Drone-based Microgravity Experiments
April 27, 2018

Department of Electrical Engineering and Computer Science
University of Central Florida
Dr. Lei Wei
Sponsored by: Northrop Grumman

Group 29
Brenise Barclay – EE
Casey Colón – EE
Hunter Fernandez – CpE
Mitchell Findley– CpE

Contents

Executive Summary

Before using products in space, we need to know how they will function under space-like conditions, namely, microgravity. With all the cost and time involved in sending something into space, it would be downright absurd to just go off of theoretical calculations or expectations of how a design would respond to the lack of gravity in space. This is why we need to have places or methods of testing different products in microgravity conditions.

Currently the options for this kind of testing are very limited. Parabolic flights through Zero Gravity Corporation is one option for this kind of testing but it is very costly, it is $5,197.50 for a maximum of 7.5 minutes of microgravity testing time in 20-30 second intervals [1]. Clearly this is not a realistic option for most researchers who are self-funded. Unfortunately, the other options are equally, if not more, out of reach. These other options include suborbital spaceflight, a drop tower or drop tube, and ultimately being at the International Space Station itself. These “options” are just not practical for many scientists and researchers looking to perform microgravity experiments; this is the motivation behind our project.

We set a goal of providing a low-cost, easily accessible way to conduct experiments under microgravity or reduced gravity conditions. We plan to achieve this using a drone shell to control velocity around an evacuated experimental chamber. Our drone is equipped with on-board data collection and instrumentation to record parameters to characterize the payload environment, such as acceleration, in the future additional parameters could be added. Additionally, we have a camera fixed on the experimental chamber to record particle movements during testing. Furthermore, live acceleration data from inside the chamber is displayed on the researcher’s laptop and that data is automatically stored. After experiments, all data is collected and uploaded to a non-relational database.

Our design shows that a drone based solution has the potential to make the execution of experiments in a simulated microgravity environment more accessible by reducing cost, as well as eliminating the need to travel long distances or fight for a slot on the schedule. This design also has the potential to drastically improve the repeatability of conducting microgravity experiments. Instead of having to wait for the next opening at a drop tower, or the next launch, scientists only need to wait the time it takes to swap a drained battery for a fully charged one.

Throughout this paper we will explain more and more about our design and how we achieved it. We will explain the lengths that we went through to design a product that was safe for the public. We will also provide a user manual that shows the user just how to operate the drone so that by and large they will know just what to do.

Additionally, we have learned a lot through this project about working as a team. We had to update and modify our specifications and requirements after meeting with our mechanical and aerospace engineering teammates but that has only made our project better. We also had to coordinate with our computer science teammates to determine what kinds of systems worked best for both our group and theirs. We had to learn to sometimes let go of what we thought was best and to listen to their ideas and feedback just as they listed to ours. We made sure that we were as responsive and as helpful as we possibly could be.

2 Project Description

In this chapter we discuss project motivation, goals, objectives, requirements specifications, provide our house of quality analysis, block diagrams, and user operations manual. This chapter gives additional background on our project such as why it is necessary, the problem that it is solving, and the market that we see using our product. We also discuss our goals and specific requirements, some self-created and others given to us by the customer; these are all summed up in the house of quality analysis figure for a quick glance. We also provide high level block diagrams for the hardware and software of our project. Additionally our user operations manual is in this chapter to educate the user of our product on how to properly operate it.

2.1 Project Motivation

A few different methods and products capable of allowing scientists to conduct experiments such as particle collisions and chemical reactions in a microgravity or reduced-gravity environment currently exist. This section aims to discuss the advantages and disadvantages of each, and to point out the motivation behind our design.

The first method is to send the experiment materials to the actual microgravity environment of space aboard a rocket. This method, while the most obvious and by far the most expensive of the methods we will discuss, produces the best quality results, since the experiment is actually conducted in the desired environment. The enormous price and difficulty of sending such experiments into space make this option more or less impossible for any scientist or company without the proper connections. This is the problem that we aim to solve. We want to make the execution of scientific experiments in a microgravity environment more accessible so that we can learn more about how things react when the force of gravity is less of a factor.

The next method we will look at is most simply referred to as a drop tower. The concept of a drop tower in the realm of reduced-gravity experiments is to drop the experimental apparatus and enclosure toward earth, allowing it to experience free-fall. As the apparatus accelerates downward, the normal force acting on the experiment materials inside it will decrease, resulting in a simulated reduced-gravity environment. The apparatus is usually enclosed in some kind of shell that reduces the effects of air resistance, which increases the quality of microgravity because the downward acceleration of the enclosure can more closely match the acceleration due to gravity. Capable of achieving a simulated gravity on the order of 10^{-6}□, this method actually achieves the most significant reduction of gravity of the methods that will be covered in this discussion [1]. Another advantage to the drop tower is that the experiment materials payload is only limited by the design of the drop tower. Even with the additional cost, if any, of using a drop tower built for large experiment payloads, it would likely still be much cheaper than the majority of other available options. However, this method is not without its drawbacks. As a drop tower is not mobile, experimenters must make the trip to the drop tower site with their experiment materials. This presents a problem because there are not many operational drop tower sites that are available for use, meaning there is a good chance that conducting an experiment or series of experiments could require an out-of-state or even out-of-country trip. This is neither cheap nor convenient for scientists, and therefore not a viable option for the vast majority. In short, the use of a drop tower for conducting experiments in a microgravity environment provides a relatively low-cost, high-quality, mid-scale platform for time spent in microgravity and accessibility due to scheduling and traveling difficulties.

Next, we will look at the option of conducting experiments in simulated microgravity environment via parabolic flight. The most famous example of reduced-gravity aircraft is NASA's "Weightless Wonders", also known as the "Vomit Comet". Zero Gravity Corporation also operates reduced-gravity aircraft [2]. This method starts with such aircraft by flying upward at a forty-five degree angle. Once the target altitude is reached, the pilot reduces thrust and levels the nose of the aircraft to start the simulation of reduced gravity. It must also be noted that to reach the target level of microgravity, the aircraft must maintain an engine thrust that perfectly counteracts the effects of drag due to air resistance. Reduced-gravity is continued through the apex, and on the way down as well, as the aircraft points downward and uses engine thrust to match Earth's acceleration due to gravity to the best of its ability [3]. This parabolic flight allows for approximately twenty-five seconds of reduced-gravity time. This experiment window is much larger than what we saw with the drop tower option. During these flights, the parabolic flight pattern is usually repeated forty to sixty times, giving the potential for a large amount of data to be collected. This method, however, faces similar challenges as those of the drop tower, as the scientists must travel and secure space for their experiment materials aboard the flight. A flight aboard a reduced-gravity aircraft is also impractical for most researchers because of its price, typically costing over $5000 [4]. This option trades good-quality microgravity and high amount of data collection per experiment for difficulty of accessibility and extreme cost.

The shortcomings of the previous methods for simulating the levels of reduced-gravity suitable for microgravity experiments provide the motivation behind this project. We recognize the significant effects gravity has on scientific experiments such as particle collision and chemical reactions and the need for continued research in these areas. The problem is, as we have stated, accessibility of the resources necessary to simulate suitable environments for testing. We propose to solve this problem with the use of an unmanned aircraft system, or drone, as our vessel. Given the increasing popularity and availability of drones, we believe we can achieve a good quality of reduced-gravity simulation at a relatively low price when compared to previous methods. After discussions with our interdisciplinary counterparts on the project, we have decided to take an approach very similar to the parabolic flight previously discussed. Given our potential size, weight, and maximum altitude limitations, this method should maximize the microgravity window for experiment execution. Our flight pattern will be discussed in further detail in subsequent sections. In addition to low-cost and good microgravity quality, our design will allow for a quick turnaround time between experiments, with the only limiting factor being the time it takes to recharge the battery. This period could be cut down even further if the experimenter has a spare fully charged battery to swap with the drained one. The trade-off here is that our available experiment materials payload size will be reduced. Successful implementation of our design would allow researchers to continue to advance our knowledge of the effects of a microgravity environment.

2.2 Objectives

- The system will be capable of simulating useful levels of reduced-gravity in an environment suitable for the on-board execution of microgravity experiments. It is understood that a low-budget system such as this may not produce the same level of microgravity attainable with more expensive and proven methods such as the drop tower or parabolic flight. However, as long as the implementation of our design provides a usable and repeatable level of microgravity, this objective will be satisfied.

- The system will be low-cost. Our intent is to make this system accessible to any scientist, group, or company with the desire to conduct experiments in a reduced-gravity environment. A low-cost system available to the masses would open up the potential for a much larger pool of microgravity research.

- The system will be capable of simulating periods of reduced-gravity that are long enough to allow for complete execution of discrete experiments. An inherent challenge with our approach is a limitation on our maximum altitude, set by the Federal Aviation Administration. This has a significant effect on the amount of time our system can successfully simulate microgravity. This altitude limitation, however, will not keep us from achieving our objective, as some experiments can be performed in as little as half of one second, which is much smaller than our target period [5].

- The system will be safe and easy to use. We intend to make this system safe and easy to use so that anyone with the desire to use it to conduct microgravity experiments can do so safely, regardless of their technological skills or drone experience. We do not want anyone to reject the use of our product due to intimidation. We must also ensure that our system complies with any government organization-mandated safety regulations. These regulations and standards will be detailed in a later section.

- The system will allow for storage and retrieval of experimental materials data, such as acceleration, gravity, and temperature. Storage of this data is crucial, as it allows for a thorough analysis by researchers, as well as serves to verify that our goal microgravity environment conditions were actually achieved during the experiment.

- The system will allow for on-board video recording of the events occurring in the experimental chamber. This goes hand in hand with the need to store sensor data. The experimenter must be able to see how the experiment reacted to the absence of normal Earth gravity. Combining the recorded video with the stored sensor data, researchers can get a more complete picture of exactly what happened during their experiment.

- The system will be able to stay connected for data transmissions of up to 400 feet using WIFI. This is necessary to send the large amount of data required by experiments that would use this microgravity system at the speeds that would make the data viewable live.

- The system will be capable of reverse playback and live stream of data. This will provide an additional layer of redundancy and exponentially increase its usability for demonstration. This will be mimicked with playback of the stored video recording and sensor data. If packet loss in live stream the onboard stored data will be able to recover the lost data.

2.3 Requirements Specifications

The requirements and specifications are as follows:

- Drone able to reach altitudes capable of sustaining microgravity testing environment lasting at least 5 seconds.

- Drone capable of consistent upward and downward acceleration while carrying test load of at least 1 lb, with dimensions of approximately 7-1/2" wide x 5-1/8" tall x 14-3/8" long.

- Drone power supply with sufficient charge to allow drone to accelerate to target altitude with enough momentum to keep climbing after power to motors is switched off, cut power to motors to start simulation of reduced gravity, make a thrust-assisted descent, and recover from downward acceleration with enough time to avoid damage from ground impact.

- High capacity and lightweight external power supply capable of supporting multiple sensors and experimentation equipment.

- Light-weight and power efficient sensors for observing various properties around the drone and experimental payload.

- Onboard processing component to read, store, and transmit sensor readings and flight data to a ground station.

- Low-cost and small-size camera for visually observing experimental payload which is minimally able to support 720p video.

- Stable WIFI connection at 400 feet.

- Live stream and display of experimental data

With an open ended list of project specifications in regards to the drone-based microgravity experiment, several of the requirements are rough estimates and target goals in order to meet or exceed a “minimum” standard. These include the minimum sustained testing time of 5 seconds for a success and an optimal goal of 9 seconds which would be challenging due to the limitation in drone altitude achievement, and the test load weight of at least one pound which is ideally the most weight available to a large drone not performing a parabolic flight pattern. For testing purposes this weight would be the absolute ideal and include housing for the experiment, but a more specific requirement would simply be to verify microgravity can be achieved during testing. As for the other requirements, the need for both power efficient and lightweight sensors and testing equipment is paramount as multiple sensors are required by the given documentation not including the sensors necessary for the drone to operate remotely. For the payload environment specifically: temperature, drag, acceleration, and velocity sensors are all possibilities for inclusion with acceleration being a specific requirement as well as a camera to capture and transmit live video data. Secondarily to low-weight and low-power drain components, low-priced parts are key to meet the background requirement of a cheap and reusable system for microgravity testing which is generally not readily available for a low cost.

The requirements and specifications articulated above and on the previous page are applicable to our final, full-size microgravity drone prototype. Here, we will also discuss our requirements for our test drone configuration. The purpose of the test drone configuration is to provide us with feedback from our electrical systems and sensors so that we can verify that these systems are working properly and as intended. This test drone configuration will also serve as our contingency requirements verification vehicle in the unfortunate event that the final, full-size microgravity drone prototype is fatally damaged or otherwise unusable.

The test drone configuration will provide us with sufficient data to verify that both the built-in flight controller accelerometer and the high-precision external, gravity-sensing accelerometer are working properly and reliably. The test drone configuration will have the ability to save accelerometer data to on-board memory. The test drone configuration will have the ability to provide greater than 2 seconds of reduced gravity for experimental testing. The test drone configuration will be capable of providing reduced-gravity levels of less than 0.1g. The test drone configuration will have the ability to reach altitudes greater than 300 feet. The test drone configuration will be capable of a sustained flight time greater than 5 minutes under normal flight conditions, and greater than 15 seconds of flight during parabolic, reduced-gravity experimental flight patterns.

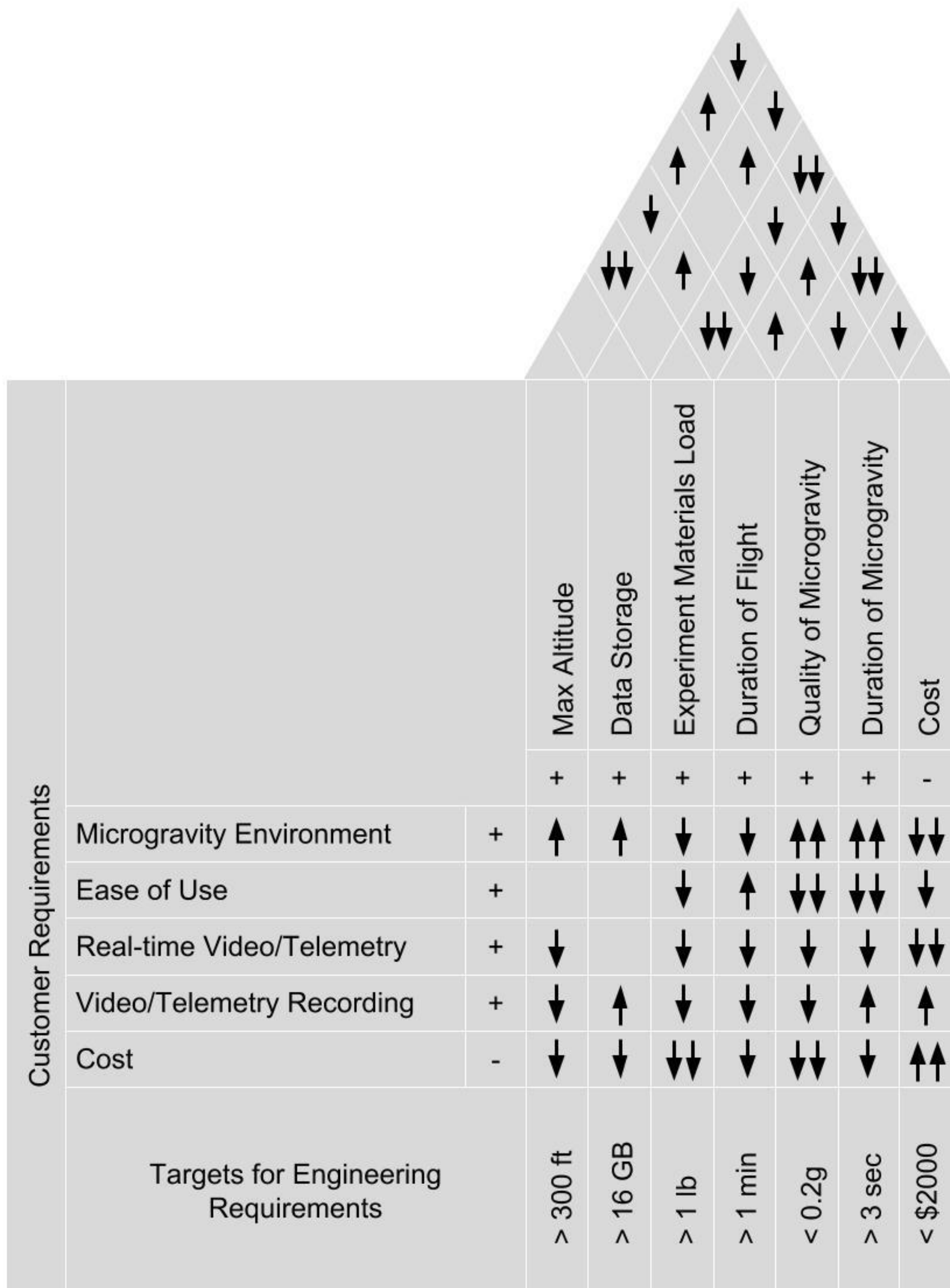
2.4 House of Quality Analysis

The house of quality diagram shown in Figure 2.4.1 shows our target engineering requirements based on the needs and wants of the customer. We made the all of the targets measurable and made sure that some of them could be demonstrated in front of our faculty review board within ten-minute time frame. By doing this we were able to successfully show that we met our engineering requirements and succeeded at the project.

As can be seen in Figure 2.4.1, the House of Quality diagram, the maximization of any one of our engineering goals comes at the cost of others. Compromises must be made between market requirements and realistic engineering requirements in strategic areas in order for our project to be successful and to stay within budgetary constraints. The guidelines set by the Federal Aviation Administration suggest a maximum altitude of no greater than 400 feet [6]. This limitation is unfortunate because it decreased the amount of time we were able to keep the experiment materials in a simulated microgravity environment. Another important factor we considered was the quality of microgravity achieved during the experiment. Increases in the target quality of microgravity result in a more challenging project, and thus a higher cost to us and, ultimately, to the individual or group that will use the system. The requirement of having a live experimental video feed on the drone presented another challenge. The implementation of the live video feed required an additional dedicated microprocessor to handle video data processing and packaging, and sending this data to the transmitter. This added data processing and transmission added to the cost of the system, as well as put additional strain on the drone’s power supply until compensated for adequately during battery selection.

The House of Quality diagram helped us keep track of these challenges and remind us of the tradeoffs we faced. It enabled us to stay focused on exactly what we needed to deliver without getting side-tracked on what might look nice or be cheaper. By ensuring that every component we selected was able to meet the target engineering requirements laid out by the House of Quality diagram, we were well on our way to a successful project. As such, we put a lot of thought into what engineering requirements we could realistically deliver given our constraints while still meeting the customer’s needs.

Figure 2.4.1: House of Quality Diagram

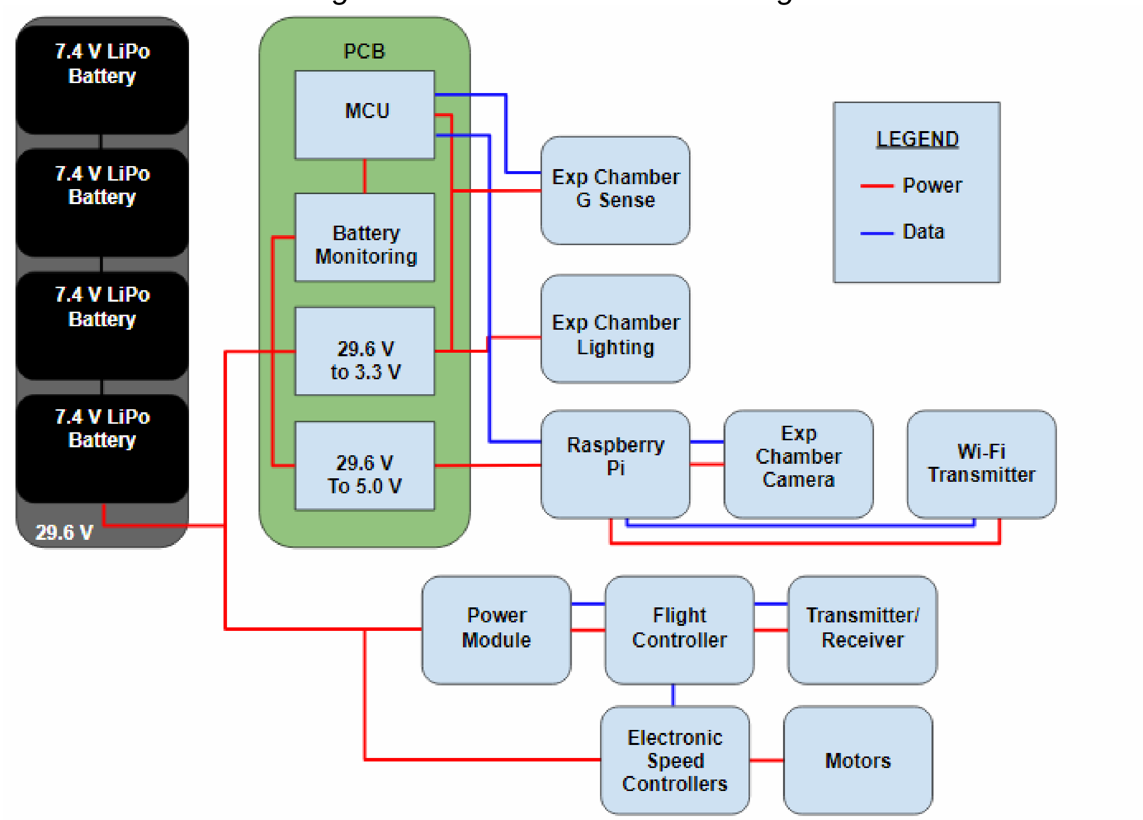


We also had to make sure that our targets for engineering requirements were actually measurable. This posed a bit of a challenge since our project must be done outside since the drone has to go quite high, and quite fast but we made sure that we were able to demonstrate at least some functions of it indoors. After brainstorming we were able to list engineering requirements that could be measured indoors as well.

2.5 Block Diagrams

This section includes two block diagrams, namely, a hardware block diagram and a software diagram. Additionally, a brief explanation of each block diagrams is given along with some insight as to the thinking behind the block diagrams. We found that creating block diagrams really helped us to see what needed to be done. This is because although the block diagrams are very high level and do not provide specific technical details, they provide a good overview. Seeing the different parts of the project laid out in the block diagrams also allowed us to divide up the leadership of the various parts of the project.

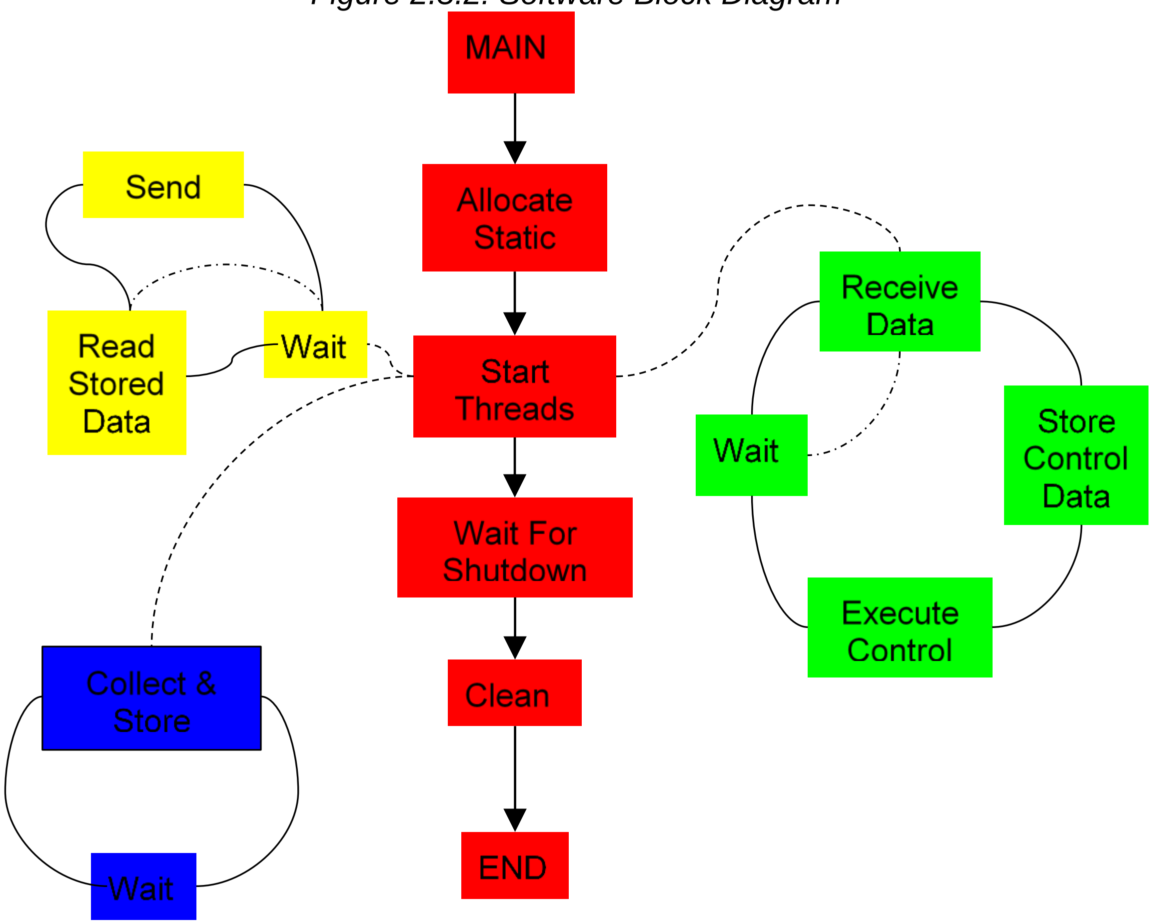
Figure 2.5.1: Hardware Block Diagram



The hardware block diagram shown in Figure 2.5.1 shows the projected components and how we initially expected them to connect to one another. Each block is also color coded to show which team members were responsible for each component and the main leads on it. This was done so that nothing would fall through the cracks and so that each component would be worked on and done on time. The tasks were broken up as to utilize the strengths of each team member and to ensure that the job was done right.

The software block diagram shown in Figure 2.5.2 is a high-level overview of how software will be programmed. The diagram shows what needed to be accomplished by the software due to the various components and gives an outline of how that was achieved. We created this diagram because we knew that having the big picture in mind when designing the software would prove helpful as it would increase the likelihood that nothing would be left out or forgotten. However, this diagram just shows our initial thoughts on how the software should be structured and was subject to edit and change as we move forward.

Figure 2.5.2: Software Block Diagram



2.6 User Operations Manual

In this section directions on how to use our system are provided along with safety information and warnings. We realize that this is a very important section because no one likes receiving a product with no instructions! As such we have done all that we can to ensure that our customer will be able to figure out how to use the drone properly by reading our user operations manual. This is also imperative for safety, since without an operations manual someone might try fiddling around in order to get the drone to work and injure themselves or others. As such we provide safety warnings and instructions when it comes to operating and handling the drone below.

Furthermore, we have made it easy to follow and have provided numbered lists in order to make it simple to read. By making the user operation manual as simple and straightforward as possible we will increase the likelihood that the user of the drone will actually read it. Although this is not actually our responsibility, by increasing the chance that the customer will read the operating instructions we are going a step further to provide the public with safety. This is because if the user of the drone reads the operation instructions then they will know the safety measures that they need to take and as such the public will be safer. The detailed user operation manual can be found below.

Warning

The drone is an electrical device with various electrical components. It is not to be operated in the rain or stored improperly. The drone should be stored as is detailed in this manual. Additionally, the batteries of the drone are to be properly charged according to the details provided in this manual. Failure to do so could result in fire and injury or even death. The drone has propellers that will be spinning very fast when the drone is in operation and can cause injury. Please keep hands clear of the propellers when the drone is powered on. Additionally all safety recommendations detailed in this manual should be followed in order to maintain the safety of yourself and others.

Safety

The drone should never be operated in an area of high traffic. The drone could fail for some reason and fall down onto someone which could cause serious injury due to the size and speed of the drone. The drone should instead be operated in an open field that is either on private property or secluded so that there will be a low amount of people around the testing area. This will ensure that some unsuspecting bystander will not accidentally walk right through the testing area and be in potential danger. Additionally, those doing the research themselves should be careful when performing experiments. Researchers should never walk underneath the drone while it is in operation.

Furthermore, [Figure 2.6.1] below shows an example of a safety sign that can be set up to make sure that everyone is aware of the area where the drone is being tested since it will be going hundreds of feet up in the air and may not be seen.

This will not only aid in maintaining the safety of those around the drone, but will allow for protection of the researchers as a sign like the one shown will alert those around of the experiments being done.

Figure 2.6.1: Example Safety Sign

Storing the Drone

Good storage is essential for continued predictable operation of the device. The drone will need to be stored in a relatively cool and dry place. The drone needs to be cool, if not, the electronics onboard can be severely damaged causing unpredictable functionality. The same could occur with a large amount of moisture as not only could it cause damage to the metal and motors, but it can far more easily cause shorts on the board destroying the electronics and causing unpredictable function. Removing the battery while not in operation is another big thing as it will not only cause the battery to last longer but it will help prevent accidental use and prevent unpredictable use if damaged since without power the drone will not run.

Operating the Drone

Taking the drone out of storage and starting the operating setup can be achieved in a rather easy set of steps:

1. Place the drone on a flat surface
2. Make sure to have an SD card inside the raspberry pi

3. Connect the telemetry transmitter to the flight controller and point the antennae appropriately
4. Connect the WIFI transmitter to a raspberry pi USB port
5. Place and secure the battery to the drone in the appropriate position
6. Connect the battery correctly, positive (red) to positive and negative (black) to negative.
7. The drone is now setup for standard operation

Steps for Ground Control and Operation:

1. Turn on your ground control computer
2. Make sure you have the appropriate software installed
   a. Mission Planner http://ardupilot.org/planner/index.html#home
   b. Custom drone Software for live experimental data viewing and playback
   c. Make sure all software is up to date
3. Attach the USB telemetry transmitter to ground control computer
4. Check your telemetry connection to the drone using Mission Planner
5. Connect the WIFI transmitter to the USB port on the ground control computer
6. Check your WIFI connection to drone by using the custom drone experimental software. Live experimental data should be viewable.
7. Make sure the drone software is up to date.
8. Map your waypoints (MAV Link commands) using Mission Planner.
   a. There is a standard template of waypoints to use to achieve microgravity.
   b. Using your own set of waypoints is not guaranteed to achieve microgravity.
9. Start execution from the Mission Planner console
10. View live experimental data
11. Execute a landing and carefully shut down the drone

Steps to Verify the Experiment Data:

1. Disconnect and remove the drone battery
2. Remove the SD card from the raspberry pi
3. Use the custom experiment software to verify and place any data missing from the live stream in its appropriate spot. Export the experiment data file and log it in the connected database.
4. You can now remove the experimental data from the SD card and place it back in the drone ready to run again
5. If a second run is needed place the battery back in the start at the ground control and operation steps if not reverse the drone setup and place it in a good storage space described above.

Note: If all of the steps listed in this user operations manual are not followed the drone may not operate correctly and injury could result. Please read the whole manual carefully and adhere to the safety warnings, operational, and storage information. Furthermore, this drone is intended for scientific experiments and is not for use in any other application.

3 Research Related to Project Definition

In this chapter we discuss other ways that a microgravity environment can be achieved for scientific testing. We discuss how these methods relate and are similar to our project or how they differ and why our drone based design is needed. Additionally we breakdown the relevant technologies and how they apply to our project. Such as flight controller, flight control, power source, communication, and sensor and navigation technologies. We explain a little bit about these technologies and how we used them to our advantage. Next we go into strategic components and part selections, we discuss how we selected each component and why. We explain the significance of each component and utilize tables to compare their features. So all in all this chapter gives some background on the field of our project and introduces the foundation for our project both in theory and physical part selection.

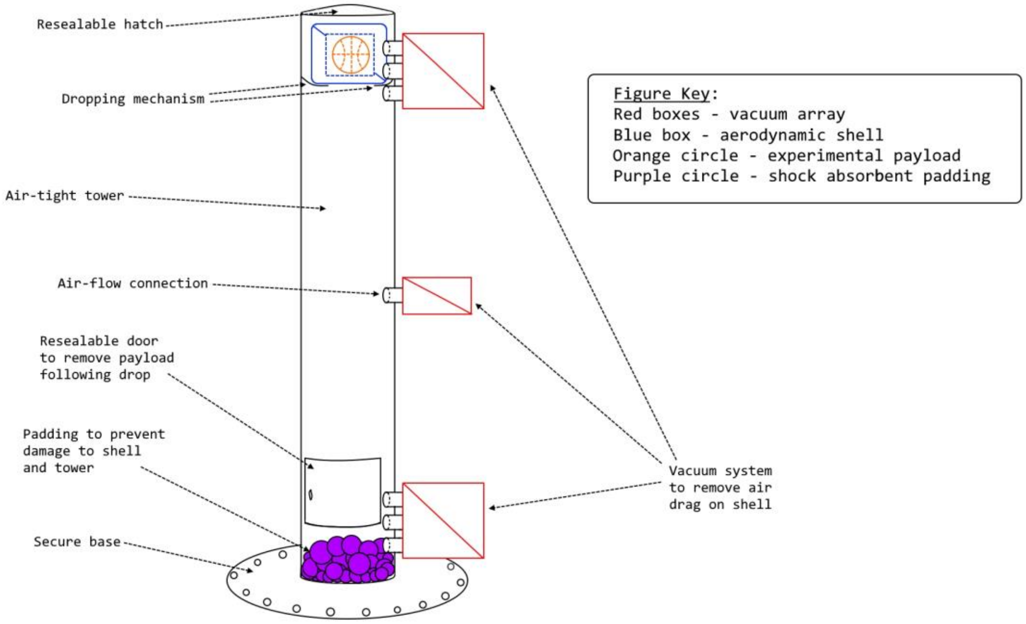
3.1 Existing Similar Products and Projects

In this section we explore similar products and projects that are already in existence. We describe them in detail and point out their restrictions and limitations. Such similar products are the Bremen Drop Tower, ZERO-G parabolic flights, and ultimately of course testing on the international space station in space. The purpose of this section is to provide further background on the current options for doing experiments in microgravity testing. Furthermore, the need for our drone will be clearly seen as the shortfalls of each method are pointed out and explored.

In order to perform successful and effective microgravity testing on Earth, strict conditions must be met involving expensive equipment. One method for performing these experiments involves a large, airtight tower which is vacuumed sealed before experimental equipment is dropped down the tower. For longer durations of induced microgravity, the drop tower must be very tall and have a large fall-space to allow for various equipment sizes.

In the image below [Figure 3.1.1], a conceptual representation of a microgravity drop tower is shown to illustrate how these experiments are performed. In this example an airtight tower has a payload placed within it at the top of the tower on a collapsing platform. Then a complex array of vacuums removes all the air in the tower to reduce air drag to as close to zero as possible, with any minor drag remaining being compensated for by an aerodynamic shell encapsulating the experimental payload. Once the tower has no more air, the payload is dropped down the tower to land on shock absorbent padding in order to prevent damage to the shell or the bottom of the tower due to high velocity impact.

Figure 3.1.1: Conceptual representation of a drop tower microgravity setup.



Permission not necessary to use this figure as it is open source content.

Currently, The Center of Applied Space Technology and Microgravity (ZARM) in Germany is the top facility for this exact kind of testing, with the world's largest drop tower, the Bremen Drop Tower, standing at 146 meters (479 feet) tall which allows for a microgravity testing time of approximately 4.74 seconds [9]. They have also implemented a method of "catapulting" a capsule from the bottom of the tower all the way to the top in order to effectively double the testing time to 9.3 seconds.

ZARM's facilities are open to any researchers to use as long as their experiments are approved by facility staff. Bremen Drop Tower can only be operated safely three times per operation day, which is only four days out of a week with Mondays reserved for tower maintenance. A combination of these two factors has led to the intense overbooking of the facilities at ZARM, with their own web page including the line "nowadays the Bremen Drop Tower is usually booked out for one year in advance". This means that researchers must get their experiment approved and wait around a minimum of one year in order to test at ZARM, not to mention the need for transportation and rooming in Germany. Rooming may even need to last over an entire "campaign" which ZARM defines as "8 to 24 drops [over] 1 to 3 weeks [at] two drops a day". Many of these restrictions have created a tremendous issue for researchers seeking long-term experiments as 9.3 seconds may not suffice as far as testing goes. Another complication with the Bremen Drop Tower is in their precise calculation requirement in order to use the catapult system and its capsule. There are many other rules and regulations specified in their 42 page user manual which must be strictly met in order to use the tower. It isn’t that other systems will be equation or regulation-less but these are specifically quite lengthy and complex and include other systems which are not native to a specific experiment by researchers.

Bremen Drop Tower and other similar drop towers are not the only method for simulating microgravity while on Earth. With proper calculation and an experienced enough pilot, a parabolic flight maneuver can be performed during which microgravity will be achieved. Various companies can provide this service but one of the most relevant would be the Zero Gravity Corporation, or ZERO-G, a company with a modified Boeing 727, known as G-FORCE ONE, which has taken the UCF Aerospace Department, Microgravity Laboratory, and Physics Department on these flights [1]. ZERO-G provides microgravity testing which include up to 30 parabolas per flight with parabolas that simulate Martian, Lunar, and zero-gravity conditions. In order to test on G-FORCE ONE, various conditions must be met and there is a cost associated with their services as they are a private corporation. To reserve a single day of flight for one person, and experimental equipment which is hand-held or free-floating, the cost is $14,300 plus 5% tax. For larger experiments requiring equipment to be setup in an area, a 10’x10’ space (a section) which includes seating for 5 people, the cost for one day of flight is $38,500 plus 5% tax. These costs only grow as more days of flight are added, up to four days of flight for one seat is $24,310 plus 5% tax, while a section is $131,000 plus 5% tax. Many researchers may not have this kind of money especially when they are just testing smaller experiments that do not necessarily require 100 sq. feet of space to test but are too large to be held in hand. Perhaps they are small enough to be handheld but the cost is still far too high at a minimum of $14,300 not including tax. With well-funded researchers, flights such as those provided by ZERO-G provide microgravity experiment durations between 20-30 seconds per parabola which can happen 15-30 times depending on conditions. A drawback, however, would be the requisite airspace to perform these maneuvers is 10 miles. This limitation specifically keeps other companies from providing this parabolic flight maneuvers as they must have excessive starting capital and get various permissions from the FAA.

Earth is not the only location to test microgravity conditions, as the condition is already prevalent in space. Testing on the International Space Station is also possible and is already done by NASA and various other space agencies. Researchers will not be able to test their experiments without the approval of these space agencies which may be excessively hard to achieve considering the intense limitations to weight, space, and flight availability. There’s also no way for researchers to test their experiment firsthand, and any data which is physically collected or observed would need to be collected by astronauts on the station. Thus, it is entirely impractical to test microgravity for the average researcher in space, which is where solutions such as ours become much more practical.

3.2 Relevant Technologies

In this section technologies that are relevant to our project are explored and discussed; technologies such as flight controller, flight control, power source, communication, and sensor and navigation technologies. The technologies that are heavily used in our project are discussed more and broken down into subsections to give a clearer overview of how they relate and why we went with the technology that we did. Furthermore, the technologies that aren't as directly related to our project but are still pertinent to our project, we explore and briefly overview them.

3.2.1 Flight Controller Technologies

The flight controller is the brain behind drone flight. Flight controllers perform many functions to allow drones to fly predictably and safely. One of these functions is to control how fast each of the rotors spins, which steers the drone and controls its altitude and speed. This function can also be accomplished by an electronic speed controller (ESC), which will be discussed later in further detail. Flight controllers exist in different levels of sophistication, with the cheapest and simplest versions costing in the order of 10 dollars. As with the more expensive flight controller configurations, these may or may not include amenities such as a magnetometer, or compass, for directional sensing, accelerometers, and GPS. As can be expected, the precision of such modules is lower than what the more expensive options can provide. At the other end of the spectrum, the most expensive flight controller options may include features like different flight modes, such as GPS lock, altitude lock, orientation mode, and a non-stabilized manual mode [8].

Table 3.2.1.1: Flight Controller Architecture Comparison

Features	F1	F3	F4	F7
Example Chip	STM32F1036BT6	STM32F303CCT6	STM32F405RGT6	STM32F475VG
Processor Speed	72 MHz	72 MHz	168 MHz	216 MHz
Number of UART	2	3	3-5	8
Flash Memory	128 KB	256 KB	1 MB	1 MB

There are four main types of flight controller CPU's (Central Processing Units) to consider. We will refer to these different types as “F1”, “F3”, “F4”, and “F7”, as they are commonly called. While different, all of these employ an STM32 processor, which is the brain of a flight controller [39]. In this section, we will briefly outline the differences between these types. This can be seen in Table 3.2.1.1 below, the features of the chips are succinctly put for easy viewing and comparison. This was very helpful in our studies of flight controllers as it helped us to quickly see the differences of the flight controller architectures without having to switch between multiple information sources. This saved us valuable time and allowed us to better allocate our time onto other things.

The table above provides a simple and straightforward comparison of examples of chips that employ the F1, F3, F4, and F7 type technologies. The F1 type has the lowest processing capabilities of the four, and is now considered by most to be outdated. The F3 type has advantages over the F1 such as a dedicated floating point unit (FPU) to allow for faster calculations, and an additional UART to allow for an extra device. The F3 also has an integrated 5 volt regulator, meaning the board could be powered directly from a LiPo battery [39]. Type F4 flight controllers provide much higher processing speeds than F1 and F3, and can have as many as five UART's. A drawback to the F4 is that, like the F1, it does not have built-in inverters on its UART ports, while the F3 and F7 types do. The F7 has the fastest processing speeds of the four types, at 216 MHz, but uses a larger chip to do so, leaving less room for other components on the circuit board. The F7 is also a newer technology, so the functionality of it is somewhat limited compared to its potential, but this will improve as the technology matures.

3.2.2 Flight Control Software

In addition to flight controller selection, there also exists a need to load software with flight control algorithms. One of the most popular examples is the open-source community OpenPilot [8]. Another popular open-source option is ArduPilot, which is capable of controlling many different types of autonomous vehicles including multirotor drones, which makes this a possible candidate [10]. The features of each of these packages are detailed below.

3.2.2.1 OpenPilot

OpenPilot is an open-source flight controller software package designed to work with multi-rotor drones, as well as fixed wing aircraft. When OpenPilot was founded in 2009 by David Ankers, Angus Peart, and Vassilis Varveropoulos, its original philosophy of use was as a learning tool for developers of unmanned aerial vehicle platforms. This software is compatible with hardware such as an inertial navigation system board, a main control board, a GPS receiver, and a 2.4 GHz serial communications ground station link [11]. OpenPilot also has its own custom-designed flight controller board called the AHRS, which stands for Attitude Reference Heading System [12]. As we are considering the OpenPilot software package for this system, we will take a look at the AHRS, as well as other flight controllers like the CC3D that may be compatible with OpenPilot.

3.2.2.2 ArduPilot

Like OpenPilot, ArduPilot is an open-source software designed for use with unmanned aerial vehicles. However, there seems to be more information readily available on the features, compatibilities, and intended uses of the ArduPilot suite. This software is the only autopilot platform compatible of controlling all of the following vehicles: multirotor drones, fixed-wing and VTOL model aircraft, model helicopters, ground rovers, model boats, model submarines, and antenna trackers [10]. The ArduPilot software consists of navigation software, which runs on the actual flight controller, and ground station software to control the vehicle. Examples of this ground station software include Mission Planner, APM Planner, QGroundControl, MavProxy, and Tower. ArduPilot runs on a number of different hardware platforms, which makes this a good potential candidate for our system. These platforms include: Intel Aero, APM 2.X (Atmel Mega), BeagleBone Blue and PXF Mini, The Cube (Pixhawk 2), Erle-Brain, Intel Minnowboard, Navio2, Parrot Bebop, PixRacer, Qualcomm SnapDragon, Virtual Robotics VRBrain, and the Xilinx SoC Zynq processor. APM (ArduPilotMega) is also supported by ArduPilot, but only older software versions [10].

3.2.2.3 Cleanflight

Another open-source software package option is Cleanflight. Similar to ArduPilot and OpenPilot, Cleanflight can be used to control both multirotor and fixed-wing aircraft. An additional note is that Cleanflight is compatible with a variety of motor counts, including tricopters, quadcopters, hexacopters, and octocopters [13]. This is important because while our substitute test drone will likely have four motors, our final working prototype is planned to have six. This software package is also compatible with 32-bit equipped flight controllers.

3.2.3 Power Source Technologies

For our drone to be able to carry the target experiment materials payload through the flight path necessary to successfully achieve our microgravity goals, careful consideration must be taken when selecting a power source. A few different options exist, some cheaper and more popular, while others are more expensive and harder to find. These options are discussed below.

In addition to finding the most suitable battery option to power our system, we must also decide how many batteries our prototype drone will need, if it requires more than one. Choosing the battery cell configuration of the battery we select is also an important issue, as certain flight controllers have limited compatibility across different types. Both of these topics will be further investigated in later sections.

3.2.3.1 Nickel-Metal Hydride (NiMH) Battery

Nickel-Metal Hydride is an older battery technology that has typically been used for making general, more common rechargeable batteries such as AA, and AAA due to their low self-discharge percentage [19]. These types of batteries can be found certain drone remotes that we might be purchasing however they are not the types of batteries we would find to power the actual drone.

3.2.3.2 Lithium-Polymer (Li-Po) Battery

While Lithium-Polymer batteries are a relatively new battery technology for the drone community, they are the most popular battery option that we will consider for our system. Lithium-Polymer batteries come in three different electrolyte technologies: a dry solid, a porous chemical compound, or a gel-like consistency. The gel-like consistency is most common of the three types today, and this gel characteristic makes them less likely to leak electrolytes when compared to Lithium-ion batteries [16]. Lithium-Polymer batteries are lightweight, low-profile, robust and flexible. Also, because of their decreased likelihood of leakage, these batteries are one of the safer options. Disadvantages of the Lithium-Polymer technology include their shorter lifespan (number of usable charge cycles), when compared to Nickel-Metal Hydride batteries, and lower energy density when compared to Lithium-ion batteries. Also, because Lithium-Polymer batteries are more costly to manufacture, they are usually more expensive than the other battery types discussed in this section [16]. Considering the popularity of this type of battery and its lightweight and low-profile characteristics, the Lithium-Polymer will likely be what we choose to power our system.

3.2.3.3 Lithium-Ion (Li-ion) Battery

Lithium-ion is another battery technology we will consider when searching for a suitable power source for our system. Advantages to this technology over others is higher power density, lower cost, and absence of the memory effect. Because Lithium-ion battery technology has been around for some time now, options of this type should be cheap and easy to find. Memory effect is when a battery becomes harder to charge over time [16]. Lithium-ion batteries do have their disadvantages as well. These batteries suffer from aging and could potentially combust, making them a somewhat unsafe option when compared to Lithium-Polymer.

3.2.4 Communication System Technologies

In this section, we will discuss the research we conducted into different available communication system technologies that we will consider for our project. A general overview of this research includes measurement units relevant to data transmission, different frequencies of communication, and various other bits of collected information. Using this information helped to learn more about technologies such as long-range Wi-Fi, mobile networks, 915 MHz transmissions, and Bluetooth communication systems.

3.2.4.1 Transmission Research Summary

For this project to be completed successfully, the drone requires some kind of connection to the ground computer. Whether the connection is used to transmit just drone flight controller telemetry data, or live video of the experiment, that connection is a vital portion of successfully completing the project and required an array of knowledge related to long-range data transmission. Upon initial assessment, the transmission of data was going to be handled entirely via Wi-Fi, including telemetry and live drone perspective flight video. During research into how to make this possible we learned of things such as antenna design and signal strength over a distance in terms of dBi (or decibels-isotropic, units based on a hypothetical antenna which is lossless) [46]. One of the considerations for a potential ground-station transmitter/receiver was a 24dBi parabolic antenna [48], which was determined to be far too powerful for our needs and would likely result in loss of connection if drone maneuvers were spread over a large area [46][47]. Another consideration which was determined to be inefficient for our purposes was a 15dBi omni-directional antenna [49]. This antenna was substantially more powerful than necessary and would also broadcast its signal in a massive area around the drone/ground-station making testing incredibly difficult. An explanation as to why is provided in-depth in section 4.1 Standards, but a short summary is that Wi-Fi signals are not allowed to interfere with each other, and such a strong, directionless signal would certainly interfere with any other signals in range. More research into Wi-Fi transmissions over a long distance [35] was conducted to ensure a proper setup was achieved with relevant information such as a link budget, or an accounting of all gains and losses for a transmitter through a medium [51].

Much of the information which helped us determine what to investigate further came from various forum answers on electronics.stackexchange.com. In one specific answer, the important of link budget is explained in the context of determining range of signal transmission [52]. In this answer post there was also further emphasis placed on antenna design (omni-directional versus parabolic) as it heavily affects transmission range and effectiveness. Additionally, the importance of choosing a Wi-Fi protocol was brought up as packet delay could cause connection timeouts. An explanation of the different units used for measuring transmission power and a “rule-of-thumb” style guide for determining dBi and how it relates to range of transmission [53]. This rule-of-thumb roughly translated a 6dB (decibels, not decibels-isotropic) gain with an effective doubling of transmission range due to a roughly four-factor increase in power, a useful tip even if not perfectly accurate. Also included in this answer was a breakdown of generic Wi-Fi ranges (such as a “standard” router antenna which has a roughly

2dBi gain) and what they mean to transmission area (in regards to vertical area covered versus horizontal). Another useful collection of information related to transmission power and Wi-Fi interaction [54] provided the final bit of anecdotes to learn further on and make more confident decisions.

Although this collection of information was very specific to an example, it added a general sense of information which would be relevant to know in order to make an informed decision. However, even all this information did not give us a solid determination of how signal transmission was going to be handled with regards to a link transmitting live video data and telemetry data. As such, we began examining already commercially available drones such as the Autel Robotics X-Star (and X-Star Premium) to see how a professional company in the field handled data transmission [55]. With an examination of the Autel's X-Star support Youtube channel [56], as well as independent website information [58], and a product q/a thread on the product's amazon page [57], we noticed that video data and control data were not being broadcast on the same frequency. Specifically, video data was being transmitted over a 900MHz frequency while telemetry and flight controls were being transmitted over a 5GHz (or 2.4GHz depending on X-Star model) frequency. The splitting of these signals allows for Autel's drone to transmit up to 2K live video feed at altitudes of 400 feet and variably long ranges [55] (mostly anecdotal due to restrictions on drone flight) without interrupting control or sensor data. With that idea in mind, we began researching what frequencies were usable for transmission of data with regards to video and telemetry data. One source which provided more information on digital spectrum modulation (or DSM), a technology used for digital two-way communication in remote controlled vehicles [59]. In particular, there's a breakdown and comparison of remote control vehicle communication technologies and points for assisting in learning more information on remote communication technologies.

After being pointed in an appropriate direction in regards to transmission frequencies, several transmitters were selected for possible use. Specifically, some candidates for a transmitter/receiver pair for telemetry data we looked into were already compatible with the ArduPilot flight controller we had been considering [60]. Aside from being a fully compatible telemetry set, they were also in the 900MHz frequency band (specifically 915MHz) which meant it would not interfere with video transmission data sent on another band. Using the calculations from the previously undergone research (specifically the research related to transmission range and dBi) a rough estimate of the maximum gain was 20dB [53]. If the relationship between dB and transmission range holds for other frequencies as it would in the Wi-Fi range (such that for every gain increase of 6, the transmission range is doubled) that would be at least an 8-fold transmission distance increase from the base distance of an approximately 0dB transmission. Without solid information concerning this base distance, it is difficult to determine an estimated range, however, we are confident that it will be sufficient as the system is designed for recreational drones and will likely encompass a large range. There were additionally “whip” or compact antennae which were compatible and worth consideration to ensure optimal range as well as minimal weight/size impact [61]. During discussions with our computer science counterparts, we determined that a telemetry transmission set from a larger kit [62] could also meet the full functionality we were seeking as well as maintain compatibility with the ArduPilot flight controller assuming that was our final choice.

3.2.4.2 Long-Range Wi-Fi

Long-range Wi-Fi is one of the technologies we are considering for our project’s communication system solution. The frequency in particular that we will be looking at is 2.4 GHz, due to its popularity among already existing radio-controlled vehicle transmitter/receivers. Frequencies in the Wi-Fi range are potential candidates for our system because they are smaller, cheaper, and simpler to implement than other options. Some listed drawbacks to using long-range Wi-Fi for our communications solution include poor signal penetration - meaning these connections are effectively limited to line of sight - and far less range when compared to GSM (Global System for Mobile Communications) and CDMA (Code Division Multiple-Access) cellular options [35]. If we ultimately choose to use this approach in our system, we can live with these drawbacks. Our drone is only intended to fly in the vertical direction and will stay under the FAA’s suggested max altitude of 400 feet, meaning that, as long as we have use high-gain directional antennas, the needs of our system should be well within the range limitations.

3.2.4.3 Mobile Networks

Another potential communication system technology that we will investigate is the use of a 4G network connection. The advantage to this approach is that the problem of communication range is virtually eliminated. This could ultimately make it possible to control our drone from anywhere a cellular phone call could be received. A computer, cell phone, or data-enabled tablet could be used to communicate with a server to perform data storage and visualisation via the network, and the server would establish network communications with a 4G LTE dongle onboard the drone system [36]. One example implementation of this approach is shown in [Figure 3.2.4.3.1].

The use of a 4G network for communication with our drone is an interesting approach - and while it is possible that the initial implementation could be more difficult than the long-range Wi-Fi option - when finished, this method may provide us with additional functionalities that could make our system easier to use for experimenters.

Figure 3.2.4.3.1: Implementation of 4G network as communication solution

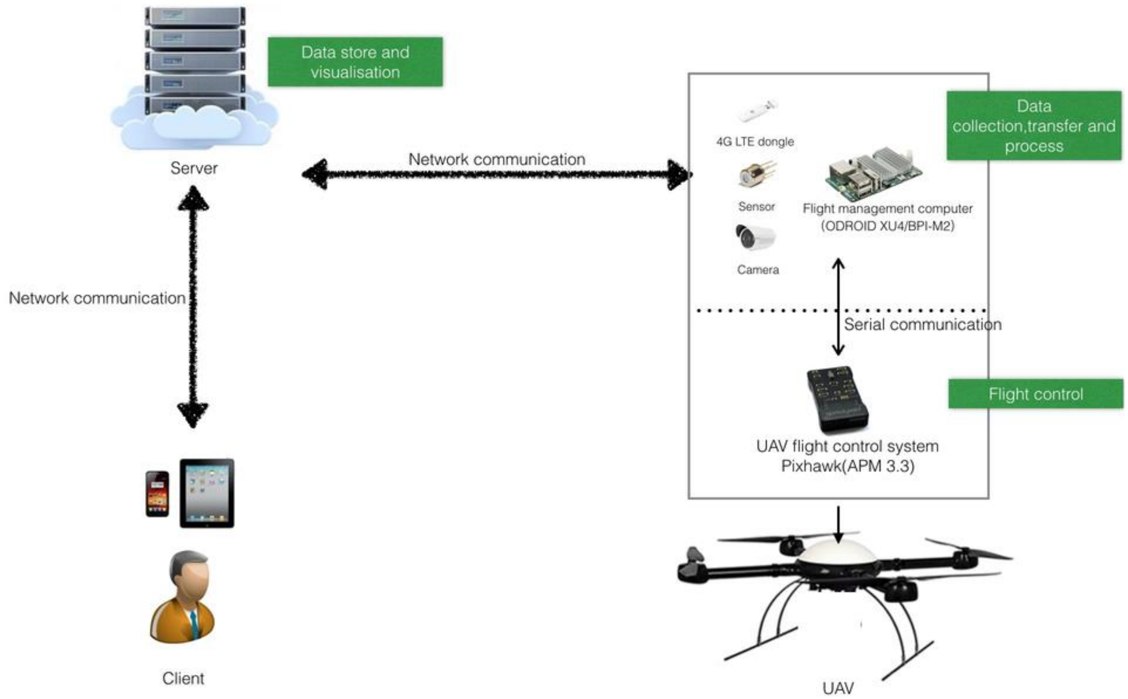


Figure used with permission from WiredCraft.com.

3.2.4.4 900 MHz

Although there are many frequencies in which transmission of data can be achieved, many are regulated by government agencies or limited in their capacity/availability. For frequencies in the 900 MHz band, there is often a wide availability of frequencies in this range as well as enough channels to likely avoid interruptions from other sources. A specific limitation of this technology, however, is the compatibility of devices being directly related to those which can access this frequency or have adapters available to do so. This frequency range isn't as widely used as a technology such as Wi-Fi and is not readily available in devices other than adapters, transmitters, and receivers. As such, we will consider the use of the 900 MHz range for smaller portions of data such as telemetry rather than attempting to convert all data for transfer through this non-standard frequency.

3.2.4.5 Bluetooth

Bluetooth is a another technology we will consider for a part of our overall communication system, as there currently exists some drones that are compatible with it. The primary issue with Bluetooth, however, is the lack of sufficient communication range which would limit control of the drone to being within visual range of the pilot. Even with a significant amount of signal boost it is unlikely Bluetooth will be able to reach high altitudes to control the drone throughout flight.

3.2.5 Sensor and Navigation Technologies

Our system will employ several different sensor and navigation technologies to accomplish our objectives of providing drone flight data such as acceleration, airspeed, and altitude, and experiment environment data such as temperature and level of reduced-gravity achieved. These technologies are detailed in the following pages.

3.2.5.1 Acceleration Sensing

An accelerometer is an electromechanical device that can measure acceleration forces in static form, such as Earth's gravitational pull, or vibrational form, like an earthquake [17]. This technology will be useful because we aim to gather acceleration data during flight. One accelerometer will be used to measure the acceleration of the drone and another will be located inside or near the experimental chamber to gather data on the level of microgravity achieved. There exist a few different accelerometer types - capacitive MEMS, piezoelectric, and piezoresistive - each with their own advantages and disadvantages. In most cases, a capacitive MEMS accelerometer is best for slow or low frequency motion sensing applications [18]. Piezoelectric accelerometers are typically the best option for fast or very high frequency motion sensing, like vibration sensing, while piezoresistive accelerometers are best for less sensitive operations like shock testing due to their usable range.

Aside from choosing the right accelerometer to perform each function using the notes above, we must also take into consideration the frequency response or bandwidth of the accelerometer. We won't be able to get accurate results if our chosen accelerometer does not include the frequency of motion, vibration, or shock that we are hoping to measure. This specification shows the maximum deviation of sensitivity over a frequency range, meaning that, while vibration sensing piezoelectric accelerometers work well at high frequencies, they are likely to be less sensitive or accurate at lower frequencies [18]. Similarly, a capacitive MEMS accelerometer may be able to sense accurately at relatively low frequencies, but sensitivity and thus accuracy drop off as the frequency of acceleration increases. A great way to observe how accelerometers behave is to plot the frequency responses of the different types.

As can be observed in accelerometer frequency response plots, capacitive MEMS accelerometers perform well at low frequencies - all the way down to DC, in fact - but accuracy begins to suffer at frequencies greater than 200 Hz. It is very important that we select the accelerometer type that works best for the functions and conditions that we plan to use it.

3.2.5.2 Altitude Sensing

We will need to employ altitude sensing technology as part of our objective to provide drone in-flight telemetry. To accomplish this, we plan use an altimeter. We will consider two types of altimeters: a barometric altimeter and a GPS altimeter. A barometric altimeter is an instrument that determines altitude based on atmospheric pressure [21]. A GPS altimeter is implemented in one of two ways. The first method takes absolute position in 3-dimensional space and compares it to the ellipsoid math model that approximates mean sea level. The GPS then corrects using a built-in table for the difference between the ellipsoid model and mean sea level [22]. The second method makes the assumption that the altimeter is resting on the Earth's surface and uses a built-in lookup table to determine the geographical surface altitude at that GPS location [22]. Since we are considering altimeters for flight use, it would be illogical to use a GPS altimeter with the functionality of the second method and therefore, this method will be eliminated from further consideration.

Various flight controller options include either a built-in altimeter or a connection for an external altimeter module. Using a flight controller with a built-in altimeter will simplify our objective, however we will need to find out whether or not our selected unit employs a barometric or GPS functionality and test its accuracy.

3.2.5.3 Velocity Sensing

Velocity sensing is a functionality that our mechanical and aerospace interdisciplinary counterparts expressed a desire for, so we will take it into consideration. Velocity sensing or measuring in terms of airspeed has been around for a relatively long time in the form of pitot tubes. A pitot tube is a pressure measurement instrument used to measure fluid flow velocity. This consists of a tube, containing a fluid, that is pointing directly in the direction of fluid flow [23]. Flow velocity, or airspeed, can then be calculated using Bernoulli's equation relating stagnation pressure, static pressure, and dynamic pressure.

As it is our intention to design our system such that the only movement is in the vertical direction - meaning a straight up and down flight - other options may exist allowing us to mimic this functionality. We could add code to our software package that would take the data from the drone flight accelerometer and integrate it to calculate velocity. Another potential way to mimic this functionality would be to use data taken from the altimeter module. Again, since our planned drone flight is only in the vertical direction, we can treat the altitude data from the altimeter as displacement data. If we keep track of the time since drone lift-off, we can divide the measured displacement by flight time to get the velocity data we need - since displacement divided by time equals velocity.

3.2.5.4 Temperature Sensing

We will need to include temperature sensing technologies in our considerations because it is our goal to provide data that tracks the temperature inside the experimental chamber throughout the duration of flight. Various types of temperature sensors exist, and we will discuss relevant and realistic options in this section.

The first type we will look at is a thermistor. The name comes from combining the words THERM-ally sensitive res-ISTOR. A thermistor changes its physical resistance when exposed to changes in temperature. Because this type is usually made from oxides of nickel, manganese, or cobalt coated in glass, they are easily damaged. The advantage of thermistors is their accuracy, repeatability, and speed of response to any changes in temperature [33]. Despite the delicate nature of this temperature sensing technology, the advantages that come with it suggest that we further investigate thermistors as an option for use in our experimental chamber.

Another temperature sensing technology we will consider is the Resistive Temperature Detector, or RTD. These are precision temperature sensors made from high-purity conductors such as platinum, copper, or nickel wound into a coil. The coiled conductor’s resistance changes as a function of temperature, like the thermistor. The output of this type of temperature sensing technology is extremely linear, producing very accurate measurements [33]. The trade-off here is that RTD’s have very poor thermal sensitivity, meaning temperature changes produce very small changes in the output. Although the RTD is capable of producing very accurate measurements, its low thermal sensitivity makes it a poor candidate for our intended use.

The last temperature sensing technology we will discuss is the thermocouple, which is the most commonly used temperature sensor type. Thermocouples are easy to use, have a fast response to temperature changes and a very wide temperature range. This technology consists of two junctions of different metals - copper and constantan, for example - that are either welded or crimped together [33]. One junction is kept at a constant temperature for reference, called the cold junction, and the other junction, referred to as the hot junction, is for measuring. When the two junctions are at different temperatures, a voltage is developed across the junction [33]. This voltage is then used to calculate the difference in temperature.

After researching different types of temperature sensing technologies, we have decided that thermistors and thermocouples will need further consideration for our needs. The poor thermal sensitivity of the RTD technology suggests that it will not be a good candidate for our experimental chamber temperature sensor.

3.2.5.5 Directional Sensing

A magnetometer measures the direction of something with respect to a magnetic field at a particular location [20]. On a drone this location would be the magnetic north pole, it is used to measure the direction and location of the drone with respect to said magnetic north pole. It is an aid to the accelerometer and gyroscope as it is yet another sensor that can determine direction and location. We will definitely want the drone to properly respond to the programmed flight path that it is programmed so that its experimental chamber will experience reduced gravity. For this to be done the direction and location sensors on the drone need to be quite accurate, that is why having a magnetometer in addition to the accelerometer and gyroscope is very useful. As such we are very grateful that our chosen flight controller, the ArduPilot Mega 2.8, includes a magnetometer. We will definitely be utilizing it to the fullest and expect its inclusion to be very helpful.

3.2.5.6 Location Sensing

The inclusion of a GPS module in our system is being considered, as it would provide additional data and functionality. Some of the flight controllers we've researched so far have a kit that comes with a GPS module. With the low cost and potential simplicity of integrating the GPS module into our system, it is worth a closer look. It would provide an added feature that could be potentially very useful and helpful. If it is in the budget and within our time constraints to add a GPS module we will certainly do so. However due to the fact that we expect to be flying the drone in a tight area, it is not necessarily a must have for our project. We must instead first focus on the components that are vital to achieving our goal of providing a reduced gravity environment.

3.3 Strategic Components and Part Selection

In this section we talk about the thought process and research that went into selecting various components and parts for our design. We have tables under most of the subsections that follow which compare some of the different component options we considered for our project. This is so that the types of parts we were considering selecting and the varied features between them can be seen at a quick glance. We also go into detail as to our thought process behind selection and expound upon why we selected the components we did.

3.3.1 Flight Controller Selection

Seeing that our project is centered on controlling a drone in a particular manner so that an environment of reduced gravity can be produced, we knew that selecting a flight controller had to be done with great thought. Furthermore, the flight controller had to be one of the first things we selected since we had to make sure that the other components that we would be connecting to the flight controller were compatible with it. Some examples of such components are telemetry devices, a camera, GPS module, etc. Additionally, the components needed varied based upon which flight controller we selected as some controllers had key features already embedded in them while others lacked said features. With this in mind we began exploring flight controller options with our desired functionality and budgetary constraints as factors in our selection, the details and results of which are as follows.

We started basic with the Crius All In One Pro flight controller and moved our way up to the ArduPilot Mega 2.8, next we considered the PixHawk PX4. [Table 3.3.1.1] below shows the features of the three flight controllers mentioned and was used to compare them to one another. The Crius All In One Pro, ArduPilot Mega 2.8, and PixHawk PX4 all have the sensors desired, a gyroscope, accelerometer, magnetometer, barometer, and are GPS compatible. However the Crius All In One Pro does not have flash memory storage and this feature could be helpful in our data logging process. So we eliminated the Crius All In One Pro flight controller due to the fact that it doesn't have this added feature whereas the other options do. The ArduPilot Mega 2.8 and PixHawk PX4 have data storage capabilities and their capacities can be enlarged by inserting an SD card. So since the features of the ArduPilot Mega 2.8 and PixHawk PX4 are comparable, next the cost of each flight controller had to be considered. The PixHawk PX4, due to its increased amount of features is significantly more expensive than the ArduPilot Mega 2.8.

Table 3.3.1.1: Flight Controller Comparison

The Crius All In One Pro	ArduPilot Mega 2.8	PixHawk PX4
6-axis gyroscope	3-axis gyroscope	3-axis 16 bit gyroscope
6-axis accelerometer	Accelerometer	3-axis 14 bit accelerometer
3-axis digital magnetometer	Magnetometer	6 axis magnetometer
Barometer	Barometer	Barometer
None	Onboard 4 MB Dataflash chip	32 bit 2M flash memory
GPS Compatible	Off-board GPS compatible	GPS Compatible
4 General Purpose Serial Ports	1 Telemetry Port	2 Telemetry Ports

Therefore for our purposes and due to our limited budget, it makes more sense to go with the ArduPilot Mega 2.8 flight controller. However, a further limitation of the ArduPilot Mega involved available telemetry ports. Our Computer Science teammates requested we have two telemetry ports to allow for easier interfacing with autonomous flight commands. Even though we ended up selecting the ArduPilot Mega 2.8 flight controller due to its reasonable price, we ended up switching to the PixHawk PX4 which provided the main features we needed to meet our project requirements and came with two telemetry ports.

3.3.2 Telemetry Sensor Selection

Next we had to decide on which telemetry sensors to get, we had to make sure that we would be able to properly record and transmit all of the necessary measurements. At first we looked into getting a receiver and transmitter separately that included sophisticated telemetry such as the X8R FrSky receiver, and FrSky Taranis X9D Plus transmitter. However upon deeper research it was found that these products would go above and beyond what we actually need for our application and as such were more costly than necessary. So due to budgetary constraints, we began looking for a lower end transmitter and receiver that would still be able to provide the needed telemetry for this project. We found that the telemetry kit that could be purchased in a bundled package with the ArduPilot Mega 2.8 flight controller would actually fit our purposes quite nicely. So we decided on this telemetry kit as it allowed us to save funds in our tight budget. Available telemetry sensors with the ArduPilot Mega 2.8 kit are an external GPS module, an external airspeed sensor module, a built-in magnetometer for directional sensing, a built-in 3-axis accelerometer, and a built-in barometric pressure sensor for altitude sensing. The kit also allows for the use of an external compass, which is included with the GPS module. All of the additional components included with the ArduPilot Mega 2.8 were compatible with the PixHawk PX4 which meant that our initial flight controller selection included all of the sensors necessary for us to accomplish our in-flight telemetry objective despite switching flight controller models.

3.3.3 Transmitter and Receiver Selection

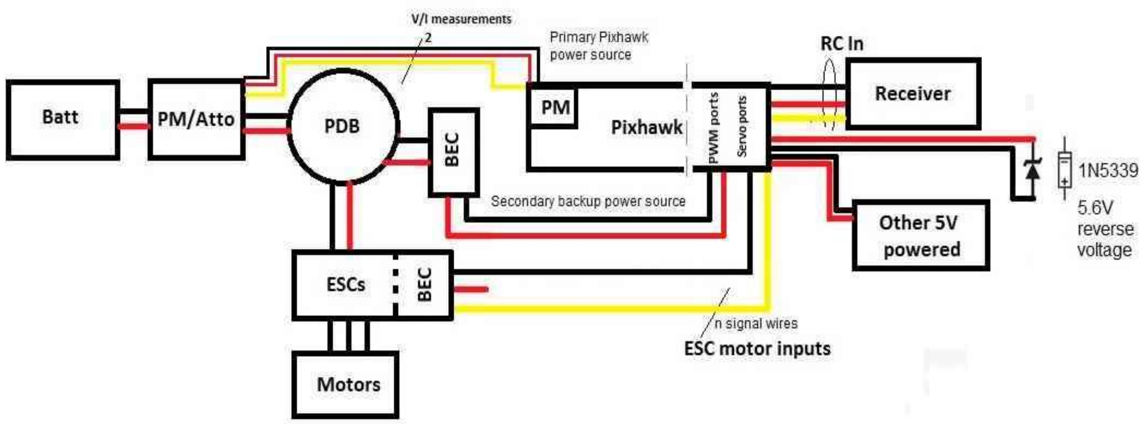
Although we briefly discussed a transmitter and receiver pair in the previous section when dealing with telemetry, we also need to have a remote control to at least initially control the drone. So we began to search for more modest transmitter and receivers that may not have much telemetry capabilities but will be able to fly the drone effectively. Upon searching we found two possible options that seemed to have around the features we needed and that were within our budget. These were the Turnigy 9X 9Ch Transmitter with a Module and 8ch Receiver, and the Flysky FS-i6X 2.4GHz 10CH Transmitter with a iA6B Receiver. They cost $62.12 and $59.40 respectively, they both fit within our budget since we had allocated a total of $100 for the transmitter and receiver and these options come with both so they are actually less than expected. Since they are both within budget and seem to be a good fit for our project, choosing between the two became a little bit of a challenge. We took to the reviews to find out the reliabilities and ease of use of each of the two transmitter and receiver pairs. We found that the Flysky FS-i6X 2.4GHz 10CH Transmitter with a iA6B Receiver seems to be the better choice as the Turnigy 9X 9Ch Transmitter with a Module and 8ch Receiver had a few reviews that said that it does not come with an owner's manual and that it is not for first time drone configurers. Since we are definitely new to setting up transceivers the Flysky FS-i6X 2.4GHz 10CH Transmitter with an iA6B Receiver seemed like the way to go, especially since it is cheaper!

3.3.4 Deciding How to Power the Drone

In this section just how to power the drone is discussed. As can be seen in [Figure 3.3.4.1], there are many components that need to be powered and that figure shows one way of doing it. In this section different options for powering the drone will be discussed and evaluated.

Following the selection of the telemetry kit, we decided to tackle the issue of power. The supply of power to the flight controller is a very important part as it distributes the correct amount of voltage to different components. Without the proper regulation provided by the power distribution board, too much voltage could be delivered to a component and cause it to break. Additionally, seeing that we need to take off very fast we knew that we needed to be very thoughtful in selecting the power components for the flight controller.

Figure 3.3.4.1: ArduPilot Doc [62]



Permission not necessary to use this figure as it is open source content.

With this in mind we began searching for the right way to deliver power for our project. There are a few main ways to go about providing proper power to the flight controller, a voltage regulator along with a power distribution board, a power distribution with an internal voltage regulator, or a power module. As implied, there are different types of power distribution boards, some are powered by vBat while others are not. The boards that are powered by vBat can be directly connected to the battery whereas the boards that are not must be connected to a voltage regulator that is connected to the batter and has stepped down the battery voltage for the power distribution board. Thus a power distribution board that does not run on vBat would create an added need for a voltage regulator and in turn another expenditure. So we checked our flight controller to see if its power distribution board was run on vBat or not, we found that it was not due to the fact that it is usually used in conjunction with a power module. Power modules are useful and actually very helpful because they not only provide the steady stepped down voltage that the flight controller needs, but they also have current and voltage sensors so that a warning can be given when the battery is low. Additionally, when the battery gets below a certain point a safety measure can be triggered that causes the drone to land [8]. This battery monitoring feature proved to be very useful and in fact necessary for our purposes. We needed to know when the drone batteries were dying so that we could end testing and bring it down, especially during testing when we were be figuring out the full capabilities of our design. After discovering that a power module would be the best way to distribute power to the drone, we then had to figure out which power module to use. Upon a quick search it was evident that for the ArduPilot Mega 2.8 flight controller there is a standard power module that will work, it is the APM Arduflyer Power Module V1.0. We didn’t feel the need to continue searching for other power modules since this one was already known to be compatible with our initial ArduPilot Mega 2.8 flight controller as well as the PixHawk PX4 flight controller. Fortunately, we were able to locate a bundle that included not only the telemetry kit we decided on earlier but also the APM Arduflyer Power Module V1.0 for a reasonable amount so we decided on this.

However while this power module would work perfectly for our test drone, we later realized that for the big drone we would need a different power module that would be able to take in over the 33V nominal voltage our batteries would provide. With this in mind we selected the Holybro/PixHawk APM 2.5 Power Module for our big drone as we found that it could handle up to 42V.

3.3.5 Electronic Speed Controller Selection

Table 3.3.5.1, on the following page, shows the three electronic speed controllers that we considered and explored, the Flyduino Kiss 24A Race Edition, DSHOT Bullet 30A, and SPEDIX 30A 3-6S. The electronic speed controllers vary the motor's speed and direction and thus selecting a controller that could effectively and efficiently do what we needed for this project was very important. Both the continuous and burst current rating of the electronic speed controllers are important because we wanted to go as fast as we could for a short period of time so we needed to make sure that the electronic speed controller we selected could handle the amount of current that we were going to be delivering to it so that it would operate smoothly. The weight was obviously a big factor as well as we were looking to reduce the weight of our drone anywhere we could. Likewise the dimensions of the electronic speed controller dictate how much area the controllers will take up and seeing that our space is limited this was also a factor to be considered. We also considered the fact that the input voltage that the controller takes would restrict us when selecting a battery or vice versa. For example, if there was a certain battery that we deemed the best we would have to select a controller that could work with that input. We felt that perhaps the most important feature in selecting a controller was bi-directional support. Our mechanical and aerospace engineering teammates asked us to select electronic speed controllers that were able to allow the current to flow in the reverse direction in order to propel the drone downward at quite a fast speed. Therefore if we found that this was not able to be supported and programmed into the software, then the electronic speed controller would not fit our purposes. With that being said it seemed that the Flyduino KISS 24A Race Edition controller had to be eliminated from our choices as the only way to reverse the motor direction is to physically solder the wires differently. That left the DSHOT Bullet 30A, and the SPEDIX 30A 3-6S electronic speed controllers. When we reached this point in the electronic speed controller selection process we did not go ahead and nail down exactly which electronic speed controller we would use since we wanted to discuss it further with our mechanical and aerospace engineering teammates to see their thoughts. We wanted to make sure that the speed controller we selected could actually perform at a high enough level to achieve reduced gravity and we knew that our MAE teammates could provide valuable input that could potentially save us from having to go back and re-order a different controller.

The electronic speed controllers shown in [Table 3.3.5.1] were initially thought to be good enough for our purposes, this is because when we first set out to select electronic speed controllers we did not yet have details on the motors that our mechanical and aerospace engineering teammates were thinking of selecting. After meeting with our teammates they informed us that their target was to be able to use six motors. We realized that in order to meet this requirement we had to go back to the drawing board and restart the search for electronic speed controllers given this new information since as can be seen in [Table 3.3.5.1], all of the electronic speed controllers we initially researched only supported three motors. Additionally, we garnered more information on the motors from our mechanical and aerospace teammates, they told us that in order to achieve our goal of creating a microgravity environment that they needed to select motors with a max current draw of 65A. So we began researching new electronic speed controllers and the results of that search can be seen in [Table 3.3.5.2]. Having the electronic speed controllers all compared in one table allowed us to quickly and easily view all of the various features and differences in the devices. Doing this allowed is to quickly compare and contrast the various controllers without having to flip between specification sheets.

Table 3.3.5.1: Initial Electronic Speed Controller Comparison

Features	**Flyduino KISS 24A Race Edition**	**DSHOT Bullet 30A**	**SPEDIX 30A 3-6S**
Continuous current rating	24A	30A	30A
Burst current rating	30A	35A	40A
Weight (grams)	3.53 (without wires)	3.9 (without wires)	5.5 (with wires)
Dimensions	19x27mm	19.6x19.6mm	25x13mm
Input Voltage	2-5S LiPo	2-4S LiPo	3-4S LiPo
Bi-directional Support	Only through soldering the wires in the opposite places	Yes can be changed to bi-directional in BLHeli_S program	Yes can be changed to bi-directional in BLHeli_S program
Motors supported	3	3	3
Price	$21.99	$12.99	$11.95

As can be seen in [Table 3.3.5.2] the first two electronic speed controllers we were comparing in the updated search can only handle up to 22.2V, this proved to be a problem since our mechanical and aerospace teammates later informed us that they would need 29.6V of power. With this new information we searched again for electronic speed controllers and found that the Turnigy dlux 40A Mk2 Brushless Speed Controller w/8A S-BEC would meet our voltage needs. As for being able to handle the current draw of our motors, initially our mechanical and aerospace teammates thought that they would select a motor with a current draw of 65A as we mentioned previously. However, the motors they ended up selecting were model KDE4012XF-400 brushless motors with a max current draw of 30.6A. So the continuous and burst current rating of 60A and 70A respectively, of the Turnigy dlux 40A Mk2 Brushless Speed Controller w/8A S-BEC will meet our needs. However although we established that this electronic speed controller should be perfect for our project, we wanted to once again consult with our mechanical and aerospace teammates just to make sure we were all on the same page. After speaking with them and getting their approval we went ahead and ordered six of these electronic speed controller and eagerly awaited their arrival to begin testing with them.

Table 3.3.5.2: Updated Electronic Speed Controller Comparison

Features	TURNIGY TRUST 70A SBEC Brushless Speed Controller	HobbyKing YEP 80A (2~6S) SBEC Brushless Speed Controller	Turnigy dlux 40A Mk2 Brushless Speed Controller w/8A S-BEC
Continuous current rating	60A	80A	**60A**
Burst current rating	70A	100A	**70A**
Weight (grams)	65 (without wires)	70 (with wires)	**80 (without wires)**
Dimensions	61x37mm	50 x 30mm	**81 x 35.2mm**
Input Voltage	2-6S LiPo	2-6S LiPo	**2-8S LiPo**
Bi-directional	Yes	Yes	**Yes**
Price	$31.97	$36.26	**$46.30**

3.3.6 Battery Selection

Subsequently we moved on to battery selection for our drone. In this section our thought process and research steps in selecting a battery will be seen and discussed. Trying to selecting the right battery for our drone was very challenging. There were so many options available but yet there were also so many constraints! We realized that for our project we did not necessarily need a battery that would last a long time since we foresee the end user just swapping out the battery after one or two experiments. These experiments will only last under a minute so a long amount of flight time is not really needed. However we did need the batteries to be lightweight. We hoped that since we were not requiring long battery life that we would be able to find batteries that were lightweight enough to fit our purpose. So we set out to find the perfect balance between our desire to find a battery that could sustain the flight time we wanted while yet being lightweight and compact. This flight time vs weight tradeoff is a common one in battery selection and can prove quite difficult. We hoped that with our low requirement for flight time we could minimize the weight and size of the battery and thereby improve our overall design.

With that being said, battery lifetime and weight were not the only variables involved in our battery selection. We also had to choose batteries that were compatible with our flight controller. We didn’t want to end up with the perfect battery in every regard except for the fact that it was not compatible with our chosen flight controller! We knew that would result in much wasted time and frustration and as such, we looked into what kind of battery would be best for our flight controller. For the ArduPilot Mega 2.8 flight controller, a lithium polymer (LiPo) battery is suggested. So that narrowed down our search a bit although there are still many variations of lithium polymer batteries that we had to consider. The power module we used for the big drone works with a LiPo battery up to 10S so that gave us additional flexibility. Before actually beginning the search for batteries we figured that it would be a good idea to figure out what the different terminology actually means, such as ‘2000mAh’ or ‘50C.’ The capacity of a battery is written as ‘2000mAh’ this equals 2 Amp Hours and by looking at this number you can get a good feel on how much power the battery can hold which extends to its lifetime. ‘50C’ helps to find the discharge rating (or ‘C rating’) of the battery [15]. The discharge rating of the battery will be 50 times the capacity of the battery in Amp Hours. So for a battery with a capacity of 2000mAh and a discharge rating marked of 50C, the discharge rating in amps would be 100; this means that the battery can withstand a maximum sustained load of 100A. We used the word phrase ‘sustained load’ because although the battery can handle more of a load for only a short period of time (burst discharge rating), we are dealing here with the continuous discharge rating.

After learning more about the meaning and implications of the terms used to describe batteries we began searching for the right one. We found it extremely helpful to already have the knowledge of what the various terms used in battery specifications mean. Because of this we were able to intelligently search rather than just blindly looking for a battery without really knowing what the specifications were telling us. The comparisons of a few batteries we considered are seen in [Table 3.3.6.1] below. The differences in features as well as their similarities can easily been seen and compared.

[Table 3.3.6.1] was very helpful to see in an organized manner the different features and offerings of the batteries we were considering. It shows a comparison of the Turnigy 1500mAh 3S 25C Lipo Pack, Turnigy 1300mAh 2S 20C Lipo Pack, and Turnigy 1000mAh 2S 20C LiPoly Pack batteries. We have already discussed the continuous and burst discharge ratings and the meaning that those hold. As such we needed to select a battery that would be able to withstand the sustained load that we would place on it along with the burst load we would place on it for a brief period. We worked with our mechanical and aerospace engineering teammates to figure out exactly what our expected load would be in this regard. We realized that when we had this nailed down we would be able to better weed out batteries that would and would not work but early in our research we were also looking at the other features of the batteries. We saw the large range in weights just in these three batteries, from 59 to 127 grams. That was a big deal for our purposes, since as mentioned previously, we were looking for the lightest possible battery that could help us successfully achieve our goal of creating a reduced microgravity environment. This goal meant that weight was one of the most important factors in our considerations. However with that being said, we still needed to make sure that the battery had enough voltage and capacity for our project. These were also areas where we worked together with our mechanical and aerospace engineering team members to get a good estimate of what the target goal was. As we mentioned initially, battery selection was very challenging as it depends on many variables. Due to this [Table 3.3.6.1] was only our initial, very preliminary battery search results which gave us a good start and helped us to see how the batteries on the market vary in size, weight and voltage.

Table 3.3.6.1: Initial Battery Comparison

Features	**Turnigy 1500mAh 3S 25C Lipo Pack**	**Turnigy 1300mAh 2S 20C Lipo Pack**	**Turnigy 1000mAh 2S 20C LiPoly Pack**
Continuous discharge rating	25C	20C	20C
Burst discharge rating	35C	30C	30C
Weight (grams)	127	81	59
Dimensions	73 x 33 x 27mm	73 x 35 x 17mm	72 x 34 x 14mm
Number of Cells	3	2	2
Minimum Capacity (mAh)	1500	1300	1000
Voltage (V)	11.1	7.4	7.4
Price	$12.63	$4.80	$4.36

After later conferring with our aerospace and mechanical engineering teammates, we discovered that although batteries such as the ones shown in [Table 3.3.6.1] could be used to power the drone, it would require them to pick motors that would not be capable of giving the thrust that we needed. As such, we realized that these batteries, while a good starting point, were not the right ones for our drone. So we began searching for batteries that would be able to properly power the motors that our mechanical and aerospace teammates would be choosing. Based on the specifications that they told us it was determined that two four cell LiPo batteries seemed be best. [Table 3.3.6.2] below shows the batteries that we narrowed it down to and their various features. Comparing the features in a table allowed us to more easily see where the batteries differ and examine which one would be best suited for our purposes.

Table 3.3.6.2: More Powerful Battery Comparison

Features	**Turnigy 2200mAh 4S 30C Lipo Pack**	**Turnigy nano-tech 2200mah 4S 50C Lipo Pack**	**ZIPPY Compact 2200mAh 4S 25C Lipo Pack**
Continuous discharge rating	30C	25C	25C
Burst discharge rating	40C	50C	35C
Weight (grams)	258	218	210
Dimensions	105 x 34 x 35mm	100 x 35 x 30mm	107 x 28 x 34mm
Number of Cells	4	4	4
Minimum Capacity (mAh)	2200	2200	2200
Voltage (V)	14.8	14.8	14.8
Price	$22.02	$20.73	$16.80

As mentioned before, the weight of the battery was very important for our design, we wanted the lightest battery that we could have that would still be powerful enough to power our drone. Looking at [Table 3.3.6.2] above, we see that the ZIPPY Compact 2200mAh 4S 25C Lipo Pack is the lightest and the smallest but it has a burst discharge rating of 35C. At this point we realized that we needed to check with our aerospace and mechanical counterparts to make sure that, that burst discharge rating would be enough for the motors that they would be selecting. We decided that if the burst discharge rating of the ZIPPY Compact battery was enough then we would go with it since it was the cheapest, lightest, and smallest battery out of the three we were considering. However we felt that if the burst discharge rating needed to be higher we had two other options, that of the Turnigy 2200mAh 4S 30C Lipo Pack, and the Turnigy nano-tech 2200mah 4S 25~50C Lipo Pack, since each of those batteries have a rating of 40C and 50C respectively.

These were the conclusions we initially drew having not yet been informed of the final decision on motors by our mechanical and aerospace engineering teammates. We figured that when we were provided with which motors they have selected we will be able to choose a battery quickly. We felt it would be beneficial to go ahead looking for batteries before getting the final motor selection because it would help us to get a feel for the market, and that it did.

However, it is a good thing that we did not select our battery without again speaking with our mechanical and aerospace engineering teammates. Upon speaking with them again they had revised their motor choice and as such provided us with updated maximum current draw numbers for the motors. They informed us that each motor would have a max current of 65A, which meant that for six motors, our battery would need to be able to handle a max current draw of 390A. (Note this current draw was later revised down to 184A due to the change in motors mentioned in the previous section but this was after our batteries had been selected, this is why for the duration of the battery selection section this original value of 390A will be referred to). This was a much larger current draw than we originally anticipated and as such we had to embark on a new search to find battery options that would support that high of a current draw. This was yet another example of the real world preparation that we gained from this project in working together with other departments. Although we had already searched for batteries twice and thought that we had narrowed it down to batteries that best suited our needs, we had to again modify our search requirements and find the best battery for our design. In the real world, designs change and you have to work with other departments and do your best to provide them with their needs. So with this in mind we began a new search for the lightest, smallest battery that would be able to still provide the needed voltage and support the large current draw of the motors. [Table 3.3.6.2] provides a neat and compact breakdown of the various batteries that we found that would be able to support our voltage and current draw requirements.

[Table 3.3.6.3] shows the three batteries that we considered after receiving the new updated requirements, the ZIPPY Compact 5800mAh 3s 60c Lipo Pack, the ZIPPY Compact 5800mAh 4s 60c Lipo Pack, and the ZIPPY Compact 5000mAh 6s 60c Lipo Pack w/ XT90. To determine around what range of batteries we should be looking for, in terms of mAh, discharge rating, and cell count, we had to perform some basic calculations. First our mechanical and aerospace teammates informed us that the optimal voltage would be 29.6 V but that 22.2 V could possibly be accepted. Additionally, we were looking for the lowest weight and size that we could so we decided to look at the possibility of using multiple batteries with a lower cell count rather than one large battery. As for the mAh and discharge rating, we were provided with the graph shown in Figure 3.3.6.1 below. This shows that the total flight time would be less than fifteen seconds and it shows the length of time that the motors would be at full thrust. Given that the burst discharge rating on a battery allows for about ten seconds of current draw at that level, based on our flight time and thrust requirements, we were able to look for a burst current rating that would be able to sustain 390A of current draw. In order to calculate the maximum amount of current that a battery can support we multiplied the C rating by the capacity of our battery in amps. We were also able to get an idea of the capacity our battery needed to have by calculating the flight time that we would get from our battery based on our current draw. This was obtained by multiplying the desired flight time by the capacity of our battery. Doing these calculations for different values we found the range that our battery needed to be in, it needed to have a capacity of around 5000-5800mAh and a 70C-80C burst discharge rating. With that we set out to find potential batteries for our project.

Figure 3.3.6.1: Throttle (%) vs Time (s)

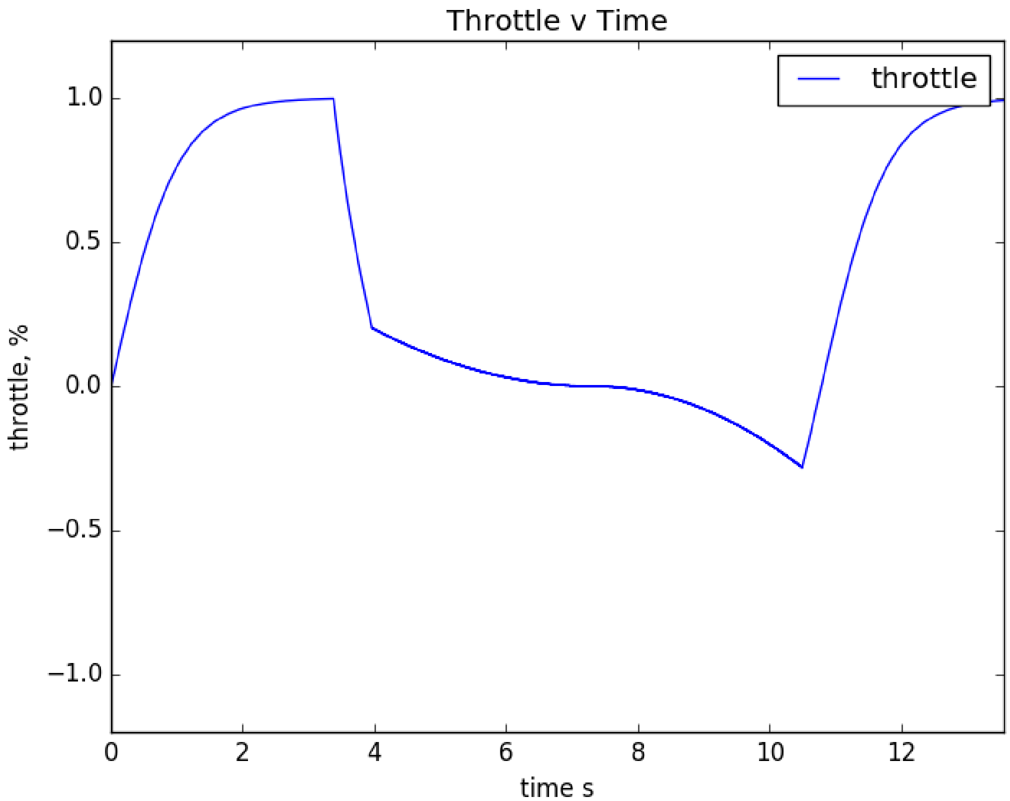


Figure used with permission from MAE team.

Looking at the [Table 3.3.6.3] we found that the ZIPPY Compact 5800mAh 3s 60c Lipo Pack battery had a burst discharge rating of 70C and as such it would be able to withstand the maximum current draw of 390A. Additionally, it was the lightest of the three ZIPPY battery options due to its cell count of only three. We determined that we could use two of those batteries to provide the acceptable 22.2 V so the total weight would be 896g.

However we also found that the ZIPPY Compact 5800mAh 4s 60c Lipo Pack battery could be used seeing that it also has a 70C burst discharge rating. The benefit of this battery was that while it was a little heavier than its three cell counterpart, two of them it would provide the more desired 29.6 V. However this battery was more expensive and with our limited budget that was also something that we had to take into consideration. It is always a tradeoff between features and cost and it is always difficult making that decision. In the end it must be decided if the added feature outweighs the added cost and this can be very difficult to do. However, we recognized that making the right decision was vital as it could have ramifications on the whole project.

Table 3.3.6.3: Updated Battery Comparison

Features	**ZIPPY Compact 5800mAh 3s 60c Lipo Pack**	**ZIPPY Compact 5800mAh 4s 60c Lipo Pack**	**ZIPPY Compact 5000mAh 6s 60c Lipo Pack w/ XT90**	**Turnigy nano-tech Shorty 4200mah 2S2P 65~130C Hardcase Lipo Pack**
Continuous discharge rating	60C	60C	60C	**65C**
Burst discharge rating	70C	70C	70C	**130C**
Weight (grams)	448	576	830	**190**
Dimensions	158x45x32mm	152x46x41mm	157x45x60mm	**96x46.4x25mm**
Number of Cells	3	4	6	**2**
Minimum Capacity (mAh)	5800	5800	5800	**4200**
Voltage (V)	11.1	14.8	22.2	**7.4**
Price	$49.71	$63.37	$81.72	**$27.90**
Number of Batteries needed	2	2	1	**3 or 4**

Additionally, we also wanted to consider using one big battery to power the drone but unfortunately we could not find one that would meet our requirements. The closest one that we found was the ZIPPY Compact 5000mAh 6s 60c Lipo Pack w/ XT90 shown in [Table 3.3.6.3]. Upon first review we thought it might work since it has a burst discharge rating of 70C, however it only has a capacity of 5000mAh, this means that its discharge rating needed to be 80C. So although we wanted to compare going with a single larger battery over multiple batteries, it seemed like a single large battery was not an option with our specifications.

Lastly, we considered a Turnigy nano-tech Shorty 4200mah 2S2P 65~130C Hardcase Lipo Pack. This battery was actually brought to our attention by one of our aerospace engineering teammates. So we had to review the specifications of the battery in order to determine if it would even be possible to power our drone using this battery. As mentioned earlier, we needed to calculate the maximum amount of current that the battery can withstand. It has a capacity of 4200mAh and a burst discharge rating of 120C so we found that it can withstand a max current draw of 504A, which is above and beyond our max current of 390A. We found this to be interesting since we had previously looked into getting a higher capacity battery with a lower burst discharge rating however we found that this combination of a very high discharge rating and lower capacity also worked. Next we needed to see if this lower capacity battery would be able to give us the amount of flight time that we need. Being cautious in the calculations (using the maximum current draw rather than the average just to be safe) we found that this battery should give us 38 seconds of flight time. This is more than double the under 15 seconds that we actually needed. So we concluded that Turnigy nano-tech Shorty 4200mah 2S2P 65~130C Hardcase Lipo Pack battery was a viable option for our project.

Additionally, this battery was the cheapest, smallest, and most lightweight of the new batteries that we considered, which was definitely desired. However, the reason why the battery is so much smaller and inexpensive is due to the fact that it is only a two cell battery. As discussed earlier, in order to power the motors needed to give us the thrust required we need to provide a minimum voltage of 22.2V or better yet 29.6V. This meant that we would need to have either 3 or 4 of these Turnigy nano-tech Shorty 4200mah 2s2p 65~130C Hardcase Lipo Pack batteries in order to power our drone. So we had to take the cost and space requirements of doing so into consideration. We found that even though we would have to purchase either 3 or 4 of these batteries, it would still be cheaper to do so than the other viable battery options we found. Furthermore, the battery has smaller dimensions and a hard case so we realized that it would be easier to stack them. So upon further conferring with our mechanical and aerospace engineering teammates, we decided upon using this battery.

Along with selecting a battery for our main drone, a battery for our test drone also had to be selected. When selecting a battery for our test drone we had to take into consideration the fact that it was of a different size and capacity than our main drone. In the next section our process in selecting our test drone and the drone we selected will be discussed however for completeness of this battery selection section we will discuss battery selection for the test drone here. We wanted to power our test drone with one or two of the batteries that we selected for our main drone. As we knew that this would reduce costs and allow us to be able to do testing faster since we would be able to swap out the drained batteries for fully charged ones. However, since our main drone needed more power than our test drone, we were concerned that the electronic speed controllers or motors may not be able to handle the increased voltage. As such, we had to investigate the specifications of our electronic speed controllers and motors to see what they could handle. We found that the electronic speed controllers were rated for up to 14.8V as were the motors. This was a big relief to us because it meant that we could use two of the Turnigy nano-tech Shorty 4200mah 2s2p 65~130C Hardcase Lipo Pack batteries that we selected for our main drone, wired in series, for our test drone. While we initially used these two batteries we realized that for lightweightness and ease it would be better to have a dedicated test drone battery. We did a quick search for a battery that would meet our test drone requirements and selected the Turnigy 2200mah 3S 25C lipo pack battery for our test drone.

While the Turnigy nano-tech Shorty 4200mah 2s2p 65~130C Hardcase Lipo Pack battery has many benefits as have been discussed above, one of the downfalls of this battery was that it came with a “build your own connector” connection. This meant that a connector of our choice had to soldered on as opposed to the other batteries which came with the connector already soldered on. However we since we are engineers this was not out of our abilities, it was just a minor inconvenience. Since the battery gave us the ability to choose our own connector, we decided to research which connector would be best. Given the fact that we initially thought that we would be using the same battery for the test drone as we will be for our main drone, we decided to look into the electronic speed controller connections of the test drone. As will be discussed in the next section, we selected a test drone, and this test drone came with electronic speed controllers that already had Deans connectors soldered on. So we determined that it would be advantageous to use Deans connectors for all of our batteries rather than trying to get adapters which would be an unnecessary added cost. However after later switching to a dedicated test drone battery, we decided to change out the battery connectors to EC8 connectors which were rated for the type of current draw that we would be pulling with all six of our motors on the big drone.

3.3.7 Test Drone Selection

Next we decided to begin figuring out what test drone options were available on the market. We needed a drone to test all of our electronic components to make sure that they were functioning correctly. We couldn’t just wait to place our electronics on the drone that our mechanical and aerospace teammates would build because we knew that it would not be complete for a few months. So our plan was to test our components and software as much as we could on a test drone somewhat similar to their drone design to ensure proper functionality. Then when the drone was finished being built by the mechanical and aerospace engineers we would be able to integrate our electronics with their design and make the tweaks and modifications necessary to get the electronics to function correctly on their drone. An approach like the one we took for this project could also prove useful for the real world as someone who wishes to do microgravity testing may already have a drone and our electronics would be able to be modified to suit their drone rather than requiring them to use ours. Although we mainly decided to go this route due to our time constraints and project requirements. By having a backup drone and way to test our drone sooner rather than later, we knew that we would give ourselves a much higher chance of success than we otherwise would have had. With this in mind, we set out to search for test drone that would serve our purposes.

Table 3.3.7.1: Test Drone Comparison

Features	**Lynxmotion Crazy2Fly Drone (Base Combo Kit)**	**Lynxmotion Crazy2Fly Drone (T-Motor Combo Kit)**	**Lynxmotion Hunter VTail 400 Drone (Base Combo Kit)**
ESC's Continuous current rating	12A	12A	12A
ESC's Burst current rating	16A	15A	16A
Weight (grams)	250	250	326
Dimensions	38cm(L) x 38cm(W) x 11cm(H)	38cm(L) x 38cm(W) x 11cm(H)	35cm(L) x 38cm(W) x 14.3cm(H)
Ideal Input Voltage	2-4S LiPo	2-4S LiPo	2-4S LiPo
Motors	4 - 28x30mm - 1000kv Brushless DC motors	4 - T-Motor - 22x08 1100kv Brushless DC Motors	4 - 28x30mm - 1000kv Brushless motors
Not included and required for flight	Flight controller, battery, charger and remote (TX/RX)	Flight controller, battery, charger and remote (TX/RX)	Flight controller, battery, charger and remote (TX/RX)
Price	$274.39	$334.03	$439.38

As seen in [Table 3.3.7.1], upon searching for a test drone we found three potential drones that seemed to fit what we needed, the Lynxmotion Crazy2Fly Drone (Base Combo Kit), Lynxmotion Crazy2Fly Drone (T-Motor Combo Kit), and the Lynxmotion Hunter VTail 400 Drone (Base Combo Kit). These drones seemed like good options because they all came with motors, and electronic speed controllers but lacked a flight controller, battery, and remote. This lack of completeness was seen as beneficial to us because it would let us test our flight controller and telemetry. Additionally, although there were electronic speed controllers included, they all needed to be programmed. So all of these drones seemed to have the base features that we desired with the ability to use our components.

While all of these drones seem similar so far, they have different weights and dimensions that we had to examine. We found that the Lynxmotion Hunter VTail 400 Drone (Base Combo Kit) was smaller than the other two drones but it was 76 grams heavier than them. This was important because seeing that was already starting off heavier than the other two drones, we needed to make sure that when we add our electronics to it, the motor will be able to sustain the added weight.

Beyond their weight and dimensions the drones also differed in their motors. We found that while all of the drones had four brushless motors, they had different specifications. For example the Lynxmotion Crazy2Fly Drone (Base Combo Kit) and Lynxmotion Hunter VTail 400 Drone (Base Combo Kit), had 1000kV motors whereas the Lynxmotion Crazy2Fly Drone (T-Motor Combo Kit) had T-motors with 1100kV. The challenge for us became how to distinguish which motors we needed for our project. Doing some research we found that the kv rating of the motor is directly related to the RPM, in fact kv has the units (RPM/Volts). Beyond that we also found that the higher the motor’s kv the higher its top speed will be but the slower it will accelerate while a motor with a lower kv will accelerate faster [14]. This posed a bit of a problem for us as we needed the drone to go very fast and to accelerate quickly. However these motors were just for the test drone so seeing that it is smaller than our actual drone that our mechanical and aerospace teammates would be building we knew that we could not necessarily expect it to be able to achieve the same speeds as the big drone.

Lastly the price of the drones also needed to be considered, although all of the drones seemed to fit our purpose, only one was actually within the $300 we allocated in our budget for a test drone. So we had to seriously assess if we should go over budget in this area and try to reallocate the money so that we would still be able to purchase all of the parts we needed. We needed to figure out not only if we could afford to go over budget on this item but also if we even needed to. The Lynxmotion Crazy2Fly Drone (Base Combo Kit) was actually under budget and we found that it could potentially provide all of the functionality that we needed, if this is the case we knew that we wouldn’t need to spend more than we previously thought necessary. At the same time, we concluded that if the only available test drones that would actually meet our needs were over budget then we would have no choice but to purchase the drone and then try and shift some money around in our budget in order to have a successful project.

We ended up finalizing our drone selection after research it a bit further and to speaking to our mechanical and aerospace engineering teammates. We decided to go with the Lynxmotion Crazy2Fly Drone (Base Combo Kit) due to the reasons we have previously discussed. Due to the fact that we made this decision with care and not hastiness we found that our test drone served our purposes well.

3.3.8 Computational Components

On the drone data is being collected via the flight controller's integrated sensors as well as from sensors included to measure experimental data. Both of these streams of data are transmitted live to the ground-station, and only the flight controller has a built-in connection. With this considered, the experimental sensor data is taken in from the sensor to a microcontroller on the drone, relayed to a single-board computer and then sent to the ground-station. This separation of computer components was done to avoid overloading one system with several sensor inputs and video transmission. As such the drone's microcontroller functions to read and consolidate any external sensor data, and the single-board computer processes and streams the live video feed and consolidated sensor data before transmitting that data to the ground-station. Additionally, the single-board computer acts as redundant storage of data and video in the case of transmission loss. A research summary which helped us come to this conclusion has been included as subsection 3.3.8.1.

3.3.8.1 Research Summary

As part of the experimentation process, a live video feed being transmitted by the drone showing the experimental payload must be handled by the on-board processor. Initially, we were unsure of what we would use in this regard, and believed a simple microprocessor would suffice. However, knowledge of previous information from various classes led to the conclusion that more research was necessary before continuing.

Upon researching of the topic based on this previous knowledge, information in a post on electronics.stackexchange highlighted the immense complications with regards to the implementation of a microprocessor for video processing. Specifically, the limited processing power with regards to MIPS (or million instructions per second) and throughput (the amount of bits usable within the processor for instruction/data manipulation) would severely bottleneck the system [43]. Limited MIPS capability relates to the speed of the processor for all intents and purposes in this section.

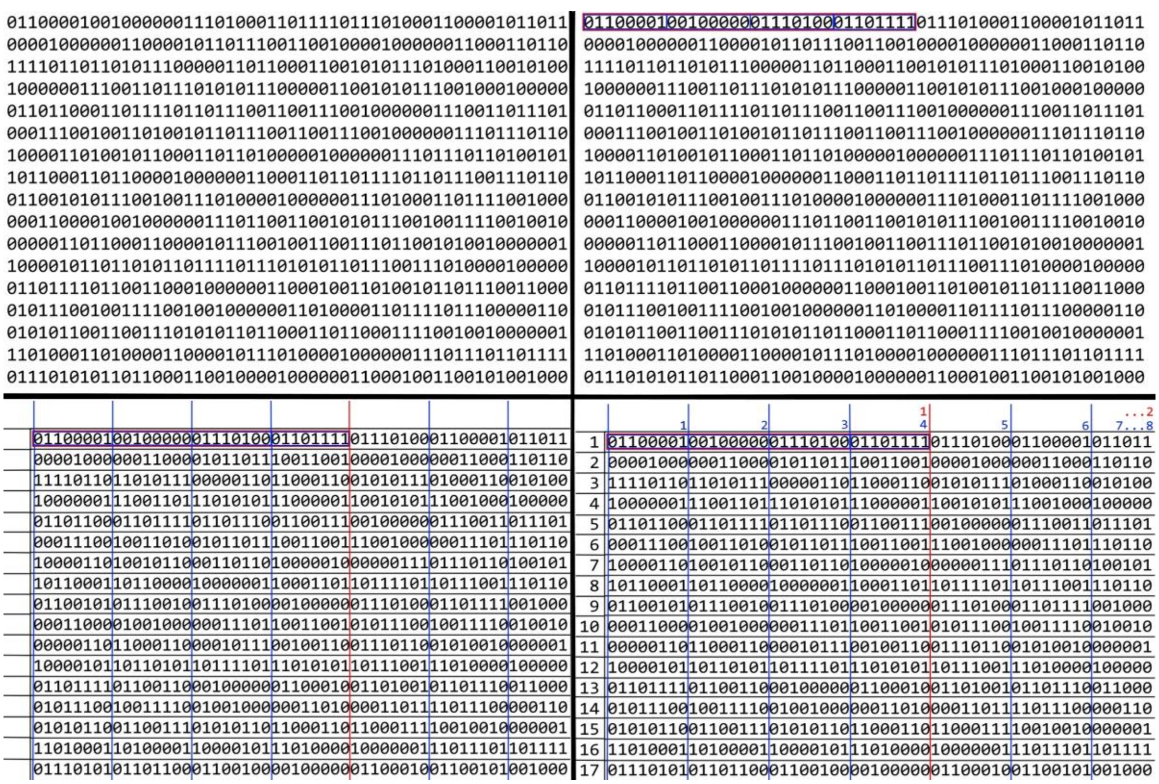
Using [Figure 3.3.8.1.1] as reference, the image shows that a collection of pixels is really made of a large collection of bits (each pixel having its own collection of bits). This is the case for every image which constitutes a frame in a video, such that the camera is constantly passing excessively large collections of bits to the processing unit for transmission and storage.

Figure 3.3.8.1.1: Simplified conceptual example of the composition of an image based on the bits used to form the image's pixels. The "pixels" outlined in black boxes are not representative of the actual size of the pixels of this image.

Permission not necessary to use this figure as it is open source content.

Expanding further on those bits, Figure 3.3.8.1.2(a-d) starts by showing a collection of bits [Figure 3.3.8.1.2(a)] which represent a theoretical pixel. Then in [Figure 3.3.8.1.2(b)] bounding boxes are added such that an individual blue box shows the throughput of the average microprocessor (usually 8 bits) while the red box shows the throughput of a single-board computer (usually 32 bits). There are four blue boxes needed to match the amount of bits encompassed in the red box which illustrates how much more efficient a single-board computer would be when compared to a microprocessor when going through the massive amount of bits required. If it takes a microprocessor four times as long to do what a single-board computer can do assuming an equal processing speed, Figures 3.3.8.1.2(c-d) show just how much more processing is being done by the latter system than that of the former. In just 34 clocks, the single-board computer would be done with this section of bits while the microprocessor would need 136. In reality this divide is only furthered by the processing speed of a microprocessor versus a single-board computer. A microprocessor's speed is generally in the 1-100 megahertz (1,000,000 - 100,000,000 Hz) range while a single-board computer is generally in the 1-3 gigahertz (1,000,000,000 - 3,000,000,000) range. Processors can only process as fast as they can switch electrical charges which is determined in hertz, meaning more hertz is more speed and the single-board computer could have up to a 3000 multiplier speed increase over a microprocessor. Taking into account the amount of images which will need to be processed (i.e. 30 images per second for a 30fps video) during the entirety of the experimental capture, it is clearly apparent that a single-board computer is the system of choice in this regard. There are additional issues with regards to program memory (such as RAM) but already it became apparent a microcontroller or development board would not suffice for even low definition video streaming. Following this information, research began into more powerful processors which involved less setup to video stream in order to reduce the complications involved related to software setup and video streaming. During research a standard for devices which allow simple video streaming known as the USB video device class (or UVC) was discovered and was kept in consideration for any additional future considerations [44]. In order to have on board processing which was compatible with UVC, we looked into single-board computers such as the Raspberry Pi which include integrated USB ports. Raspberry Pi even have camera modules which can be directly integrated onto the motherboard to allow for live video streaming without the need for a UVC compliant camera setup [45]. With the additional functionality provided by the Raspberry Pi, it was clear this would be the choice for handling the live video stream between the drone and the ground-station.

Figure 3.3.8.1.2(a-d): Compilation of figures used to explain processor throughput in regards to an image's bits. Figure 3.3.8.1.2(a) is the upper-left image, Figure 3.3.8.1.2(b) is the upper-right image, Figure 3.3.8.1.2(c) is the lower-left image, and Figure 3.3.8.1.2(d) is the lower-right image.



Permission not necessary to use this figure as it is open source content.

3.3.8.2 Microcontroller Selection

For the microcontroller portion of the drone's onboard electronics there were a variety of options which could have been used to achieve the goal of reading and possibly storing miscellaneous sensor data at the discretion of other experimenters. There were options as straightforward as development boards created by Texas Instruments or Arduino which have microcontroller chips installed on an already designed PCB. Alternatively, the more in-depth and direct route was that of choosing an MCU chip to add to a printed circuit board of our own design. In either case, to achieve maximum functionality it will additionally be necessary that the microcontroller be compatible with analog and digital I/O, have enough ports to accommodate at least one additional experimental sensor, and have multiple communication modules to mitigate the need for constant switching of protocols. Starting with a development board by T.I., the MSP430F5529 LaunchPad Evaluation Kit, was the most complex setup as it accommodates for a significant amount of additional GPIO connections [Table 3.3.8.2.1]. For the microcontroller on this board it wasn't feasible to take the chip and transplant it onto a PCB of our own design due to the massive amount of GPIO pins which would need to be connected. With space and weight both being concerns having a separate board just for a microcontroller was the least flexible option in terms of the overall drone design. Another development board to be considered is the Arduino 101, which is a somewhat simpler chip on a printed board. Arduino is also known for their use of Java which is a much simpler language to work with than the T.I. version of C (a much lower-level language). Similar to the previous T.I. board, the use of an entire Arduino development board was not feasible with consideration of space constraints on the drone. The chip powering the Uno, however, was a feasible option for a custom PCB setup which led to further research into individual chips to mount on our design.

Printing a custom circuit board to house and operate a microcontroller was the most cost and space efficient solution for interacting with additional sensors, considering we already had need of a PCB for power regulation to other drone subsystems. It was not the most flexible design choice, however, as any deficiencies in the chip, or any later design changes (such as switching to a different chip model) would not be addressable in a timely fashion. Even minor PCB alterations would cause massive stalls as it took days of design alterations, waiting for shipment, and then final assembly and testing. Despite this shortcoming, we pursued this option further and were explored chip possibilities such as the Atmel ATMEGA328-PU and an MSP430FR2311. In the case of the ATMEGA328, there was sufficient flash storage for a small amount of data to be saved but not enough for consistent data storage such as when there are multiple flights or when there are several sensors all storing their data. Additionally the CPU architecture was 8-bit which meant the throughput was quite low compared to other available chips. The benefit, however, to choosing the ATMEGA is that it is the chip used on Arduino development boards (such as the Arduino Uno) which means easier programming. Regarding T.I.'s FR2311 chip the throughput was already higher than the ATMEGA with a 16-bit CPU architecture, but the flash memory was only 4 KB which was quite low. Such low flash memory prompted us to consider the other systems going on to the drone, specifically the single-board computer we would be utilizing. With sufficient storage capacity on the single-board computer (SBC) we would be able to mitigate low memory space by only storing SBC rather than on the MCU's limited flash space. This meant that the FR2311's doubled throughput as compared to the ATMEGA328 was the best option for our overall design. With further considerations such as the separate communication modules which allowed for multiple communication modes to be run simultaneously and low-power consumption modes, it was decided the FR2311 would best suit our needs when fitted to our printed circuit board.

Table 3.3.8.2.1: Microcontroller Comparison

Features	**MSP-EXP430F5529 USB LaunchPad**	**MSP430FR2311 MCU**	**Arduino 101**
CPU Operating Frequency	25 MHz	16 MHz	32 MHz
CPU Cores	1	1	2
Non-volatile Memory	128 KB	4 KB	196 KB
RAM	10 KB	1 KB	24 KB
GPIO Pins	63	16	24
Additional Ports	1x USB2.0	None	1x USB2.0
Supply Voltage Rating	5V or 3.3V	3.3V	7-12V
Price	$12.99 (up to $54.99 with additional features)	$1.56	$30.00

3.3.8.3 Single-board Computer Selection

When it comes to real-time video streaming in regards to the experimental payload camera feed, a microcontroller is not a viable solution. Limitations include processing rate (often <1GHz), RAM space (often <1MB), and processor throughput (often <32bits) which on their own present bottlenecks for large, constant data streams but together make microcontrollers ineffective. As such, we integrated a single-board computer into our design to handle the live data transfer from the drone to the ground-station. With regards to choices, there are few options as well known and widely accepted as the Raspberry Pi series, which is all we will be considering for this functionality. Most other options are far too expensive for our budget limitations or are very specialized in their design to handle certain tasks better than others. For general purpose use and low cost the Pi series was the best choice. In regards to model the two we considered initially were the Pi 3 Model B and the previous Pi 2 Model B. Our design was ideally setup to function regardless of exact model as the Pi series of computers receives mostly hardware updates rather than design changes and should result in models being interchangeable as long as hardware performance is adequately met. Our design uses a Raspberry Pi 3 Model B which has 1GB of RAM, a quad core 1.2 GHz Broadcom 64bit CPU, 4 USB 2 ports, and additional slots for hardware additions. Some hardware additions worth mentioning include camera modules which we considered for the experimental live-video feed dependent on quality and price. With this single-board computer in place were minimally capable of streaming video in 360p at 30fps but ideally 720p at 30fps or 480p at 60fps. Our ideal values would be limited by the strength and reliability of our transmissions from the drone to the ground, but without considerations to that we would be capable of reaching one of those two sets of values.

3.3.9 Accelerometer Selection

In this section, we will compare accelerometer options that we considered for use to monitor the gravity levels in our experimental chamber environment. This can be seen in [Table 3.3.9.1]. We placed the three main accelerometers that we were considering in a table to neatly and easily compare their features and differences. This really aided us in making a decision as we were able to quickly see the pros and cons of each device and make a decision based on them. Without this we would have had to go back and forth looking at each of the data sheets of the accelerometers that we were considering and this would have wasted valuable time that could have been used elsewhere to further our project. As such we decided to list out the features in a table.

Although the flight controller we selected has a built in accelerometer due to the nature of the experiments that our project is geared towards we felt that it would be best to have an additional accelerometer inside of the experimental chamber. Researchers would want the most precise measurements possible when it comes to measuring the acceleration so we felt the first step to this would be to go about looking for an accelerometer for just the experimental chamber. Upon doing some research about accelerometers we found that they can either have an analog or digital output, as such we needed to figure out which type of output would be best for our project. We found that the main deciding factor in whether or not to use an analog or digital accelerometer seemed to be the type of hardware that you are working with [40]. With our hardware we can support either type of output so we decided to look at both types of accelerometers, digital and analog. [Table 3.3.9.1] above shows three of the different accelerometers that we considered for the experimental chamber, the SparkFun Triple Axis Accelerometer Breakout - ADXL345, the SparkFun Triple Axis Accelerometer Breakout - ADXL335, and the SparkFun Triple Axis Accelerometer Breakout - LIS331. As can be seen by the table, while they all have similar input voltage ranges, the current draw varies from 10uA to 320uA. The sensing ranges of the accelerometers also go from +/- 3g all the way up to +/-24g. The greater the sensitivity in the accelerometer the more accurate the reading will be so the more sensitivity the better [40]. We found that the price for all of the accelerometers were relatively cheap, all of them are under twenty dollars so it was not a factor in our decision. We ultimately decided to go with the SparkFun Triple Axis Accelerometer Breakout - ADXL345 with the digital output and sensitivity of +/- 16g. We felt that this accelerometer would be the best one for our purposes.

Table 3.3.9.1: Accelerometer Comparison

Features	**SparkFun Triple Axis Accelerometer Breakout - ADXL345**	**SparkFun Triple Axis Accelerometer Breakout - ADXL335**	**SparkFun Triple Axis Accelerometer Breakout - LIS331**
Digital or Analog	Digital	Analog	Digital
Input Voltage	2.0-3.6V	1.8-3.6V	2.16-3.6V
Current Draw	40uA	320uA	10uA
Sensing Range	+/-16g	+/-3g	+/-6g, +/-12g, +/-24g
Price	$15.98	$13.31	$8.86

3.4 Parts Selection Summary

Due to a variety of factors we have not come to many conclusions on specific part selection at this time. Many parts work for our purposes but a true determination of effectiveness versus cost will need to be done before moving forward on any purchases. With the research we currently have, however, it can be said that a decision can be made and acted upon swiftly once a finalized drone design has been chosen and we can begin calculations for total system weight, as well as attempt to predict the full technical challenge to be overcome by the drone’s electronic systems. Should the drone be very large, we will likely need technologies that include the Turnigy 1500mAh 3s 25C Lipo Pack, the most weight intensive battery which provides the most mAh of the batteries we examined. Should the drone be smaller however, every bit of weight which can be reduced may become weight we can carry for the experimental payload, meaning a smaller battery such as the Turnigy 1000mAh 2S 20C LiPoly Pack may be more appropriate. We may even need a balance of combination, but before passing any kind of judgement our group has decided it is more advantageous to allow our colleagues in the mechanical and aerospace engineering departments to give us at least a draft design. Should any complications arrive such as delays in design, we are poised to make selections based on a drone model kit as collected in Table 3.3.7.1 in section 3.3.7. In this scenario a more general design would be applied which we will aim to apply to any drone within reasonable estimates.

Another important note on parts selection addresses the flight controller and how the main difference we are facing is in pricing (with other features included but not necessarily determining factors). With a more solid idea of the budget we are facing we may be able to invest more in having a stable and additional-feature-filled flight controller which would reduce the need for separate modules for GPS or telemetry. Once again, however, we are in the position where a general template can be created without any further input regarding design or budget as we know the ultimate budgetary goal is price versus efficiency (such that price is justified when compared to gains in efficiency).

Finally, to address the most important aspects, the PCB which we will be designing as well as the computational components of the drone, the idea is to have a design which is both open-ended and cost effective. Examining the microcontroller choices presented in Table 3.3.8.2.1 in section 3.3.8.2 Microcontroller Selection, it is likely we will go with a single-chip design by T.I. along the lines of the MSP430F6775A. It is both the most cost effective and generic part (in that it is simply the chip, not a full development board) with sufficient GPIO support for multiple sensors. Ideally any sensors relevant to experimental data will be able to be added to the system as necessary with the microcontroller able to support as many as possible. Along with the microcontroller it was determined there will be a single-board computer to handle live video feed of the experiment. We will most definitely be using a single-board computer in the Raspberry Pi family due to their current wide use and predicted ease of upgrade. Most likely it will be the Pi 3 Model B, the current version of the series in production which is more than capable of supporting cameras with solid resolutions and frame rates.

3.4.1 The Ordering of Parts

Due to the interdisciplinary nature of our project when it came to actually ordering parts we had to go through a bit of bureaucracy. It took us a while to get the information on how we could purchase parts. We asked our fellow mechanical and aerospace engineering teammates and they in turn inquired of their professors. We didn't want to go ahead and order parts without going through the proper channel since if we did so we would run the risk of not being reimbursed. However we did have to purchase some relatively inexpensive components in order to do as much breadboard testing as possible. As can be seen and is explained further in later sections.

As it turns out we found out that there indeed can be no reimbursements from the mechanical engineering department who was in charge of our project. In fact the advisor for our project (more directly the advisor for the mechanical and aerospace engineering students), Dr. Nader, informed us that one person should be in charge of forwarding all ordering forms to him for approval. Our aerospace engineering teammate Nicholas Jones volunteered to take care of all the ordering. As such we had to pass our final parts list to him so that he could fill out the necessary purchasing forms and forward them to Dr. Nader for approval. [Figure 3.4.1.1] below shows that Nicholas forwarded the purchasing forms to Dr. Nader and that they were approved by him. Despite the multi-step ordering process, we eventually received the parts and were able to use them to work on our project.

Figure 3.4.1.1: Proof of the Ordering of Parts

From: Marino Nader [mailto:[redacted]]
Sent: Friday, December 01, 2017 2:57 PM
To: Nicholas Jones <njones86@Knights.ucf.edu>
Subject: Re: purchase orders

Approved.

Dr. M. Nader

Lecturer

Mechanical and Aerospace Engineering Dept.

University of Central Florida

Orlando, FL 32816-2450

From: Nicholas Jones <[redacted]>
Sent: Tuesday, November 28, 2017 2:14:40 PM
To: Marino Nader
Subject: purchase orders

I've attached the first set of purchase forms, mostly for the ECE team. Let me know if there is any additional information you need.

Thanks,

Nicholas Jones

While we were waiting for our parts we laid down as much of the groundwork as we could so that we would be ready to begin further testing as soon as we received the parts. At first we were concerned that with the lack of information and interdepartmental politics we would be unable to get parts ordered as soon as we needed to. However after a slow start, we eventually got the info that we needed on purchasing parts and since we already had a list of all of the final parts that we needed we were able to pass the list on for purchase immediately. So although we at first felt as if we were behind with our parts ordering, we did what we had to do in order to catch up with testing and other various tasks that could not be completed until the parts arrived. Furthermore, in the months that followed we made sure to submit all parts that we needed to be ordered to our mechanical engineering teammate as soon as it became evident that we needed them in order to avoid any unnecessary delays in part acquisition.

4 Related Standards and Design Constraints

In this chapter we discuss the standards that relate to our project and what impact they have had on our project. We also discuss design constraints that we faced such as economic, time, environmental, social, political, ethical, health, safety, manufacturability, and sustainability constraints. We expound upon which design constraints have more of an impact on our particular project and how we have worked within those while still achieving our goals.

4.1 Standards

In this section we discuss various standards and regulations that apply to our project and explain how we adhered to them and encouraged our mechanical and aerospace counterparts to do the same. First we discuss the standards that are related to our project. Our research on drone standards and restrictions are discussed. Next we discuss more specifically the impact that these standards had on our design.

4.1.1 Related Standards

Regarding the many different factors which comprise the project as a whole, there are collections of standards, regulations, and common practices to take into account before proceeding into any individual project portion. As it were, drones are already incredibly complex devices with many regulations which reflect that status. Concerning the construction of a drone in general, we can only assume that our associates in mechanical and aerospace engineering strictly adhered to proper production standards to ensure overall safety for all members of the project team. With regards to the electrical and embedded computing portions of the project, there are standards relating to proper WiFi signal usage, drone operation, and printable circuit board setup. Since we also coordinated with our colleagues in computer science, some programming fell to them, however as engineers we did our best to ensure that they adhered to proper coding standards and followed the prevalent coding convention related to the languages we used. This ensured that as electrical and computer engineers we could go into the code without in-depth knowledge and understand what was essentially occurring in a clear and cohesive way.

To address the most pressing concerns of drone operation, an understanding of the regulations imposed by the Federal Aviation Administration (FAA) is required prior to any flight testing for the project. With regards to drones, or unmanned aircraft systems (UAS), the FAA imposed standard regulations in 2016 concerning the proper operation of drones as well as the requirements for pilots of drones. Specifically, these rules known as Part 107 [24] include operational limitations such as weight limitations, visual line-of-sight (VLOS) usage, and time/airspace restrictions. Including any electronic components we wish to include on the drone and the addition of a payload, there is a strict weight limit of 55 lbs (or 25kg). The payload must also be “securely attached” and cannot “adversely affect the flight characteristics or controllability of the aircraft”. Meaning any experimental payload must be easily accessible but also secure, not to mention how it must be seamlessly integrated into the drone’s design to prevent flight issues. With regard to VLOS, the drone can only be flown by an FAA registered drone pilot within their visual range on a day with “minimal weather visibility of 3 miles”. This includes an airspace restriction on altitude which cannot exceed 400 feet, and can only be operated during daylight or “civil twilight” (just before/after daylight). Concerning a pilot in particular, they must “qualify for a remote pilot certificate” which means the pilot has “demonstrate[d] aeronautical knowledge” via an FAA administered exam or through a pilot’s certificate. If in the future we were to decide to exceed the height limit of 400 feet in order to try and achieve an even better sustained microgravity environment, we would need to filling out a Part 107 Waiver Application on the FAA’s website, a process which can minimally take 90 days [25]. Overall, the FAA regulations concerning drones will be strictly followed starting at discussions of pilot certification for various project members who are interested, and an immediate waiver application filing when applicable.

Another principle concern specifically for us as electrical and computer engineers came in the form of the main circuit board. There are many different organizations with standards concerning the proper assembly of printed circuit boards in a multitude of situations. Mainly, however, we adhered to policies laid out by the Association Connecting Electronics Industries (IPC) in their IPC-A-600: Acceptability of Printed Boards documentation (specifically the IPC-A-600H-2010) [26]. This document is said to “portray specific criteria of the requirements of current IPC specifications” and includes visual representations of these standards [27]. We further determined that we might also refer to the J-STD-001F: Requirements for Soldered Electrical and Electronic Assemblies also by IPC but hosted by the Goddard Space Flight Center of NASA’s electronic library [28]. Given that the J-STD-001F documentation is primarily technical specifications and is essentially an incredibly long collection of written information we realized that it might not be entirely be relevant for a board such as ours. However we felt that it was worth having that information close-by for extra assurance on standard if needed. However for the most part the IPC-A-600H-2010 was what we relied on as it is a comprehensive guide on acceptable quality using photographic examples. We found this to be much easier to understand and work with, as we could immediately see when we had made a mistake that had been classified as “nonconforming”.

One system which has to follow very strict standards and is therefore very straightforward to implementing comes from the proposed Wi-Fi signal system which will transmit instructions to the drone from the ground-station. Wi-Fi standardization has been carefully controlled by the Institute of Electrical and

Electronics Engineers (IEEE) in that every form of Wi-Fi technology must be in some way compliant with the original standard they set in 1997 [29]. Starting at 802.11 Wi-Fi technology is standardized based on “updates” to the previous technology in either minor or major ways which has led to the current prevalent system 802.11ac. This standard has even been adopted by the American National Standards Institute (ANSI) for widespread use and adoption and provides both 5 GHz high-speed connections and 2.4 GHz long-range connections [30]. With a long-range antenna using the 2.4 GHz band we were able to connect to both the drone’s microcontroller/flight controller and the ground-station computer due to this strict enforcement of standards.

Though more loosely related to us, the programming portion of the project is quite important and was worked on and looked over by a multitude of different individuals. As such, we worked with our group mates from the computer science department to enact strict coding standards, or conventions, based around the languages used to code for the project. Every language has their own variation of coding conventions but generally include formats for “file organization, indentation, comments, declarations, statements, whitespace, naming conventions, programming practices, programming principles, ‘rules of thumb’” and other relevant coding concepts [31]. Many large companies, even those related to consumer products adhere to coding conventions or implement their own “in-house” practices to ensure workplace cohesion. In a blog post on their company website, online marketplace Jet, specifically outlines the need for coding conventions to ensure “high quality code” which is “easy to understand, modify, and debug”, all important to streamlining our project specifically [32].

4.1.2 Design Impact

With regards to our specific design and how they will affect our standards, we will most certainly be aiming for a design which successfully follows as many industry standards as possible while not violating any restrictions. In order to do so, the design of the drone by our mechanical and aerospace engineer colleagues will be able to strictly meet the FAA’s forty-pound maximum weight restriction. We will also ensure that the design is capable of stable flight while under load to continue to work within FAA restrictions. As per the printable circuit board, we will ensure to a degree of certainty IPC standards are met in terms of both soldering, circuit design, and overall circuit condition. These standards are not going to take precedent over budget and time constraints which will be significantly more impactful on the project’s “health” overall. Barring any kind of catastrophic circuit board failure, we will likely be able to meet the standards set by IPC and still meet the deadline with a high-quality finished product. Our design will not likely impact Wi-Fi transmission but may need to change depending on long-distance signal quality. IEEE’s strict enforcement of standard on Wi-Fi signals will benefit us should the need for a different antenna arise. As for code design, we will aim to enforce a strict coding convention between our group and the group of our computer science associates. This will ultimately benefit both our groups mutually in creating an effective system for the drone to operate on as well as promote future adoption by any researchers seeking to implement our system for conducting microgravity experiments, this is why these standard are in place and should be adhered to. We seek to do just that with our project and to maintain good standards in our work.

4.2 Realistic Design Constraints

In this section we go through various design constraints that we had to face from economic and time constraints to manufacturability and sustainability. We discuss how these constraints affected our project and how some had more of an impact on our project than others. The reasons why the various constraints affected us and some didn't are also explored.

4.2.1 Economic and Time Constraints

Economic and time constraints presented us with a very real challenge, as we were limited by the agreed upon budget and we had the requirement of providing a working prototype of our system by the end of the second semester. With these constraints in mind, we had to make sure to keep our design, implementation, and test goals realistic so that they could be achieved on time and under budget.

The economic constraint placed direct limitations on the quality level, size, and number of components that we were able to purchase. We started by looking at our objectives and potential design solutions, and decided from there what our priorities were and where we could make compromises. Our budget limitation had a large impact on our power source options. The more expensive drone batteries and battery chargers can easily cost more than $200. We knew that in selecting the batteries for our project we would have to take our budge into consideration. We knew that the use of cheaper batteries could potentially decrease our flight time and as such placed high importance on quality battery selection through the reallocation of budget funds if necessary. Budgetary constraints also limited the quality of camera that we could use to live display and store the live video streaming of the experimental chamber during flight. Cost also limited our options in selecting the model of flight controller for our drone. We must had to select a flight controller that was cheap enough to allow us to purchase our other necessary components, while still giving us enough functionality and tunability to accomplish our microgravity and safety objectives. Ideally the economic constraints that were placed on our project will not be a limiting factor on the versatility of the design to be applied when there are no budget concerns. This would allow researchers with higher budgets to apply concepts from the prototype we are developing and create more refined versions with more batteries to power the drone, finer-tuned sensors, or a different flight controller which would allow for more precise maneuvers and stable data collection.

Time constraints directly limited the level of complexity with which we could implement our drone and experimental environment design. Due to our two semester deadline, we had to make compromises in functionality and quality. We had to be realistic with our goals in order for us to be successful in our creation of a stable prototype. This means that we had to weigh the possibilities of functions like full flight automation with emergency backup manual flight capability vs purely manual flight. We knew that full flight automation would potentially make the quality of microgravity achieved during experiments higher and much more repeatable, but that it could present additional programming challenges. The option of fully manual flight would likely be less difficult to implement, but it would cause the system to be less user-friendly, and greatly reduce the ability of the system to produce consistent results in microgravity quality. As such we decided to attempt full flight automation feeling that although it would be a challenge it was worth attempting. We also felt that circuit board complexity should be kept at a manageable level by avoiding parts that would require intense calibration to function accurately due to the limited test runs we would be able to conduct. Time also made exploring the option of applying for permits to fly the drone at high altitudes a high priority early-on in the project lifespan in order to avoid delays and have all the necessary permissions for regular testing. Additionally, the time it takes to receive parts was also taken into consideration. We knew that we could not spend the so much time designing the printed circuit board, for example, that we would not be able to get it back from fabrication in time to do adequate testing! As such we had to make sure to manage our time wisely so that our project was able to have the best possible outcome. Our table with project milestones and their accompanying dates can be seen in Chapter 8. This table helped us to work within the time constraints that we have and to ensure that we remained on track. This has a real world application because beyond academia we will always have deadlines that we will have to meet and you never want to miss a deadline or tell a client that a product will be done earlier than it can actually be completed. As such the time constraints of this project proved to be quite a good exercise in time management.

4.2.2 Environmental, Social, and Political Constraints

For our project environmental constraints were one of the main types of constraints that were presenting us with challenges. We were trying to create a reduced gravity environment while being on the earth which has a gravitational pull of approximately 9.81 m/s^2. If there was less of a gravitational pull here on earth’s surface then it would have most likely been easier to create a chamber with reduced gravity, however the earth’s gravitational pull is what it is, and as such we had to work with this constraint to achieve our goal.

In terms of social constraints, these posed less of an issue for us. Our goal was to provide a reduced gravity environment to perform testing, and as such there were less social factors involved as there might have been had our project been different. We did not really have to deal with the scrutiny of the public, this may have been because our product is not intended for general use. It is rather meant to serve a niche of the population in the science and research fields. So fortunately we did not have to deal with many social constraints since no one questioned whether it was socially acceptable for us to try and create a reduced gravity environment for scientific testing, this probably did not occur because it does not go outside of any social norms. In our culture today we have become increasingly used to drones being flown for various purposes and they are social accepted for the most part, so creating a new application for drones did not raise any social ire.

Political constraints are another type of constraint that we did not have to worry too much about. This was probably due to the fact that there is no lobby against microgravity drone testing or a real political motive that would cause someone to oppose our goal. If anything the main concern that we felt might be raised was that of safety and informing the public if the testing would affect them or potentially cause them harm. Nonetheless we decided to do some research into just what kind of political constraints we might be facing. We found that earlier this year the Unmanned Aircraft Systems Act was enacted in Florida [66]. This act describes a few regulations that relate to drones. One such regulation is that "a person may not knowingly or willfully: 1. Operate a drone over a critical infrastructure facility; 2. Allow a drone to make contact with a critical infrastructure facility, including any person or object on the premises of or within the facility; or 3. Allow a drone to come within a distance of a critical infrastructure facility that is close enough to interfere with the operations of or cause a disturbance to the facility. [66]" According to the legislation 'critical infrastructure facility' means "completely enclosed by a fence or other physical barrier that is obviously designed to exclude intruders, or if clearly marked with a sign or signs which indicate that entry is forbidden and which are posted on the property in a manner reasonably likely to come to the attention of intruders [66]." So according to this act we could not operate our drone over any such 'critical infrastructure facility.' This was not a problem since we operated both our test drone and actual drone in an open field away from any such facility. So although this legislation certainly applied to our project and it was good to know, this Unmanned Aircraft Systems Act did not really constitute a major political constraint for our project since we were already planning to work within it. It was reassuring to know however, that there was not any further restrictions or requirements in this newly passed act that would affect our project further.

The current assessments on environmental, social, and political constraints above and how they affected our project were just our preliminary thoughts, and we knew that in the future more insight might be gained into the further ramifications of these constraints. However upon starting our project we felt like the environmental constraints seemed to play the biggest role out of the three constraints presented in this section. This is due to the fact that our project revolved around creating a reduced gravity environment within an environment that contains 9.81 m/s^2 of gravity. Social constraints were not much of an issue since the product is meant for a niche scientific community and furthermore since drones are socially accepted. Political constraints had the least impact on our project since there was no real political lobby that fought against the drone we sought to produce.

4.2.3 Ethical, Health, and Safety Constraints

Ethics are very important in any field and especially in engineering. In the workforce we would not want to cut any corners in an effort to save money or quickly fix a problem as this would be unethical and potentially unsafe. This was true for our reduced gravity drone project as well, if we were to say, for whatever reason, choose an electronic speed controller that could not really handle the current we knew we'd be providing, and we passed it off to the mechanical and aerospace engineers telling them that it could handle it that would be unethical. In this example we would be simply lying and in the end it wouldn't benefit us because our project wouldn't work either. Due to the unethical nature of this we refrained from doing anything of the sort during our project. Additionally, in other instances we could've avoided taking necessary precautions in order to avoid facing the challenges at hand. However we knew that this would be unethical behavior since we would not be doing our due diligence to ensure the safety and reliability of the drone so we refrained from this behavior. We appreciate the internal ethical constraints that we faced and did our best to be ethical in every aspect of this project. As far as outside ethical concerns, those were less prevalent, since we wasn't any controversial issue that were trying to tackle or something that we were doing that could be considered unethical. We were simply trying to provide a way for researchers to test in a reduced gravity environment using drone technology, this is not something that would be deemed unethical in any way shape or form. However with that being said our product is a drone with a camera and that could be used unethically. So we explicitly stated in our user operations manual that our drone is intended for scientific experiments and is not for use in any other application. This is intended to protect us if anyone decides to use our product for unethical purposes.

Let us now examine health constraints, the product that we were creating, a drone that is programmed to achieve a few seconds of reduced gravity for testing, did not immediately seem to have any health implications. However upon deeper thought we realized that even though we were not producing a product that would inherently have any chemicals or potentially dangerous materials, we were constructing this drone with the intent for it to be used for scientific experiments. As such we realized that we did need to specify in the owner's manual for our drone that proper safety precautions should be taken especially when it comes to the types of chemicals used. Spillage or a chemical reaction could occur and we wouldn't want someone being injured if this were to happen. This brings us to the next topic of concern, safety constraints.

Moving on to safety constraints, it is clear why safety would be an important factor to consider in any project, we did not want anyone get hurt and as such we did all that we could to prevent such an occurrence. For our project we had to operate a drone, drones have propellers that spin very fast and can cause real bodily harm if not handled properly. So we had to ensure that proper safety protocol was followed and listed in our owner's manual with the correct operation explained. Another issue that we considered was if the drone were to unexpectedly fall from the sky. This could have occurred due to a sudden lack of power or some other motor failure, and we recognized that if the drone were to fall from the sky, anyone in its direct path could be seriously injured. This is why we stipulated that the drone not be flown around large groups of people and that it ideally should be operated in a secluded area where foot traffic is very minimal.

When we were considering the possibility of the drone falling, we contemplated having some kind of sensor that would stop the drone from operating if there was something under it but we realized that this was not really practical with our time constraints. Also since this drone is meant for scientific use and for the use of the general public, we felt it was likely that the drone would be operated in a rural or at least a low traffic area anyway and that stipulating that the drone not be used in a busy area with groups of people around would be enough of a safety measure. Additionally, we did pick components that have the ability to alert us when the battery is low as to prevent the drone from falling out of the sky due to a lack of power. The safety of the user of the drone along with those around the drone have been taken into consideration and precautions have been taken to ensure their safety.

4.2.4 Manufacturability and Sustainability Constraints

The manufacturability constraints we faced were mostly dealt with by our mechanical and aerospace engineering teammates. We were fortunate to be able to have team members who specialize, to an extent, in the actual manufacturing side of things. Our mechanical engineering teammates in particular had to face many manufacturability constraints, from choosing the right materials to the actual practicality of being able to manufacture the drone, they had to overcome many manufacturability constraints.

If our drone design were to be produced on a large scale we would have to face a whole different set of manufacturability constraints, our design may not be able to be made in a factory the same way it was built by hand. In fact it is more probable than not that a new design would have to be made for a large scale production of it. However, our goal was to produce a niche product that could be used by members of the scientific community to conduct experiments specifically pertaining to microgravity experiments. So the likelihood that our drone design would need to be produced on a large scale is slim. Regardless, as previously mentioned, we still faced manufacturability constraints in producing just this one drone. Our ability to use different materials and parts was limited to what was available to be delivered or purchased within our limited time frame and our equally limited budget. As such we had to be wise with our purchases and manufacturing decisions.

It is always a plus to have a sustainable design, and to do so requires taking various factors into consideration. In a world where our landfills are filling up and earth's inhabitants are becoming more and more aware of our ability to reduce this, we are becoming to dive more deeply into sustainable designs. In general we don't want to engage in unsustainable or environmentally unfriendly manufacturing as it is not only a bad business move but it is also socially frowned upon. For our specific project however, we were focusing on creating a niche product to mainly be utilized mainly by those in the scientific community who wish to do microgravity research. And as such, the manufacturing of our drone design may only be done a few times rather than many, many times. Thus sustainability constraints were less of a concern for us. With that being said, we did have the issue of ensuring that the drone could sustain a certain amount of tests, and that it would be reliable. We did not want the drone to only be able to be used for a couple of experiments and then no longer be usable. So when picking components and materials we thought about how long they would be able to be operated.

These are our current views on manufacturability and sustainable design. In the future if this product does become mass produced these topics will have to be looked into in more detail. However as for now the manufacturability aspect of our design was mostly handled by our mechanical and aerospace engineering teammates and the sustainability of our design was only somewhat been considered. This is because we were just focusing on building a working prototype, more of a proof of concept for our sponsor Northrop Grumman. However we still included this section as we felt like it was something that definitely needed to be discussed.

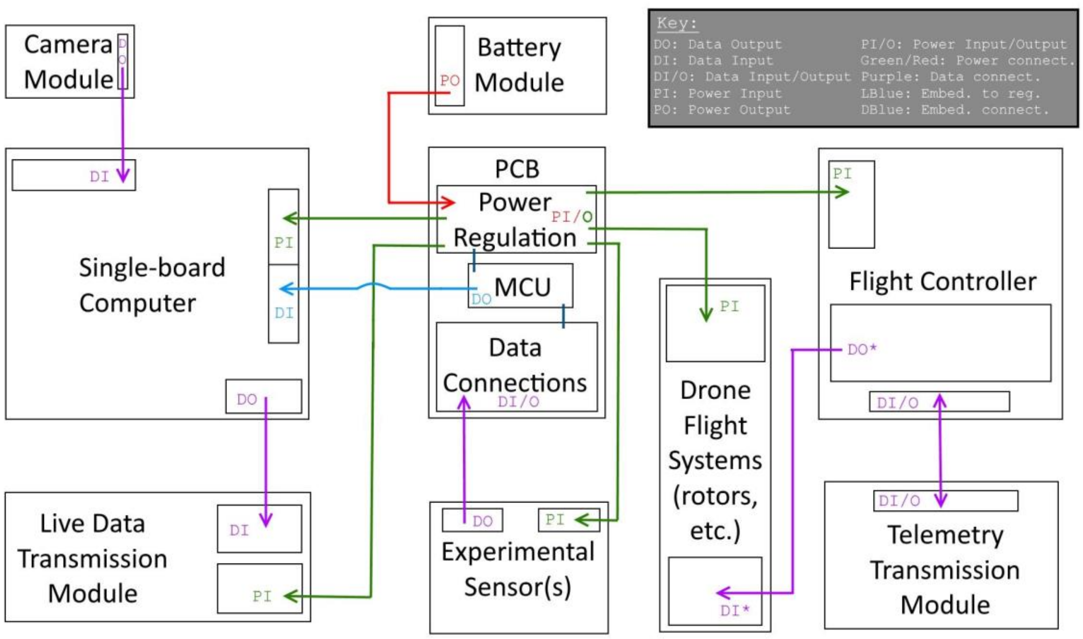
5 Project Hardware and Software Design

In this section we discuss project hardware and software design by providing related diagrams and schematics along with breaking down each subsystem of our project. Additionally, on the software design side an overview is given along with information on some details of it such as Wi-Fi data packets, stored experiment video, and threads.

5.1 Initial Design Architectures and Related Diagrams

In this section, we will outline the different components and technologies that will be utilized in our project. We will also provide a description of how everything will be connected using a figure as reference.

Figure 5.1.1: Abstract architecture diagram for the drone's electronic systems.



Breaking down the image we are using [Figure 5.1.1] as reference for a conceptual connection diagram between components, there's a lot of information which will be helpful to our design process. Before beginning, it is important to note that this is an abstract diagram, so the connections, components, and modules, may not accurately reflect actual wired connections, physical components, or component collections (physical modules). With that in mind, the Battery Module will output power directly to the Power Regulation section of our central PCB where the voltage/current will be split as necessary to the other electronic components. This connection being colored red does not imply anything about the connection, but rather distinguishes the connection is one that may be unique from other similar power connections such as those colored green. Exactly how it is unique will be determined in later iterations of the design but a conceptual reason is that all the module's power is going to regulation but not all of the power is being supplied out by the regulator.

Once the regulator has broken down the voltage being supplied, it will distribute the voltage necessary for other components/modules to function. One such component includes the Flight Controller and its Telemetry Transmission Module. As of now, the Telemetry Transmission Module will likely be power by the Flight Controller via a direct connection depending on compatibility of the two components, but that connection has not been determined to be internal (such as an add-on component) or external (such as a wire) and has not been added to the diagram at this time. In the Flight Controller component, there's a purple data connection labelled with an asterisk (DO*) which represents the analog output of the flight controller to the Drone Flight Systems. Although technically a data connection, it directly controls the power output of the rotors and can be seen as a mixed (data/power) connection.

Another notable feature of the diagram is the dark blue (DBlue in key) connections between the MCU and Power Regulation. This connection represents the entirety of the microcontroller's embedded connections between the MCU and the Power Regulation/Data Connections sections via the PCB. It is distinct in that the components are affixed to the PCB and will not be connected with a physical wire but rather a PCB connection. Due to the embedded nature of the MCU, the light blue (LBlue in key) connection between the MCU and Single-board Computer will need to be a hybrid embedded/wire connection. Hence the key referring to this connection as "Embed. to reg." or embedded connection to regular connection.

As the diagram is abstract and not a finalized connection architecture, the exact nature of every component and the connections it shares are unknown. Hence the inclusion of an external power connection between the Live Data Transmission Module and Power Regulation. During physical design, the Live Data Transmission Module may draw power directly from the Single-board Computer via one USB in a data/power hybrid connection or two USBs (one for data, one for power). The design as presented may even be how the module draws power, however, for the time being the connection is shown to be external and will be updated following further information on the exact features of this module. For certain, however, the Camera Module will not take an external power connection as it will draw power via an add-on port from the Single-board Computer, or from a USB port (in either case with the connection acting as a hybrid data/power connection).

One final portion of the diagram to address is the separate module for Experimental Sensor(s) and the generic nature of this module. For testing purposes we are aiming for a single sensor (an accelerometer) to measure whether microgravity was achieved for the experimental payload. In practice, however, the design may need to incorporate more sensors for a thorough experiment beyond simply an accelerometer. To accommodate for this, the abstract diagram simply has a module for sensor(s) rather than specifically adding in every sensor possibility which can be used for experimentation. Dependent on the amount of sensors, the amount of purple data connections leading to the Data Connections section of the PCB and the green power connections from the Power Regulation section of the PCB would change, but the remaining systems would not.

5.2 Printed Circuit Board Design, Breadboard Test, and Schematics

Starting at the most critical subsystem for our drone's electronic hardware design, the main printed circuit board, it is important to note the functionality the board is intended to provide. Primarily, the printed circuit board acts as a "hub" for all the subsystems to route through in order to get power and in some cases transfer data. With a solid set up, the printed circuit board controls the flow of power coming from the battery into the various components ensuring they all receive the proper operating voltage they require. Additionally, the PCB serves to route signals from sensors to the microcontroller as necessary to keep systems which may need to be swapped out from being attached directly to other components.

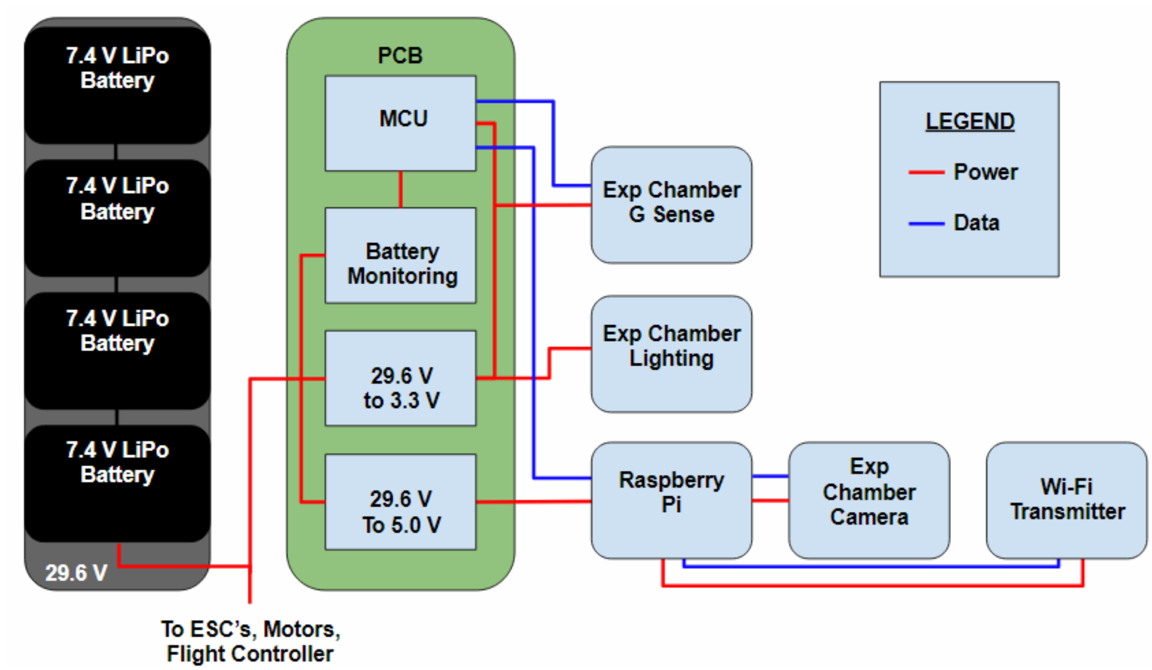
5.2.1 High-Level Input/Output Diagram

In this section, we have included a high-level input/output diagram that outlines the functions of our printed circuit board, along with a short discussion to clarify details.

Furthermore, this clear and distinct figure allows for a quick glance of the printed circuit board design and allows for ease of explanation. Rather than going into all of the details we can easily show this high-level diagram to explain our design to our mechanical and aerospace engineering teammates so that they know what we are doing and where we stand without boring them with the electrical schematic which is not really relevant to them.

As can be seen in the figure below, we have designed the printed circuit board to take power from our LiPo batteries, condition it for our needs, and distribute it to various components. The PCB also routes signals from the accelerometer to the microcontroller, and from the the microcontroller to the Raspberry Pi. The PCB provides conditioned power to the microcontroller and the experimental chamber systems, namely the accelerometer and the LED lighting. The PCB also provides power to the Raspberry Pi computer from a universal serial bus (USB) port. Our design also allows for onboard battery monitoring via resistor voltage division to step down unregulated power, connected to an analog to digital converter (ADC) pin, physical pin 1 on the microcontroller. The Raspberry Pi routes power to the experimental chamber camera board. The electronic speed controllers (ESC's) handle the power distribution to the motors, and the power module drives the flight controller, and transmitter/receiver.

Figure 5.2.1.1: PCB Input/Output Diagram



For the experimental chamber systems, the Raspberry Pi, and the electronic speed controllers to work properly and reliably, they must receive stable DC power within the range dictated by their respective architectures. With various factors potentially affecting the stability of the power coming from our LiPo batteries, as well as the different voltages required by our different systems receiving power, we needed to have regulators on our printed circuit board. As shown in the figure, one regulator converts the 29.6 volts from the four series-connected LiPo batteries down to the 3.3 volts necessary to power the experimental chamber systems and the microcontroller, while another regulator conditions the 5 volts necessary to power the Raspberry Pi. We did not require the PCB to route power directly from the battery to the electronic speed controllers since a splitter was used. No on-board regulation for these systems was necessary since the ESC's were designed to be connected directly to the battery anyway, and they handle their own power regulation and distribution to the motors. In the sections to follow, we discuss the options available to implement the necessary regulation.

5.2.2 Voltage Regulation Methods

As shown in the high-level input-output diagram in the previous section, we needed two different regulators: one to step the 29.6 VDC supply voltage down to 3.3 volts and the other to step the supply voltage down to 5 volts. This section describes the various methods that are commonly used to regulate voltage. We look at the differences between voltage divider circuits, linear voltage regulators and switching voltage regulators. We also discuss the different functionalities of switching voltage regulators.

5.2.2.1 Voltage Divider Circuits

A voltage divider circuit is the simplest technique for regulating or stepping down voltage. It is implemented by two or more resistors in series, with the input voltage being distributed across each of the resistors according to the ratio of each resistor to the total series resistance. For example, if we have an input voltage of 15 volts applied to three series resistors of the same value, the voltage across each resistor will be 15 divided by 3, or 5 volts. This is the simplest and worst technique for voltage regulation because of the power that is dissipated in the resistors. It is for this reason that we did not choose voltage divider circuits for our regulation needs.

5.2.2.2 Linear Voltage Regulators

Linear voltage regulators are one of the most basic methods to regulate voltage. These regulators have three pins and work by adjusting the effective series resistance of the regulator based on a feedback voltage. This means that linear voltage regulators essentially behave like a voltage divider circuit, allowing the regulator to output a constant voltage regardless of what current load is placed on it, as long as the maximum current capacity is not exceeded [41].

One of the requirements of a linear voltage regulator is that there must be a voltage drop across the regulator itself, typically around 2 volts. This requirement, however, did not affect our regulator choice since we expected a much larger voltage drop across the regulators. Our regulator input voltages coming from the 29.6 volt LiPo supply needed to be converted to two different voltages, 3.3 volts and 5 volts, making the voltage drops across the regulators 26.3 volts and 24.6 volts, respectively.

Another problem associated with linear voltage regulators is their inherent lack of power efficiency. Because of the required voltage drop, large amounts of power are dissipated into the regulator. For example, in the minimum case of a 2 volt drop, 2 watts would be dissipated by the regulator with a 1 amp load - since 2 volts multiplied by 1 amp is equal to 2 watts. This means that the effective use of linear voltage regulators is limited to low-power applications [41]. For our purposes, we would see about 26.3 watts and 24.6 watts going to waste - all while heating up our printed circuit board and surrounding components - assuming a 1 amp load for each. This inefficiency and potential amount of power going to waste made linear regulators a bad option for our needs. Considering the amount of power our drone's electronic speed controllers and motors demand, we could not afford to let it be wasted in the form of heat on our printed circuit board.

5.2.2.3 Switching Voltage Regulators

Switching voltage regulators provide a higher efficiency alternative to linear voltage regulators. This means that switching voltage regulators are a viable option for high-power applications. Switching regulators, however, are more complicated as they generally have more pins and require more external components to be connected. Another issue with switching voltage regulators is their tendency to cause problems in nearby circuits due to electromagnetic interference noise [41]. This noise, caused by the magnetic field generated by the inductors connected to the circuit, must be kept in consideration when designing the printed circuit board. This is discussed further in the PCB Considerations and Pitfalls section. Although this method of voltage regulation is slightly more complex than the ones previously discussed, it is the most power-efficient option and thus, it is the method we chose to implement on our printed circuit board. With our decision in mind, it is time to discuss the switching method in further detail, and to outline the different topologies of DC switching voltage regulators.

<u>Step-Down Regulators (Buck Converters)</u>
Step-down regulators, also known as buck converters, are the most common type of switching regulator. Buck converters are used to take a DC input voltage and step it down to a lower DC voltage of the same polarity [42].

<u>Step-Up Regulators (Boost Converters)</u>
Step-up regulators, also commonly referred to as boost converters, are used to take a DC input voltage and step it up, or "boost" it, to a higher DC voltage of the same polarity [42].

<u>Step-Down and Step-Up Capabilities (Buck-Boost Converters)</u>
As the name states, buck-boost converters combine the buck and boost functionalities, allowing the circuit to step down the voltage to the desired level - when a battery is fully charged, for example - and step up the voltage when the battery voltage level drops below the threshold of the buck converter [42]. This type of converter makes it possible to get the most out of a single battery charge,

since it continues to work as the voltage of the battery lowers, boosting as needed.

Switching voltage regulators use a technique known as pulse width modulation to accomplish the task. A simple explanation of how pulse width modulation works is that a feedback loop adjusts or corrects the output voltage by changing the On time of the switching element in the converter [42]. For example, imagine applying a series of square wave pulses to an L-C filter - a circuit with an inductor series-connected to a capacitor, with the output voltage taken at the node between the two. Pulse width modulation relies on the concept of duty cycle, which is defined as the switch On time (or the time that the voltage is high) divided by the total period of the pulse [42]. The series of square wave pulses is filtered by the circuit and provides a DC output voltage that is equal to the product of the peak pulse amplitude and the duty cycle. Thus, changing the amount of time the switch is in the On state directly affects the output voltage of the circuit [42]. This is what allows one to achieve both buck and boost functionalities with switching regulators.

5.2.3 Voltage Regulator Design Process

In this section, we discuss factors that need to be considered in order to successfully implement our voltage regulators. Components needed to be selected that would produce the most accurate and stable results. Mainly, we explain in detail our voltage regulator selection as well as the different types and sizes of capacitors, inductors, and diodes and provide an overview of the process of making selections that were the best for our purposes. This process is described in further detail in source [63] from the references section.

5.2.3.1 Voltage Regulator Integrated Circuit Selection

So we have discussed the different types of voltage regulators and we have narrowed our selection. We know that we need voltage regulators that can handle high power loads, if necessary, since the regulator will need to convert the 29.6 volts from our LiPo batteries down to 3.3 volts and 5 volts, meaning there would be a relatively large voltage drop across the regulator. It is for this reason that we chose to use dc switching voltage regulators, specifically the step-down or buck topology. We decided not to employ the buck-boost topology because it would increase the complexity of our design, and while it could be helpful in some cases, it was not necessary for our purposes.

Once we selected our regulator type and topology, it was time to start the voltage regulator circuit design process. The first step in this design process was to select the specific voltage regulator integrated circuit model and package type that best fits our needs [63]. Searching for options that could convert 29.6 volts down to 3.3 volts and 5 volts, we came to decide that the LM2575 voltage regulator integrated circuit was an acceptable choice - specifically, the LM2575-3.3WU for stepping down to 3.3 volts and the LM2575-5.0WU for stepping down to 5 volts. The LM2575 also has the option to be implemented as an adjustable output voltage regulator, which would eliminate the need for us to purchase two different integrated circuits. We decided not to use this approach, however, since it would increase the number of external components needed, making our printed circuit board design more challenging. Another reason we decided against this is because the design called for the use of external resistors, which would draw more power, dissipate more heat to the board and surrounding components and likely decrease efficiency, something we could not afford to make compromises on.

After building and testing Revision A of our printed circuit board, we decided to make some minor changes to improve the efficiency and reliability of our design. One of these changes was to use the LM2576-5.0WU, instead of the LM2575-5.0WU, to provide the power regulation necessary to reliably power the Raspberry Pi. We made this change because the LM2576 is rated to supply 3 amps, while the LM2575 was only rated to supply 1 amp. This was driven by a suggestion in the Raspberry Pi documentation to use a supply capable of supplying 2.5 amps.

5.2.3.2 Input Capacitor Selection

The next step in the voltage regulator design process is to select an appropriate input capacitor. This capacitor, commonly referred to as a bypass capacitor in this configuration, is needed between the input pin and the ground pin to prevent large voltage transients from appearing at the input [63]. For stable operation of our converter, this capacitor will need to be a low equivalent series resistance (ESR) aluminum or solid tantalum capacitor. Electrolytic capacitors can be used for this application, though with most electrolytic capacitors, the capacitance value decreases and the equivalent series resistance increases with lower temperatures [63]. As electrolytic capacitors are very common and cheap and we will not be subjecting them to low temperatures, we have chosen this type for our input capacitor. As is stated in the LM2575 voltage regulator datasheet, a 47 microFarad, 25 volt aluminum electrolytic capacitor placed close to the input and ground pins provides sufficient bypassing [63].

With our change in regulator selection for 5 volt power came a need to choose new external components capable of handling the 3 amp rating. Our new input capacitor selection was chosen to be a 100 microFarad, 50 volt aluminium electrolytic capacitor. We also chose a new input capacitor for the 3.3 volt circuit with a 50 volt rating to increase safety.

5.2.3.3 Catch Diode Selection

Since the LM2575 is a step-down switching converter, it requires a fast diode to provide a return path for the inductor current when the switch turns off [63]. Now that we have selected an input capacitor, we must find a diode appropriate for this application. Rectifier diodes are a very significant source of losses within switching regulators. This means that choosing the rectifier that best fits into the the converter design is an important process. Schottky diodes provide the best performance due to fast switching speed and low forward voltage drop [63]. They are also the most efficient, especially in low output voltage applications such as ours. Fast recovery and Ultra-Fast Recovery diodes are some other options to consider, although some types of these diodes have an abrupt turnoff characteristic that may cause instability or electromagnetic interference within the circuit or nearby circuits [63]. Another factor we need to take into consideration when selecting our catch diode is the reverse voltage rating of the diode. This rating should be at least 1.25 times the maximum input voltage, according to the LM2575 datasheet [63]. In our case this means that the diode would need to have a reverse voltage rating of roughly at least 40 volts, since our maximum input voltage will be 29.6 volts. The following tables from the LM2575 datasheet provide comparisons of various diode types and packages that will help us in our selection.

The tables that follow (Table 5.2.3.3.1 and Table 5.2.3.3.2) show our Schottky Diode and Ultra-Fast Recovery Diode Comparison. We placed these tables in our document in order to show the process that we went through in selecting which diode to go with. These tables helped us to efficiently see the features of each type of diode and to make a decision on which would be best for our project based on that. Additionally, it made ordering easy as we could just look up the part number and order it without having to think about which data corresponded to which diode. Below more information on our diode selection is given and a further explanation of our process in selecting them.

As is apparent from the diode comparison tables, there seems to be many more options available using Schottky diodes versus Ultra-Fast Recovery diodes. This along with the potential of electromagnetic interference noise has led us to decide on Schottky diodes for our applications. We will be using the through-hole package, and we have chosen the 1N5819 40 volt diode in order to allow for a margin of error. Also, these diodes are common and inexpensive, which is always helpful.

Once again, our change in regulator selection for 5 volt power drove us to choose a new catch diode capable of handling the 3 amp rating. Our new catch diode selection was chosen to be an SB560 60 volt, 5 amp diode. This new diode was more than robust enough to handle our regulation needs.

Table 5.2.3.3.1: Schottky Diode Comparison [63]

	Schottky			
	1.0 A		3.0 A	
	Surface Mount	Through-Hole	Surface Mount	Through-Hole
20 V	SK12	1N5817 SR102	SK32 MBRD320	1N5820 MBR320 SR302
30 V	MBRS130LT3 SK13	1N5818 SR103 11DQ03	SK33 MBRD330	1N5821 MBR330 SR303 31DQ03
40 V	MBRS140T3 SK14 10BQ040	1N5819 SR104 11DQ04	MBRS340T3 MBRD340 30WQ04 SK34	1N5822 MBR340 SR304 31DQ04
50 V	MBRS150 10BQ050	MBR150 SR105 11DQ05	MBRD350 SK35 30WQ05	MBR350 SR305 11DQ05

Table used with permission from onsemi.com.

Table 5.2.3.3.2: Ultra-Fast Recovery Diode Comparison [63]

	Ultra-Fast Recovery			
	1.0 A		3.0 A	
	Surface Mount	Through-Hole	Surface Mount	Through-Hole
30 V	MURS120T3	MUR120 11DF1 HER102	MURS320T3	None
40 V	10BF10	None	MURD320	MUR320 30WF10 MUR420
50 V	None	None	31DF1HER302	None

Table used with permission from onsemi.com.

5.2.3.4 Inductor Selection

Inductor selection is another important topic to discuss, because the magnetic components are the cornerstone of all switching power supplies, and the type of inductor used can have a significant effect on the performance of the circuit. Using the wrong inductor for the job can cause voltage spikes generated by the rate of transitions in current within the switching power supply [63]. These spikes can hinder regulator performance and even damage components in the circuit. Another problem we needed to keep in mind with the use of an inductor is radio frequency interference (RFI) and the electromagnetic interference noise (EMI) caused by the magnetic field generated in the inductor [63].

There are two modes that the LM2575 voltage regulator can operate in - continuous and discontinuous. When the regulator is operating in continuous mode, the current flows through the inductor continuously and never falls to zero. The regulator operates in continuous mode when loads are relatively heavy. This is because when the loads are heavy, the switching is done faster, and the inductor does not have a chance to fully discharge [63]. If the regulator is powering a light load, however, it will be forced into discontinuous mode. While in discontinuous mode, the current in the inductor is allowed to fall to zero for a period of time. This is because when the load is light, the regulator switching is slower. While continuous mode is generally the preferred mode of operation, our 3.3 volt regulator will likely operate in discontinuous mode, since the loads it will drive are light - in the microAmps to hundreds of microAmps range drawn by the accelerometer in the experimental chamber. Our 5 volt regulator, however, will likely operate in the continuous mode for at least part of the time, since the Raspberry Pi is a potentially heavy load - it can draw up to 1 Amp.

To make it a little more simple for us to figure out the inductor that will be appropriate for our needs the LM2575 voltage regulator datasheet has included an Inductor Value Selection Guide. This guide contains a chart for each of the LM2575 regulator options, including 3.3 volt, 5 volt, 12 volt, 15 volt, and adjustable output voltages. These charts show the different ranges of maximum load current versus maximum input voltage and points out which inductor value is best for that range. We only used at the 3.3 volt and 5 volt charts, since those were the only two types of regulators we were using.

The way these charts work is, knowing the maximum current requirement for the load and the maximum input voltage to the voltage regulator, you follow the line for each and see where they intersect. The inductor listed for the range that the intersection falls under is the best selection. Assuming a maximum regulator input voltage of 29.6 volts and a maximum load current rating of 0.6 Amps for our 3.3 volt applications, we were able to determine the inductor that best fits our needs was the L330 inductor, which is 330 microHenries. Assuming a maximum regulator input voltage of 29.6 volts and a maximum load current rating of 0.9

Amps, we found that the L330 inductor was the best choice for our 5 volt applications as well.

Figure 5.2.3.4.1: LM2575 3.3 Volt Inductor Selection Guide

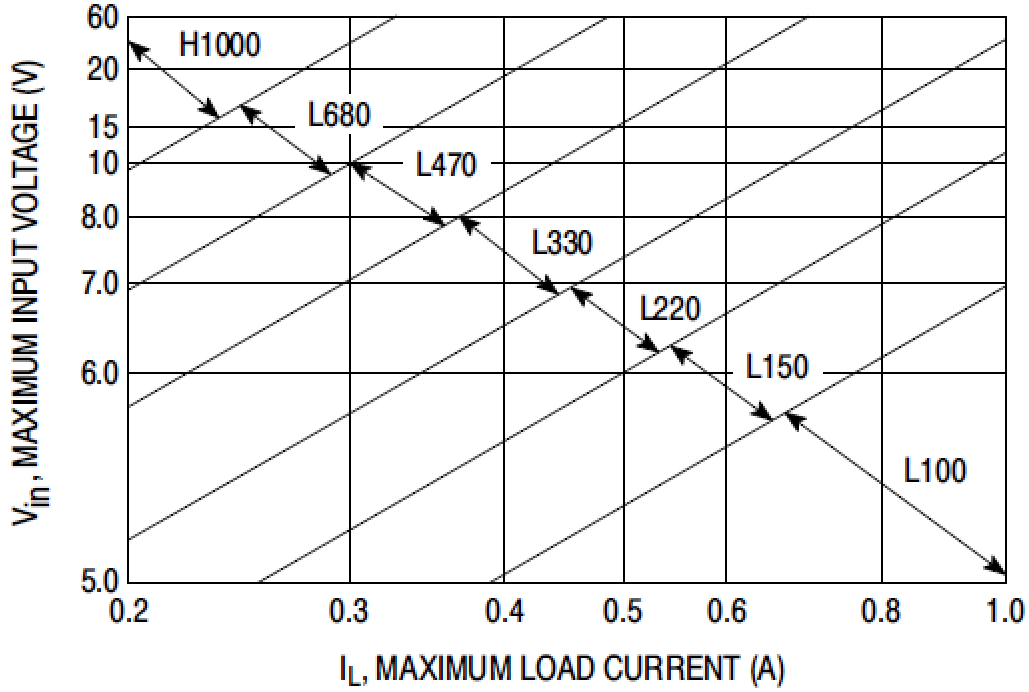


Figure used with permission from onsemi.com.

Figure 5.2.3.4.2: LM2575 5 Volt Inductor Selection Guide

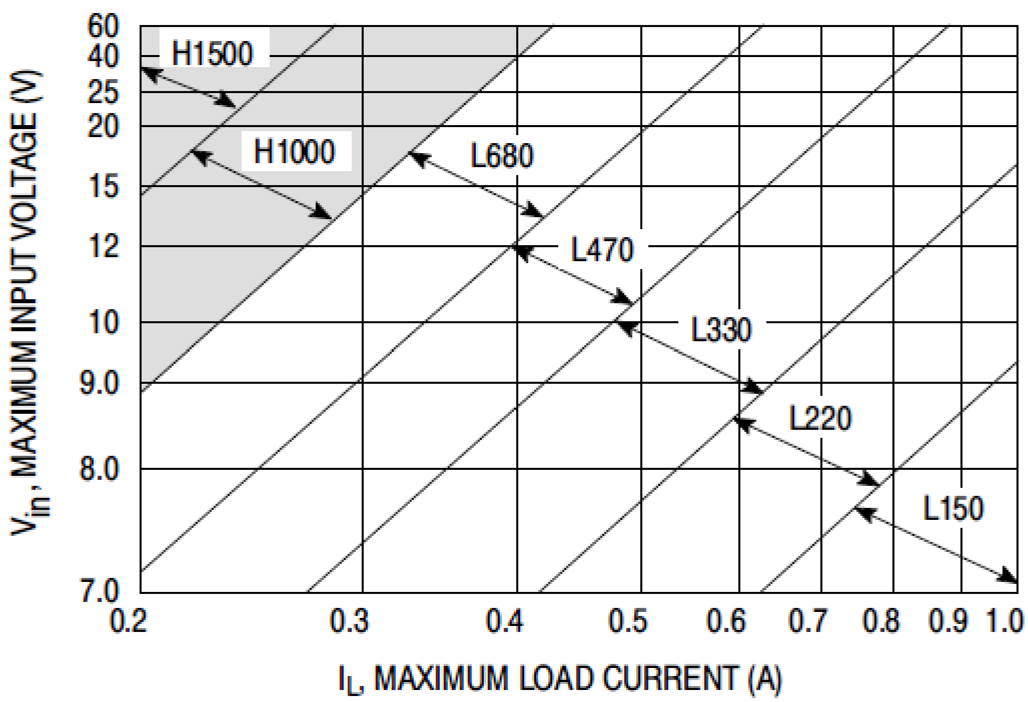


Figure used with permission from onsemi.com.

It is important to note that the compatibility of these inductors with their respective voltage regulator circuits may be affected by the transition from breadboard testing to the printed circuit board. Factors like component placement and spacing will need to be considered, as well as the length and width of the traces on the printed circuit board.

With the changes necessary for Revision B, we were also required to choose a new inductor that would be suitable for the new regulator. The inductor selection guide from the LM2576 documentation is shown below.

Figure 5.2.3.4.3: LM2576 5 Volt Inductor Selection Guide

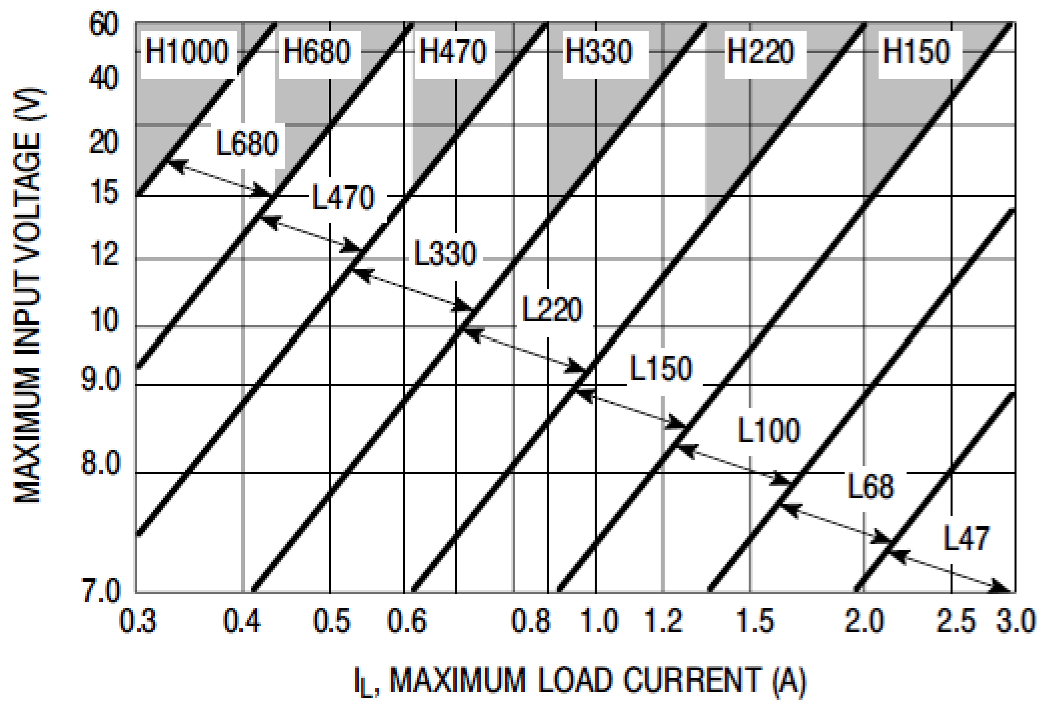


Figure used with permission from onsemi.com.

From the chart above, we were able to find that a 150 microHenry inductor would be best to handle the higher current ranges that we required with our new design [73].

5.2.3.5 Output Capacitor Selection

The two main functions of an output capacitor are to filter the output and provide regulator loop stability [63]. An output capacitor, particularly one with low equivalent series resistance (ESR), is necessary to ensure low output ripple voltage and good stability. The output capacitor's ESR and peak inductor ripple current value are the two main factors that control the output ripple voltage value. This equivalent series resistance value is related to factors such as capacitance value, voltage rating, physical size, and the construction of the capacitor [63].

As mentioned above, it is important to have an output capacitor with a low equivalent series resistance value. However, it is also important that this value is not too low, as this would leave the system more vulnerable to Radio-Frequency Interference and Electromagnetic Interference issues[63]. This means that the selected output capacitor's equivalent series resistance value must be within the proper range. Based on information provided in the LM2575 datasheet, we selected 470 microFarad, 25 volt electrolytic capacitors to use for our circuit's output capacitors.

With the requirements of our new design's 5 volt power regulation circuit, we needed to choose a new capacitor for safe and reliable filtering. Based on the information provide in the LM2576 datasheet, we selected a 1000 microFarad, 50 volt electrolytic capacitor to use at the output of our 5 volt circuit. We also replaced the output capacitor of our 3.3 volt circuit with one of the same value at 470 microFarads, but with a 50 volt rating to increase safety margins.

5.2.4 Voltage Regulator Breadboard Testing

In this section, we describe the process by which we tested our voltage regulator on the breadboard. We have also included pictures of the breadboard test setup and materials needed. These pictures how the exact setup that we used in order to do our breadboard testing. That way, if replication is desired it is easily achievable. Additionally, an explanation is given in order to guide the reader on just exactly what they are looking at and how each test set up was done. The images and pictures are all of good quality so that it can be clearly seen what was done and conclusions drawn from it.

Figure 5.2.4.1: Voltage Regulator Circuit Test Setup [63]

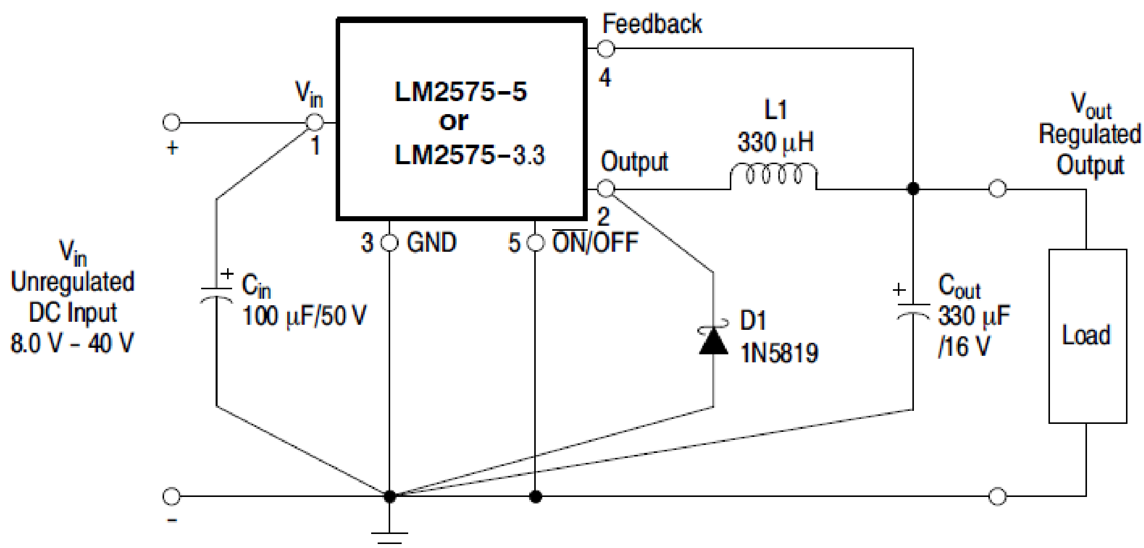


Figure used with permission from onsemi.com.

The circuit diagram above is what we used to perform our voltage regulation. One thing to note is the fact that either the LM2575-5 or the LM2575-3.3 will work with the same peripheral components shown in the diagram above. Once the circuit construction and breadboard testing of the 3.3 volt regulator is complete, the only change that needs to be made is the voltage regulator integrated circuit itself. This will serve to simplify and speed up the breadboard test process for our voltage regulators.

Despite the changes necessary with Revision B of the circuit board, Figure 5.2.4.1 still serves as an acceptable guide, just with a different catch diode and different values for the input and output capacitors.

5.2.4.1 Voltage Regulator Breadboard Test Procedure

This section lists the materials needed and procedure that was taken to perform testing on the breadboard. Following the design procedure outlined in the previous subsections, we have selected, and acquired, the components necessary to move forward with breadboard testing of our voltage regulator circuits. The materials needed for breadboard testing are listed below.

- 3 x Digital Multimeter, alligator clips
- 2 x 12 V (8) AA battery pack, 16 AA batteries
- 2 x Breadboard, jumper wires
- 1 x LM2576-5.0WU DC switching voltage regulator
- 1 x LM2575-3.3WU DC switching voltage regulator
- 1 x 150 uH inductor (bobbin-based or toroidal)
- 1 x 100 uF input capacitor (electrolytic)
- 1 x 1000 uF input capacitor (electrolytic)
- 1 x SB560 60 V Schottky diode
- 1 x 330 uH inductor (bobbin-based or toroidal)
- 1 x 47 uF input capacitor (electrolytic)
- 1 x 470 uF output capacitor (electrolytic)
- 1 x 1N5819 40 V Schottky diode

During this test, we placed the two 12 volt battery packs in series to generate 24 VDC on the power bus of the two breadboards. One breadboard had the 5 volt regulator circuit on it and the other had the 3.3 volt regulator circuit. We were able to take the unregulated 24 VDC from the battery packs and step it down to regulated outputs of 5 volts and 3.3 volts. The steps necessary to complete breadboard testing of our voltage regulator circuits are outlined below.

1. Construct one voltage regulator circuit on the first breadboard following the configuration shown in Figure 5.2.4.2, using the LM2576-5.0WU voltage regulator integrated circuit and the components listed above.
2. The bobbin type inductor may be used in this circuit, however, using these on the printed circuit board may result in radio-frequency and

electromagnetic interference in within the circuit. Using the toroidal inductor on the printed circuit board would reduce this interference.

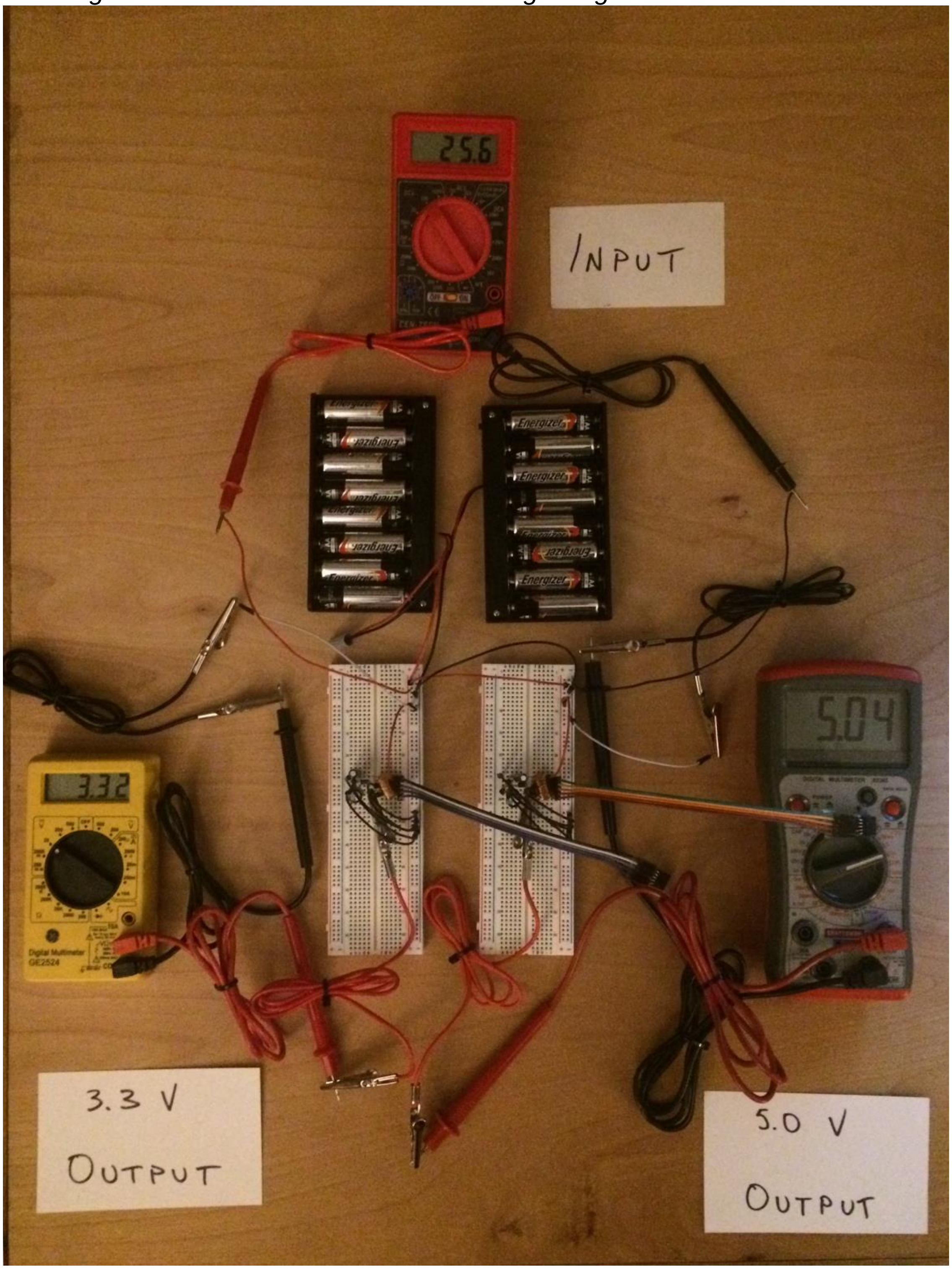
3. Repeat step 1 for the second breadboard, this time using the LM2575-3.3WU voltage regulator integrated circuit.
4. An important note here is to employ a single-point-ground, and to try and keep the leads as short as possible to increase the stability and accuracy of the system.
5. Insert eight AA batteries into each battery pack to reach 12 volts. Connect the two battery packs in series on the power bus of the breadboard to reach 24 volts. This voltage closely reflects the DC voltage we will be taking from the prototype drone's power supply. Make sure the power switch on the battery packs is in the 'off' position and be careful not to touch the battery leads together.
6. Set the first multimeter - which we will refer to as multimeter A - to the correct DC voltage range and connect the leads to the power bus to monitor the input voltage coming from the battery packs.
7. Set the other two multimeters to the correct DC voltage ranges. Connect the leads of one - multimeter B - to the output of the 5 volt regulator circuit and the leads of the other - multimeter C - to the output of the 3.3 volt regulator circuit.
8. Once the circuits and multimeters are connected and properly grounded, turn the power switch on each battery pack to the 'on' position.
9. Multimeter A at the unregulated input should read 24 ± 2 volts, depending on the freshness of the batteries used. Multimeter B at the output of the 5 volt regulator circuit should read 5.0 volts, and Multimeter C at the output of the 3.3 volt regulator circuit should read 3.3 volts.

If the values on our multimeters match the values of their respective multimeters articulated in step 9 of the test procedure, then we have successfully tested verified the performance of our 5.0 volt and 3.3 volt regulator circuits. Verification of the proper function of these circuits means that they can be incorporated into the printed circuit board design. This is very important because we do not want to send for the printed circuit board to be fabricated and when our circuits do not work. This is why we had to test our voltage regulator and the results and images of our voltage regulator breadboard testing can be seen in section 5.2.4.2 which follows. Without such testing we would be skipping a very important step and our whole project could be put at a standstill if we received the printed circuit board and it was not functioning correctly.

5.2.4.2 Voltage Regulator Breadboard Test Results

This section shows the results of the breadboard testing we conducted prior to designing the printed circuit board. As shown in Figure 5.2.4.2.1 below, there were multimeters used - one at the 25.6 volt unregulated input from the battery packs, one at the regulated 3.3 volt output, and one at the regulated 5.0 volt output. Additionally, the specific breadboard circuit make up can be seen along with the batteries used to provide the voltage. We tried to provide good quality images and a detailed explanation so that if someone else wanted to duplicate our testing they could easily do so. Furthermore, our results are discussed and conclusions are drawn.

Figure 5.2.4.2.1: 3.3 V and 5.0 V Voltage Regulator Breadboard Test



Following the procedure outlined in section 5.2.4.1, we were able to complete breadboard testing of the two voltage regulator circuits. As mentioned in this procedure, we connected the two battery packs in series, giving us 25.6 volts which is within our expected margins. Then we applied the 25.6 volts to the power bus on each of the breadboards, making sure to leave the battery pack power switches in the Off position. After constructing the voltage regulator circuits, one on each breadboard, according to the test procedure and figure 5.2.4.2, we attached the leads of the three multimeters to their respective test points and turned each to the correct setting.

Once everything was set up, we turned the power switches on the battery packs to the On position. Watching the two output multimeters to monitor the performance of our regulator circuits, we saw the voltages displayed quickly settled at 3.32 volts and 5.04 volts, as shown in figure 5.2.4.2.1. These voltages were each within one percent of our target values and they do not approach the absolute maximum voltage ratings listed for each of the modules they will be connected to. This was a good indication that our voltage regulator breadboard testing was a success and that we were ready to move on to other subsystem testing in preparation for our printed circuit board design.

5.2.5 Battery Monitoring

In this section, we discuss the need to incorporate battery monitoring functionality into our printed circuit board. We cover what options are available that would allow for battery monitoring and what method best suits our needs for this project.

We have done some research and found some battery alarms for Li-Po batteries that plug into the JST-XH charge plug. The battery alarm has an LED display that shows the total battery voltage along with each cell voltage. It has a modifiable voltage threshold and when the battery level falls below it an alarm will sound. This alarm is loud enough to be heard from the ground even while being surrounded in the casing of the drone. So this battery alarm might prove very useful in our design.

However, although the battery alarm mentioned would inform us when we need to land, we would not be able to read off the voltage from the LED display while it is on the drone.

Our ultimate choice for battery monitoring was to use one of the pins on the PCB-mounted microcontroller. We chose physical pin 1 on the microcontroller because it has analog-to-digital converter functionality. Stepping the unregulated voltage from the batteries down to a voltage safe for the microcontroller pin using a resistor voltage divider circuit allows a simple solution. Once this was done, all that was required was to read the analog value at the pin and convert that to a voltage for battery monitoring.

5.2.6 PCB Considerations and Pitfalls

This section points out some details that we needed to keep in mind as well as potential problems one could face when designing a printed circuit board. Topics that are covered include component placement and spacing, traces, grounding, and heat considerations, and soldering.

5.2.6.1 Component Placement and Spacing

Component placement and spacing plays a very important role in the design of a printed circuit board. According to reference [68], it is generally best to place components only on the top side of the board. Heat sensitive components need to be kept far away from heat generating components to keep systems stable and operating properly [67]. In our case, that means that we had to make sure that the microcontroller was placed as far away as possible from the voltage regulator circuits on the circuit board, as these circuits could potentially generate heat.

It is also important to keep components of analog circuits away from those of digital and high frequency signals [67]. This means it is best to split up the board into two sections - one for digital components and traces, and the other for analog components and their traces. This was another reason we kept the analog voltage regulation circuits separate from the microcontroller, since it takes digital data from the digital accelerometer it is connected to. Another good practice is to place appropriate components very close to each other, while still allowing room for trace routing. This saves space on the printed circuit board and reduces trace length. Trace considerations are outlined in the next section.

5.2.6.2 Traces

The size, layout, and orientation of traces on the printed circuit board are another important factor to consider during the design process. As mentioned in section 5.2.6.1, it is important to keep analog traces away from digital and high frequency traces in order to prevent capacitive coupling, which increases noise in signals [67]. It is also considered good practice to keep traces as short as possible, as longer traces increase resistance and the potential for signal noise [68]. These traces should also be the appropriate width, according to their intended use. Traces intended to carry higher current should be wider than those connected to light loads [68]. Increasing the trace width results in a lower trace resistance, which means less power lost through the trace and less heat is generated. This can be seen in the equation for resistance (R) below, obtained from reference [67].

$$R = \frac{(Resistivity \times Length)}{(T\text{hickness} \times Width\text{h})}$$

It can be a challenge to place components such that traces are as short as possible, as wide as necessary, and so that they can connect those components efficiently with the reduction of signal noise. One possible way to simplify trace paths is to use a multi-layer design for the printed circuit board. This frees up vital space on the surface of the printed circuit board and makes it possible to run traces that would otherwise overlap if they were on the same layer. It is a good practice, however, to exchange wiring directions between layers [69]. For example, if the traces on the top layer of the printed circuit board are generally horizontally-oriented, then the traces on the layer below that would be vertically-oriented. This practice helps to guard against broadside crosstalk problems [70].

Other things to take into consideration when routing traces include the spacing of the traces from mounting holes, from each other, and the shape of the traces themselves. The traces need to be spaced far enough apart so that the manufacturer or printed circuit board vendor is able to reliably complete etching without the risk of a short developing between adjacent traces. A gap of 0.007 to 0.010 inches between all adjacent pads and traces is recommended to mitigate this risk [71]. Leaving adequate space between traces and screw mounting holes is also important. If this is not observed, it could potentially create a shock hazard, and could lead to damaged components [71]. Leave more space here to ensure safe printed circuit board assembly. When routing traces, it is also a good practice to avoid creating any 90 degree angles with the trace. This increases the likelihood of manufacturing errors in the etching process, with the trace being potentially etched narrower than intended, and also increases the possibility of producing a trace that is not fully etched, resulting in a short [71]. A good way to avoid this is to use two 45 degree angles to turn a trace.

5.2.6.3 Grounding Considerations

Another important factor to take into consideration when designing a printed circuit board is grounding. This is perhaps one of the most important things to get right in the process, because not doing so can result in all kinds of problems that could cause circuits on the board to behave unreliably and erratically. In this section, we discuss problems that could arise from employing bad grounding techniques, and how to avoid the potential issues associated with them.

It is bad practice to have many different ground connections all over the printed circuit board. Having too many separate grounds amongst circuits could cause erratic and unpredictable behavior of circuits, potentially resulting in bad control signals, inaccurate data, and dirty power. This is because, if the grounds for nearby circuits of the same type are separate, noise or heat generated by the neighboring circuit could be transferred to the ground of another. This transfer between the two circuits could result in a potential difference between their respective grounds. The possibility of differences in trace characteristics leading to the grounds - trace length and width, for example - could also present problems in the form of voltage drops through the trace [71]. These are some of the reasons that it is almost always recommended to have a ground plane within the printed circuit board as one of its layers. Having a ground plane reduces the power waste within the onboard circuits by eliminating the need for a bunch of long traces to make connections to ground. The ground plane makes it possible for all of the necessary components and circuits to have a common ground, which helps to eliminate the problem of referencing a voltage or signal to two different grounds. This ground plane gives all traces the same reference point for measuring voltage [71]. Another advantage to having a ground plane is that it can serve as a vessel to draw heat away from other crucial areas of the printed circuit board. In order to make connections to the ground plane, vias are used. Vias are essentially conductive cavities in the printed circuit board that connect traces and pads on the surface of the board to deeper conductive layers of the board. Another use of these vias will also be mentioned in section 5.2.6.4.

It is also considered best practice to keep digital and analog grounds separate. This is because voltage and current spikes from digital circuits can generate noise or interference in analog circuits, making them potentially unstable [69]. This suggests that the printed circuit board should be generally split up into two sections: analog and digital. Since it is best to keep their grounds separate, there should be a separate dedicated ground plane for each of them. Setting the board up this way simplifies things and also provides a solution to another issue. Having conductive traces and a ground plane, separated by an insulator can lead to capacitive coupling between the two and the possibility of noise and stray voltage spikes. Restricting the analog and digital traces exclusively to areas above their respective ground planes, reduces the capacitive coupling between the two types of circuits [69].

5.2.6.4 Avoiding Heat Issues

Heat generation is a factor that must be constantly evaluated throughout the printed circuit board design process. This starts with component choice. For example, the voltage regulator integrated circuit that we chose to use on our board is a DC switching voltage regulator. These are more efficient than linear voltage regulators because they do not rely on voltage divider circuits to step down voltage. Voltage divider circuits use resistive components to provide a voltage drop, which wastes power and introduces unwanted heat into our system. This unwanted heat is one of the reasons we decided not to employ linear voltage regulation technology.

Another point in the PCB design process at which heat must be considered is when deciding where to place components. Poorly planned component placement can result in long traces on the circuit board, and as shown in the equation for resistance in section 5.2.6.2, longer traces result in higher trace resistances. Higher trace resistances mean more power is dissipated through the traces in the form of heat. Poorly planned component placement can also result in the transfer of heat from robust circuits to more sensitive circuits, as discussed in section 5.2.6.1. These heat problems can be mitigated by placing the appropriate circuits close to each other - while still allowing room for traces - and making sure components are oriented in a way such that minimum trace length can be achieved. Making sure that trace widths appropriately match the requirements of the circuit is also a good way to reduce the possibility of excess heat, as described in section 5.2.6.2.

As previously mentioned, vias are useful in the design of printed circuit boards. These allow for the connection of surface conductors to inner conductive layers of the board, such as ground planes. This combination of vias and inner conductive layers provide a great way to deal with heat on the circuit board. The vias can be used to move heat from one side of the PCB to the other by giving it a conductive path [67]. This heat handling capability could be even further improved by adding heatsinks on the back side of the circuit board.

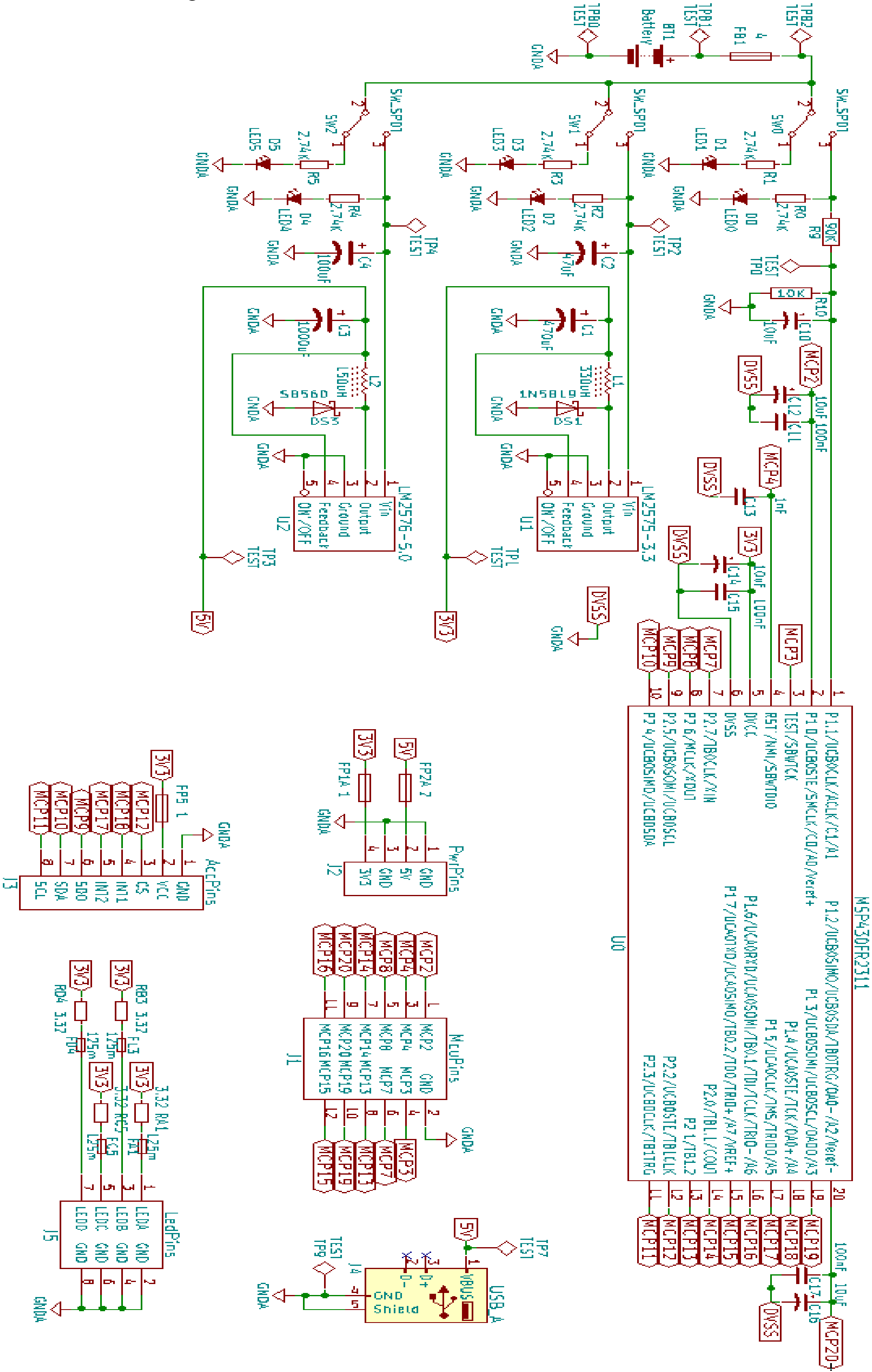
5.2.6.5 Soldering

Another factor to keep in mind throughout the process of designing the printed circuit board is soldering. It is important to make sure that the pads, holes, and traces are laid out in a way that allows for clean and efficient solder connections to be made with the components. In this section we briefly discuss methods that could help one avoid problems and make the soldering process easier. It is recommended to solder in order from small to large components [69]. Soldering larger components first could make it harder to access any nearby small components. Starting with small components like surface mount devices and finishing with larger components like through-hole capacitors, toroidal inductors, and terminal blocks should make the soldering process a little easier [69]. When soldering, it is important not to move the component or the board as the solder dries, as this could cause a “cold” solder joint. This type of solder joint often results in an unreliable electrical connection and can cause erratic and unpredictable behavior in affected circuits. A proper solder joint appears shiny and metallic, not grainy, uneven, or dull.

5.2.7 Printed Circuit Board Schematic

In this section, we provide the schematic for our printed circuit board. This schematic shows the connections between the DC power source, the microcontroller, the voltage regulation circuits, the battery monitoring circuit, and the pin headers for external connections.

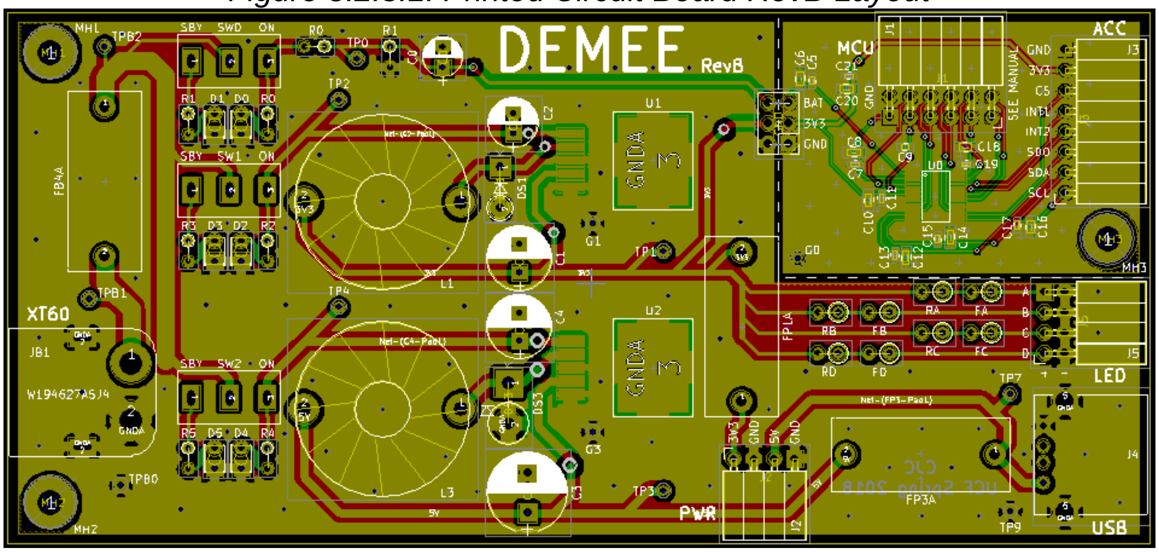
Figure 5.2.7.1: Printed Circuit Board Schematic



5.2.8 Printed Circuit Board Layout

In this section, we have included the final layout for our printed circuit board. This layout is shown in Figure 5.2.8.1. We also briefly discuss the driving factors for final component choice and placement.

Figure 5.2.8.1: Printed Circuit Board RevB Layout



As can be seen in the figure above, battery power is applied at the XT60 connector on the left side of the board and is run through a 4 amp glass fuse for protection before terminating as a 29.6 volt rail for use by the board's circuits. To the immediate right of this rail are three SPDT slide switches, along with red (standby) and green (on) status LEDs and their necessary resistors. The battery monitoring circuit, seen at the top of the layout, uses two resistors for voltage division and a 10 microFarad capacitor for noise filtering before terminating as an accessible pin at the edge of the power section's ground plane. Just below the battery monitoring circuit is the 3.3 volt regulator circuit. This circuit contains the LM2575-3.3WU regulator previously mentioned, along with the necessary external components. Following the output trace of this circuit reveals three branches. One branch terminates as an accessible pin at the edge of the power section's ground plane, another takes power to external pins for use with the experimental chamber LEDs, and the third takes power to an external header as a spare 3.3 volt connection. Below the 3.3 volt regulator circuit is the 5 volt regulator circuit. Similarly, this circuit contains the LM2576-5.0WU regulator and the necessary external components. The output trace from this circuit takes power to the USB port to power the Raspberry Pi, and to the PWR header as a spare 5 volt connection.

At the top-right corner of the layout, one can see the microcontroller section of the board, shown with a dashed line. In this area of the board are the surface-mounted microcontroller, it's decoupling capacitors, and its traces that terminate in female headers for use with the accelerometer and Raspberry Pi. This section has a separate ground plane from the power section of the board. These two ground planes are joined at a single point, with a jumper connection. This was done in an attempt to keep noise from the switching regulators away from the microcontroller traces.

5.3 Second Subsystem - Flight Controller

Another critical subsystem would be the flight controller and telemetry sensors which will make up the entirety of the drone’s flight capability. Including the telemetry within the flight controller itself would be ideal but barring that as a possibility we would conceptually separate the two despite how closely they will work together. For the flight controller we are looking to have a direct connection to the printed circuit board for power supply conditioning and possibly for communication with the independent microcontroller, whether that is one-way communication or two-way. The flight controller will need to be able to at least gather sensor readings from the telemetry sensors and be able to safely engage a landing protocol in the event of a transmitter connection loss. Independent telemetry sensors will need to be connected to the flight controller via the printed circuit board in such a way that they can provide data which the controller needs to be piloted effectively. Keeping in mind that it is our goal the make our system as easy to use as possible, we aim to completely automate drone flight. This will not only keep the system simple to the user, but will also improve the repeatability of experimental chamber conditions.

The flight controller will mainly communicate directly with ground control. This will be accomplished using the telemetry transmitters and MAV Link protocol. The telemetry transmitters use a 915 MHz frequency. The rest of the communication will be directly with the single board computer, raspberry pi. Communication between the raspberry pi and the flight controller will be over gpio pins.

The flight controller will be will be connected to the brushless electronic speed controllers. This will give the drone its flying capabilities. The flight controller will use the speed controllers to determine the direction and amplitude of the rotors. As well as determining telemetry capabilities and calculating flight parameters it will show the current status of the battery. As the battery for the design will not allow a high amount of flying time it will be crucial that the current state of the battery is known at all times.

5.4 Third Subsystem - Experiment Payload

Grouping up the various experimental payload components into one subsystem, the main focus of the project can be confined to that of environmental chamber sensors and a moderate resolution video camera. For the experimental payload sensors, all the connections will go to the microcontroller directly with any power connections being set up on the printed circuit board. The sensors will provide data which the microcontroller can manipulate and store as necessary within its own subsystem. An added benefit of the sensors going to the microcontroller rather than the single-board controller is avoidance of data overload. With just one or two sensors, the single-board computer will likely be able to handle the data as well as maintain the video stream, but as more sensors are added there will be more processor time being spent on simple data collection rather than buffering and streaming video. In regards to the experimental video, a power connection will be routed from the PCB if necessary, but it will otherwise transmit all data to the single-board computer without interference of the microcontroller.

This subsystem is split into two parts. There will be data that is directly sent to the single board computer, Raspberry Pi, and there will be data sent to the microcontroller, computed then sent to the single-board computer.

The video data, due to its high load and USB interface will be sent directly to the Raspberry Pi. The Raspberry Pi will take this data and compress it if necessary because video data is so large when in raw form, combine it with the experimental data from the microcontroller and send it to the ground controller for viewing.

The experimental data routed through the microcontroller will be somewhat pre-processed this will let the Raspberry Pi focus on the video data and transmissions to the ground station. As a safe “minimum” this was decided to be an accelerometer as an additional confirmation of reduced gravity.

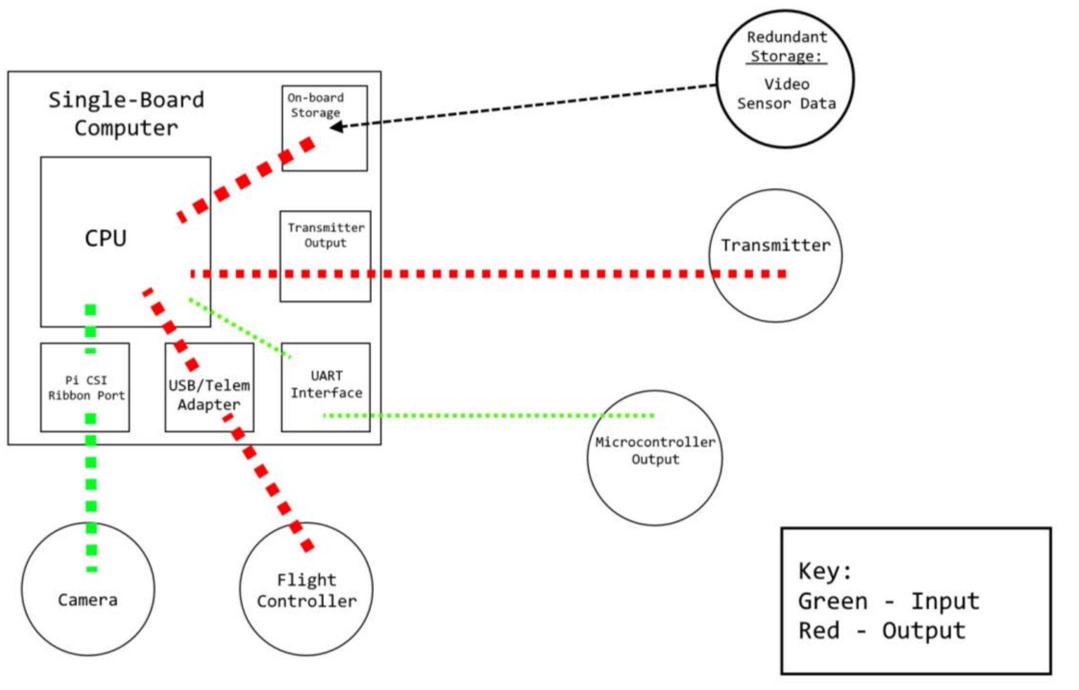
5.5 Fourth Subsystem - On-Board Computing

As mentioned in the previous subsystem, the microcontroller and single-board computer are vital towards experimental data collection and transmission. Both of these components together make up the custom computational subsystem for the drone.

5.5.1 Single-Board Computer

Mainly the single-board computer is used for live-video transmission of the experimental payload to the ground-station and redundantly storing sensor and video data. One additional functionality our Computer Science teammates wanted to achieve was the ability to send commands via a telemetry port integration from a pre-programmed script. As such the Raspberry Pi is loaded with a simple, lightweight Linux OS to run both Python and Java programming language-based scripts. Python is used for autonomous flight scripts and video commands while Java is used for interfacing with the MCU. Further on this topic of data storage, the Pi supports expansive storage in the form of an SD card up to 32 GB so that it can redundantly store video data and sensor data in the case of transmission failures. There is no need to store any telemetry data as the flight controller keeps logs on its hardware in case of transmission failure.

Figure 5.5.1.1: A conceptual diagram of the data being handled by the single-board computer.

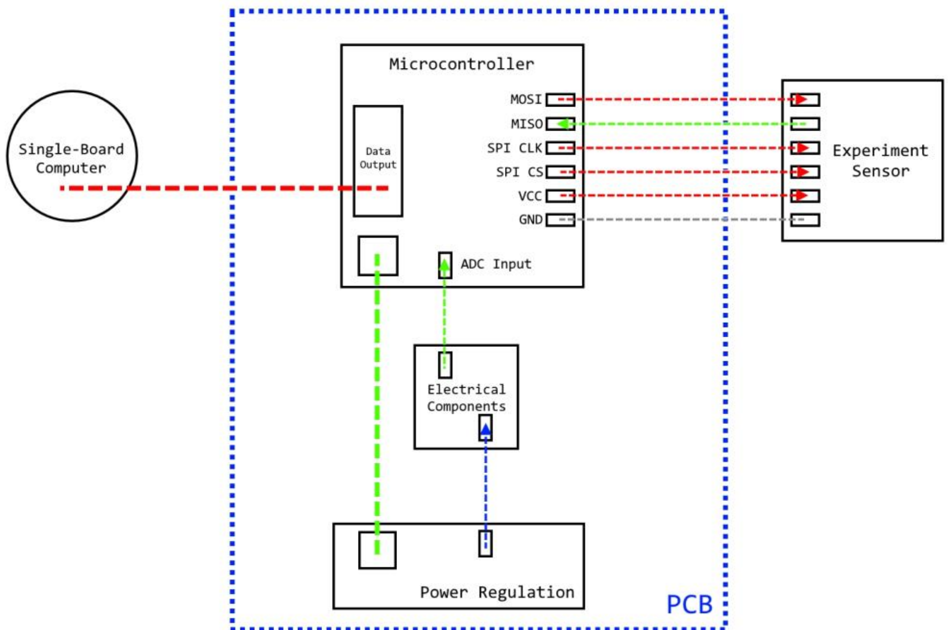


In the image included above, [Figure 5.5.1.1], an abstract representation of the data being handled by the CPU on the single-board computer is shown with along with the data streams of various related components. The data lines (shown in green for input or red for output) conceptually represent how much data is being sent to the components such that the highest input/output connection is the redundant data storage which is taking in the video data and sensor data consistently. In this diagram there's no sensor data being transmitted as it may not update consistently based on the latency between the transmitter connection and the difference in clock rate between the single-board computer and the microcontroller which will result in the Raspberry Pi updating the storage far more regularly than it will be transmitting all the data. To clarify, the transmission data rate is not guaranteed to be consistent and is therefore represented by a thinner line, while the storage data rate will be consistently high (due to the additional camera data) and is therefore represented by a thicker line.

5.5.2 Microcontroller

For the microcontroller, the main task it performs is to simply take in data constantly during flight from the experimental sensor (accelerometer) and transfer that data to the Raspberry Pi for redundant storage. Further functionality as to live transmission of this accelerometer data was set to be a further improvement if there was sufficient time. As the microcontroller is on the PCB, it will be directly related to any power-regulation system which it is capable of analog reading data had we needed to implement redundant battery monitoring. For sensor communication, the microcontroller uses the SPI communication scheme in a 4-wire master device setting. Had we chosen to use a sensor which outputted analog data, this also could have been a use for the ADC (analog-to-digital converter) ports to be processed and then handled as digital data. Between this subsystem and the previous one it can be said they form one larger critical subsystem but are quite distinct in how they contribute to experimentation as a whole and as such they are split into two critical subsystems.

Figure 5.5.2.1: A conceptual diagram of the data being handled by the microcontroller on the PCB.

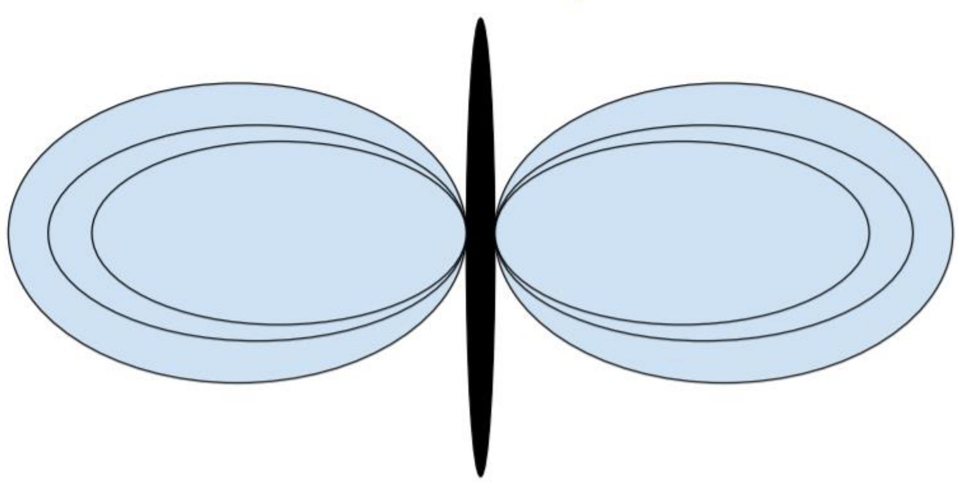


Shown above in [Figure 5.5.2.1], the microcontroller is an embedded component of the PCB and is connected to various components in different ways. Based on current our experimental sensor’s digital outputs the controller utilizes SPI before changing to a UART connection with the Raspberry Pi to transmit the data out. In the figure, the sensor is connected to enable SPI communication with all the compatible pins labelled in their generic names. Also, an electrical connection between the power regulation section of the PCB is routed to some electrical components (such as resistors, capacitors, inductors, etc) to produce an analog signal representing the amount of battery power remaining. Although this was not actually incorporated in the final design, it was included here to serve as an example of expanded functionality with our overall design and approach to the project.

5.6 Fifth Subsystem - Transmitter/Receiver

Then the final subsystem is the drone's independent transmitter/receiver and the flight controller's associated transmitter/receiver. For the flight controller's transmitter/receiver, it will likely be entirely independent of the PCB with power being drawn directly from the flight controller. As for the independent transmitter/receiver there will likely be a power connection directly from the PCB as well as a data connection (likely via USB or possibly GPIO) with the single-board computer. Neither transmitter will be on the same frequency to prevent channel management and possible data corruption leading to likely candidates for the transmission frequencies to be 900 MHz for flight controller data/telemetry and 2.4 GHz (or standard Wi-Fi) for single-board computer communication. With this data being separated onto two frequencies, an ideal video transmission will be a stable 480p at 60fps but more realistically the connection may only be capable of 30fps video. As for flight controller telemetry data, as long as the 900 MHz transmitter/receiver pairs are strong enough, stable connection should be maintained throughout flight without interruption due to the lack of signals operating at this frequency to cause interruptions. Power for the WIFI will be drawn from a usb 2.0 port on the Raspberry Pi 3 b. The Wi-Fi will be a 2.4 GHz transmission with an omni antenna. The antenna will need to be directed outward making it level with the ground providing the best signal strength for the ground station, shown in the figure below [5.6.1] [72].

Figure 5.6.1: Omni-Directional Antenna Waves



5.7 Software Design

In this section first an overview of the software design is given and then a more in-depth discussion of the software design is provided along with a summary of the software design. The software design in this project is just as important as hardware as it facilitates the proper functioning of the hardware. The following is section is quite long due to the fact that software design must be adequately discussed and explained.

5.7.1 Software Design Overview

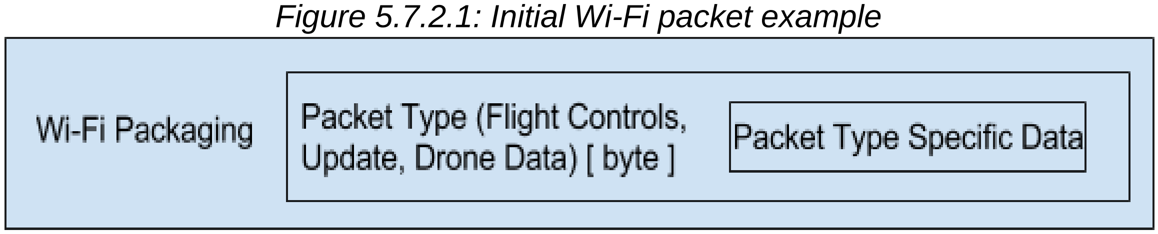
Here a more detailed overview of the software design will be given starting at the microcontroller and Wi-Fi connectivity. The WiFi will be hosted on the single board computer using the WiFi dongle hostapd, dhcpcd, and dnsmasq. This will be the main way of communication between the ground station and the pi.

After this is established the camera will be attached and verified working. Then simple video transmission will be calibrated and tested up to 400 feet between the single board computer and base station as to assure package loss will not completely disrupt the video at potential max range. The PCB will then be connected to the single board computer to establish communication. The single board computer then being fully connected for flight. Last we must initialize the flight controller. The flight controller has a lot open source code, ArduPilot.

5.7.2 Data Packets

The data packets will be the only information to and from the single board computer. These packets will have 4 different types consisting of Data, Commands, Status, and MAVLink.

Figure 5.7.2.1: Initial Wi-Fi packet example



Moving to flight control packets (MAVLink). These are also a one way communication from the control center to the flight controller. The packets will be sent over 915 MHz and will contain telemetry information provided by the open source selected ArduPilot system.

Unlike the previous mentioned data packet type, Drone data is a one way communication from the single board computer to the ground control. These packets will be transmitted using UDP protocol. As mentioned, the data in this packet will be related to live feed video, or acceleration. The live video feed will has its own port separated from the rest of that data as it is not handled by the java coding and is a large amount of data on its own. These packets are not guaranteed to reach their destination and we will have no way of knowing if the actually do. This makes them perfect for redundant live data.
The commands and statuses will be sent over tcp from the single board computer to the ground station. This will take up a third port. The commands will be simple start and stop procedures. The statuses will be success, failed, error.

5.7.3 Live Experiment Video

The experimental video will be taken one of three ways. No matter which way the video has to be stored locally. A proposed solution was to have a phone or Go-Pro attached to the copter and to turn on the video pre-flight. Another solution was to have its own camera which can be done by having the microcontroller send a signal to start the recording or can be another board entirely and start the controller when you setup the experiment. The best method by far was to use a single board computer (raspberry pi 3) with which we could easily transmit the video and store it simultaneously. For a video in high resolution at a target of 1080p (2.1 megapixels) the size would be:

$$2.1 * 2^{20} * (color\ depth) * (number\ of\ seconds) = Size_{raw}.$$

The Raspberry Pi 3 will send the video using a WiFi USB dongle and a USB camera or interfaceable one. This is a lot to handle as well as storing the video. Using a pre-existing library gstreamer.

Gstreamer library supports a range of applications from simple playback, audio/video streaming to complex audio (mixing) and video (non-linear editing) processing [64]. This with the gstreamer plugin gstreamer-plugins-tee which will allow the split of data into multiple pads (branch data flow) [64]. This is echoed on the plugin page:

“Branching the data flow is useful when e.g. capturing a video where the video is shown on the screen and also encoded and written to a file.” [64]

5.7.4 Threads

Threads are independent set of instructions run by an operating system as such (independently) [34]. This is handled by the operating system run separate threads concurrently or asynchronously [34]. Threads run concurrently are not run at the same time, they are run piece by piece so let for example: Thread A and thread B are running concurrently on some operating system. The operating system will split execution time between the threads so it may execute A for 2 seconds (this would be an extremely long period of time) then it executes B for 2 seconds … then A… This can be caused by having more threads than processors which is often the case [34]. If Threads A and B were executing in an asynchronously fashion A would be executing instructions while B is also executing instructions [34]. This can only happen with a minimum of two processors. There are several reasons to use multiple threads:

"

- *Work that can be executed, or data that can be operated on, by multiple tasks simultaneously:*
- *Block for potentially long I/O waits*
- *Use many CPU cycles in some places but not others*
- *Must respond to asynchronous events*
- *Some work is more important than other work (priority interrupts)" [34]*

The copter has several instances for individual threads. This can be very beneficial with multi core processor. When Listening for incoming commands from ground control it can collect data in is its own independent loop that can create software interrupts to signal for data to be written and sent as the buffer fills.

5.7.4.1 Listening / Receiving

The Ground Control will be listening on a port for the single board computer. This will be an independent thread that will loop for the entirety of the program. During the initialization phase of the program a network socket will be opened for listening attached to a port. The program will then listen using TCP (Transmission Control Protocol) and wait for the single board computer to initiate a connection on the active port set and when the processor is ready the thread will continue. At this point the thread will read the incoming message packet entirely or fail if the packet is not valid. As a TCP connection should resend lost packets, packet loss will not be tolerated.

5.7.4.2 Collecting Data

The single board computer will be collecting data from 3 sources; Flight Controller, accelerometer, and live feed camera. This collection will be entirely un-dependent on the camera API. Since the camera must stream data at a higher rate than any other on board system by a significant amount it is handled by a refined raspberry pi library. The camera writes to a buffer as it should as it will require a buffer either way then the data will be written to a pre-allocated buffer and 'tee'd' to a write process and UDP stream process. The accelerometer data is handled by the java. The same java that is connected to the ground station and is send over UDP as well. The accelerometer data will be collected by identifying the padding over the single board computer UART. These serial messages are padded as such; ‘\n’ → ‘X’ || ‘Y’ || ‘Z’ → ‘=’ then there is a 16 bit value for the axis. This is to ensure invalid data is not received.

5.7.4.3 Sending Data / Multicast

In order for the Copter to acquire the host address of the Ground Control it will need to read a multicast sent out in the sending thread. This thread will occasionally send out a small multicast (broadcasted to LAN) packet containing basic information. This information will be high usefully when connecting and check the Copter to make sure it is running an acceptable program version or needs an update.

The main purpose of the sending thread is to send Drone Data to the ground control during flight in which case most of these packets will not need to be sent over TCP as packet loss could be acceptable as these packets should be a never ending stream. It was quite possible that these packets may sent over multicast but was decided that standard UDP would do. This could be quite useful for demonstrations and with a strong enough processor they could be encrypted for security but I do not believe entirely necessary for the current prototype. Most of this data will be raw but the Control Center should have plenty of processing power to make calculation with the data provided.

5.7.5 Error Handling

Error handling is always a finicky process to complete as you want to be as specific as possible with the smallest amount of output as possible. With good error handling a problem that could require hours of looking for a *bug*, can be handled within minutes or even put through a life cycle process of its own: *thrown->caught->subprocess*.

These errors will be most likely to occur in the Copter from transmissions or streaming at the Control Center. Packets sent over the air can easily be lost/disrupted and this can cause non-uniform data to be received. These calls will be handled in a variety of ways. The Control Center will have to just ignore the non-uniform data and maybe inform the user but in most cases an ignore will do just fine. The Copter will have to inform the user because if inaccurate data is received then the wrong MAVLink command could be called for a Flight Controller causing a potential crash.

5.7.6 Mission Planner

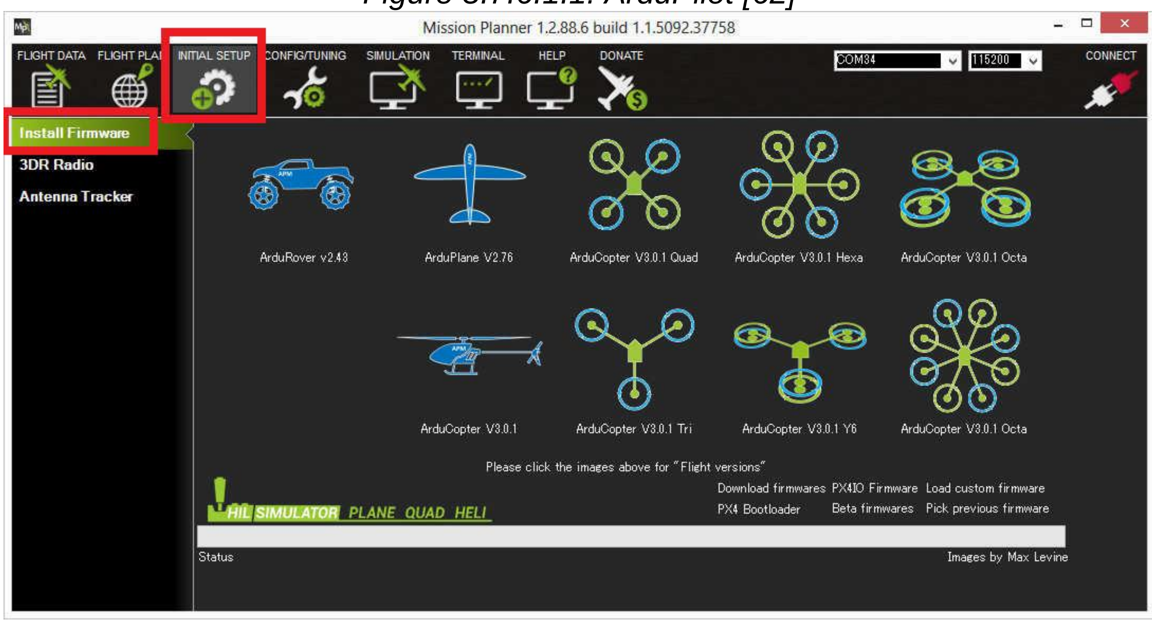
Mission Planner is the ground station open source software use for communication to the flight controller offering status of the sensors, battery, and control of the copter [62]. This ground station software is made to directly integrate with the ArduPilot firmware installed on the flight controller [62]. Mission Planner is even made to install the firmware on to the flight controller and set up parameters for modules connected to the board and flight [62].

The ground station will mainly display waypoints and telemetry information. This will occur over the 915 MHz transceiver plugged directly into the flight controller. The flight controller will receive messages over the transceiver using MAVLink protocol [62].

5.7.6.1 Initial Setup / Firmware Installation

In this section the initial setup and firmware installation will be discussed and explained. This will be done using images as well as text documentation. We will show pictures from the firmware installation and explain how we worked with the software in order to make sure that it was working right for our purposes.

Figure 5.7.6.1.1: ArduPilot [62]



Permission not necessary to use this figure as it is open source content.

Mission controller will be the application use to install the firmware and set the parameters for flight. First and foremost is selection of vehicle type. The Copter we will be using is a hexa-copter (6 rotors) specific ArduPilot firmware. With the correct vehicle selected it will open up more options for mandatory hardware and optional hardware.

Mandatory hardware includes Frame Type, Accel Calibration, Compass, Radio Calibration, Flight Modes, and FailSafe [62]. These all have extensive parameters per feature / module. This can be shown by example. These are the parameters for the battery. These serve towards sensing the current level of the battery.

Below a table can be seen that provide information about our PX4 flight controller and its specifications. This was provided in order to give a better view of the technology that we are working with. Further explanation will be given and the table will be referenced. Having this table proved very useful as we were able to clearly see the parameter name and their description and range. It allowed us to program smartly and more easily.

Table 5.7.6.1.2: ArduPilot [62]

Parameter Name	**Description**	**Range (Units)**
BATT_MONITOR	Controls enabling monitoring of the battery's voltage and current	0 - 7
BATT_VOLT_PIN	Battery Voltage sensing pin	-1 – 2, 13, 100 (disabled / board type)
BATT_CURR_PIN	Battery Current sensing pin	-1, 1 – 3, 14, 101 (disabled / board type)
BATT_VOLT_MULT	Voltage Multiplier, obtain actual battery voltage (BATT_VOLT_PIN * VOLT_MULT)	(Float)
BATT_AMP_PERVOLT	Amps per volt	ampere per volt
BATT_AMP_OFFSET	AMP offset, Voltage offset at zero current on current sensor	(volt)
BATT_CAPACITY	Battery capacity	(milliampere hour in increments of 50)
BATT_WATT_MAX	Maximum allowed power (Watts)	(watt)
BATT_SERIAL_NUM	Battery serial number	(auto for SMBus or -1)
BATT_LOW_TIMER	Low voltage timeout	0 - 120 1 (seconds)
BATT_LOW_TYPE	Low voltage type (Voltage type used for detection of low voltage event)	0 or 1 (Raw Voltage, Sag Compensated Voltage)

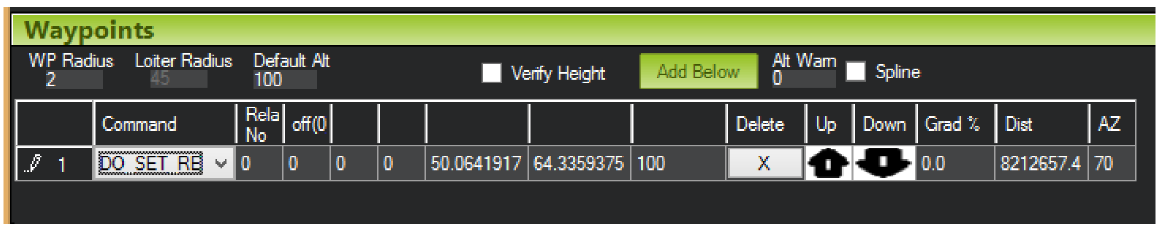
The battery parameter values BATT_MONITOR, BATT_VOLT_PIN, BATT_CURR_PIN, BATT_VOLT_MULT, BATT_AMP_PERVOLT,

BATT_AMP_OFFSET, BATT_CAPACITY, BATT_WATT_MAX, and BATT_SERIAL_NUM are all replicated for a second backup battery.

5.7.6.2 MAVLink

Next we will discuss MAVLink and how that applies to our project. A figure will be shown that depicts the waypoints and explains the steps that will need to be taken in order to properly program our drone and get it to perform the way we want it to. This is very important to the success of our project since if we don't program the flight coordinates correctly the drone will not go at the speed and time that we want it to and will as a result, not produce a microgravity environment which is the whole point of the project. So as such we paid very close attention to this.

Figure 5.7.6.2.1: ArduPilot [62]



Permission not necessary to use this figure as it is open source content.

MAVLink is a communication protocol used to interface between the ArduPilot firmware and a ground station. We will be using a 915 MHz transmitter to exchange the MAVLink protocol signals. This protocol is used for all 3 ArduPilot vehicle firmware types; rover, copter, plane. If an unsupported command is sent over MAVLink protocol it will be dropped [62]. All waypoint commands are sent in a MAVLink_mission_item_message. A MAVLink_mission_item_message can contain three types of commands and usually has 7 parameters of which some can be null; Navigation, DO, Condition [62].

Navigation commands are used to control the vehicle's direction, takeoff and landing. To achieve the flight plan necessary to simulate zero gravity the commands will only consist of MAV_CMD_NAV_TAKEOFF and MAV_CMD_NAV_LAND [62]. DO commands are for executing auxiliary functions. They are not for changing flight position.

Condition commands are used to delay DO commands till a constraint is met. This is mainly our altitude changes. Precisely the main function that will help us achieve micro-gravity is MAV_CMD_CONDITION_CHANGE_ALT [62]. This will have the copter ascend or descend at specific rate in units of meters per second (parameter 1) until the craft reaches a specified altitude (parameter 7). As displayed in the table below.

Table 5.7.6.2.2: ArduPilot [62]

CMD	P1	P2	P3	P4	P5	P6	P7
MAV_CMD_NAV_TAKEOFF	Min. pitch	Null	Null	Yaw Angle	Latitude	Longitude	Altitude
MAV_CMD_NAV_LAND	Abort Altitude	Null	Null	Yaw Angle	Latitude	Longitude	Altitude
MAV_CMD_CONDITION_CHANGE_ALT	m/s	null	null	null	null	null	End Altitude

These commands are a relative few compared to the entire list. The bulk of commands are all referable at: https://github.com/mavlink/mavlink/blob/master/message_definitions/v1.0/common.xml.

5.8 Summary of Design

In time, the drone's physical design will be handled by our mechanical and aerospace engineering counterparts and eventually passed on to us to include in this section. Until then, however, our electronic component hardware and the key ideas of the software can be summarized in one single overarching sentiment. We are aiming to have a lightweight, highly portable, power efficient, and low-cost design to promote small-scale microgravity testing amongst researchers who do not have access to other large scale alternatives. With regards to the electronic hardware, we will have a central circuit board routing power throughout the system to the various components and possibly linking up the data connections of other components to allow for swap-ability of individual components. From this central power and data hub, other components will branch off forming subsystems of the overall design which include the long-range communications antenna, the dedicated microcontroller for experimental data sensors, the experimental data sensors themselves, and the flight controller. Communication between the experimental data sensors will be limited to the microcontroller which may communicate with the flight controller but not the antenna. The flight controller can access the antenna to communicate with the ground-station computer where the pilot will be relaying flight instructions and monitoring telemetry data.

As for the software design, several core ideas including simplicity of code, safety of design, and depth of solution summarize the goals in a general sense. Specifically we are aiming to have a system which uses both the microcontroller and flight controller to handle various processes which can be broken down into code threads. These threads will specify actions for the drone to take while in flight without interrupting the drone's ability to maintain flight. Also the software for the transmitter/receiver of the drone will be based around Wi-Fi packets being sent to and from the ground-station to use the most standard and efficient solution available. To enable efficient communication the software system will be designed around the TCP multicast principles and send packets at a frequency which loss is not necessarily going to lead to a critical failure. Considering the constant stream of data being sent out it is likely that packet loss will not cause problems as long as it isn't a constant loss for a sustained period of time. Also for consideration is the idea of packet encryption which may not be entirely necessary as long as the drone's flight commands cannot be manually overridden via the packets. Should the packets contain flight data being sent to the drone encryption will likely be a high priority to prevent drone theft or unauthorized use of the drone during testing. With regards to the data display, there will be a focus on the need to clearly and cleanly show the relevant numbers and give access to functionality which allows for more information which can aid in the experimental process.

6 Prototype Construction and Coding

In this chapter we discuss our integrated schematics, PCB vendor and assembly, and final coding plan. We discuss our process in creating them and display them in order to expound on them.

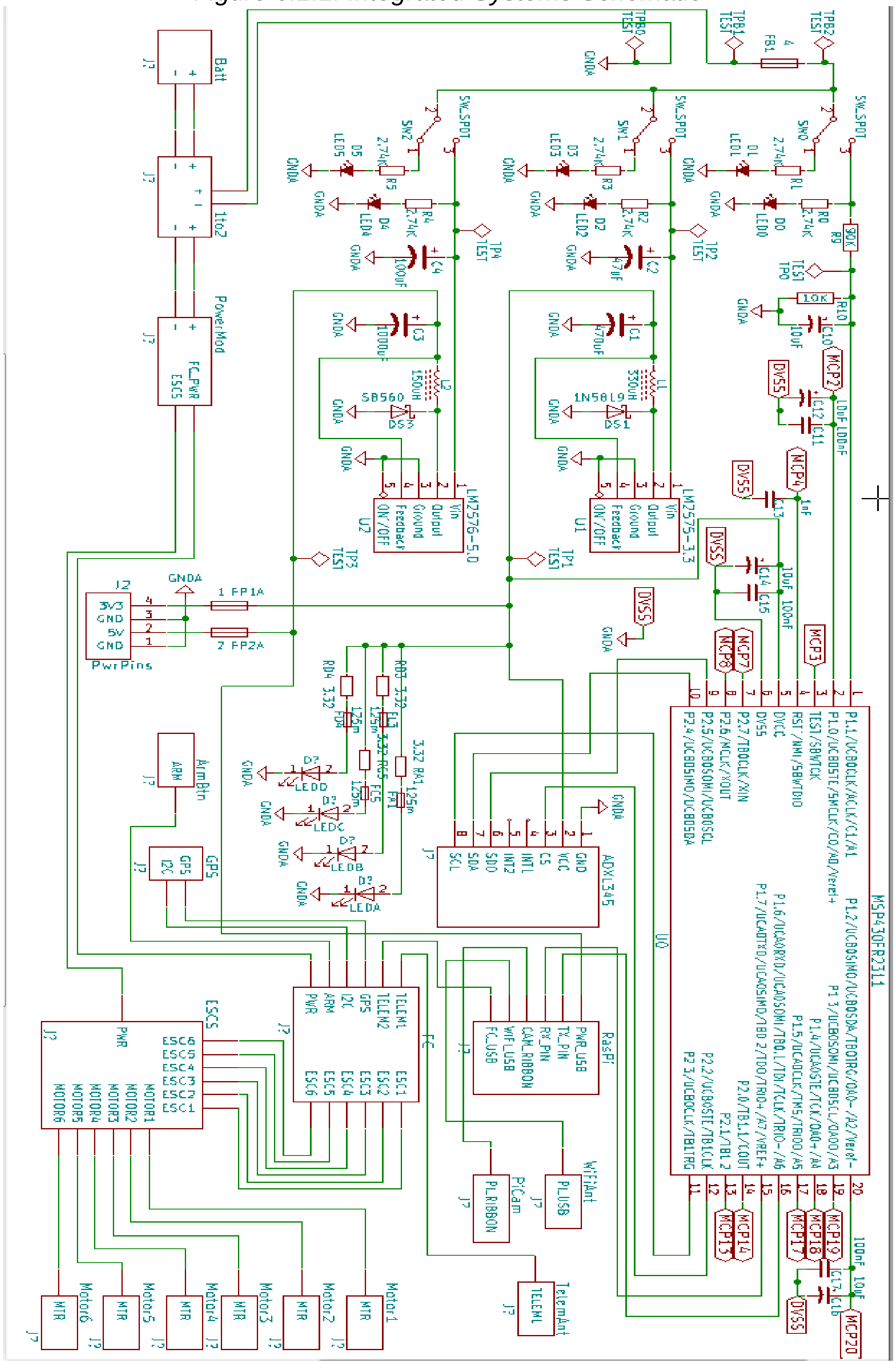
6.1 Integrated Schematic

In this section, we will provide an integrated systems schematic that will include all components and systems that will be connected to the printed circuit board. We will also briefly review the roles of these systems, and explain the types of connections that need to be made with them.

[Figure 6.1.1], displayed on the following page, shows all of the connections that will be made between all subsystems. As can be seen at the top of the schematic, the LiPo battery power source will connect to the circuit board. The application of power from there will then be controlled by a board-mounted switch. Throwing the switch will apply power to the MSP430FR2311 microcontroller through the battery monitoring circuit. This simple circuit uses the concept of voltage division and two resistors to step down the voltage to a safe range for the analog to digital converter pin of the microcontroller. When programmed, this system serves to monitor the battery voltage by sampling and converting the analog signal into digital levels, mapping those digital levels to voltages and comparing them to nominal values.

Throwing the switch also applies power to the 3.3 volt and 5.0 volt switching regulator circuits. As demonstrated in section 5.2, these circuits efficiently step down the battery voltage to levels that can be safely used by our other subsystems. The output of the 3.3 volt regulator is connected to the DVCC pin on the MSP430FR2311, which powers the microcontroller. This 3.3 volts also serves to power the ADXL345 accelerometer, seen just under the microcontroller in the schematic. The SCL, SDA, SDO, and active low CS pins on the ADXL345 are connected to the P2.3, P2.4, P2.5, and P2.2 digital GPIO pins on the MSP430FR2311 to allow for SPI communication between the two. The final subsystem that uses the 3.3 volt regulated power is the experimental chamber lighting. This simple circuit uses 4 light emitting diodes (LEDs) to allow us to see what is happening in what would otherwise be a dark chamber. Placing a series-connected resistor before each of the LEDs ensures that the diodes operate within a safe current range. At the output of the 5.0 volt regulator circuit, we have a connection to the 5 volt micro USB port on the Raspberry Pi board. The Raspberry Pi will then have a connection to the Raspberry Pi camera module via ribbon cable. This subsystem will allow us to record the events occurring inside the experimental chamber. There will also be a WiFi adapter connected to the Raspberry Pi via one of its USB ports, as shown in the schematic.

Figure 6.1.1: Integrated Systems Schematic



Also seen in the integrated schematic are the connections between the power module, the flight controller, the telemetry antenna, the electronic speed controllers, the motors.

6.2 PCB Software and Vendor

In this section, we will briefly discuss what software package we have chosen to complete our printed circuit board design, and how we will proceed to have it manufactured.

In searching for appropriate printed circuit board design software, we found some popular choices like Autodesk's Eagle and Altium. Initially, we aimed to use the Eagle PCB design software because it seemed to be the most common, and most common usually means more component libraries and better support, among other advantages. However, the Eagle software was very heavy in the sense that it was a very large file to download, and it was taxing on the processing ability of the computer. This led us to return to our search for printed circuit board design software, this time looking for a package with a lighter load on the computer. Having heard about KiCad from others who had previously design PCBs, we decided to look into it. After researching the system requirements and finding that the software was, in fact, lighter than packages like Eagle, we decided to move forward with KiCad.

With the amount of high-quality and cost effective printed circuit board manufacturing options we have found available, it was not a difficult task to find a potential vendor. We decided to use PCBWay as our circuit board vendor. PCBWay had very competitive prices, and depending on the complexity of design, they offered 24-hour build times. Combine the cheap cost, 24-hour build time, and 3-4 day shipping estimates, and PCBWay was a clear choice. After receiving two different orders from this vendor, we can say that quality is not the highest, but acceptable for our needs.

6.3 Final Coding Plan

Initially we were unsure of the division of work between our Computer Engineering group members and our Computer Science colleagues when it came to coding the project. Eventually the code pertaining to unloading, interpreting, manipulating, and displaying data from the drone is done in some parts by Java and other similar object-oriented coding language with user-interface support. This code was one of the main responsibilities of our Computer Science colleagues as they had more experience dealing with object-oriented programming and were more prepared to make a standalone program capable of data interpretation and manipulation.

For the code running on the flight controller of the drone, the Computer Engineering members of our group worked to ensure proper initial setup while our Computer Science counterparts worked with specific functionality regarding custom flight maneuvers and autonomous flight planning. Most of the autonomous scripts running on the Raspberry Pi were done in Python using libraries which allowed the issuing of MAVLink commands. As for the code running on the microcontroller of the drone, this was handled by our Computer Engineering group members as they had experience directly dealing with hardware components (such as the experimental data sensors and serial communications) and did not require the experience of an individual in Computer Science. The microcontroller was programmed with the C-programming language using TI libraries to handle serial interfacing and clock manipulation.

After we received all of our parts we were able to distribute components to realize our coding plan and break down the responsibilities between the Computer Engineers and the Computer Scientists.

7 Prototype Test Plan

In this chapter we discuss our hardware and software test environment, along with the hardware and software specific testing that we did. We explain the areas we had to use for hardware and software testing and the tests that we performed to test our project. The testing environment utilized is very important because if you have a testing environment that is nothing like the environment that you will actually be operating your design in, it will be useless. So we wanted to make sure that we were testing our hardware and software in an environment that was similar to where our drone would actually be operating. As for our hardware and software specific testing, we describe how we tested them using various methods based upon the testing that the component needed.

7.1 Hardware Test Environment

With regards to the hardware systems of the drone, a majority of testing needed to be done out in the field following some in-lab testing. Starting with drone components such as motors and actuators, attaching the components to the drone frame happened place in a lab or at a specific group member's home. One of the locations assembly was done was the Texas Instruments (T.I.) Lab in the engineering building on campus or the neighboring machining workshop. With the sensors, antenna, microcontroller, and flight controller, there was a need to connect all the components to a main circuit board. This main circuit board and all of the connections were thoroughly tested to ensure they received adequate power without causing any damage to the PCB. In the Senior Design Lab on campus, there was electrical testing equipment such as multimeters/oscilloscopes to test the connections. Most of our hardware tests took place in the Senior Design lab to verify hardware on the ground before any kind of flight test. We also had a smaller test drone which enabled for electronics testing before final drone assembly. Testing on this drone allowed for minor flight tests without seriously endangering the electronic components or the final drone before it could be adequately stress tested.

Once we had most of our flight components, we moved our testing outdoors to run the motors and gather thrust data to determine if adequate thrust production was taking place before any take-off. For most of our low-altitude testing the drone could be flown in any public location without the need for FAA approval, which meant most testing was conducted in an open field where the area is clear of bystanders and private property. During our final testing we even attempted to reach out to UCF to attain proper flight permits on campus, but our tests concluded before any such permit could be acquired. For our final testing and competition flight which required a high-altitude flight to verify the carrying power of the drone's motors as well as the sensors ability to gather data we were going to use a vast stretch of deserted property in Palm Bay, FL. Although we were unable to get the drone in a state able to perform high-altitude flight, this location was within proper FAA guidelines and public property which would have allowed for thorough testing with minimal chance to cause any kind of damage or injury.

7.2 Hardware Specific Testing

Specifically for any testing involving the hardware, there needs to be comprehensive analysis of the various components on the drone. Breaking down the drone into components is essential before taking it out for any kind of flight testing as budgetary constraints will not allow for rebuilding the drone in the case of flight failure and drone damage. Starting at the drone's actual hardware such as any motors and actuators can be tested simply for proper connectivity to the drone components and to ensure no parts are damaged while being installed. With that done the main circuit board of the drone can be checked node to node with a multimeter to ensure all connections are stable and outputting the correct voltages. This step is essential as any errors in PCB assembly could lead to voltage surges in other components, possibly damaging these components or the PCB itself.

Before attaching any sensors or battery components to the PCB, they are tested individually on a breadboard to verify accuracy of sensor measurement at ground level as well as battery electrical integrity. Once on the PCB, a multimeter is used to verify the integrity of the electrical connections as well as the integrity of the circuit itself to mitigate the risk of voltage surges. Once each of the PCB power distributions is tested, all the components secured on the drone, a short "test flight" using the flight controller's open source software can ensure that the flight controller works as a component and in its functionality of engaging or disengaging various drone components specific to flight maneuverability. The drone may not be able to fly at this time without careful calibration of the flight controller, but this is mostly a software concern. Attaching the microcontroller which will control the sensors will be an essential step towards actually measuring the impact of a microgravity testing environment and completing the project goal. As such, the microcontroller would be added to the PCB, the PCB re-verified, and then the sensors tested by the microcontroller using a simple program which engages the sensors and verifies they are outputting similarly to how they did before drone attachment. Another crucial component for the drone to receive commands at a long distance is the long-distance antenna which can be affixed directly onto the drone and added to the PCB.

Testing of the drone's single-board computer will be a straightforward power-on check to determine whether the system is functioning. Then, following some software installations on the single-board's storage unit (a microSD card), there were tests for any defects. The camera module is then be attached or connected to the board, and run to verify that there are no manufacturing faults or damages to the camera. Finally, if all these systems work they are attached directly to the drone for later flight testing.

Afterwards, the drone’s flight controller is calibrated to ideally enable a short, low-altitude test flight to ensure the drone is able to fly without any kind of cargo-load. With the drone flying, a test load can be added to the drone in the designated load-bearing section to test whether the drone is still able to fly with an additional load as well as to ensure the flight controller’s calibration is holding stable. A solid connection via the antenna, as well as verification that the load is being carried with minimal flight impact is the last test to conduct before various software tests and before seeking further flight approval. Following approval from the FAA for high-altitude flights on our modified drone, a test can be conducted (without experimental load) to determine if the drone is capable of safely recovering from a free-fall before impacting the ground. At this time it is absolutely crucial that the drone be able to safely recover with plenty of altitude to spare as a failure following the addition of a test load would likely be detrimental to the project as a whole.

During this testing the antenna’s range would also be tested to ensure connection is stable throughout flight to help mitigate any kind of crashing (mid-air or otherwise). Should the drone have any connection issues, they would also have to be addressed prior to addition to a test-load during a high-altitude flight. Once the experimental test load is added, however, the drone’s full functionality is essentially tested to ensure it is able to recover from accelerated free-fall with additional weight. Another test flight follows this to include sensor measurements to ensure they function in capturing data throughout the test-flight. Essentially the drone is fully functional at this point in operation as per the project requirements, but additional hardware testing may include that of a base-transmitter connected to the ground-control computer. Should the computer fail or the transmitter be unable to establish long-range connections, the drone would need to run completely autonomously beginning at the time of launch.

7.3 Software Test Environment

For the software components of the project, there are a variety of environments to test the full functionality of both the drone and the ground-based control station. On the drone there is an on-board microprocessor to handle data collection, a single-board computer to handle live video transmission, and a flight controller which will handle drone flight operations. Software will also be running on the ground-station and will thusly need to be tested on there as well. After programming a dev-board for the microprocessor and single-board computer, the full extent of its functionality is tested with the drone on the ground and connected to a computer. On the computer, the program is run in the coding environment it was developed on through the drone’s microprocessor and relaying the data back to the computer. The microprocessor comes with its own development environment being programmed in C/Cpp. At this time debugging commences and any errors or abnormalities would be ‘ironed out’. Once perfected, the code is moved to the PCB microcontroller. As for the flight controller, it will need to be run on both the ground-control computer as well as the drone in order to be tested. The Raspberry Pi runs its own operating system where any testing related to the Pi is done. This includes the experimental video storage and receiving of data from the microcontroller, including the transmission of the live video, which requires the transmitter/receiver pair be setup between the “ground” and drone. As long as the drone meets FAA regulations, we are able to test the drone’s flight capabilities and the flight controller’s functionality at any physical location where it is not going to be obstructed (such as a city with tall buildings). Ideally out in an open field the drone can be launched, simple flight maneuvers tested, and landed to verify that the flight controller has been properly configured.

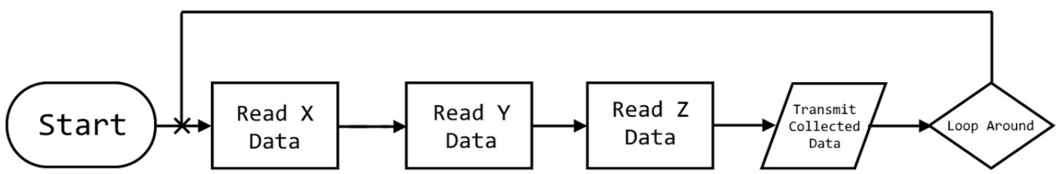
After a successful simulation using the ground control software and a successful live test, the next test takes place on the ground-control computer where a program for analyzing and displaying the experimental data is tested. Eventually, however, the program needs to be standalone in which case it runs through an executable on Windows based machines for easy access and setup. Once all these factors are in place, the final testing environment varies drastically depending on various factors. Firstly, we verify that the drone and its software are completely within FAA guidelines before any kind of significant flight test, as to not violate any legal constraints. Secondly, we seek permits related to flying the drone at high altitudes from the FAA, and find an open field to enable safe testing without the fear of damaging property or injuring bystanders. Alternatively, we reach out to our sponsors Northrop Grumman to determine if they have access or knowledge of any flight-zones which we test in without FAA approval as they are already designated for drone/aircraft flight. Whichever route is chosen, the final testing will take place in an open field where the software will run on both the drone (on its microcontroller and flight controller) and the ground-station computer. In that location we will be able to test the full functionality of the drone’s experimental load capabilities, data collection accuracy, and overall flight maneuver success.

7.4 Software Specific Testing

In order to fully test the software components of the project it is broken down into the various components which have code to test. Starting at flight data, the sensors are programmed to activate and record data. The details of the testing already done and to be done is discussed in this section. Furthermore, a flowchart can be seen with is accompanying explanation.

This is tested by having the drone microcontroller connected directly to a computer and running a simple program to activate any sensors attached and collect their data for a short period of time by sending it to be stored in the data unit of the drone. Once the data has been verified (not for accuracy of the measurements but for the existence of the content), the microcontroller is activated and left running for several minutes to confirm data is being sent out over the entire duration of a “flight”. A simple program which follows the flowchart shown in [Figure 7.4.1] is used at this point in testing to verify the microcontroller is properly using SPI communication and sending out the data to be interpreted by the single-board computer’s processor.

Figure 7.4.1: MCU software flowchart.



Another component with a software portion of testing is the single-board computer being used to store and transmit experimental data. For the first portion of testing, a simple operating system is installed onto the Pi to enable further use for the project. Once booted up, a program which enables the use of the camera (whether that’s drivers and a command line or a full on interface) is installed and tested. If a Pi inherent operating system is used with a Pi camera module, the command line of the operating system should allow for the downloading and installation of camera drivers as well as commands which enabled the camera to record and store data. These are in the form of Python/C libraries which emulate video via collections of quick succession pictures, and are to be checked for accuracy of video quality and integrity of overall content.

Once the video integrity is verified, the next program to be tested should be that of the external transmitter for the Pi’s experimental feed. Testing this in the simplest form involves setting up the transmitter with the Pi and setting up the receiver with a computer to determine if the two systems can communicate. At first the data being sent is just video which has been tee’d and simultaneously becomes live video as it stores locally. In order to progress further, the microcontroller is tested such that it is sending the sensor data properly to the Pi and the Pi is able to interpret that data and work with it. This system is to be considered working once the microcontroller is shown to successfully send data to the Pi, and then the Pi successfully transmits live video and the received data to the “ground” computer. Included in this consideration is the redundant storing of data on the Pi’s SD card.

Next comes the addressing of the drone's flight control communication with the ground station's receiver. Breaking this down a bit, a single program with multiple functions (or several programs with singular functions) is setup to ensure strong communication between the drone and the "ground-station" by first attempting a connection and transfer of data. After a successful connection, the connection is verified as secure and be able to continuously transfer data such as altitude. To test whether the connection holds at high altitudes, the drone and ground-station computer can be separated across various distances of ideally open field (to minimize signal noise) or less ideally across a long distance (such as one end of the university to another). If the second test is used the range can be expected to be reduced as the signal may encounter large amounts noise, but the further the two devices can be separated before this kind of disconnect is too great, the further it can be somewhat assumed the drone can fly vertically.

Addressing the base-station, testing out the software is to be done in two-parts. The first part is integration of open-source code for the drone's flight controller which should provide a multitude of flight functions right at setup. Using, simply, the base-code provided we are able to see that any provided functions related to the drone's flight patterns and data relaying work as intended. The drone's live experimental camera needs to be integrated into the program or displayed in a separate window such that it live transmits the feed with minimal data loss. Once the ground-station is shown to activate and view the camera, similar range testing to that of data transmission described earlier in this section can be used as well as further testing showing the full extent of the camera's functionality. Then the base-code is altered to allow for new parameters or flight functions such as the acceleration assisted drop which will simulate the microgravity environment as well as an altitude recovery maneuver to ensure the drone does not crash. Testing these functions does not occur until the code can be absolutely verified to work in such a way that the drone is not damaged based on failure to recover from free-fall. A way of testing this somewhat safely is to tether the drone to the ground in some fashion that allows us to observe how long the downward acceleration persists and at what conditions the recovery functions trigger.

Using various mathematical and physics-based formulas, it is determined whether the descent is the proper speed to create the testing environment as well as approximately what altitude the drone needs to begin altitude recovery as to not crash into the ground. After extremely cautious analysis, the functionality of these new maneuvers is tested at low altitudes which should allow for recovery of drone components in a worst-case scenario program failure. Should the program work as intended, however, the ground-station should have proper calibration to be used for the entire flight control system of the drone at the time of full functionality testing. Any safety features need to be assessed and possibly overridden to allow for full experiment functionality. Following this testing, the ground-station must also have a program that allows for retrieving, manipulating, and displaying collected data and video in a useful way once the drone has concluded its flight. As the sensors have already been tested to determine that they work and collect data, the ground-station must now show that it can access the data following a successful flight.

One final program that may be necessary involves automating the entirety of the process from drone launch to final data display, in such a way that human interaction is unnecessary. With this the drone needs functionality to transmit all stored data upon landing, reporting on the status of the battery or batteries, and possibly entering a standby mode which reduces power consumption until a launch initialization request is made by the ground-station. Ultimately this is not necessary as long as the remaining functionality is present, but would be an interesting result which may entice researchers seeking to use this method to invest due to the sheer simplicity of the procedure.

8 Results and Lessons Learned

Following all of our component testing and several integration meetings, we managed to have the fully assembled drone ready for test-flights before the final competition flight. Initially we encountered issues with our first few test flights where manual RC control caused the drone to capsize. Also, the drone had a drifting issue under manual control where the drone gentle drifted pitch down until set back on the ground. Later test-flights corrected this issue over several iterations of manual trim tuning which took place over the course of several days. Eventually we moved on to autonomous testing but ran into a similar capsizing which we were looking to address. Another manual flight, however, proved unfavorable and the midsection of the drone collapsed under the impact of a crash preventing any further testing. During that final flight we did manage to record the impact and accelerometer data which was successfully stored on the Raspberry Pi's SD card.

With that in mind there were many things to learn from this setback starting with the most important which dealt with the nature of our project. Interdisciplinary work has many different people working on different things that eventually need to come together; a delay in one group can lead to a cascading delay of sorts especially when hardware components are limited. As such, we learned that working closely with our teammates constantly to keep a solid timeframe is vital as well as ordering sufficient extra hardware such that multiple people can all work at the same time. We also learned that ordering components needs to be thoroughly done the first time as waiting for part re-orders can take weeks to address which are massive slowdowns projects such as ours cannot always afford. These are among the most valuable lessons we learned from this project along with lessons we each learned individually during our time on the project.

9 Administrative Content

In this chapter we discuss how we stayed on track using milestones and we also talk about our budget and finances. A timeline of when we set goals for various tasks to be completed can be seen in this chapter along with why we felt such a timeline was important. Additionally, in the budget and finance discussions we discuss how our limited budget impacted us and how we feel dealing with being on a team and only having a limited amount allocated to us opened us up to how things are in the real world.

9.1 Milestone Discussion

In this section, we will lay out our project milestones for Senior Design 1 and Senior Design 2. We will also discuss the driving forces behind our selected goals and dates, as well as how we plan to get from one milestone to the next.

Table 9.1.1: Senior Design 1 Project Milestones

Senior Design 1	
Select Project	8/31/17
Divide and Conquer Document	9/8/17
Updated Divide and Conquer Document	10/20/17
Submit 60 Page Document	11/3/17
Final Components List	11/10/17
Submit 100 Page Document	11/17/17
Breadboard Testing for PCB	11/20/17
Finalize PCB Design	11/27/17
Submit Final Document	12/4/17

Table 9.1.1 shows some definite due dates for when we had to turn items in, and also some tentative dates that we listed as goals for when we wanted to have certain tasks completed or for progress to be made on them. It was important for us to keep track of project milestones that we were meeting or failing to meet so that we could get back on track and successfully complete the project. Additionally, we added more milestones for senior design 2 as the months passed and we had a better grasp on what was achieved during our first semester and what needed to be done in the next.

As can be seen above we wanted to have our final components list by November 10th because we wanted to have them ordered by that date. We felt that this was imperative because we knew that we needed to begin testing as soon as possible and that things may take longer than expected to arrive. We actually aimed to order the parts as soon as possible, but our absolute deadline was the 10th of November.

Table 9.1.2: Senior Design 2 Project Milestones

Senior Design 2	
Order PCB	1/8/18
PCB Verification Testing	1/15/18
Substitute Drone Electronics Testing	1/22/18
Build Prototype	3/9/18
Prototype HW/SW Integration	3/16/18
First Prototype Test	3/23/18
Initial Prototype Update	4/6/18
Final Presentation	4/18/18
Submit Final Report	4/27/18

We made a tentative list of milestones for senior design 2 with things that we knew that we would have to do just to give us an overall view on the semester. We knew that we would have to build and test the prototype, make modifications to that prototype, and make a final presentation. So we felt that although we did not yet know all of the specifics, adding these milestones was helpful to keep us mindful of what needed to be done.

9.2 Budget and Finance Discussions

In this section the impact budget and finances had on our project and on decisions that we made are discussed. We speak about how we had to use our interpersonal skills to make sure that we had the funding that we needed for our components and also how we had to get creative in order to work within the budget that we were allocated.

9.2.1 Budget

Table 9.2.1.1: Estimated Budget

Block	**Item**	**Price (estimated)**
Transmitter/ Receiver	Long-range wifi Tx/Rx	$ 50
	High-gain antenna	$ 50
Power Source	Rechargeable Battery	$ 300
	Battery Charger	$ 100
Flight Controller	Flight controller and peripherals	$ 150
Telemetry Sensors	Accelerometers, Temp sensors, barometer, magnetometer, etc...	$ 100
Environment Sensors	Temp sensors, high precision accelerometers	$ 200
Video Recorder	720p or greater video camera	$ 300
Media Storage Device	> 4 GB memory	$ 50
Drone FPV Video	240p or greater video camera	$ 300
DPDU	PCB design	$ 100
Development and Test Substitute Drone		$ 300
Total Estimated Budget		**$ 2000[1]**

Table 9.2.1.1, shown above, was a tentative budget that we started out with knowing that we would update it as more details about the project are obtained.

The above [Table 9.2.1.1] was our initial budget but we found that we were able to get most of what we need for the amount shown in [Table 9.2.1.2].

Table 9.2.1.2: Actual Costs to Date

Block	**Item**	**Price (estimated)**
Transmitter/ Receiver	Long-range wifi Tx/Rx	Included in flight controller kit
	High-gain antenna	Included in flight controller kit
Power Source	Rechargeable Batteries	$111.60
	Battery Charger	Free
Flight Controller	Flight controller kit and peripherals	$87.11
Telemetry Sensors	Accelerometers, Temp sensors, barometer, magnetometer, etc...	Included in flight controller kit
Environment Sensors	Temp sensors, high precision accelerometers	$17.95
Media Storage Device	> 4 GB memory	$13.99
Experiment Chamber Camera	240p or greater video camera	$ 23.49
DPDU	PCB design	$240.05
Development and Test Substitute Drone		$274.39
Motors		$482.79
Electronic Speed Controllers		$277.80
Structural Materials		$380.12
Total Expenses to Date		**$1909.29**

As can be seen by [Table 9.1.1.2] above, so far our total expenses have only been $1909.29, this is great since our mechanical and aerospace teammates also need to order parts and will have considerably higher costs than we will. By cutting our costs as much as possible we will be giving them more budgetary flexibility.

9.2.2 Finance Discussions

We came up with the values in our budget by doing a quick check on what similar component parts seemed to be going for in the market. For those things that we weren't really sure of we had to make an educated guess of what we thought it would be. This posed some problems as when we actually began doing heavy research into what parts to select we realized that some components were more expected than we expected. This led to us having to be creative in order to work within our budget. For example, rather than just getting one fancy transmitter and receiver that included all of our telemetry needs, we had to decide rather, to go a different route. This different route was to get a simple remote that would allow us to initially control the drone and then a separate telemetry set. It would have been easier to just get it all packaged but our budget did not allow for that, so rather than going over budget where we didn't necessarily have to, we found a workaround. This is just one example of where budgetary constraints forced us to be creative when selecting parts. Although it was difficult, we feel that this was good practice and preparation for the real world as when we are working for a company and given a budget, we will not be able to just reach into our pockets and decide to fund it ourselves if we go over budget, no we will have to strictly adhere to the amount given us while still meeting the task.

Another wrinkle to our budget and finances was that of collaboration. As an interdisciplinary project we are also working with mechanical and aerospace engineering students along with computer science students. Our whole project was given a budget and as such we had to think about them when making our budget and purchasing parts. We couldn't just simply buy the top of the top equipment and forget about all of the costs that go into the building of the actual drone structure. That would be unwise and unkind and in the end it would ultimately not result in a better product or happy teammates. This is another real world preparation since good relationships with co-workers are essential for product success. No one is going to want to work with a team that is always thinking only of themselves rather than the group effort and as such could result in a lack of productivity and innovation.

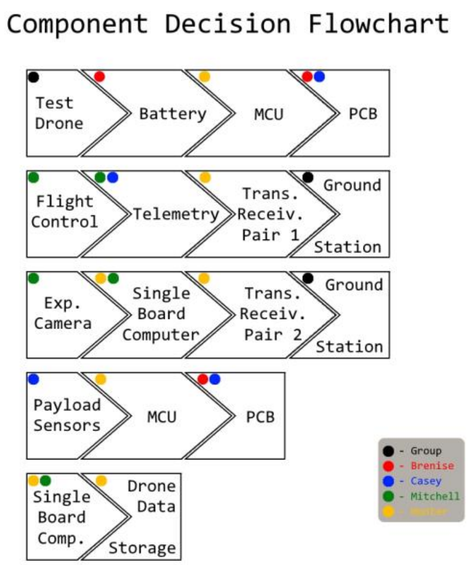
With this being said, we are immensely grateful to Northrop Grumman for sponsoring our project. As college students we certainly do not have the funds to finance a project like this and it would not have been possible without their support. Company leaders who decide to sponsor student projects and research are fostering an environment of learning that they could eventually reap the benefits of. We say this because of all the things we learned so far in this project involving budgetary constraints and teamwork when it comes to finances.

9.3 Project Member Responsibilities

In this section we will be breaking down who was leading the project group on the various systems which would constitute our test drone. We also show the flowchart used to decide who would be choosing various components and the component choices which would follow as a result of the choice. Specifically, if a project group member was in charge of a particular system the chart assigned them to research and make the decision on components related to that system or to review the decision made by another member to ensure that system was being adequately served.

We found that this flowchart was especially helpful to us when it came to project planning and decisions. For example, once we determined the lead of power systems we didn't need to wonder who was looking into battery selection. Additionally, the person in charge of power systems knew that while their partner as shown earlier in [Figure 2.5.1], is there to help and lend advice, ultimately there is some freedom for that power system lead to select the best battery and present it to the group. This is why we established leads for each system, we found that without having a dedicated lead, everyone was left in an unclear state.

Figure 9.3.1(a-b): Compilation of figures representing project member responsibilities. Figure 9.3.1(a) is the Component Decision Flowchart (left chart and key) and Figure 9.3.1(b) is the Test Drone System Leads table (right table).



Test Drone System Leads

System	Lead
Test Drone	Brenise
Power Systems	Brenise
MCU/Data Storage	Hunter
PCB	Casey
Flight Controller	Mitchell
Sensors	Casey
Communications	Hunter
Single-Board Computer	Mitchell
Experimental Camera	Mitchell

In the figure above [Figure 9.3.1(a-b)] both the Component Decision Flowchart and Test Drone System Leads table have been included. For the former [Figure 9.3.1(a)], the flowchart breaks down the various systems of the test drone into the necessary components and relates them to other dependent components.

There were four components which could be chosen at the beginning of this narrowing process, with a fifth component flow (the vertical bottom box) being derived from a previous flow (the vertical middle box). When a group member chose a particular component (marked by colored dots) that component's specifications were then taken into account in regards to the component choices which followed (i.e. with knowledge of the Test Drone, we were able to choose an adequate battery to power the systems).

Also in the figure, our Test Drone System Leads table [Figure 9.3.1(b)] shows the breakdown of our test drone into the systems which relate to it and the project group member who was responsible for leading in the development of that system. Each system lead was to choose the components which would best fit the particular system and ensure component cohesion between interconnected systems. Additionally, the development of a particular system was to be done by or reviewed by the system leads once the assembly of the test drone began. As such, a system lead determines if the system is going to fulfill the assigned purpose adequately and ensures it is implemented correctly.

10 Conclusion

This section contains a brief summary of each section in the report. Enough details are given in order to give the gist of each section however unimportant details are withheld to make for quick and easy reading.

In chapter 2 entitled Project Description, we describe our project motivation and objectives along with our requirement specifications. These motivations were the lack of inexpensive, easily accessible, quality microgravity testing options. Our objectives were to be able to fill this void in the scientific research and experimental sphere with a drone based solution. Our goal was to provide a drone that could provide an experimental chamber where microgravity testing could be conducted easily and at a low cost. As for our specific requirement specifications and target engineering goals, they are depicted succinctly in our House of Quality diagram (Figure 2.4.1). They include: being able to reach a max altitude of greater than 300 feet, having greater than 16GB of storage capabilities, the ability to have an experimental material load greater than one pound, a flight duration of more than one minute, to produce a reduced gravity environment where the gravity in the experimental chamber is less than 0.2g, that the reduced gravity environment last for more than 3 seconds, and lastly that it is within a budget of $2000.

In chapter 3 entitled Research Related to Project Definition, we discuss the research we conducted to discover the extent of the need for our product and similar products were currently available along with the relevant technologies. Furthermore, we also discuss part selection and possible architectures for our project. In terms of similar products we discuss The Center of Applied Space Technology and Microgravity in Germany and their drop tower along with ZERO-G's parabolic flights to discuss the quality and duration of microgravity that they provide. In the relevant technologies subsection the following technologies were explored: flight controller, flight control software, power source, communication system, and sensor and navigation. As for the part selection subsection it is quite lengthy, we dive into the process that went behind our selection of a flight controller, telemetry sensor, transmitter, receiver, power components, electronic speed controllers, batteries, and test drone.

In chapter 4 entitled Related Standards and Realistic Design Constraints, we discuss standards and how they impact our design along with realistic design constraints such as economic, time, environmental, social, political, ethical, health, safety, manufacturability, and sustainability constraints. We discuss the regulations in place for drones and the complex standards that surround them, additionally, we discuss how we have confidence (and will ensure to a certain extent) that our mechanical and aerospace engineering teammates and our computer science teammates will adhere to these standards. In terms of realistic design constraints we address each of the aforementioned constraints. The economic and time constraints section expounds on the restrictions that our budget and limited amount of time had on our design. Had we had an unlimited amount of money and time we would most definitely have been able to produce a better product but we had to realize our economic and time constraints and work within them to produce a great product regardless. The environmental constraints are just simple and obvious, we are trying to produce a reduced gravity environment while being on the earth which has a gravitational pull of approximately 9.81 m/s^2, which significantly affects our design. As for the social and political constraints, we felt these were less of an issue for our type of project. Our project is one aimed at a specific niche of the scientific community, those desiring to perform microgravity experiments in an inexpensive manner, and as such there aren't many social or political constraints. The ethical, health and safety constraints on the other hand, especially the safety constraints, applied more to our project as we are working with a drone that could cause potential injury if it were to fall on someone. As such we definitely thought about ways we could maintain public safety and the safety of the user of the drone and those are outlined in this subsection. Lastly in this section was the subsection of manufacturability, and sustainability constraints, we addressed this as best as we good seeing that we are not intending for this to be a widely produced design.

In chapter five entitled, Project Hardware and Software Design Details, we elaborate on our design and break it down into five subsections. These subsections are breadboard test, and schematics, flight controller, experiment payload, microcontroller and data storage, and transmitter/receiver. In each section we further expound on each topic and how it relates to our design. In the Initial Design Architectures and Related Diagrams section we provide an abstract architecture diagram for the drone's electronic systems. This is a conceptual figure that gives a very helpful overview of the drone's electronic systems. Furthermore, section 5.2 deals with the printed circuit board design, breadboard tests, and schematics. This is quite a large section since it is very important and requires quite a few images and explanations to properly explain what is being done. Another long section is that of section 5.7 which deals with the software design. Many tables were presented and images that explain the software design and the research that was done on software.

In chapter six entitled, Project Prototype Construction and Coding, our integrated schematics are placed and information is given about our printed circuit board design. Additionally, information about our PCB vendor and assembly along with our final coding plan is given. Section 6.1 contains an integrated systems schematic that shows all of the connections that were made between the printed circuit board and other subsystems. When we received all of our parts we were able to begin assemble our test drone and started developing prototypes of our control system for the actual drone. This leads us to our next chapter in which more about testing is discussed.

In chapter seven entitled, Prototype Testing Plan, we discuss our hardware test environment, hardware specific testing, software test environment, software specific testing. In this chapter we discuss the specific testing of the hardware and software that we did. We discuss how we used the Texas Instruments lab on campus for preliminary testing of the hardware. Next we explained how after all of the components were attached to the drone and we were ready for flight, we had to find an area where we could safely test the drone. We had to find an area that has low traffic as to ensure the safety of the public. We also discuss how individual components such as sensors were able to be breadboard tested before actually attaching them to the drone.

In chapter eight entitled, Results and Lessons Learned, we briefly discuss the outcome of our project between our whole interdisciplinary team and the lessons we learned by participating in this project.

Lastly chapter nine deals with administrative content such as milestone, budget, and finance discussions. This section is clear cut, we discuss the goals we set for ourselves to ensure that we remained on track and towards a successful project, along with a budget to make sure we remained within our means. We discuss the importance of setting goals both big and small, and provide two tables which detail the milestones and their corresponding dates. We additionally discuss a little bit more about the impact our budget had on us and how finances caused us to be more creative in part selection. Furthermore, we discuss how having to work within a budget forced us to work with our mechanical and aerospace engineering teammates as well as our computer science teammates. We couldn’t just pick whatever we wanted regardless of what our other teammates thought, we had to collaborate with them and listen to their thoughts and ideas on various parts and methods.

Appendix

Appendix A - Copyright Permissions

Figure A.1: Copyright Permission for WiredCraft Figure (3.2.4.3.1)

Copyright Permission   Inbox x

Casey Colón [redacted]   Nov 27 (3 days ago)
to info

Hi there,

I'm an Electrical Engineering student at the University of Central Florida. I'm working with a group in Senior Design and we are designing a drone system. One of the communication technologies we have considered is the 4G network connection, as described in the blog titled "Make your personal drone fly even farther with a 4G network connection", authored by Charlie Li in May of 2016. I'm writing to request permission to use certain pictures, diagrams, and information from this blog in the form of communication technology comparisons in our report. All of these will be cited of course. Please reply at your earliest convenience.

Thank you

...

Dominika wydmuch <dominika@wiredcraft.con   Nov 29 (1 day ago)
to me

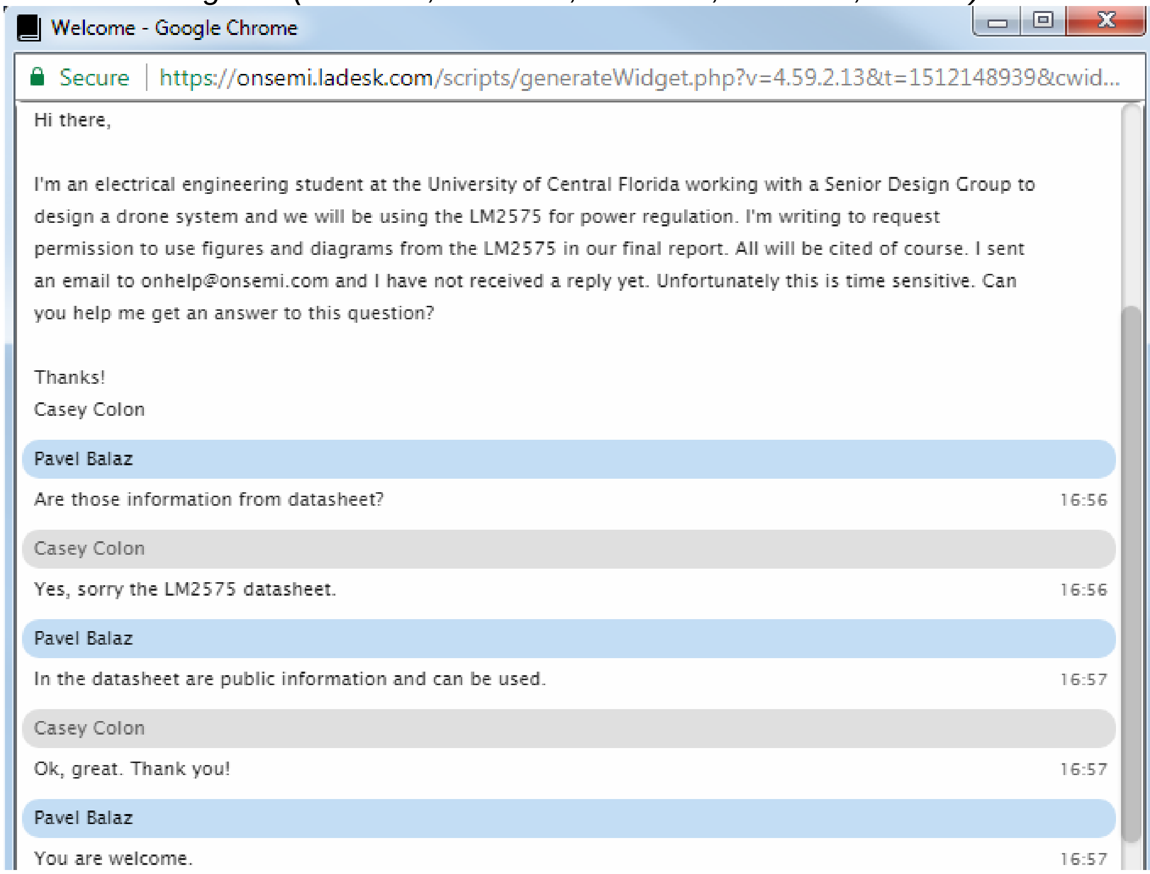
Hi Casey,

You can go ahead! We would be happy to receive a copy of your report in the future.

Good luck and have a nice day.

Best,
Dominika

Figure A.2: Copyright Permission for ON Semiconductor LM2575 Datasheet Figures (5.2.3.3.1, 5.2.3.3.2, 5.2.3.4.1, 5.2.3.4.2, 5.2.4.2)



Appendix B - References

[1] Gozerog.com. (2017). Zero Gravity Corporation.Retrieved 9/5/2017, from https://www.gozerog.com/

[2] Microgravity and Drop Towers. (not dated). Retrieved 10/29/2017, from http://www.esa.int/Education/Microgravity_and_drop_towers

[3] Reduced-gravity aircraft. (09/27/2017). Retrieved 10/30/2017, from https://en.wikipedia.org/wiki/Reduced-gravity_aircraft

[4] What is the Vomit Comet? (04/30/13). Retrieved 10/31/2017, from https://www.livescience.com/29182-what-is-the-vomit-comet.html

[5] Drop Everything! (04/10/09). Retrieved 10/31/2017, from https://www.nasa.gov/audience/foreducators/k-4/features/F_Drop_Everything.html

[6] Where to Fly. (07/26/2017). Retrieved 10/31/2017, from

https://www.faa.gov/uas/where_to_fly/

[7] Common Power Module — Copter documentation. (not dated). Retrieved 11/01/2017 from, http://ardupilot.org/copter/docs/common-3dr-power-module.html

[8] Flight Controllers: The Processor Behind Every Multi-Rotor Flight. (6/03/2014). Retrieved 11/02/2017 from, http://www.tomshardware.com/reviews/multi-rotor-quadcopter-fpv,3828-2.html

[9] The Center of Applied Space Technology and Microgravity (ZARM) Bremen Drop Tower information. (2017). Retrieved 11/01/2017 from, https://www.zarm.uni-bremen.de/en/drop-tower/general-information.html

[10] ArduPilot. (10/11/2017). Retrieved 11/02/2017 from, https://en.wikipedia.org/wiki/ArduPilot#Common_to_all_vehicles

[11] OpenPilot. (10/11/2017). Retrieved 11/02/2017 from, https://en.wikipedia.org/wiki/OpenPilot

[12] Attitude and heading reference system. (07/20/2017). Retrieved 11/02/2017 from, https://en.wikipedia.org/wiki/Attitude_and_heading_reference_system

[13] Cleanflight. (not dated). Retrieved 11/02/2017 from, http://cleanflight.com/

[14] Understanding R/C Brushless Motor Ratings. (not dated). Retrieved 11/02/2017 from, http://www.dewsburymac.org.uk/Understanding%20Brushless.pdf

[15] A Guide to Understanding LiPo Batteries. (not dated). Retrieved 11/02/2017 from, https://rogershobbycenter.com/lipoguide/

[16] Lithium Ion vs. Lithium Polymer Batteries - Which is Better? (06/20/2017). Retrieved 11/02/2017 from, http://blog.ravpower.com/2017/06/lithium-ion-vs-lithium-polymer-batteries/

[17] A beginner's guide to accelerometers. (not dated). Retrieved 11/02/2017 from, https://www.dimensionengineering.com/info/accelerometers

[18] Accelerometer Specifications: Deciphering an Accelerometer's Datasheet. (03/18/2016). Retrieved 10/16/2017 from,

https://blog.mide.com/accelerometer-specifications-decoding-a-datasheet

[19] Nickel-based Batteries Information – Battery University (not dated). Retrieved 11/02/2017 from, http://batteryuniversity.com/learn/article/nickel_based_batteries

[20] Magnetometer (not dated). Retrieved 11/02/2017 from, https://en.wikipedia.org/wiki/Magnetometer

[21] Altimeter. (09/04/2017). Retrieved 11/03/2017 from, https://en.wikipedia.org/wiki/Altimeter

[22] How To: Pick an Altimeter. (01/04/2016). Retrieved 11/03/2017 from, https://www.mountaineers.org/blog/how-to-pick-an-altimeter

[23] Pitot tube. (08/24/2017). Retrieved 11/03/2017 from, https://en.wikipedia.org/wiki/Pitot_tube

[24] Federal Aviation Administration Part 107 Summary (06/21/2016). Retrieved 11/02/2017 from, https://www.faa.gov/uas/media/Part_107_Summary.pdf

[25] Federal Aviation Administration Part 107 Waiver Application (07/14/2017). Retrieved 11/02/2017 from, https://www.faa.gov/uas/request_waiver/

[26] IPC-A-600H: Acceptability of Printed Boards (04/13/2010). Retrieved 11/02/2017 from, http://shop.ipc.org/IPC-A-600H-English-D

[27] IPC-6012 or IPC-A-600 Which Standard is Right One For Me (07/20/16) Retrieved 11/03/2017 from, https://www.eptac.com/wp-content/uploads/2016/06/eptac_07_20_16.pdf

[28] J-STD-001F: Requirements for Soldered Electrical and Electronic Assemblies (02/2012). Retrieved 11/03/2017 from, https://elibrary.gsfc.nasa.gov/_assets/doclibBidder/tech_docs/IPS-J-STD-001F.pdf

[29] IEEE 802.11 Wi-Fi Standard (not dated). Retrieved 11/03/2017 from, https://en.wikipedia.org/wiki/IEEE_802.11

[30] Wireless Standards 802.11a, 802.11b/g/n, and 802.11ac (09/29/17). Retrieved 11/03/2017 from, https://www.lifewire.com/wireless-standards-802-11a-802-11b-g-n-and-802-11ac-816553

[31] Coding Conventions (not dated). Retrieved 11/03/2017 from, https://en.wikipedia.org/wiki/Coding_conventions

[32] Coding Standards: Why Do We Need Them, and How Do We Develop Them? (06/27/16). Retrieved 11/03/2017 from, https://tech.jet.com/blog/2016/06-27-coding-standards-need-develop/

[33] Temperature Sensors. (not dated). Retrieved 11/03/2017 from, http://www.electronics-tutorials.ws/io/io_3.html

[34] POSIX Threads Programming. (03/07/2017). Retrieved 11/03/2017 from, https://computing.llnl.gov/tutorials/pthreads/

[35] Long-range Wi-Fi. (not dated). Retrieved 11/03/2017 from, https://en.wikipedia.org/wiki/Long-range_Wi-Fi

[36] Make Your Personal Drone Fly Even Farther With a 4G Network Connection. (05/03/2016). Retrieved 11/03/2017 from, https://wiredcraft.com/blog/drone-copter-uav-4g-network/

[37] Beej's Guide to Network Programming. (June 8, 2016). Retrieved 11/03/2017 from, http://beej.us/guide/bgnet/output/html/singlepage/bgnet.html

[38] The Global Positioning System. (not dated). Retrieved 11/03/2017 from, https://www.gps.gov/systems/gps/

[39] F1, F3, F4, and F7 Flight Controller Differences Explained. (10/30/2017). Retrieved 11/03/2017 from, https://oscarliang.com/f1-f3-f4-flight-controller/

[40] What You Need to Know to Choose an Accelerometer. (10/17/2013). Retrieved 11/13/2017 from, https://www.digikey.com/en/articles/techzone/2013/oct/what-you-need-to-know-to-choose-an-accelerometer

[41] Types of Voltage Regulators. (11/4/2017). Retrieved 11/16/2017 from, https://www.lifewire.com/types-of-voltage-regulators-818851

[42] Switching Regulator Fundamentals. (9/2016). Retrieved 11/16/2017 from, http://www.ti.com/lit/an/snva559a/snva559a.pdf

[43] *Answer on the thread:* “Transmitting a video stream through a microcontroller’s wifi” (November 15, 2012). Retrieved 11/05/17 from,

https://electronics.stackexchange.com/a/50038

[44] USB video device class (not dated). Retrieved 11/05/17 from, https://en.wikipedia.org/wiki/USB_video_device_class

[45] Video conferencing with the Raspberry Pi (04/27/2015). Retrieved 11/05/17 from, https://www.element14.com/community/community/design-challenges/sci-fi-your-pi/blog/2015/04/27/video-conferencing-with-the-raspberry-pi

[46] How To Buy a Long Range WiFi Antenna (07/01/2016). Retrieved 11/05/17 from, http://www.wirelesshack.org/long-range-wifi-antenna.html

[47] What is the best WiFi antenna for me? (not dated). Retrieved 11/05/17 from, http://www.radiolabs.com/Articles/wifi-antenna.html

[48] TP-Link TL-ANT2424B 2.4GHz 24dBi Grid Parabolic Antenna (not dated). Retrieved 11/05/17 from, https://www.newegg.com/Product/Product.aspx?Item=N82E16833997212

[49] Premiertek ANT-OMNI-2415 Outdoor 2.4GHz 15dBi OMNI-Directional High-Gain N-Type Female Antenna for Wi-Fi (not dated). Retried 11/05/17 from, https://www.newegg.com/Product/Product.aspx?Item=N82E16833993017

[50] Link budget (not dated). Retried 11/05/17 from, https://en.wikipedia.org/wiki/Link_budget

[51] *Answer on the thread:* “Does altitude affect wifi signals?” (07/18/2014). Retrieved 11/05/17 from, https://electronics.stackexchange.com/a/115817

[52] *Answer on the thread:* “How to get good wifi range for my Raspberry Pi robot?” (03/21/2017). Retrieved 11/05/17 from, https://electronics.stackexchange.com/a/293800

[53] *Answer on the thread:* “WiFi Signal Boosting” (07/07/2012). Retrieved 11/05/17 from, https://electronics.stackexchange.com/a/35227

[54] Autel Robotics X-Star Premium Drone with 4K Camera, 1.2-mile HD Live View & Hard Case (White) (not dated). Retrieved 11/05/17 from, https://www.amazon.com/Autel-Robotics-X-Star-Premium-Camera/dp/B01B1H82LY/ref=sr_1_1?s=toys-and-

games&ie=UTF8&qid=1509919191&sr=1-1&keywords=Live%2BView%2BDrones&refinements=p_n_feature_six_browse-bin%3A7029316011&th=1

[55] *At video timestamps 3:31 and 3:40* Autel Robotics Tutorial: Getting to Know Your Remote Controller (06/08/2016). Retrieved 11/05/17 from, https://www.youtube.com/watch?v=AhYk9K0sh9Q&index=49&list=PLZtzsCyCj6TzudysYMYsRY73lZ_-79-P9

[56] *Answer in the Q&A section for [54] by Jaffer* (07/12/2016). Retrieved 11/05/17 from, https://www.amazon.com/ask/questions/Tx29WUTNL8DIXTL/ref=ask_al_aap_hza?_encoding=UTF8&sort=helpful

[57] 10 Reasons the Autel Robotics X-Star Premium is the Best Value Drone for Sale in 2016 (07/08/2016). Retrieved 11/05/17 from, https://vhdrones.com/blogs/news/10-reasons-the-autel-robotics-x-star-premium-is-the-best-value-drone-for-sale-in-2016

[58] What Are DSM RC Controllers and Receivers and What Do They Do? (last updated 07/16/2017). Retrieved 11/05/17 from, https://www.thoughtco.com/dsm-rc-controllers-and-receivers-2862526

[59] HKPilot Transceiver Telemetry Radio Set V2 (915Mhz) (not dated). Retrieved 11/05/17 from, https://hobbyking.com/en_us/hkpilot-transceiver-telemetry-radio-set-v2-915mhz.html

[60] 915Mhz Micro Whip Antenna (SMA-RP) (not dated). Retrieved 11/05/17 from, https://hobbyking.com/en_us/915mhz-micro-whip-antenna-sma-rp.html

[61] Hobbypower Pixhawk PX4 Flight Controller … FPV Quadcopter Multirotor (not dated). Retrieved 11/09/17 from, https://www.amazon.com/Hobbypower-Controller-Telemetry-Quadcopter-Multirotor/dp/B01NB9M746/

[62] ArduPilot Copter Documentation (2016). Retrieved 11/17/17 from, http://ardupilot.org/copter/index.html

[63] LM2575 - Adjustable Output Voltage, Step-Down Switching Regulator. (6/2009). Retrieved 11/10/2017 from, https://www.onsemi.com/pub/Collateral/LM2575-D.PDF

[64] Gstreamer Open Source Media Framework (09/18/2017). Retrieved 11/17/2017 from,

https://gstreamer.freedesktop.org/

[65] RC LiPo Battery Connector Types, Choosing, Soldering. (2017). Retrieved 11/30/2017 from, https://www.rchelicopterfun.com/rc-lipo-battery-connector

[66] Myfloridahouse.gov (2017). Retrieved 11/30/2017 from, http://www.myfloridahouse.gov/Sections/Documents/loaddoc.aspx?FileName=_h1027er.docx&DocumentType=Bill&BillNumber=1027&Session=2017

[67] Practical PCB Layout Tips Every Designer Needs to Know. (11/13/2015). Retrieved 12/01/2017 from, https://www.allaboutcircuits.com/technical-articles/practical-pcb-layout-tips/

[68] Tips for Designing PCBs. (not dated). Retrieved 12/01/2017 from, https://www.expresspcb.com/tips-for-designing-pcbs/

[69] Tips on how to properly design/layout a Printed Circuit Board. (not dated). Retrieved 12/01/2017 from, http://www.onmyphd.com/?p=pcb.design

[70] The Best Multilayer PCB Design Tips for Circuit Board Layout. (11/08/2017). Retrieved 12/01/2017 from, https://resources.altium.com/pcb-design-blog/the-best-multilayer-pcb-design-tips-for-circuit-board-layout

[71] The Top 10 PCB Routing Tips for Beginners. (not dated). Retrieved 12/01/2017 from, https://www.autodesk.com/products/eagle/blog/top-10-pcb-routing-tips-beginners/

[72] Omni Antenna vs. Directional Antenna. (2/27/2007). Retrieved 12/01/2017 from, https://www.cisco.com/c/en/us/support/docs/wireless-mobility/wireless-lan-wlan/82068-omni-vs-direct.html

[73] LM2576 - 3.0 A, 15 V, Step-Down Switching Regulator. (1/2006). Retrieved 03/03/2017 from, http://www.onsemi.com/pub/Collateral/LM2576-D.PDF