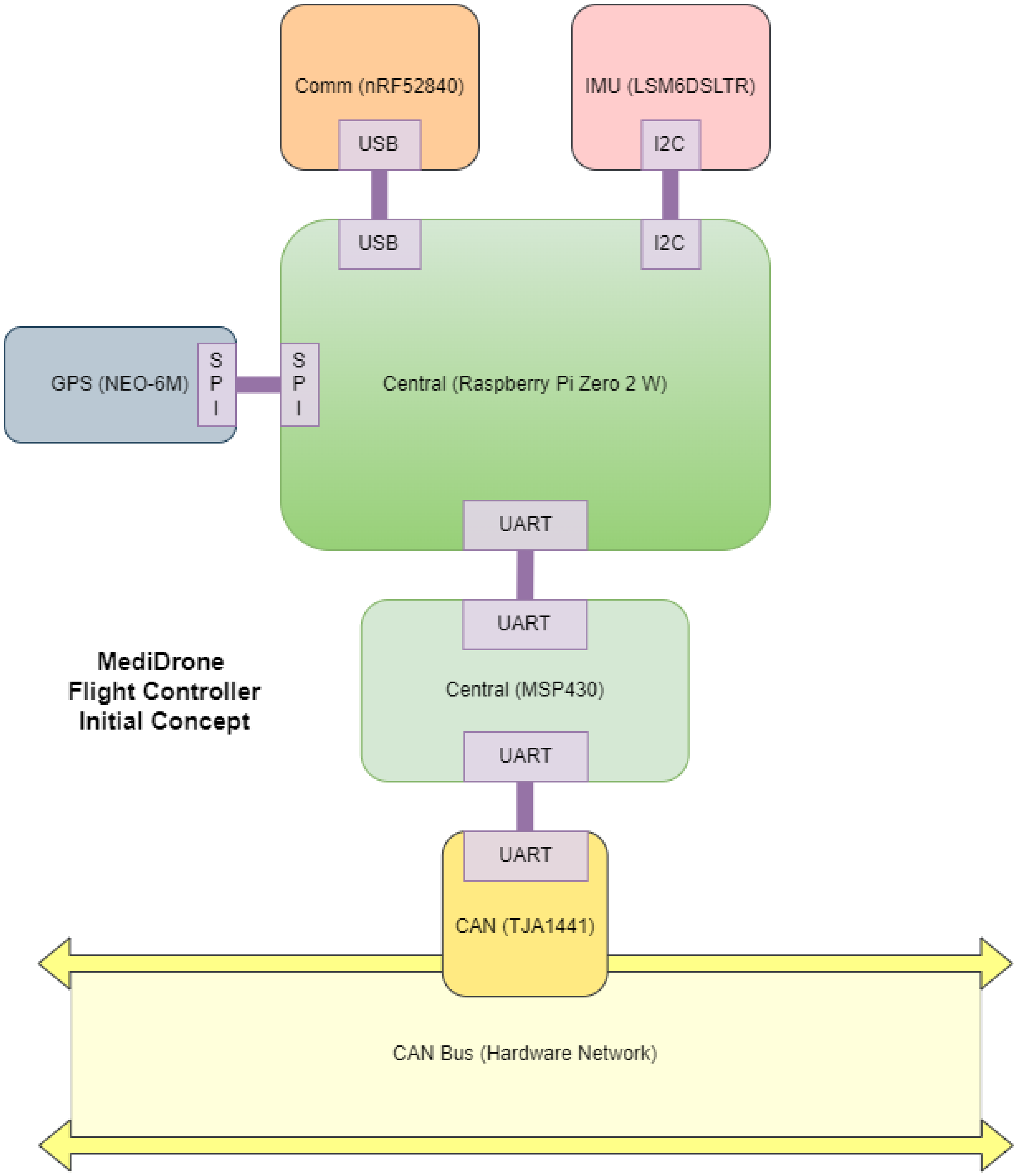


Figure 59

6.1.2) Hardware Network

The main flight controller board will be connected to various PCBs throughout the drone. A few of these are sensor boards, brushless motor boards, payload boards, and a few others. It's important to note that the drone's brushless, magnetic, motors will be causing a rather large, electrically noisy environment for the drone's hardware. It's also imperative that the signals sent to and from the different PCBs in this network are high-integrity and the system is robust to the point where little to very few signals are distorted or lost. This is crucial since there can be rapid, severe consequences for a single miscommunication. This is needed not only to improve the quality of the product, but to ensure the safety of the surrounding environment. Given this defining circumstance, the system required a way of transmitting data safely.

6.1.2.1) CAN

Understanding that the system must deal with a very noisy environment, the communication protocol we decided to utilize in the system was the Controller Area Network (CAN) protocol. This, two-wire, serial protocol was developed for, and is frequently used in, in-vehicle networks that are prone to high quantities of electrical disturbances. The CAN protocol utilizes a bus, where every board, device, or "node" connected to it can broadcast to. Every node is therefore capable of sending and receiving signals from it. To do so, every node utilizes a controller and a CAN transceiver, essentially translating input data into CAN frames. For the specifics on MediDrone's CAN transceivers and controllers, more information is provided in the upcoming subsections.

6.1.2.1.1) Protocol Overview

Luckily, most of the weight of implementing this protocol is done through the CAN transceivers. However, it is still important to understand how this protocol actually works for later debugging and improvements. Something interesting about the protocol is that it's peer-to-peer, allowing every node to broadcast to the bus as they desire. There are certain rules in place though. Firstly, a node can only write to the bus when they see the bus is not busy. Secondly, the protocol is based upon node priorities. For instance, if two nodes decide to broadcast simultaneously, the node with the higher priority, or lowest arbitration ID, is allowed to broadcast first. Each node has a distinct arbitration ID and each frame written to the bus contains an arbitration ID. Every node can receive every frame, but the node with the arbitration ID is able to accept the given data. The two signals, CANH and CANL (CANHigh and CANLow), on the bus can be protected from electromagnetic interference with a twisted pair, as the two signals form a differential pair. The bus also needs to be terminated with two 120 Ω resistors. There are multiple different ways CAN addresses errors, including but not limited to Cyclic Redundancy Check (CRC), bit stuffing, frame checking, acknowledgement checks, and more.

6.1.2.1.2) CAN vs. Common Serial Communications

	CAN	I2C	SPI	UART
Max Bit Rate	1Mbps	5Mbps	10Mbps	5Mbps
Advantages	1) Robust 2) Error detection 3) Multiple nodes 5) 2 wires	1) Reliable 2) 2 wires 3) Multiple nodes	1) High-speed 2) Continuous transmission 3) Multiple nodes 4) Full-duplex	1) Full-duplex 2) 2 wires 3) Simplicity 4) Minimal error checking
Disadvantages	1) Slower than peers 3) Half-duplex	1) Address conflicts 2) Half-duplex	1) Multiple wires 2) Max of 1 master 3) No error checking	1) Max of 2 nodes

Table 30

The table above only outlines a few of the advantages and disadvantages of the most popular serial communication protocols and the CAN protocol. The greatest weights when deciding how to construct the drone's hardware system are fault tolerance and robustness.

6.1.2.1.3) Node Transceivers

The following transceivers all are potential candidates for the network's CAN nodes. Each has some minor differences that will be listed below. All these transceivers share the same physical attributes of having two pins for CANH and CANL (bus lines), and share the two pins, Tx and Rx, in which the CAN controller (microcontroller) writes and reads from.

6.1.2.1.3.1) TJA1441

This particular transceiver offers the use of the CAN FD (Fast Data rate) protocol which allows the data rate on the bus lines to vary from the standard CAN data rate (1Mbps) to 5x the data rate (5Mbps) by overclocking the payload portion of the CAN frame. CAN FD is just an extension of CAN and shares all the robust features that it offers. This chip comes in three variations (A, B, and D); all of which have minor different pinouts and offer different combinations of their normal and sleep modes.

6.1.2.1.3.2) TCAN1462-Q1

This Texas Instruments IC offers CAN FD that supports a data rate up to 8Mbps for larger networks. This, just like the TJA1441, and more likely than not all CAN FD transceivers, are backwards compatible with the classical CAN protocol. Common modes this chip supports include the normal mode and low-power standby mode that supports remote wake-up requests. The datasheet also advertises that the chip meets the CiA 601-4 Signal Improvement Capability (SIC) specification, and that the device reduces signal ringing and the dominant-to-recessive (differential pair) edges and therefore allows better extraction of CAN FD's full capabilities.

6.1.2.1.3.3) SN65HVD23x

Another Texas Instruments CAN transceiver, this chip operates off a supply voltage of 3.3v compared to the prior two chips above operating at 5v. The signaling rate for this transceiver has a max rate of 1Mbps due to the fact it only supports the CAN protocol, not the extension, CAN FD, the previous two ICs offered. This chip supports the following three modes: high-speed, slope control, and the low-power standby mode once again. This chip seems more simplistic than the other two ICs, which may be beneficial in the long term. However, following some research, this chip is found on a very inexpensive breakout board which may come in handy during the testing and prototyping stages of MediDrone's development.

6.1.2.1.4) Node Controllers

A CAN node in this network once again consists of a CAN controller and a CAN transceiver. The can transceivers all have Tx and Rx pins meaning interfacing is very simplistic. For the controllers, we do not need anything extensive whatsoever, and if anything, a lightweight controller to simply read from a sensor or write to a motor, when needed. The microcontrollers will all be programmed uniquely given whatever node they're stationed at and what their purpose is. A few requirements of the controller will be to: 1) decode a CAN frame into an instruction from input data, 2) encode an instruction into a CAN frame to send to the bus, and 3) perform simple tasks like reading and writing from its GPIO pins. Everything else isn't necessarily required but a few extra GPIO pins and other extended interfaces or modules on the chip wouldn't be a bad idea to leave the nodes with the possibility for extension.

6.1.2.1.4.1) MSP430

As discussed earlier, we are intending to design the MediDrone with MSP430s. These are scalable and general-purpose controllers designed to poll a single sensor or control an entire complex embedded system with many inputs and outputs. Again, for all nodes, we do not want a heavy package. The MSP430 comes in many, many, forms but there are a few controllers we are looking at specifically. First, the MSP430G2353 comes in 20-pin packages and contains a 10-bit ADC, I2C, SPI, UART, and 16 GPIO pins. The main concern for this microcontroller is the lack of RAM, with only 0.25KB. Again, these shouldn't need to be flashed with much but 0.25KB is surely on the lighter side if these nodes are desired to be extendable. This consideration led to a controller with a bit larger of RAM (1KB of SRAM), the . This microcontroller also offers serial interfaces of SPI, I2C, UART, and 17/21 GPIO pins depending on the package. This IC has a 10-bit ADC as well. It will be easier to pinpoint the exact microcontroller that will be used for the CAN controllers once the software is written and tested to ensure a microcontroller with the appropriate components is used.

6.1.2.2) Sensor Nodes

The drone is intended to have various sensing nodes within its hardware network. The main sensors needed are ultrasonic sensors, time-of-flight (ToF) sensors, and a thermistor. The drone will have 6 obstacle avoidance boards on every positive and negative axis of direction. In other words, imagining a cube, every face of the cube will have an obstacle avoidance sensing board placed extending normal to the drone. These boards are intended to be small and will contain both a time-of-flight (ToF) sensor and an ultrasonic sensor to maximize both sensors' benefits. There will be another node to deal with the payload cooling. This thermistor will be located inside the medical box itself.

6.1.2.2.1) Obstacle Avoidance

Obstacle avoidance is imperative with any robot or autonomous vehicle. As mentioned above, the combination of an ultrasonic sensor and time-of-flight (ToF) sensor will be embedded into one board per side of the drone.

6.1.2.2.1.1) Ultrasonic

The ultrasonic sensor the drone will utilize is the popular HC-SR04 circuit. The main negatives to sound sensing when compared to laser/light sensing are the inaccurate outputs and range. In the drone's case, a big factor is the time it takes to sense when sending/receiving a signal. Ultrasonic sensing is considerably slower than time-of-flight (ToF) but is susceptible to fewer external factors and therefore can be reliable for a drone when used in conjunction with a ToF sensor. The next section highlights some of the advantages and disadvantages of both sensors. Both sensors will require immense testing to ensure the safety of the drone.

6.1.2.2.1.2) Time-of-Flight

Time-of-flight (ToF) sensors generally work by emitting a certain light, usually an infrared light, and record the time it takes to receive the reflected light. For obstacle avoidance on the drone, some key characteristics were that this was a 1D (1 dimension) ToF sensor, that the sensor was relatively small, and it operated at a frequency capable of sending and receiving light before the receiver had moved off of position (due to the drone's movement). Most ToF sensors on the market record the minimum distance in which the drone will need to detect an object before crashing. When researching for a specific sensor, it was also important to ensure this sensor would be capable of operating as expected when dealing with external sources of light, like sunlight as the drone will most likely be facing during frequent usage. The first IC under consideration was the TMF8805. This IC operates only at 3v and can be communicated with I2C. The processor within the chip can operate with an external clock of 75Mhz, nominally 4.7Mhz. Looking into the chip's datasheet, the helpful table below offered more insight on the max range the sensor may be able to detect given the lux of the sunlight.

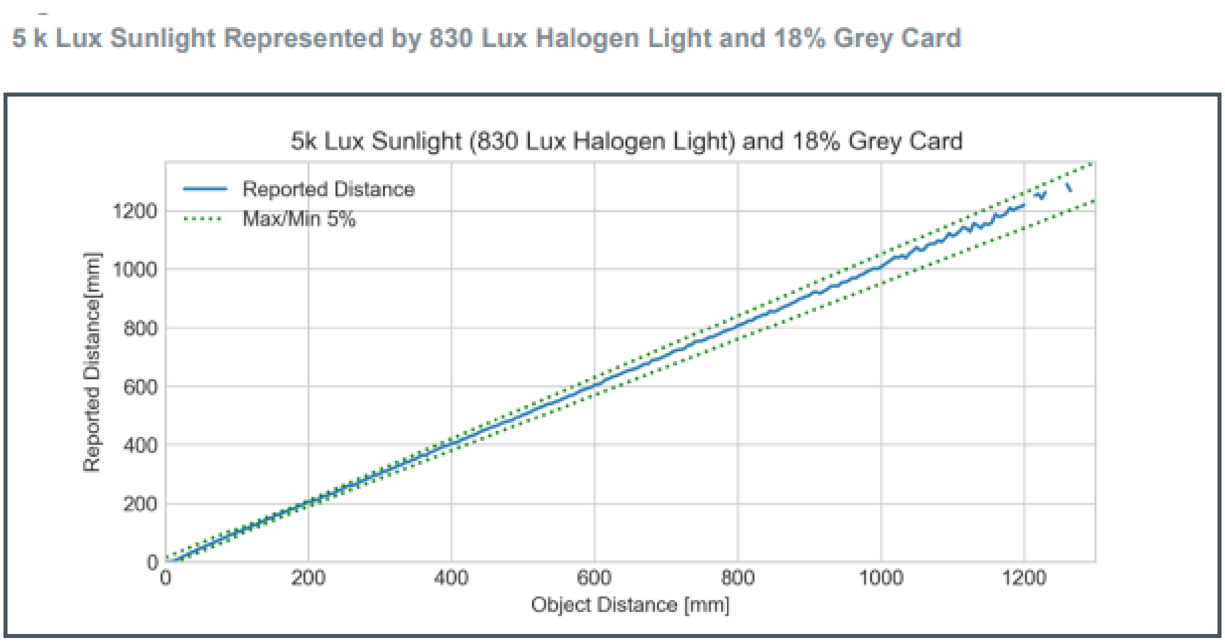


Figure 60

The datasheet also provides other charts providing less lux intensity, with the recorded distance tapering off slightly less. Given the factors of drone movement and ambient light interferences, testing will be imperative when going about integrating the sensor into the hardware network. For reference, the table below provides estimated lux values with corresponding environments.

Illuminance	Example
0.002 lux	Moonless clear night sky
0.2 lux	Design minimum for emergency lighting (AS2293).
0.27 – 1 lux	Full moon on a clear night
3.4 lux	Dark limit of civil twilight under a clear sky
50 lux	Family living room
80 lux	Hallway/toilet
100 lux	Very dark overcast day
300 – 500 lux	Sunrise or sunset on a clear day. Well-lit office area.
1,000 lux	Overcast day; typical TV studio lighting
10,000 – 25,000 lux	Full daylight (not direct sun)
32,000 – 130,000 lux	Direct sunlight

Table 31

Predicting worst case scenarios for this sensor is the safest way to ensure the quality of the drone. To estimate, given the chart from the datasheet and the table above, if the sensor is experiencing direct sunlight, it may achieve potentially less than a foot of detectable distance. In best case scenarios (little to no light), it looks like it can achieve up to ~10 feet at a very fast rate. The rate at which it can detect distance on even the average case, this ToF sensor should be beneficial. But, to ensure the drone knows its limits, the slower ultrasonic sensor will be used in conjunction. Below shows a quick comparison of the benefits and negatives of the ultrasonic and ToF sensors.

	Ultrasonic (HC-SR04 module)	Time-of-Flight (TMF8805)
Advantages	1) Reliable in most environments	1) Fast 2) Small package 3) Very Accurate
Disadvantages	1) Susceptible to faster moving drone speeds 2) Large	1) Susceptible to external light, darker-colored objects 2) More expensive than ultrasonic

Table 32

6.1.2.2.2) Payload Sensing

The drone's goal is to deliver refrigerated medical supplies to a desired location in a quick, efficient manner. Doing so, it would be wise to measure the current temperature of the payload's contents to ensure the trip isn't wasted. The sensing can be done with a simple thermistor on a CAN node, inserted particularly into the payload itself. The chosen thermistor is the NXFT15XH103FEAB035 which will be amplified and fed into the 10-bit ADC from the MSP430 on the node.

6.1.2.3) Motor Nodes

There will be four separate motor nodes, there will potentially be five nodes if the payload servo motor doesn't combine with the payload's temperature sensing node. It would be ideal to keep them together for simplicity reasons, but it depends more so on the mechanical design. The main four nodes, however, will be containing the electronic speed controllers (esc) and the brushless motors themselves. If the payload motor node is separate from the sensing node, it will utilize the same CAN transceiver and MSP430 the rest of the nodes in the network are using.

6.1.2.3.1) Electronic Speed Controllers

We have decided to go with the VESC, open-source, electronic speed controller. This controller was already designed and intended to be within a CAN network. Instead of using the CAN transceiver in their schematic, MediDrone's escs will be equipped with the same transceiver for the rest of the nodes in the hardware network. For more information, refer to the schematics section below.

6.1.2.3.2) Brushless Motors

The brushless motors are one of the more important pieces to the drone. A lot of consideration needs to be taken to ensure the right motors are selected as there are many different specifications that go into a single motor.

6.1.2.3.3) Servo Motor

The servo motor is simply to open and close a clasp that is holding the weight of the medical payload. This can be designed a few different ways, but this section is mainly concerned with the proper servo motor for the application. The main specification that is necessary for the drone is the maximum weight the gears and internal dc motor can support. It'd also be important to give a little wiggle room here as multiple different payloads can have slightly varying weights; we will go with a greater weight than what the drone is able to hold. The motor that MediDrone will administer is Goteck's SER0011. This metal-geared servo motor has a rated stall-torque of 2.3Kg-cm at 4.8v and 2.5Kg-cm at a maximum 6v. In other words, while the shaft is in a locked position, the motor can support up to 2.3 kilograms on a one-centimeter length arm. 2.3Kg, or ~5lbs, is more than what the MediDrone will be holding and can even support given the brushless motors and mechanical structure. This motor only weighs around 3% of 1 lb making it hardly a burden for the drone.

6.1.2.3.4) Hall Effect Sensor

The servo motor node, or payload node, will integrate a hall sensor to detect when the payload is close to the drone. This can be helpful for many reasons. For starters, if we decide to have a wireless circuit within the insulated payload box, it will have to initiate a bluetooth connection to the flight controller when the box is close to the drone. To summarize the process, the hall sensor on the drone side will detect a magnetic field from a magnet on the payload box when it is flush against the drone. This can also be helpful to command the servo to stop turning when the payload is pulled all the way up to the drone. Currently, the designed board seen in the schematics section is equipped with Texas Instruments', surface mount, DRV5033. As far as the idea goes, the datasheet even has "Docking Detection" listed as the first practical application. This reassures the idea that this could be a valuable electrical component to the drone during the unloading/loading sequences.

6.1.2.4) Network Diagrams

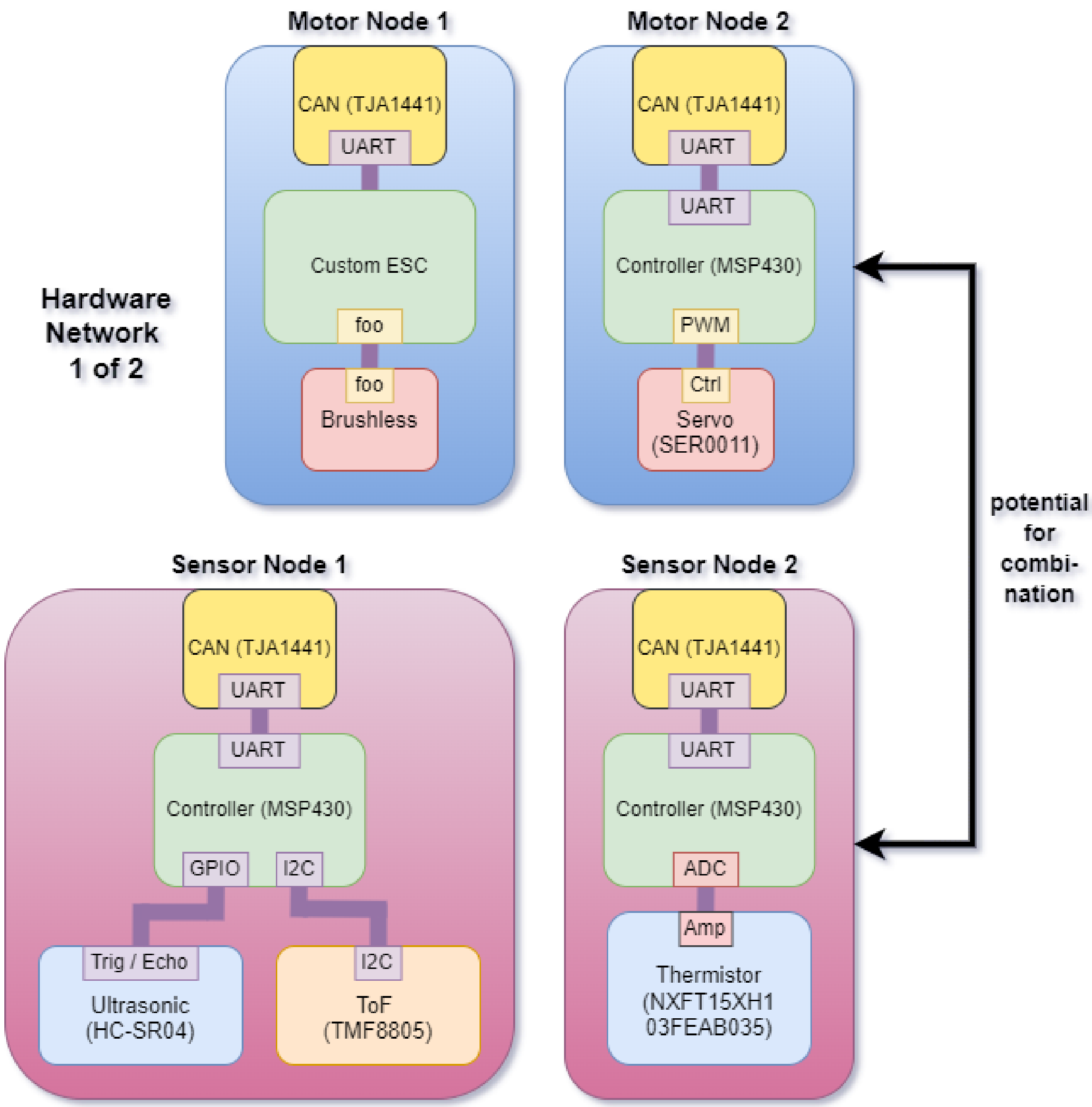

Motor Node 1
Motor Node 2
CAN (TJA1441)
UART
Custom ESC
foo
foo
Brushless
CAN (TJA1441)
UART
UART
Controller (MSP430)
PWM
Ctrl
Servo
(SER0011)
Hardware
Network
1 of 2
potential
for
combi-
nation
Sensor Node 1
Sensor Node 2
CAN (TJA1441)
UART
UART
Controller (MSP430)
GPIO
I2C
Trig / Echo
Ultrasonic
(HC-SR04)
I2C
ToF
(TMF8805)
CAN (TJA1441)
UART
UART
Controller (MSP430)
ADC
Amp
Thermistor
(NXFT15XH1
03FEAB035)

Figure 61

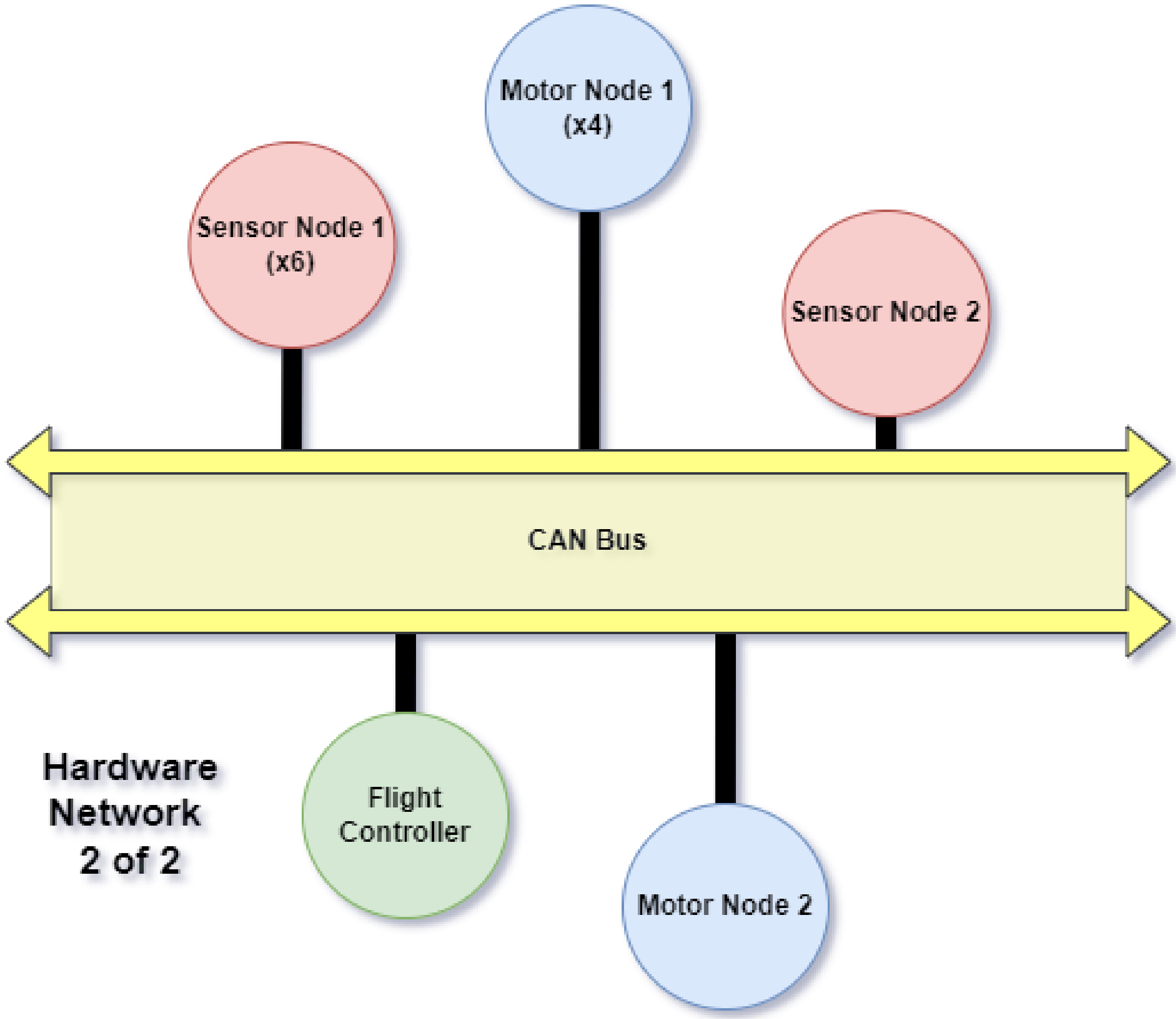


Figure 62

Final Design:

MediDrone's obstacle avoidance circuits underwent a second revision after Senior Design 1, with the primary changes made to the CAN communication interface, and some minor modifications throughout the boards. The boards utilize two proximity detection technologies to ensure reliable and safe flight. The two technologies, time-of-flight (ToF) using the TMF8805 and ultrasonic with the HC-SR04, complement each other's strengths and weaknesses. The HC-SR04 is a simpler ultrasonic breakout board that directly solders into the obstacle avoidance node's PCB, while the TMF8805 is a surface-mount device that uses infrared light to measure distance and can be communicated with via I2C. The TMF8805 has advantages over ultrasound in several aspects. Time-of-flight (ToF) technology is superior to ultrasound in terms of distance measurement because it uses light, specifically infrared light, instead of sound waves. Light travels faster than sound, but it also has more disturbance to filter out. For instance, a bright or hazy day may produce more noise for light sensors than for ultrasonic sensors. The TMF8805, which uses infrared light, can accurately measure distances beyond 3 meters in typical scenarios. The chip's datasheet includes a graph that shows the relationship between typical distances and ambient light, as noise.

Based on empirical analysis using an 18% grey card, the manufacturer determined that the sensor's ability to capture objects is impacted by a greater presence of light at a distance of approximately 2.5 meters. However, the sensor remains reliable up until that point, providing enough time for the drone to change direction when faced with an obstacle. This data illustrates the main limitation of ToF technology, which is why MediDrone's hardware includes an additional ultrasonic sensor. To implement the necessary firmware updates, the TMF8805 required a ram patch and was downloaded through I2C using the MSP430 as an intermediary. The software was flashed with a large number of I2C sequences that were converted from Intel Hex Records through a Python script. The following figure shows a snapshot of the communication between the MSP430 and the TMF8805 over I2C.

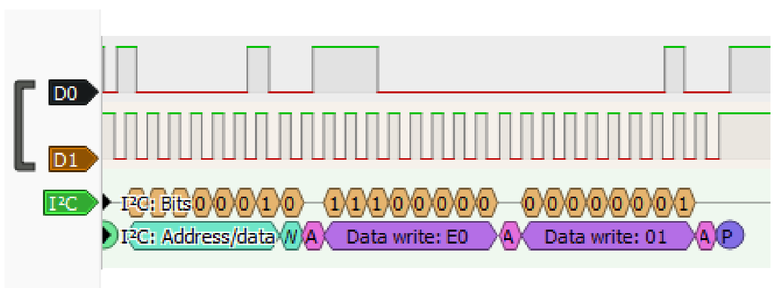


Figure: MSP430 I2C Write To TMF8805

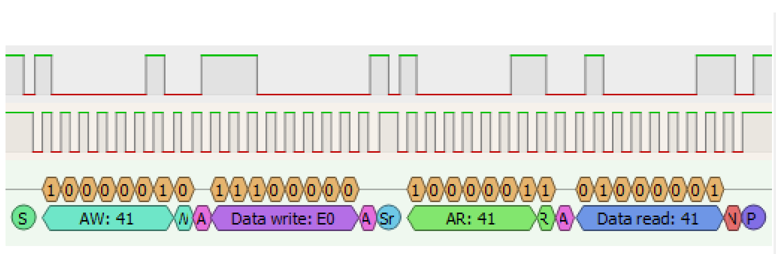


Figure: MSP430 I2C Read From TMF8805

The HC-SR04 offers a simple interface to a microcontroller consisting of two digital I/O, namely a trigger and an echo. To initiate its operation, the trigger is raised to a high level for about 10 microseconds. The echo signal is then raised for a duration that is directly proportional to the distance of an object. This duration can be transformed into distance by using the formula provided below.

$$distance = \frac{time\ taken\ x\ speed\ of\ sound}{2}$$

The above equation is derived from the basic principle of speed equals distance divided by time. By utilizing the constant speed of sound, the distance to and from the object can be calculated. The formula further divides the value by two to determine the distance to the object alone. In order to capture the rising and falling edges of the echo signal, the obstacle avoidance board's MSP430 employs timer interrupts. The following image is a record obtained from another debugging session using a logic analyzer, which displays a constant polling of the trigger signal transitioning from low to high and back to low, along with the corresponding input signal from the echo.

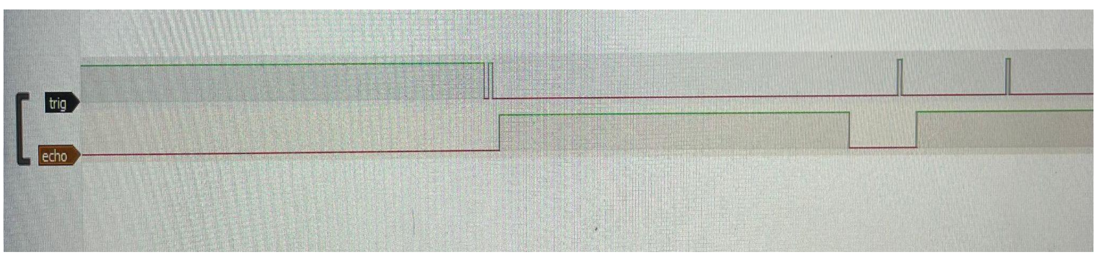
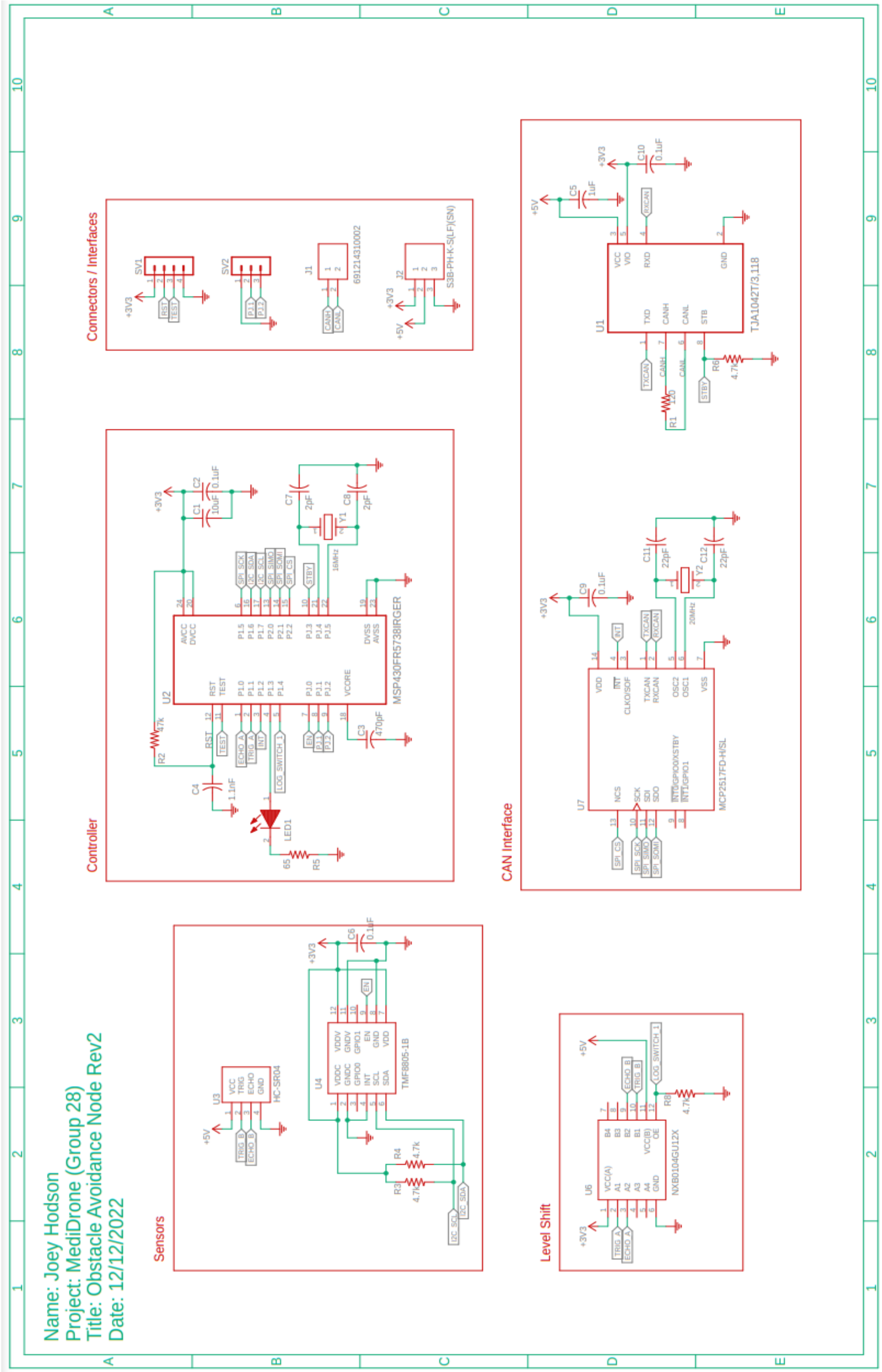


Figure: HCSR04 Trigger and Echo Capture

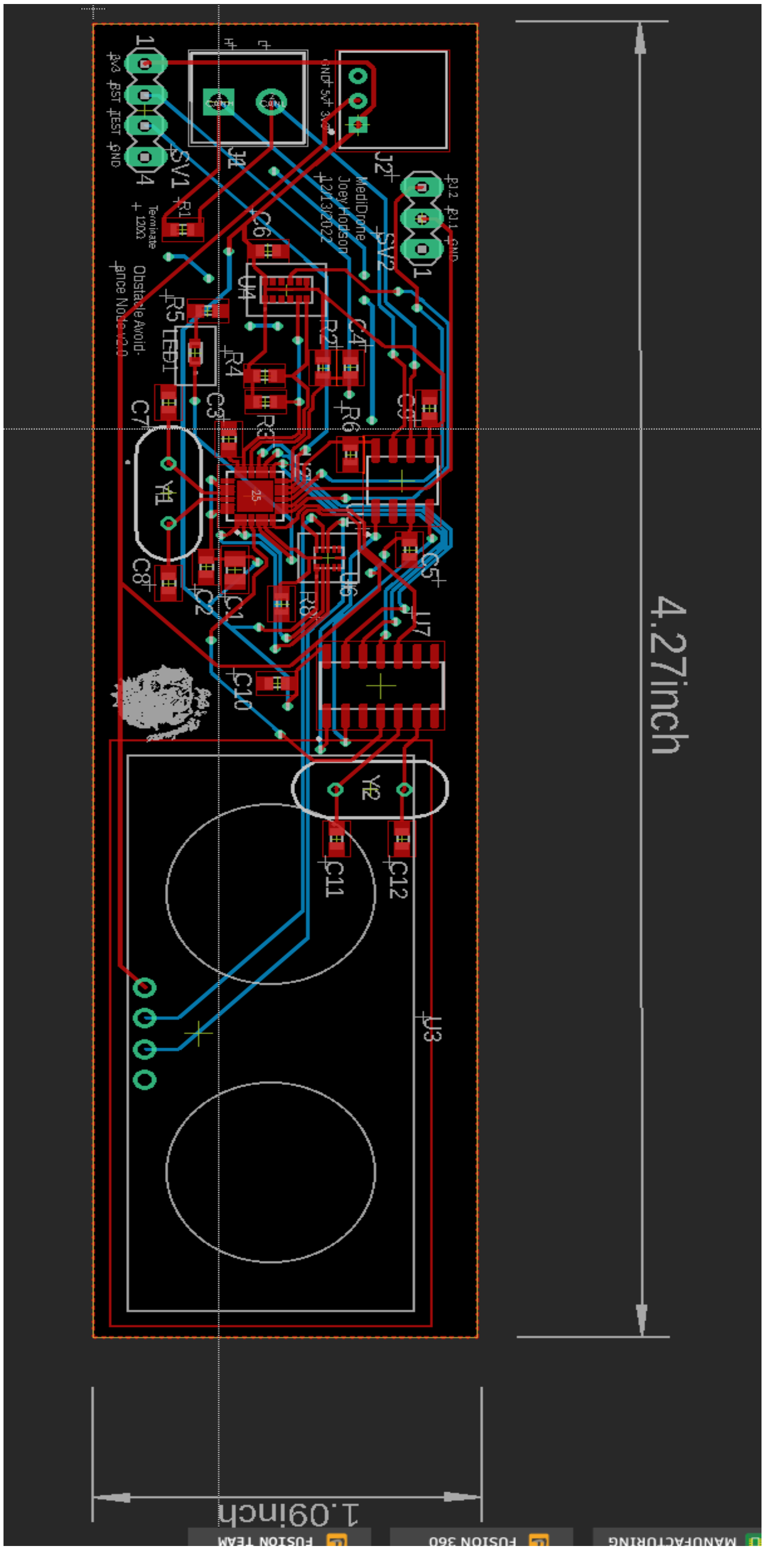
Revision 2 Schematic and Board:

Schematic:



Name: Joey Hodson
Project: MediDrone (Group 28)
Title: Obstacle Avoidance Node Rev2
Date: 12/12/2022
Connectors / Interfaces
Controller
CAN Interface
Sensors
Level Shift
MSP430FR5738IRGER
TJA1042T/3,118
MCP2517FD-H/SL
HC-SR04
TMF8805-1B
NXB0104GU12X

Board:



Final Build:

6.2) Precision Landing

For the MediDrone to be completely autonomous, it must be able to land harmlessly and precisely. When in route, the GPS module will be able to get the drone to within 15 feet of its landing pad, according to the NEO-6m series' documentation. The rest of the work needs to be done with a different system that the drone had embedded into itself and/or potentially within a landing pad's hardware.

6.2.1) Considerations

A lot of autonomous drones utilize computer vision with a camera and distinct target pad that it's trained to recognize. Others use an infrared beacon designed landing pad for the drone to pick up infrared signals from a particular position / direction. Lastly, newer technologies are using Angle of Arrival (AoA) which is now available in certain Bluetooth 5 versions. All these methods are valid considerations, yet some have greater benefits than they do consequences. This will be noted in greater detail in the following few sections.

6.2.1.1) Machine Learning

Potentially the most commonly used method, machine learning offers drones the ability to detect a specific pattern on a landing pad with limited to no hardware on the landing pad side. Also, the hardware for the drone would also be considerably light, not in the sense of physical weight, however, its physical weight too as PiCam's do not weigh very much. However, the major consequence of machine learning is its heavy software implementation. Firstly, the Pi Zero would have to be preloaded with an already trained model to detect the target. The training involved could be strenuous and not time-friendly for MediDrone's timeline as this would be a multi-step process of training to ensure the model would have a high prediction accuracy. Machine learning itself is just another complexity that we, as the developers, are trying to avoid. Another flaw to this method stems from environmental factors. If the environment is foggy, ill-illuminated, or any other harsh condition that impacts the result of an image being taken, the system could fail at much greater rates.

6.2.1.2) Infrared Beacon

Infrared Beacons are another commonly used method to ensure a precise landing of an autonomous drone. The idea here is the most simplistic when compared to both machine learning and Angle of Arrival (AoA). Essentially, a beacon on the ground will be continuously transmitting infrared signals from all sides of the box (beacon). Each diode will be transmitting at slightly different frequencies for the receiver to be able to know what side of the beacon it is receiving from. The drone, carrying the receiver must be able to distinguish each signal from one another. On the top surface of the beacon, the signal sent out will be the key signal for the drone to pick up on. This will be the frequency it knows to slowly lower onto. There are multiple ways to truly design this but the overall concept is just that. We plan to design our own, keeping in mind a good reference like the MarkOne beacon listed below.

6.2.1.2.1) APHD1608SF4CPRV-P22

This IR emitter is under consideration for the design of the beacon. The peak intensity of the diode is at around 880 nm wavelength. It also has a 100° viewing angle, offering a wide range of angle for the photodiode to pick up on. These are priced relatively lightly and the beacon would support upwards of 20 of them, 4 per side. The beacon would house a microcontroller (MSP430) to demultiplex a signal with a varying frequency to each array of diodes per side.

6.2.1.2.2) IR-Lock (MarkOne) Beacon V1.1

The Beacon V1.1, by IR-Lock, is a commonly used beacon to help safely land an autonomous drone. This beacon claims it can transmit to a receiver, with interference, from over 45 feet. This just highlights the strong capabilities of going with an IR-beacon rather than AoA or computer vision. This would also be a desirable beacon to fall back on given time constraints, if need be.

6.2.1.3) Angle of Arrival

Angle of Arrival, or AoA for short, is a relatively new technology introduced in Bluetooth 5.1. AoA is slowly being integrated into aerial vehicles to offer just another technique to land an aerial vehicle precisely and safely. The technology has a lot of other uses but has been shown to be effective in this application specifically. To explain in slightly more detail, AoA relies on a 2-D array of antennas and a Bluetooth receiver to pick up a signal being sent out by a Bluetooth transceiver. Each antenna is multiplexed and read 'simultaneously' to figure out the specific angle the signal is coming in from. The controller, connected with the antennas, is able to fuse all inputs together to form an exact location of the transmitter. For MediDrone's purposes, the landing pad, with said controller, must then transmit a command to the drone, instructing it to move in a certain direction. For more specifics, Bluetooth 5.1 appended the Constant Tone Extension (CTE) to the signals' packet format. This added frame is set at a constant frequency with varying delays. This will happen continuously until the drone has landed safely on the pad. Due to the recency of this technology, it may be somewhat difficult and expensive to implement. However, there is still an option to integrate this technology into the project as it would be very intriguing to do so.

6.2.1.3.1) XPLR-AOA-1 kit

The first thing to consider here is that this kit isn't even accessible to the public quite yet. You must provide an inquiry to even be able to demo the pad and transmitter. So, it's important not to spend too much time on this consideration but to mention that an inquiry was made to U-blox on our plans to use the kit. With that said, this kit contains both a pcb and a transmitter. The transmitter here simply transmits a signal, with a specific delay and frequency in its CTE, to be picked up by the antennas. The 5 antennas are located on the board the kit here provides. More thought and consideration will be put into this kit if U-blox responds to the team's inquiry. That is not to mention the price isn't even publicly available either and may be unreasonable for our desired budget. Other kits like the ISP1907-AOA-DK by Nordic Semi are priced considerably, so this option may be a wipe as a whole.

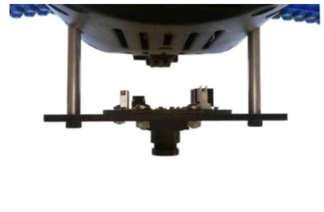
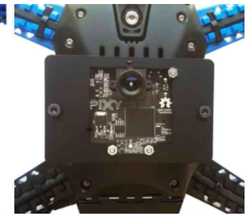
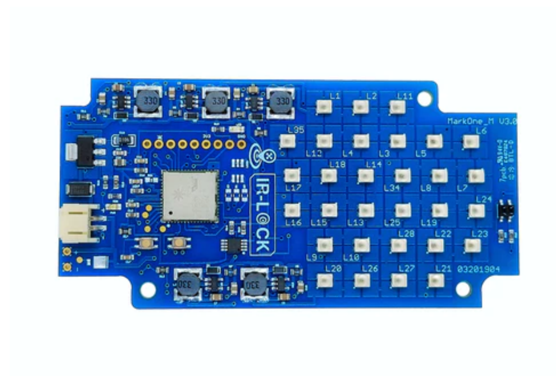
6.2.2) IR Implementation for Precision Landing

Given our finances for this project, as well as availability and abundance of infrared transmitter/receiver technology, it would be optimal for us to research the use of Long Range IR for precision landing of MediDrone.

We will research the feasibility of designing and developing an IR transmitter and receiver for installation on the drone and the landing pad. The IR-LOCK Sensor and MarkOne Beacon is one of the most used precision landing systems for autonomous commercial drone systems; however, the Sensor is listed at $349 USD and the Beacon is listed at $299 USD. These products, though ideal for our project, would be pushing the budget for this project with our finances.

Thankfully, both the IR-LOCK Sensor and MarkOne beacon are open source hardware and have documentation we can analyze to gain a better understanding of how these technologies implement infrared transmission, lidar, and GPS to achieve precision landing.

IR-Lock Sensor Drone Mounted Sensor 3.6mm lens installed for infrared detection Connects to flight controller via I2C Pixy software utilizes GPU for auto-tracking software for IR beacon Operates as drone camera and sensor Wouldn't be feasible for our project as it is way out of our budget, but provides good insight as to the state of	 Cost: $349 https://irlock.com/collections/ir-markers/products/ir-lock-sensor-precision-landing-kit

this technology and how its implemented	  **IR-LOCK Drone Mounted** https://ardupilot.org/copter/docs/precision-landing-with-irlock.html
Precision Landing IR Beacon for Landing Pad Designed for use with IR-LOCK Sensor Dimensions: 10cm × 5 cm × 1 cm 30 Infrared Emitting Diodes Max Detection Range: 15+ meters Voltage input: 10.2-14V (Battery Powered) Has chip for Pulse Width Modulation Placed directly in center of target	 **MarkOne Beacon V3.0** Cost: $299 

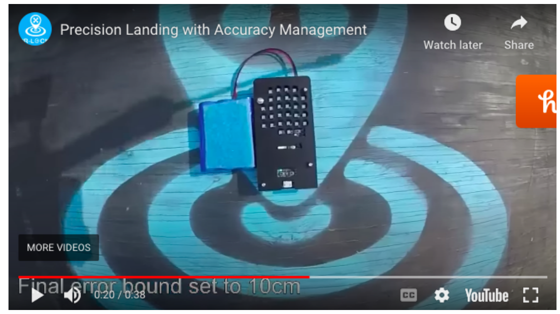
landing zone Would not be feasible for our project as it is way out of our budget, but provides good insight as to the state of this technology and how its implemented	https://irlock.readme.io/docs/2-power-markone-beacon  https://diydrones.com/profiles/blogs/precision-landing-with-accuracy-management

Table 33

The MarkOne Beacon provides great insight into beacon technology. At its core, without its implementation to the IR-LOCK device and Pixy, the MarkOne is simply an infrared beacon with emitting diodes. Though we can't take this exact design, we can very easily implement this simple idea of having a board or boards placed strategically laid out about the landing pad to best home MediDrone into its precise landing position.

6.2.2.1) Optimal Beacon Transmitter System

In order for the MediDrone team to determine the layout of the landing zone, a comparison between use of a single-beacon or multi-beacon system is required to implement the optimal system for landing zone.

6.2.2.1.1) Single-Beacon System

With a single-beacon system, we will have a designated landing zone with a single beacon located directly in the center. To the drone IR sensor, the beacon will be solely locked onto and the drone will guide itself to loiter and descend when it is directly above the beacon. This would be more direct and most likely require less calculation aboard the drone's software, the type of calculations needed will be more elaborated on in 6.2.2.1.2.

A single-beacon system seems to be the most popular system for autonomous IR landing. We will be researching how a multi-beacon system could be implemented, but on initial assumption it appears that the single-beacon will be less complex for software and less demanding for hardware. If the drone camera sensor detects the one beacon it can just home onto that point and make its way directly downwards on it from above. We would only have to power one beacon rather than multiple, and we would not have to set the beacons at an exact distance from each other. As this type of beacon system is more popularly used and researched, it may be more feasible to stick with just one IR board than multiple in our landing zone.

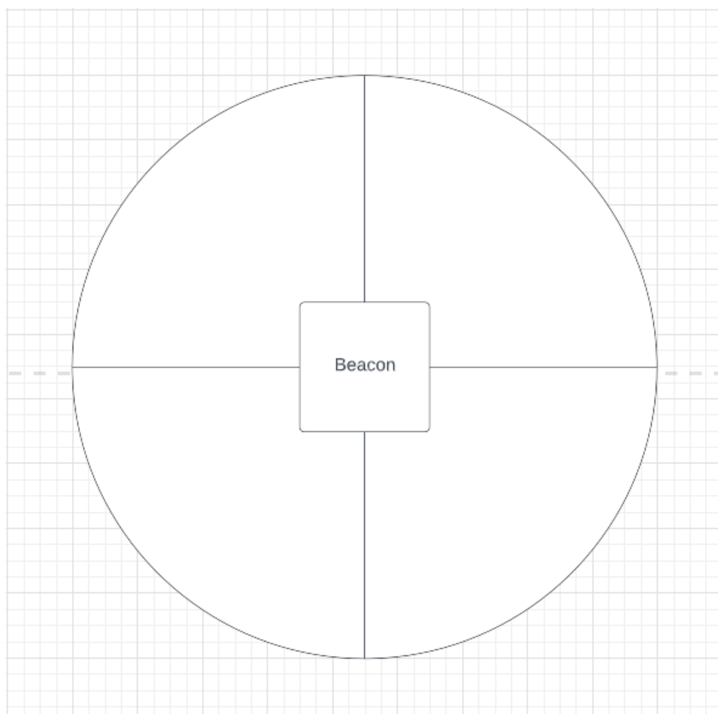


Figure 63

6.2.2.1.2) Multi-Beacon System

Given our lack of a graphic processing unit, we wouldn't be able to correct the drone's landing orientation through methods such as machine learning or visual algorithms. This rules out the idea of using infrared identification. Infrared identification would allow for us to have different shaped infrared beacons, such as an N for North or E for East, on the landing zone for the sensor to recognize as cardinal coordinates; however, with GPS and a built-in compass we could utilize these to orient the drone upon descent. With these aspects we don’t have the need to analyze how to determine which beacon is which in a multi-beacon system.

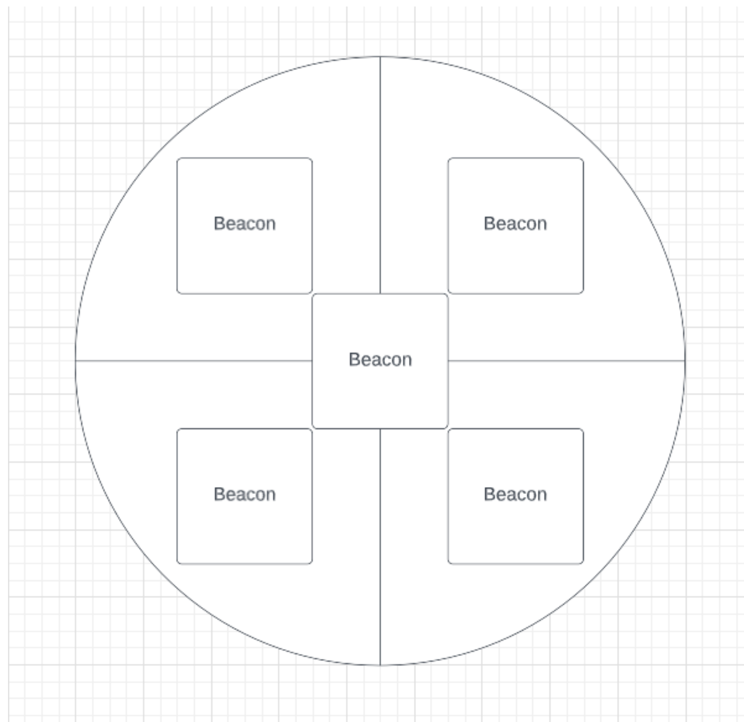


Figure 64

If the multiple-beacon system proves to be a feasible design, then we will calculate the specific angles and distances of this procedure. Depending on how high the drone is in the air, how closely together the beacons are placed next to each other, and the boresight angle when the drone is loitering above the landing zone, the beacons at will seemingly converge to be a singular spot of light. As the drone descends, the beacons, according to the camera, will seemingly begin to separate. This is similar to the Screen Door Effect, where as you approach a screen or monitor you can start to make out the individual pixels of the display. WIth multiple beacons, this effect will take place to hone the drone to the singular beacon, then as it descends the beacons will separate and the drone can use these multiple beacons as a way to triangulate landing. Triangulation sounds to be a feasible design goal for the infrared landing. The distance between the beacons and the boresight angles to the drone will ideally assist the drone in landing precisely in the target area.

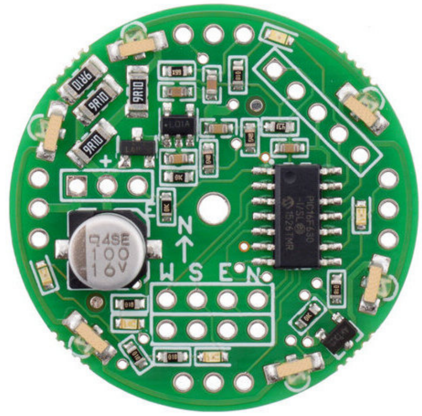
To enable a multiple-beacon system, software adjustments will have to be made; however, hardware adjustments won't need to be. Essentially, if a single-beacon system proves to be just as feasible, then the only adjustment we'll have to make is hardware-wise. We will be making multiple beacons to test with in the event of hardware failure anyways. So, it would be implemented just as simply as a single-beacon system would be.

6.2.2.1.3) Beacon Board Analysis

To better understand the optimal design for the beacon's board, we will analyze board models products for IR beacons.

Pololu IR Beacon Development Board This board is made for having two robots play a game of tag by transmitting and detecting each other, so that one can chase and the other can evade. Circular Model 1.35" diameter Detection Range: 6 inches to 15 feet IR modulation frequency 56 kHz Operating voltage: 6 - 16 V IR Emitters: 6 IR detectors: 4	

<table>
<tr>
<td>This board acts as both a transmitter and as a receiver. The beacons switch between receiving and transmitting IR signals so that it doesn’t confuse reflections with their transmissions.

The design for transmitting may be analyzed more later but as of now we are not interested in having the beacon do any receiving of signals.</td>
<td>

https://www.pololu.com/product/704</td>
</tr>
<tr>
<td>AiM IR Beacon Transmitter

This beacon is made for tracking times between laps for kart races. It is placed at the start/finish line of the race track and transmits to the driver the time between each crossing of the line.

Ideal for close range transmission where GPS isn’t accurate

Dimensions:
4.84 x 76.1 x 1.85 inches

IR Emitters: 6

Low Frequency:
battery powered with four AA batteries
At Low Frequency detection range up to 30 feet
High Frequency:
external wiring harness and 12V lipo battery
At High Frequency detection range up to 60 feet</td>
<td>
</td>
</tr>
</table>

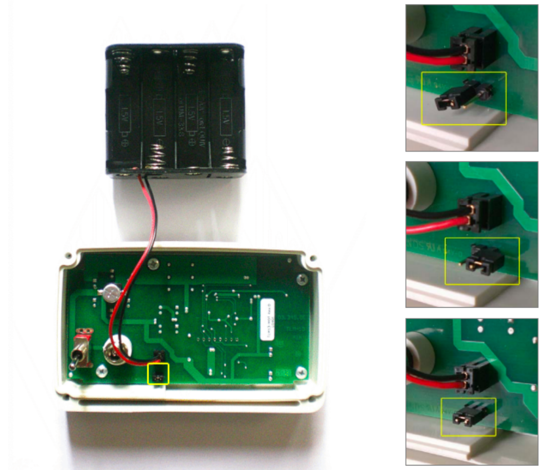
Sends coded/modulated IR light signal via IR-LEDS The signals are composed of blinking light-on/light-off phases, the times and frequency between light pulses is determined and synced with the receiver so the receiver can confirm that the IR light it is detecting is the beacon's signal and not interference This type of IR transmitter would be ideal for Medidrone's IR beacon as it demonstrates how PWM is a necessary function for our beacon.	 https://ucexperiment.wordpress.com/2012/02/21/aim-ir-beacon-transmitter-substitute/ https://aimlive.com/wp-content/uploads/Optical-Lap-Transmitter-Data-Sheet.pdf

Table 34

6.2.2.1.4) Beacon Board Design

A transmitter beacon board's design can range from simple to complex. A transmitter beacon board in its simplest design would be a board fixed with infrared LED's and a power source. For the time being, we will just be designing the board to be turned on when the power source, being a battery, is connected to the circuit or implementing a switch. The switch would be flipped on when the drone is approaching its destination. This action would connect the circuit of the LED's to the power source and the LED's would stay on continuously until it is turned off again. A stretch goal for this design could be to implement a way to remotely flip the switch, such as via RFID, rather than just an analog toggle switch.

The ultimate goal for the beacon's design would be to have:

- the beacon be remotely turned on by a switch via RFID
- Integrated circuit mounted to the board can be programmed to have PWM at a desired frequency
- Infrared LED's flash at modulated frequency
- Once drone lands, beacon is remotely turned off

Figure 65

Because of interference from reflections of IR light from surfaces such as glass and/or water, it's imperative that we look into ways of filtering out reflected IR light in order to prevent the sensor mounted to the drone to not be confused as to where the landing site is.

A method of filtering out interfering IR signals is to modulate the frequency of the IR signals coming from our IR beacon. By modulating the frequency of the signals, we can program the IR receiver mounted on the drone to seek and only lock onto a specific, consistent frequency coming from the beacon.

The use of Pulse Width Modulation (PWM) would be the most efficient technique of modulation for our system. The process of converting PWM from an analog signal to be processed digitally aboard the drone would be the most reliable way to ensure that the signal is processed accurately and precisely.

PWM can be easily achievable by using a microcontroller, such as an Arduino or Raspberry Pi; however, we are going to be designing our own PCB to make up our IR beacon. And so, instead of using a microcontroller, we are turning our attention towards integrated circuits that can generate accurate time delays. The IC that seemingly has the most potential to fit with our goal of PWM-infrared beacon is the LM555 Timer.

6.2.2.1.4.1) LM555 Timer

The LM555 timer is an integrated circuit from Texas Instruments. It is constructed of 8-pins connecting together a system of comparators and a flip-flop. These components ensure a steady and consistent PWM for signals that will be pulsed at a duty cycle of 50%. This kind of operating mode for the timer is called an astable operation. Below is a schematic of how the LM555 timer could be used to pulse an IR LED by building it to operate in an astable mode.

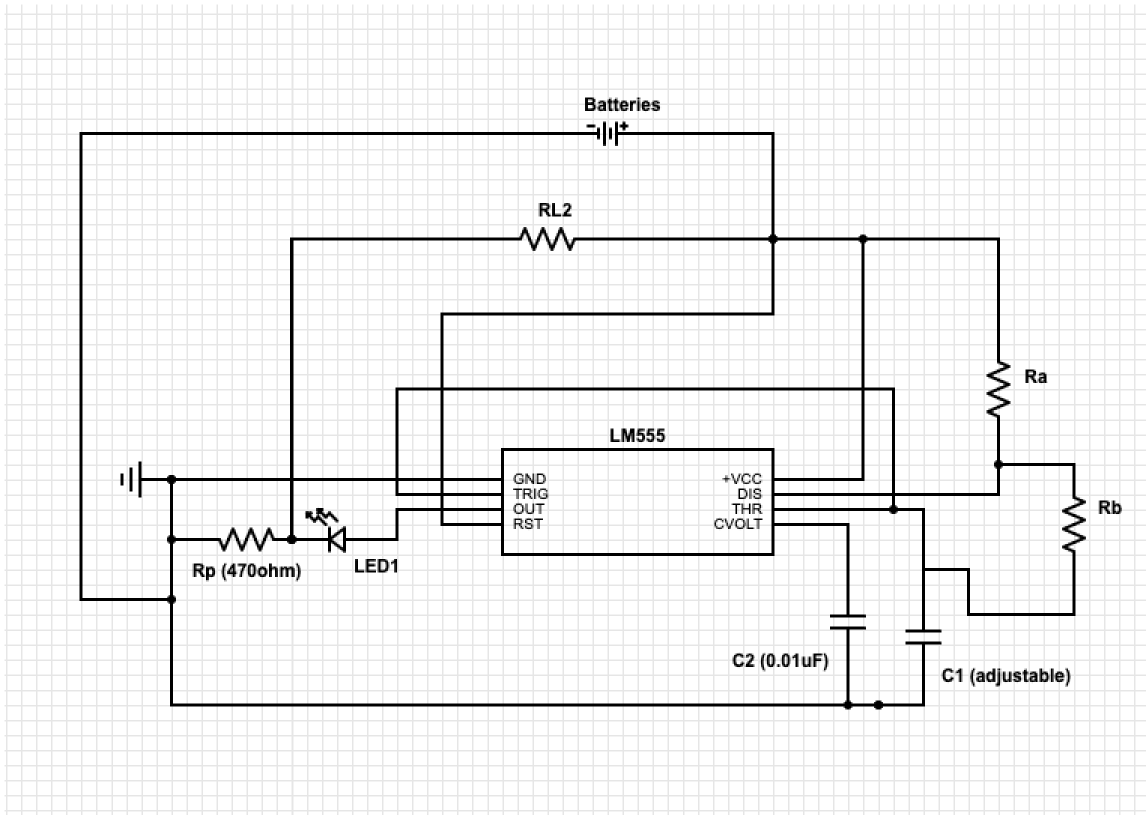


Figure 66

The frequency is determined by the relationship of capacitance over time delay. The charge/discharge time is determined by the relationship of the resistors. The equations for these are given in the data sheet for the LM555.

Output High:

$$t_1 = 0.693(Ra + Rb) * C$$

Output Low:

$$t_2 = 0.693(Rb) * C$$

Total Period:

$$T = t_1 + t_2 = 0.693(Ra + 2Rb) * C$$

Frequency of Oscillation:

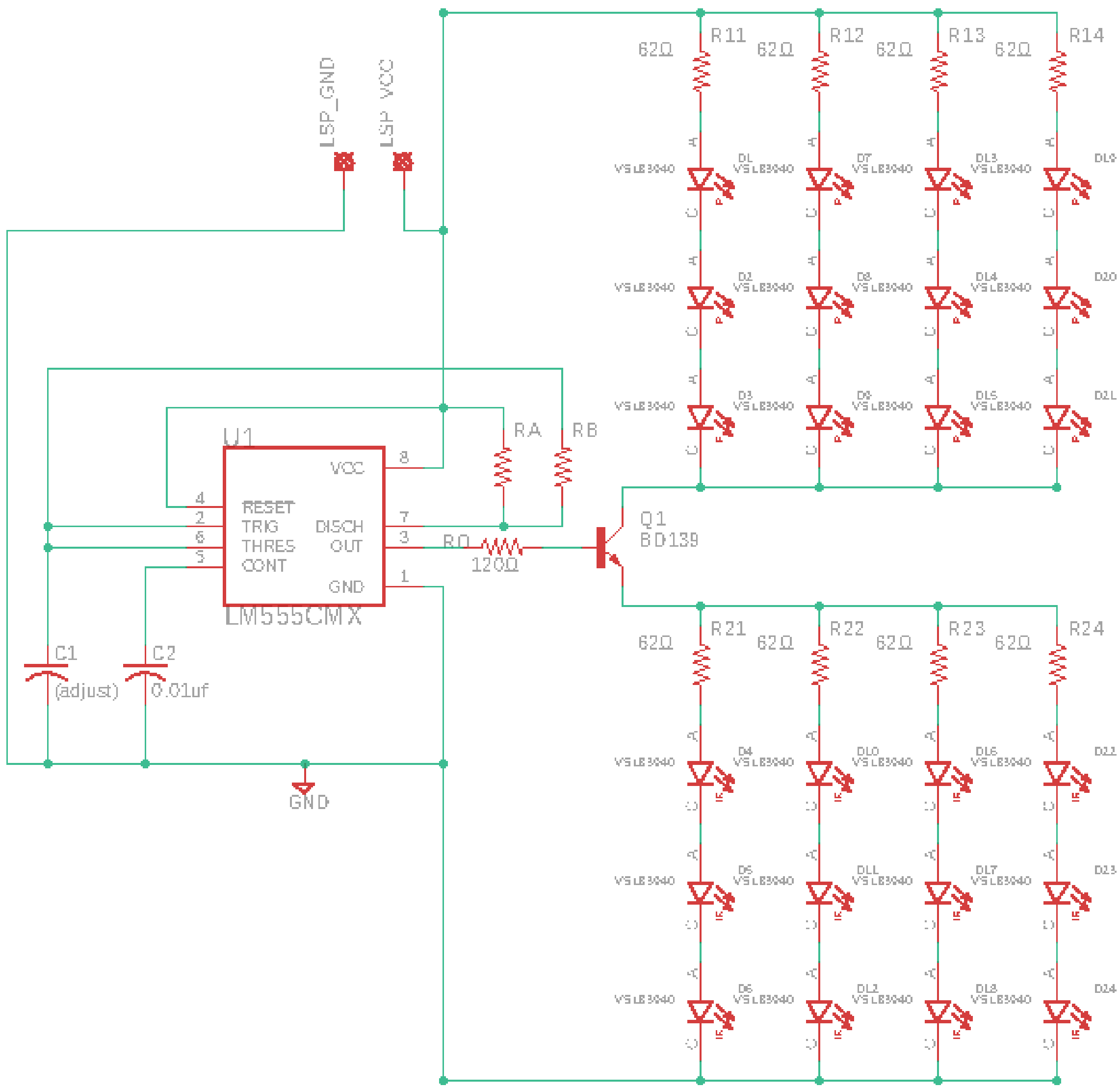
$$f = 1/T = 1.44 / [(Ra + 2Rb) * C]$$

Duty Cycle Equation:

$D = Rb / (Ra + 2Rb)$

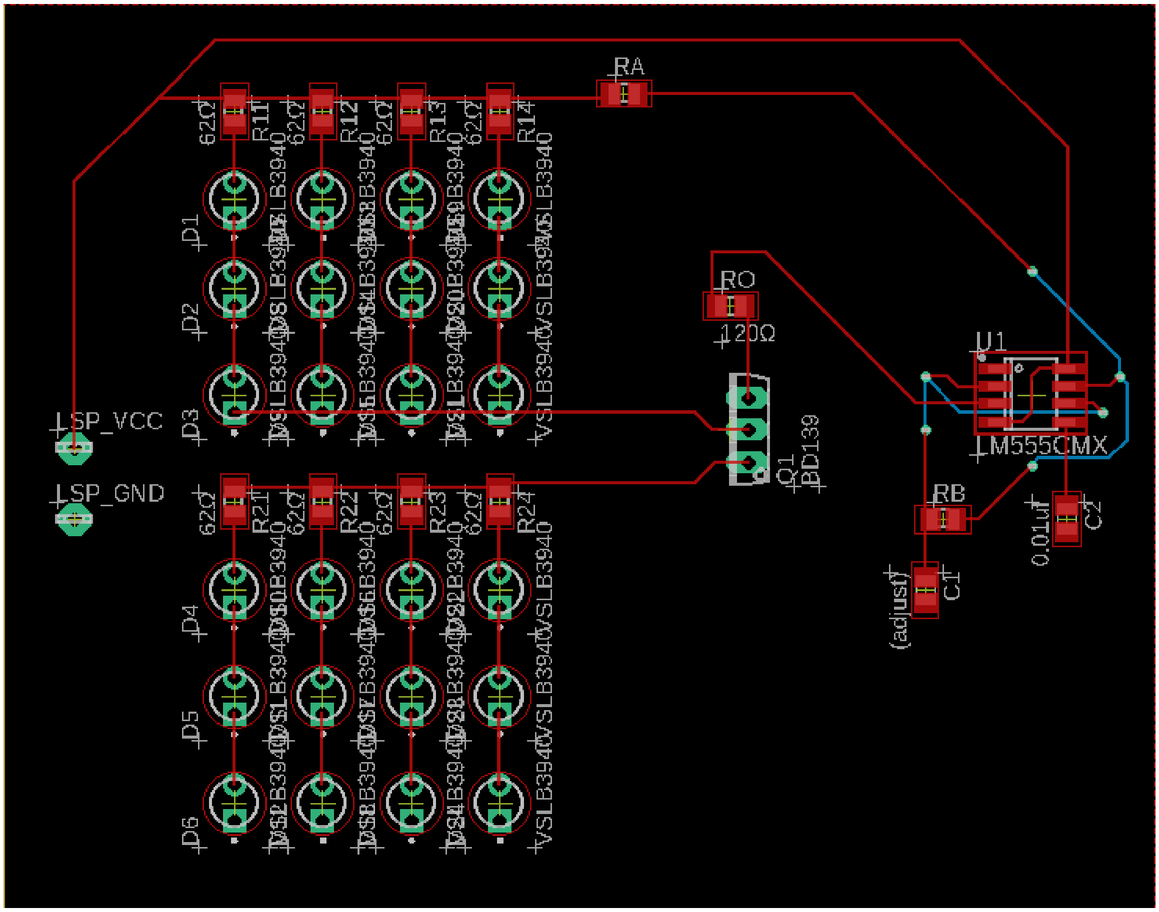
We'll follow this astable operating mode build and adjust the frequency to whichever frequency will work with the receiver sensor mounted for the drone. The supplied voltage through the battery pack will be adjusted for the number of LEDs connected to the output. The IR-LEDs we will be testing with are the ADA388 5mm IR LEDs. These have a wavelength of 940nm, which is a standard amount for IR LED wavelength. We will test on a breadboard until we reach a beacon we believe will work, then we'll order a two-layer PCB.

We researched the most optimal frequency for the precision beacon and determined that 38kHz would be best for filtering out interference and natural noise from sunlight reflection. A duty-cycle of 50% is optimal for having an equal and stable pulse. Using the schematic for astable mode, we designed a beacon board composed of two arrays of IR-LEDs. Using our research from the Wii remote, we made a design where the two arrays will appear separate from up-close, and will appear converged from a distance of about 6 feet. This will allow for the drone's camera to see a single beacon while descending, and then will be able to use the separate arrays to determine landing orientation.



For the input voltage we will be using a 9V battery. Using equations from the datasheet, we calculated the values of the resistors, Ra and Rb, and capacitor, C2, to determine the optimal frequency for our IR LED's to be detectable on a sunny Florida day. Ra valued at 68ohms, Rb at 1.8kohms, and both capacitors at 0.01uF.

The input voltage will go into the top LED array composed of four columns with three IR-LEDs. Once the currents go through the LEDs in the collector of the transistor, the output current from the LM555 timer will be sent through the transistor base. These currents will then add together to make the current going through the emitter. This current going from the emitter will then pass into the bottom LED array, then to ground. This timer layout allows for the arrays to alternate flashing. We then designed the PCB with this schematic:



With our breadboard testing being successful, we sent the GBR file we made on Eagle to JLCPCB to have it printed. With the five pcb's that are shipping in, we used different resistor and capacitor values to achieve different frequencies. We then tested with the receiver which frequency is more accurate for the sensor to detect. We also designed and 3D printed an enclosure for the beacon, to shield the electronics and exhibit the IRLED's.

6.2.2.2) Optimal IR Receiver System for Drone

For the IR receiver sensor, we will need to have a sensor that picks up on infrared signals and can process these signals as data to send to the microcontroller to communicate to the flight controller. To understand how an IR sensor would receive these signals, we shall analyze existing applications of IR receivers.

6.2.2.2.1) Wii Remote

The Wii's infrared beacon is a thin bar that's to be placed at the top or bottom of the monitor/screen to which the console is connected. This beacon has a row at each end of the bar composed of five infrared light emitting diodes, IR-LEDs. The IR-LEDs are positioned in a parabolic fashion. Thus this positioning allows for a wider cone. So, the IR-LEDs on the ends are angled outwardly, and as it goes down the row towards the center, each IR-LED is then angled more inward towards the center of the bar. The optical sensor located on the top face of the Wii Remote is used as a reader of the infrared beacon.

The optical sensor on the top of the Wii Remote is made of a simple thermal camera with connections via data lines to the controller built into the Wii Remote. The optical sensor registers infrared light. When using the Wii Remote for a Wii game, the end of the remote with the camera is facing towards the beacon. The beacon recognizes the two rows of light, which are clumped closely together, as two points of IR rays. The camera then uses the angles of the points to determine which direction to move the cursor on the screen. If the camera registers the point on the right side of the beacon to be above the point on the left side, the controller will calculate the angle between the two points to register the tilt of the controller and adjust the on screen cursor.

The Wii Remote can be a distance of six to sixteen feet away from the beacon to register the angles between the two IR points and the camera. This range is the focal point of the sensor relative to the beacon. In this range, the sensor works optimally to accurately register the cursor's movements relative to the beacon's placement. At a distance less than six feet, the sensor has trouble seeing both beams of the IR LEDs and/or can see the individual IR LEDs that make up each of the two rows. At a distance of greater than sixteen feet, the sensor will then begin to clump the two IR beams as one, similar to the effect of the LEDs up close and how the two beams make for two sources of light for the sensor to detect.

The sensor can still detect IR rays coming from the beacon. This is what matters most to the scope of what we want the sensor to perform. With a single beacon system, the sensor would just need to detect the signal and the dry will fly towards it. With the multi beacon system, the sensor would detect the signal as a single source until it reaches the focal point. Once it reaches the focal point the sources will begin to diverge and the sensor can register these points and the flight controller will triangulate itself to land towards the center. Once the sensor closes in on the center beacon, it will lose sight of the beacons around the center.

6.2.2.2.2) IMX462-99/127 IR-CUT Starlight

This sensor captures high-quality imaging in visible light as well as near-infrared (NIR) band. NIR infrared is the wavelength of infrared we are hoping to detect, as it is above visible light but also less than 1000 nm. Its field of view (FOV) is at 127.9 degrees, which means it will be able to take in all rays from the beacons until very close to the center point of the landing zone. It's compatible with our microcontroller through transmitting data and clock lines via I2C. Its frames refresh at 25fps, which seems to be a value that would work well with how the drone will be descending. With resolution of 1920 x 1080, it could offer clear cut imaging on the IR source. Priced at $15.90.

6.2.2.2.3) Pixy2.1 Smart Vision Sensor

This sensor works much like the IR-Lock we analyzed in 6.2.2). The Pixy2.1 is the smaller yet faster version of it. The Pixy2.1 can be programmed with infrared detecting libraries and sends data via I2C to the microcontroller. It has an FOV of 60 horizontal x 40 vertical degrees. The Pixy2 detects color by frequency and wavelength, which would be perfect for our goal for the sensor. It consumes 100 mA typical and requires 5V input. It does have an IR-Cut filter which will allow for crisp infrared detection. Can detect objects up to 70 meters away. Priced at $69.90

6.2.3) Testing

Aboard Medidrone, we will have equipped an ArduCam b0205. The objective of the camera is to detect the PWM signal from the IR beacon, positioned on the landing pad, from a distance between 10-20 meters (depending on the day's weather conditions). We found a good deal on the ArduCam b0205 as well as libraries that we were able to use for infrared PWM detection.

With the breadboard model working well, and the PCB assembled we conducted testing with the software.

6.3) Power Distribution

- **Overview**
- **Types of PDB's**
- **MediDrone Power Capacity**
- **Design or Buy**

All hardware included on MediDrone must be carefully and reliably powered. This section will outline methods of power distribution from Lipo batteries. Lipo batteries must be able to supply sufficient and constant current to all peripherals in a regulated manner. A Power Distribution Board (PDB) is required in order to regulate the power delivered from the positive and negative leads of the battery to peripherals including but not limited to motors, sensors, communications, and other powered systems. Power is provided to motors through Electronic Speed Controller's (ESC's); the flight controller board is directly powered through the power distribution board and delivers power to its peripherals including communications, GPS, and various sensors. The Power Distribution Board may also directly power a camera such as a GoPro.

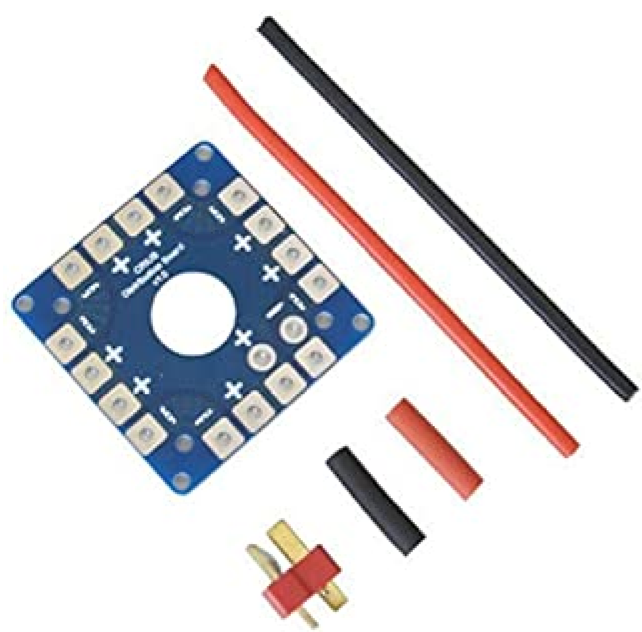
Power Distribution Boards preferable to the MediDrone project will, at the very least, consist of lead input from battery, and at least four ESC (electric speed controller) output leads, voltage regulators (for peripherals FPV and VTx), and motor connection pads supporting signal exchange for motors and communication between flight controller and ESC's individually. Design versus buy is a very important factor for the success of the power distribution board.

Types of Available Power Distribution Boards (PDB's)

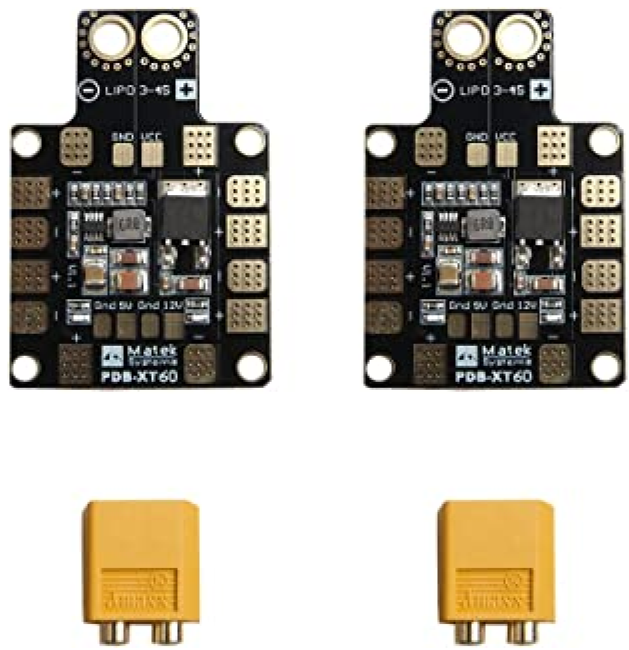
A PDB must be chosen based off size, layout, and capacity (voltage and current), below are general available PDB types and configurations. Available PDB's may include integrated flight controller, and/or integrated current sensors. PDBs also may vary in the amount of ESC's supported, modern designs have allowed PDB's to include 4 ESC lead sets on the same board, fully supporting a quad copter's speed controllers and motors. MediDrone may not purchase an integrated flight controller board due to the ability to design a flight controller in-house. Using a pre-designed and bought PDB which is capable of delivering power 4 ESC's gives the main constraint of taking choice away from the amount of motors, with this configuration, MediDrone will be a quad-copter. Other ideas involving a hex-copter or octo-copter will effectively be muted, however, MediDrone plans on building a quad-copter, minimizing this constraint.

For any reason, if MediDrone wanted to support a hex-copter, six-axis PDB's exist and are available. Six-axis PDB's are not as common as four-axis PDB's however there is an example in the figure below of a PDB capable of being used for a quad or hex copter style of UAV.

Many PDBs offered may include a built in flight controller (FC), these are priced higher than PDB's which do not offer flight controller as part of the same board, or stacked boards. Observed are PDBs including flight control function from $75+ although these would introduce their own constraints or require the use of a second PDB or FC if more functions, outside of the predetermined scope of a purchased PDB/FC combo, are required. .

Vendor Available PDB Variety	
Simple PDB:	**daier 2Pcs ESC Speed Controller Power Distribution Board PDB for Quadcopter Multirotor 3-8**  $8.99

<table>
<tr><td colspan="2">The daier ESC may work however due to its low pricing and apparent low quality, this is not an ideal choice as a PDB for MediDrone. This type of PDB does not include 5v and 3.3V for peripherals, however, for an inexpensive backup, it is a good idea to either experiment with these or outfit one with additional specification.</td></tr>
<tr><td>Moderate PDB:

SD2 Used this PDB</td><td>Acxico 1 pcs Drone Power Distribution Board XT60 3-4S 9-18V 5V 12V Output PDB


$11.79</td></tr>
<tr><td colspan="2">The Acxico PDB seems to offer the bare bones of what MediDrone requires, being supply to four speed controllers, 3.3V and 5V pin DC supply. Due to low price, these may be lower quality and can fail, which would fail mediDrone. This product can serve as the PDB for MediDrone.</td></tr>
<tr><td>Complex PDB:</td><td>TAROT TL2996 High Current Distribution Board Power Distribution Management Module 12S 480A for DIY 4-axis / 6-axis Drone Kit</td></tr>
</table>

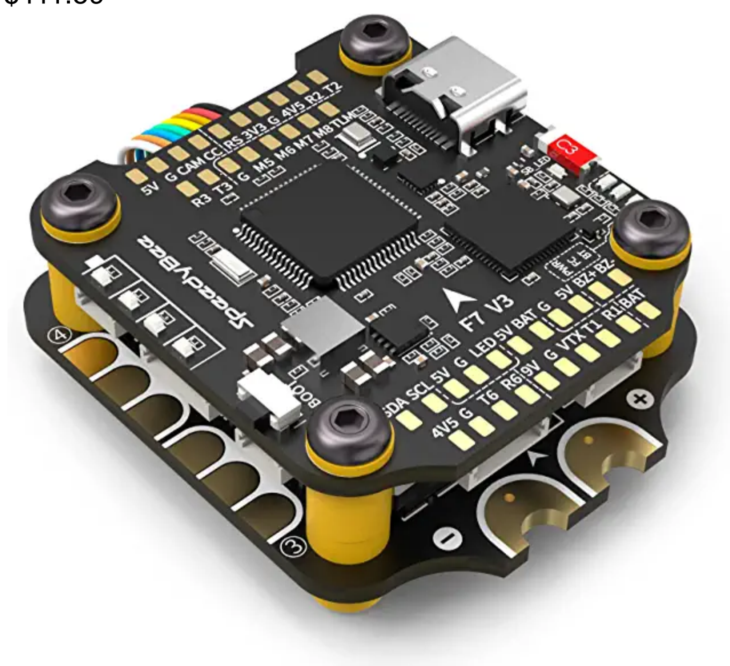
	 $61.98 *PDB and Flight Controller Stack* **SpeedyBee V3 F7 Flight Controller Stack: 30x30 Drone FC Stack with 4in1 50A ESC BL32, Wireless Betaflight** $111.59 
The above two products include flight controllers as well as power distribution. These are unnecessary for Medidrone, as a flight controller is being designed in-house. If the in-house flight controller proves incapable, these are good options to use.	

Table 35

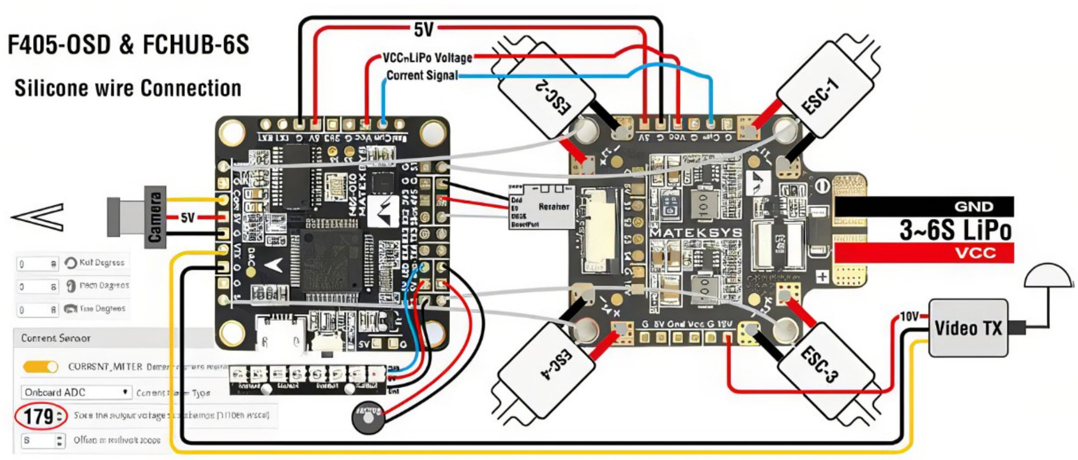


Figure 67

Example system shown [Simple Drone Circuit Diagram - Drone HD Wallpaper Regimage.Org]()

MediDrone will be designing a flight controller, eliminating the need for a flight controller integrated power distribution board or versa a flight controller with built in power distribution board is unnecessary for MediDrone with current scope. Above shows a flight controller with separate power distribution board in a system whereas other systems displayed further above are integrated together on the same printed circuit board.

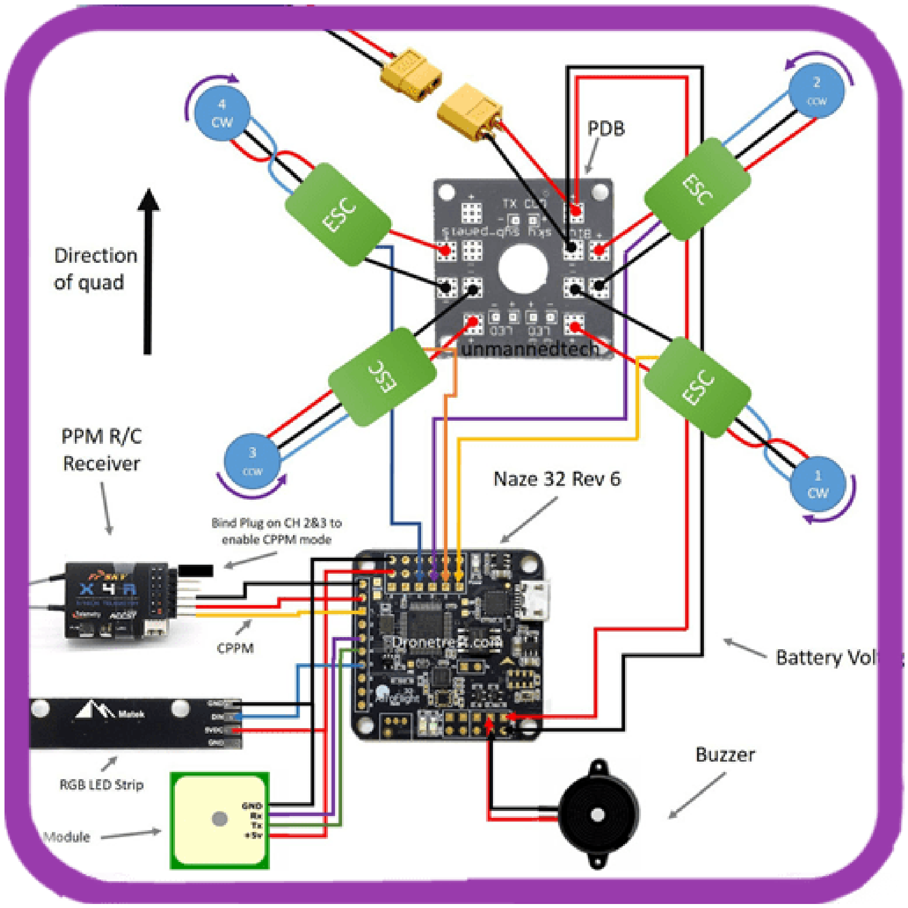


Figure 68

These images serve as functional diagrams showing power distribution board use with flight controller board and peripherals in a graphic superimposed over board images. The PDB's shown support four flight controllers and other peripherals. The interaction between the PDB and the flight controller (FC) is shown above, the relationship summed as the PDB supplying power to the ESC's and FC whereas the FC supplies voltage signals to ESC's and the PDB supplies power to the FC.

<u>MediDrone PDB Necessities</u>

Each electric speed controller will pull at least 20A, for a quadcopter, instantaneously 80A current will be needed to supply four ESC's with maximum rated power. The motors will not each be pulling 20A at all times, however the PDB must be able to deliver max rated, and specified currents from the LiPo battery.

Speed controller with a higher current rating than 20A can also be used for the MediDrone. Higher rated controllers include but are not limited to 30A, 40A, 50A, or 60A. 30A may serve to be more beneficial for MediDrone in order to carry a payload with more weight such as blood versus a medical payload of less weight such as a vaccine vial. With higher rated current comes further drain on the LiPo batteries, serving as a constraint toward using controllers which draw more current. If the MediDrone were to use each controller at 30A or 40A, four speed controllers would draw, 120A and 160A respectively if current drawn reached max rating from the controllers.

Most peripherals, to be powered through the flight controller or separate 5V bus, will operate properly on much lower current than motors require. The flight controller itself, including peripherals should pull around or less than 3 Amps including all modules

PDB Peripheral Type	**Power Specification**
Electric Speed Controller Min (including motor)	20A*4= 80A
	30A*4= 120A
	40A*4=160A
	60A*4= 240A
Flight Controller	DC 5V=<3A
Peripherals on 3.3V	>1A
Other Peripherals on 5V	>1A
*Note most peripherals, communications, and powered devices are powered through the flight controller, therefore, flight controller encompasses the power delivery of some sensors, RF modules, and processors, 3.3V and 5V pins will also be available on the PDB for these peripherals to receive power if not from flight controller..	

Table 36

<u>Design Considerations</u>

To design a power distribution board, programs can be used such as Autodesk Eagle, KiCad, or Altium. The process of designing a printed circuit board includes the schematic and the board saved in .sch and .brd file types specifically. Designing also requires assembling components within a bill of materials (BOM) generated by the design program specifically to component types utilized in the project. Surface mount type components will be used on a PCB designed in-house to be reflowed using hot air and solder paste. The constraints attached to this process include design or technical flaws in either the program itself, or physically implementing the design in the laboratory. Designing a PDB allows for further customization of a long bus for peripherals to be delivered power from.

Schematic design encompasses choosing the proper step down chip per datasheet, choosing the correct library derived symbols, drawing traces, and testing simulation. Power electronic design is very sensitive, and highly skilled form of design, encompassing heavy weight on decisions around component choice.

The board design, saved as file type .brd in Autodesk Eagle allows for the board to be modeled physically around the schematics. Component footprints derived from libraries and the internet must be placed on the board design in a simple but effective form allowing the assembler leeway rather than difficulty. Footprints will most likely be for surface mount components to be hot air reflowed to the board. The components must be named in such a way that, if printed on the board, the assembler can decipher with a legend or notes, to ensure a proper build.

The final board can either be printed through a printed circuit board “boardhouse” such as JLCPCB. University of Central Florida has also invested in a PCB milling machine and reflow oven, available for use by senior design participants. This is the in-house method but also a riskier method. A PCB house will be effective, but limited in shipping time and operations around the December time period.

LiPo Battery Connectors XT60 Connectors Were Used

Battery connectors are chosen based on the max rated current to be drawn from the battery. See above photographs including LiPo batteries, connectors between batteries and the PDB must be rated to pull the current of the entire drone system before the power is distributed to specific peripherals which make up the drone system.

Anderson power pole connectors would serve decently to connect power from LiPo battery pack to the PDB. This connector type is mostly used for powering amateur radio systems which draw high amounts of current. Features of anderson power poled include a very distinct click to lock the connectors together, these are fairly easy to purchase and install as well. Anderson power poles mainly come in 15, 30, and 45 Amp variations, less than what is required in our system. These can not be used on the battery connector to the PDB as they are not rated high enough, however if connectors from the PDB to the speed controllers need to be replaced or fabricated, they would be ideal between the PDB and ESC's due to their tight locking ability while maintaining electric contact.

XT150 connectors are available for theoretical use for power supply to the MediDrone as the highly powered MediDrone required connectors of a large current rating. These connectors feature a load capacity of 150 Amps, necessary to provide at least 30A to four electric speed controllers. This type of connector includes a 6mm pin, already compatible with a standard LiPo connector.

AS150 Connectors carry the capacity of running a max of 200 Amps constantly, these would be the most ideal connectors in terms of current rating. Main constraint of using AS150 connectors is a modification required on the Lipo battery for a 7mm pin rather than a 6mm pin.

Final Design:

When designing an aerial vehicle such as a drone, it is crucial to optimize for weight and range. MediDrone is equipped with Lithium-Polymer (LiPo) cells, as current battery technologies permit. The drone has employed 4-series, 14.8v batteries with two different charges - 7.2Ah and 1.5Ah, to meet the aforementioned consideration. The reason for having different charges is to optimize for range and weight. Initially, the 7.2Ah battery seemed to be excessively heavy, making it challenging for the drone to lift against the thrust of its motors. On the other hand, the 1.5Ah cells weighed around 6 ounces together, which is slightly more than one-fourth of the weight of the bigger battery. With this battery, the drone lifted off the ground much more smoothly, in line with the expectation for a drone. The lighter weight of the drone also appeared to enhance flight stability. We finalized the drone with a 6s, 3.3Ah battery.

Each motor is controlled by a 40-amp ESC. Therefore, the power supply needed to accommodate this. Not only the power supply, but the max-continuous battery current discharge as well. The 1.5Ah battery was rated for 120C as its charging/discharging rate. 1.5 x 120 provided a continuous discharge of about 180 amps. This results in a leftover 20 amps for the rest of the electronics on the drone, this being more than sufficient. The 7.2Ah battery had a similar rating and obviously could discharge the current that the motors sank.

The power distribution board allowed for 4 channels, each with 40 amp traces. This was ideal. It is also worth noting that every power connection in the system is done through XT60 connectors. The 60 implies the max amperage, this being 60 amps. This amount of continuous current is sufficient for every connection. The 5v regulator onboard the PDB could only source 2.5 amps. This was concerning early on as, after some calculations, it appeared the total electronics would sink roughly 3 amps total. After some findings, each of the four obstacle avoidance boards, on average, pulled under 100 milliamps continuously. The flight controller and beacon receiver did not pull near 2 amps combined; thus, the regulator was sufficient. These findings imply that, if more hardware were to be added, like another CAN node, then it may be best to either swap the regulator on the distribution board, design a new distribution board, or to find a more suitable replacement.

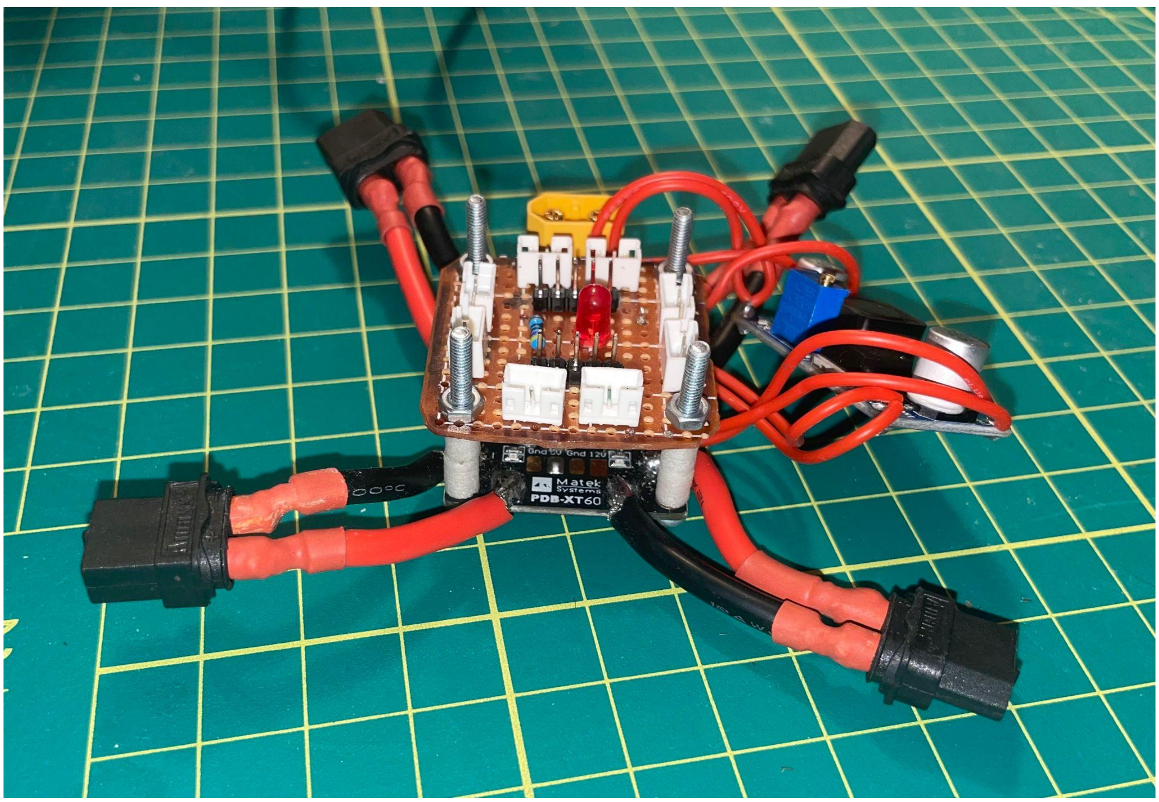


Figure: Final Power Distribution Iteration

6.4) Battery Pack 6s Batteries Used

The battery pack is one of, if not the most, critical parts of the drone. This is because having a stable power source that is able to supply the drone's power demands is critical to its function. The battery must provide power at the voltage required by the motors and supply the peak current that the motors draw, so that the motors can perform as expected. Battery packs are constructed from individual battery cells connected in series or parallel *see battery configurations below*. Safety is a very serious factor in which battery pack to use or construct in-house. When handling battery packs, MediDrone team will check cells for damages before, during, and after use. Charging and discharging of these cells must be done to standard practice in order to maintain their safety and prevent battery cell damage from occuring. Battery cell damage is very hazardous to the battery pack an the personnel involved. Proper PPE will be worn at all times work is being done toward a battery pack including but not limited to protective eye wear and limiting loose articles of clothing. Proper safety practices will be objective and paramount when working with battery packs as PPE will not be very effective if improper electrical or handling practices occur.

Figure: Final Battery Iteration

6.4.1) Battery Chemistry

The battery chemistry type is what determines the exact physical mechanism of how the battery cell works. Thus, cells are typically grouped by their chemistry since batteries of a certain chemistry typically have similar parameters and characteristics. Each battery chemistry generally determines the nominal voltage, peak current, capacity, weight, size, change-discharge profile, and thermal characteristics of each cell, which are used to determine the remaining parameters of the pack. The specific battery cell chosen from this family can slightly adjust these values, but overall, they are largely determined by the chemistry. The most common battery chemistries used today are lead acid, nickel-cadmium, nickel-metal hydride, lithium-ion, and lithium-polymer. Figure X shows a comparison of energy density by volume vs. energy density by mass for each of these battery chemistries.

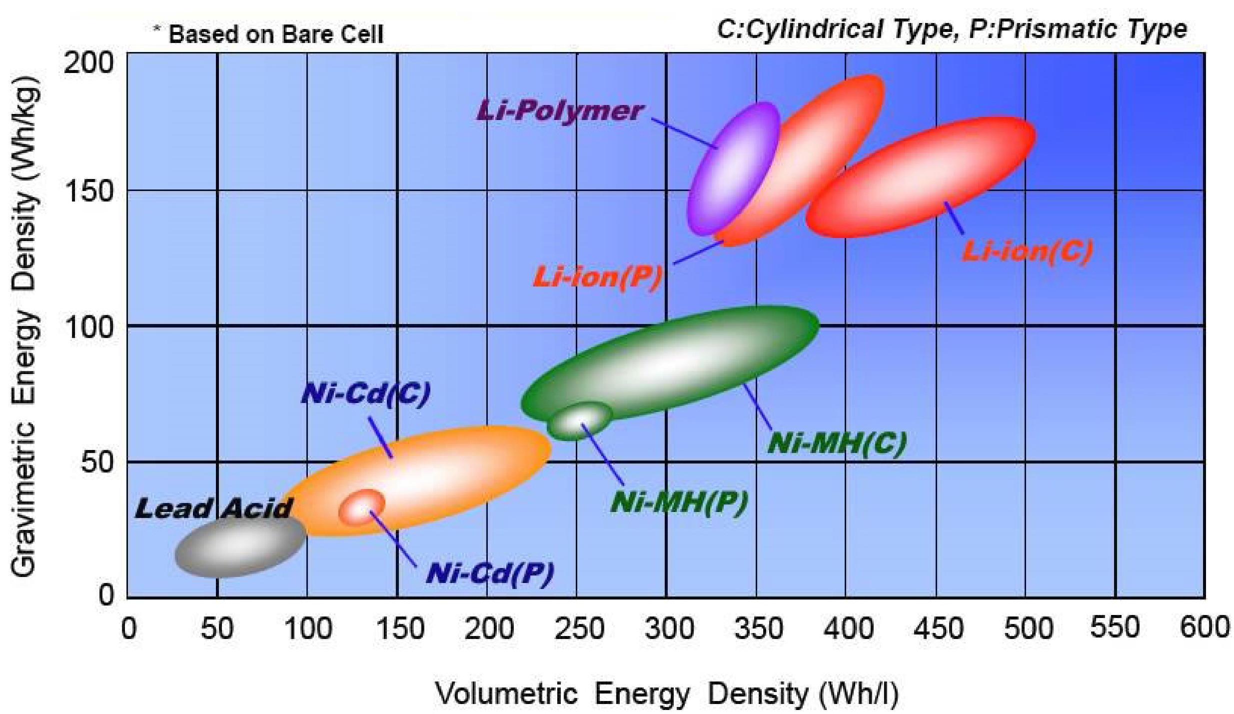


Figure 69: *Comparison of energy densities for various battery chemistries. Source: NASA*

6.4.1.1) Lead-Acid

Lead acid batteries are most likely used for electric vehicles such as scooters, wheelchairs, bikes, and boats. These are powerful and are able to deliver a high capacity to devices. These batteries are known for reliability and capability. Unlike LiPo batteries, lead acid batteries are able to withstand bad practices in charging in cases where charging is slow, fast, or overcharging. Lead acid batteries, while powerful, bring too many constraints for use with a quadcopter project. The largest constraint, and reasoning of Medidrone to not utilize lead acid type batteries is the weight of these batteries.

6.4.1.2) Nickel-Cadmium

Nickel-cadmium (NiCd) type batteries are a portable and rechargeable type cell. Applications of NiCd cells include car batteries, main feature supplying a short burst of high current at ex. 12 volts in order to start a car. These are not ideal for a long stride of high current, but are ideal for low current for a longer amount of time. These type rechargeable batteries are most common in households, at lower voltages include AAA, AAA, C and D type battery cells. NiCd batteries are less available due to the sourcing of Cadmium, these could be used to power 3.3V and 5V peripherals on the drone but are not ideal for long periods of power delivery at high current to electric speed controllers.

6.4.1.3) Nickel-Metal Hydride

Nickel-metal hydride (NiMH) is similar to the Nickel-cadmium type battery in the fact both are rechargeable. These batteries by nature, carry a capacity two to three times that of NiCd batteries. These batteries are known to self-discharge more than NiCd cells. Overcharge of Nickel-metal hydride batteries is bad practice, and will destroy these type of cells. NiMH batteries, ideal for short bursts of current up to 1A such as a digital camera, are not ideal for ESC's but can be used to provide power to peripherals.

6.4.1.4) Lithium-Ion

Lithium-ion (Li-ion) contains a higher specific energy than NiMH batteries. These batteries are rechargeable and serve applications in phones, electric vehicles, and aerospace such as satellites. If Li-ion batteries become damaged, these cells feature a very high risk of combustion due to flammable electrolyte. Flammability of batteries is reduced with proper charging and discharging practices. If these batteries become damaged, using them further, presents the greatest fire risk. Lithium-ion batteries are paired with battery management systems, and power distribution in order to maintain proper discharging to the device and to keep from damaging the battery's cells.

6.4.1.5) Lithium-Polymer Utilized by MediDrone

Lithium-polymer (Li-po) batteries are the most ideal for MediDrone. These batteries are rechargeable, and can supply a large amount of current for a moderate amount of time (enough time to deliver payload from locations A to B and possibly back from B to A). These contain a higher specific energy than the other battery types above. Applications of Li-Po batteries include portable devices, radio controlled aircraft, and other applications similar to MediDrone. LiPo batteries come in different packs named by the amount of cells they contain. These packs can either be assembled in-house or purchased as a pack. The popular types of Li-Po packs include but are not limited to 2S, 3S, and 4S signifying two, three, and four cell Li-Po batteries respectively. Each lipo battery cell or pack contains a capacity rating and max discharge rating. Choosing the correct cells to purchase, amount of cells required, and configuration to apply will be paramount to the power delivery system of the MediDrone.

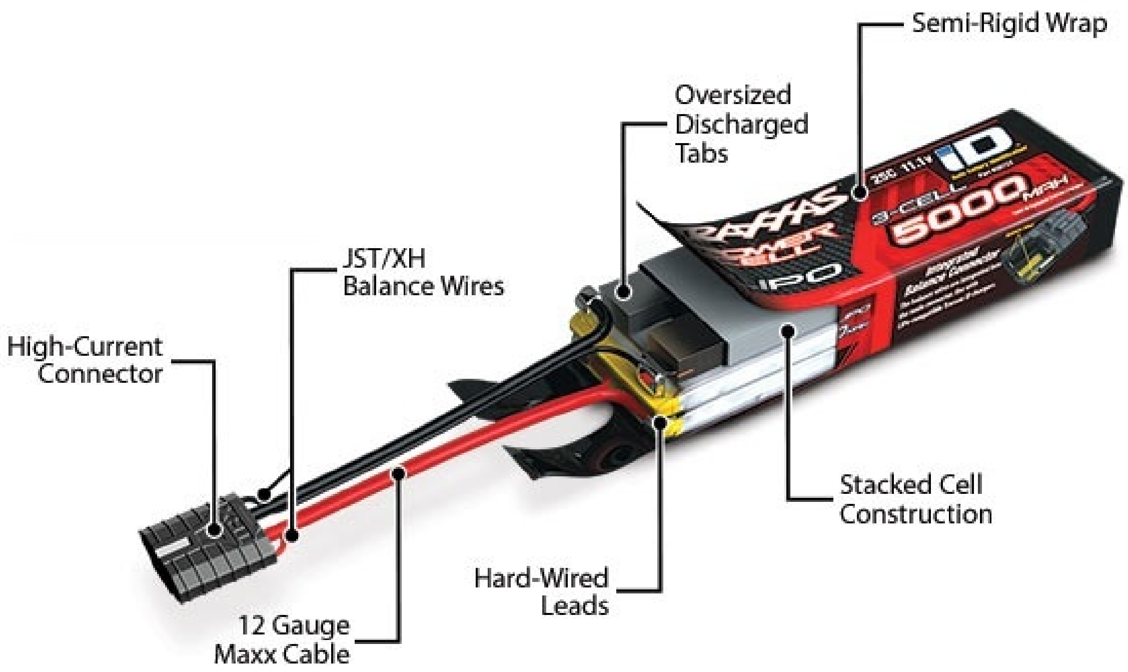


Figure 70

https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.dronetrest.com%2Ft%2Flipo-batteries-a-guide-to-using-and-looking-after-your-batteries%2F1278&psig=AOvVaw36dwhHPkDO1milmONk7huZ&ust=1669848193495000&source=images&cd=vfe&ved=0CA8QjRxqFwoTCPihm7W71PsCFQAAAAAdAAAAABAG

Li-Po batteries require proper charging practice to avoid damage. Any damaged cells must be disposed of immediately before use to avoid combustion. Li-Po batteries are light weight, unlike lead acid, a feature complimentary to the MediDrone. In order to carry a larger load, the weight of battery, electronics, and quad copter frame must be minimal. Li-Po batteries will meet weight, capacity, and safety requirements of the MediDrone.

Previous projects of similar scope (quadcopter with payload) utilize a 4S LiPo battery capacity 75C and discharge of 1300 mAh. This type of LiPo is available around $10 to $30 from various vendors including Amazon. While precedent is a good indicator of products to seek for the MediDrone, the MediDrone team will continue ongoing research to seek the most optimal LiPo battery and configuration.

6.4.2) Battery Configuration LiPo 6S Configuration Used

Once the particular battery cell has been selected, the configuration of the battery pack can be determined. The configuration determines the overall voltage, peak current, capacity, weight, size, and cost of the battery pack. The configuration is two numbers: the number of cells in series, and the number of cells in parallel.

Series connection is when the negative terminal of one battery is connected to the positive of another. The resulting positive and negative terminals with no joining connections will make up the potential difference when adding the two (or more) batteries' voltages.

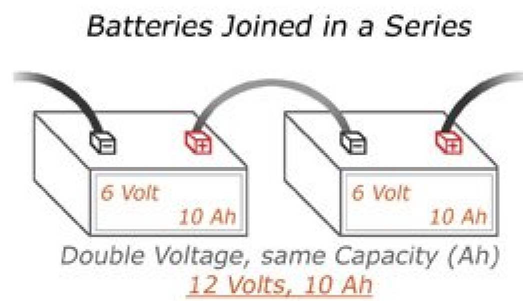


Figure 71: *Batteries connected in series.*
Source: batterystuff.com

Parallel connections between batteries do not alter the voltage levels like series connections. Instead of summing the voltages, we sum the charge stored within the batteries. For example, if we have two batteries of 5 volts, each with 100mAh oh charge, adding two in parallel will produce a battery of 5 volts with 200mAh of total charge. Doubling the total capacity of charge and allotted current for a particular system.

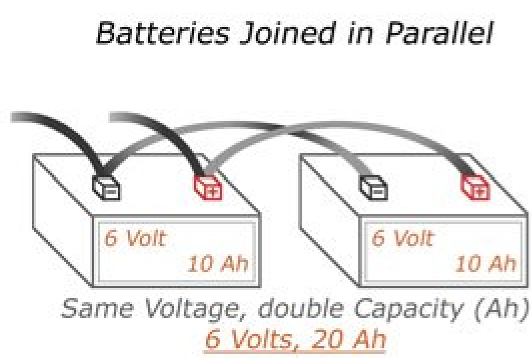


Figure 72: *Batteries connected in parallel*
Source: [batterystuff.com]

LiPo battery cells are connected in parallel in order to create a standard xS type LiPo battery pack. Medidrone will seek to purchase a pre-configured LiPo battery pack but will maintain the ability to design and implement an in-house LiPo battery configuration.

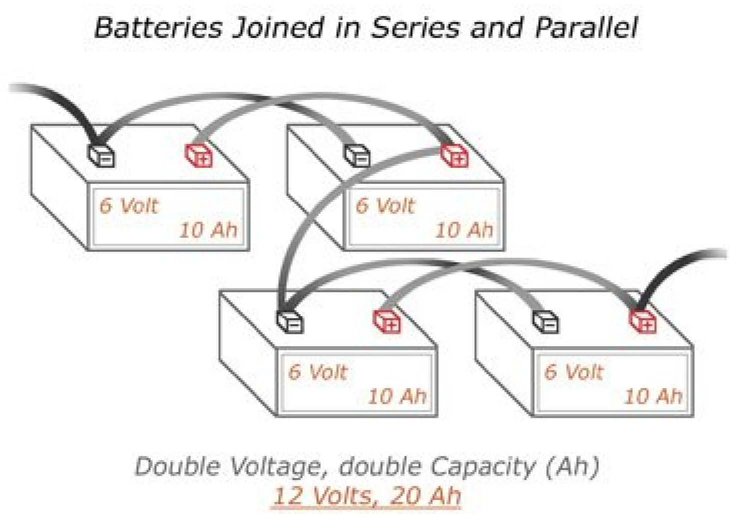


Figure 73: *Batteries connected in series and parallel*
Source: [batterystuff.com]

Series and Parallel configurations used in tandem can meet complex power electronics necessities. These configurations require further calculations and verification in order to meet specification of the load, and maintain safety of the electrical system.

The formulas which determine the pack characteristics from the cell characteristics and pack configuration are:

# of cells overall = (# of cells in series) * (# of cells in parallel)
pack voltage = (cell voltage) * (# of cells in series)
pack peak current = (cell peak current) * (# of cells in parallel)
pack capacity = (cell capacity) * (# of cells in parallel)

pack weight = (cell weight) * (# of cells overall)
pack volume = (cell packing efficiency) * (cell volume) * (# of cells overall)
pack cost = (cell cost) * (# of cells overall)

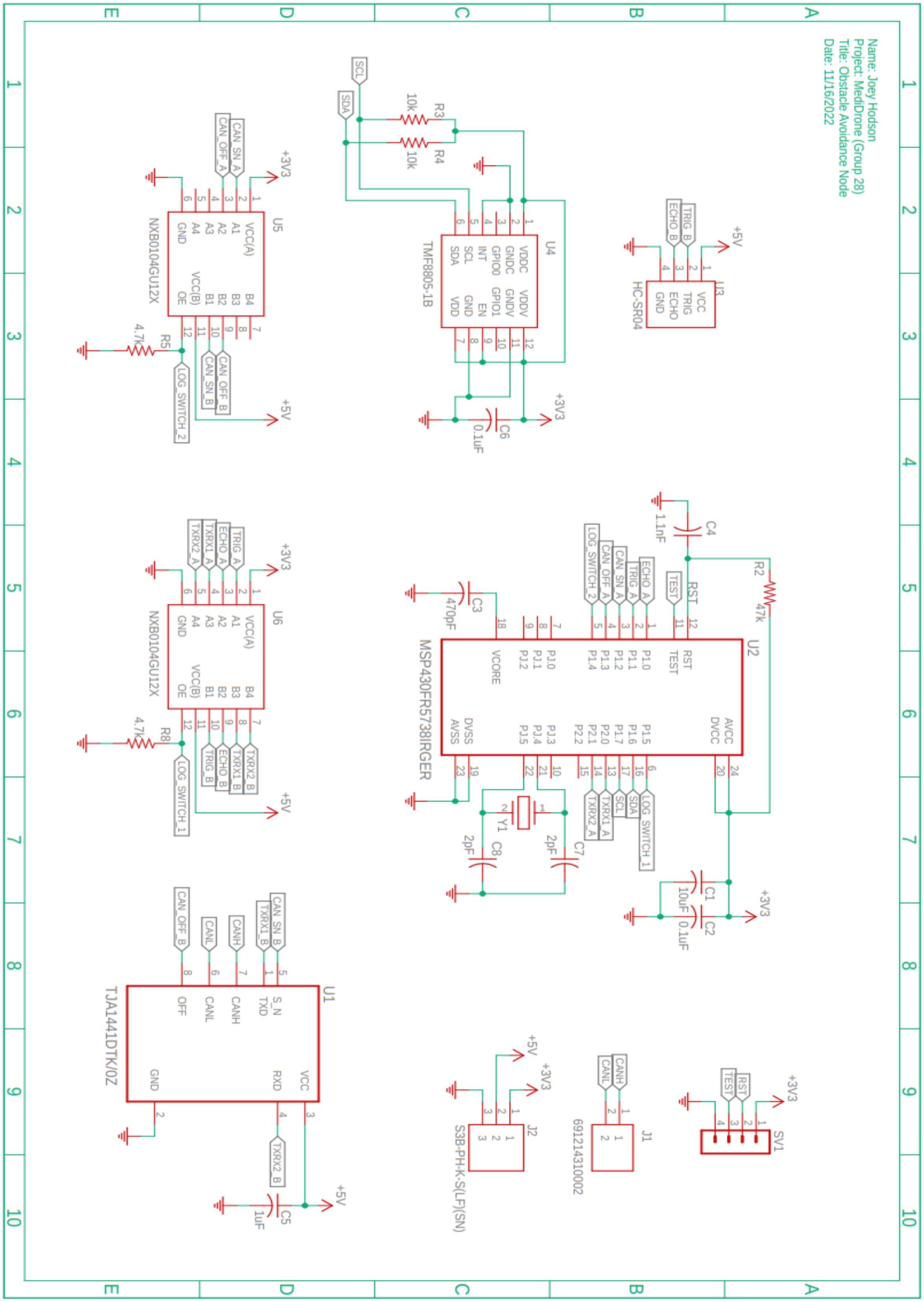
Proper electrical safety will be practiced at all times when MediDrone team members work with battery packs and configurations. MediDrone team ensures that batteries will be disposed of if damaged and will never accidentally short a LiPo battery. Having the privilege to work in a laboratory space with expensive and limited equipment requires that the highest safety, laboratory, and procedural standards be followed at all times. Any battery pack made in house will require housing, either market bought or 3D prinied in-house. Standards regarding battery housing will be followed to ensure the battery pack will not be damaged as battery damage can lead to flammability or skin contamination. LiPo batteries are safe to use and operate as long as all safety standards are followed by all MediDrone team members.

6.5) Schematics

The schematics in the following sections were made using AutoDesk Eagle. All schematics displayed are in-house designs from various MediDrone team members. Different libraries for Eagle were found with the help of ComponentSearchEngine.com and SnapEDA. A few of the schematic symbols were made on SnapEDA with their easy-to-use interface. Other programs available for use include Altium and KiCAD, each with their own strengths and weaknesses. Medidrone utilizes mostly in-house schematics supported by Eagle and KiCAD.

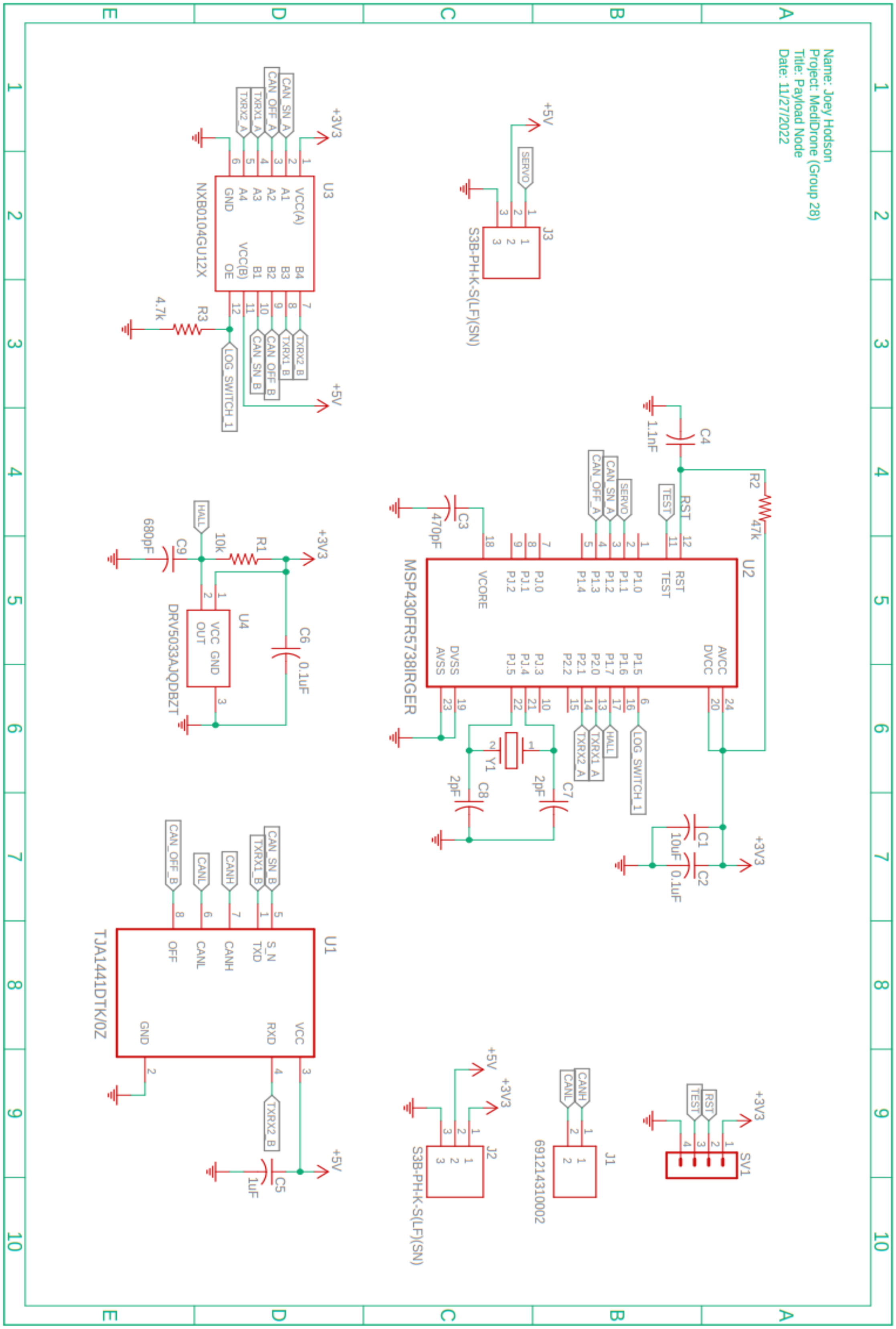
6.5.1) Obstacle Avoidance Nodes

The schematic above represents sensor node 1 from the network diagrams in the previous few sections. The schematic above will easily transform into the other network nodes on the drone. For example, the msp430 and surrounding circuit to the chip will be nearly identical across the nodes. The CAN transceiver, level shifter, and connectors will all also be across the other nodes. The only unique components for this node includes the extra level shifter, the ultrasonic board, and the ToF sensor.


Name: Joey Hodson
Project: MediDrone (Group 28)
Title: Obstacle Avoidance Node
Date: 11/16/2022
U3
HC-SR04
U4
TMF8805-1B
U5
NXB0104GU12X
U2
MSP430FR5738IRGER
U6
NXB0104GU12X
U1
TJA1441DTK/0Z
SV1
J1
691214310002
J2
S3B-PH-K-S(LF)(SN)

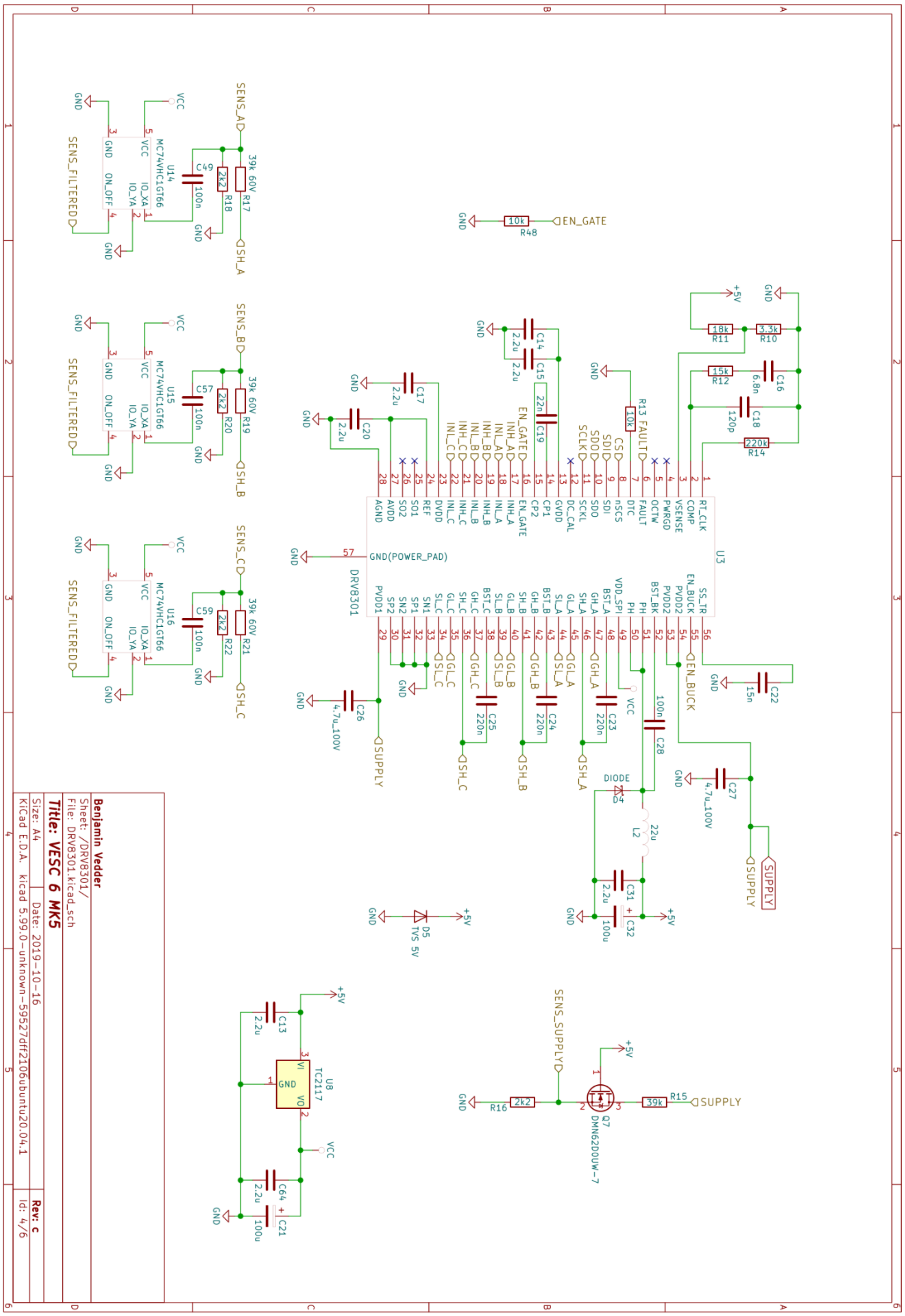
6.5.2) Payload Node

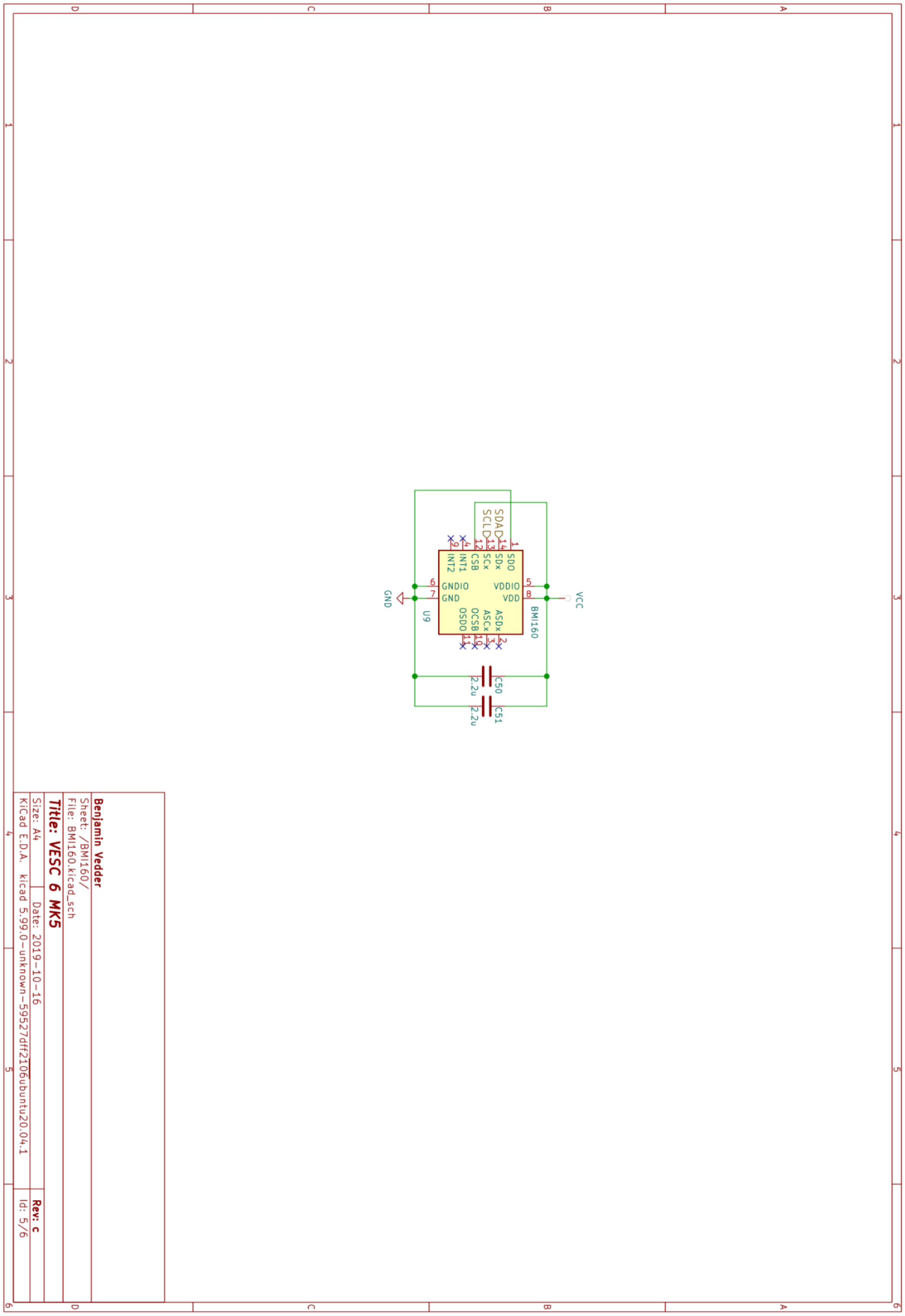
The rest of the boards, like the payload node, connected to the CAN hardware network will share similar schematics as the controller, transceiver, and level shifters are needed for all the circuits. Many of the minor components are therefore similar as well. The main difference between the payload node and the obstacle avoidance nodes is first, it uses only one level shifter chip rather than two since there are less 5v chips on board and more 3v3 chips. This makes it easier to interface directly to the 3v3 controller on board directly. Also, this circuit obviously doesn’t contain any sensors for obstacle avoidance but instead has a new connector for the servo motor that will be connected to this node, controlling the rise and fall of the connected payload box to the drone. More information on the specific motor is found in the previous Hardware Network section. Also, as a late development, a new idea to implement a payload sensor was formed. The problem stemmed from our temperature sensor being inside the payload box. There was no good way to have the payload box have a temperature sensor within, but stay connected, via physical wires, to the drone as the payload will be disconnected and connected repeatedly. Instead, a bluetooth circuit, with a temperature sensor, will be placed within the drone. The payload will also have a magnet on the top of the box. This payload node (drone side) will have a hall effect sensor measuring the magnetic field so when the box is detected as “loaded” to the drone, the flight controller will begin a bluetooth connection to the temperature sensing circuit within the payload. This eliminates all physical electrical connections to the payload box.



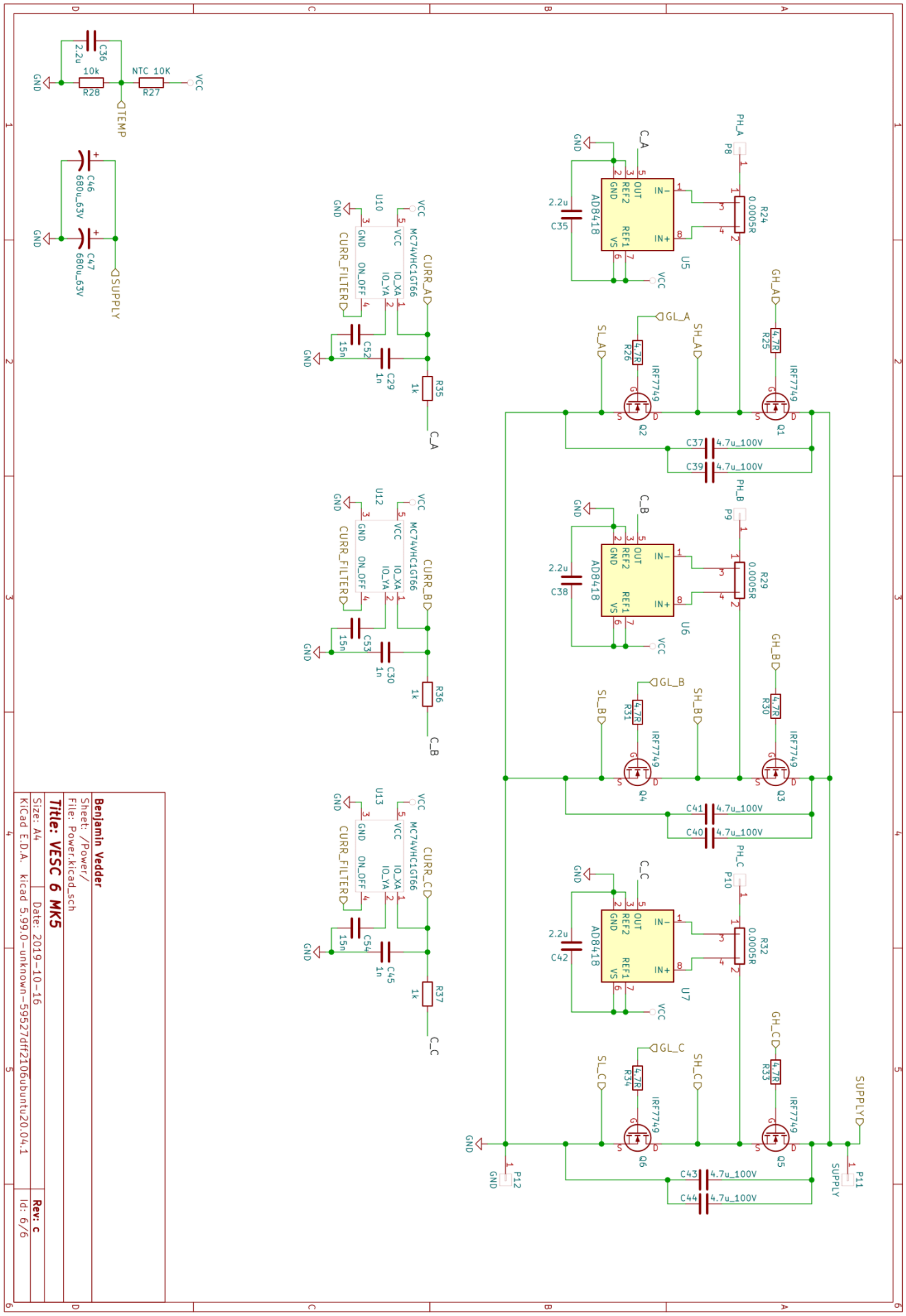
Name: Joey Hodson
Project: MediDrone (Group 28)
Title: Payload Node
Date: 11/27/2022
MSP430FR5738IRGER
NXB0104GU12X
DRV5033AJQDBZT
TJA1441DTK/0Z
S3B-PH-K-S(LF)(SN)
691214310002

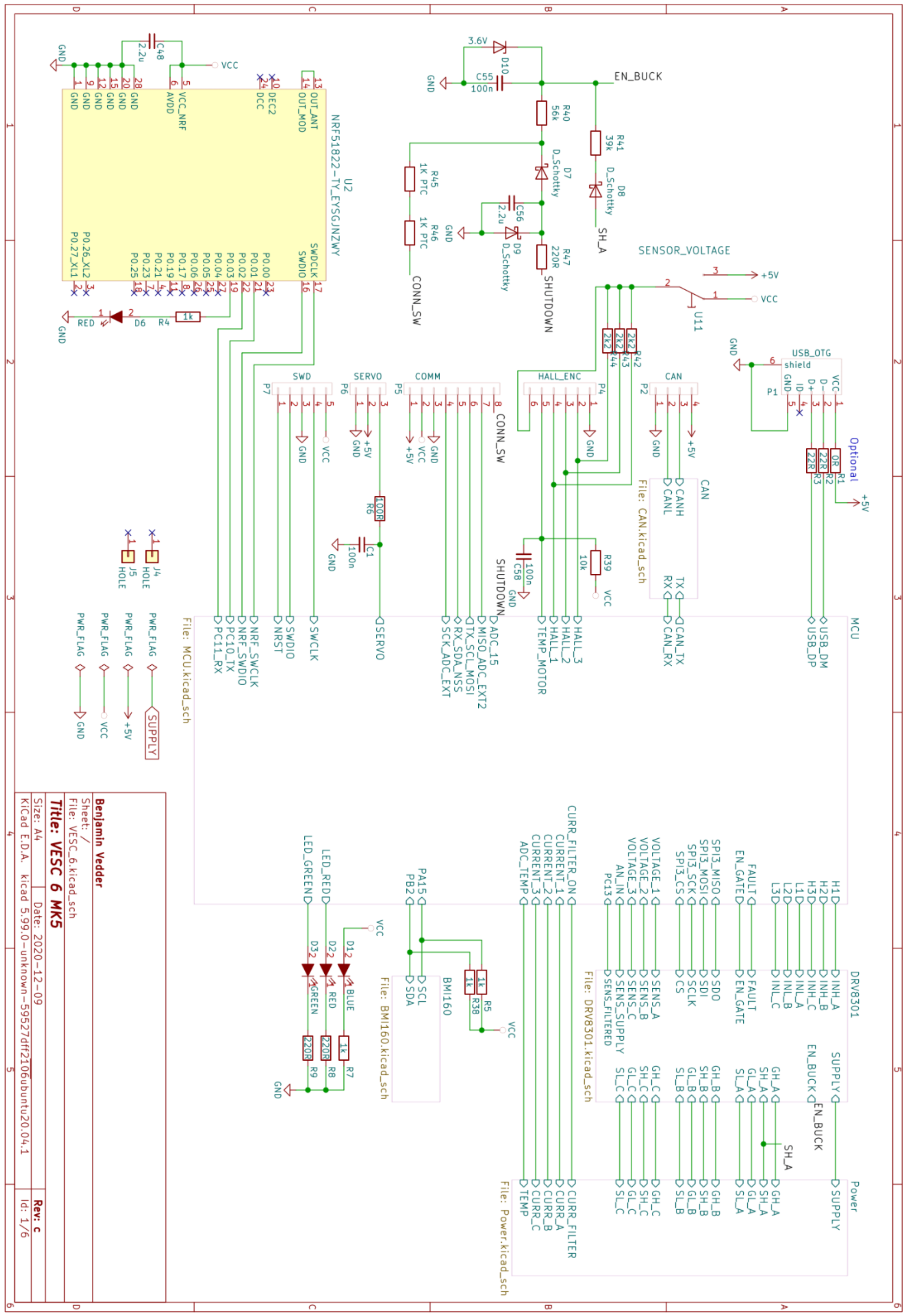
6.5.3) Electronic Speed Controllers

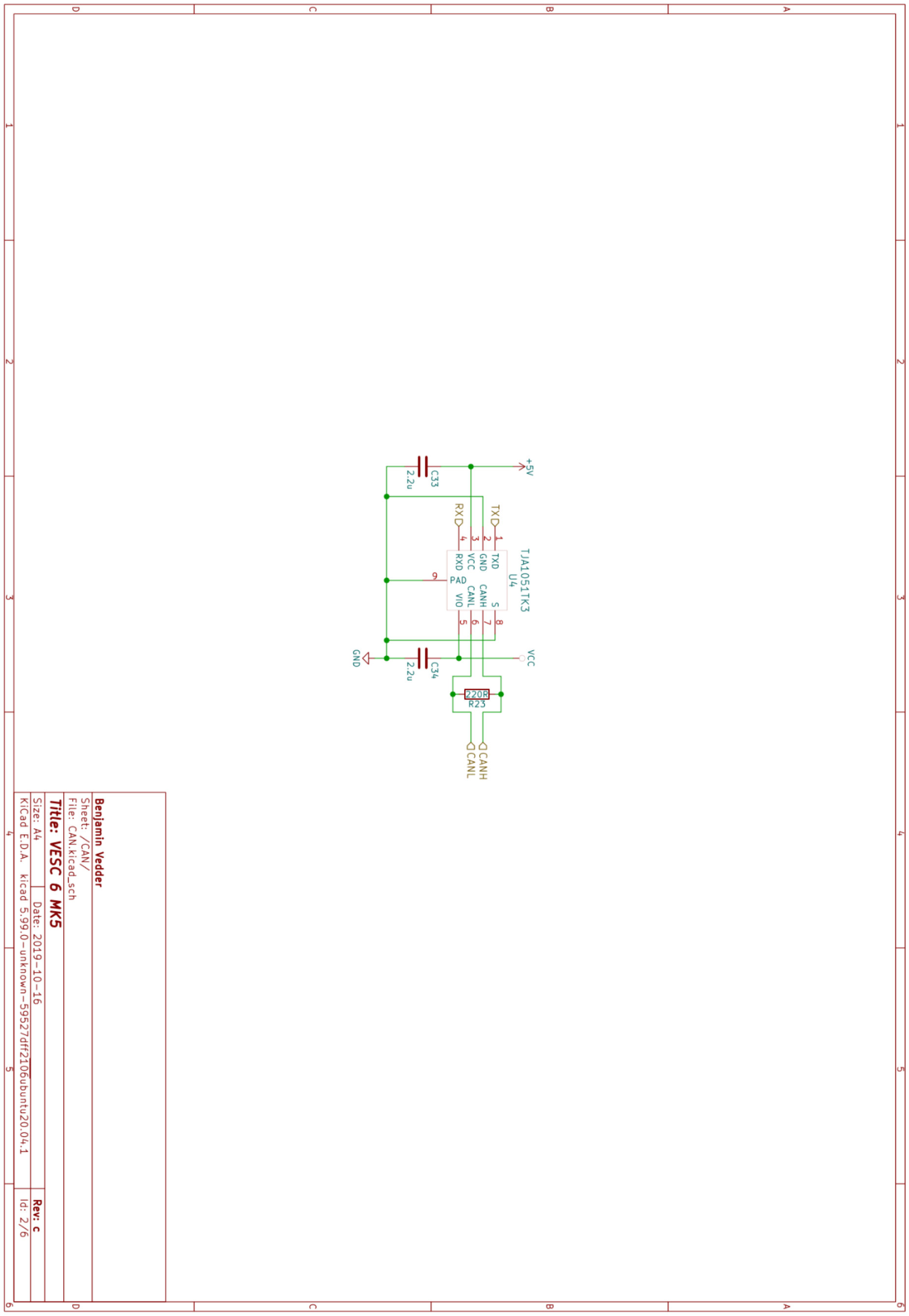




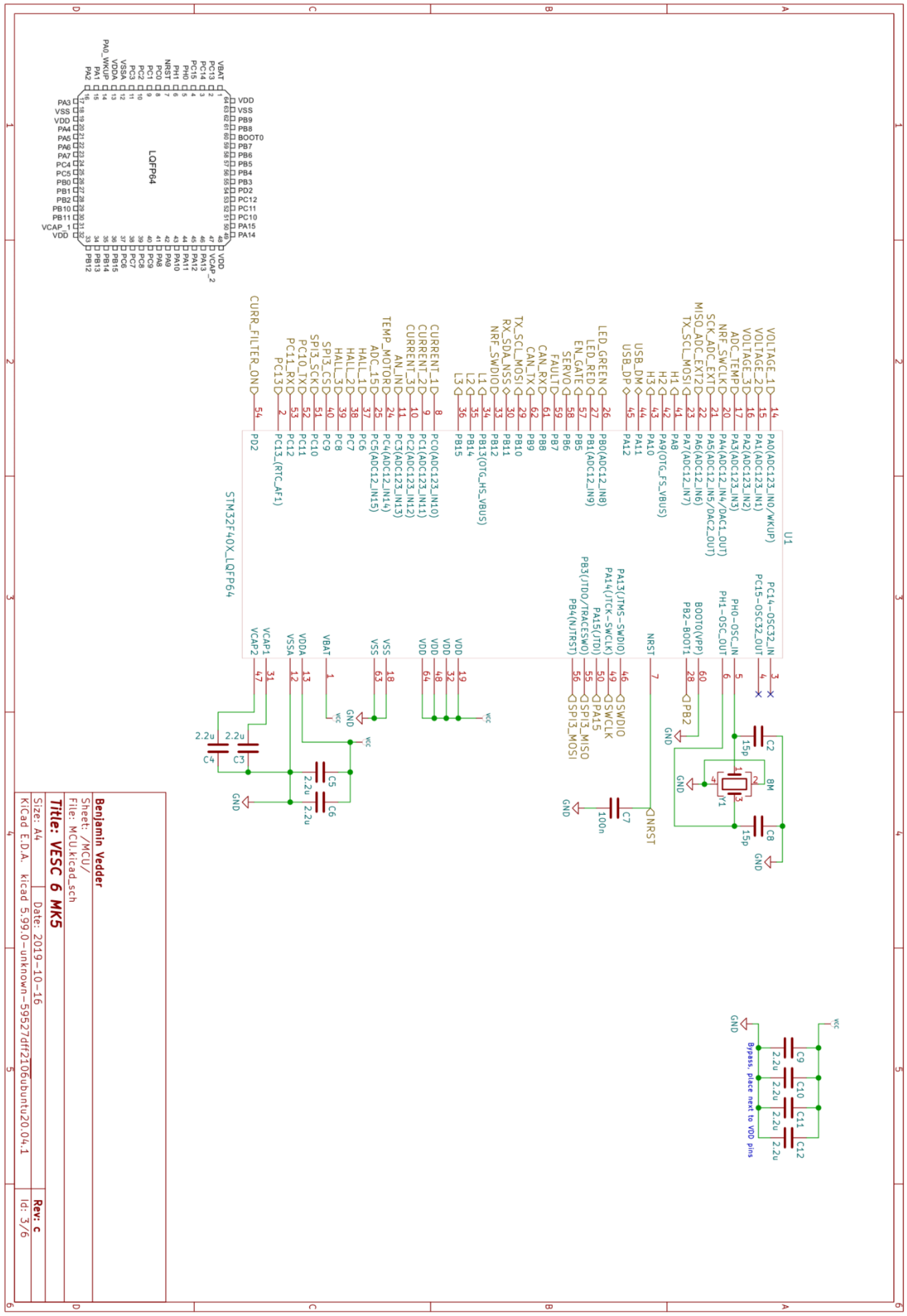
SDA
SCL
SDO
SDx
SCx
CSB
INT1
INT2
GNDIO
GND
VDDIO
VDD
ASDx
ASCx
OCSB
OSDO
BM160
U9
GND
VCC
C50
2.2u
C51
2.2u
Benjamin Vedder
Sheet: /BMI160/
File: BMI160.kicad_sch
Title: VESC 6 MK5
Size: A4
Date: 2019-10-16
Rev: c
KiCad E.D.A. kicad 5.99.0-unknown-59527dff21~106~ubuntu20.04.1
Id: 5/6


Benjamin Vedder
Sheet: /Power/
File: Power.kicad_sch
Title: VESC 6 MK5
Size: A4
Date: 2019-10-16
Rev: c
KiCad E.D.A. kicad 5.99.0-unknown-59527dff2106ubuntu20.04.1
Id: 6/6


Benjamin Vedder
Sheet: /
File: VESC_6.kicad_sch
Title: VESC 6 MK5
Size: A4
Date: 2020-12-09
KiCad E.D.A. kicad 5.99.0-unknown-59527dff2106ubuntu20.04.1
Rev: c
Id: 1/6
NRF51822-TY_EYSGJNZWY
U2
MCU
File: MCU.kicad_sch
DRV8301
File: DRV8301.kicad_sch
BMI160
File: BMI160.kicad_sch
Power
File: Power.kicad_sch
CAN
File: CAN.kicad_sch
SWD
SERVO
COMM
HALL_ENC
USB_OTG
SENSOR_VOLTAGE
EN_BUCK
SHUTDOWN
CONN_SW
Optional



C33
2.2u
+5V
RXD
TXD
TJA1051TK3
U4
RXD
VCC
GND
TXD
PAD
CANL
CANH
VIO
S
GND
C34
2.2u
VCC
220R
R23
CANL
CANH
Benjamin Vedder
Sheet: /CAN/
File: CAN.kicad_sch
Title: VESC 6 MK5
Size: A4
Date: 2019-10-16
KiCad E.D.A. kicad 5.99.0-unknown-59527dff21~106~ubuntu20.04.1
Rev: c
Id: 2/6

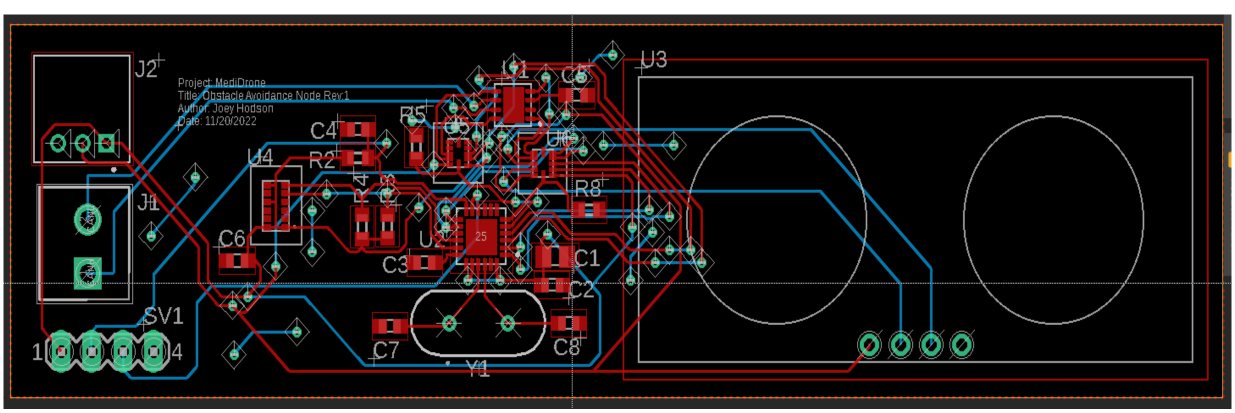

Benjamin Vedder
Sheet: /MCU/
File: MCU.kicad_sch
Title: VESC 6 MK5
Size: A4
Date: 2019-10-16
KiCad E.D.A. kicad 5.99.0-unknown-59527dff2106ubuntu20.04.1
Rev: c
Id: 3/6
STM32F40X_LQFP64
U1
LQFP64
Bypass, place next to VDD pins

6.6) Board Layouts

Laying out a board requires a bit of a different expertise than designing a schematic. Rather than simply picking components, component orientation and proximities to one another require attention unlike designing the schematic. Most of the circuits connected to the CAN bus shared similar schematics and even similar board layouts. Both, however, for each distinct node, required tweaking and modifications to reach their respective desired performances.

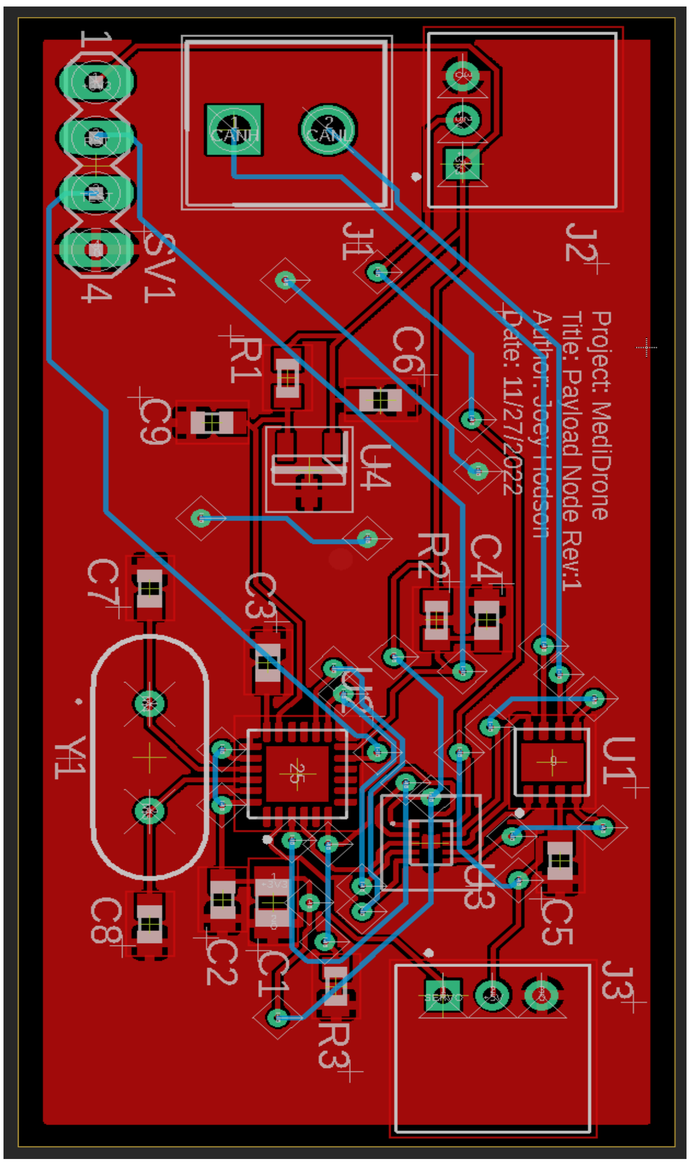
6.6.1) Obstacle Avoidance Nodes

There will be 4 obstacle avoidance nodes oriented on the drone as discussed in the earlier sections. All four of these nodes will face a north, east, south, or west direction. Each of these boards are equipped with both a ToF and ultrasonic sensor. The specific ultrasonic sensor we chose is actually already a breakout board and will therefore act as a shield to the obstacle avoidance node. It will slot into the pin headers on our designed board and will be soldered in place. The main difficulty with this board was thinking about the max size requirements and how it should be oriented on the drone. This is a slim board, it will be able to fit nicely on each of the sides, not taking up too much room. Another big consideration for this layout was giving the ToF sensor, **U4**, plenty of breathing room from other components besides it. For example, the connectors on the left side of the board are spaced far enough from the sensor where it should not pick these up which could cause for another revision. This node also has a ground plane. **J2** is the 3v, 5v, and ground connections. **J1** is a screw terminal leading to the CAN bus, while **SV1** is the Spy-Bi-Wire, 2-wire, programming interface which will be flashed with the help of a fet from a Texas Instruments LaunchPad which the team has on hand.



6.6.2) Payload Node

This board is much smaller than the obstacle avoidance node from the previous section. Sitting at roughly the same height, this board is only 2 inches long, keeping its weight minimal while also making it easy to position on the bottom of the center platform of the drone. **U4** on this board is the hall sensor which will detect the magnet on top of the payload box when in range (fastened adjacent to the drone’s bottom). This board has one more connector than the previous board, sharing the same, right-angle, JST connector as **J2** on this board and the previous board. This new connector, **J3**, is the female connection to the servo motor which will be fastened nearby.



7. Software Design Details

7.1) Software Component Diagram

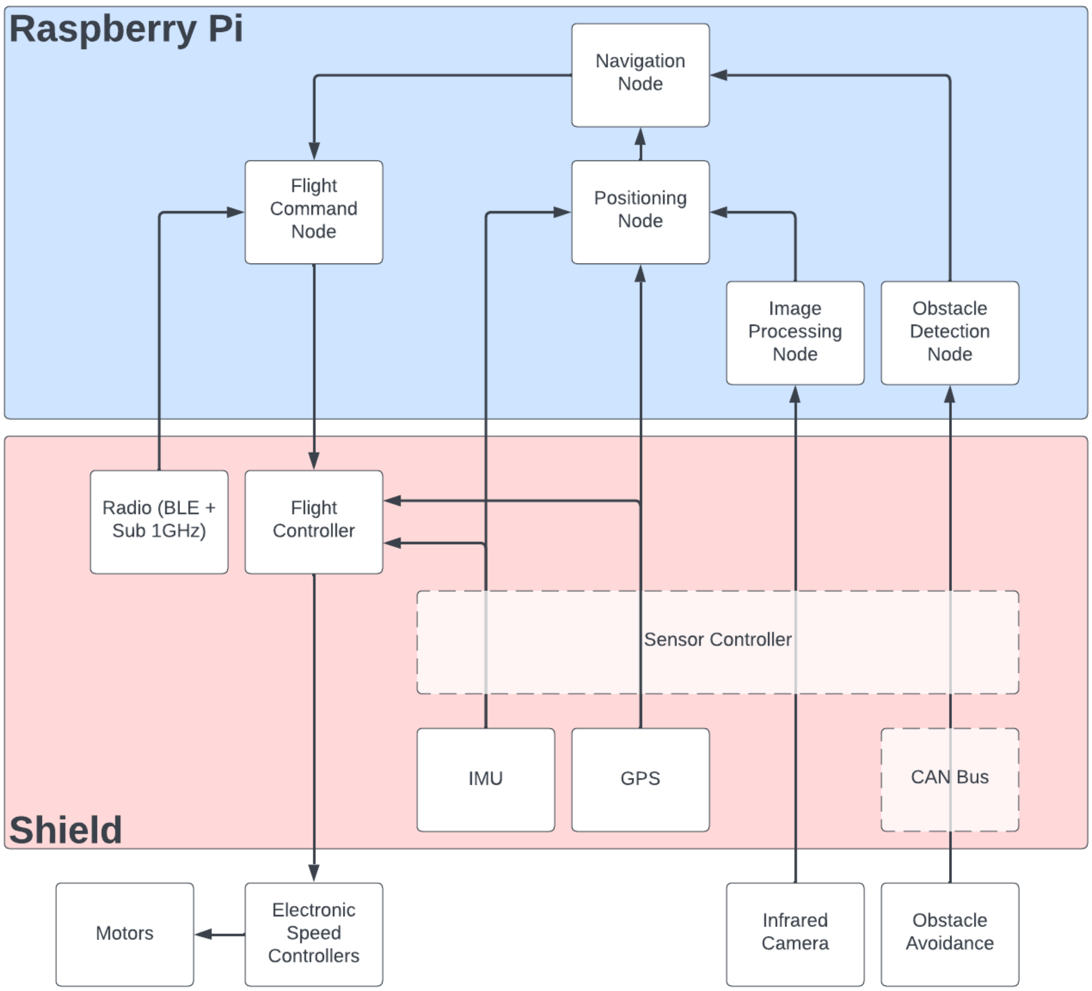


Figure 74

7.2) Physical Layer

The physical layer of the software is all of the code that runs outside of the Raspberry Pi computer. This is all of the firmware on the microcontrollers that will be doing a variety of tasks on the drone. Some of this code is already written for us, but for the microcontrollers that we are implementing, we will have to do some modifications to fit our specific use case.

In particular, this level of abstraction goes between devices, so the software of this layer will need to handle the physical communication between devices. For communication between the Raspberry Pi and microcontrollers, we will implement a full duplex SPI interface for communication, to allow for the highest throughput we can achieve. On the Raspberry Pi, this is done by implementing existing libraries into our ROS code, to speak to the hardware.

7.2.1) Electronic Speed Controllers

In our abstraction, the ESC only takes in one input signal, and produces no output.

Many electronic speed controllers do not run any custom firmware. They accept the input speed as a signal (often pulse width modulation), and drive the motor from 0 to 100% power based off of that signal. All of the calibration is done on the side of the remote control, instead.

At minimum rating, 20 Amp speed controllers are required. MediDrone aims to use a higher rating than 20 Amp ESC's as more current may be required to generate enough lift for the MediDrone to fly successfully.

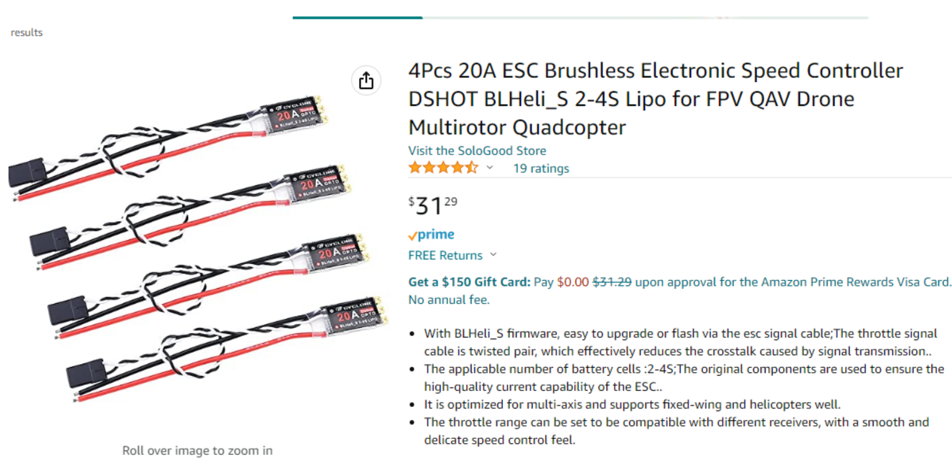


Figure 75

MediDrone is considering use of 60A ESCs to generate the required amount of lift to ensure the payload can physically transport from point A to B. 60A speed controllers will drain the LiPo battery pack in a more rapid fashion than 20A or a variety of lower rated ESCs. The battery configuration will play a large role in the amount of flight time MediDrone can achieve, the ESCs must be rated to supply motors with enough current to ensure the payload's mass be offset for lift.

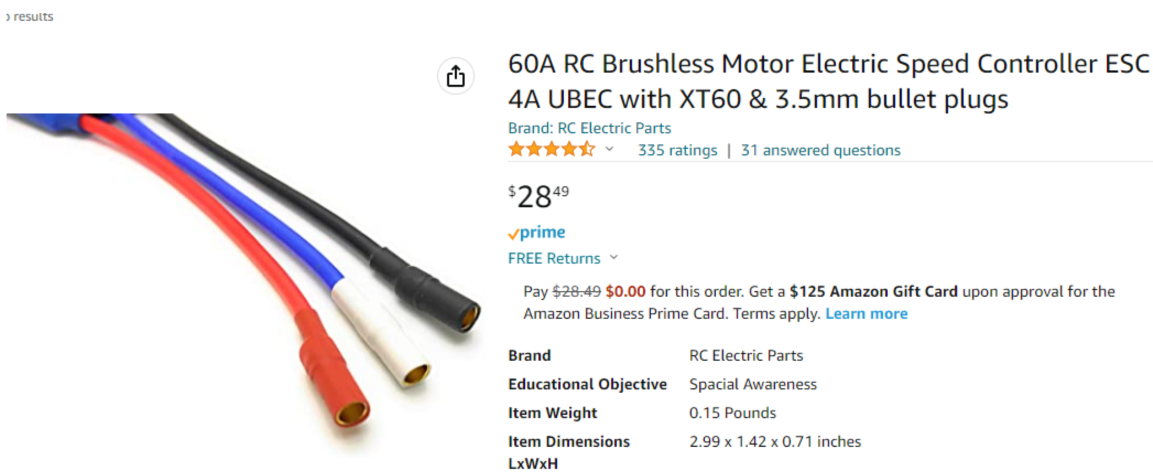


Figure 76

Pricing of ESCs vary, the images above show prices for a set of four 20A and a single 60A ESC respectively. The set of four, 20A ESCs are aout the price of a single 60A ESC. If the MediDrone were to be scaled down to a smaller frame size, and payload, 20A ESCs would serve more ideally. MediDrone will monitor costs and mechanical ability to lift mass X with various ESC's and power distribution options.

7.2.3) Flight Controller

In our model, the flight controller takes in 8 inputs: 4 from the Flight Command node (desired roll, pitch, yaw, and altitude) and 4 from the IMU (current roll, pitch, yaw, and altitude). The flight controller produces 4 outputs (motor power percentage for each of the 4 motors).

On an over the counter flight controller, there is no software here which we need to write. We simply "plug and play" each of these inputs with each of the outputs, and the drone can fly. Each of these flight controllers comes with software which supports tuning of the control parameters for the specific drone.

However, for this use case, our drone will have two different physical configurations: with and without the payload. These two configurations require two different sets of flight controller calibration parameters. For an over the counter flight controller, we can achieve this by recalibrating it while the drone is landed. The Raspberry Pi is a full Linux computer, so we can write a Python script which runs the proprietary flight controller software, updates the flight controller parameters to hard coded values, and re-flashes the flight controller.

However, it is possible that this configuration does not end up working out, due to the flight controller not being designed to be reconfigured dynamically. In this case, we have a plan to design our own flight controller. The theory of flight controllers is already well established: the control system (pictured below) takes in current angles from the IMU as well as remote target parameters, computes the error function (difference between current and desired angles), and feeds that into a simple PID controller to produce the desired power on roll, pitch and yaw.

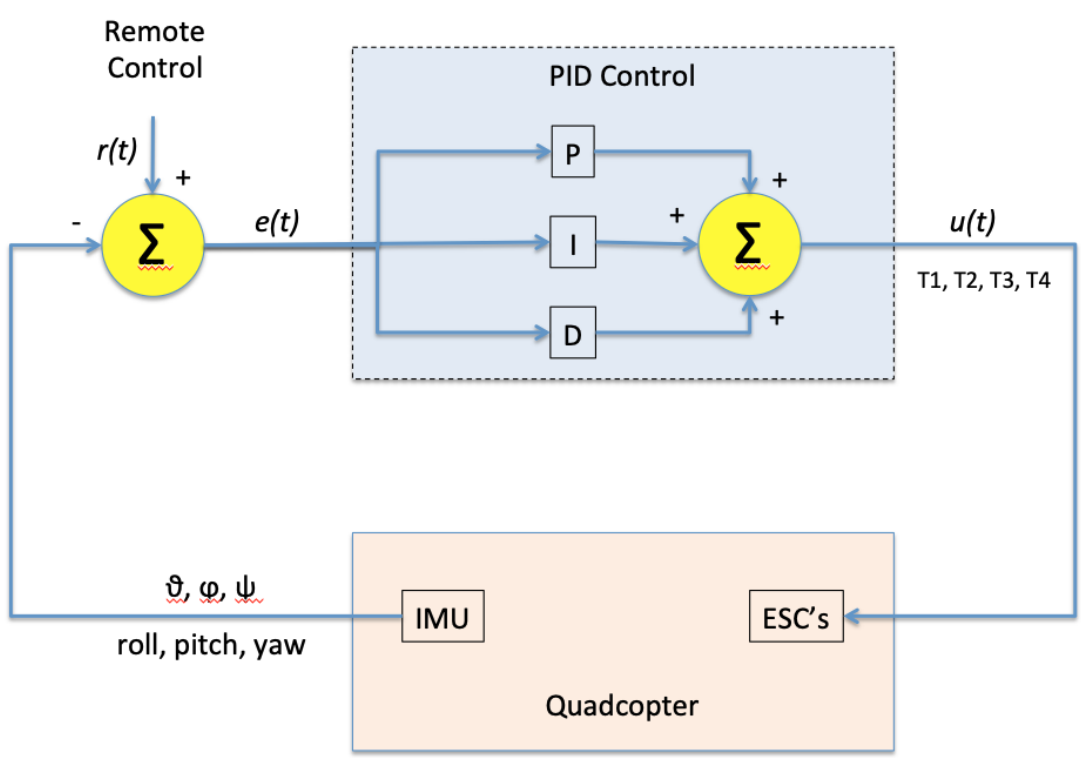


Figure 78

After, we must determine the power sent to the motors themselves. We currently have the amount we would like to drive the drone in each of its angular degrees of freedom, but we use the physical configuration of the drone to turn these into individual motor powers using a process called motor mixing. In the motor configuration pictured below, the formulas to compute each of the motor powers is shown. Note that thrust refers to the vertical height of the drone, which is controlled by the IMU barometer altitude reading, not the IMU gyroscope headings.

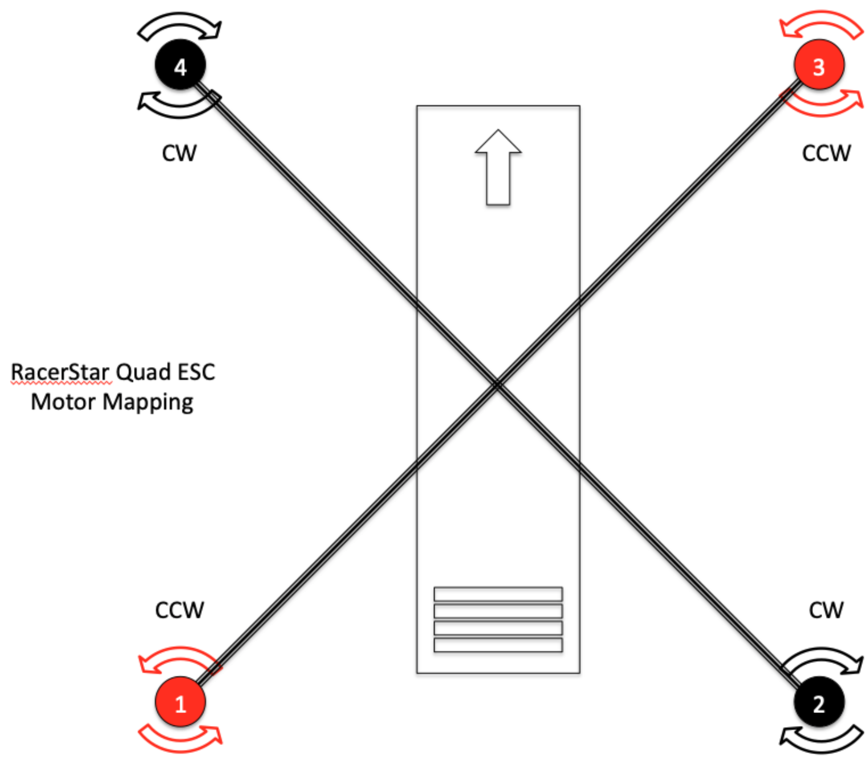


Figure 79

Motor 1 (CCW - back left) = Thrust + Yaw − Pitch − Roll

Motor 2 (CW − back right) = Thrust - Yaw − Pitch + Roll

Motor 3 (CCW − front right) = Thrust + Yaw + Pitch + Roll

Motor 4 (CW − front left) = Thrust − Yaw + Pitch - Roll

We will use a TI C2000 real-time microcontroller, a line of MCUs designed for control applications by providing high performance and low IO latency. TI provides libraries in C for this line of microcontrollers to handle all of the IO. Thus, the software for the flight controller can be boiled down to a simple main loop: read in the inputs, carry out the PID computations, calculate the motor mixing, and send the outputs to the motors. The software also will support a signal via an extra GPIO pin to allow the PID parameters to be toggled. Because the flight controller is only for this drone, we can hard code in the PID parameters, although it would be trivial to accept the parameters from the Raspberry Pi.

There is a wealth of open source flight controller projects (pictured below) we can take reference and inspiration from. In particular, the software designs of LibrePilot and Paparazzi are simple and robust enough for us to implement our own versions of. The key component we need to reference is the implementation of the PID controller, since the rest of the program is straightforward.

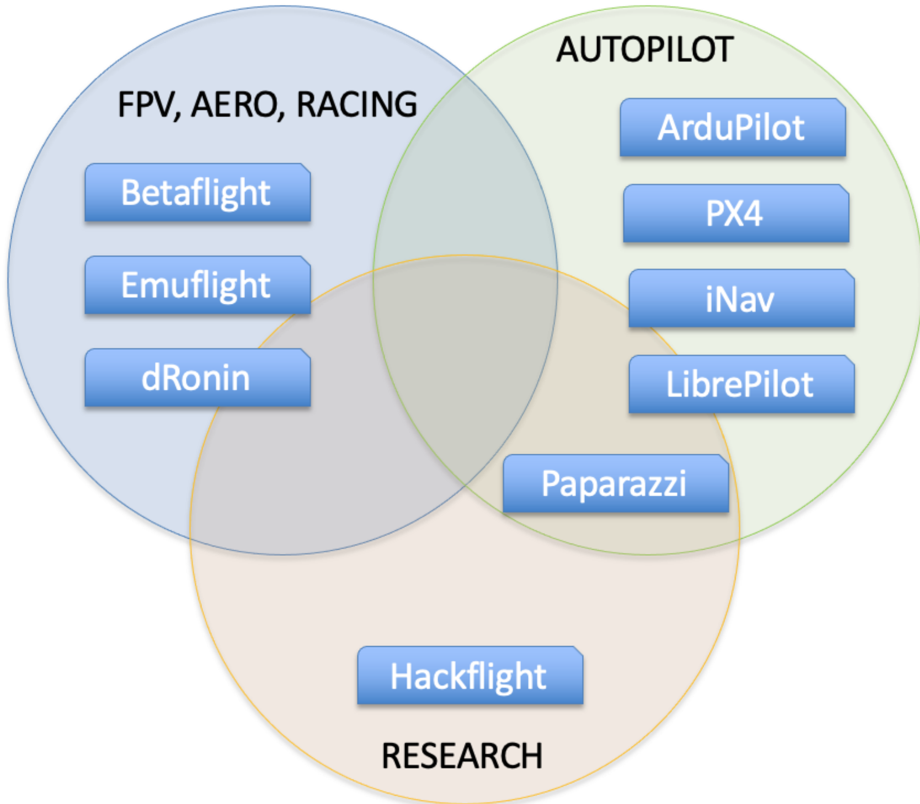


Figure 80

7.2.4) Sensor Controller

In our software, the Sensor Controller will manage the IO to each of the sensors on the drone, and output them to the Raspberry Pi. There is one key exception: the flight controller must have maximum throughput and minimum latency to the IMU, so we will have the Sensor Controller on the same bus as the flight controller to IMU, and have it listen to its communication to the flight controller. That way, we do not stall the flight controller by occupying the IMU with queries. The MCU must have enough free pins to support all of this IO. We will use a TI MSP432 general purpose ARM MCU to implement this Sensor Controller in C.

7.2.4.1) TI RTOS

One key feature about the MSP432 is that its operating system is TI-RTOS, which stands for Real-Time Operating System. This OS allows for programs to pause and interrupt each other based on their hierarchy. The figure below shows the four main levels of hierarchy in TI-RTOS.

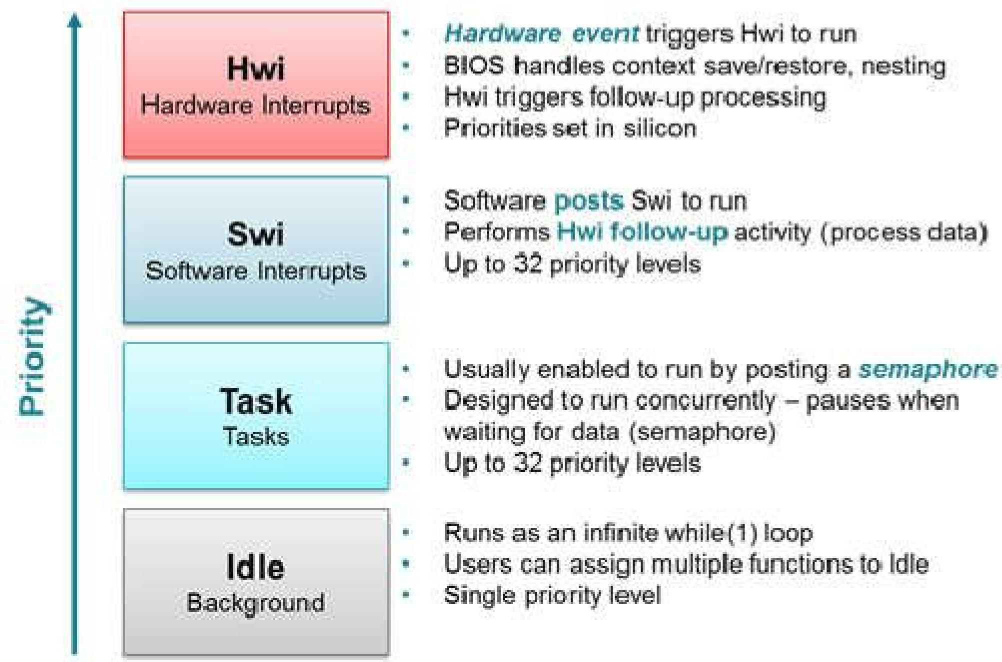


Figure 81

The hardware interrupts pausing the software and tasks allow for the MCU to determine exactly when the clock ticks of hardware communications we are implementing to occur. Then, we can use software interrupts and tasks to handle the actual process of reading and writing via IO. Finally, we can use tasks to manage when, what, and how often we read from each of the sensors. The IMU takes priority, since this MCU does not have control over its communications. Thus, we can use RTOS tasks to prioritize the IMU over the other sensors. The figure below shows an example of how each of these processes usurp each other in an RTOS context.

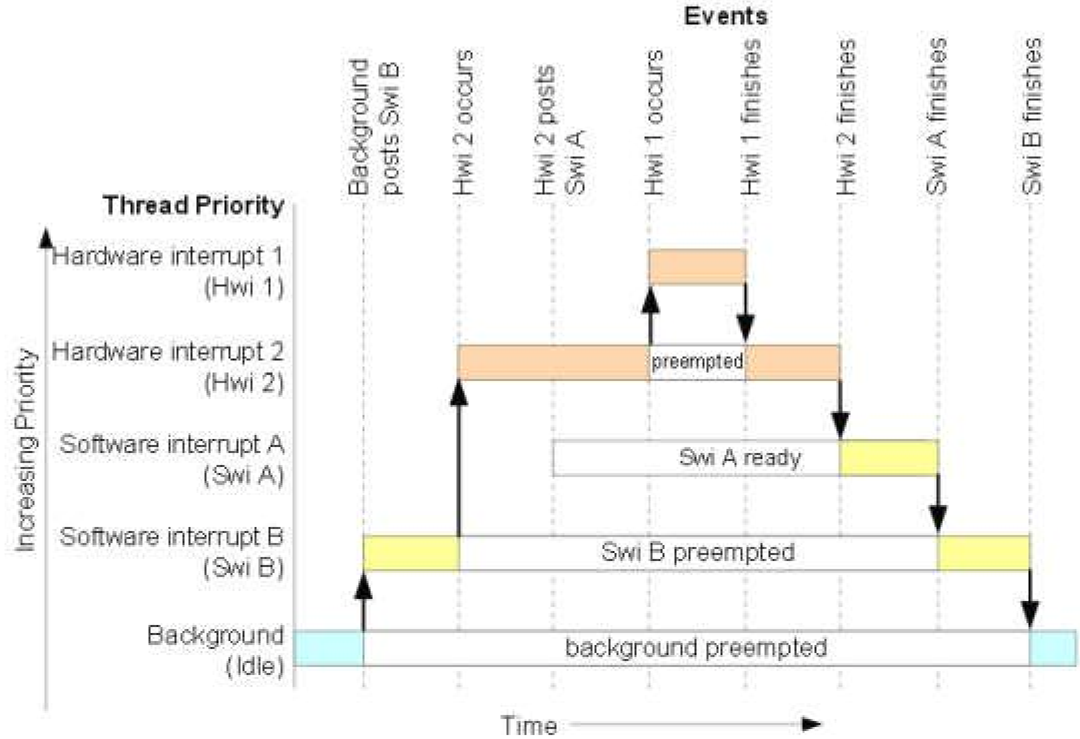
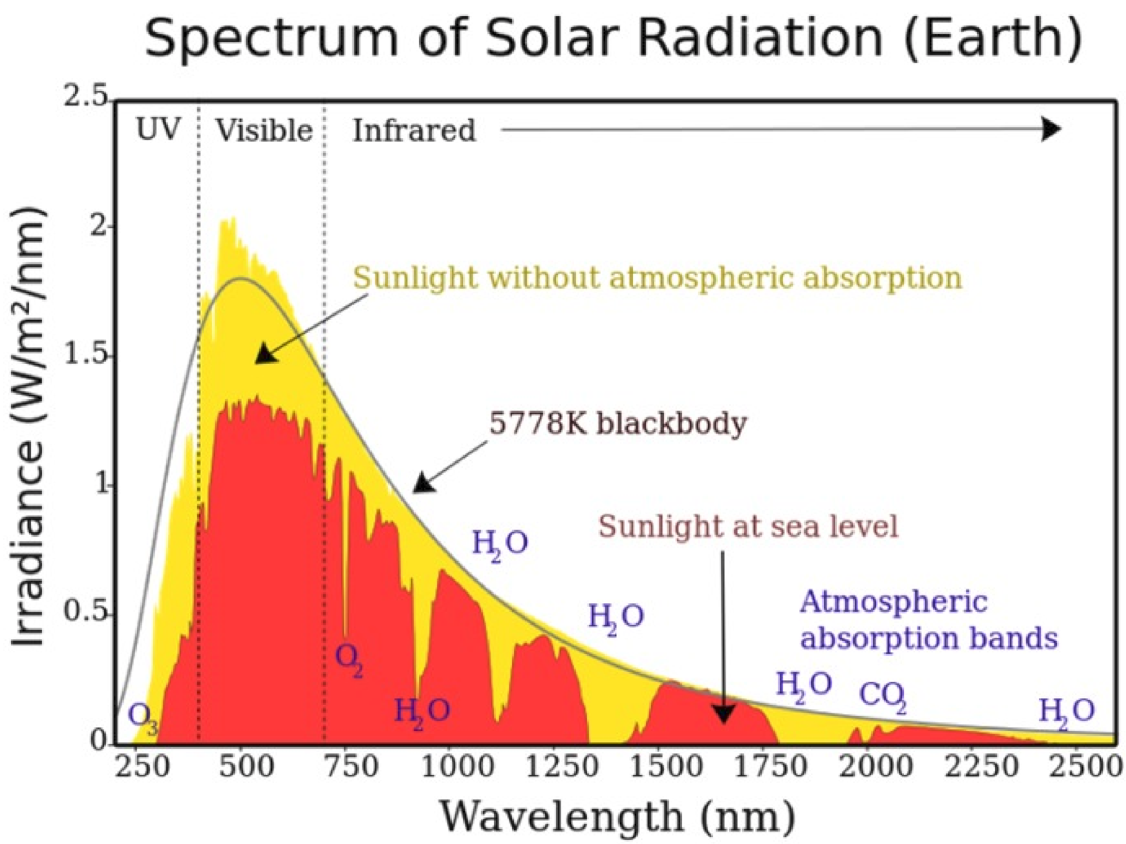


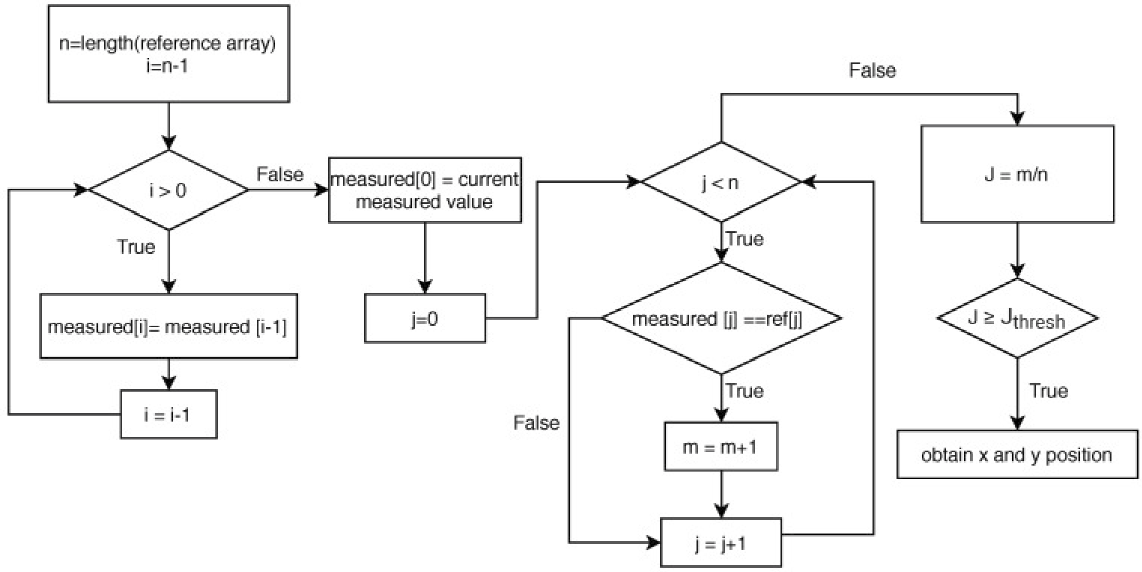
Figure 82

7.3) Precision Landing

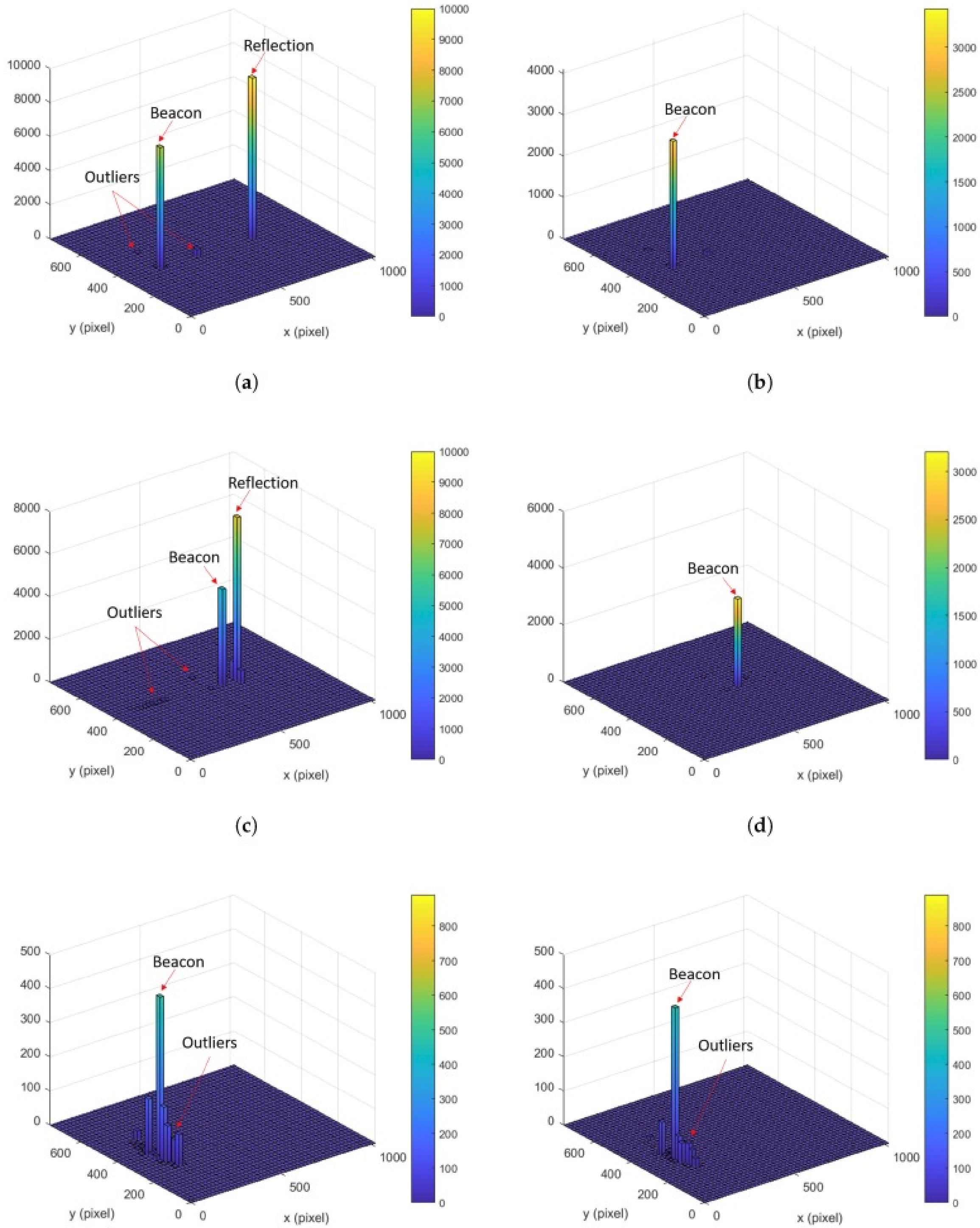
There is a large problem of interference that must be dealt with. The IR beacon will appear as a bright spot on the IR camera, just like any other bright light on a normal visible light camera. However, the Sun still produces infrared light, as well as visible and UV light (shown in the figure below). This means that it is still possible to have interference from the Sun due to reflective surfaces reflecting sunlight into the camera.



This problem can be dealt with by using signal modulation. We are able to control the IR beacon, so we can make it flash at a fixed frequency. Now that we know this frequency, we know how often the image should contain the beacon, and how often the image should contain only interference. A diagram for the signal modulation algorithm is shown below. The idea of the algorithm is to keep track of when the IR beacon should be on or off by keeping a clock on the device. Then, when taking a burst of photos, for each of then we can determine whether or not the beacon was on during that photo. We can then determine for each pixel, whether or not it was “right enough”, in order to make it through the filter. If a pixel is on when the beacon is on, and off when the beacon is off, this is weighted into the count which measures the similarity between the reference value and the measured value.



This algorithm has been tested in a variety of outdoor conditions in a study of up to 60 ft next to a sunlight reflector. The results from 20, 40, and 60 ft. are shown below. On the left is the raw data read in by the camera, checking if the pixel was bright or dark, and on the right is the data after it has been modulated by the algorithm. This gives us the confidence to be able to find the IR beacon in a variety of outdoor conditions.

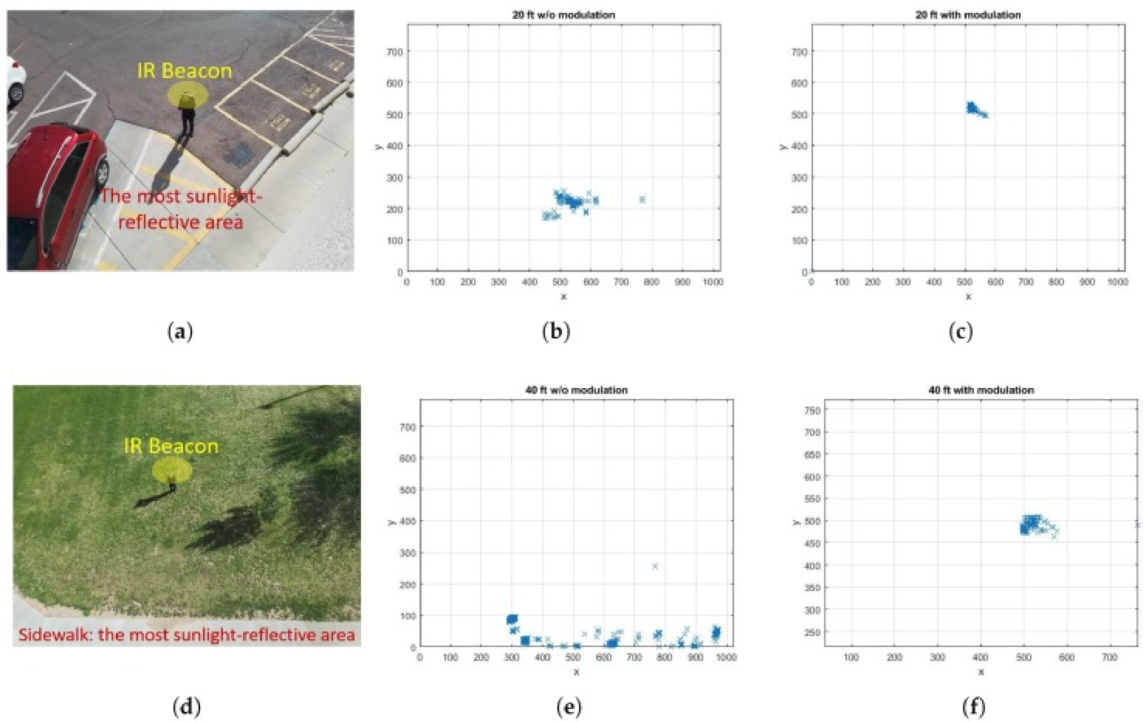

Reflection
Beacon
Outliers
Beacon
Reflection
Beacon
Outliers
Beacon
Beacon
Outliers
Beacon
Outliers
y (pixel)
x (pixel)

(a) (b)

(c) (d)

(e) (f)

Now that this output is properly showing the beacon, we can take the average of all matching positions to find the estimated coordinates of the beacon. This is rerun multiple times per second to produce a tracking dot on the beacon. This is shown in the figure below, where the left is the average of all matching positions without the modulation, and on the right is the position with the modulation.



```
procedure ProcessFrame(f, frames, states, counts, threshhold, T)
    Enqueue(frames, f)
    while Time(Front(frames)) + T/2 < Time(f) do
        Dequeue(frames)
    end while
    f' ← Front(frames)                         ▷ Frame with opposite beacon state
    positions ← ∅
    for each position (x, y) do
        if f[x, y] ≠ f'[x, y] then
            s ← 1
        else
            s ← 0
        end if
        counts[x, y] ← counts[x, y] + s − Front(states[x, y])
        Dequeue(states[x, y])                  ▷ Update buffer of previous states
        Enqueue(states[x, y], s)
        if counts[x, y] ≥ threshhold then
            Insert(positions, (x, y))
        end if
    end for
    return Average(positions)
```

8. Administrative Content

8.1) Parts List

Table 37, below, provides an estimated total cost for the drone including the components talked about in the previous two sections. We are not considering any spares or duplicates here, but will more likely than not require some during the testing and prototyping phases of the design.

Item #	Item	Part Number (if applicable)	Qty
1	Brushless Motors	n/a	x4
2	ESC Mosfet	n/a	x4
3	LiPo Battery Cell	n/a	x20
4	Power Distribution	n/a	x1

	Board		
5	Battery Management System	n/a	x1
6	XT60 Connectors (male & female)	n/a	x1
7	JST Connector Kit	n/a	x1
8	Infrared Beacon Receiver	n/a	x1
9	Infrared Beacon Emitter	APHD1608SF4CPRV-P22	x20
10	Infrared Beacon Controller (MSP430xx)	n/a	x1
11	Thermistor	NXFT15XH103FEAB035	x1
12	Motor Propellers	n/a	x4
13	Payload Servo Motor	SER0011	x1
14	CAN Controller (MSP430xx)	n/a	x9
15	CAN Transceiver	TJA1441	x14
16	IMU	LSM6DSLTR	x1
17	ToF Sensor	TMF8805	x6
18	Ultrasonic Sensor	HC-SR04	x6
19	GPS Antenna	GPSMOD2515	x1
20	GPS	NEO-6M	x1
21	Central Controller (Raspberry Pi Zero 2 W)	n/a	x1

22	Central Controller (MSP430xx)	n/a	x1
23	RF Module (Rx & Tx)	nRF52840	x2
24	Peltier Refrigeration Tablet:	Bolsen TEC1-12704	x2
25	Peltier Testing Module:	KOOBOOK 12V thermoelectric Peltier cooler Refrigeration Cooling Fan System Heatsink DIY Kit	x1
26	Fasteners	n/a	x20
27	Chapie	Seth, Sponsorship	XTBD
28	PLA Filament: -Seth, Amazon	OVERTURE PLA Filament 1.75mm 1Kg	x1
29	LM555	LM555CMX/NOPBTR-ND	x5
30	EMITTER IR 940NM 65MA SMD	1080-1351-2-ND	x5
31	RES 470 OHM 5% 1/4W AXIAL	470QBK-ND	x5
32	CAP CER 10000PF 50V X7R RADIAL	399-4206-ND	x5
33	RES 10K OHM 5% 1/2W AXIAL	10KH-ND	x15
34	CAP ALUM 470UF 20% 16V RADIAL	399-ESK477M016AG3AA-ND	x5
35	HOLDER CR2477	BH1000GINACTIVE-N	x2

	PBT PLAS UL94V-0	D	

Table 37

8.2) Milestone Discussion

One of the many important aspects to ensure that our project is completed effectively and on time is to set milestones and deadlines for them. The list below shows both major and milestones and their appropriate deadlines as of today, set in place by the team. **Table 38** marks both the hard and soft deadlines for Senior Design I, Fall 2022. The adjacent table below, **Table 38**, contains the tentative deadlines for Senior Design II, Spring 2023. **Table 38**'s tasks can be broken up further, but for now, without official deadlines, the table is composed of the main tasks that will be needed.

8.2.1) Fall 2022

During Fall of 2022, the team is working to brainstorm, design, and research implementation details for MediDrone. The following table provides insight on important deadlines.

Deadlines	**Hard/Soft**	**Task**	**Completed (y/n/ip)**
September 16, 2022	Hard	Divide & Conquer v1.0	y
October 7, 2022	Hard	Divide & Conquer v2.0	y
November 4, 2022	Hard	90 Page Design Document Draft	y
November 7, 2022	Soft	• Majority of Research • Final BOM	y
November 18, 2022	Hard	135 Page Scope of Work (Draft)	y
December 1, 2022	Soft	Proposal For Funding	n (exempt)
December 6, 2022	Hard	180 Page Scope of Work (Final)	y

Table 38

8.2.2) Spring 2023

During spring of the new year, the team plans to build and test MediDrone and all of its subsystems. **Table 39** highlights important dates to keep in mind. The first and obvious step is to order our parts. On top of the parts, however, we plan to buy breakout boards to test a lot of the components individually. Other parts, electrical components for example, the team will be able to breadboard and test via subcircuits. Testing and designing simultaneously, will allow us to catch a lot of bugs early on. A 'subsystem' as noted in the table below, is the set of mechanical design (including the payload), all electrical designs (including schematic and pcb ordering), and software design.

Deadlines (Tentative)	**Hard/Soft**	**Task**	**Completed (y/n/ip)**
January 13, 2023	Soft	Breakout Board / Individual Component Ordering	Y (partly)
January 25, 2023	Hard	Individual Component Testing	Y (partly)
January 30, 2023	Hard	All Individual Part Ordering	Y
February 17, 2023	Soft	All subsystem Rev1 design	Y (partly)
February 22, 2023	Soft	All subsystem Rev1 testing	Y
March 3, 2023	Soft	All subsystem Rev2 design	Y
March 10, 2023	Soft	All subsystem Rev2 testing	Y
March 24, 2023	Hard	Drone Assembly	Y
March 31, 2023	Hard	Drone (w/o payload) Functional	Y
April 7, 2023	Hard	Drone (w/ payload) Functional	Y
April 14, 2023	Hard	Extended Tests	Y
April 21, 2023	Hard	Final Demonstration	Y

Table 39