Senior Design II
Final Project Document

SCUAV

Safe Construction Unmanned Aerial Vehicle

Department of Electrical Engineering and Computer Science
University of Central Florida
Dr. Lei Wei

Group 13:

Alan Hernandez	--	Computer Engineering
Baian Elmazry	--	Computer Engineering
Nicola DaSilva	--	Computer Engineering
Veronica Love	--	Electrical Engineering

Customers: Dr. Lei Wei, Dr. Samuel Richie, Construction companies
Advisors: Dr. Gita Sukthankar, Mr. Saif Ghassan, Mr. Fred Cass

Table of Contents

List of Figures

List of Tables

1.0 Executive Summary

Safe Construction Unmanned Aerial Vehicle (SCUAV) seeks to revolutionize the construction industry through the use of autonomous drones capable of construction in metropolitan areas. Our project aims to address the issues of safety and cost in the modern day. This project implements the use of a drone to pick up pre-cut Styrofoam blocks from the construction material site and assemble a structure at the build site. The Styrofoam blocks represent the construction materials which would be used in a full scale design of our project.

Our project integrates various components to perform autonomous flight and to complete its task of stacking blocks. An essential component to our project is the PixHawk flight controller. This component takes information from its sensors to direct the flight of the drone. The Pixhawk sends pulse position modulation, PPM, signals to each of the drone's motors to control its throttle. In this way the Pixhawk determines the speed and direction of the drone. The Pixhawk not only receives sensory information, but also commands from a Raspberry-Pi for flight control. The Raspberry Pi is on board the UAV and communicates to a base station computer via Wi-Fi signals. The base station computer loads the program on the Raspberry-Pi. Python was used as the programming language for all of the autonomous flight. The base station computer in combination with a mobile application allows easy user interface with the drone. The mobile application has several buttons to initial the autonomous flight of the drone and the construction of the blocks. The options on the application are end, arm, disarm, take-off, and RTL, return to launch.

The drone begins its mission at the base station, flies to the construction site to obtain a block, and builds the blocks at the build site. The Raspberry-Pi uses the GPS coordinates of the base station, material site, and build site for the Pixhawk to navigate the drone to the precise location. While at the material site, the drone uses the OpenMV camera to recognize April tags on the Styrofoam blocks. And to grasp the Styrofoam block at the material site and place it down at the build site, the drone uses a rangefinder and a claw gripper. A custom printed circuit board, PCB, drives the claw and rangefinder and is connected to the Raspberry-Pi along with the camera. The OpenMV camera is capable of object detection and image processing and it indicates to the Raspberry-Pi when an April tag is detected. When the UAV is at the material site, the Raspberry-Pi detects an April tag and sends a command to the PCB for the claw to close. Before the claw closes, the correct distance needs to be detected from the rangefinder. The claw closes on a nub attached to the Styrofoam block. When the UAV is at the build site, the Raspberry-Pi sends a command to the PCB for the claw to open and place the Styrofoam block. The process of transferring blocks and constructing them at the build site is repeated until the structure is complete. The claw, rangefinder, and camera are all at the bottom of the drone's frame. At the completion of the structure, the drone returns to the base station. This overview serves as a bigger picture for our project. The decision for each of our components, their implementation in our project, and their integration with other components are described in the following sections.

2.0 Project Description

The project description section gives an overall view of what Safe Construction Unmanned Aerial Vehicle (SCUAV) is all about. The purpose of this project is to provide a safe solution to construction in metropolitan areas. A single user could deploy and operate an autonomous drone with minimal input for initialization of construction.

2.1 Motivation and Function

Our motivation for this project stems from our interest in drones and their potential applications in future. After watching a TED talk by Raffaello D'Andrea on drones, we were inspired by the potential of using drone technology to create complex structures. The video displayed teams of drones working together to complete various tasks such as building tall towers of staggered bricks and creating bridges by wrapping lines of rope together.

The future of automation will become a driving force in industry. Automation will touch industries in transportation, manufacturing, data analysis and more. Our hope for this project is to revolutionize the construction industry through the use of drones. Our group believes that there is a lot of potential for improving the traditional approach for building structures with the use drone technology.

We envision that drones will be especially useful in developing metropolitan areas. Using drones would be a solution that improves safety, is easier to use, more efficient, cheaper, and more compact that the traditional ways structures are built. We will be implementing our project using OpenCV and AI algorithms to automate the construction process.

2.2 Goals and Objectives

The drone should be light enough for a single person to transport and deploy. The drone should be able produce enough lift to transport a single construction material at a time with ease. The drone should be able to stay in flight until the construction of the structure is complete. The drone should be able to recognize the different objects in the construction site. It should also know when the construction material is correctly placed. The drone should detect when the construction is complete and return to base. The drone should house the processing unit for image recognition.

Our drone will be using a flight controller to control the flight motors as well as communicate with PCB. The flight controller will be used to communicate and receive data from the PCB, the inertial measurement unit, and the Raspberry-Pi. The PCB will play a significant role in this project by controlling the servo motors for the gripper that would grab each construction material for transportation, and determine the phase the drone is currently on such as searching for base station or retrieving object for construction. Our project flow diagram is shown in Figure 58. It will specify the timeline in completing the project in both Senior Design 1 and Senior Design 2.

2.3 Design Description

Our team designed a drone that selects blocks of Styrofoam using object recognition and moves them from one location to another to build a structure. The blocks of Styrofoam in our project represent the building materials used at a construction site. The two sites our drone travels to and from to transport materials are the build site and the raw materials site. Our drone places the blocks of Styrofoam in a particular order at the build site.

To create a structure out of Styrofoam blocks, the drone performs three main functions: object recognition, flight control, and microcontroller design. The first function of the drone that will be explored is the flight control. The flight controller receives sensory information from the IMU, barometer, magnetometer, and GPS. Based on this input sensor information, the flight controller directs the drone's motors to navigate the drone. The flight controller also adjusts the speed, direction, and angular position of the drone. For instance, one of the sensors used to navigate the drone is a GPS. With the information the GPS provides, the microcontroller knows the physical location of the drone. The GPS also provides information for navigating the drone in the lateral and longitudinal direction. Another sensor used by the UAV is an IMU. The IMU's accelerometer provides the speed of the drone to the flight controller. It is also used to provide the angular position of the drone based on the pitch, roll, and yaw angles provided by the IMU's gyroscope. The IMU's acceleration and rotational information help the microprocessor maintain stable flight. The magnetometer sensor helps orient the drone by detecting the magnetic field of the Earth. And the barometer sensor is used to measure the drones vertical distance from the ground. Along with the sensory information, the Raspberry Pi also sends commands to the PixHawk flight controller to control the flight of the UAV.

The drone controls the speed, direction, and angular position of the drone via electronic speed controllers, or ESCs. Based on the input information, the flight controller sends commands of how much electrical power to send to the ESCs. ESCs are devices which control the speed and angular motion of a servo motor. This is done by sending pulse position modulation, PPM, signals from the microcontroller to the ESCs. Based on these signals, the ESCs control the amount of power sent to the motors. The way the power is sent to the motors from the ESCs controls both its RPM and angular motion. Servos motors usually have an angular range from zero to 180 degrees. The motors in turn control the speed and angle of the propellers. And the propellers direct the navigation of the drone. That is how the flight controller navigates the drone and controls the drones speed, direction, and height based on input information. The engineering specifications for the design and the drone are shown in the House of Quality in Table 1.

Our project uses the OpenMV camera to detect the Styrofoam blocks. This camera is capable of object detection and image processing. Our group uses April tags as a simple way for the camera to view images from afar. April tags are similar to QR codes, but they are larger and can be detected at a further height. The camera is connected to the Raspberry-Pi and signals when an April tag is detected. Based on this information, the Raspberry-Pi knows to signal the PCB to open the claw and when to initiate the rangefinder to begin the process of closing the claw, since the claw only closes when the correct distance is detected from the rangefinder.

Our project uses an ultrasonic rangefinder to determine how far the drone is above a block. When the drone is within a certain distance from the block, the drone's gripper closes and grasps a block. The rangefinder is only used when closing the claw, it does not measure distances during the whole flight of the drone. Because of this, the rangefinder only begins detecting distances once the drone is above a block. The camera is used to determine when the claw should be opened and when the rangefinder should start measuring distances to close the claw once the block is within range. The camera is connected to the Raspberry Pi. When the camera detects an April tag, the Raspberry Pi communicates to the PCB to either open the claw or measure rangefinder distances to close the claw. Figure 1 illustrates a flow chart of how the synchronization of the claw, rangefinder, and Raspberry Pi works.

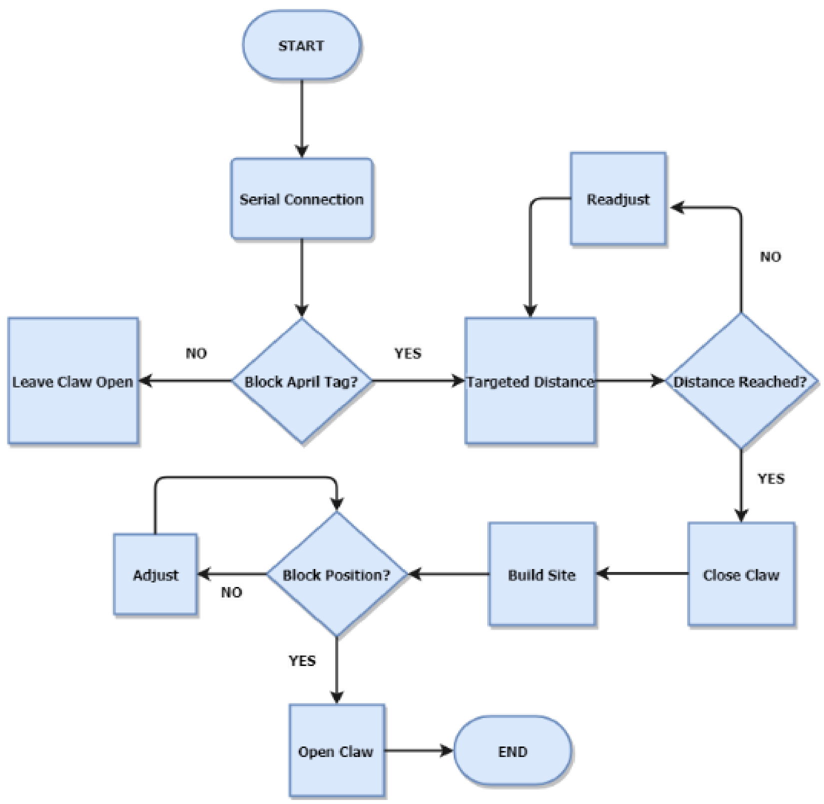
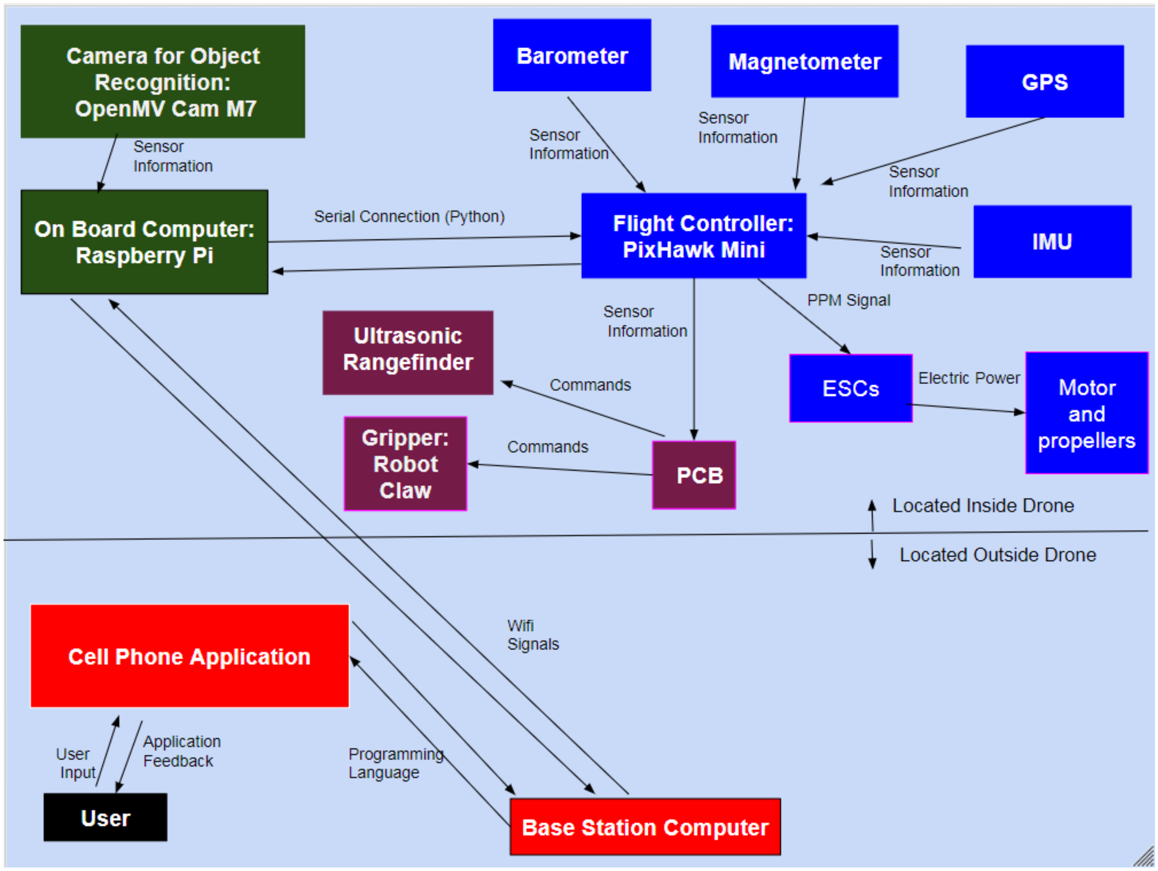


Figure 1: Claw, Rangefinder, and Raspberry Pi Flowchart

2.4 Block Diagram

The figure below displays the Block Diagram for this project. Each block is labeled with an important component required for SCUAV's design. Noted is the part each member plays in constructing this project.



LEGEND:

Figure 2: The Block Diagram Shows the Primary Responsibilities of the Team Members.

- ❖ To power the drone components such as the microcontroller, on board computer, gripper, GPS, rangefinder, camera, IMU, and the motors, a wiring harness or a power distribution board will be used.

❖ **Block Diagram Status:** All of the blocks are completed.

2.4.1 Secondary Responsibilities:

- Alan Hernandez aided Baian Elmazry with object recognition research and Nicola DaSilva with hardware setup.
- Baian Elmazry aided Alan Hernandez with developing drone's pathfinding algorithms as well as implementing the Bluetooth application.
- Veronica Love aided Nicola DaSilva in the flight controller microcontroller and electronics
- Nicola DaSilva will aid Veronica Love with the PCB design

2.5 Requirement Specification

- Overall Design Requirements
    - Max weight or 15 lbs., 10 lbs. for drone and 5 lbs. for base station
    - The blocks that the drone will be lifting need to be at least 6 cubic inches.
- Drone
    - Weight: Max 10 lb.
    - Dimension: Max 2 cubic feet
    - Signal Range: Maximum of 50 ft. indoors/120 ft. outdoors from base station.
    - Cargo Capacity: Minimum 3 lb. individual drone. Minimum 10 lb. multiple drones.
    - Flight Height: Maximum 7ft.
    - Altitude Accuracy within 6 inches
    - Housing for on board computer, microcontroller, gripper, battery, camera, IMU, GPS, motors, ESCs, rangefinder, and PCB
    - Autonomous
    - Lands at base station once structure build is complete
    - Place the blocks of Styrofoam in a particular order at the base station site using object recognition
- Base Station
    - User's laptop
    - Bluetooth application
- PCB
    - Controls the drone's gripper and rangefinder

2.6 House of Quality Analysis

The House of Quality Analysis is shown in Table 1. The House of Quality Analysis shows the things that must be considered when planning the design of SCUAV. Many factors must be taken into consideration in order for this project to be successful.

Table 1: House of Quality

			Engineering Requirements								
			1) Signal Range from Base Station	2) Flight Height Range	3) Cost	4) Altitude Accuracy	5) Dimension	6) Weight	7) Carrying Capacity	8) Block Size	9) Structure Size
			+	+	-	+	-	-	+	+	+
User Requirement	1) Performance	+	↑↑	↑↑	↑↑	↑↑	↓↓	↓↓	↑↑	↑	↑
	2) Durability	+			↑↑			↑↑		↑	↑
	3) Cost	-	↑	↑		↑	↑	↓↓	↑		
	4) Ease of Use	+	↑	↑	↑	↑	↓↓	↓↓	↑↑		
	5) Battery Life	+			↑↑						
Targets for Engineering Requirements			Indoors: <50 ft. Outdoors: <120 ft.	< 7 ft.	< $1000	+/- 6 inches	< 2 cubic feet	Drone Weight: < 10 lb. Base Station Weight: < 5 lb. Total project weight: <15 lb.	Individual drone: > 0.5 lb. Multiple drones: > 1 lb.	At least 12 inches by 12 inches	At least 288 inches by 288 inches

3.0 Research

This section examines three important aspects for the decision making process for this project. These aspects consist of learning from other existing products and projects, using technologies that are relevant to the project, and a discussion of the selected parts choices. The existing products and projects section discusses related ideas similar to the current proposed project and it sheds a light on what can be learned or innovated in this project. The relevant technologies section describes some the technologies that are applicable to possible usage. Lastly, the parts selection describes the reasons for why some the parts for this project are chosen. This section also compares some of the devices in their category.

3.1 Existing Products and Relevant Technologies

There are several products and/or relevant technologies that are suitable to draw inspiration for SCUAV's design and purpose. The project on flight assembled architecture is one of the key inspirations for this project. It follows the same idea of building a structure using a drone. The goal of this project is to innovate that idea and make it better for the construction industry. The interesting aspect of the Construction with Quadrotor Teams is the drones working together as a team to build a structure. This concept is the future innovation for SCUAV's project. Lastly, the Autonomous Tracker Loader would also be a future endeavor for SCUAV because the autonomous construction robotic team would increase meaning that in the near future it is possible for robots to be building structures completely without human assistance.

3.1.1 Flight Assembled Architecture

There exist a few different implementations for constructing structures with drones. This particular source uses drones to build a structure and each drone collaborates well with his fellow drone. This project done by a university in Switzerland is one of the inspirations for SCUAV. Common attributes seen involve a dedicated camera off the drone, and a specialized gripper on the drone [1]. A camera is placed viewing the operating area the drones are working within. The camera and additional hardware running computer vision tracks the drone's positions and backend software helps avoid collision. A specialized gripper on the drone, usually meant to pick up a specific type of object, is used to grab the blocks during transportation. The drone references the objects using computer vision to aid in navigation from the raw material site to the construction zone along with other sensor equipment that may be commonly found on drones today such as an inertial measurement unit or IMU. With the IMU allowing more data collection of the drone's environment for stable and safe transportation. Complex algorithms are used to stabilize the drone's movements through changes in the motors voltage. Through the use of machine learning techniques, a reliable means of reaction to environment changes allow the drone to quickly adapt to new changes. This would aid in our project by allowing the drone to avoid obstacles and continue its construction without interruption.

3.1.2 Construction with Quadrotor Teams

At the Vijay Kumar GRASP Lab of the University of Pennsylvania, Q. Lindsey and D. Mellinger designed construction quadrotor teams [2]. Quadrotors were used to lift modular parts and assemble cubic structures. The quadrotors were specifically designed for applications within the construction industry. Because of its similarity to our SQUAV project, seeing how they designed their gripper and how the quadrotors worked in teams was particularly interesting to us. There were several methods within their design which ensured that the quadrotor securely grasped the object as well as the object being securely connected with the rest of the components to build the structure. The gripper that Q. Lindsey and D. Mellinger designed was able to lift objects both vertically and horizontally, making the quadrotor more versatile.They designed the structure parts to be easily clamped onto by having the ends of the modular parts be a bigger dimension than the rest of the part. This allows the modular parts to be lifted up when they are standing up vertically. When the modular parts were laying down horizontally, they could still be easily lifted by the drone's claw since the shape of the gripper clamp was designed for the modular part to easily fit inside. Along with this, the parts had electromagnets embedded inside of them to aid in the placement of the part since once the part is close enough, it will attach to the surrounding parts containing magnets. Not only is the structure able to snap into place through magnets, but the drone can determine whether parts are successfully placed and retry if they are not. The video of their demonstration gave our team a lot of insight into ways to ensure an accurate lift and placement of materials.

Another interesting aspect of their project was that they had multiple quadrotors working in a team to construct the structure. Doing this gives the advantage of being able to build structure more quickly. Constructing a second drone to work with our first one is a task that our team would enjoy doing if it could be accomplished within the time frame of the class [3].

Figure 3: Autonomous Construction Quadrotor [3]

3.1.3 Build Robotics - Autonomous Tractor Loader

The Autonomous Tractor Loader (ATL) project was designed by the Built Robotics company. The company's autonomous robot is designed to help the construction industry in the building process. The robot's goal is to successfully dig away all the soil necessary for the foundation of a structure to be built. It is programmed to know how deep to dig, where to start the dig, and what location should the soil be placed after it has been dug out. The Lidar rangefinder is one of the sensors used to determine how far away the robot is from an object like the dig site. An additional sensor is placed to prevent any collisions from happening whether it be another vehicle or a human. The robot is also designed to handle the environment on a construction zone site. An interesting aspect of the company's design is the robot's ability to run a full day (24 hours) of work without tiring. This feature will be good for when the progress of a construction site might be running behind schedule or the company might simply want to get a good start on their project.

The Built Robotics company has a mission to make this robot available on the market to everyone at a reasonable price and they are racing to have this product on the market before self-driving cars become popular. The company wants to help reduce some of the hazardous situations that happens on construction sites and replace most of the human workers eventually with their system. There are mixed feelings about how machines will take over jobs, but the company claims that the integration of machines will open new jobs in the future.

Some things to notice about this project is the robot's ability to be autonomous, how it uses its Lidar sensors, and how it focuses on the safety of the environment. These three aspects are related to SCUAV's purpose. SCUAV's mission, as previously stated before, is to build a structure autonomously. SCUAV wants to help make the construction site safer for workers and others who happen to be traveling near the construction site. Sensors will be used to tell how far something is from the drone, what Styrofoam block to pick, and where to place that same block.

Figure 4: Autonomous Tractor Loader from Built Robotics [4]

3.2 Parts Selection

When selecting components careful consideration on each of the parts selected was performed in order to design a workable project. Parts were chosen based on their specifications, features, dimensions, price, and if they are compatible for the drone's design. Below describes the well thought out though process of each part.

3.2.1 Power Options

Below the power options are discussed.

Tethered Wire Power

Drones that use a tethered wire power supply can remain in the air indefinitely. This is the greatest advantage of using a tethered wire for power. All other power options have a maximum flight time before needing to be recharged or undergo some sort of maintenance. Another advantage to using a tethered wire is it allows for data to be quickly be transferred between devices in the drone and the computer. However, the obvious disadvantage to this option is that the drone will have a fixed range for which it can fly [5].

Solar Power

Having our drone be solar powered is another option for sending power to devices in our drone. Solar power harvests energy from the sun and converts it to electric energy. Solar cells can store energy for use when not in the presence of the sun, so the drone will still be able to fly indoors and at night. However, having to harvest solar energy before flying the drone in these environments would make the drone less practical for use in construction when the drone may need to operate at night, inside of structures, or in weather providing little sunlight. There are, however, many advantages to using solar power in our drone design. Solar power does not have fuel that needs to be replaced or batteries to recharged. Because of this, there is less regular maintenance needed for drones powered by solar energy which can continuously operate as long as sunlight is present. Solar powered drones are also more environmentally friendly than the battery power option.

Fuel Cell

Fuel cells are another method of providing power to drones. Fuel cells electrical energy for systems by combining a fuel with oxygen. This fuel could be hydrogen. The reaction that takes place to provide energy from fuel cells environmentally friendly since its only byproduct is distilled water. Fuel cells also have high energy density than the other power options. They use oxygen available in the air for half of the reaction for producing energy. Because of this, more energy can be stored for a given amount of space in fuel cells than in batteries. The energy density of fuel cells are about three times greater than batteries used in drones. Along with this, fuel cells can be recharged at a much faster rate than batteries can. For instance, hydrogen gas fuel cells can be re-filled in as little time as two minutes. Another advantage of using fuel cells over batteries is that they allow drones to fly for a longer time and fly a greater distance than drones powered with batteries. Fuel cells also differ from batteries in the way they produce power. Batteries store energy and can provide it when it is needed in the drone. However, an advantage of fuel cells is that it can produces the energy as it is required in the system which removes the need to store energy. However, some disadvantages of using fuel cells is that it is both much heavier and largely more expensive than the other power options. It would be very difficult to implement a fuel cell into out drone design since creating the reaction for energy production in fuel cells is very difficult. This is because, in the case of using hydrogen as the fuel, hydrogen and oxygen are not found in their pure state in nature and they need to be obtained from an alternate source, such as air and water. However, the processes for obtaining these elements from other compounds and sources is both complex and expensive.

LiPo Battery Power

Lithium Polymer batteries, also known as LiPo batteries, are the most common type of battery used in drones. There are many advantages to using Lipo batteries, which explain their widespread popularity. When comparing LiPo to NiMh and NiCd batteries, LiPo batteries are lighter weight. Compared to other batteries, LiPo batteries do not discharge very rapidly. LiPo batteries are also capable of disch. LiPo batteries have a battery life of about 20 minutes. The chemical composition of LiPo batteries allow for it to have a higher energy density than most other batteries of a comparable weight.

NiMH and NiCd Batteries

NiMH and NiCd are older battery types than lithium batteries. They are also not as commonly used as lithium batteries because they are heavier and have a lower energy density than lithium batteries. Nominal voltage is usually the midpoint of an operating range.

Power Option Comparisons

Our team decided to use LiPo batteries for our drone design. We decided against using a tethered wire because of the range limitations. Despite the many advantages of a solar powered drone, we decided that it would greatly increase the complexity of the project, and we wanted to focus on the computer vision aspects of the project. Along with this, we decided the fuel cell power option would not be practical for our project design because of the its weight, increased cost, and increased complexity. And when comparing the different types of batteries, we realized that LiPo batteries offered the best capabilities for our drone.

LiPo Battery Ratings

There are many different types of LiPo batteries. And it is essential for our project's success to select an appropriate battery. For this reason, we have included the specifications for LiPo batteries within our paper. LiPo batteries are composed of individual rectangular cells. Each of these cells has a combination of metal and chemicals and they create an electrical charge when stored with other cells. Each LiPo battery cell has a nominal voltage of about 3.6 volts. The amount and orientation of LiPo battery cells will determine its voltage and capacity. The cells can be connected in series and parallel. Cells connected in parallel will increase the capacity, while cells connected in series increase the voltage. A battery with two cells in series will provide 7.2 volts and having three cells in series will provide 11.1 volts. The number and orientation of battery cells are important and the information is specified on the package. For instance, for three cells in series and two cells in parallel, the notation would be "3S2P".

Table 2: Power Comparisons for Drone

Power Comparisons					
	Tethered Wire	**Solar Power**	**Fuel Cell**	**LiPo Battery**	**NiMH and NiCd Battery**
Advantages	-Can run continuously	-Does not have a limited flight range -Fuel does not need to be replaced or recharged -Environmentally friendly	-Does not have a limited flight range -Long flight time -powerful	-Does not have a limited flight range -Lightweight -Practical implementation -Widespread use	-Does not have a limited flight range -Practical implementation
Disadvantages	-Has a limited flight range	-Needs to harvest solar energy to fly	-Complex -Expensive	-Has a limited flight time	-Heavier and lower energy density than LiPo batteries -Has a limited flight time

Another important specification is the battery capacity. The capacity of batteries used in drones play a vital role their flight time. This value is given in amp hours and it determines the length of time a battery can provide energy. However, the larger the capacity of a battery, the heavier the battery is. The weight of the drone is also a vital factor for drone flight time. Because of this capacity and weight tradeoff, a greater battery capacity does not directly increase the drone flight time because of the added weight it also adds.

A battery's discharge rate, or C rating, is another important battery consideration. It determines how quickly energy can be drawn from the battery. It is essential that we select an appropriate discharge rate for our drone. For instance, if the motors draw more power from a battery than the battery can provide, the battery could become damaged. If a situation occurs where the drone's battery supply is damaged, the drone could end up crashing. Some applications have short periods where a greater discharge is necessary. Sometimes LiPo batteries specify a burst discharge rate, or burst C, which indicates the discharge rate that can be provided by a battery given that it last for a short period, usually 15 to 30 seconds. When our drone is picking up and releasing objects, power will be provided to the claw motor. When the claw motor is being used, more current will be drawn for a short period of time. Because of this, the battery power supply for the claw motor should have an appropriate burst discharge rate specified.

Each LiPo battery cell has a nominal voltage of about 3.6 volts. And the voltage of each cell should remain between 3.0 volts and 4.2 volts. Voltage on each cell of the battery that is outside this range could cause the LiPo battery chemicals to be unstable, which could potentially start a fire. The voltage supplied to LiPo batteries are very important and improper use of LiPo batteries can lead to safety hazards. There are different methods to being indicated when a battery is getting low. For instance, some flight controllers can monitor the battery's voltage and provide an indication.

LiPo Battery Safety

It is important to know how to safely handle LiPo batteries to reduce the risk of injury and damaged equipment. Various safety measures for LiPo batteries should be observed when charging, storing, and handling them.

Proper charging of LiPo batteries is essential for safety and for the battery's longevity. It is important to charge LiPo batteries in a fireproof location or in a LiPo safe bag, which is fire resistive. The location at which a battery is charged is another important consideration. Before charging a LiPo battery, it should be observed for any signs of damage. When charging a LiPo battery, it is important to check that the battery shows no physical signs of being damaged. There are various signs of damage that indicate a battery is no longer usable. This includes overheating, swelling, or leaking and the emission of unusual smells, noise, or smoke. After checking the battery for damage before charging it, one should still be attentive to the battery for signs of damage while it is being charged. The battery charger itself should also be tested before being used to ensure it is properly functioning before re-charging LiPo batteries with it.

We also found that damage incurred on LiPo batteries can be avoided through proper handling. LiPo batteries will be hot after providing power to a device or system. To prevent damage to LiPo batteries it is also important to wait until the battery is cooled before coming into contact with it. The flight times of safety precautions should also be observed when storing LiPo batteries. Each voltage cell should be charged close to 3.8 volts, or 40 to 50% charge and should not be left fully charged. This is not as important when the battery is being stored for a short period of time, about a day, before being used again and is more of a precaution for long term storage. It is important to note that when batteries are obtained, the charge will not be 100% but will be around 40 to 75%. Having the proper voltage in each of the voltage cells is essential for LiPo batteries as the cell should not drop below 3.0 volts or exceed 4.2 volts. An important feature present in all LiPo battery chargers is its ability to balance charge. The voltage levels of each of the cells are monitored and set to all be the same. This equalization of cell voltage is very important since the battery cells will need to be recharged around the same time after being equalized. Along with this, LiPo batteries should also be stored at room temperature. The batteries should also be stored in a fireproof location or Lipo Safe bag.

Fuses
Our team to use a fuse as an extra precaution when testing our equipment. A fuse is a device that is used to protect electronic equipment in a circuit. It is placed in series in a circuit and is made of a low resistance conductive material. Every fuse has a maximum voltage rating and a max amount of current that can flow through the conductive material before it begins to melt. As the material melts, a space forms. This causes electrical arcing to occur. This means that current will arc across the spacing. However, as the current increases, the space will eventually become too wide and the current flow is terminated. Since the fuse is placed in series with the circuit, the circuit will become open as the fuse disconnects. In this way, fuses are safety devices that protect electronic equipment from damaging with overcurrent. We decided that this could save us from having to buy extra parts from failed equipment. It also protects electrical equipment from over temperature damage and reduces the risk of fires. For these reasons, our team decided to obtain fuses for our testing procedure despite the added cost.

There are AC and DC fuses. DC fuses are usually larger than AC fuses. DC fuse is always at a constant positive voltage which allows for an electrical arc between the melted wires. This electrical arc can be avoided by increasing the distance between the electrodes. Another method used to help reduce the arc, which is seen in some DC fuses, is having a spring to aid in pulling the melted electrodes apart. While the voltage of AC fuses oscillates between positive and negative voltage and do not have a constant positive voltage like DC fuses. AC fuses switch between positive and negative about 50 to 60 times a second. Because of this, a AC fuse’s voltage will cross zero point. Having points of no current helps break the electrical arc that form inside of fuses because as soon as the current reaches zero, the arc will self-terminate. Since the electric arc is more easily terminated in an AC fuse, the electrodes of AC fuses can be closer together than electrodes of AC fuses. This allows for AC fuses to be smaller than DC fuses. Size is one of the main differences between AC and DC fuses.

Another difference between AC and DC fuses is their voltage rating. This is due to the electric arc between the melted electrodes being easier to stop in AC versus DC fuses. Because of this, AC fuses are usually able handle higher voltages. There are also many different types of AC fuses. AC fuses are classified into high voltage and low voltage fuses. For applications using power above 1 kilo-volt, high power fuses are used. And for power below 1 kilo-volt, low power fuses are generally used. Within high and low voltage fuses there are also various types. One type of AC fuse which our team found particularly useful is the rewireable fuse. Once the fuse has been blown, the fuse wire can be easily replaced and reused.

Despite the advantage of the different types of AC fuses and the smaller size, our team decided to use the DC fuse. The power sent to the various devices within our project will all be DC. It is likely that our power source for the project will be a battery supplying DC power to our devices. Because of this, we will need to use a DC fuse. DC voltage does not alternate, so the fuse needs to electrodes need to be placed far apart. If they are not, in the case of an AC fuse, when the fuse opens the gap will be too small. This can cause two very dangerous situations. The arc can continuously melt everything around it and catch fire, or the current will arc and be hot enough to melt the fuse back together and continue to heat up until catching fire.

Power Regulation

The power for the drone's motors will be handled by a LiPo battery. The Raspberry Pi and the printed circuit board need a power supply. There are various ways this power can be provided. If the power regulation is handled on the printed circuit board, either a linear voltage regulator, a switching regulator, or Zener diodes could be implemented in the printed circuit board design. Another is to use a LiPo battery expansion board to power the Raspberry Pi and a portable charger to power the PCB.

In this section we will compare the different power regulation options. As mentioned previously, linear voltage regulators, switching regulators, and Zener diodes can all be used as voltage regulators. For any input voltage, a voltage regulator can provide a constant output voltage. These integrated circuits are useful for providing devices with constant voltages from a power supply source. The primary role of voltage regulators is to either step down or step up voltage to meet the voltage requirements of different devices. However, they also act as a buffer and protect components from being damaged. And they are placed between an input voltage and a load to shield devices from voltage spikes from the power supply. Depending on their application, some voltage regulators need to be designed to have improved performances such as lower noise or lower output ripple voltage.

Linear Voltage Regulator

Linear voltage regulators are simple 3-pins voltage regulators. A constant output voltage is able to be maintained through a negative feedback control loop. One disadvantage to linear voltage regulators is that it has a large minimum voltage drop across it. This large difference in potential voltage is not desired because it causes the power dissipation to increase. As higher voltage levels are sent to the voltage regulator, more power is dissipated through the voltage regulator. For instance, if 7 volts are sent the voltage regulator, 2 watts could be dissipated. But when 10 volts are sent to the same voltage regulator, 5 watts would dissipate in the voltage regulator. Linear voltage regulators usually have low efficiency, typically below 50%. Because of the high-power dissipation and low efficiency of linear voltage regulators, they are commonly used in applications which are low cost and low power. At low power levels, the efficiency of linear voltage regulators is better since there is less of a potential created in the regulator. Another disadvantage to linear regulators is that they take up a lot of space. However, an advantage of linear voltage regulators is that they have low output ripple voltage. They also respond very quickly to line changes and they have lower electromagnetic interference and noise than switching regulators. Linear voltage regulators can either be series of shunt.

Switching Regulators
Switching regulators are voltage regulators which can handle high input voltages. Because of this, they can be used for a wider range of applications. They have much better efficiencies than linear voltage regulators, their power efficiency is usually around 85% and can be achieved in the 90% range. However, a disadvantage to switching voltage regulators is that extra components are needed. These extra components. Along with this, switching regulators generate electric noise which can affect the components the voltage regulator is connected to. As the frequency of switching regulators increase, so does the noise. Switching regulators can be step up (such as boost), step down (such as buck), and inverter voltage regulators (such as buck-boost). Good efficiency is essential for power conversions, so switching regulators are used for these applications.

Zener Diodes
A Zener diode is a device that will provide voltage regulation for components by shunting to ground any voltage above its Zener diode breakdown voltage. A disadvantage to Zener diodes is that they can only be applied to low power applications. However, sometimes a high input power can be sufficiently reduced to a lower voltage level through different devices such as a large enough resistor. Although Zener diodes can act as voltage regulators within a circuit, they do not perform as well as linear regulators do. The reason for the performance differences of Zener diodes acting as linear regulators and linear regulators is that Zener diodes to do not have a feedback loop for the changes in voltage from the power supply and the temperature changes that may occur. However, voltage regulators are integrated circuits that have this essential feedback loop designed within them.

Power Regulation Decision
After considering the various methods of regulating voltage, our group was between using a switching regulator for providing a regulated voltage and using a Lithium battery expansion board. Switching regulators are highly efficient and would provide a reliable source of power for the Raspberry Pi and the PCB. Performing the power regulation through a switching regulator on the PCB would also be a lighter weight option compared to having a portable charger to provide power to the PCB. We decided to use a LiPo expansion board to power the Raspberry Pi and to use a switching regulator to provide power to the PCB. However, in case of any issues with providing power from a switching regulator on the PCB, we will be able to easily be able to use a portable charger as an alternative power supply.

3.2.2 Gripper: Robot Claw
Our team decided to investigate the different types of grippers available for our drone. We compared the advantages of using a robot claw to the advantages of using a suction cup as method for lifting objects.

Suction Cup Gripper
A common way that drones lift objects is through a suction cup gripper. These grippers grasp object by creating a vacuum. Some disadvantages to using a suction cup is that the airlines can get clogged if surfaces are dirty. Flat surfaces are most suitable for suction cups. And surfaces which not flat, like curved or pointed surfaces, may not have enough surface area for the suction cup to lift the object. Another disadvantage of the suction cup method is that it is unable to lift porous objects [6].

Table 3: Power Regulation Analysis

	Linear Voltage Regulator	Switching Regulator	Zener Diode
Applications	-Used in low cost and low power applications	-Can handle high input voltages -Can be used for power conversion	-Used in low power applications -Do not perform as well as linear voltage regulators

Magnetic Gripper
Other options we explored were electromagnet and permanent magnet gripper to lift objects. These are both examples of magnetic grippers which are used to lift ferromagnetic materials.

An advantage of magnetic grippers is that the grasping of the object can be done very quickly. Along with this, magnetic grippers can easily be controlled. However, if there is just a bit of oil on the surface of an object, the grasp can be greatly reduced. Along with this, the objects lifted by magnetic grippers may remain magnetized after being released. The greatest disadvantage of magnetic grippers is that it is only able to lift ferromagnetic materials [7]. If a material is ferromagnetic, it means that the material can be magnetized. Ferromagnetic materials are usually metal, for example iron and nickel are ferromagnetic. Along with this, if the gripper is moving too fast, objects being lifted by magnetic grippers can sometimes slip out of the grasp of the gripper.

The EPM, or Electro Permanent Magnet Gripper, is a magnetic gripper that we noticed was being used in different drone designs. It is a device that has advantages from both electro and permanent magnets. After conducting research, we realized why this gripper was popular for drones. It can easily interface with a Pixhawk microcontroller. It is easily interfaced within a drone as it only needs to receive a PWM servo input. Along with this, there are many online resources of how to connect this gripper to a microcontroller because of its common use for drones. Another advantage to this gripper is its low power draw. During steady state, the EPM draws less than 1mW of power. It is also capable of lifting heavy loads. Its carrying capacity is one kilogram, about 2.2 pounds.

Mechanical Gripper

Mechanical grippers often used in drones to grasp objects. And there are many different types of mechanical grippers available in the market.

An advantage of both suction and magnetic grippers, is that they can pick up objects of different sizes as long as they as flat. However, mechanical grippers may not be able to do this. If a claw design is used for a mechanical gripper, it will have a maximum dimension for the objects it is able to lift. Mechanical gripper designs using in drones are typically claws that wrap around objects. Because of this, the size of the claw and the amount the claw is able to extend when in the opened position will determine the size of the object the claw is able to lift. Our project is meant to lift construction materials. Since construction materials are not always the same size, using a gripper that does not have a fixed dimension it is able to lift would be an advantage.

To address the issue of having a fixed dimension able to be lifted by the gripper, our team explored different gripper designs. The first design explored by our team was a traditional craw gripper and the second design explored was a bar clamp gripper. The bar gripper would use a bar wrench to pick up the cube from its sides. We decided on a traditional claw instead because of the extra weight a bar wrench would add.

Gripper Comparison

There are many advantages of using the Electro Permanent Magnet Gripper, or EPM. It has low power draw, can lift heavy objects, is able to lift objects of various sizes, is commonly used in drones, and is easily controlled. Despite the many advantages of magnetic grippers, we will not be able to implement this gripper in our project. Magnetic grippers are only capable of lifting ferromagnetic materials and the objects used in our project will not be metal. For this reason, we will not be able to use a magnetic gripper in our design.

Our team also decided against using suction cup grippers in our design. There are various conditions that need to be met for suction cup gripper to correctly grasp object. Objects must be nonporous and flat. Along with this, the surface of the object cannot be dirty, or else the airline could become clogged. Since our project is designed to be used at the construction site, the materials our drone will be lifting will likely have dust and dirt on their surfaces. Mechanical grippers are usually unable to grasp object past a certain dimension. However, the disadvantage of will not be an issue for our team since we will can design mechanical grippers that do not have this limitation. The gripper we ultimately decided on is a mechanical gripper. The gripper comparisons is shown in Table 4.

3.2.3 Microcontroller

The microcontroller unit is an essential component for SCUAV's functionality. It is the nervous system of all the drone's capabilities. This type microcontroller will act as the drone's flight controller, which receives all of the sensory information from the Raspberry Pi, the IMU, and the GPS. Based on the information the flight controller receives, the controller sends signals to the propellers' motors, and to the electronic speed controllers. The flight controller will also be able to adjust the speed, direction and angular position of the drone. Figure 29, in Section 5, shows a pictorial view of how the microcontroller was first thought out to work within SCUAV's system. During the drone's mission, the flight controller must send all the data that is going in and out of it to the base station so that the user can monitor what is happening on board of the drone and if any abrupt changes need to be made.

There are several possible options to choose from when considering which flight controller is the best option to use for this project. Option one is the Pixhawk, the second option is the Arduino Uno, and the other option would be the BeagleBone Black board. The comparisons of these products are discussed below.

3.2.3.1 Pixhawk

The Pixhawk flight controller brand has several types of flight controllers ready to purchase for any project. This section discusses two in particular and they are the 3DR Pixhawk Mini and Pixhawk Autopilot. Table 5 gives a preview to the features and specifications of both microcontrollers.

3DR Pixhawk Mini

The 3D Robotics company is the designers of the 3DR Pixhawk Mini. This miniature Pixhawk flight controller comes from the family of the Pixhawk generation. What makes it unique is the capability of performing all the same tasks the original Pixhawk can do despite its size. All the new improvements also gives this flight controller an edge over the original. Some of these improvements are the primary and secondary IMU MPU9250 and ICM20608 sensors are more reliable for flight, a microJST rather than the DF13 connectors, supports 4S batteries, its size is one-third the original, and the GPS module is a separate device that can be easily plugged into the flight controller and it has been recently updated for better flight performance. All of the information mentioned can be found on 3D Robotics website which can be found in Appendix C of this document. Figure 4 below gives a preview of all of the components on the 3DR Pixhawk Mini.

Pixhawk Autopilot

The creators of the Pixhawk Autopilot designed this flight controller with hobbyists, amateurs and researches in mind for their robotic adventures. The 3D Robotics company also contributed to the design of this Pixhawk. Its size is suitable for most drone projects and there is an additional co-processor which serves as a failsafe option. Instead of just having a general IMU sensor, the sensors of the Pixhawk are broken into gyroscopes, accelerometers and magnetometer. With each of these components working together, they can compete with the IMU in the 3DR Pixhawk Mini. The maximum operational ratings for the power module input, the servo rail input, and the USB power input is 4.8V to 5.4V respectively. There is also a GPS module that connects to this device. Appendix C houses all the information relevant to this section and gives credit to the Pixhawk company.

Table 4: Gripper Comparisons

	Suction Cup	Magnetic Gripper	Mechanical Gripper (Bar Wrench Design)	Mechanical Gripper (Traditional Design)
Advantages	-Commonly used on drones	-Quickly pick up objects -Easily controlled	-Can pick up objects of varying sizes	-Commonly used -Light weight
Disadvantages	-Surface must be flat -Cannot create contact with porous materials -Items can slip out of suction cup grip	-Can only lift ferromagnetic materials	-Not traditional, may be harder to implement -Heavier than traditional gripper	-Objects must have a handle for the claw to grip to

Pixhawk Comparisons

The 3DR Pixhawk Mini could be a possible top contender for the final microcontroller selection. Its size and how it can be implemented makes it standout when compared to the Pixhawk Autopilot. Majority of all its features and specifications are the same. The 3D Robotics website claims that their mini Pixhawk offers 60% more in a smaller package when compared to the original Pixhawk. It is based on the PX4 hardware system and it has many more upgrades when compared to the Pixhawk. One interesting aspect of the 3DR Pixhawk Mini is the fact that their GPS module is not part of its internal design. It is a separate device that connects externally to the Pixhawk Mini. The module has its own features and dimensions and it may even be more reliable than the original based on the new design upgrades.

The Pixhawk Autopilot is bigger than the 3DR Pixhawk Mini but most of its design specifications are similar to the Pixhawk Mini. One thing that sets it apart from the other device is the possibility of 6S. LIPO batteries being implemented although the standard is 4S LIPO batteries. The Pixhawk website provides a guide on how implement this. Recently discovered was the barometer located within the Pixhawk and it served an as essential part for this project's programming. m

3.2.3.2 Arduino

The Arduino is home to some of the most incredible projects developed out there. The company's product is built into simple projects like RC toys to complex projects like SCUAV. All the information mentioned below corresponds to Appendices section of this document.

Arduino Uno R3

According to Arduino's website, the Arduino Uno R3 board is the most commonly used board in the entire Arduino family. All the resources for this board are easy to access via the website and the implementation process is also a very smooth one. The board's features and specifications can be seen in Table 5. There are two ways to power this device: USB connection or external power supply. This feature would be useful for when the all of the drone parts are put together and there might a need to use one power source, USB port, to connect to another component in the design.

Arduino Mini 05

The Arduino Mini 05 is being considered in this project because its size alone could be very valuable in the design aspect of the project. There has been multiple revisions of this particular board but the Mini 05 is the newest model. The new revisions consist of having all the board's components on top of the board, a reset button, and there is a package for the ATmega328 despite it being originally being ATmega168. Just like the Arduino Uno R3, Arduino provides all the documentation and code implementations needed to use this product.

Arduino Comparison

When comparing both Arduinos to each other the Arduino Uno R3 seems to be more compatible with the design for SCUAV. Both Arduinos have the similar specifications. One ideal feature of the Arduino Mini is the size of the board. Despite the Arduino Uno's board being bigger, it will be ideal for this project because the components on the board meet our design requirements. It would also be easier to add more specifications onto this board since an IMU is needed for this project. Another similarity would be easy access to all specification documents and code examples which are located on the company's website [8].

3.2.3.3 BeagleBone Black

The BeagleBone device is also another possible contender since it has IMU capabilities. This notable feature is important for the drone to fly. Another interesting aspect of this device is there is an easy connection access to the Raspberry Pi and the Base Station for this project. The device has two PRU 32-bit microcontrollers [9]. More information on this board is mentioned in Section 3.2.10.

3.2.3.4 Microcontroller Decision

Since all the possible microcontrollers have been discussed previous, now is the time to narrow the decision between the top contenders and make a final decision. From the Pixhawk section, the original Pixhawk seems to be a good choice since it is easier to implement in our drone design. For the Arduino section, the Arduino Uno R3 is a better alternative because its specifications meet the requirements for SCUAV's design. When comparing the Arduino Uno to the Pixhawk, the Arduino Uno is on the cheaper side of the planned budget but the Pixhawk will have less components to implement into our design. The Arduino also does not have any IMU components on the board. When adding the BeagleBone Black to the comparison list, it would seem that the device may have a higher leg up when comparing the communication level within the drone but this does not necessarily mean that this is a good thing. The BeagleBone board would allow for direct communication to the Raspberry Pi connected to the camera as well as the base station. One thing to note about both the Arduino and the BeagleBone is they both lack the safety switch feature and automatic failover. The final decision will be to use the Pixhawk Autopilot flight controller. The reasoning behind this is the fact that the design implementation will be cleaner on the drone frame, the drone will have a better chance of having stable flight especially when the grippers build the Styrofoam structure, and the overall implementation will be compatible with the parts chosen for SCUAV's design. Table 5 gives a comparison of the different microcontrollers.

3.2.4 IMU

The acronym IMU represents the term Inertial Measurement Unit. The IMU in drone terminology comprises of a gyroscope, an accelerometer, and a magnetometer. This device plays an essential role in the drone's functionality. The main job of the IMU is to keep the drone stable during flight. It communicates with the microcontroller by providing the amount of speed needed to keep the drone in the air, signaling the rotational forces that are around the drone, and report any changes that need to be made in order for the drone to stay in the air successfully. With all the information the microcontroller received, it will then tell the rest of the drone's components to behave correspondingly to the new instructions the microcontroller was given. The flight controller chosen for this project does not have a single unit or slot dedicated to the IMU, instead it has the three sensor components embedded separately into the flight controller's hardware design.

The flight controller chosen for this project does not have a single unit or slot dedicated to the IMU, instead it has the three sensor components embedded separately into the flight controller's hardware design. This is very convenient for this component of the project and for the overall cost budgeted for SCUAV's design.

Gyroscope

The gyroscope sensor is responsible of detecting rotational motion and any changes in an object's orientation. It is used to measure angular velocity. Angular velocity is defined as a measurement of speed of rotation [11]. This sensor is one of the essential parts for flight stability since it will be able to tell if the drone is unbalanced during the drone's flight. If the drone is unstable an alert will be sent to the flight controller and the drone will try to readjust itself. It is important to note that acceleration of an aircraft does not affect what the gyroscope measures [11]. There are three main applications for gyroscopes: angular velocity sensing, angle sensing, and control mechanisms [12]. Angular velocity sensing pertains to the amount of angular velocity produced. Angle sensing deals with the angular velocity produced by the gyroscope's own movement. Lastly, control mechanisms aims to detect vibrations produced by external factors, like wind, and transmits these vibrations back to the microcontroller. All three of these applications work together to produce the proper functionality of the gyroscope sensor.

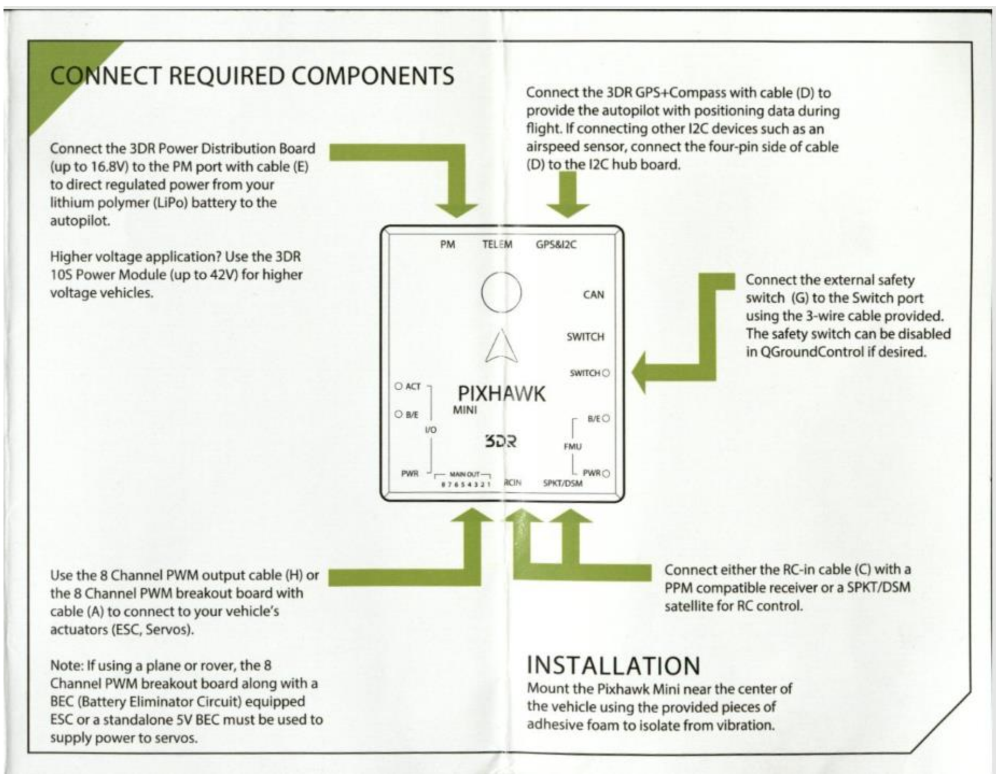


Figure 5: 3DR Pixhawk Mini Anatomy [10]

Accelerometer

The accelerometer sensor measures the acceleration of gravity on an object. It uses the same component that gyroscope uses, which is MEMS (Micro Electro-Mechanical Systems). MEMS is similar silicon integrated circuits but they are mostly mechanical [13]. This sensor is capable of measuring acceleration in 3D which is useful for measuring the orientation of the drone relative to the earth's surface [13]. This 3D feature would collaborate well with the gyroscope's implementation on the flight controller. For more insight on how the accelerometer measures acceleration specifically, it uses the relation between its internal capacitance plates [14].

Table 5: Microcontroller Comparison [10]

	3DR Pixhawk Mini	Pixhawk Autopilot	Arduino Uno R3	Arduino Mini 05	Beagle Bone Black
Specifications					
Processor	32-bit ARM Cortex M4 with NuttX RTOS	32-bit STM32F427 Cortex M4 core with FPU	ATmega328P	ATmega328	AM335x 1 GHz ARM Cortex-A8
PWM/servo outputs	8	14	14	14	?
Interfaces	UART, I2C, CAN	UART, I2C, CAN	SRAM, EEPROM, Flash Memory	SRAM, EEPROM, Flash Memory	SDRAM, Flash Memory, UART, I2C
Connectors	Micro JST connectors	DF-13 connectors	I/O pins	I/O pins	I/O pins
Voltage	4.1 - 5.5V	4.8V - 5.4V	5 V	5 V	5 V
Weight	15.8 grams	38 grams	25 grams	Less than 2 grams	39.68 grams
Size (W ×T ×L)	38 mm × 12 mm × 43 mm	50 mm × 15.5 mm × 81.5 mm	68.6 mm × 53.4 mm	30 mm × 18 mm	53.34 mm × 4.76 mm × 86.36 mm

Magnetometer

The magnetometer sensor is mostly used for assisting in calibrating the drone against orientation drift [13]. This additional sensor makes up the basic definition of what an IMU does for a drone when it is connected to a flight controller. This sensor basically measures the magnetic fields surrounding the drone's environment [`15]. In the world of Physics, it is good to have a device like this to make sure that the drone runs smoothly.

3.2.5 GPS Module

The GPS module is a crucial part of the drone's functionality. The GPS application will allow for the microcontroller to know the position of the drone at all times. This information is essential because it allows for the drone to navigate from its starting position to the materials site, then to the building site and lastly back to the base station. Getting the navigational information correct will be able to meet the objective of this project. The Pixhawk flight controller has a slot available to plug in the GPS module. Possible GPS modules to consider are the uBlox NEO-M8N GPS GNSS + Compass Receiver, 3DR uBlox LEA-6H High-Performance Receiver, and Trimble MB-One GNSS Receiver. The comparison between these suggestions are located in Table 6. The uBlox NEO-M8N GPS GNSS + Compass Receiver will be chosen for this project simply because it suits the purpose of this project and there is a good cost bundle that includes this device with the flight controller chosen. Although the 3DR uBlox LEA-6H High-Performance Receiver is also a good choice, it was not picked because of the cost and the navigational update rate is half the NEO-M8N GPS. The Trimble MB-One GNSS Receiver is an okay GPS module system but the other two systems seem to fix the drone's design better.

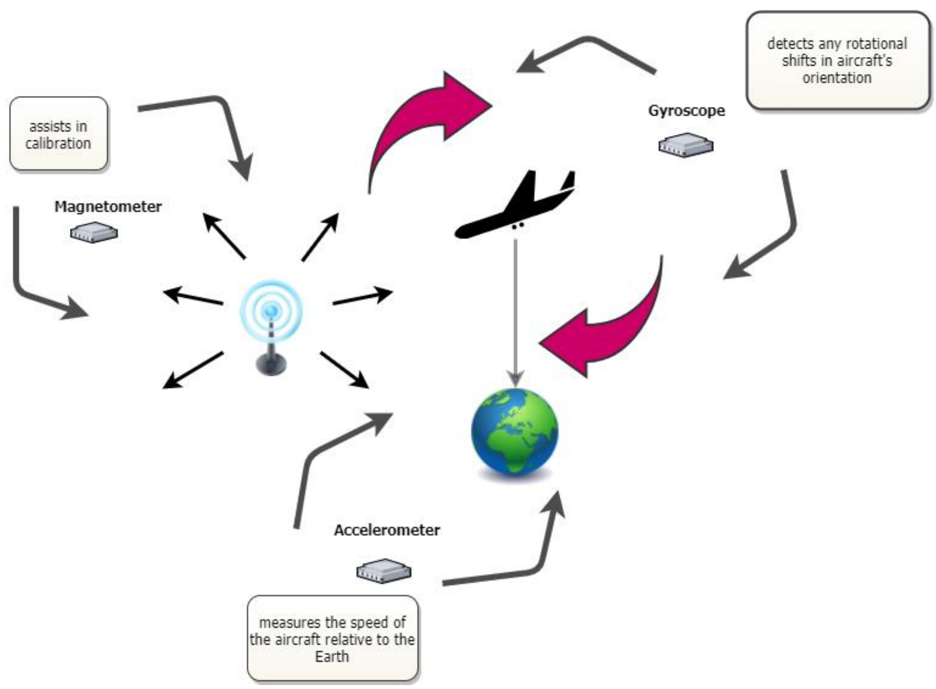


Figure 6: IMU Sensors Working Together

3.2.6 Rangefinder: Infrared and Ultrasonic Sensors

The rangefinder component's functionality is to tell how far away an object is from the drone during its flight. This will be useful for when the drone is trying pick up a Styrofoam block with the grippers and place the same block in a specific location. The rangefinder will also be helpful for telling the drone how high the constructed structure currently is. Another advantage of having a rangefinder is to allow the drone to avoid obstacles that could especially cause harm to the drone and the objects themselves. All of this information will be signaled to the microcontroller and the drone will behave correspondingly to the given data. The sections below describe possible rangefinders and Table 7 compares them.

Lidar-Lite v3 Rangefinder

The Lidar-Lite v3 rangefinder would work perfectly for keeping SCUAV above ground at a particular height, avoid unnecessary objects, and telling the drone's grippers how far away the block is to pick and how to place it in the structure. This device has an easy installation process when connecting it to the Pixhawk. It can use either the I2C port or one of the PWM slots. A demonstration of this can be found in Figure 6. This device also collaborates will with the IMU component.

On technical terms, the Lidar-Lite rangefinder measures the distance of an object by calculating the time delay between the transmission of a laser signal and its reception after reflecting off the same object [20]. Using the speed of light, the laser translate it into distance [20]. This distance is then transmitted back to the microcontroller and the drone acts correspondingly if an object seems to be close in its flight range.

Table 6: GPS Module Considerations [16,17,18]

Specifications	uBlox NEO-M8N GPS	3DR uBlox LEA-6H	Trimble MB-One GNSS
Interfaces	UART, USB, SPI, DDC	UART, USB, SPI, DDC	UART, USB, CAN, SAMTEC
Voltage Supply	2.7 V - 3.6 V	2.7 V - 3.6 V	3.2 V - 4.5 V
GNSS Types	GPS, GALILEO, GLONASS, SBAS	GPS, GALILEO, GLONASS, SBAS	GPS, GLONASS, SBAS
Receiver Type	72 channels uBlox M8 engine	50 channels	240 channels
Navigation Update Rate	10 Hz	5 Hz	50 Hz
Features	Flash, SAW, LNA, TCXO	Flash, LNA, SAW, TCXO	Flash, LNA, SAW
Low Noise Regulator	3.3 V	3.3 V	None
Antenna Supply & Supervisor	None	Yes	Yes
Size	37 x 37 x 12 mm	38 x 38 x 8.5 mm	71 x 46 x 11 mm
Weight	22.4 g	16.8 g	24 g

Note: Highlighted in green is the chosen component for this project.

LeddarTech LeddarOne Optical Rangefinder (3.3 V UART)
The LedderTech LeddarOne Optical Rangefinder is another possible candidate in choosing which rangefinder to use. This rangefinder is an alternative low-cost and

compact device that detects and senses the environment of a targeted object [21]. According to the Robot shop website, this sensor is suitable for level sensing and proximity detection [22]. There are several similarities between this rangefinder and the one mentioned above.

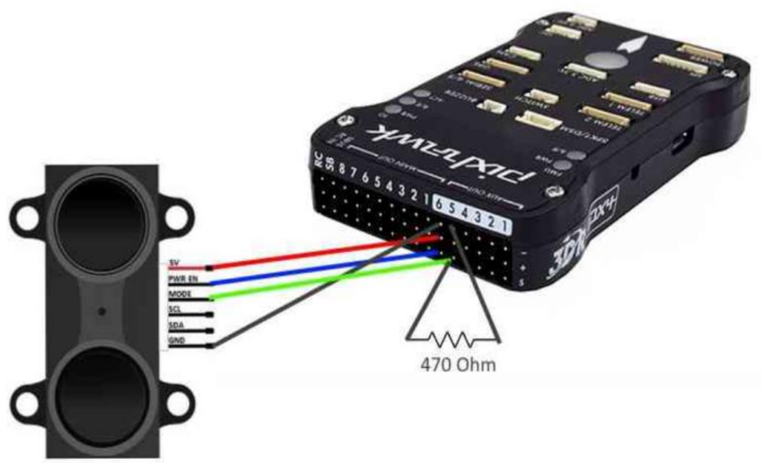


Figure 7: Rangefinder Installation into the Pixhawk Using the PWM Interface [19]

Ultrasonic Rangefinder

The rangefinder we decided to use in our project is the HC-SR04. We decided on this rangefinder because it is much cheaper and simpler than the other rangefinders. Our team's initial plan was to have the Lidar-Lite v3 or the LeddarOne communicate with the flight controller. However, the HC-SR04 ultrasonic rangefinder is commonly used with the ATmega328 microcontroller. Because of this, the HC-SR04 was implemented with our PCB design. In the next paragraphs it will be described how HC-SR04 works and how it can be controlled from the ATmega328.

This device emits ultrasound waves at 40 kHz. Ultrasound waves are those above 20 kilohertz and inaudible to humans. The rangefinder consists of a power, ground, echo, and trigger pin. The trigger pin emits ultrasonic sound wave in eight 40 kilohertz bursts. To initiate the bursts, the trigger pin must be held high for 10 microseconds. Once the waves are emitted, the timer begins counting. When the sound wave is interrupted by an object, it will bounce back and be received by the echo pin. The timer is then stopped, and the echo pin will be held high for the same amount of time that the sound wave traveled [76]. The timing for the rangefinder can be seen in Figure 8.

The HC-SR04 rangefinder can be easily control from the ATmega328 microcontroller. The distance an object is away from the rangefinder can be calculated from the total time a sound wave travels. This is the time from when the trigger pin emits the sounds waves to when they are received back by the echo pin. The distance away an object is can be calculated from the following equation:

d=(t*0.034)/2

Where *d* is the distance away an object is away in centimeters and *t* is the total time the sound wave traveled in microseconds. 343.5 meters per second is the speed of sound waves. The speed of sound converted to centimeters per microsecond is 0.034. The distance traveled by the sound wave is divided in half to obtain the distance to the object.

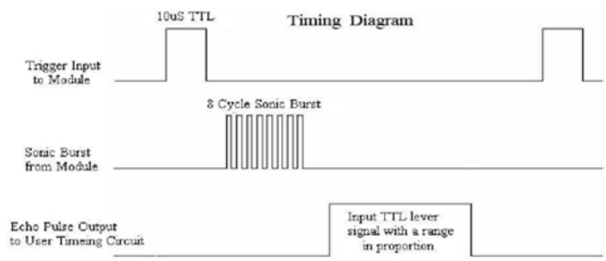


Figure 8: Rangefinder Timing Diagram [77]

Rangefinder Comparison

The table below compares the specifications between the two possible rangefinders. The Rangefinder Conclusion section describes which rangefinder is appropriate for SCUAV's project design. The information found is referenced in the Appendices section.

Rangefinder Conclusions

The rangefinder that will be chosen is ultrasonic rangefinder. We decided on this rangefinder because it is significantly cheaper than the other rangefinders. Along with this, it is more straightforward to use. The HC-SR04 rangefinder is commonly controlled by ATmega microcontrollers, so there is a lot of resources available for this. Instead of having to control the rangefinder from the flight controller, the ultrasonic sensor can be directly controlled from the ATmega328 chip on the printed circuit board.

3.2.7 ESCs

Electronic speed controllers or ESCs controls the speed, direction, and angular position of an aircraft or vehicle. More specifically, ESCs is used to change the speed of a motor, its route and performs as a brake [25]. The microcontroller will send commands to the ESCs to tell them how much power to supply the motors. The ESCs chosen will be able to work with the motors chose for SCUAV's design.

3.2.7.1 Types of ESCs

There are two types of electronic speed controllers and they are brushed electronic speed controllers and brushless electronic speed controllers. An important thing to note is just as the name implies, a brushed ESC is for a brushed motor and the same applies to brushless ESCs and motors. Brushed ESCs are cheap and their main focus is to alternate the amount of voltage being supplied to the motor. This

ESC has no interest in the positioning of the motor itself [26]. Brushless ESCs on the other hand are a bit more expensive, can last longer and have a high performance rate [25]. One definite way to tell this ESC apart from the other one is the brushless ESC has three wires while the brushed ESC has two wires. The brushless ESC determines how much power to supply a motor by a small voltage that is supplied via the third wire that is not being currently used to power the motor. This small voltage is equivalent to the how fast the motor is turning [26].

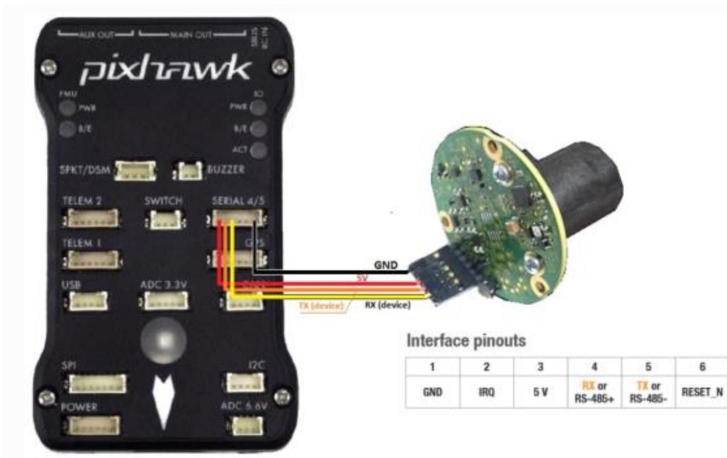


Figure 9: LeddarOne Rangefinder Installation with Pixhawk [23]

3.2.7.2 Chosen ESC

The brushless ESC will the component chosen to implement into SCUAV's design. It was chosen simply because it has better performance quality, will last longer and it will be compatible with brushless motors who are also efficient to use in this project. There is an example, figure 34, of how the brushless ESC is wired is shown in Section 5.

3.2.8 Motors, Propellers, and Lift

The motors and propellers are responsible for allowing the drone to take flight. They help the drone to maintain flight while it is in the air. Selecting the right motors and propellers that are compatible with them is important when deciding on SCUAV's design. We found that based on the size of the propellers, the battery size should be selected. Larger propellers require a lower battery voltage rating. For our propellers, which are 10 by 4.5, we should use a 11.1V, 3 cell battery but instead we have 16.8 V batteries. This battery size supplies us with more power that is need to lift the weight of the drone.

3.2.8.1 Motors

Our drone will have a motor to drive each of the propellers on the drone. After investigating whether brushed or brushless motors would be better to use for our design, we decided on using brushless motors. We quickly learned that brushless motors are used in most of today's technology. To understand the advantages of brushless DC motors over brushed DC motors, it is important to understand the technology of each of these motors.

Table 7: Rangefinder Options [22, 24]

Specifications	Lidar-Lite v3	Ledder Tech Leddar One	Ultrasonic Rangefinder
Size	20 x 48 x 40 mm	50.8 mm	1.3x0.4x0.1 inches
Weight	22 g	25 g	9 grams
Range	40 m (131 ft)	40 m (130 ft)	500 cm
Power	4.75 V - 5 V	5 V	5V
Accuracy	+/- 2.5 cm	+/- 5 cm	-
Interface	I2C, PWM	3.3 V UART	No standard communication protocol
Repetition Rate	500 Hz	100 Hz	NA
Beam Divergence	8 mRadian	3°	NA
Resolution	+/- 1 cm	3 mm	0.3 cm
Laser Wavelength	905 nm / 1.3 watts	850 nm	NA
Operating Temperature Range	-20 °C to 60 °C	-45 °C to 85 °C	-
Price	$149.99	$115.00	$3.00

Note: Highlighted in green is the chosen component for this project.

Brushed Motor

The first rudimentary DC motor was invented by Ernst Werner von Siemens in 1856 and since then has been improved. The modern DC brushed motor uses two permanent magnets, one positive and one negative, that do not move and are on the outside curve of the circular DC motor. These stationary magnets along with are part of the stator.

Table 8: Brushed ESC vs Brushless ESC

Differences	Brushed ESC	Brushless ESC
Cost	Cheaper	More expensive
Motor Position	Does not affect	Effects
Performance Rate	Low	High
Reliability	Short usage	Longer usage
Number of Wires	2	3

Note: The component highlighted in green was chosen for this project.

This will also dictate the direction of current through the coils. As the ring segments rotate and switch brushes, the current through the coils reverse. This current reversal will reverse the north and south polarity of the magnetic field through the coils. As the coils change polarity, they will be will be attracted to the magnets they were previously repelled by and vice versa. This polarity reversal will occur as the coil reaches the magnet of its opposite polarity, causing the rotor to spin continuously. Although the type of brushless DC motor described is a permanent magnet DC motor, there are also DC motors that which replace the permanent magnets with stationary electromagnets.

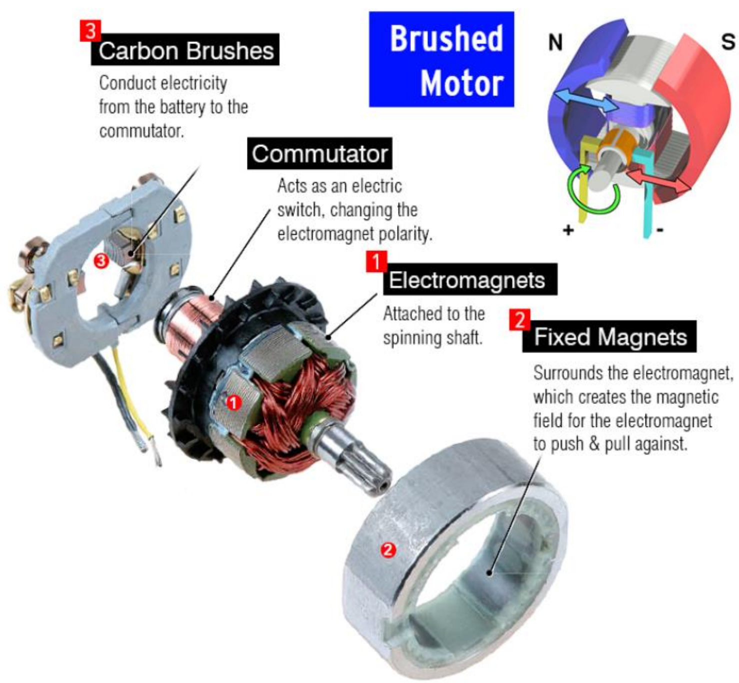


Figure 10: Brushed Motor Anatomy [27]

Brushless Motor
The first brushless DC motor was introduced in 1962. The brushless DC motor is used in many applications today. In a brushed DC motor, the stator has the stationary permanent magnets and electromagnets in the rotor are rotating. However, in the brushless DC motor, the permanent magnets are the ones rotating, acting as the rotor of the device, while the electromagnets are stationary and act as the stator of the device.

There are usually four sets of stationary coils that turn into electromagnets when energized. So, each of the four coils will have a north and south polarity. The electronic controller is responsible for energizing the coils and so it replaces the need for the brushes and commutator that were used in the brushed DC motor. The permanent magnets are attracted to opposite polarities and repulsed by like polarities. The electronic controller will control the polarities of the electromagnets to make the permanent magnets constantly rotate. The electronic controller will switch the electrical charge sent to the coils so that each of the permanent magnets constantly have coils of opposite polarities right in front of them and ones of like polarities right behind it. The electronic controller will know the position of the permanent magnets relative to the coils from a sensor for the electronic controller to know how to energize the coils.

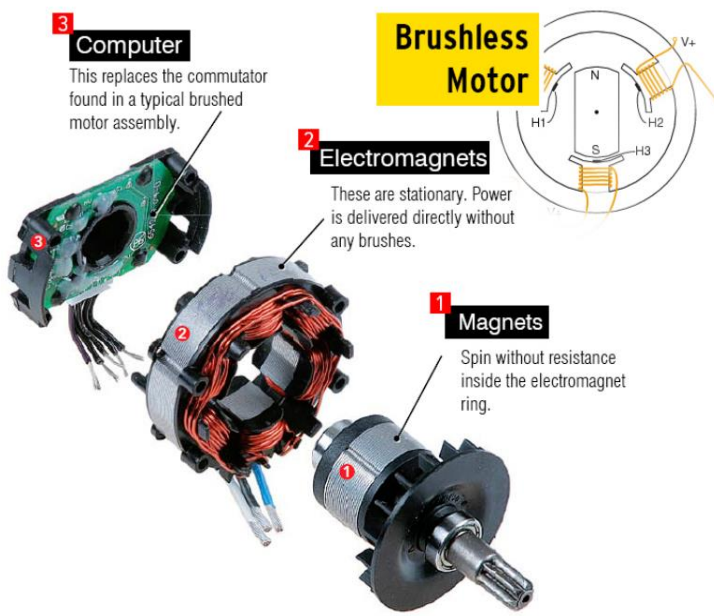


Figure 11: Brushless Motor Anatomy [28]

Brushed vs. Brushless Motor Comparison
The advantages that we found with using brushless motors is that brushless motors require no commutator or brushes. Having no brushes is an advantage for products requiring long term use since the brushes require cleaning and, over time, replacement. The long-term use of brushless motors makes them reliable. Along with this, brushless DC motors are advantageous for devices requiring high torque

over a vast speed range. The contact that the brushes have with the armature will cause friction as the speed of the motor increases. This friction can cause sparking which is another disadvantage of the brushed DC motor. The friction at high speeds also reduces useful torque in the motor which makes the brushed DC motor unable to reach the same high speeds of the brushless DC motor. Along with this, the brushless DC motor has higher heat dissipation. The coils of the brushless motor dissipate heat more effectively. This is because the coils are stationary and are connected to the case. These thermal advantages allow the motor size to be reduced. And the reduced size and high efficiency of the brushless DC motor allow for a higher output power to size ratio. Brushless DC motors also generate less electrical noise. They are quieter because there are no brushes to rub against the commutator. Brushless motors have a higher power density than brushed motors. The power density of a motor is its torque per size ratio.

However, brushed DC motors also have some advantages over brushless ones. For instance, brushed DC motor have a lower cost of construction. A brushless DC motor also requires the use of an electronic speed controller which increases the cost and is less practical. The brushed DC motor will only require a controller if the speed needs to be varied. Brushless DC motors can be used in more rugged environments since no extra electronics are required and it is able to be controlled with simple two wire control. Because the motor will be applied to drones in our project, the brushless DC motor was selected. Drones are high power devices and the brushless DC motor was selected because of its overall efficiency, power density, ability to reach higher speeds, smaller size, better power dissipation, lower noise, long-life and reliability. Having a motor which can be smaller in size is important when designing drones so having smaller and lighter components is critical. Table 9 below shows this comparison below and it highlights which component was chosen for SCUAV’s design.

Inner-Rotor vs. Outer-Rotor Brushless Motors

Once our team decided on using brushless DC motors, we compared inner-rotor and outer-rotor DC brushless motors. For both inner and outer rotor DC motors, the permanent magnet is the moving part acting as the rotor. However, for the outer rotor design, the permanent magnets are on the outside of the motor and the coils are on the inside. And for the inner rotor design, the permanent magnets are on the inside and the coils are on the outside. An advantage of the outer-rotor design is the low cogging torque. Cogging torque is the torque from the permanent magnets being much more attracted to the steel cores than to the coils between the steel cores. Cogging torque is undesirable because it increases vibration and noise and causes torque and speed ripple. Along with this, the rotor of the motor tends to expand at high speeds from the centrifugal force. Having the rotor on the outside of the motor allows an expansion in the permanent magnets without any collision with the coils. To avoid this collision, inner-runner motors are designed to have a gap between the rotor and the coils. However, this gap allows for magnetic flux leakages which decreases its efficiency. The outer-rotor motor is also able to produce more torque than inner rotor motors. This can be described by analyzing the motor constants, Kv and Kt. Motors are described by Kv ratings. Kv is the motor velocity constant which determines a motor's speed of rotation for a given voltage in units of rpm/V. Kt is the motor torque constant. It is the amount of output torque for a given current. Kt and Kv are inversely proportional and a lower Kv will result in a higher Kt. Since outer rotor motors generally have higher Kt, they produce less torque for a given current. And for application needing to drive large propellers, more torque is needed That is why the majority of fixed wing aircraft and multirotors, which sometimes have to power larger propellers, use outer rotor motors.

Table 9: Brushed vs Brushless

Brushed Motor	**Advantages**	**Brushless Motor**
Lower cost of construction		Longer-term use
Has a controller		Provides a higher torque over vast speed range
		Can reach higher speeds
		Better overall efficiency
		Higher power density
		Higher heat dissipation
		Can be used in more rugged environments
		Generates less electrical noise
	Disadvantages	
Requires more maintenance		Less practical (requires a controller)
May cause sparking		Increase in cost
Not as reliable		

Note: Highlighted in green is the chosen component for this project.

However, there are also advantages to having an inner rotor design. Having the rotor on the inside means that there is lower rotor inertia. This allows for faster acceleration of the rotor which is an advantage for applications needing to be quickly started and stopped. However, in disk drive and fan applications the high rotor inertia of the outer-rotor motor is more beneficial. Inner rotor motors also able to reach higher speeds than outer rotor motors. For speeds above 6,000 rpm, inner rotor motors are needed. Motors found in drones range from 1,000 to 20,000 rpm, so if our design exceeds 6,000 rpm, we may have to use an inner rotor motor. Inner rotor motors also tend to have higher Kv. Higher Kv are useful in applications such as remote control cars. The inner-rotor brushless motor also has better heat dissipation. The main benefits of inner rotor motors are having higher heat dissipation, faster acceleration and higher speed capabilities. And the main benefits of outer rotor motors is improved efficiency. Although inner rotor motors have higher speed capabilities, our team realized our drone design will not likely require motor speeds of above 6,000 rpm. Along with this, not producing enough torque is a significant defect of inner rotor motors. Our team decided that the outer rotor motor would be the best design because of the improved efficiency and higher torque.

When selecting motors it is important to pay attention to the maximum current supplied and the motor's working current. These motor specifications will determine the type of electronic speed controller with the correct current rating to use. The ESCs supply power to the motors and because of this, they should have a larger current draw than the motors. An example of a correct motor and ESC selection based on current ratings is the selection of a 25 max amp ESC when the motor has a max current rating of 23 amps and a working current of 19 amps. It can be seen that the ESCs have higher current ratings than the motors.

3.2.8.2 Propellers

The propellers are controlled by the microcontroller via the signals the microcontroller receives. These signals are sent from the microcontroller to the motors and then to the propellers. The propellers in turn navigate the drone while it is in flight. When picking the right propeller it all comes down to picking the most appropriate type for the project, its size, the number of blades, the pitch, the amount of power it is issued in order to fly and many other factors.

3.2.8.2.1 Types of Propellers

There are several different types of propellers used in many drone projects. These types all serve a purpose for what kind of drone is being built. Some might be needed for industrial purposes, others might be needed for fun lightweight projects or some might be needed to have that professional look and they might be safer for the environment. The table below, Table 10, discusses the pros and cons of each material used for the propellers. Carbon fiber will be the material chosen for this project.

Wooden:

Wooden propellers have very low flexibility, heavy, and very inefficient if the drone crashes [29]. The advantages are the propellers are well balanced and they have a more solid structure. This material is classified in the medium cost range because the propellers will more than likely have to be replaced if they are slightly damaged.

Carbon Fiber:

Carbon fiber propellers are light, durable, and seems to be favored when wanting a powerful machine [29]. One of the disadvantages of this material is damage level is high, meaning that these propellers could possibly cause damage to the entire drone system if hit in mid-air. What makes this material expensive is the material mixed with carbon-fiber. Knowing this, the flexibility of the propeller varies. Propeller guards would be great investment for this type of blade.

Plastic:
Propellers made from plastic are usually pretty cheap, light, and flexible. The cons to this material are it has a low performance rate, there is not a good airflow, and durability is low [29]. On the positive side this material is less likely to get damaged and it is cheaper for hobbyists to use. The cost to replace these propellers is pretty cheap and would be suitable for this project in the building process.

Figure 12: Example Types of the Propellers Mentioned Above [31]

3.2.8.2.2 Pitch

The pitch of the drone is the distance travelled by the drone in one single propeller rotation [30]. If the propeller has a low pitch, then it will likely have a high amount of torque. This is a good thing for the motor since the motor will be able to operate at a lower current rate and the drone will be able to keep steady during flight [30]. If the propeller has a high pitch, then there will be a low amount of torque thus making the drone unsteady.

3.2.8.2.3 Size

The size of the propeller matters when carrying the weight of the drone. The larger the propeller the harder it is to carry the drone simply because it is hard to change the speed of the propellers at a fast rate. Bigger propellers in theory may look like they would do a good job but they are a real disadvantage to drone flight. Smaller propellers are recommended since they allow for the speed of the drone to change at a better rate. The blades of the smaller propeller spins faster allowing the motors to operate at the speed they prefer with ease [30]. One thing to note about size of the propellers is the large propellers are useful for drones that

3.2.8.2.4 Number of Blades

The number of blades to use for a drone varies but the most efficient is using two blades. Adding more blades will not necessarily mean that the drone will gain more thrust [31]. Two blades will be chosen for this project's design. Once the blades are added this drone's structure, the blades will have to calibrated to work properly during flight. To calibrate these blades, one will have to be rotated in a clockwise direction and the other in an anti-clockwise direction. More details on how this calibration works can be found in the Hardware Implementation section of this document.

Table 10: Types of Propellers

	Wooden	**Carbon Fiber**	**Plastic**
Weight	Heavy	Light	Light
Cost	Medium	Expensive	Cheap
Performance	Medium	High	Low
Durability	Low	High	Low
Flexibility	Low	Medium	High
Damage level	High	High	Low

Note: The component highlighted in green is chosen for this project.

3.2.8.3 Lift: Flight Capability

When making decisions on our drone's design, it is important to predict the performance of our drone. Before testing the drone and integrating its component whether the drone can support its weight requirements. There are several key factors which will affect a drone's flight. The weight of the drone, the motor's characteristics, and the characteristics of the propellers all play vital roles in the success of a drone's flight. Each drone will have a difference flight ranges, carrying capacities, and flight times based on its drone design.

There are several equations to aid in these predictions for a drone's success. For instance, the required thrust for each motor can be calculated. The equation for each motor's thrust is the total weight of the drone multiplied by 2 and divided by the number of motors of the drone. Along with this, it is useful to know each motor's thrust since each combination of motor and propeller should be able to provide the together with the propellers should provide enough combined thrust to be greater than twice the flighting weight of the drone. It is also important to calculate the payload capacity of the drone.

The weights of each of the components such be tested before integrating the drone's components. Doing this will allow the drone's flight capabilities to be tested. A list of all the drone's components weights has been listed in Table 11.

Table 11: Component Weights

Drone Component	Weight
Microcontroller: Pixhawk Mini	15.8 grams
Microcontroller: GPS module	22.4 grams
Drone Frame	478 grams
Battery	
Ultrasonic Range Sensor	15 grams
Drone Gripper	123 grams
Motor x6	318 grams
OpenMV Camera	16 grams
ESC x6	150 grams
Landing gear	
Propeller x6	54 grams
Raspberry Pi	45 grams
PCB Weight	About 50 grams
Styrofoam load	200 grams
Power distribution board	0 grams
Motor for Gripper	42 grams
Battery Charger (PCB power supply)	About 80 grams
LiPo Battery Expansion Board (Raspberry Pi power supply)	77.11 grams
Total	1,697.7 grams

We found that the type of battery which we select will greatly influence the carrying capacity and flight time of our drone. For our project we will be using the A2212 1000KV Brushless Outrunner Motor. And the propellers we will be using in combination are 10 by 4.5 inches. Our motors will draw a maximum of 13 amps. And based on the information from the datasheet for the motors used, given in Figure x, the maximum thrust that each of motors can provide with 10 by 4.5 inch propellers is about 802 grams. If each of the motors provide their maximum thrust, this leads to a total thrust of 802 grams *6, which is 4,812 grams. Since the thrust should, in general, be double the weight of the drone, the weight of the drone should be less than 2,406 at the minimum. However, after conducting research, our team realized that it is better to have the weight of the drone be closer to 30 or 40% of the weight of the drone. Since we wanted to be more conservative, we calculated the maximum weight of the drone to be 1,443.6 grams for the weight being 30% of the maximum thrust, and 1,924.8 grams for the drone weight if it is 40% of the maximum thrust. Along with this, the drone needs to be even lighter to support the drone being able to hover. When calculating the total weight of the drone, an additional 20% of the drone weight needs to be added to its total. All the thrust calculations will need to be based on this total.

We also calculated the needed voltage amount from the battery to be 11.1 volts. We came to this conclusion based on the maximum amount of voltage that the motors were able to draw, which was found to be no more 11 volts based on the motor datasheet. The datasheet for the motors we would be using also showed that we would need a battery that is two to three cells. So the motors we are using are not able to support more powerful battery like a four, five, or six cell battery. Each battery cell holds 3.7 volts, so a two cell battery is 7.4 volts and a three cell battery is 11.1 volts. Based on this information we decided to use a three cell battery since we will try to use the maximum thrust that the motors are able to provide and will need to run the motors at a voltage closer to 11 volts to do this.

Another reason we decided to go with a 3s battery is because the recommended battery size for our drone frame is 3s to 4s. A 4s is usually recommended for smaller propeller sizes, and is recommended for a propeller which is 8 inches. However, when using propellers which are larger and around 10 inches, it is recommended to use a 3s battery. This is because the smaller propellers will need to have a higher rpm and thus need to run at a higher voltage and require a battery that can supply a higher voltage. And a 4s battery is able to supply 3.7 more volts than a 3s battery is able to. However, larger propellers are able to run at a lower rpm and supply the same thrust as a smaller propeller running at a higher rpm. This relationship is shown in the Figure 13 with the motor datasheet information. And since we will be using large 10 inch by 4.5 inch propellers, we will be using a 11.1 volt, or 3s battery, instead of the 14.8, or 4s battery.
A major contribution to the flight time of the drone is also based on the milli-amp-hour of the battery. A battery which has a higher milli-amp-hour provided by the battery usually means that is able to support greater vehicle weights. An important consideration when selecting the drone battery is to make sure that the battery is able to supply a sufficient amount of current. If the motors draw more current than the battery is able to support, then the motors will overheat and the battery could also burst into flames. The C rating of the battery determines how much current

draw it is able to support. And the milli-amp per hour rating determines how fast the current is released from the battery. Because of this, the milli-amp per hour rating is strongly related to the drone's flight time. If the milli-amp per hour rating is very high, the battery will release the battery current very quickly and the battery will need to be recharged in a shorter amount of time. Figure 11 shows the equation used to find the C rating of the battery. We decided to use a 5500 milli-amp per hour battery which has a C rating of 35C. When 5500 is divided by 1000 to give 5.5, which is the capacity of the battery, and multiplied by 35, which is the C-rating of the battery, is gives a maximum discharge current value of 192.5 amps, 5.5 multiplied by 35. This value is far above the current discharge value required for our drone. Since each of our motors require a maximum current draw of 13 amps, the battery needs to be able to provide at least 78 amps. Since our battery is able to provide 192.5 amps, it is able to provide a current draw well above the required 78 amps. This means that motors will not be trying to draw more current than is available and put our battery at risk of bursting into flames. Using a battery which provides a larger current draw and has a higher C-rating is usually heavier than batteries with a smaller C-rating, which is the downside of using a C-rating which is larger. Larger C-rated batteries usually also have a larger milli-amp per hour rating which means a longer flight time. So it is a trade-off, as the C-rating increases, the flight time is usually able to be longer, but the weight of the drone will increase. The reason we decided to use a battery that can provide a more current and has a higher C-rating is because we wanted a battery that has a higher milli-amp per hour rating. Selecting a higher C-rating usually means that the battery weight will significantly increase, however we were able to find a battery which was very light weight while also having the advantage a large milli-amp per hour rating of 5500.

Maximum Discharge Current = C-Rating x Capacity

Figure 13: Equation for Battery Rating

We also used a very useful online calculator to help determine the type of battery we would need. We inputted the following information, we would be using six motors and the maximum current draw per motor was found to be 13 amps, based on the motor datasheet. We battery we believe will be appropriate for our drone is a 3s battery, so it provides 11.1 volts, which has a milli-amp per hour rating of 5500. Based on all this information inputted into the calculator, we determined the approximate flight time of our drone to be around 14 minutes, or 11 minutes if following a more conservative calculation model.

3.2.9 Camera

To accomplish our main goal for SCUAV, a camera will be needed to aid the drone's capabilities in recognizing objects it needs pick up and put down, recognize the building site, and determining its path. A camera with a good processor and good lens will ease our workload. It will transfer the object recognition to the MCU and will be able to navigate the drone to the potential positions. The ideal camera should have a clear image resolution in order to clearly detect the styrofoam blocks and other construction material and also to detect the building site for where to place those materials. Below is a discussion on how the thought process of selecting a camera was discussed among the group and a final decision will be made.

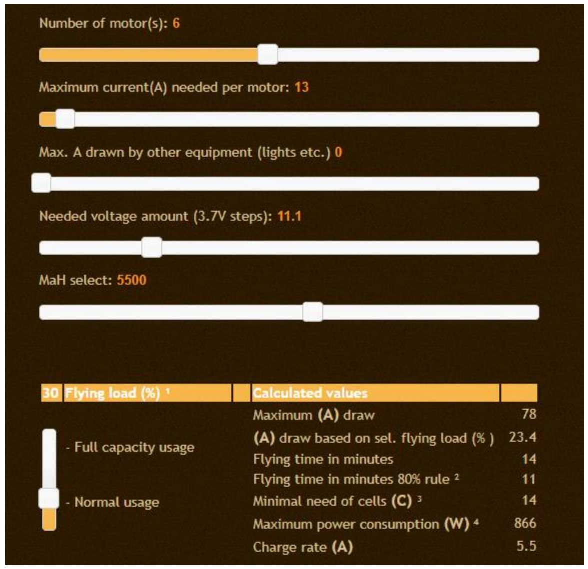


Figure 14: Flight time Calculator

Camera comparison

There are a wide variety of camera selections. SCUAV's main choice at first was the CMUcam5 Pixy Image Sensor, which was recommended by Dr. Samuel Richie. Other options for the camera include the Raspberry-Pi's Camera Module v2, which allows to use the PiCamera library to take pictures and videos and also the OpenCV library. The OpenMV Cam M7 Image Sensor is also a good option, due to its ARM processor running at 216MHz.

Propeller	Gear Ratio	Volts	Amps	Watts	RPM	Speed (mph)	Thrust (g)	Thrust (oz)	Temp (C)
GWS HD 8x4	1	7	3.35	23	6630	25.1	226	7.97	
GWS HD 8x4	1	7.9	4.1	32	7410	28.1	287	10.12	
GWS HD 8x4	1	8.9	4.85	43	8220	31.1	347	12.24	
GWS HD 8x4	1	9.9	5.65	55	8940	33.9	420	14.82	
GWS HD 8x4	1	10.9	6.5	70	9660	36.6	495	17.46	
GWS HD 9x5	1	6.9	5.5	37	6000	28.4	348	12.28	
GWS HD 9x5	1	7.9	6.7	52	6660	31.5	436	15.38	
GWS HD 9x5	1	8.9	7.85	69	7290	34.5	526	18.55	
GWS HD 9x5	1	9.9	9.25	91	7920	37.5	627	22.12	
APC E 10x5	1	6.9	7	48	5610	26.6	406	14.32	
APC E 10x5	1	7.9	8.45	66	6120	29.0	505	17.81	
APC E 10x5	1	8.9	9.9	88	6690	31.7	604	21.31	
APC E 10x5	1	9.9	11.45	113	7170	34.0	702	24.76	
APC E 10x5	1	10.9	13	141	7650	36.2	802	28.29	
GWS HD 10x6	1	6.9	7.2	49	5610	31.9	424	14.96	
GWS HD 10x6	1	7.9	8.7	68	6180	35.1	526	18.55	
GWS HD 10x6	1	8.9	10.1	89	6690	38.0	617	21.76	
GWS HD 10x6	1	9.9	11.7	115	7200	40.9	722	25.47	
GWS HD 10x6	1	10.9	13.25	144	7680	43.6	817	28.82	
GWS HD 10x8	1	10.8	18.2	196	6390	48.4	733	25.86	

Figure 15: A2212 1000KV Brushless Outrunner Motor Datasheet Information

3.2.9.1 Raspberry-Pi Camera Module v2

The Camera Module v2 is a beginner-friendly camera that can be used in any Raspberry-Pi devices. It can be used to take photographs and high-definition videos. The Camera Module v2 has an 8 megapixel Sony IMX219 image sensor. It is capable of producing 3280 x 2464 pixel static images and support up to 1080p30 video. [32] This camera is also very easy to program any object recognition techniques needed to make this project a success.

As mentioned before, this camera has great object recognition capabilities, as it can be programmable using OpenCV and the PiCamera libraries. It is supported by the Raspbian, a Raspberry-Pi operating system. What makes the Camera Module one of our choices for the main camera is that it allows to utilize OpenCV. Very easy to use, this camera is very popular among robotic enthusiasts very easy to learn from. [33] It has an ARM based processor, BCM2835, connected to the Raspberry-Pi via CSI bus. It has a higher bandwidth link which carries pixel data from the camera back to the processor. A disadvantage to the camera module would be that it requires an additional Raspberry-Pi microcontroller for the camera to function. We only need the Raspberry-Pi to function as an on-board computer which will be later explained.

3.2.9.2 OpenMV Cam M7

The OpenMV Cam M7 camera is a small, low-power microcontroller board that allows you to manipulate images using Machine Vision. Machine vision slightly differs from computer vision in the sense that the objects to be viewed are already known and almost all observed events are predictable. It has become a candidate for our camera selection because it has an ARM based processor that runs at 216 MHz and uses 512KB of RAM and 2MB of flash, which enables 640x480 grayscale and RGB 565 high definition images and videos. The OpenMV image sensor contains substantial object recognition capabilities because it is able to read up to 16 colors per image. [34] Each of those color has a number of distinct blob in which you can detect the position, size and centroid of each blob. Similar to the Raspberry-Pi Camera Module v2 mentioned earlier, it is capable to use computer vision libraries to aid its object detection capabilities using either high-level Python in MicroPython operating system. It helps as an alternative to using low-level languages like C/C++ although it was still an option.

A drawback for this device is that does not support using the OpenCV library. It isn’t necessarily a drawback but the OpenCV library helps developing computer vision algorithms easier and is more open source. An advantage we have discovered throughout the development of the project is that OpenMV libraries are much to learn from. The comparison between using OpenCV and other software will be mentioned later. Other applications for this image sensor include marker tracking, color tracking, frame differencing, template matching, rectangle/circle detection, data matrix decoding/detection, image capture/video recording, optical flow and more computer vision techniques. The Optical Flow application in the image sensor will help SCUAV determine its stability in the air. As seen in Figure

14, the camera also has an I2C bus and an Asynchronous Serial Bus so it can interface with other microcontrollers. We want to lean more on the I2C bus due to having a much better communication with multiple master devices and being less susceptible to noise compared to other communication interfaces. Throughout the project development, however, I2C communication is not suitable for the OpenMV Cam M7 because it performs poorly as a slave. We want to have our on-board computer as the I2C master. The OpenMV would end have to be in a system call in order to feed the I2C hardware data for a master device to get anything other than zeros. In other words, the OpenMV Cam M7 would have to be waiting for the Raspberry-Pi to tell it what to do before it can do anything. The USB protocol is then used for sending data to the on-board computer. It also shows all I/O pins. At $65.00, the OpenMV Cam M7 image sensor is affordable enough to be one of our choices as the main camera for SCUAV.

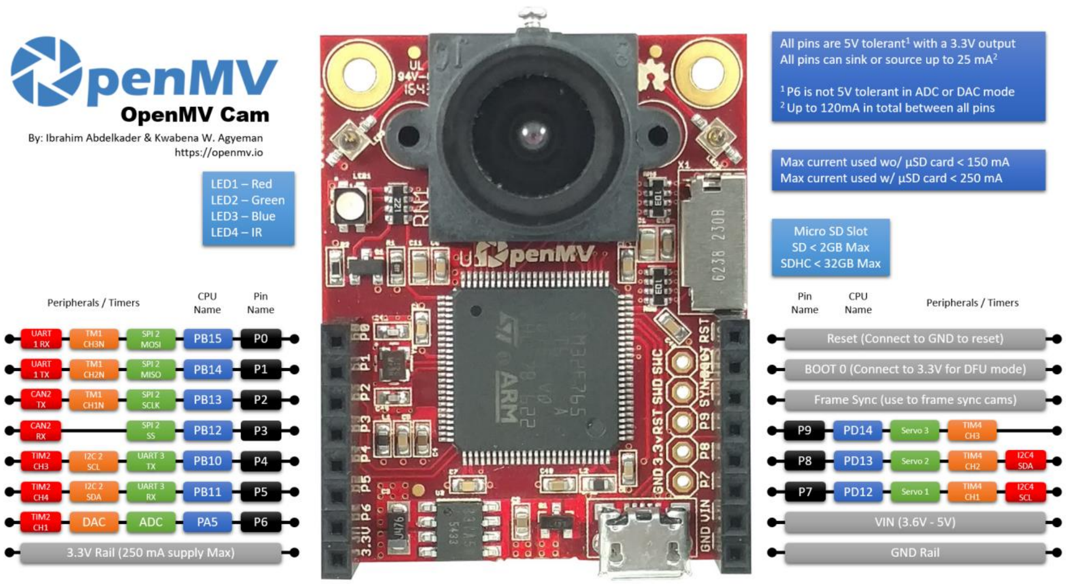


Figure 16: OpenMV Cam M7 Pins [34]

3.2.9.3 CMUcam5 Pixy Image Sensor

The CMUcam5 Pixy is a small image sensor that is capable of connecting with other microcontroller devices. Recommended by Dr. Richie, the CMUcam5 Pixy processes images and sends only important information to the microcontroller without outputting large amounts of unnecessary data. The Pixy is able to send information to different microcontroller using any communication interface. Even without a microcontroller, the Pixy can manipulate events using digital or analog output.

The CMUcam5 Pixy supports a variety of microcontrollers such as the Arduino, Raspberry-Pi, Beaglebone, and more. It processes an image at a rate of 20 milliseconds. It uses an Omnivision OV9715 image sensor that enable 1280x800. The Pixy has an NXP LPC4330 processor that runs at 204 MHz and has 264 KB of RAM and 1MB of flash. This device is one of our choices for the main camera because it is capable of “teaching” an object the user is interested. Simply, hold the object in front of the lens and hold the button on the top right of the device as shown in Figure 14. Its RGB LED provides feedback regarding the object it is looking at. The Pixy will then create a statistical model of the color contained in the object and will store in flash. With this statistical model, Pixy can find object with similar colors. A drawback of the CMUcam5 Pixy is that is does not support Computer Vision libraries. The CMUcam5 Pixy’s cost is at a decent price of $69.00. In Table 12, a comparison of all three cameras mentioned is shown along with its specifications. Ultimately, the OpenMV Cam M7 has been chosen as the main camera for SCUAV due to its better performance and its versatile computer vision capabilities. Applications in how to use the chosen camera will mentioned in the Computer Vision section.

Figure 17: CMUcam5 Pixy [35]

All the camera comparisons can be seen in Table 12. The table shows all the unique specifications that each camera has. The positives and negatives are also shown for the different options. At the bottom of the table is the cost comparison. Some of the cameras have similar features and all of them can be good candidates for SCUAV’s design. Another thing to notice is the some of the components of the cameras are made by the same manufacturer but they have different part numbers or they are just a different model designed to specifically work within that camera type.

3.2.10 On-board Computer (on drone)

In order to help monitor local operations on the flight controller we would need an on-board computer. We would also need an on-board in order send back data to the web interface. This can also be used to extract data and live feed from the camera and send it to the web interface for the user to better understand the current construction phase. The Raspberry-Pi is a perfect example of an ideal on-board computer. For SCUAV, we had the choice of either using BeagleBone Black or the Raspberry-Pi 3 Model B.

Table 12: Camera Comparison

Specifications	**Camera Module v2**	**OpenMV Cam M7**	**CMUcam5 Pixy**
Image Sensor	Sony IMX219	Omnivision OV7725	Omnivision OV9715
Processor	BCM2835	STM32F765VI ARM Cortex M7	NXP LPC4330
RAM	N/A	512 KB	264 KB
Resolution	640x480	640x480	1280x800
Communication	N/A	SPI, USB, I2C, Asynchronous Serial	UART, I2C, SPI, USB, Digital, Analog
Computer Vision	Yes	Yes	Yes
Price	$26.39	$65.00	$69.00

Note: Highlighted in green is the chosen component for this project

3.2.10.1 Raspberry-Pi

Both the BeagleBone and Raspberry-Pi function as a desktop computer and can be used to build smart devices such gaming devices, phones, servers, and more. It is very beginner-friendly and uses a Linux-like operating system called Raspbian, which is pre-installed in the device. The Raspberry-Pi uses an ARM processor and runs at 1.2GHz with 1GB of RAM. An advantage the Pi has is that it is capable of using Bluetooth and comes with an on-board BCM43438 Wireless LAN, which is needed to communicate with and send data from the flight controller to the base station. [36] This device draws power at 150-210 mA at 5V under varying conditions. A drawback from this device is that it requires a microSD card to operate. It is a main choice for our on-board computer because of its simplicity and its open source capabilities. A list of specifications for the Raspberry-Pi is shown below in Table 13.

Figure 16 below shows the Raspberry-Pi 3's physical layout. The Pi contains four USB ports, and 40 extended GPIO pins. On the bottom, there is the MicroSD port needed for loading the operating and storing data. We would be able to use it store data from the flight controller and send it to the base station via Wi-Fi or Bluetooth.

Figure 18: Raspberry-Pi 3 [36]

3.2.10.2 BeagleBone Black

Unlike the Raspberry-Pi, the BeagleBone Black is not much for beginners and a little more complex to learn from. Similar to the Raspberry-Pi, it can function like a computer and used to build smart devices. The BeagleBone Black have more features as it can function for embedded systems projects and low power programming. [37] As shown in Figure 17, the BeagleBone Black hasa GPIO capability of 65 pins, 15 more than the Pi and a total of 92 connection pins. A disadvantage the BeagleBone has is that it only has an on-board Ethernet port and one USB host for network communication. Although this microcontroller has a default operating system of Angstrom, it is compatible with other operating systems, like Linux, Android, Debian, and more. This is not ideal for this project to function since most of these components intertwine with each other and all the connection space is needed.

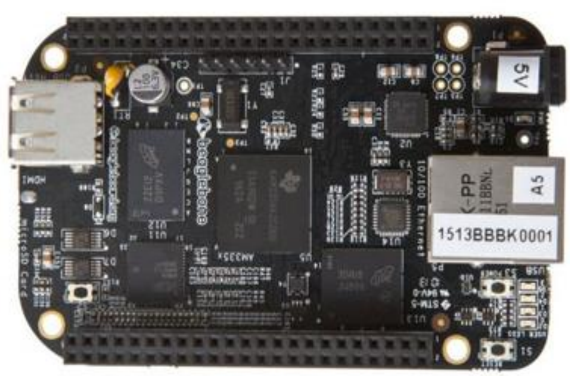


Figure 19: BeagleBone Black

At 5 volts, the BeagleBone Black has a power draw of up to 460 mA under varying conditions, more than the Raspberry-Pi and might not be as ideal. Another advantage BeagleBone has is that it comes with 4GB on-board storage while Raspberry-Pi has no onboard storage and requires a MicroSD card to store data.

This device is a candidate for SCUAV's on-board computer, as shown in Table 13 for comparison, because of its versatile capabilities and connectivity. Ultimately, the Raspberry-Pi 3 Model B was chosen as the project's on-board computer due to its simplicity in setup, programming and wireless communication. It has a huge community support that makes it easier for open source.

Table 13: On-board Computer Comparison

Specifications	Raspberry-Pi 3 Model B	BeagleBone Black
Memory	1GB LPDDR2	512MB DDR3L
Processor	1.2GHz 64-bit Quad-core ARMv8 CPU	1GHz TI Sitara AM3359 ARM Cortex A8
Networking	802.11 Wireless LAN, Bluetooth	1 Ethernet port
Operating System	Raspbian	Angstrom
GPIO	40 pins	65 pins
Power	150-350 mA @ 5V	210-460 mA @ 5V
Cost	$35.00	$45.00

Note: Highlighted in green is the component chosen for this project.

3.2.11 Computer Vision Software

One of the most important aspects for the SCUAV project is the use of computer vision. Being one of the world-changing topics today, Computer vision, according to the British Machine Vision Association, is the science that aims to enhance the capability of machines and computers to visually sense the world around them [38]. It focuses on automatic extraction, analysis, and image processing.

To achieve SCUAV's purpose, the software team will use Computer Vision to aid the drone's capability of detecting any object in its way as well determining what construction materials it is picking up, how to pick them up and where to place them. We have chosen our ideal camera needed to achieve this goal, which will be the OpenMV Cam M7. A possible second camera may be used to aid the drone's path.

Computer vision uses heavy convolution, matrices and linear algebra. [39] Convolution helps alter the image's appearances such as blurring, smoothing, getting specific edges (horizontal or vertical), and more. The use of Computer Vision will require software, such as OpenCV, Tensor Flow, Clarifai and others. Since we are using the OpenMV image sensor as our main camera, we shall use its open source software which provides us a learning environments and allows to write high level Python scripts to aid the drone's computer/machine vision methodologies. Python is the chosen language for our computer vision implementation due to its simplicity in syntax and multithreading. The image sensor aids us to perform edge detection, color detection, template matching, optical flow, frame differencing, and more.

To use the OpenMV Cam M7, one can simply connect the camera to any microcontroller. For this project, we will integrate the image sensor with the flight controller in order to receive its data. We will also connect the image sensor to the on board computer for live stream in the web interface application. This will help the user visualize the entire construction process. Drivers must be installed in order to get things started. The OpenMV IDE will be launched once everything is installed properly. Any out-of-date firmware will be updated as necessary. A USB cable is needed to connect the image sensor to the computer, I2C is used to connect to the flight controller. The OpenMV is smart about connection as well as automatically filter out all serial ports that are not part of the image sensor. The IDE has a frame buffer viewer which allows the user to view the live stream of the OpenMV Cam. As mentioned earlier, we will use the live stream on the web interface.

3.2.11.1 Application

Computer Vision serves a purpose in a wide number of applications. A major application it is involved in is robotics, especially when building humanoid robots, drones or other machines. Another way Computer Vision is involved is in security. For example, it can used when detecting a person's face when using an ATM or spotting intruders invading a home. Image restoration is also a major application especially when restoring broken artifacts or to even for optical character recognition. We are hopeful to revolutionize computer vision in this project by recognizing industrial objects, picking them up and placing them in its appropriate locations.

Software utilized to compute such applications are OpenCV, OpenMV, Clarifai, Tensor Flow, and more. OpenCV is known as the most popular software as it aims mainly at real-time computer vision. Tensor Flow on the other hand is used as a machine learning library for data flow programming while it is used for computer vision library. Tensor Flow is a little more complex as it is mostly a symbolic math library for Deep Learning. Clarifai solely focuses on image recognition and it works by using a statistical model that shows the probability of what object it is seeing. Clarifai can also learn what object it is viewing by simply repeating place the object in front of the camera. OpenMV is a machine vision library that supports image recognition and also can learn about what object it is viewing. It supports very little OpenCV unlike other software, but also includes many application as stated before. OpenMV is simple to use and it has its own development environment, hence why this software will be used for the computer vision development.

3.2.11.2 Object Recognition Methods

Several object recognition methods are used in practice when implementing computer vision. It is still a challenging task in computer vision. We humans are able to recognize images with minimal or no effort although these images may vary depending on its size or scale as well as the color and its edges. But even then us humans fail to recognize objects to the most accurate level.

3.2.11.2.1 Edge Detection

Edge detection checks all edges of an image, inside and out. Color is not a factor in this object recognition method. Being one of the most important tools in computer vision, it uses a variety of mathematical. However, SCUAV would probably need the use of both color and edge detection to recognize materials. The figure below, Figure 18, demonstrates how edge detection works.

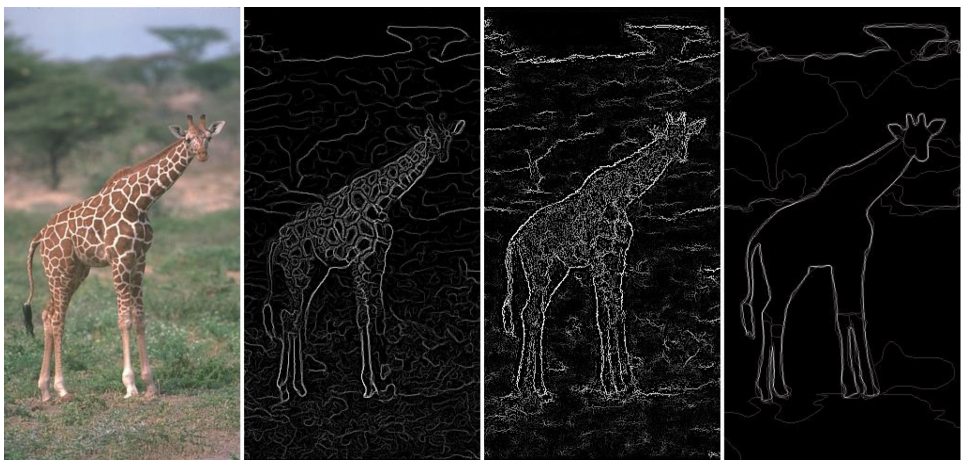


Figure 20: Edge Detection

Several algorithms are used to perform edge detection such Sobel and Canny. The Sobel algorithm or operator is used in image processing and mainly focuses on edge detection methods. [40] It performs two-dimensional spatial gradient measurement on an image and focuses on regions with high spatial frequency that corresponds to edges. This algorithm consists of a pair of 3x3 convolution kernels. One of the kernels is simply rotated 90°. As mentioned earlier, these kernel convolutions shown in Figure 19 are designed to maximally respond to edges running horizontally and vertically relative to the pixel grid.

The Canny algorithm is designed to be detector. It takes an input a grayscale image and as an output it produces an image displaying the positions of tracked intensity discontinuities. [41] The algorithm starts by smoothing the image enough to eliminate noise. It then finds the intensity gradients of the image. Apply non-maximum suppression (edge-thinning) to eliminate invalid response to edge detection, then apply double threshold to find the potential edges. Last step for edge detection to suppress all the other weak edges that are not connected to the strong edges.

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Figure 21: Sobel Convolution Kernels

3.2.11.2.2 Divide and Conquer

Used in practice in all sorts of algorithm development, this can also be used in computer vision. The problem must be divided into sub problems similar to the original problem. Then those sub problems need to be solved recursively. [42] Subproblems must have a base case. Finally, combine all sub problem solution to come up with the final solution. Divide and Conquer should be used when similar subproblems are not computed many times. Otherwise, paradigms like Dynamic Programming and Memoization should be used. Binary search is an example for divide and conquer, as the same subproblem should not be evaluated again.

In the case for computer vision, the divide and conquer algorithm is used in appearance-based methods. We can consider all positions as a set. We must find the best position in the cell. The base case would be if the bound is too large then prune the cell, otherwise divide the cell into subcells. When the cell is small enough, stop. This technique is guaranteed to find all matches that meet the standard. Another divide and conquer example for computer vision would be 3D object reconstruction from a single 2D line drawing. One can divide the complex single line drawing into different simple line drawings, then rebuild the 3D shapes form these simpler drawings. Finally, merge the 3D shapes into one complete object. Other examples for this method would be for finding event trajectories and detection as mentioned in the COCOA project by UCF's CRCV. [43]

3.2.11.2.3 Gradient Matching

This appearance-based object recognition method can be a choice for our computer vision development. In gradient matching, edges are mostly robust to illumination changes and it does not throw away a lot of information for it compares image gradients. In order to perform this method, a computation of pixel distance is needed. That computation will be used as a function in both pixel intensity and pixel position. This method can also be used in color, which would be enormously beneficial for SCUAV as it may need color recognition.

3.2.11.2.4 Scale-invariant feature transform (SIFT)

SIFT is a feature-based object-recognition method that focuses on key points of an object. These are extracted from a set of referenced images and stored in a database. In order for a new object to be recognized, each of its features needs to be compared individually with the database and find matching features based on Euclidean distance of their feature vectors. For example, if we have 100 random images and we want to find a specific image in that database, we would match that image with the entire database. Within there, it will take the highest key value that is closest to its match. However, this algorithm might be slow for SCUAV due to iteration and searching through the entire database.

3.2.11.2.5 Template Matching

Template matching is a brute force algorithm for object recognition and can be another form of edge detection. A simple sub-image is then used as a template. A pixel-by-pixel comparison is then used on the template with the image being scanned for. The template is placed at every possible pixel of that image. Utilizing a similarity metric, find the pixel with the maximum match. That is the spot that has the most similar pattern to the template.

However, this object recognition method has some flaws for our project due to its inefficiency. If there is no matching spot in the desired image and the template, a match will still be found. A change in shape/size/color in the object with respect to the template leads to a false match, which is called affine variant. Although this object recognition algorithm is rarely used, it is a candidate for SCUAV because of its simplicity.

3.2.11.2.6 Speeded Up Robust Features (SURF)

SURF is an object recognition method similar to SIFT. Based on the same principles as SIFT, it detects the specific feature of an image and compares with the images in the database, but details in each step differently. It consists of three parts: interest point detection, local neighborhood description and matching. It highly more efficient than SIFT because instead of scaling using cascade filters, it utilizes square scaled filters. It is much faster to use filter images using squares with the help of integrals. [44] We have decided to choose SURF as our object recognition method due to its superb efficiency.

Figure 22: SURF Object recognition

3.2.11.2.7 Features from Accelerated Segment Test (FAST)

FAST algorithm for corner detection is a fast-enough algorithm that allows to work in real-time applications like SLAM and other video processing applications. Its detector uses a 16-pixel circle around the desired image to classify whether candidate point p is a corner. Each pixel in the circle is labeled 1 to 16 clockwise as denoted in Figure 21. [45]

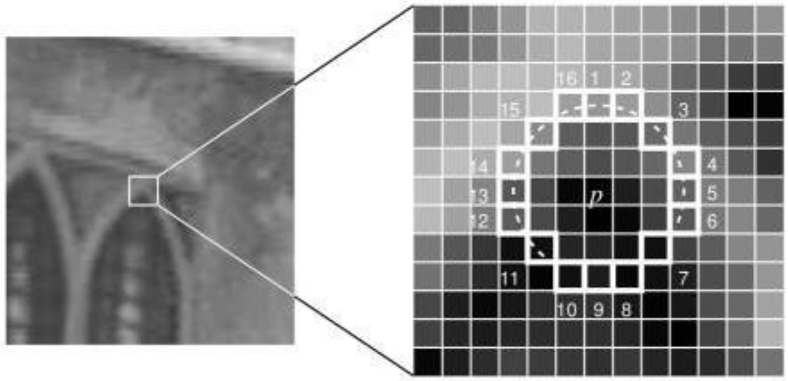


Figure 23: FAST Feature Detection

If there's a set of certain number of contiguous pixels are brighter than candidate point p plus a threshold value t, OR they are are darker than candidate point p minus a threshold value, then the candidate point is considered a corner. A high speed test is then performed to exclude the amount of non-corners. This is proven to be faster than other algorithms mentioned but has its weaknesses. This includes the high speed results are thrown away, multiple features are detected adjacent to one another, and the pixel choice is not the best because its efficiency is dependent on the distribution of the corner appearances.

3.2.11.2.8 Binary Robust Independent Elementary Features (BRIEF)

BRIEF algorithm is an alternative to using SIFT and SURF as those two algorithms take up a lot of memory. For this reason, SIFT uses 128-dim vector for descriptors for thousands of features, which takes up 512 bytes of memory. SURF similarly takes up a lot of space with descriptors (256 bytes). More memory taken means less efficiency for matching. SIFT can still use hashing for converting its floating-point number descriptors to binary strings. These binary strings can be used to match features using the Hamming distance. It provides a better speed-up because Hamming distance just XOR and applies bit count, which is rapid in the CPU. This, however, does not solve the memory problem.

This is where BRIEF comes to play. According to the OpenCV documentation, BRIEF provides a shortcut for finding binary strings without the use of descriptors. A smooth patch of the image is taken and selects a set of some location pairs. Some pixel intensity is done on these pairs. Once these binary strings are obtained, use Hamming distance to match the descriptors. As memory efficient as it is, it is only a feature descriptor and a detector, so SIFT, SURF, or other detectors will need to be used. BRIEF could possibly be a disadvantage to SCUAV's programming side of the drone's functionality to this project.

3.2.11.2.9 Oriented FAST and Rotated BRIEF (ORB)

Having explained one of the fastest feature detector algorithm like FAST and memory efficient feature descriptor BRIEF, the ORB algorithm has been developed to be an alternative to SIFT and SURF. It is basically a combination FAST detector and BRIEF descriptor along with other modifications. This algorithm is another possible contender for programming SCUAV's camera.

Since FAST does not compute for orientation of the key points,ORB calculates the intensity of the centroid of the image patch with its corner at center [46]. The orientation is given by the direction of the vector from the corner point to the centroid. Rotation invariance (image is transformed and corner locations do not change) is improved by calculating moments.

ORB uses BRIEF as descriptors, but BRIEF is not friendly with rotation. With ORB, BRIEF can be rotated according to its orientation of key points. For any set of n binary strings, a 2xn matrix is formed containing its pixel coordinates. This algorithm approximates its angle to increments of 12 degrees and create a lookup table of pre-calculated BRIEF patterns. An important property for BRIEF is that each bit feature has a large variance and a mean of 0.5. It becomes more distributed when the bit feature is oriented along the key point direction. To resolve such issues, ORB performs a greedy search to find the high variance and the closest mean to 0.5, making it work well with noise. Below in Figure 22, two images are being compared using ORB to determine the position of the desired image(duck) with respect to the camera by matching its key points with coordinates of images stored in memory. Not only it deals well with noise, but also is more efficient than other feature detector, which is why we chose to use ORB as the object-recognition algorithm for SCUAV.

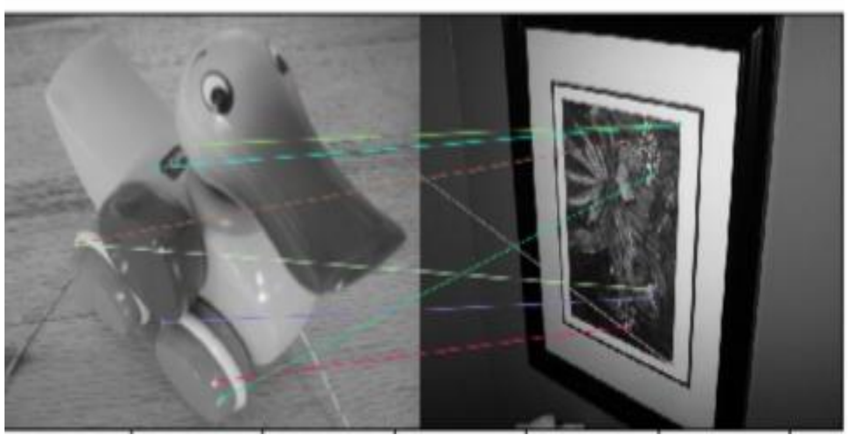


Figure 24: ORB matching best key points

3.2.11.2.10 Quick Response (QR) Code Detection

After much research with other object detection methodologies, it is proven from the OpenMV Cam M7 that color detection and gradient matching performs exceptional. However, color detection is so powerful that it is able to read all objects with certain RGB thresholds. This means, in an outdoor environment, it may difficult to precisely detect and read the target without noise.

Dr. Sukthankar mentioned that placing QR codes on our blocks will be helpful for the precision-based tasks SCUAV will perform. QR codes, as shown in Figure #, are matrix barcodes that contains information about an item. Throughout the development, we figured that having a certain function as the payload of the QR code will aid the drone's autonomous process. For example, we would have QR code #1 to tell the drone to descend and close the claw. QR code #2 would tell the drone to travel the building site. To detect the QR code, edge detection would be used draw a rectangle around the code. Due to fast readability from the camera, simple codes with different functionalities are candidates to our computer vision integration.

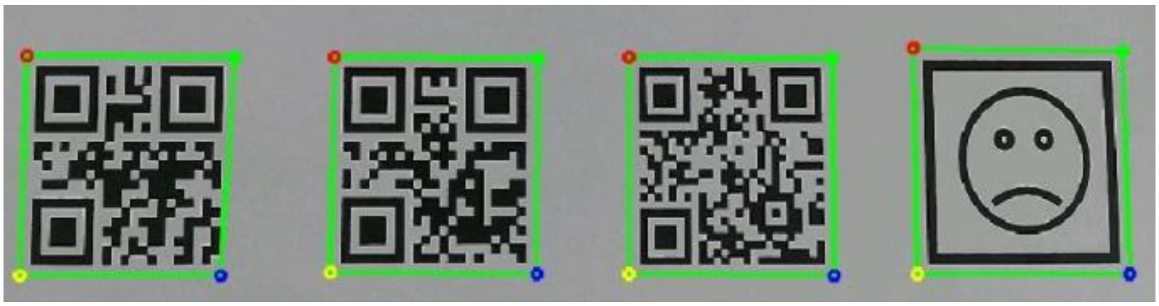


Figure 25: QR codes

3.2.11.2.11 April Tags

After rigorous testing with QR codes, we have found that it is a very efficient method in completing the project. However, we found out that it still may be difficult to detect and maintain due to lighting and the angle of which the camera is detecting it from. Also, we found out later that the QR code is not detectable unless it is completely centered and is only a few inches away from the camera lens, which will cause issues when picking up and stacking blocks.

We then came across the use of AprilTags. AprilTags, as shown in Figure 26, are visual fiducial systems similar to QR codes that can be used for a variety of functions including robotics, camera calibration and more. April-tags are robust to lighting conditions and view angle, which makes it simpler and efficient to complete the building process. Each April-tag belongs to a tag family and encodes an ID number. With the ID number, the camera can differentiate which April-tag to be used in the building process. After many testing later, we concluded that using April Tag detection is the optimal solution for SCUAV's objective.

3.2.11.3 Algorithms

Other algorithms included as a possibility for our computer vision development are Sobel, Canny, and Boost. Choosing an algorithm for SCUAV's computer vision capabilities is essential to complete its goal. The system needs to be able to perform efficiently and accurately. We originally have chosen ORB as our object-recognition algorithm for the project.

If SCUAV wants to track one of the Styrofoam blocks (not all blocks are going to be the same), and it knows what it looks likes, the it can take an image of that block, compute the ORB features and save it to memory. Then with every frame coming through the camera, we can look for ORB features and determine if the key points shown the frame are close enough to the key points stored in memory. Matching them will tell us the position and rotation of that Styrofoam block with respect to the camera.

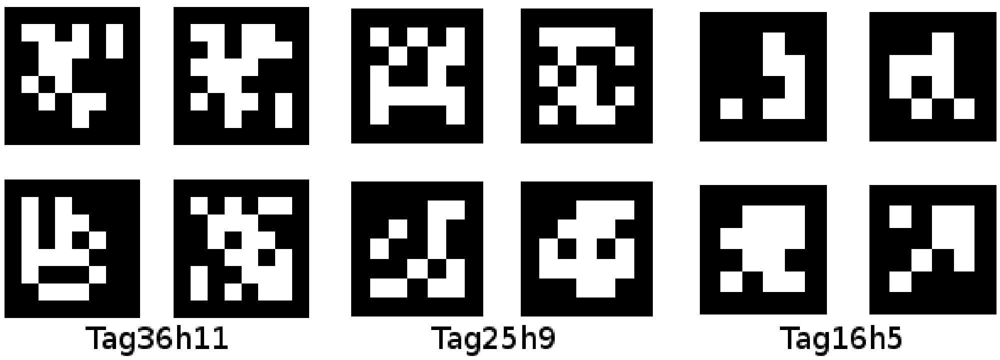


Figure 26: April Tags

Other options for implementing our computer vision algorithms would using color detection. We could use for this implementation would be giving each Styrofoam block a designated color., since not all blocks are going to be used equally. By simply grabbing the block's information by using its color, SCUAV will be able to grab the block and send it to its appropriate location at the construction site. We can also use the Optical Flow application from the camera to help determine how stable the drone is in the air. It can be used to detect translation.

For the construction site, we would probably need to use a grid system in order for the drone recognize the position of where each block would be placed. To develop this grid system, it is possible to use color detection on the grid. For example, we can match the color of the Styrofoam block with the color of each square in the grid. Option 1 shown in Figure 23 would be to divide the grid into quadrants and each quadrant is its own color. We simply place those blocks according to their color.

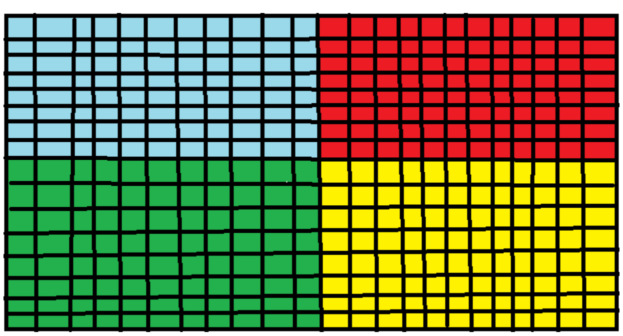


Figure 27: Option 1

Option 2 shown in Figure 24 would be make layers for the grid. Each layer has its own color and we could place these blocks according to these colors and stack them on top of each other. We could start off by having the color blue as the base of the structure, and then red as the next layer, and so on. We could also have the bottom layer having bigger blocks and decrease its size as the structure is near completion. The drone will be tasked to find a different color once a layer is completed. Option 2 seems to be the optimal approach because it guarantees the structure to be safely constructed.

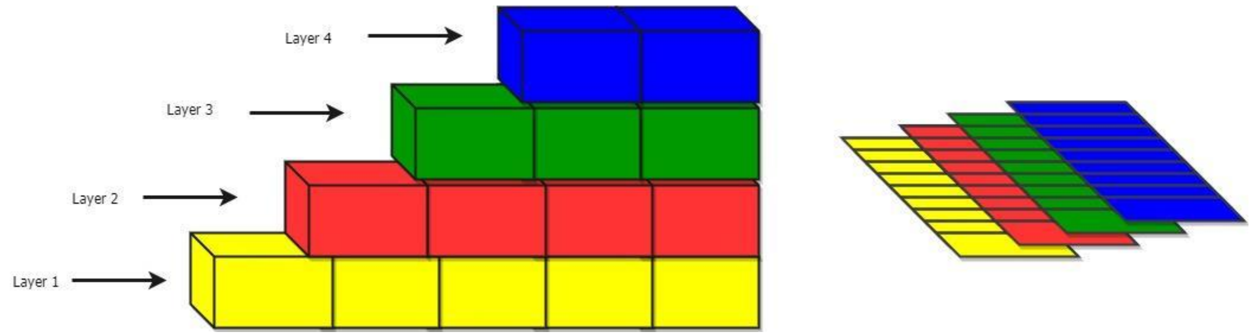


Figure 28: Option 2

After researching more on AprilTags, option 3 is now in play. As mentioned, we can the AprilTag's ID number to determine the drone's phase in construction. AprilTag 1 is the Materials Site. Once the drone is at the Materials Site, it will hover until the camera spots the first block tag. When the correct block is found, the drone will navigate towards it by determining the Euclidean distance using the Pythagorean formula since GPS will lack accuracy. Visual Servoing is also used to center the target's position relative to the drone, it will help with the stabilization and precision in picking up and stacking the blocks. When a tag is detected, the camera will determine whether it is centered or not by sending a character to the on-board computer telling the drone to move left, right, forward, or backwards until the distance of the tag's centroid and the drone is zero. More of the autonomous navigation will be mentioned in the next section.

3.2.12 Autonomous Drone Navigation

The aim for developing and completing SCUAV is provide an autonomous flying hexacopter. Different software and methods can used for automating the pilot. One of the methodologies to automate the process is to use visual servoing. The OpenMV Cam M7 is capable of using visual servoing in order to regulate the position and orientation of a robotic platform relative to a target (April Tag) using a set of visual features such as the perimeter of the rectangle drawn around the tag, the centroid of the object detected. It is mainly intended for the takeoff and landing of the hexacopter as well the stabilization of the target.

Another option to automate the construction process is to use 3D Robotics' DroneKit. DroneKit is mainly programmed in Python that allows the use of communication with the flight controller from the on-board computer. In this case, we would be able to send commands with cardinal directions, specific height it wants to lift, give the drone a customized mission, and more.

The algorithm for the navigation involves having the home location as well as two waypoints based on the materials site and the build site as shown in Figure #. Upon ignition, the drone will acquire its home location with its GPS. In our Python script, a set of GPS coordinates for each waypoint is used for the drone to navigate. When the drone travels to the first waypoint of the mission, it detects a block with an AprilTag. As mentioned earlier, the camera will send data to the Raspberry telling the drone to adjust the offset of the AprilTag until it is in the center of the screen. Once it is centered, the drone will descend and tell the MCU to activate the claw system to grab the block. When the claw is closed, the drone will take off again at one meter and navigate to the next waypoint, placing the block to its appropriate location (AprilTag). The process is then repeated until there are no blocks left. When no blocks are left in waypoint, the drone's vehicle mode will convert to "RTL" which will return to launch.

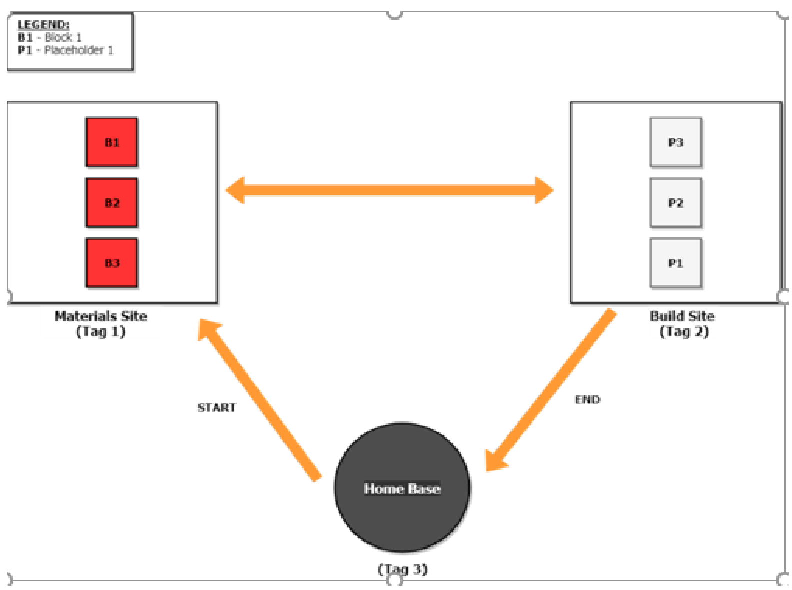


Figure 29: Navigation Path

3.2.13 Base Station

A central location or base of operation would be ideal to both monitor and control the drone, being the main objective of the base station. As the means of communication to the drone, wireless communication would best suit our application. In terms of which wireless technology to use, both Bluetooth and wifi were examined. Bluetooth offers less energy consumption for communication between devices however is limited in speeds and range. Wi-fi-ac offers a larger bandwidth to transmit data as well as a longer range than that of Bluetooth 4.0. With scalability in mind, Bluetooth falls short in regard to multiple drones transmitting data to one central location being the base station. Wi-fi will suit the needs of the project along with future development and scalability.

The base station could act a wireless hotspot that the drone would connect to. As there could be multiple drones used, the base station should be set up as a server and the drone as a client. With the base station acting as a server, it will need to host services to any drones that connect to it as well as the user monitoring the drone(s). With the base station acting as a mobile hotspot, this would allow multiple other drones to connect to the base station for coordinated construction.

Python

Python is a high level general-purpose programming language. It has been used in the robotics industry for a while now and it can be quite useful for this project. Its built in libraries allow for not only network applications but automation. Python can be used to automate terminal line commands. These commands can be used to install software packages or to update the current software. These commands can also be used to configure network settings such as setting a static ip address on a computer or establishing a hotspot to connect to.

3.2.14 Web Interface Application

With a base station in mind, a user interface is required to both monitor and send commands to the drone. There are a few different kinds of user interfaces to choose from. Among this selection could be a mobile application built for Android, iOS, or a web application. With both Android and iOS being the leading consumer markets in phones, having a drone controlled by a phone would mean a simple touch based interface to which would only require a simple download to begin controlling and monitoring the drone. However, developing and testing this would require the use of an iOS device or Android phone. When developing, some issues may not appear on the tested device but bugs could be found when launched as a product. That being said, a web application is also a feasible solution as it would be hardware independent and could run on any device such as a laptop, desktop, android, iPhone or any other computer for that matter with a web browser.

As the current design of the project calls for a base station, designing a user interface around the base station would minimize the complexity of this project. Having a user interface on a local computer hosting a web application allows for not only the local machine of the base station to monitor the drone, but also a server for a client computer to connect and view the current progress.

3.2.14.1 Hypertext Transfer Protocol

The Hypertext Transfer Protocol (HTTP) is the foundation of the World Wide Web. It is an application protocol for information systems which is defined by an international community, The World Wide Web Consortium (W3C), that is responsible for standards concerning the internet. HTTP was originally created in 1989 by Tim Berners-Lee, which set the standard of how information is transmitted on the internet along with how devices will handle such request for transmitting and receiving data.

Figure 30: Ground Control Base Station

Concerning HTTP, there are two positions for communication between two devices. One will act as a client and the other a server. A client begins communication through a request or a command to the server, which the server will return a valid response depending the protocol used. The response given by the server varies but may include the information on a web page, an error code message such as 404 Error when a page is not found, or even a redirection to another page or site. An example of client server communication is a web browser trying to access www.google.com, with an HTTP request to the server associated with that specific domain and the web page information is transmitted back from the server to the client that made the original request.

3.2.14.2 Representational State Transfer

Representational State Transfer (REST) is an architecture style for an Application Protocol Interface (API). An API outlines a set of subroutines and tools to use to build a system as a developer for a user. An API may vary from an email service to a weather application, but often simplifies tasks with a simple input and output format. A REST API often references HTTP as its means of communication between internet systems. REST follows a short set of HTTP commands issued by the client to the server:

- PUT - request to replace existing entry and updates data on server, if entry does not exist it is created
- POST - request to create a new entry to the server
- GET - request to retrieve content from the server

- DELETE - delete all data associated within delete call from client

3.2.14.3 Databases

This project will require the use of a database to keep track of the drone's progress in construction. The database will also store data from the sensors and a profile to authenticate access to control the drone. There are a variety of different types databases all of which store data but offer different functions and store the data differently.

The most common type of database used today would be the Relational Database model. Its approach for managing data with first-order predicate logic structure is as follows. Data is stored in a collection of tables. In these collections, or entities, it follows a similar structure to that of an object in Object Oriented Programming. An entity presents an abstract form of an object, action, or user affiliated with the data. The main component of a relational database is relating data in one table to data in another creating a relation between the two tables or many tables. For example, a single entry in one table record could be correlated with various other tables.

Although not as common as a relational database, a Document Oriented Database is another option to use in this project. It centers around a basic unit of data called a document. A document is an ordered set of key-value pairs. For example, a book id, title, and author associated with a book. Each document is entered in the database with these key-value pairs and grouped by its attributes and fetched according to these properties.

Table 14: Document Oriented Database Entry

BookID	867 - 5- 30986-753 - 0
Title	The Wise Man's Fear
Author	Patrick Rothfuss

With the relational model approach, a relationship between tables would be required and to be categorized by schema, detailing the structure of the database prior. In contrast the document oriented database would not require the structure to be outlined prior to implementation. This would allow data to be entered with less restriction on organization, thus empowering developers with a more agile approach to that of a relational database.

Since we are using the relational database approach, picking an appropriate database system is essential for SCUAV. For this reason, we must choose a system that is very simple and makes ease for data retrieval. Also, the database model should be flexible for meeting changing requirements and increasing amounts of data. Finally, the model must have data integrity, meaning the database must ensure accuracy and consistency of data.

The most popular relational database models used today are Oracle, MySQL, and PostgreSQL. A comparison of these databases are shown below. Factors for choosing the database system relies in the factors of simplicity, security, speed, and memory.

3.2.14.3.1 Oracle

Oracle is one of the most used databases today. Oracle has feasible scalability to work with large amounts of data, and is very powerful for managing code inside the database. Although in most businesses, one of its disadvantages is that it is not open source. Oracle is only compatible in UNIX-like operating systems, and it is a little more complex to get the hang of it.

3.2.14.3.2 PostgreSQL

PostgreSQL is a relational database management system that emphasizes in extensibility and standards compliance. PostgreSQL supports different SQL standards and is very extensible with over 12 programming languages. This system offers the following advantages. PostgreSQL is highly reliable and has very issues with crashing. This relational database model is cross-platform with other operating systems like Windows and any version of UNIX. A disadvantage would be that it cannot store data like an object. Although being open source, PostgreSQL has some learning curve.

3.2.14.3.3 MySQL

We have ultimately chosen to use the MySQL server because of the following advantages. MySQL is known to be the most secure and simple database management system, as it is used of most web based applications. This relational database model offers great scalability to facilitate almost any amount of data, up to as much as 50 million rows at a time. MySQL is thoroughly tested to prevent memory leaks. It also runs on many operating systems such as Windows, Linux, and other UNIX systems. Table 15 below compares all three database systems. MySQL's simplicity is also what makes it an advantage for our database choice in order to complete the project in timely manner.

3.2.14.4 Web Application Backend

This project will require a web application to track and send commands to the drone and structured backend to handle exchanges with the database. This web application will need to monitor, control, and authenticate use of the drone before the drone begins operation. Rather than create two applications for Android or iOS device, we have favored a single web application that will allow access to any device with a web browser which will reduce time in development and testing. It will also require the user, once authenticated, to input what structure to build and state where the drone will find the construction materials as well as where to build.

This can take some time to develop with no prior code, to reduce time in for development, we will examine commonly used backend programming languages for an API such as Node.js, and Ruby on Rails.

Table 15: RDMS Comparison

Database	Oracle	PostgreSQL	MySQL
Operating System	UNIX	UNIX, Windows, Linux	UNIX, Windows, Linux
Open Source	No	Yes	Yes
Scalability	32TB	32TB	8TB
Interfaces	JDBC	JDBC	JDBC, ODBC, Scripting

Node.js
In favor of a server-side platform, Node.js was created in 2009 by Ryan Dahl as an open source, cross platform. With the use of Javascript for event driven interfaces, Node.js has become a milestone in the web development community. It allows Javascript to be run server-side rather than client-side which means the client does not have access to view the backend code and tamper with as with previous Javascript vulnerabilities. Node.js allows for unification of web development in regards to frontend and back-end development having been regarded as two different categories, Node.js stands between the two. Benefits of Node.js include:

- Scalable and dynamic in its applications
- Same language used in back-end and front-end
- Simple to deploy and support
- Single and multi-thread support
- Can be built with REST architecture

Node.js offers excellent use in network programming. However, where Node.js falls short is in applications in need of intense CPU use. Because this programming language is newer, it is not as well document and falls prey to zero day vulnerabilities. Zero day vulnerability refers to a bug or vulnerability not known to the developer and exploited by hackers.

Ruby on Rails
Like Node.js, Ruby on Rails, Rails for short, is a server-side web application that is also open source. Rails is based on Ruby which is used as a general-purpose object-oriented programming language. Ruby itself can be used for web programming, but Ruby on Rails allows for more specialized functionality in network programming. Rails is designed to reduce development of routine tasks such as generating front-end HTML, input from user, and exchanges with a database. Benefits of Rails include:

- Model View Controller software architecture - used in implementing user interfaces
- Debugging tools built in along with
- Configure API to use well known programming styles rather than XML configuration files
- Open source, allows developers to continue the growth and functionality of Ruby on Rails

JavaScript Object Notation
In order for data to be sent between two points over a network it requires some standard on both ends of transmission. JavaScript Object Notation, or JSON, has become very common to use for this standard. JSON is a text format that can be used to exchange data between any programming language. JSON is based on object literals of JavaScript, specifically ECMAScript Language Specification [(ECMA International "The JSON Data Interchange Format" 2013)]. There are a variety of methods to represent numbers, each having its strengths and weaknesses from floats to binary. JSON offers a method to represent these numbers as a sequence of digits. All programming languages can understand a sequence of digits, although their internal processes for this may differ. Through the use of JSON text, text is represented as a sequence of Unicode code points.

Not all languages support objects, and the characteristics that follow with these object's constraints. JSON allows for simple notation of name and value pairs. With JSON offering a smaller data exchange than that of XML, it can be seen that JSON will and has become the standard in exchanges of data on a web-based interface. With JSON offering such a universal tool, other standards will also be developed and derived from JSON being near human readable text.

3.2.15 Bluetooth Application

Originally, users would be to control SCUAV's process through a web application with a relational database designed using React framework in JavaScript and MySQL. However, based on time constraints and the complexity of the project, it was simpler to use the built-in Bluetooth module from the Raspberry-Pi to communicate with a mobile device. The user will then be able to control the drone from his or her phone. Our team downloaded the Bluetooth Electronics applications from the Google Play Store as shown in Figure 31.

The application allows users to connect their phone to another device and control it with any button you can choose from. A Python script is then created to connect our phone to the Raspberry-Pi. The entire program is halted until the user is connected to Bluetooth. Once connected, the Bluetooth dispatcher is executed and integrates the drone's functionalities depending on which button is pressed. The green button will arm the drone, the blue button will disarm the drone, the up-arrow button will take off, the down-arrow button will land the drone, and finally the red button will end the program and the disconnect from Bluetooth.

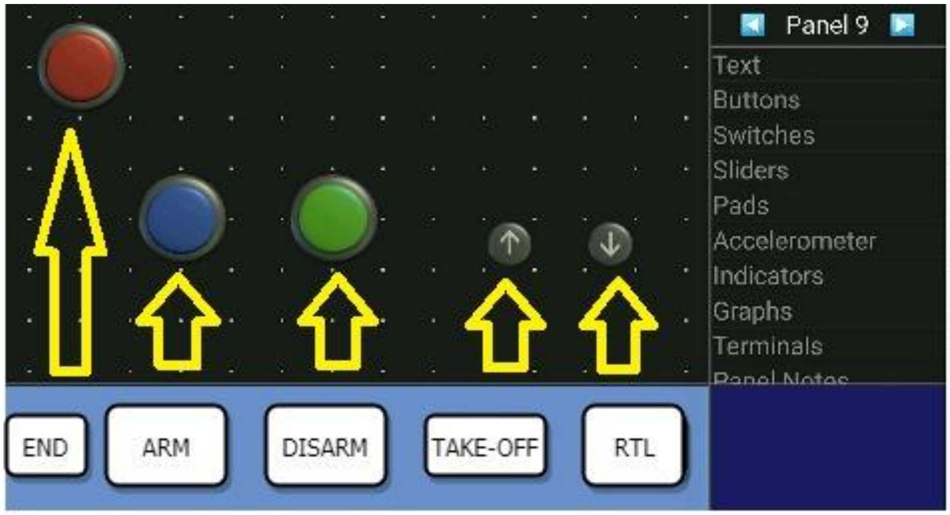


Figure 31: Bluetooth Application

3.2.16 Project Collaboration

Teamwork is vital for the development of SCUAV. Each person has a reason for their major part in the project and everyone must work together to achieve the goal. In order for everyone to remain in sync with each other, several technologies are used to maintain group contact as well as in person meetings. The ways we kept our communication within the group is discussed below.

3.2.14.1 Facebook Messenger

With the project stretching from mid-August of 2017 to late April 2018, an effective means of communication for the project needed to be established among group members. The primary means of communication for this project is Facebook Messenger. It allows group members to quickly communicate with one another, whether over a smartphone or desktop computer. Messenger allows each group member to see if their message has been not only received but read.

3.2.14.2 Skype

As meeting in person may be difficult due to distance or scheduling conflicts, an alternative method of in person meetings needed to be established. Skype offers group sessions, which allows multiple people to voice their thoughts. Skype is used professionally around the world for many business, which made it an ideal candidate for this team’s needs. It allows us to combat the troubles of traffic and distance through a simple solution that can be used anywhere with an internet connect.

3.2.14.3 Google Drive/Docs

Working as a team on a project extending many months can produce a lot of paper and documents that need to be referenced by other members at any given time. A need to quickly and easily share resources among group projects needed to be established through an affordable means. Through the power of Google Drive and

Google Docs, an effective means of collaboration of important documents and resources can be shared in a matter of seconds. This allows for rapid development and documentation at critical moments in a project, especially prior to deadlines. See Table 16.

Table 16: Team Collaboration Software

	Facebook Messenger	**Skype**	**Google Drive/Docs**
Application	Schedule Meetings	Discuss Research and Project Details	Share Files and Collaborate on Documentation

At the end of the day, a good old fashioned calling on the phone works as well incase none of the technologies described above worked. Being on the phone with one another allowed for quicker communication if someone needed to be reached as soon as possible. Being on the phone also allowed for us to make sure we kept each other's company incase someone need some support on writing this document or doing any research.

Parts List and Image

The image and Table 17 below correspond to all of the parts chosen for this project. In Senior Design 2, the parts list changed to suite all the desires need to have a completely working project. All of the parts required are the final products used and they are displayed in the figure below, Figure 32.

Figure 32: Parts List

Table 17: Parts List

Number	Item
1	Drone Frame with Propellers, Motors, and ESCs Connected
2	Taranis Remote Controller
3	Metal Gripper Claw
4	3D Printed Frame
5	Raspberry-Pi 3 Model B
6	Battery for PCB
7	Raspberry Pi PowerPack
8	3000mA and 3300mA Batteries
9	OpenMV Cam M7
10	Ultrasonic Rangefinder
11	Serial to USB Converter
12	Pixhawk Flight Controller
13	PCB
14	External Safety Switch
15	Receiver for the Remote
16	GPS Module

4.0 Related Standards and Design Constraints

Standards help bring forth innovative ideas to be further developed in industry. A set of common terms and methods of testing help in efficient overviews of related products and services. This helps industry bring change with establishing common practices found throughout many private and public organizations. Standards include not only in the physical building requirements but also coding practices. Both establish a well documented and reproducible product that would otherwise be difficult to build upon no prior knowledge with functionality, scalability, and safety in mind seen through most of industry today.

4.1 Constraints

Constraints are the points of stress in any project. They restrict the potential of a project in any number of ways. The greatest challenge in any difficult project is time constraints. It is often the bottleneck in development. It restricts the project to a set of core functions with the possibility of extended features in the future of the project’s lifetime. The economic investment budgeted for a project can greatly impact the outcome of the final product. With, for example, increased prices for shortening delivery time of essential components. Also when comparing possible selections in hardware, a reliable and stable product that is well documented is often worth the investment over a cheaper but not as well documented product. This is favorable as although the budget is diminished greater, it reducing the time in development.

Such is the case in any project that the constraint of time will be affected by decisions made in the research stage as well as design. A project must not only be reproducible to standards in manufacturing, but also conform project constraints in the future such as its sustainability. For example, the expected use in the product or service reliably over a period of time defined as the project's lifetime. Sustainability constraints can be affected by cost as well as increased or extended use of the project. Common practices today incorporate restrictions in a product or service in relation to the effects of the environment. The environment extending to the health and safety of others, and how the project can affect one’s own well being.

With this in mind, a project must adhere to local and federal regulations. This applies not only to health and safety but to in some instances social if not political encounters. New regulations in put into place every year by local and federal governments. These policies constraint not only a product or service but also an entire industry in many cases. Such can be seen in with amateur drone enthusiast, with a restriction of air space and locations seemed not appropriate to fly a drone such as near an airport.

4.1.1 Economic and Time Constraint

In order for any project to succeed it needs to meet deadlines. Deadlines with predefined goals and expectations helps ensure a completed product or service.

There are two main time restrictions concerning this project, one is the deadline for paper submissions this 2017 Fall semester on December 4. The other time constraint is the finished Safe Construction Unmanned Aerial Vehicle project due near the end of the Spring 2017 semester at the University of Central Florida. With the approaching end of the fall semester, finalization of research and general design of the project will be due with approximately a month remaining for further research and design changes. As our team focuses on documentation this semester, development and extensive testing will commence once this finished document is submitted. Senior Design 2 in the Spring semester will give us approximately 4 months to have a finished and working construction drone capable of meeting our specification requirements.

Design Constraints include but are not limited to payload capacity, flight time, and debugging. The drone itself will need to produce enough lift to not only freely escape ground level with its static weight but also carry at least a single object for construction. The drone must also be able to carry the same object it has picked up and carry it to the construction site without any problems.

4.1.2 Manufacturability and Sustainability Constraints

The idea behind manufacturability is having a product that can be easily mass produced. Things to account for in manufacturing is cost, testing, delivery, and lifespan of the product. A goal of this project is to produce an affordable solution to companies and industries interested in a mass producible and sustainable method for construction. Another goal of this project is to eventually have multiple drones collaboration with each other to provide a more practical solution for the construction companies, that would still make this product affordable and easy to maintain.

4.1.3 Environmental, Health and Safety

It is imperative that our product meets the standards for Environmental, Health and Safety. Safety is the most important aspect of this project. The final prototype must have GPS to track the position of the drone in order to meet federal regulations set by the Federal Aviation Administration. The drone must not fly above 400 ft or within a 30-mile radius of Ronald Reagan Washington National Airport. Other safety aspects to consider are the drone must not cause any damage to any structure that is already in its surrounding area, must not cause the life or any extreme injury to humans or animals, and lastly safety signs must be placed to let humans know that there are possible objects flying around. From a health point of view, the drone must not cause any harm to individuals around and it should be regularly maintained to make sure that any chemicals or debris hazardous to the community and the environment.

4.1.4 Ethical, Social, and Political

This project must take into account ethical, social, and political issues regarding not only drones but autonomous robotics. Amateur hobby drone enthusiast over the past few years have made headlines in news articles nation-wide over the past few years. With the price of drones becoming more affordable for the average consumer, politicians and special interests groups have been trying to regulate the use of drones.

This project as well as its members must uphold to the IEEE Code of Ethics. The team members must avoid conflicts of interests whenever possible and to inform affected third parties should it arise. No form of bribery can be accepted as it would conflict with the code of ethics.

4.2 Standards

Common practices used in industry lay the foundation of Standards. Standards helps establish a common set of guidelines allowing a common method of implementation accepted in most regions. These standards allow projects to extend its lifetime with design and development allowing scalability and modular functionality of a product or service. Standards can be found in every industry, with this project focusing on hardware and software.

Software standards in a project can vary. Across the design and implementation of software, depending on the application a predefined protocol outlining the date and time of a web service. This can be necessary as some projects extended over a large region, making references to different time zones a challenge. Within the scope of this project, front end development will abide by common practices in relation to JavaScript Programming. JavaScript standards will be guided by those put in place by the European Computer Manufacturers Association. Not only will this project follow standards set forth by this organization for JavaScript Programming practices, but also into JSON Data Interchange Format.

The standards set in place by JavaScript Object Notation allow flexible means of communication between different but interconnected components. Such as for example, data exchange between a client and server, allowing any form of data to be sent without a prior agreement as to what data type it may be. A global positioning system service will be needed for industrial use of this project. This will require standards put in place by the United States government. Restrictions in this instance focus on communication between the physical GPS module and third party satellites. It can range from frequency restrictions to the type of information sent.

4.2.1 ISO 8601 Standards

When handling dates and times for applications, different time formats can make synchronization difficult. The standard used today, as well as the implementation used in this project will be ISO 8601 for handling date and time information. The date and time will be handled in the API and stored database information. The standard covers formatting for:

- Dates
- Times of day
- UTC - Coordinated Universal Time
- Local times offset to UTC
- Combined date and time
- Time Intervals, and recurring intervals

The objective of this standard was to remove uncertainty with dates and times, especially across different time zones. For example, 02/01/17 could mean February 1, 2017 or January 2, 2017. The ISO 8601 sets the date format standard as:
YYYY-MM-DD
For example, November 16, 2017 would be represented as 2017-11-16, [(International Organization of Standardization "Date and time format - ISO 8601" 2017)].

4.2.2 JavaScript Programming Standards - ECMA-262

As the project will incorporate JavaScript in the front end of the web application, an examination its standards is ideal. The European Computer Manufacturers Association, known as ECMA International, has standardized the JavaScript language as ECMA-262 for implementation. To clarify, ECMAScript is the standardization specifically, and JavaScript is but one implementation of said standard.

ECMAScript is a general-purpose language, however it is used primarily in web applications. It is generally supported by modern web browsers, and focuses on object-oriented programming within a host environment. The specification document for ECMAScript covers the core aspect of the language rather than focusing on specific host environments. The cause for this is with the origin of ECMAScript focused web scripting for use on a “web-based client-server architecture” [(ECMA International "ECMAScript 2016 Language Specification" 2016)]. The document covers built-in functions, abstract operations, and much more with a comprehensive overview of the language. Documentation on built-in functions includes pseudo-code detailing steps to perform tasks. For example, the documents shows the following outline of how to use IsArray(argument) routine if the object is an array:

The abstract operation IsArray takes one argument, and performs the following steps:

1. If Type(argument) is not Object, return false.
2. If argument is an Array exotic object, return true.
3. If argument is a Proxy exotic object, then
    a. If argument.[[ProxyHandler]] is null, throw a TypeError exception.
    b. Let target be argument.[[ProxyTarget]].
    c. Return ?IsArray(target).

4. Otherwise, return false if none of the above applies.

4.2.3 JSON Data Interchange Format - ECMA-404

The standard for JavaScript Object Notation (JSON) is defined in the ECMA-404 specification. “text format that facilitates structured data interchange between all programming languages” [(ECMA International "The JSON Data Interchange Format" 2013)]. Key points that define the standard for JSON include:

- Conformance - JSON text must be Unicode-based characters that strictly complies with JSON grammar
- JSON Text - JSON is primarily a collection of Unicode based tokens, representations of an object
    - Structural tokens: “**[**, **{**, **]**, **}**, **:**, **,**,”
    - Literal tokens: true, false, and null
    - Whitespace character: tab, line

The ECMA-404 specification also outlines how data is interchanged for formatting. A JSON value may be presented as:

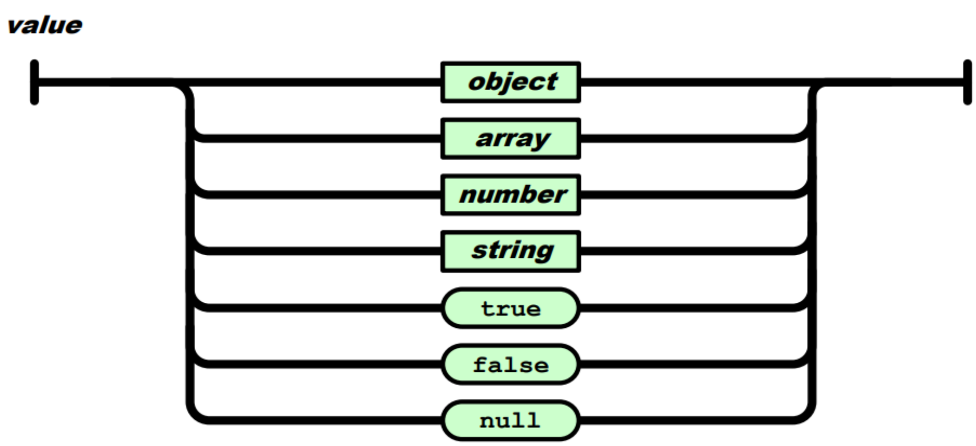


Figure 33: Representation of JSON Values

However a JSON Object would be presented as:

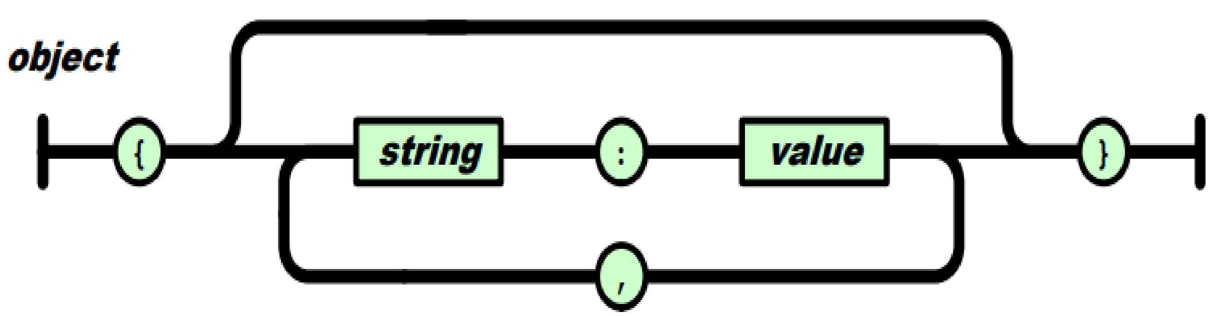


Figure 34: Representation of JSON Object

4.2.4 Global Positioning System Service Performance Standard

The Department of Defense of the United States of America, provides the standards to which global positioning system (GPS) may be used. Although there are other standards of GPS, the one to be implemented in this project shall be that described for civilians' users [(Gps.gov 2008)]. "The GPS Standard Positioning Service (SPS) is defined as follows: The SPS is a positioning and timing service provided by way of ranging signals broadcast at the GPS L1 frequency. The L1 frequency, transmitted by all satellites, contains a coarse/acquisition (C/A) code ranging signal, with a navigation data message, that is available for peaceful civil, commercial, and scientific use"[(Gps.gov 2008)].

An overview of the standard for GPS encompasses two groups, space and control. The space segment covers transmission of data over satellite. It breaks down into detailed steps of how data is relayed on the satellite, in addition to frequency of operation and diverse technologies used. The control segments discuss the Operational Control System and other subsystems. Control system is responsible for:

- "Routine satellite bus and payload status monitoring Satellite maintenance and anomaly resolution
- Management of GPS SIS performance in support of all performance standards (SPS PS and PPS PS)
- NAV message data upload operations as required to sustain performance in accordance with accuracy and integrity performance standards
- Detecting and responding to GPS SIS failures" [(Gps.gov 2008)]

4.2.5 Battery System National Electrical Code

Standard for electrical work in direct current will be required for this project. The standards set forth by the United States government, outlines the practices used throughout the nation. These practices cover the operation, maintenance, and inspection of electrical equipment with regard to ensuring the safety of the workplace. This practices are put into place to avoid injury or legal action with the use of a product or service that may have resulted in an unexpected negative outcome.

An important topic within this standard is battery systems. A battery system must always be considered to be live, being that there is current or stored electrical energy within the system. Insulation for cables is a must as it aids in diminished likelihood of a short in the circuit. Consideration of the placement of a battery in a system must be taken. The battery must have proper ventilation to avoid overheating and operating within a reasonable degree in temperature. The temperature restriction is put into place with extensive testing restricting temperature when the battery may begin to cause damage to itself, shortening of its lifespan or the overall system. These thresholds on temperature ensure safety standards.

5.0 Hardware Initial Design Details

This section corresponds to the building of the hardware components for SCUAV. Below is a thought-out discussion of how all these components will work together and perform the idea behind SCUAV's purpose. Several figures and images are shown to help understand the design process. The design method for this project has changed since the Initial Document.

5.1 Microcontroller Details

When implementing the central control station of the drone's design, each component that corresponds to the hardware design is crucial towards this implementation must cooperate with each other in order to maintain the drone's performance in a safe environment and also reach the goal of the project. The operation of the microcontroller is divided into three segments. Segment 1 is where the microcontroller receives information. In segment 2, the microcontroller decides what to do with the information it has been given. For segment 3, the microcontroller delivers instructions to various components. Figure 29 shows how all the segments collaborate with each other in the old ideal design. The new design is displayed in a later figure.

The pins section of Pixhawk Mini can be seen in Figure 33. All the components that pertains to how the drone works have to be designed or plugged into the flight controller in a specific way. There are options for how these components can be plugged in.

5.1.1 Segment 1 Implementation: Receives

Once the microcontroller is turned on, an initial boot is signaled to all the components plugged into it. The microcontroller receives data from the Raspberry Pi, IMU, and the GPS after it signals that the drone has been turned on. The GPS module would signal the drone's current location. The rangefinder will let the microcontroller know what is going on around the drones environment in terms of distances away from objects. The camera on the hand will have a clear visual of what is actually taking place around the drone. It will report back all that it has seen. The IMU will just be waiting for when the drone is in actual flight to send data to the Raspberry Pi. The figure below, Figure 30, shows a close up view of how the components will be connected to the flight controller. As displayed, the Pixhawk will be placed on top of the drone frame along with the GPS Module, the IMU, and all the other sensors. The IMU is already a part of the Pixhawk on board configuration. The schematics for the IMU integration into the Pixhawk can be seen in Figure 31.

5.1.2 Segment 2 Implementation: Decision Making

When the microcontroller receives all the data from the IMU, Raspberry Pi, and GPS module, it deciphers all the information it is given. From the information given from the IMU, the flight controller will decide on how to keep the drone stable by sending the correct signals to the appropriate components. The kind of information that the IMU supplies is the amount of speed the drone needs to be flying at, the amount of rotational forces around the drone, and keeps track of any changes that the drone might need to do. The flight controller keeps a log of all this information. Based on the information given from the GPS module, the flight controller will determine what to do with the drone's location. The flight controller will either fly the drone to the materials site, the construction site or the base station. The position of the drone will also help to know how far away an object is for the gripper to pick an object and place it on the structure. After the GPS module issues the location signal, the flight controller will now be able to use the signals from the Raspberry Pi to know how far away something is from the drone and the drone makes the decision to either fly higher or lower to avoid, pick up, or place an object. The camera signals all the visual capabilities that the drone sees. Based on these signals flight controller can decide what object it needs to pick up and where to place it using object recognition code.

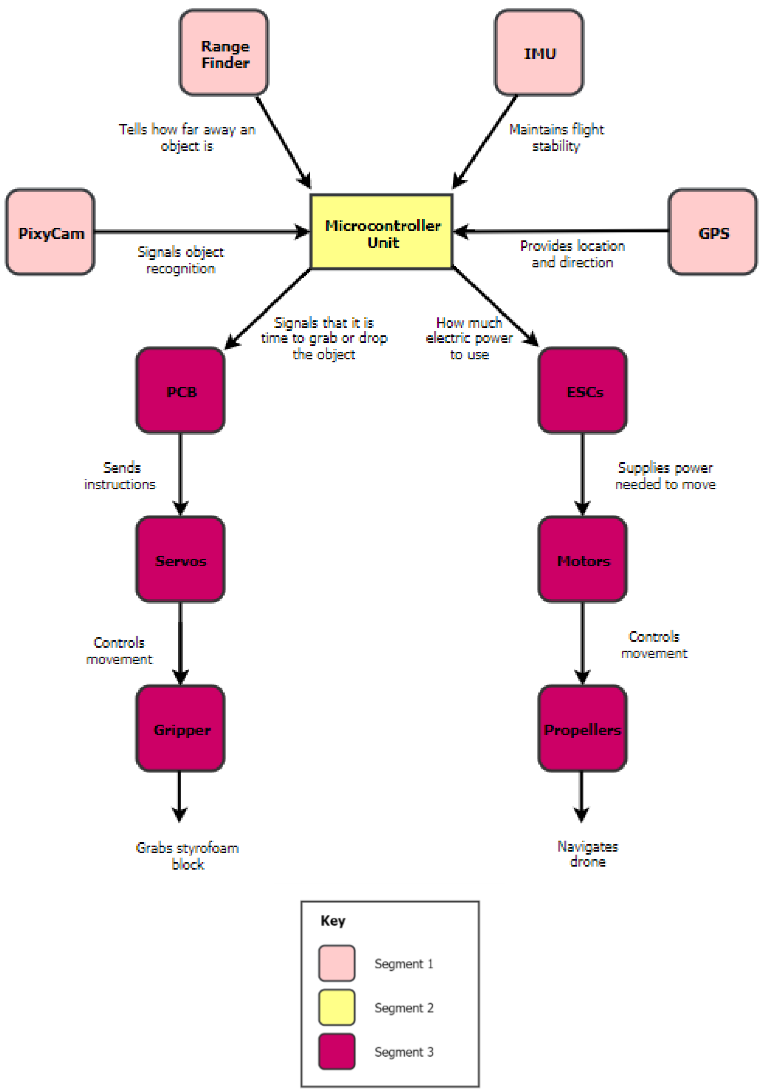


Figure 35: Microcontroller Implementation

5.1.3 Segment 3 Implementation: Deliver

Now that the flight controller has a clear idea on what instructions to give the drone, it translates the information to the components that heavily depend on the flight controller's decision. For the ESCs, the flight controller tells them how much speed to supply the motors which in turn control the propellers. The flight controller has to deliver the correct information to the ESCs because this will allow the drone to fly stable and navigate to its destination. The PCB on the other hand, waits for the Raspberry Pi to tell its microcontroller when it should move the servos attached to the gripper. Based on the message the PCB receives, the gripper would either open and grab the Styrofoam block or release the Styrofoam block onto the structure. The illustration below depicts an example of how both processes work.

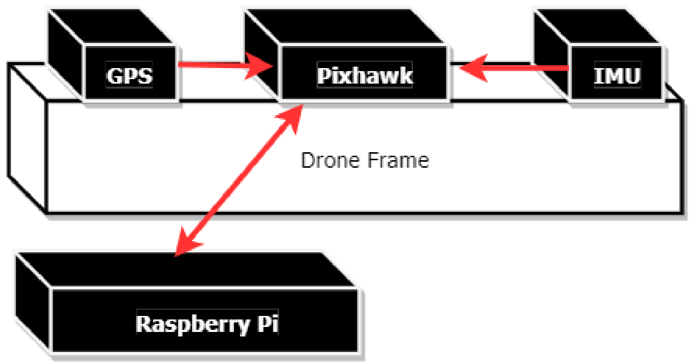


Figure 36: Segment 1 Implementation

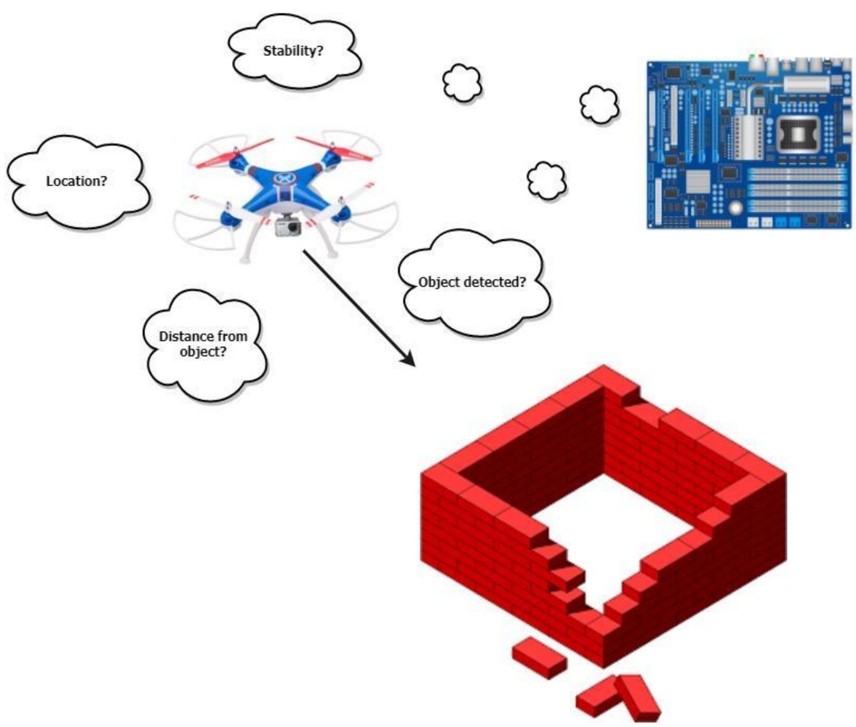


Figure 37: Raspberry Pi Decision Making Process

5.1.4 Flight Controller Pins

The figure below, Figure 33, depicts the possible pins outs for all of the parts essential for the communication aspect of the drone's functionality. Also shown in the figure is electrical wiring inside the ports and which wires go to the correct spot on the flight controller. Lastly, which pins corresponds to each motor is displayed. In the case of SCUAV's design, six pins will be used to connect six motors since the drone is a hexacopter. This pin out chart will be useful for designing the PCB component of this project. There are many possible opportunities for what devices can be plugged into both the PCB, microcontroller, and the flight controller.

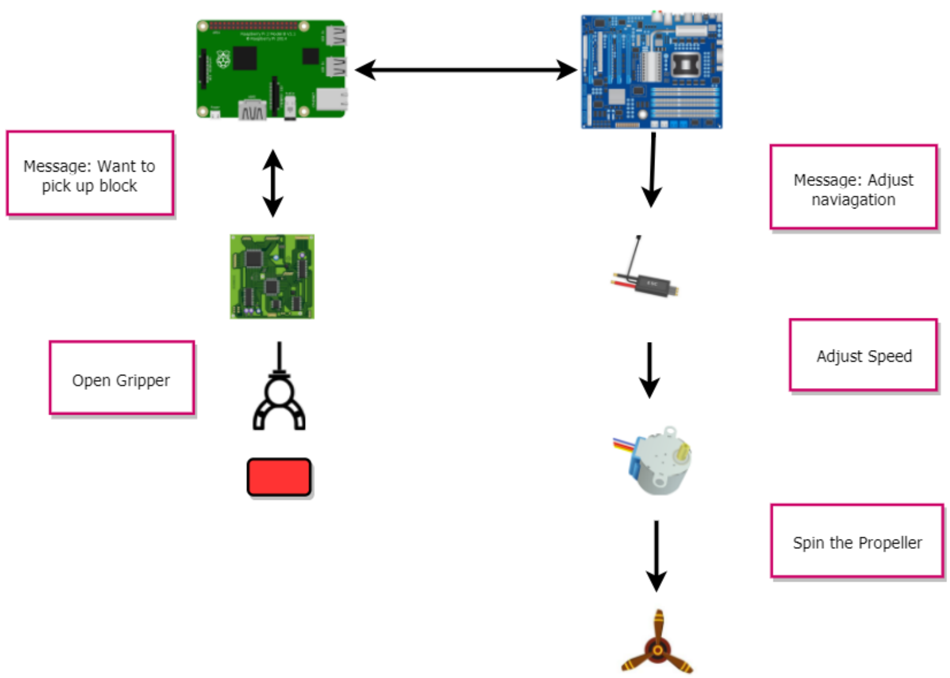


Figure 38: Controllers Delivering Messages

5.2 PCB Design and Schematics

PCBs, or printed circuit boards, are crucial to all powered devices and nearly every electronic product uses at least one PCB. A printed circuit board consists of the electrical interconnections between a device's components. There are many important steps to making a PCB. To summarize the process, a PCB schematic must first be designed according to the needs of a device, then the PCB files can be created in a software tool, these files are then sent to be manufactures, and then finally the PCB components can be mounted onto the board. We decided to research important information for each step of this process, the PCB's design, creation in software, fabrication, and component placement.

5.2.1 PCB Design

When creating a PCB in the software, there are several important considerations. The accuracy and completeness of a PCB's software files will determine its success. The accuracy of the PCB board's files should be inspected before manufacturing the board. For this, the max current and voltage levels of each node in the PCB design should be also be known. Knowing the max input current and voltage rating for each device and the current and voltage levels at the outputs of the devices will help determine the expected levels at each node. Once the expected levels at the input and output of each PCB component is known, the drawn schematic in the PCB software can be reviewed for any electrical errors in the PCB's design. When designing the schematic of the PCB, it is important to consider that some components of the PCB's design may have to be adjusted based on different design constraints such as size, budget, and availability. There are also various other considerations to be made before arriving at the final design. This includes electrical and thermal considerations such as power management.

PIN OUTS

Custom installations may require custom made cables. Here's a handy description of all of Pixhawk Mini's connectors and what they do. Just in case...

TELEM PORT

1 (red)	VCC	+5V
2 (blk)	TX1 (OUT)	+3.3V
3 (blk)	RX1 (IN)	+3.3V
4 (blk)	GND	GND

CAN PORT

1 (red)	VCC	+5V
2 (blk)	CAN-H	+3.3V
3 (blk)	CAN-L	+3.3V
4 (blk)	GND	GND

SAFETY SWITCH PORT

1 (red)	VCC	+3.3V
2 (blk)	IO_LED_SAFETY	GND
3 (blk)	SAFETY	GND

GPS & I2C PORT

1 (red)	SCL	+3.3V
2 (blk)	SDA	+3.3V
3 (blk)	VCC	+5V
4 (blk)	TX3	+3.3V
5 (blk)	RX3	+3.3V
6 (blk)	GND	GND

POWER INPUT PORT

1 (red)	VCC	+5V
2 (blk)	VCC	+5V
3 (blk)	CURRENT	+3.3V
4 (blk)	VOLTAGE	+3.3V
5 (blk)	GND	GND
6 (blk)	GND	GND

CHANNEL PIN OUTS

PIN	Multirotors	4 Channel Planes	Rovers
Pin 1	Motor 1	Aileron	-
Pin 2	Motor 2	Elevator	-
Pin 3	Motor 3	Throttle	Throttle
Pin 4	Motor 4	Rudder	Steering
Pin 5	Motor 5	-	-
Pin 6	Motor 6	-	-
Pin 7	Motor 7	-	-
Pin 8	Motor 8		-

Figure 39: Port Pin Outs for Flight Controller [10]

Another important factor to a successful PCB is its completeness. The PCB design also needs to have enough details in its design for correct manufacturing. For instance, the PCB should have the pin numbers, names, values, and ratings of each component specified. Another important thing to consider when reviewing the PCB is that the number of signal and plane layers in the first half of the PCB stack up should be the same number of signal and plane layers in the second half. Allowing this symmetry will avoid warping during the manufacturing process.

The number of place PCB layers in another thing to consider when designing the PCB. PCBs can be designed to have just one or several layers, they can also be single or double sided with copper layers on each side. Extra layers and double-sided PCB layers allow for more PCB components. And two layer PCBs are the most commonly manufactured as it is an industry standard. We decided that two layers should be sufficient board space for our project application.

5.2.2 PCB Software

PCB software is used to create PCB files, such as Gerber or CAD files, needed to manufacture the board. When looking for PCB software our team found many different options available. Some of the higher end software that are very easy to use but is more expensive. While others are more affordable but have less functionality. It is crucial that the PCB software we select can support the features we need to build our design. Some software features we know will be important are adequate board space, advanced via design ability, and the numbers of layers available for design. Often when a PCB software is less expensive, these important design features will become less adequate. We decided to compare three different types of software, Eagle, Circuit Studio, and OrCad.

Eagle CAD

Eagle CAD is the second most widely used types of PCB software. SparkFun Electronics, Element14, Dangerous Prototypes, and Adafruit are popular websites for electronic DIY tutorials as well as electronic distribution websites. All these popular websites use solely Eagle software. This has helped Eagle have a very strong online community with an abundance of tutorials and videos available. However, within the Eagle CAD software, there are not as good of documentation like videos and tutorials for the software as there is for Altium. The DIY websites not only help make a strong online community, but it has helped make this software extremely popular among hobbyists.

Some other reasons for Eagle’s popularity in the hobbyist community is that it is easy to use and has good features while still being relatively inexpensive. The software cost starts at $15 month or $100 a year for the standard version, and $65 a month or $500 a year for the premium version. Autodesk acquired Cadsoft, the makers of Eagle. When this happened, Eagle changed from a purchasing to a subscription model. Not having the option to own the Eagle is a disadvantage of this software since long time users are not able to own the software. The different versions of Eagle CAD are intended for different users. There is Freeware, Eagle Learn, Eagle Make, and Eagle Business. As expected, the free version of the software has various limitations. It cannot be used for commercial purposes and there is a limit to two schematic sheets and two signal plane layers. Although the Freeware version would not suite our needs, if we were to select this software we would look into the other versions available.

An advantage of using Eagle CAD is that it is very versatile. It can run on many types of operational software such as Windows, Mac, and Linux. Another advantage is that there are many pre-made libraries in Eagle used for different platforms including popular ones such as, Arduino and Arduino shield, BeagleBone, and Raspberry Pi. A disadvantage with Eagle compared to Altium is that it allows each PCB board to have 16 signal layers and 16 plane layers, which is less than Altium provides.

Altium

Altium is the most widely used PCB software. It is the standard PCB software used by professionals, but not as commonly used by hobbyists as cheaper softwares are. Circuit Maker, Circuit Studio, and Altium Designer are the versions of PCB software offered by Altium. These three software types are built on the same technology, but are each targeted at different user. Circuit Maker is for the maker. It has high-end capability and is free to use. Circuit Studio is for the professional, and Altium Designer is for the advanced professional. If our team were to get Altium, Circuit Studio is the version we would want. So in this section we may refer specifically to Circuit Studio when discussing the Altium PCB software.The versions offered at Altium can be bought for an annual subscription or can be bought perpetually. Circuit Studio costs $995 for a perpetual license.

One disadvantage of Altium is that there has not been very much collaboration of Altium within the maker community, which has been around for ten years. Comparing the communities of Altium with Eagle, Altium does not have as strong of a community as Eagle. However, Altium has better documentation and tutorials within its software. Another disadvantage is that unlike Eagle, which can run on many types of operational software, Circuit Studio can only run on Windows.

However, there are numerous advantages to using Circuit Studio. For instance, they have a limit of 16 signal layers and 32 plane layer, which is more than most software provide. It also has one of the most comprehensive sets of features. Like Eagle, Altium has an extensive library. Most component manufacturers offer libraries for Eagle and Altium. And Circuit Studio's component library has over 350,000 parts. Live, integrated supply chain management is another feature of Circuit Studio which helps keep track of what parts you need and which parts are available.

An interview with the Ted Pawela, the chief marketing officer at Altium gave us a lot of insight into the PCB technology used at this company. Ted explained that for many PCB software features available today, Altium was the first company to offer them. For instance, they were the first company to offer rigid flex. They were also the first company with circuit board design software when it released Protel in 1987, Altium's first software. Protel was created to make PCB design more available for users. Prior to its release, PCB software existed only on mainframe and high-end workstations. Altium was also the first company to introduce true 3D board modeling. This tool is useful to visualize a design and can help software users find errors. 3D modeling allows the user to work with 3D design files and understand how the PCB and of its components mate with the PCB's mechanical enclosures. A model for the circuit board can be created and it can be verified whether it fits inside of the enclosure model. The 3D capability of Altium allows the PCB to be rendered real time with the same database that is used for electrical layout.

Altium also has some unique features like a uniform design platform. This means that when a design is created, the release documents associated with it are automatically created. So changing anything in the documentation adjusts the design. Altium syncs the capabilities of the document with what is done in the design. Any changes to either the documentation or the design changes bidirectionally. This consistency between the documentation and design helps avoid many errors. Another great feature of Altium is its software development kit which gives users the ability to create any tools they need. This gives users a lot of flexibility, any feature lacking for their design can be developed. This makes Altium very customizable. Along with this, all the Altium tools are in one unified environment so that they are easily accessible.

OrCad

OrCad is the third most widely used PCB software. Like Altium, OrCad is an industry standard. It is generally used by professionals instead of hobbyists. When comparing Eagle to Altium, which is also an industry standard, Orcad is not as powerful of a tool and does not have the same advanced features. For example, each of the tools in Altium are separated into different applications instead of being integrated into one. This is a major disadvantage of OrCad since it requires users to switch between the different open application tabs to create any PCB board. Along with this, the 3D rendering is not as powerful and Altium’s. The starting price of an Orcad license is $2300.

PCB Software Decision

A commonality between all three of these types of software is having no limit to the number of pins or the number of layers in the PCB design. Other PCB software types limit the number of signal layers and pins. Another common feature is that all three software types offer free versions prior to purchasing the software. This will allow our team to test out a PCB software before making purchases.

However, there are many difference between the types of software which make some more ideal for our design. When comparing the price of the three types of software, Eagle is the economical for short term use, but it could be most expensive for very long term users. And the starting price of a perpetual Altium license is less than Orcad’s. When comparing the ease of use for these three software, Altium has the best user interface and navigation and Eagle comes next. However, it is easier to learn the Eagle PCB software than it is to learn the Altium software. Orcad has the most difficult user interface and navigation and the greatest learning curve. Along with this, Orcad does not have an integrated design platform. Because of these things, our team has decided to rule out Orcad.

Now we are between Altium and Eagle. One difference found when comparing these types of software were the resources available. Eagle has a much stronger only community. Circuit Studio has only been around for about two years and has not has a strong community that has been allowed to develop over the years. However, Altium has better documentation within the software, so more having a strong online community would not pose much of a problem.

Altium is the most commonly used software tool used in industry because it is the most powerful tool available. It has the most advanced features and is very easy to use. Along with this, Altium provides plenty of board space, has advanced via design ability, and plenty of layers available for design. However, because of its price, it is generally not used by hobbyists. Since one of the team members already has access to Altium's Circuit Studio, this software is free for us. Because of this, our team decided on using Circuit Studio [64].

5.2.3 Component Placement

Knowing the different methods available for us to place the components onto the PCB circuit board is very important. We decided to research these methods to help us decide which one gave us the most advantages. The method chosen will be mentioned below.

Surface Mount Technology

Surface mount technology or SMT is a method of component placement where pads are used to connect component leads to copper traces. This technology was introduced in the 1960s and became popular in the late 1980s. Before this, through hole parts were the leading technology for component placement.

Some of the advantages of surface mount technology over through hole parts. For instance, SMT components can be down to one-tenth the weight and one-half to one-third the size of thru-hole components. So SMT allows for a significant reduction in weight and size of components. This reduction is size also enables SMT components to operate at higher speeds. Along with this, SMT components can be soldered onto both sides of a PCB board, as opposed to through-hole technology which can only be soldered to one side. Being able to mount components to both sides, and having a smaller component size allows for a lower component density per unit area using SMT. So SMT is the preferred technology for highly dense circuit boards.

Not only is SMT smaller and has higher speeds of operation, there are also many manufacturing benefits to SMT. Through-hole parts require drilling into the PCB board, and the cost of drilling can sometimes be 30 to 40 percent of the manufacturing cost. While SMT does not require holes to be drilled into the PCB board which allows to PCB to be made more cheaply. Surface mount component can also be placed with automated technology such as pick and place machines, while through-hole technology needs to be soldered manually. Being able to automatically place components is especially useful when there is a large quantity of components to place or boards to manufacture.

Through-hole Technology

Through-hole technology, or THT, is a method of component placement in which holes are drilled into the PCB board and components are placed by inserting component leads into holes and soldering them onto pads on the backside of the PCB board. Until the 1980s when the surface mounting technology became popular, through-hole technology was the standard component placement method.

Surface mount technology has numerous advantages over through-hole, such as being easier and cheaper to manufacture and its smaller size. However, THT is still used in applications, such as those that need high reliability. With surface mount components are merely soldered onto the surface of the board. While with through-hole technology, component leads connect to the copper traces by being inserted directly through the circuit board layer. Because of this, THT has greater reliability and is the preferred method of component placement when a PCB will undergo great environmental stress.

As mentioned in the previous section, manufacturing a PCB board is more expensive with THT since holes need to be drilled into the PCB board. However, through-hole components are less expensive than surface mount components. When comparing manually soldering components using SMT versus using THT, the through-hole method requires less skill and less advanced technology to place components.

Through-hole components are easily soldered through the traditional method of using a soldering iron. However, SMT components are harder to solder and require greater skill when using an iron, or may need to be placed through another method, such as a reflow oven or heat gun, for easier placement. Along with this, with through-hole technology, components can more easily be manually adjusted or replaced. This is a great advantage when a design is in the prototyping phase.

SMT and THT Comparison

SMT allows for higher component operating speeds, lower cost or manufacture, smaller weight, and size which allows for a lower component density per unit area. While THT allows for higher reliability and its components require less advanced skills and technology to place and are more easily adjusted and replaced.

We decided it would be best to use a combination of SMT and THT components in our design. Our first consideration will be availability. If we can only find the exact component we need available as a surface mount or through-hole component, we will go with that technology. SMT can be used for more complex components that require higher efficiency and operation speed. And THT can be used for components whose operation speed is less essential. We decided to use THT along with SMT because surface mount components can be placed more easily.

5.2.4 Soldering Methods

To make connections, one could use the breadboard, but for temporary purposes. To make permanent circuits, the components need to be connected on the PCB. Soldering is necessary for connecting electrical components together when designing the PCB, as they are circuits which should not be disturbed. Components need to be tight. The essentials of soldering and different soldering methods are explained below.

Soldering Iron

This method is used for soldering large solder joints. It is the traditional way of creating connections between components and circuits where a hot iron is used to heat the components and solder is melted between them to make a connection. This is a very efficient way of attaching miniature components onto the PCB or any of the other components if there is a need for it.

Heat Gun

Ordinary heat guns will not work for SMD soldering. The nozzles are too big and they are too hot for SMD components. A disadvantage to this method is that it takes more time. Another disadvantage to this method is it takes more skill to correctly make connection than the hot plate and reflow oven methods. The time and temperature of each of these methods will influence the quality of the component connections. However, the angle of the airflow, speed of air coming from the nozzle, and speed and location of where the nozzle is moved on the circuit board, are factors that come into play when soldering with a heat gun.

Hot Plate

This is another method for soldering components which involves applying solder to all the components at once and then placing the printed circuit board on a heated plate for the solder to melt and create component connections. It is recommended to preheat a hot plate to 212-250 degrees Fahrenheit.

This is practiced for large scale manufacturing and preheating the hot plate avoids damaging the PCB and reduces the time to reflow the solder. An advantage to using the hot plate method is that it has better heat transfer than the oven method. The PCB gets directly heated as opposed to the air around it being heated. Along with this the plastic parts would be protected? Another advantage with this method is that is practical. All that is needed is a source of heat, such as a stove a heating plate, such as a skillet, and a device for measuring surface temperatures such as a temperature gun or a thermocouple.

Reflow Oven

A reflow oven can also be used to solder PCB components. When placed in an oven, the solder paste applied between the component's contact and pad will melt, or reflow. An advantage with this method is that is practical. All that is needed is a conventional oven or toaster oven. A thermocouple or thermometer is also needed if the oven does not display its temperature. A disadvantage to this method is that components have different melting points. Plastic components have low melting point while those with metal housing have higher melting points. The components with low melting points can become overheated and damaged. Along with this, it is often very difficult to see when the PCB components are reflowed and determine when to remove the board.

Soldering Methods Overview

Another option for our team would be to not solder any of the components and have them mounted at a fabrication house. If our PCB is sent to a large-scale manufacturer, our components will likely be placed by a pick and place machine. An advantage of doing this would be that there is less chance of error when mounting the components and possibly having better solder connections if we were to manually do it. Manually soldering the PCB could also take more time than having the process automated since incorrectly soldering components would result in time spent doing rework.

Despite the advantages of having our components automatically updates, our team decided to manually place the components instead. Soldering is an essential skill in engineering and manually soldering components will give our team more experience.

Once we decided we would manually be placing the components, we will decide of the methods mentioned above we will use to place the components. THP and some SMT components have feet and can be soldered with a soldering iron. However, there are other SMT components where the contacts are underneath the device. These require either the use of a heat gun, reflow oven, or hot pad to solder the components.

Leaded Versus Lead-free Solder

Solder acts as the glue to solder components onto the PCB it can either be leaded or leadless. Solder traditionally contained lead and tin. But, once health concerns over the consumer’s use of lead arose, lead-free solder has become much more popular. The health concerns dealing with leaded solder is its primary disadvantage to lead-free solder. When working with solder, especially leaded solder, health measures such as washing hands after use need to be made.

The lead in leaded solder gives it a lower melting point and the presence of tin allows for a stronger solder connection. Because of the higher melting point of leadless solder, the soldering should have a longer application time and be applied at a higher heat than leaded solder. Also, soldering with lead-free solder will require the use of a slightly larger soldering bit because lead-free requires more heat. This is a disadvantage of lead-free solder because there is a maximum amount of heat that components can absorb. Leaded and lead free solder joints will also appear differently when correctly applied. Leaded joints will be shiny and lead-free joints will appear dull. Along with this, lead-free solder is more expensive than leaded solder.

Our team decided to use lead-free solder because of the health concerns involved with leaded solder. Although it is easier to make connections using leaded solder, leadless solder can also form very good connections if applied in the correct way [66].

Flux

Flux comes in many forms, it can come as a core wire, a liquid, or a paste. Before solder paste is applied between the solder pad and component pin, flux should be added to the pad. Flux has many functions. Its main function is to destroy passivation layers on the PCB, these are the layers which make a surface less reactive. Passivation refers to the protection of metal by coating the surface with inert material. So, flux removes these protective layers and allows the circuit board surface to be active and become wet with the applied solder.

A flux that is more active can also remove tougher oxides and in a quicker time. After the flux is applied to a pad, solder is melted between the pad and the pin component. The flux that was applied helps transfer heat within the solder and helps the solder spread out by reducing surface tension. It is best to obtain a flux which can withstand high temperatures. If it cannot, flux charing can occur and the flux will blacken the tip of the soldering iron, making the iron less effective at transferring heat. There are many options and consideration for different types of solder paste and using the right paste can allow for less rework.

Because of the corrosive nature of flux, the residue from fluxes should be cleaned after the soldering is complete so that none is left behind. Another reason to remove the flux is that is can be conductive. No-clean fluxes can be used when soldering, these fluxes claim to have no cleaning necessary. however, it is recommended to still remove these. An option when selecting a flux is to get a no-clean flux which would not require cleaning of any flux residue. However, these fluxes sometimes burn-off before a solder joint is complete. If the flux gets burned off it is not able to help with the solder connection. The burning up of the no-clean flux often occurs with lead-free solder since it requires more heat.

5.3 Component Connections

The sections below applies to how the major components or devices that need to be connected to the microcontroller will be designed. Several schematic layouts or diagram explanations can be seen in the sections below. It is vital that all of these components are planned out thoroughly. The initial plans from Senior Design 1 has changed. Problems several problems have occurred.

5.3.1 Camera and Flight Controller Connection

For the camera and flight controller connection, there are two possible options. These options are: connection via Raspberry Pi and then to the microcontroller or a direct connection to the microcontroller. A detailed explanation on both options is discussed below. Figure 34 provides a schematic view of how the OpenmV m7 camera works.

Option 1: Direct Connection to Microcontroller
For this option, the OpenmV camera will be plugged into the I2C port on the flight controller. The I2C splicer might have to be implemented in the hardware setup incase other components need to use the same port. One of the advantages of the OpenmV cam is the camera's ability to be program via the computer to have the computer vision capabilities that SCUAV needs.

The camera does not need to be plugged into the Raspberry Pi or any other device in order to communicate with the microcontroller. Allowing the microcontroller to have direct access to the camera will make it easier for drone navigation and for allowing the structure to be built more accurately. The Microcontroller Details section explains in more details as to how this collaboration works.

Option 2: Camera to Raspberry to Microcontroller
If this option is chosen instead, then the Raspberry Pi will directly be communicating with the flight controller. It will use TELEM port on the Pixhawk Mini instead of what is shown in Figure 35. This integration will require software that is compatible with the Pixhawk and it can be found on one of the user guides for Pixhawk [67]. One way to implement this is to use the MAVLink. An advantage of this option is the Raspberry Pi will be able to gather all information that is needed to be stored and then transferred over to another Raspberry Pi located on the base station. If this option is not chosen, the same information still needs to be transferred to the base station. To solve this issue, Raspberry Pi can be connected to the microcontroller via the I2C port. The Raspberry Pi will still need to be configured to work with the Pixhawk.

5.3.1a Updated Camera Connectivity
Since multiple testing procedures have been performed, a realization was made about the camera and flight controller connection. The camera does not directly connect to the flight controller. A lot of difficulties were reached with this connection. One difficulty was the I2C configuration. The OpenMV m7 Camera does not act as slave to any device. As mentioned in the Camera selection above, this camera performs poorly with this type of communication. The new design to connect this device is to use the Raspberry Pi. Figure # shows the new connectivity.

5.3.2 Rangefinder and Flight Controller Connection
For the rangefinder and the flight controller connection also has two possible options. Option one is to connect it to the microcontroller directly and option two is completely integrate the connection to the PCB. Option two would only be considered if there is not enough ports available or if it is more logical to place it there. There is software available for the installation process and for allowing any modifications to suit SCUAV's design. All of the information for the software

configuration can be found in the LeddarOne user guide [22]. Figure 35 shows the architectural design of the rangefinder that will be used.

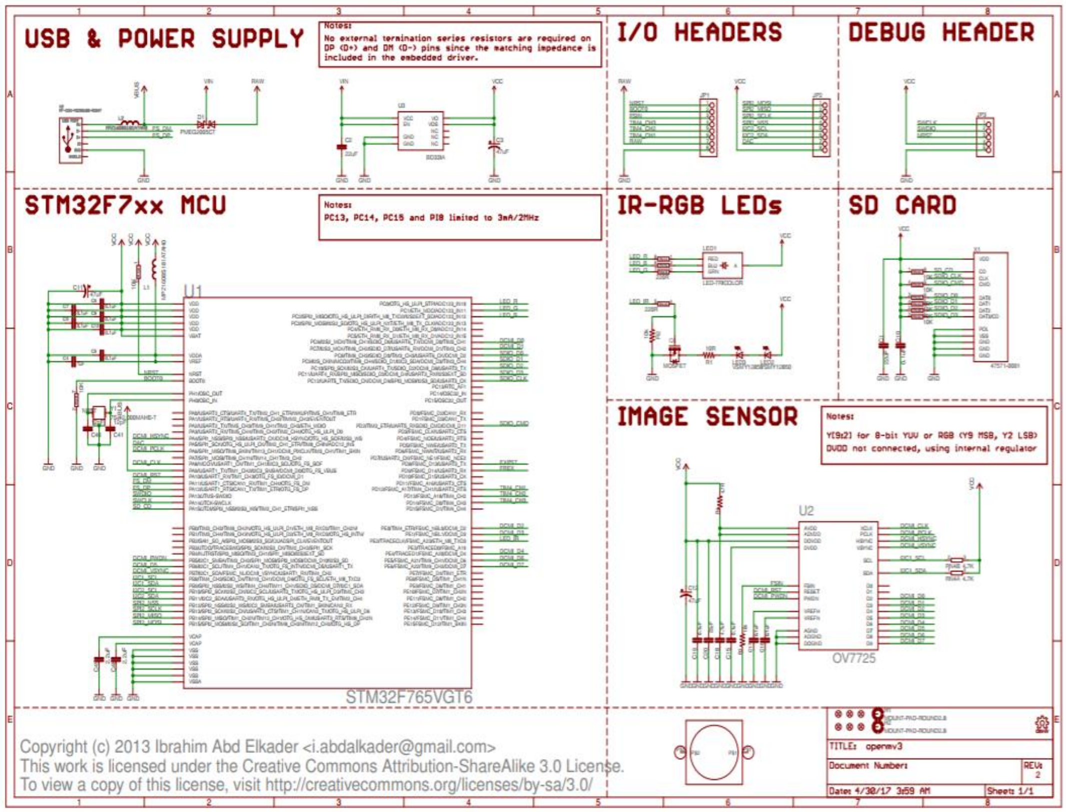


Figure 40: OpenMV M7 Camera Schematics [34]

Option 1: Connection Directly to MCU

The way the microcontroller will be connected to the Leddar One rangefinder will be via the TELEM port which is a Serial Port. Figure 30 displays another possible way but the Pixhawk Mini only has one UART serial port which is the TELEM port. If this option is chosen then the rangefinder will be attached to the bottom of the drone's frame or close to the gripper but in a spot where the sensor will be able to see its surrounding area. This will allow for the microcontroller to receive all the sensory information that the LeddarOne sees. This information will be important to report to the PCB so that the gripper will be able to pick up the styrofoam blocks and deliver them to the building site. Segment one of the microcontroller implementation explains in detail how this whole process works. This option will be used in SCUAV's design.

Option 2: Connection to PCB

If this option is chosen, then the PCB will have to be designed to accommodate the rangefinder's pin out configuration. The required pins can be seen in Figure 37. Once the LeddarOne is plugged into the PCB, it will directly send all the information to the PCB and the PCB will use it to directly control the gripper's functionality. Since the rangefinder is connected to the PCB, the PCB will have to communicate to the microcontroller all the sensory information that it receives. The microcontroller will need this information in order to avoid obstacles and also to see how far away the drone is from the material site, the build site, and the base station.

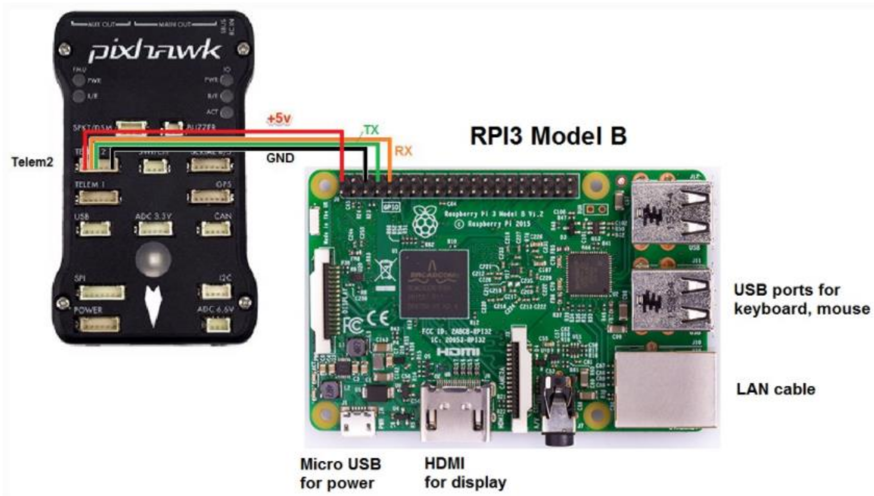


Figure 41: Port Configuration for Raspberry Pi and Pixhawk [67]

5.3.2a Updated Rangefinder Connection

Above talks about the initial design for how the flight controller was going to be the original brain of the system but this is not the case anymore. The rangefinder will be connected to the PCB unit this time. It works well on the PCB to be programmed. It needs to work for the claw functionality. Figure # shows where this device will be located on the drone and figure # shows the communication process.

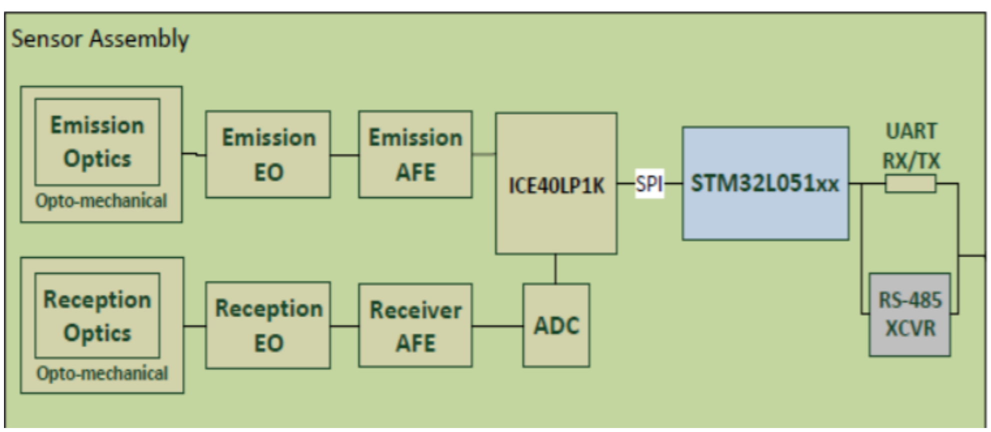


Figure 42: Architecture Layout of LeddarOne Rangefinder

5.3.3 PCB and Flight Controller Connection

For the PCB and the flight controller connection, some of the information signaled to the microcontroller will be also sent to the PCB so that the PCB will operate the gripper part of SCUAV's design. How all of this works is by first the microcontroller will receive information on if the drone is stable from IMU and where the Styrofoam block is located. The drone will fly over to the location of the block and flight stability must be checked. The drone must be mostly stable so that when the PCB gives the command to open the gripper, the drone will not wobble or tilt to the side causing the drone to crash. Once the microcontroller receives the signal that everything is fine, it will then signal the PCB to open the gripper. The drone will then decide if it is close enough to grab the block and then the PCB will signal the gripper to close its grips around the block. Flight stability will be checked again to make sure that the drone is not tilting to the side and that the block can be safely carried to the site where the structure is being built. From this point, the microcontroller will signal to the PCB that the building site has be reached and now the camera will report where the block needs to be placed specifically. Once the command has been passed on, the PCB will signal the gripper to open again and allow for the block to be released to the desired spot. The drone will then proceed to lift a little allowing for the PCB to signal the gripper to close. This entire process is repeated until the structure is fully completed or there is a termination in the drone's process. Figure 42 shows an example of the microcontroller sending a message to the PCB to pick up a block.

5.3.3a Updated PCB Connection

The PCB is now connected to the Raspberry Pi and it controls all the programming for the claw and rangefinder. This definitely worked out better than the original design. The PCB section goes into more details as to how everything is connected. Detailed figures can be seen in the images that pertains to the PCB and all of it component connections.

5.3.4 ESCs, Motors, and Propeller Connections

In this section, the ESCs, motors and propellers all need to be connected correctly because they are responsible for the drone's ability to fly. As described earlier the ESCs will be directly connected to the microcontroller, then the motors will connect to them. Finally the propellers will be the final connection made and the most visible one seen on the drone. Since SCUAV is a hexacopter six of each component will be needed.

ESCs to Motors

Figure 36 shows an example of a brushless motor being connected to an electronic speed controller. Instead of using a radio receiver, the PWM port on the Pixhawk Mini will be used. The Pixhawk has software that can be used to calibrate the ESCs and to make sure they are functioning correctly. The ESC controls the motor by signaling the amount of power needed for the motor to spin.

Motors to Propellers

Once the ESC and motor connections have been established, the propeller is attached. Before the propeller is attached, the propeller must be balanced. An important reason for centering the propeller is to avoid any unnecessary vibrations that could happen once the flight takes place. These vibrations could send false information back to the microcontroller. Now the propeller has received the instruction from the microcontroller and the navigation can begin.

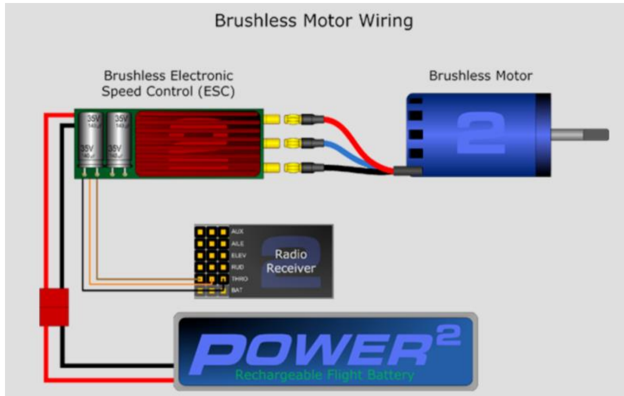


Figure 43: Example of How an ESC is Connected [69]

5.4 SCUAV Hardware Layout

The hardware layout section is comprised of possible design ideas for how all the components can be connected on the drone’s frame. So far only two designs have been made but they can be changed or more ideas will be added to this section in the future. Figure 41 and Figure 42 pertain to these designs.

Interface pinouts

1	2	3	4	5	6
GND	IRQ	5 V	RX or RS-485+	TX or RS-485-	RESET_N

Legend:
RX/TX = 3.3 UART
RS-485+ / RS-485- = RS-485

The power and interface signals are included on a six-pin standard 0.1" pitch header. The RS-485 option, in combination with MODBUS, makes it easy to integrate multiple sensors with an RS-485 network.

Figure 44: LeddarOne Pin Out [67]

Figure 40 gives an example as to what the IMU in the Pixhawk is designed to look. The only difference is the Mini has a smaller version of this integration and the sensors are updated. The sensors the Pixhawk Mini uses are MPU9250 (Accel/Gyro/Mag), ICM 20608 (Accel/Gyro), MS5611 (Baro), and HMC5983 (Compass).

Figure 45: Motor and Propeller Connection [70]

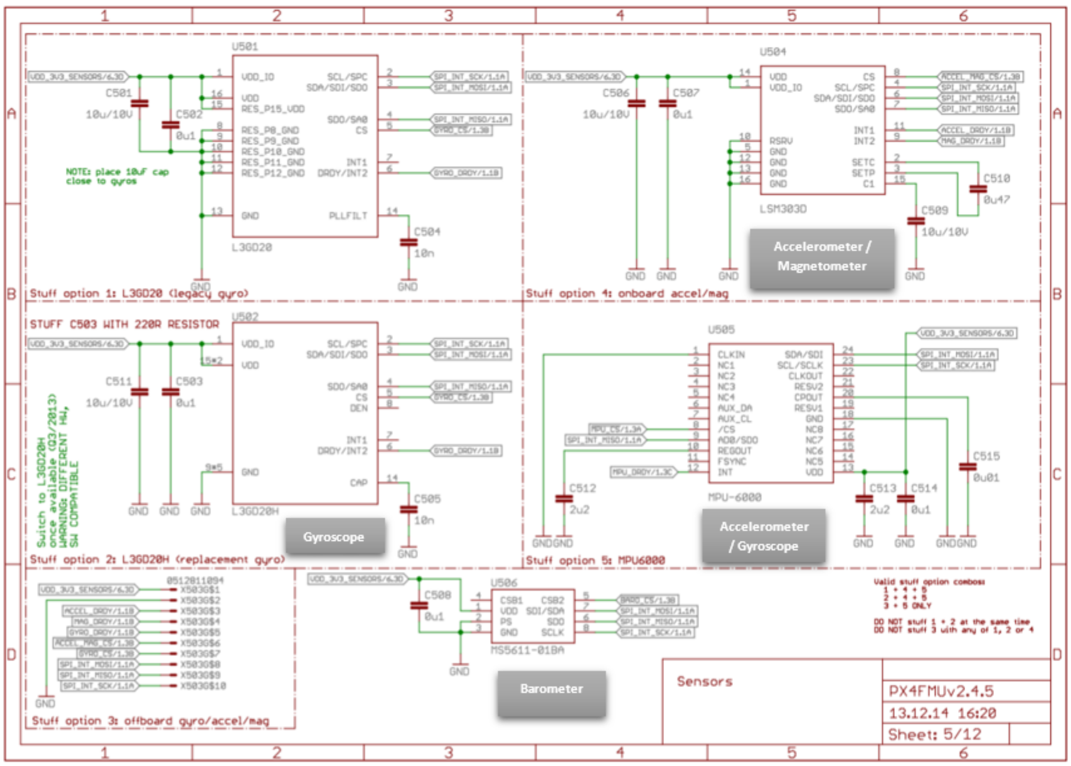


Figure 46: IMU Layout on Pixhawk [71]

Design 1

The diagram below shows a possible overall design for the drone's hardware layout. This design is subjected to change once the overall completion of the project takes place. In this design Option 1 from the Camera to Flight Controller Connection will be shown and Option 1 from the Rangefinder section. All six ESCs, motors, and propellers are connected in order into the PWM Breakout board. The ESCs connect directly to the board, then the motors connect to the ESCs, and lastly the propellers connect to the motors. All of these rely on the instructions from the microcontroller. Notice that the Pixhawk Mini is connected to the battery chosen for this project and that battery is plugged into the Power Distribution

Board. Hopefully the GPS Module works well plugged into the I2C Splitter. The PCB board is also plugged into the I2C Splitter and the goal is to have it compatible to work for this port. The external safety switch is plugged into the Switch port. This is works for when the drone is in need of an emergency landing or there needs to be a major stop of all operations for some abrupt reason.

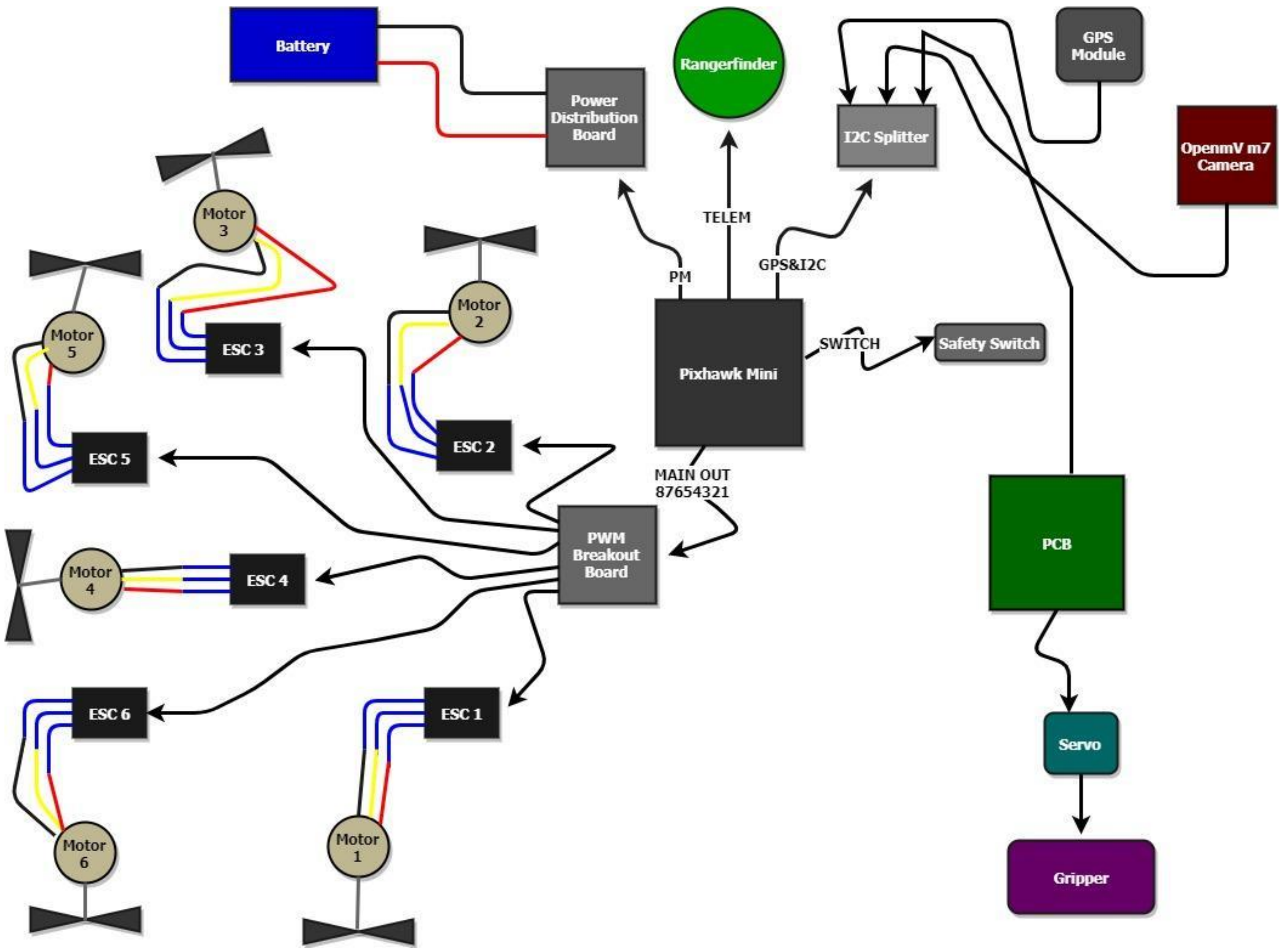


Figure 47: Design 1 Layout

Design 2

For this design Option 2 from both the Camera and Rangefinder Connection setups will be chosen. This will demonstrate that these options are also a potential possibility and will be determined if this design should be implemented instead of Design 1. Design 2 might be a little more elaborate than Design 1 but Design 1 is much easier to implement. All six ESCs, motors, and propellers are connected in order into the PWM Breakout board. The ESCs connect directly to the board, then the motors connect to the ESCs, and lastly the propellers connect to the motors. All of these rely on the instructions from the microcontroller. Notice that the Pixhawk Mini is connected to the battery chosen for this project and that battery is plugged into the Power Distribution Board. Hopefully the GPS Module works well plugged into the I2C Splitter. The PCB board is also plugged into the I2C Splitter and the goal is to have it compatible to work for this port. The external safety switch is plugged into the Switch port. This is works for when the drone is in need of an emergency landing or there needs to be a major stop of all operations for some abrupt reason. Notice the Raspberry Pi is plugged into the I2C Splitter as well. Attached to the PCB is the servo and the metal gripper chosen for this project.

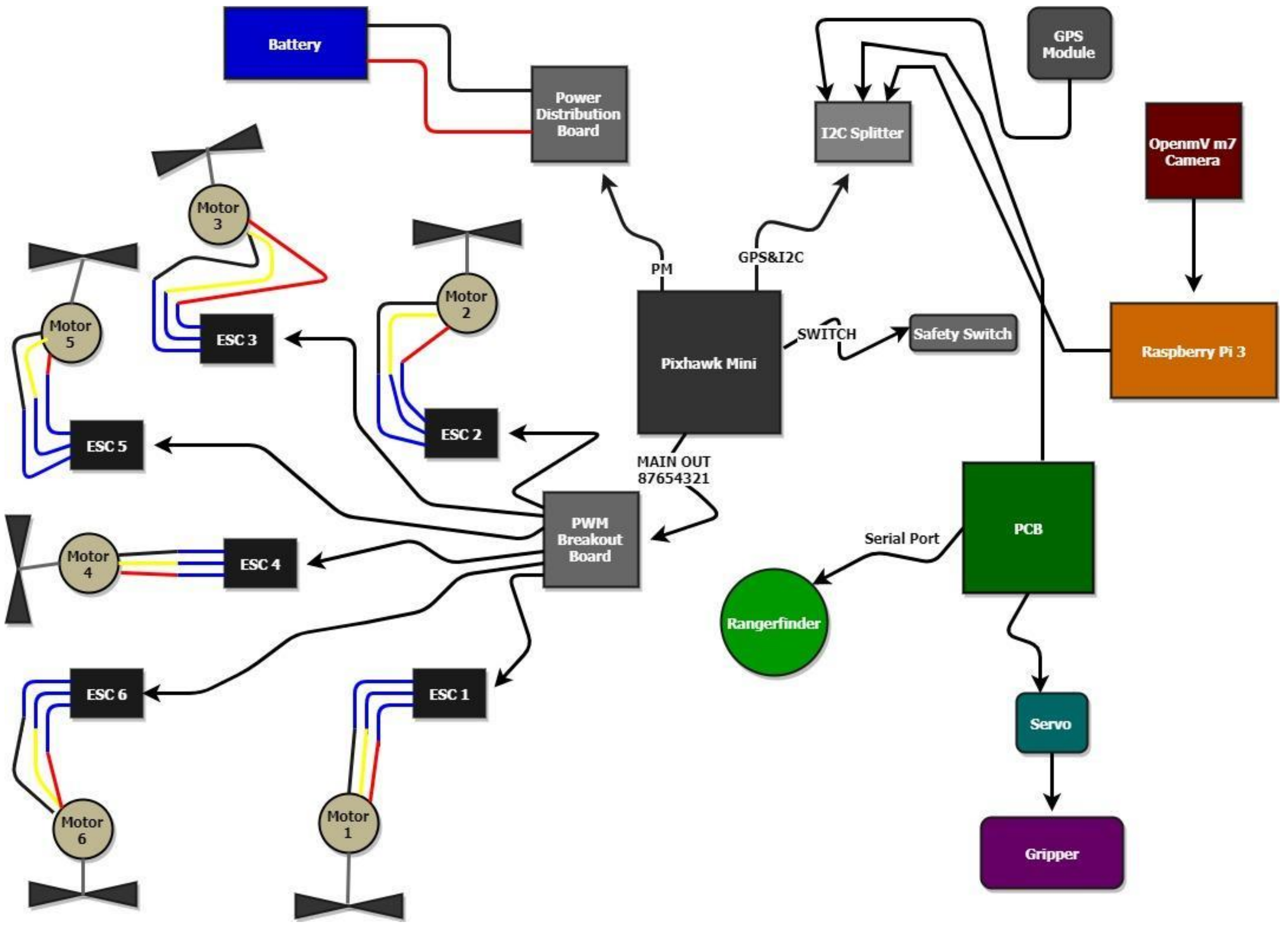


Figure 48: Design 2 Layout

The previous two designs have changed to suit the final layout of all of the hardware layout of the drone. It was tested to be successful although the multiple test issues occured with other problems like battery powering issues. Below depicts the new design of the final layout.

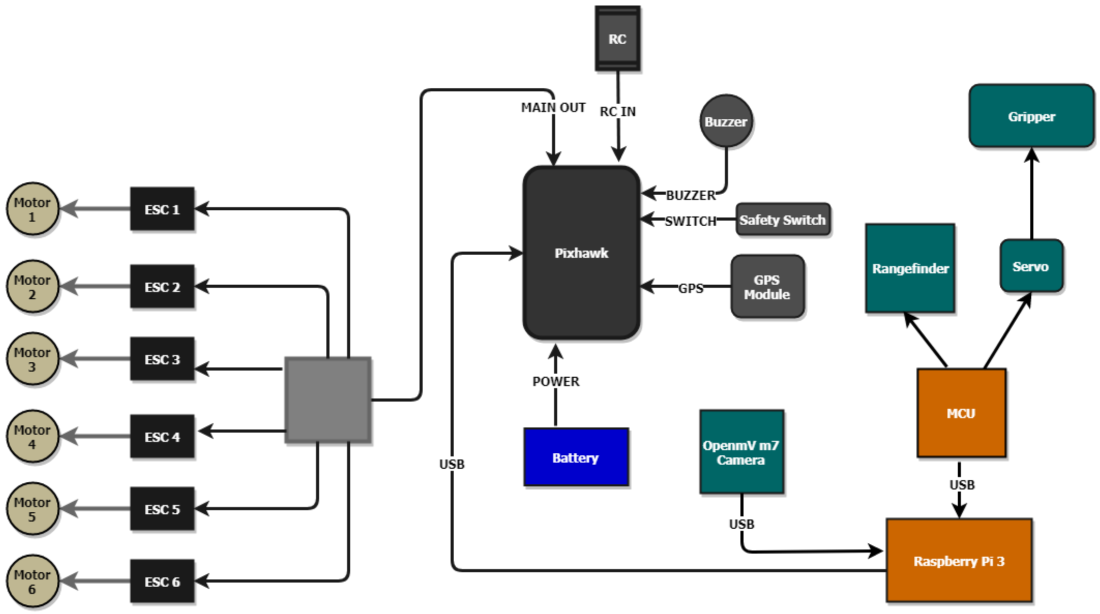


Figure 49: Final Design

6.0 Software Initial Development

This section explains the details in designing SCUAV's software system. This includes the software requirement specifications, the software design description, and finally the software and system test plan. Building good software as well as testing its performance and accuracy is essential for SCUAV to accomplish its goal.

6.1 Software Design Details

The software behind the Safe Unmanned Aerial Vehicle project will center around modularity and ease of use. It will be accomplished through development of an application programming interface for both frontend and backend use during the lifespan of the project. Through a web based user interface, the user will be able to control simple functions of a drone such as landing, returning to base, and beginning operations for construction. During future development, the API can be extended to include more functions such estimated time of completion, update of software on drone over wifi, and detecting different objects based on color and other physical features.

Figure 43 below shows the flowchart of the overall web application. The user interface will retrieve information and connect with the Web API. That is where the on-board computer will send its data from the flight controller and camera. This data includes object recognition, camera view, flight controller commands, drone position, construction phase, etc. The Web API will connect to a database which contains tables for the user information, other sensor information and construction information such as material placement and construction progress. In the next chapter we will describe the actual software test plan.

In the web application, the software team will design the frontend where users shall be able to view the route of where the drone is navigating from the materials site and base station to the construction site. Also mentioned in requirement specification in the later section, the web interface shall display options in which users can manipulate how the structure should be designed, the control system of the drone, the construction phase development and more. The interface will also display all construction information including the amount of blocks remaining, the estimated and actual time of completion, what block is chosen to pick up, the success/fail flag, etc.

6.2 Software Requirements Specifications

For SCUAV, the applicable standards for the software are as follow. The computer vision standard shall be implemented in Python for the sake of simplicity and its availability in libraries. The Raspberry-Pi on-board computer will also utilize Python for reading the data from the camera and sending that data to the base station server, thus will follow all Python related standards.

In the web interface application, the front end will feature some sort of JavaScript and will follow JavaScript standards in order to send commands to the drone. The Node.js standard will be used in the back end in order to set up the server and help with the exchanges in the databases. The SQL standards will be used to store construction information as well as user credentials and sensor data.

Assumptions will be needed to properly develop the web interface. These assumptions are as follow. One assumption is that all computer vision development is built in the OpenMV IDE, where all statistical models and image information and shall be transferred over to both the on board computer and the MCU. We will also assume that the client will use JSON to communicate with the server. The stakeholders in this project are the construction managers and building enthusiasts, so the interface will be very user friendly.

Below on Figure 43 is a use case diagram to describe the user (actor) action based on a set of events. In this case, the user is the construction manager and building enthusiasts. In the web interface, the user is able to login. The user is also capable of controlling when the drone should start or continue construction. Other events include that the user is capable toggling the drone’s navigation capability to either automatic or manual. If there is an issue within the drone or in the construction process, the user has the option to either have drone perform an emergency landing or to return the base station. Finally the user has the ability to design the structure and send it to the server.

For the computer vision part, the system should be able to detect the colors of the blocks it is seeing. The system should be able to create a statistical model of what object it is seeing and what color. The system should also be able to send the data found in the camera to the on board computer Raspberry-Pi. Code will be properly developed with proper indentation and comments for other developers to understand. All live feed found on the camera should appear in the web interface.

6.3 Software and System Test Plan

To properly test SCUAV’s software, we will implement code that will make sure the software will fail. This will help us understand its current performance and accuracy. By testing our system, this will ensure that the drone is capable detecting the correct color block and place them correctly. To test this, we must “teach” the camera how to understand colors and objects.

According to the OpenMV website, one can teach the camera what object it is seeing by simply showing the object in front of the camera. The camera will then find the images to be false and eventually will learn. It is necessary to make the camera can detect and pick up the correct color block in order to prevent structure from collapsing.

For the web application interface, the system has two distinct components to test, and as such will require multiple testing environments and strategies. The client components of the software will be tested on the engineer's computer, by the software team themselves. This will differ from using different platforms. The code functionality should not change.

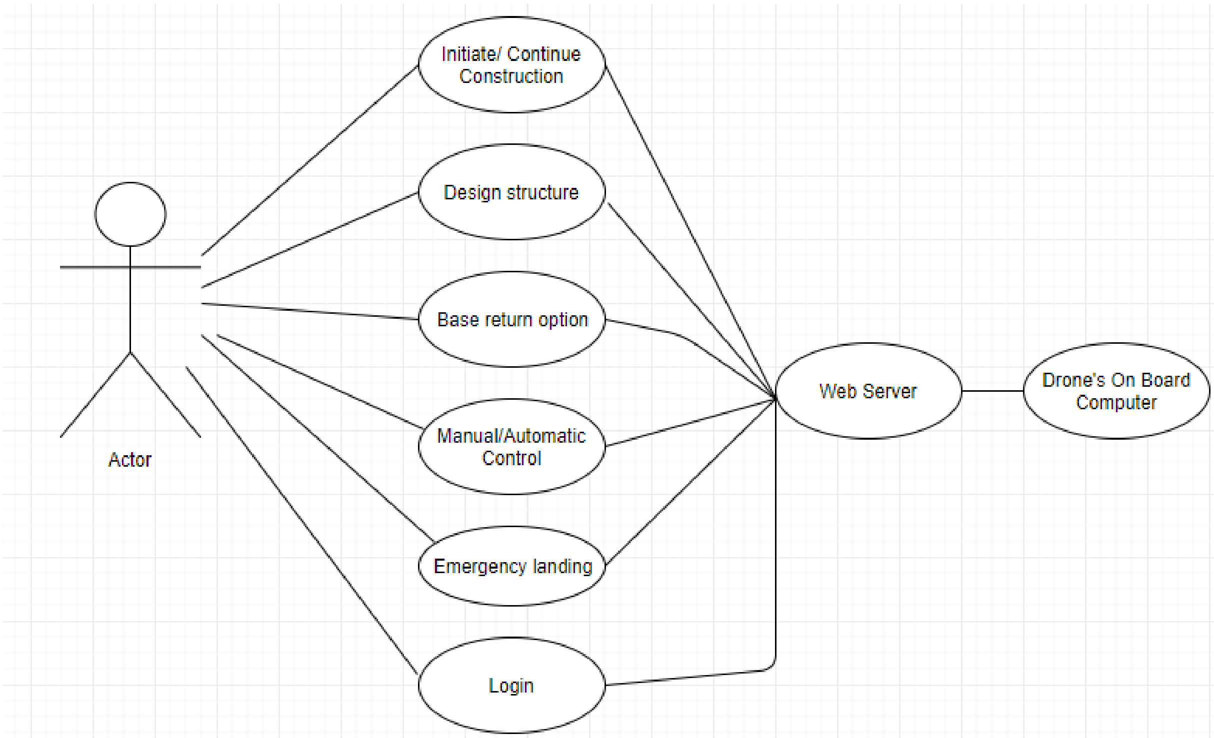


Figure 50: Use Case Diagram

The web server components will be tested on the engineer's computer and a user's computer to run the production code once the system is ready. This environment will most likely be similar to how the system will function in the wild, thus ideal.

The testing period will be halted for a particular module when:

- The module fails to compile.
- A fault that causes data corruption or crashing is found.
- A fault that interrupts the remaining test cases found.
- A fault that prevents the drone from seeing anything is found.

If no errors are encountered during testing, it does not mean that the program is running perfectly fine and free of errors. However, it means that it is sufficiently functional for user trial and eventual release.

Figure 44 above outlines the web application design. From a user standpoint, it begins with the front-end. Front end being defined as what a user sees and interacts with. Common front-ends that users interact with today are built with HTML, CSS, and JavaScript. HTML is what the world wide web began with, simple text with no fancy features seen today. CSS can be seen as the formatting and different properties of a page. JavaScript helps bring in interactive features to a page. For example, JavaScript can animate portions of a web page to dynamically change the perspective of the page, without loading another file.

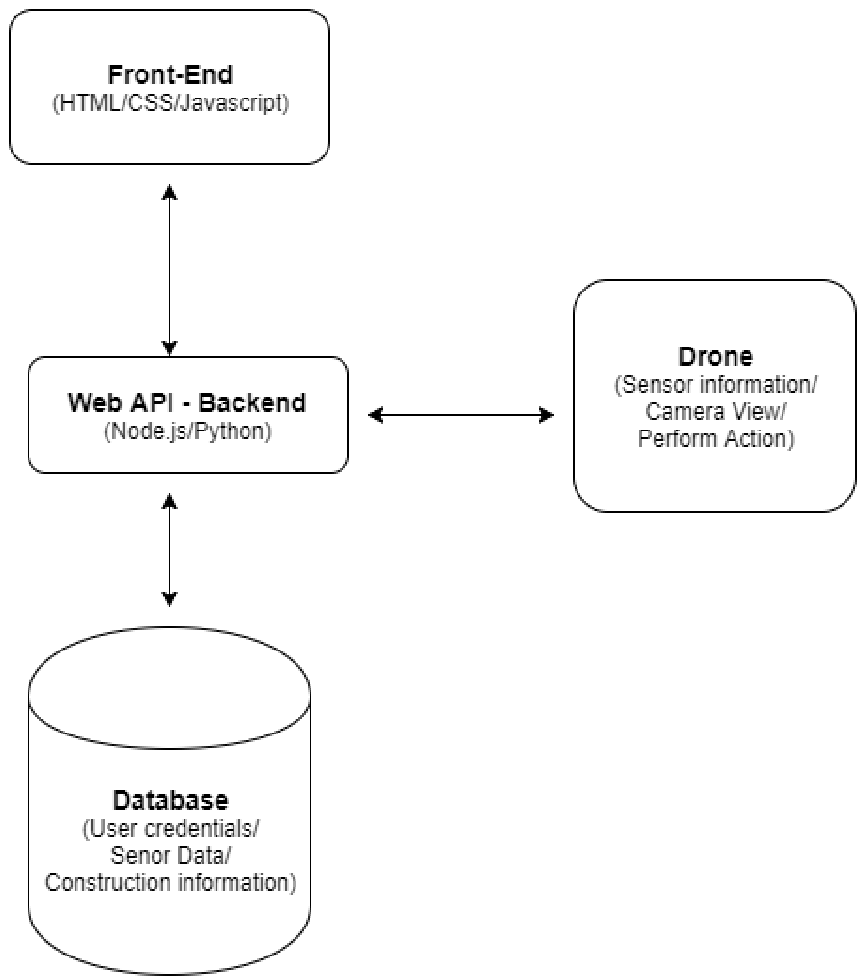


Figure 51: Web Application Design Overview

Possible test cases for the overall software are as follows. Traditional test cases would be seeing if the user if capable of logging in and registering successfully. The user shall be able to initiate the construction phase. The user shall be able to halt the construction process. Other possible test cases include allowing the user to design the structure in the web application.

Computer vision test cases include, seeing if the drone can successfully detect a Styrofoam block. Also, it should be able to descend to pick up or place down the Styrofoam block once detected. The drone should be able to avoid any obstacle in its way. Other computer vision test cases include see if the drone can successfully complete building the first layer of the structure. An important test case would be to see if the camera can send its data to the flight controller.

There should be test cases for the on-board Raspberry-Pi microcontroller as well. The computer should be able to take in data from the flight controller and live feed from the camera. All data should be transferred to the base station web server for the user to perform certain activities in the web application. The mentioned test cases are not meant to be successfully passed the first time and will be changed throughout the overall development process in Senior Design 2. More test cases will be added in the future.

The Front end that the user interacts with connects to the back-end of the web application. The backend connects all the important functions of the web application. It establishes communication with a Database that can information such as sensor data and user credentials. The backend is also in charge of establishing a connection with the drone. It allows the front end user to interact with the drone with functions and commands built on the web application.

7.0 Project Integration/Operation and Testing

This chapter will describe the SCUAV's overall project integration and overall testing. It is broken down into the prototyping and the design of the PCB and the overall hardware setup. Every component must be tested before implementing for safety and cost reasons. The overall software setup and testing are also mentioned in this chapter to blend and put everything together to begin designing the SCUAV prototype. Setting up test cases and thorough debugging will be enormously beneficial for SCUAV's design progress and project performance.

Below on Figure 45 is a pyramid stating the system architecture layers, having the PCB as the lowest level yet the most important component of SCUAV. The flight controller being the brain of SCUAV, and the web server backend will be used to communicate data from the flight controller to the web application. Finally, the user interface is the highest level of software required for SCUAV and allows user interaction. We can control and manipulate each part of the project with its appropriate programming language.

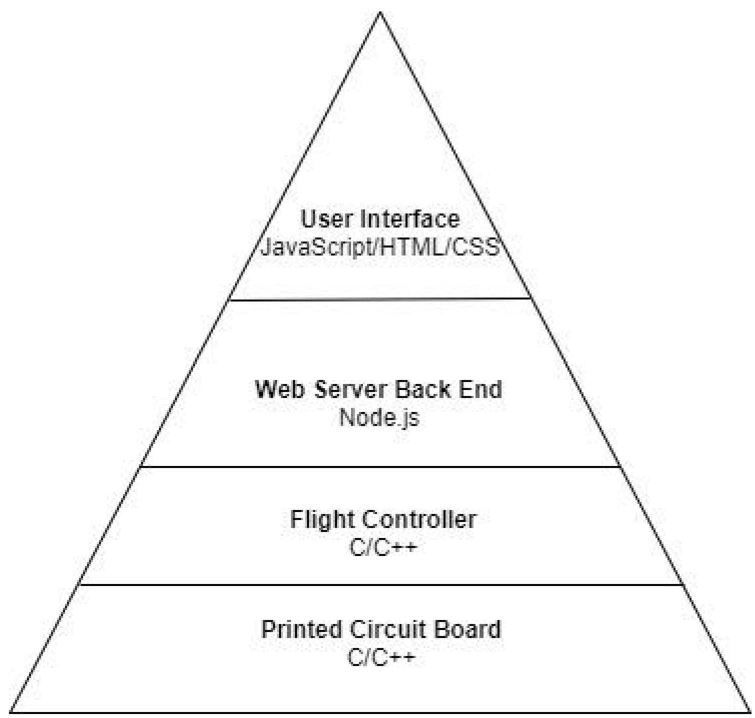


Figure 52: System Architecture Layers

As an update the above information in the figure has changed. Now the Raspberry Pi sits in the position of the Web Server Backend. And the User Interface is now a Bluetooth Application.

7.1 PCB Prototyping

We will have three stages to prototype the PCB. Prototyping is used as a preliminary model. For each component in the PCB design, three copies of the components will be ordered. The first will be used for breadboard testing, the second will be used for the protoboard testing stage, and the third will be used for the final PCB design. Since breadboards are solderless, the first component used on the breadboard is able to be used in the final PCB design.

7.1.1 Breadboard

Different options were explored for the first stage of the PCB design. We compared the advantages of using breadboards to using stripboards.

Breadboard

A breadboard is used when first testing a circuit design. Component leads are simply placed into the holes on the breadboard. Breadboards are also extremely useful for testing out each individual component in a design. We will be testing the voltage, current, and other ratings of our components to ensure their functionality prior to integrating them into our design. Breadboard are able to test all types of components, simple ones to complex integrated circuits. Along with testing each component individually, breadboards are useful for testing how each of the components can be connected together to realize a design. A breadboard will have binding posts where cables to power and ground can be easily connected. Power will be sent through the binding posts to the power rails on the PCB and from here wires can be used to send power anywhere in the breadboard. We will also be using a probe to connect to test various nodes within our design by viewing using an oscilloscope. An oscilloscope is used to view the electrical responses of our circuit to different inputs. Out of all of the types of boards available for circuit design, breadboards are the easiest to be reworked. The reason for this is that prototyping with this method does not require components to be soldered onto the circuit board to create connections.

Strip board

A strip board is fairly similar to a breadboard since it has default connections. One the back side of a strip board the wiring connects all the holes on the board either vertically or horizontally. Then, closely following the circuit design layout, the components are placed onto the strip board. It is easier if the pre-wired connections can be used so that there are no connections left to be made. However, this rarely occurs and it is likely that the wires will have to be added to board to create the remaining connections. The component leads are placed into holes on the strip board. The leads are then bent so that the component stays in place while the component gets soldered.

7.1.2 Perforated Board

Different options were explored for the second stage of the PCB design. We compared the advantages of using perforated boards to using protoboards.

Perforated Prototyping Boards

A perforated board is used in prototyping electronic designs. These boards are more permanent than other solderless boards and they are commonly used in the professional environment, where breadboards are not typically seen. Perforated boards are already etched and drilled. Perf boards can support the implementation of integrated circuit components and through hole parts. Discrete components are those which are a single unit However, surface mount components are not able to be implemented onto perforated boards. One of the main differences between strip boards and perf boards is that in a perforated board, each pad of the board is isolated. The layout for perforated boards can be laid out in PCB software. A 0.1-inch grid in the software is used. This is done to mimic physical perforated boards which consist of a square grid of 0.1 inch spacing. Another option is to describe the perforated board design in detail on paper. Once this is complete, the physical components can be soldered into the necessary holes of the perforated board. During this step, be sure that the components are placed in the correct orientation for parts with polarity. Once all the components have been added the perforated board, the interconnections between the components need to be completed. This can be done using a point to point wiring technique or wire wrapping. Now the design for the project’s circuit has been realized on the perforated board. Before testing the perforated board, the board should again be reviewed to ensure accurate connections. We’ve decided that perforated boards would be better for our design.

7.1.3.1 PCB Initial Design

The job of the PCB will be to control the robot claw of the drone as well as provide voltage and current levels being supplied to all six of the drone’s motors and the voltage and current being supplied from the LiPo battery to the power distribution board. The voltage and current level information will be sent to the on board computer to be monitored by the user. We decided to do this because it is vital that our battery provides an adequate amount of voltage and current to the ESCs through the power distribution board. The battery will supply a large amount of current, possibly 192.5 amps when the battery is fully charged, to the power distribution board. The motors, on the other hand, will operate at a maximum of 13 amps. Because of this the ESCs need to sufficiently limit the current as well as control the speed of the motors based on how frequently the voltage switches on and off. If there is a failure in one of the motors of the drone, the user can better determine its cause. Figure 47 displays how the voltage levels for the battery and motor can be detected. The voltmeter can be placed in parallel with the power delivered to the ESCs or from the battery. The voltage is then converted into digital information by an analog to digital converter. Prior to this however, the signal is amplified so that the voltage is able to be detected by the analog to digital converter. Figure 47 shows the same thing but for measuring the current. And figures 48 and 49 show the simulation of the ESC and battery voltage measurement. The simulations show that the circuit can provide a voltage and current at the necessary level. The analog to digital converter needs an input voltage between zero and five volts and an input current between 3 and 5 milli amps. The output voltage will be sent to the ATMEGA328P-PU chip. The maximum voltage for the analog to digital pin of this chip is 5 volts, so the output voltage needed to be scaled down. The ESC and battery voltage will be calculated based on the input voltage level before being sent to the Raspberry Pi.

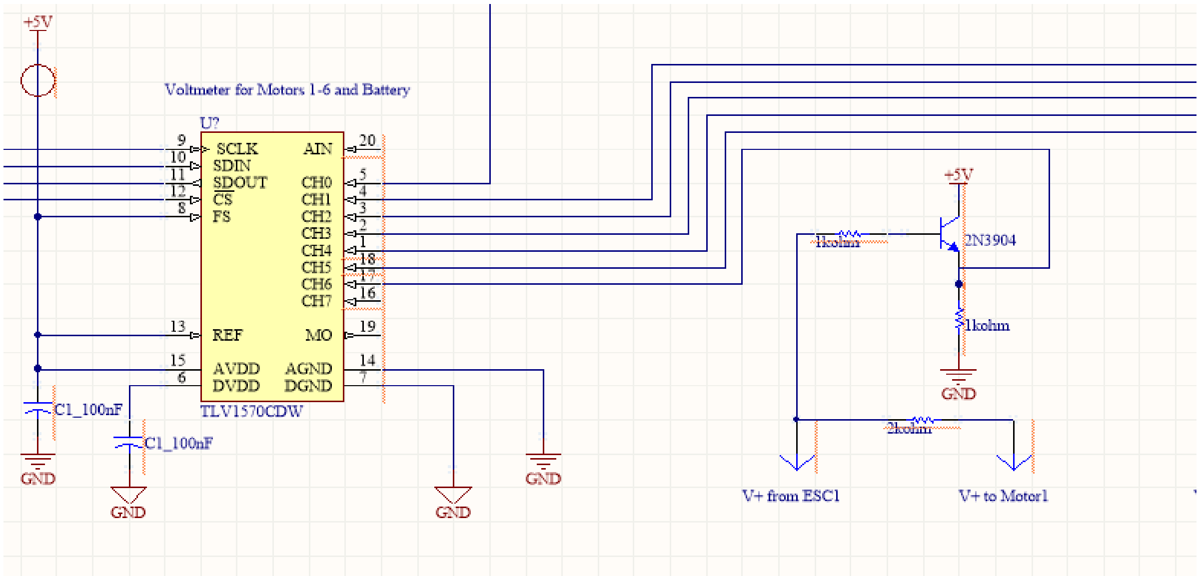


Figure 53: Voltmeter for the ESCs and Battery

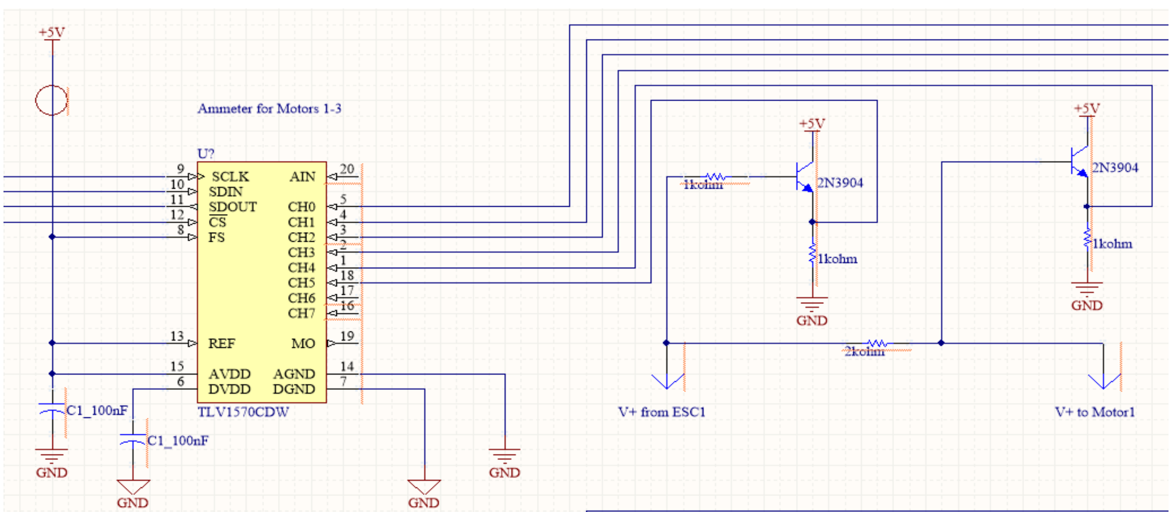


Figure 54: Ammeter for the ESCs and Battery

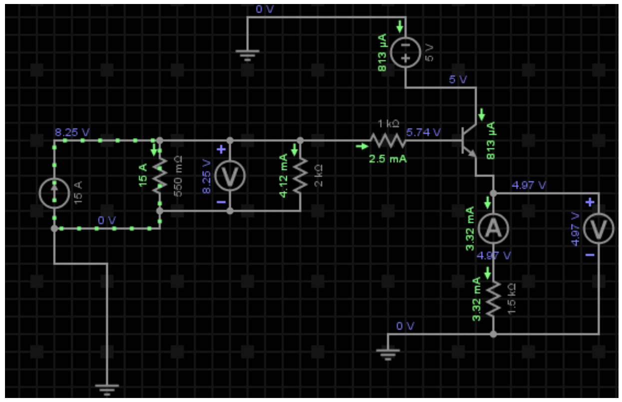


Figure 55: Simulated Circuit for ESC Current Handling

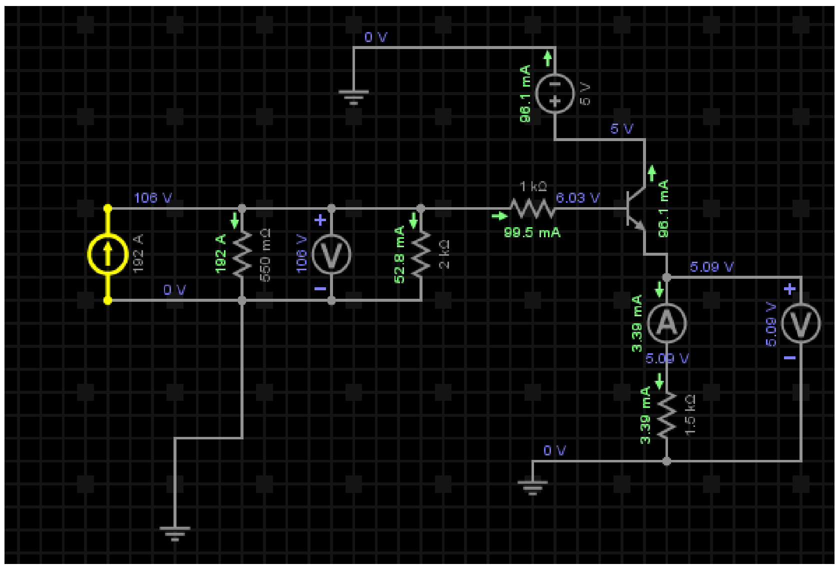


Figure 56: Simulated Circuit for Battery Current Handling

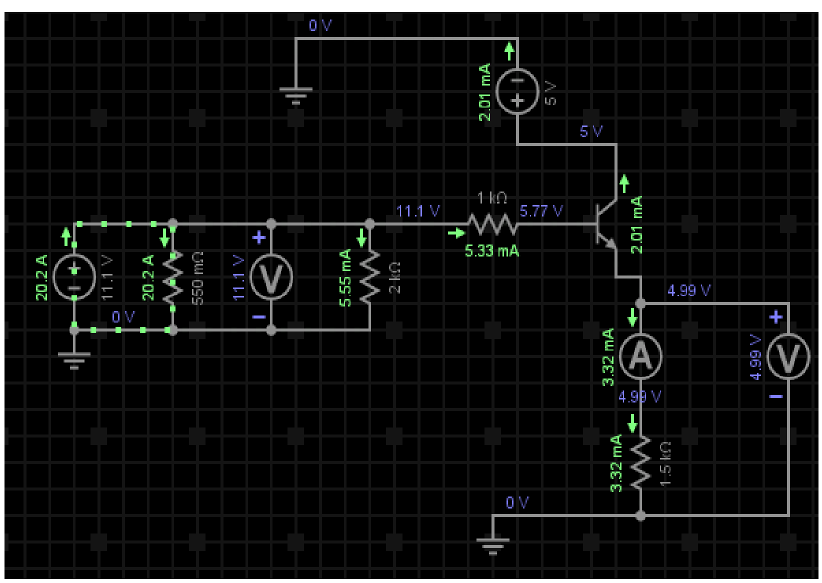


Figure 57: Simulated Circuit for ESC and Battery Voltage Handling

The PCB will communicate to the Pixhawk through the Pixhawk's I2C port. The Pixhawk will send a signal to the PCB indicating that the claw needs to either be opened or closed. This will be done with a dual channel digital potentiometer. One channel of the potentiometer will be programmed to output a pulse width modulation signal to the claw motor which will open it, and the other potentiometer channel will be pulled high when the motor needs to close the claw. Knowing which one of these command needs to be done will be done is determined by whether the drone is at the base station or the construction site. The PCB will be connected to the pixhawk controller though I2C and it will be able to determine the drone's location from the Pixhawk's GPS module. Figure 51 shows the complete PCB schematic and Figure 52 shows a closer view of the dual channel potentiometer. The potentiometer has two variable resistors which are each programmed to a specific value and scale down the input voltage differently.

7.2 Hardware Setup and Testing

We will individually test each of the electrical equipment in our project. We will power each of the components and use test its response using measurement equipment. We will analyze input and output voltage and current waveforms using an oscilloscope and obtain the current and resistance of devices using a multimeter. For individual testing of the equipment, a breadboard can be used.

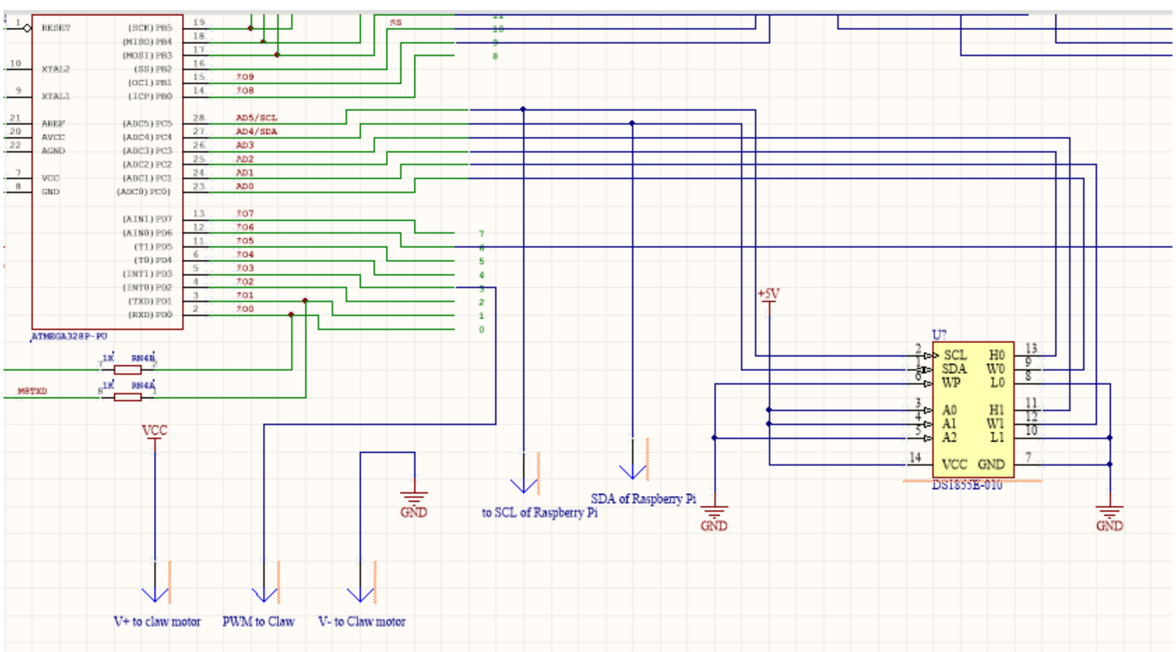


Figure 58: Claw Motor Control with Digital Potentiometer

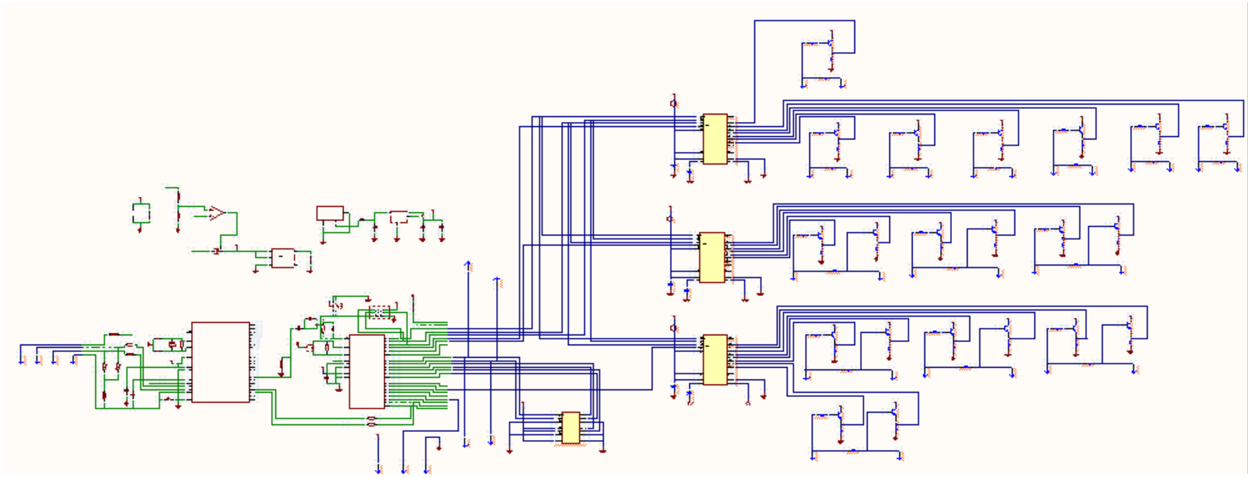


Figure 59: Complete PCB Schematic

When integrating electrical devices together and testing their performance, it will be useful to use a fuse. The fuse will help protect our equipment from being supplied too much current. The location of the fuse within a circuit is very important. Fuses should be connected between the power supply and a load. If too much current is being supplied to the load from the power source, then the fuse will open the circuit and disconnect the power from the load.

7.1.3.2 PCB Final Design

Several modifications were made to the initial design for our PCB. Instead of using the through hole version of the ATmega328 chip, we implemented the ATmega328 surface mount version. Our team also decided not to monitor the voltage and current of the battery with the PCB. We removed any circuitry dealing with battery monitoring from the PCB. Along with this, we realized there would be need for a digital potentiometer. Digital potentiometers are used to control the position of DC motors. However, our design will be using a servo motor. The position of the servo motor can be directly controlled from the ATmega328 microcontroller.

Our team realized that a rangefinder should be used to close the claw at the correct moment. The rangefinder will be used when the drone is picking up a block. As the UAV lowers to lift a block, the rangefinder will detect that the distance to the block is lowering. Once the ultrasonic rangefinder detects that block is block is close enough for the mechanical gripper to grasp the block, the servo motor closes the claw.

Figure 60 below shows the PCB design for controlling the rangefinder and the servo motor. U3 in the figure is a 16MHz crystal oscillator and U12 is a reset button.

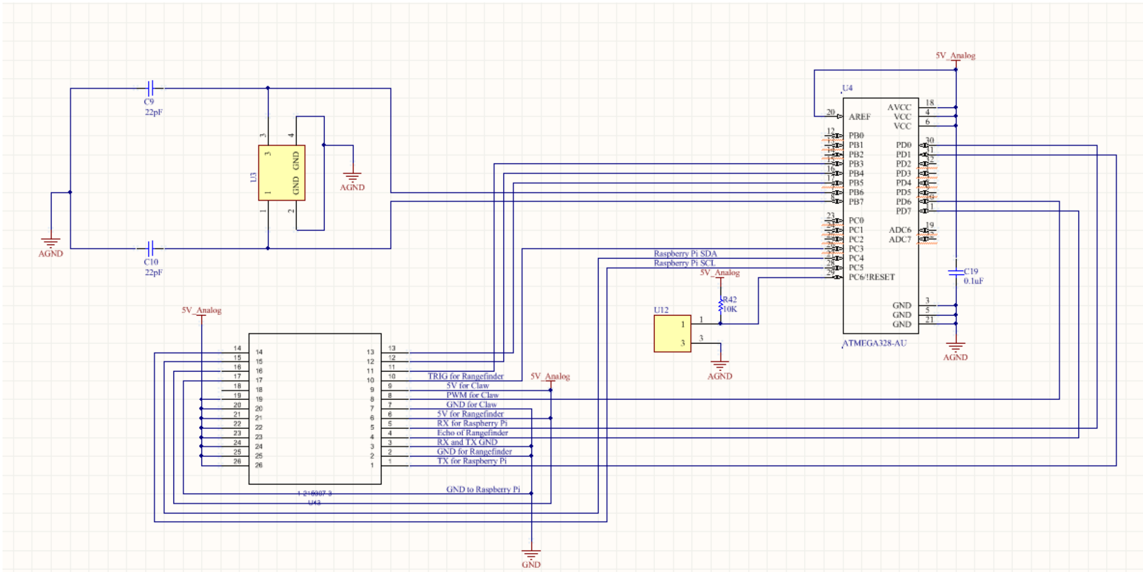


Figure 60: Final PCB Design for Claw and Servo Motor Control

7.1.3.3 PCB Programming

The SPI pins of the ATmega328 microcontroller were added to a header on the PCB for the ATmega328 chip to be easily programmed. Programming the PCB is done by using the Arduino Uno as an In-circuit Serial Programmer, or ISP. After configuring the Arduino Uno as an ISP, the code to be uploaded onto the PCB's ATmega328 chip is programmed onto the Arduino Uno. The program is then transferred from the Arduino Uno to the PCB by connecting the SPI pins and grounds. After the PCB is programmed and before it is connected to the drone's battery power supply, there is a break in power. The Atmega328 is a non-volatile device and will retain its program despite a break in power. Figure x helps describe the connections between the PCB and the Raspberry Pi, the rangefinder, and the servo motor. The figure also displays the connections used for programming the ATmega328 microcontroller.

7.2.1 Motor Testing

Motor testing is very important in our project. Motors need to be tested for the torque that they exert. This test value may be compared our calculations for this in the previous section. The current that the motors pull especially when connected to a load should also be found. The range of available current from the power supply should be able to support the maximum current that will be drawn be the motors. Along with this, the PWM characteristics for the motor should be indicated. The strength of the required PWM signal to rotate the motor also needs to be determined. Along with this, the rotation direction of the motors should be tested for. Each of the three motor states, clockwise, counterclockwise, and stopped. The heat buildup on the motor needs to be tested as well.

7.2.2 ESC Testing

For ESC testing, each ESC must be tested to make sure that the right amount of speed is being signaled to them. The flight controller has ESC Calibration software that helps to configure the ESCs to the desired speed measurements. The software is located on the Quick Starter Guide and it must be downloaded to the flight controller first with the ESCs connected. Another way to test the ESCs is to make sure that right amount of voltage and current is being processed through it and that it is being supplied to the motors. The ESCs also help with sending the direction signal to the motors and thus this will be tested when it is connected to a motor.

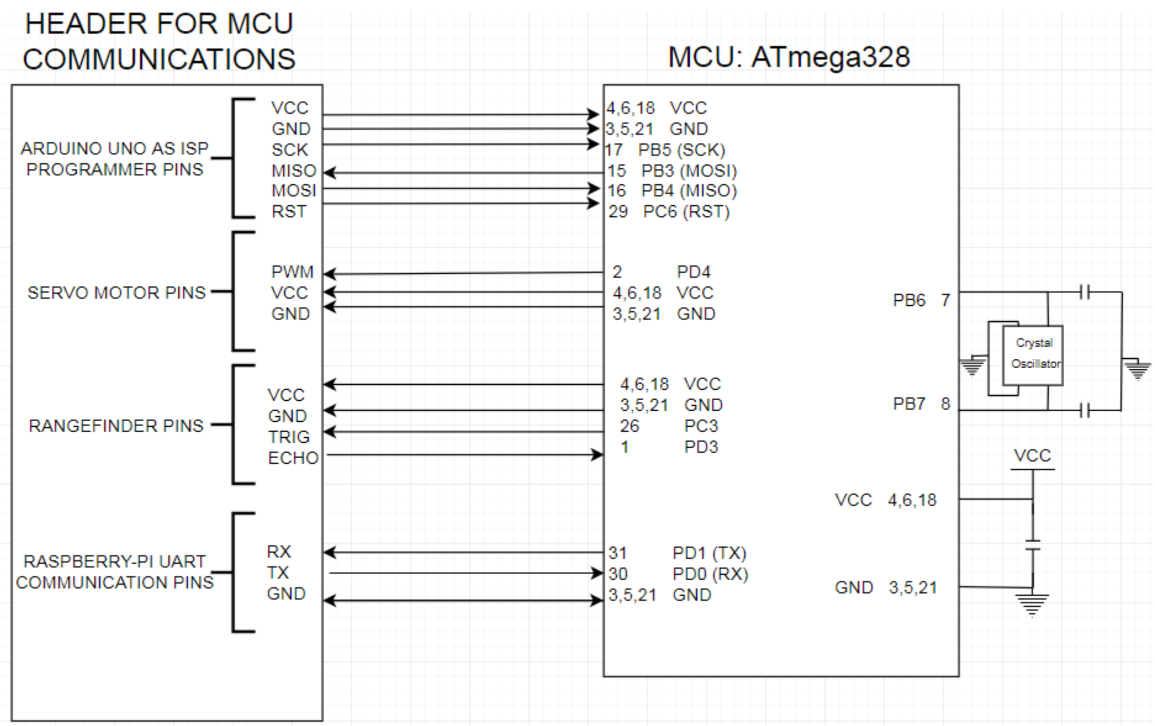


Figure 61: MCU Communications

7.2.3 Propeller Testing

Propeller testing is essential for the drone's flight and navigation. The microcontroller will be monitoring any feedback that it receives from propellers. The kind of feedback will be any noise or vibrations that propellers give off. To lower the chances of having any feedback that could potentially impact the drone's flight in a negative way, the propellers must be balanced. One way of balancing the propellers is to use a prop balancer. A prop balancer simply allows you to easily see where there is a weight unbalance in the propeller [73]. If the propeller is off-balance then the propeller can be simply sanded or masking tape can be used. Sanding is used when the heavier side of the balance needs to be made lighter in order for the propeller to be calibrated correctly. Adding masking tape to the lighter side of the balance will even out the weight distribution. The figure below, Figure 53, demonstrates how the balancer can be implemented to a propeller.

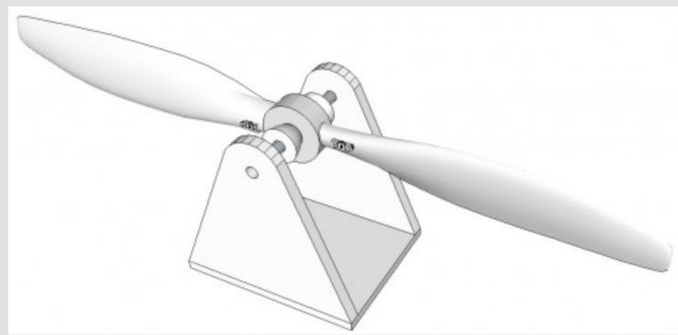
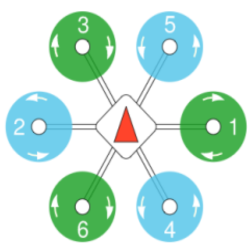

Propeller Balancer

Figure 62: Propeller Balancer [73]

Hexarotor x



Common Outputs

- **MAIN1**: motor 1
- **MAIN2**: motor 2
- **MAIN3**: motor 3
- **MAIN4**: motor 4
- **MAIN5**: motor 5
- **MAIN6**: motor 6
- **AUX1**: feed-through of RC AUX1 channel
- **AUX2**: feed-through of RC AUX2 channel
- **AUX3**: feed-through of RC AUX3 channel

Figure 63: Hexarotor Frame Type [74]

7.2.4 Flight Controller Testing

The Pixhawk Mini must be programmed to work for SCUAV's design. All of the components attached to it must be configured to work will all the settings on the flight controller. Each component plugged into it must be tested to see if there is proper communication between all the components and the microcontroller. On the software package provided by the Quick Start Guide there are multiple calibration techniques for each of the components labeled in the design. In addition to all the component configurations, the flight modes for SCUAV must be programmed and tested. Lastly, the Pixhawk must be tested when all the components are put together. This is very important for the success of the drone's functionality. All of the components must work together to obtain the objective of this project.

7.2.5 Frame Calibration

The drone's frame type must be programmed to the flight controller. The Pixhawk Mini must to be configured to suit a hexacopter. The Quick Start User guide has the software available for this. When the flight controller is first being set up the Airframe type will be asked. For SCUAV's purposes the Hexarotor x will be chosen.

Figure 56 shows the common outputs used to connect the frame to the ESCs or motors as well as which direction the propellers should spin in.

7.2.6 Camera Testing

To test the OpenmV m7 camera, the signal information will have to be tested. To do so, the Raspberry Pi will have to be tested to see if a signal is being received. The signal the on-board computer will be waiting on is the data that the camera sees. To know if the device sees anything, the drone will make its flight towards a potential object that the drone needs to pick up. If the drone does not respond to the camera's commands, modifications need to be made either on the software side or the configuration of the flight controller.

7.2.7 IMU and GPS Testing

When is the flight controller is initially booted on the computer, all the sensors in the IMU package and the GPS might have to be setup to suit the type of flying vehicle being used. This setup section can be found under the Sensor section on Pixhawk software setup menu. All of the drone's rotational forces must be tested and the drone's location must be signalled to the Pixhawk. The testing of these components is important for the drone's navigation because the drone must remain stable or level when flying, picking up the styrofoam blocks, and placing the blocks on the structure that is being built.

7.2.8 Gripper Testing

To test the gripper component, the servo(s) attached to it must be tested. The servo(s) will be tested to see if it opens and closes in the right direction once it is implemented to the gripper. The power being supplied must be tested as well. A timer might be implemented as will to allow for the gripper to open and close when picking up and placing the styrofoam blocks.

The gripper must be tested in order to see that the commands sent to the PCB are being acknowledged. When signaled to open the servo(s) attached to the gripper must spin in the direction that allows the claws to open to a certain degree. When the claws fix around the styrofoam block the servo(s) will turn in the opposite direction to close a little to make sure that the gripper is holding the block securely while it flies over to the building site. Once drone reaches the building site and it is hovering over where the block is to be placed, the servo will turn in the open direction slowly to place the block in its designated position. When the block is finished being placed, the servo will then close the gripper and wait for further instructions.

7.2.9 Rangefinder Testing

The LeddarOne must be tested to see if the sensor can pick up any of the drone's surroundings. When programming the rangefinder, the sensor must be able to pick up multiple distances away from it. So the smallest distance can be the first measurement measured and then increased to great distance as measurements are being tested. The lighting of the environment must be taken into account because it affects how accurate the measurement of an object away from the drone is. Figure 55 shows a demonstration of how this is done.

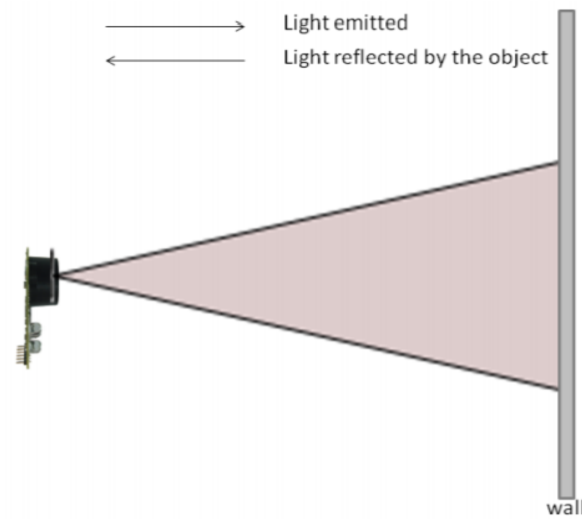


Figure 64: Demonstration of LeddarOne Testing Light [21]

7.2.10 Base Station

The base station of the drone will require some sort of physical computer capable of wireless transmission that will act as not only a web server but also a physical user interface. Current designs for the base of operation consists of a raspberry pi acting as the web server. The raspberry pi will be powered via portable battery charger with attached peripherals including a monitor and keyboard. The integration of these components must allow the user to quickly and effectively power on and boot into the web interface. To allow this functionality, a physical button mapped out the raspberry pi pins may be used which will enable a python script that will automatically power on and bring to the display the web interface. To house the hardware of the base station a pelican or brief case could be repurposed to mount the display, offer keyboard use, and a means of storing the battery along with raspberry pi. The button attached to a lever indicating that the case has been opened could begin the software routine of the Raspberry Pi.

7.3 Software Setup and Testing

The design of the software was broken into modular components. Each component focuses on a key aspect of the drone's objectives and functions. The setup between the camera and flight controller, connecting flight controller to Raspberry-Pi, and the on-board computer to the web server are thoroughly explained below. Also, the software methodology is explained below.

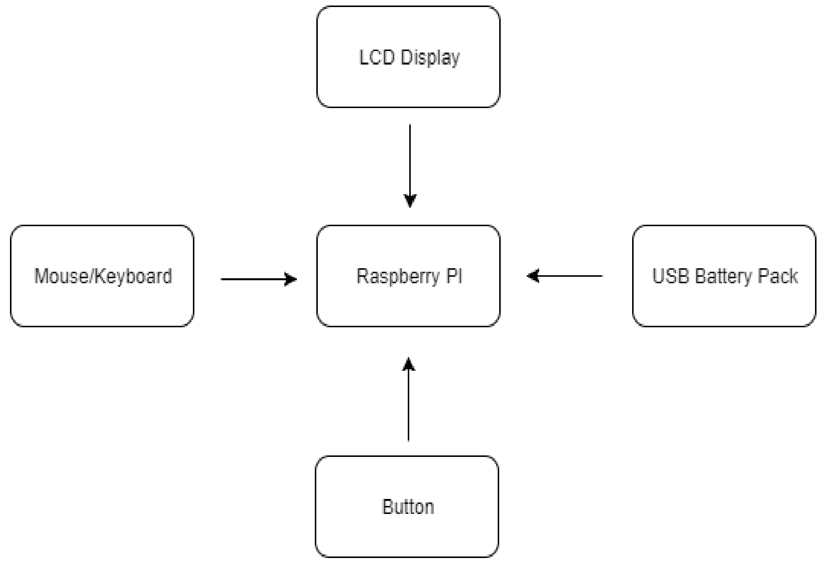


Figure 65: Base Station Diagram

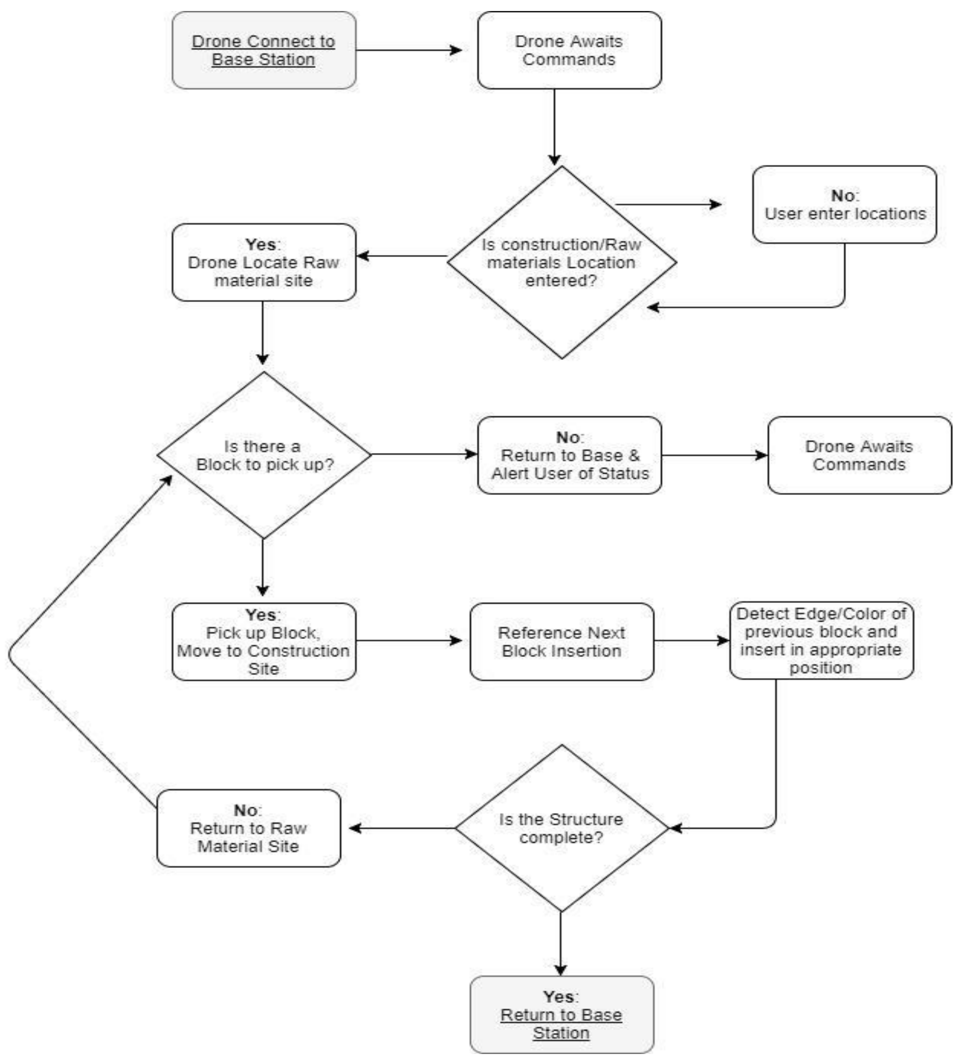


Figure 66: Drone Application Flowchart

The two figures above depict past information that has now been changed to better perform SCUAV's functionality. Now the flight path has been made more simpler and would perform much smoother. The figure of this path was mentioned in Chapter 3 of this document.

7.3.1 Camera to Flight Controller

The OpenMV camera will be able to grab data by creating a statistical model of the color contained in the object and will store in flash. With this statistical model, the OpenMV can find objects with similar colors and similar shapes. The camera will send data to the flight controller by using I2C communication. To test this, we can use its interface to view the output. The OpenMV IDE is able to show the frame buffering, histogram model of the color it is seeing, and more. Depending on what the camera sees, it can trigger the flight controller to perform certain commands such as descending the drone, pick up an object and place it in its appropriate position, avoid obstacles, and determining its completion.
As mentioned previously this setup no longer exists after much testing was conducted. The camera is best suited to be plugged into the Raspberry Pi. Below shows where the camera belongs to on the drone.

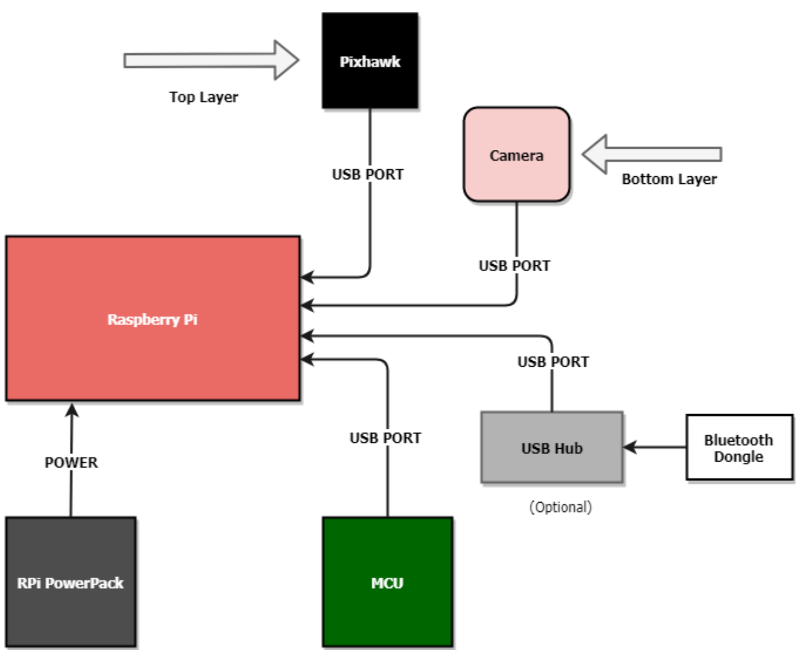


Figure 67: Middle Frame of the Drone

7.3.2 Flight Controller to Raspberry Pi

To set up the software on the flight controller, first download all the software relevant to the Pixhawk from its documentation page [75]. Based on what the company supplied follow all the steps needed to have a workable microcontroller. Second, the software needed for each component is downloaded to the microcontroller. Simple tests are run to check if all the software needed is uploaded correctly to the microcontroller. These tests comprised of breaking each component down to its own testing category and then combining all of them to make sure that the basic function of the drone is working correctly.

Testing Categories:

- IMU
    - Calibration and orientation testing
    - The impact of the drone's speed while in mid-flight
    - Combining the functionality of all three sensors
    - Flight stability
- Rangefinder
    - Height of drone during claw operation?
    - Can the drone avoid obstacles ?
    - Can the balance of the drone be maintained while the drone avoids obstacles ?
    - Laser scanning of the environment?
- GPS
    - Detection of the drone's location
    - Navigating between base station and construction site
    - Sending directional information to the microcontroller
- Camera
    - Does the Raspberry Pi receive information from the camera?
    - Successful flight to each object the camera sees?
    - Can the drone stay stable while listening to instructions sent from camera?

Once all of the testing objects are completed, the third stage of the software development begins. The flight controller will attempt to send all the information that it receives to the Raspberry Pi. To test if there was a successful connections between the two devices, simple instructions will be sent to the Raspberry Pi. For example, the microcontroller could ask if the Raspberry Pi received any longitude or latitude information from the GPS module or if it knows where the drone is currently located. If all the simple test proves to be successful, the microcontroller will attempt to combine multiple parts of its functionality and send them straight to the Pi. The Raspberry Pi will then transfer this information to the base station. If all of this proves to be successful, then the task of implementing and testing is complete.

7.3.3 Drone Raspberry Pi to Base Station Web Server

Once the on-board Raspberry Pi on the drone connects to the base station wireless access point it will begin communicating with the web server. To test the connection between the drone and base station, a simple ping from the base station to the static IP address of the drone's Raspberry Pi will verify that it is connected. The web server page will display an authentication login screen prior to access of drone functionality. To test this login page, proper user credentials must be entered, if a login error is displayed it could be caused by an incorrect entry or bug in the backend side in Node.js or current information in the database.

Once access is granted, a display showing the drone's current sensor data, progress in construction, and model of structure to be built will be shown. If the drone has not been given a task, a request for a three dimensional model will be requested to be created on the web interface consisting of simple blocks by the web user.

This three dimensional model will be handled by front end HTML and JavaScript. Once a 3D model has been created, the user will specify on the web page where the construction materials are as well as where to begin construction. To test that the data is being sent to the web server from the drone, the front end web page will display its retrieved values. If no values are displayed then there may be an issue in the backend code or an issue with the drone hardware itself. With the front end of the web server displaying the correct drone information, the user can now click on one of the action buttons displayed on the page. An action button should perform a basic task such as begin construction, suspend flight position, return to base, and land. With each action button, if the drone fails to begin construction, suspend flight, return to base, or land, any of these issues could be a backend server side issue with the code or the flight controller functions handled on the drone itself otherwise the task should be performed without issue.

7.3.4 Software Testing Methodology

The fundamental concept of software testing is evaluating a function's output given its input. In regards to the output, a value prior to evaluation is expected and depending on the actual output this function is determined to have passed or failed. Software testing can be broken into sub categories to differentiate and closer examine portions that need improvement.

Verification entails ensuring the product and service meets the conditions set place upon beginning the development phase. Validation focuses on meeting the specifications requirements given in the development phase. Basic types of software testing are composed down to two categories, white box and black box testing. Black box testing focuses on output generated given any input running on the system. White box testing covers not only the output but also the internals of the system for testing. White box testing covers validation while black box is most commonly used for verification. But again, these are the broad scopes of testing that can be seen as two different methods of testing.

In regards to black box testing, it has some advantages and disadvantages by its very nature. From the user standpoint, it can help expose failed specification requirements. The tester does not need to know how the software operates in order to test the results of a function. Also the test cases can be established once the specifications of the software are complete. In contrast to where black box holds some advantages, its disadvantages are just as plenty. When the inner workings of software are unknown, it leaves for only a small number of inputs to be tested to the testers perspective. This can be bad as it opens the door for future issues that were unknown to be solved and searched for before found. Without clear specifications, producing the optimal test cases can be difficult as it leaves the tester in the dark as to how to proceed.

White box testing has an advantage over black box testing in that it does not need the front end of an application to be develop in order to begin testing. With the inner workings of an application known, white box testing allows the tester a deeper view to the systems architecture. This helps widen the scope of features and functions to test that they are not only operating properly, but address vulnerabilities in security.

However, there are many types of testing including unit, integration, and system testing. Unit testing is the testing of a single or group of related components to produce an expected output given an input. Integration testing involves the testing of a group of components that work together to produce an output, with this category of testing falling between black box and white box testing. System testing is one of the highest levels of test, testing the software in distinct environments to ensure the system still operates dropping in the category of black box testing. As mentioned earlier in the Software Development section, testing will be split between the client and the server. Several test cases are mentioned to determine if the overall is functioning correctly. The software will be tested on the engineer's computer at first, and will then be tested on a regular user's computer. For the computer vision part, most of the testing will be based on the camera's functionality. These test cases are not final, as they will change throughout the overall development process. Newer test cases will also be added as we build. The requirements for the software as well as the software design details will be changed over time.

8.0 Administrative Content

This chapter will cover all milestones occurred during the development process of SCUAV. Below, the milestone timelines of each event are shown throughout both Senior Design 1 and Senior Design 2. Also, the budget analysis is shown on a spreadsheet specifying the cost of each component as well as the total cost. A summary of each team member's contribution will be stated below.

8.1 Team Members

Baian Elmazry contributed in the software development of SCUAV. He is responsible on working with the drone's computer vision and software navigation control. This is a crucial part of the project as it is needed for viewing blocks needed to picks, viewing its path, determining if the structure is complete and how to build the structure. Baian researched the different ways to detect the edges and detect colors. He came up with the most optimal solution for SCUAV's camera abilities. He has researched what camera will be most beneficial in completing the project. Alan Hernandez also aided Baian in his research. It was also researched how the camera would be integrated for receiving data and sending data to the on board computer. Baian also worked developing SCUAV's on board computer algorithms. An essential component of the project, the on board computer Raspberry-Pi is used to communicate data from the flight controller to the rest of the components.

The system's Web Application Interface was going to be lead by Alan Hernandez with aid from Baian Elmazry. It was to handle backend communication between the base station's web server, developed in Node.js capable of many features and functions of the drone. Functions developed for the web interface will allow manual control of the drone, updates to the drone's position and other features. This feature has now been changed to a Bluetooth Application. Alan will be responsible for establishing communication between the webserver and the drone's on board computer. A client and server will need to be created to establish this as well as for interaction between the user and the drone. The Web Interface will be designed with security in mind, focusing on functionality as well as scalability. The code developed will uphold to common coding practices and be document for further development.

Nicola DaSilva is part of the Hardware Team and will be contributing to the hardware setup, design, and implementation. She will collaborate with her team member, Veronica Love, on all hardware systems. Alan also played a part in helping with hardware test issues. Most of the programming for any hardware device such as the flight controller will be done by Nicola but the autonomous code was done by Baian. She will have to test that it is working with all of the components plugged into the flight controller. Nicola will accompany Veronica with any physical testing of all the electrical components of this project such as the PCB and breadboarding that goes along with it. Nicola will also work with Baian Elmazry to connect the OpenmV camera to the Raspberry Pi. This collaboration will be essential for all the things the drone sees and help with the building process that is designed to be performed by SCUAV.

Veronica Love is in charge of the PCB design and schematic diagram. She is the part of the hardware team and she assists Nicola DaSilva with all the hardware setup for the drone's design. Both team mates will collaborate together on all the testing that needs to be done for all the components. She will also assist in any hardware programming needed to setup any of the components. Most of the electrical components will be tested by her but she will have help from Nicola. She successfully made the PCB's MCU become programmable and it responds to the Raspberry Pi's commands. She also was helpful in any important electrical / hardware decisions were needed to be made. Overall, she performed her role perfectly.

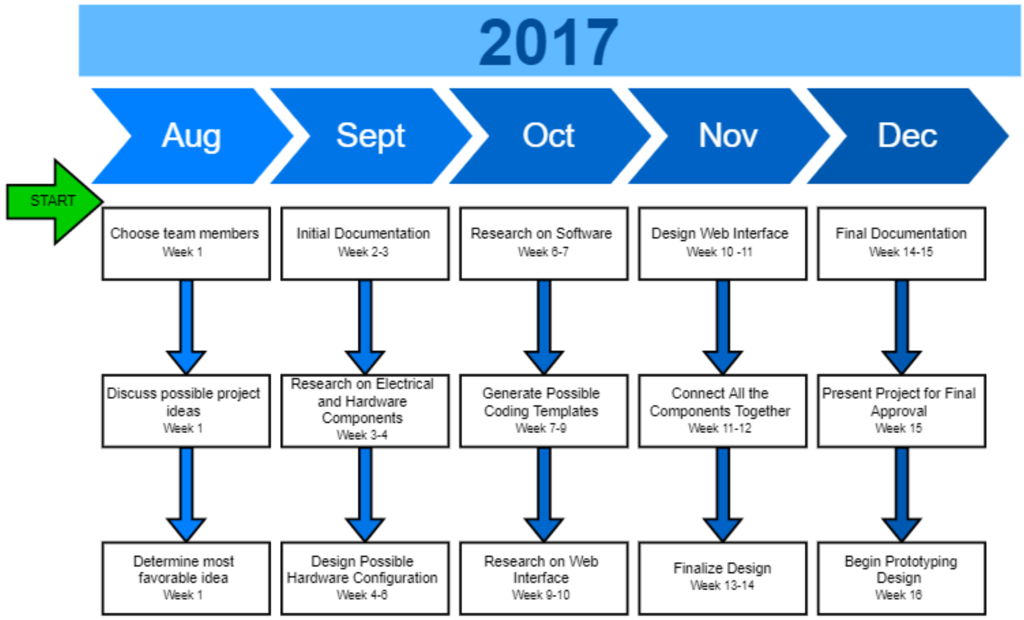
8.2 Project Milestones

Below are the project milestones flowcharts stating everything that needed to be completed from Senior Design 1 to Senior Design 2. This includes development phases in both hardware and software, and it also includes all of the documentation to be completed throughout the timeline. We are predicting that SCUAV will be fully functioning and completed in one semester with the building process starting in December 2017. The downside is that most of SCUAV's functionality was performing until some mishaps occured.

8.3 Budget Analysis

Below, there are two tables regarding the budget analysis for the project. The project will be self-financed by the team members. The first table, Table 18, shows the initial budget for SCUAV. The total cost for initial budget was a whopping $976.00, where each team member would spend approximately $244.00. On Table 19, there is final and current budget for the SCUAV. The total budget completely differs from the original budget with a total of $1,363.38, where each team member would spend approximately $681.69. Table #13 shows the comparison of the pricing of possible microcontroller from the researching days. the total cost of this project did change compared to its initial stages.

Senior Design 1 : EEL 4914C



Senior Design 2: EEL 4915L

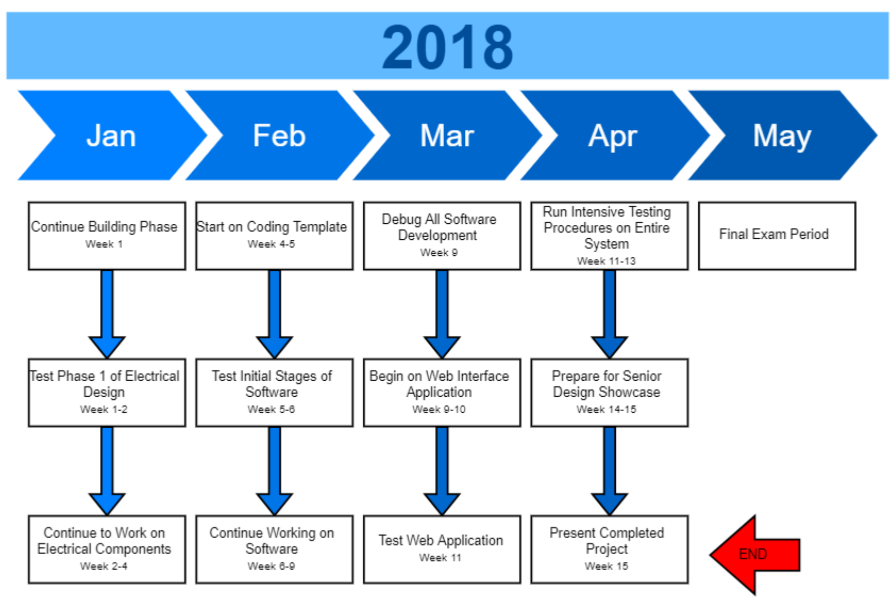


Figure 68: Project Timeline

Table 18:Initial Budget

Budget and Financing					
Category	Item Description	Item	Min. Quantity	Expected Quantity	Unit Price
Quadcopter	Microcontroller	Arduino Uno	1	1	$10.00
	ESCs	ESCs (4pack)	1	1	$20.00
	Motors	Motors (4 pack)	2	1	$40.00
	Battery	Battery	1	1	$14.00
	E-compass/Gyro/Accelerometer	BerryGPS - IMU	1	1	$50.00
	IR Range Sensor	IR Range Sensor	1	1	$120.00
	Robot Claw Kit	Robot Claw Kit	1	1	$20.00
	Odroid	Odroid	1	1	$0.00
	Camera	Camera	1	1	$70.00
	Drone Frame	Frame	1	1	$27.00
	Motor propellers	Prop (8 pack)	1	1	$10.00
	Voltage Transformer	Transformer (5pack)	1	1	$20.00
Base Station					
	Raspberry Pi		1	1	$35.00
	PCB	Estimate	1	1	$500.00
				Total Estimate:	$976.00

Table 19: Final Budget

Component	Quantity	Unit Price	Total Price
Pixhawk Flight Controller	1	$95.23	$95.23
Electronic Speed Controllers	6	$0.00	$0.00
Motors	6	$0.00	$0.00
Propellers	12	$2.79	$33.48
Battery: 3000mAh 4S	3	$49.99	$149.97
Battery: 3300mAh 4S	2	$51.50	$103.00
Balance Charger	1	$84.99	$84.99
Ultrasonic sensor	1	$3.00	$3.00
Mechanical gripper	1	$17.00	$17.00
OpenMV Cam M7	1	$65.00	$65.00
Voltage Transformer	1	$20.00	$20.00
Drone Set	1	$128.00	$128.00
Raspberry-Pi 3 Model B	2	$70.00	$140.00
Power Module connector	1	$6.79	$6.79
Power Module	1	$25.99	$25.99
Styrofoam	1	$10.00	$10.00
FrSky Taranis receiver	1	$44.97	$44.97
Frsky ACCST Taranis	1	$124.99	$124.99
PCB components	1	$202.00	$202.00
Advanced Circuits PCB	1	$93.00	$93.00
Raspberry Pi Expansion Power Board	1	$15.97	$15.97
Total		**$1,111.21**	**$1,363.38**

Table 20: Microcontroller Comparison Pricing

Item	Quantity	3DR Pixhawk Mini	Pixhawk	Arduino Uno R3
Board	1	$229.99	$95.28	$22.00
ESCs	1	$20.00	$20.00	$20.00
Battery	1	$14.00	$14.00	$14.00
IR Range Sensor	1	$120.00	$120.00	$120.00
Robot Claw Kit	1	$20.00	$20.00	$20.00
Motors	1	$40.00	$40.00	$40.00
BerryGPS - IMU	1	$0.00	$0.00	$50.00
PixyCam	1	$70.00	$70.00	$70.00
Raspberry Pi	1	$35.00	$35.00	$35.00
Drone Frame	1	$27.00	$27.00	$27.00
Motor Propellers (8 pack)	1	$10.00	$10.00	$10.00
Voltage Transformer (5 pack)	1	$20.00	$20.00	$20.00
PCB (estimate)	1	$500.00	$500.00	$500.00
Total		$1,105.99	$971.28	$948.00

The feature above outlines expected prices of comparable microcontrollers that could be implemented in the project. Each one was selected with regard to features that may be used on the drone. Both the 3DR Pixihawk Mini and Pixhawk are well established flight controllers that are well known for their use in drones. Both offer software that offers many functions, but also well supported and documented.

Where the competition can take a noticeable change among the ones listed above is in the Arduino Uno R3. At a price point of $22.00, it can be the developer’s choice when on a short budget. It offers many of the same hardware features found on the Pixhawk and 3DR Pixhawk Mini. However, where the Arduino Uno R3 falls short is in regards to software. The development for its software covers a wide range of features, but many of which have not been built with the intent of drone projects in mind.

Features found in the software of the 3DR Pixhawk Mini and Pixhawk greatly laid favor to this project as they were designed with drone’s in mind. Between these two flight controllers, price and the weight of each one was factored in into the project when making a decision on selecting the flight controller.

9.0 Project Summary or Conclusion

Working to building an autonomous drone to complete the building of a structure has helped each of us develop technical abilities as well as interpersonal skills. The operation of our project relied greatly both hardware and software. Areas of work with software within our project included the programming of the Raspberry-Pi for autonomous flight, the camera for image detection, and the PCB for claw and rangefinder operation. Areas of work with hardware within our project included the design of the PCB, the distribution of power between the battery, ESCs, and motors, and the hardware integration between the various electrical components. While hardware and software were the main focuses for our project to operate, our project also called for mechanical design. Areas of mechanical work within our project included, the design of the drone frame, modifications to the claw, and the design of the drone’s legs.

Our group was faced with many challenges when completing our project. Many of the software challenges we experienced had to do with the communication protocols used between components and the integration of the code operating the various components. Some of the greatest hardware difficulties had to do with the distribution of power between the battery, ESCs, and motors, and the compatibility of these components. There were also obstacles dealing with mechanical design of the drone, these mainly had to do with creating stable legs for the UAV. We anticipated to have more issues with the weight distribution of our drone. However, our UAV, being a hexacopter, was stable enough that any weight offset did not seem to affect the flight. We never needed to rearrange the components within the drone to redistribute the weight.

We successfully completed the integration of the camera, rangefinder, and claw with the Raspberry-Pi. This portion of our project controls the detection, lifting, and landing of blocks to stack them. We also successfully integrated the web-application, base station computer, Raspberry-Pi, and PixHawk. These components control the autonomous flight of the drone.

However, we were unable to integrate the autonomous flight of the drone with the stacking of the blocks. After proving several times that the drone could be controlled autonomously and demonstrating that blocks could be detected and stacked, our next goal was to add the operation of the camera, claw, and rangefinder one at a time to the autonomous flight of the drone.

Unfortunately, we were unable to do the full demonstration of our project. One of the motors of our drone had started to burn during a recent test flight. We found that the ESC was very hot, implying that it had drawn much more current from the battery than what it was specified for. After testing connections and running tests, we believe a likely reason for the malfunction may have been a short to the coil of the motor that malfunctioned. A short to the power distribution board would have pulled very high current from the drone's battery and possibly caused permanent damage to the board. The power distribution board seems to the source of the problem because after replacing both the motor and ESC with compatible components, the issue remained. When this issue arose we did not have enough time to fix the cause of the issue and complete our project before the end of the semester.

Despite not demonstrating the full operation of our project, our group demonstrated great progress. All of our components have been integrated together to create a sound design. Throughout this process, we developed our technical skills in fields including electrical, computer, and mechanical engineering. We learned about the process of completing a project from its investigation and design to its operation. We also learned to work as a team to complete a program task.

10.0 Appendices

10.1 Appendix A - Works Cited

[1] mwaibel, “Flight Assembled Architecture,” *YouTube*, 02-Dec-2011. [Online]. Available: https://www.youtube.com/watch?v=JnkMyfQ5YfY. [Accessed: 04-Dec-2017].

[2] dezeenmagazine, “Drones can collaborate to build architectural structures,” *YouTube*, 22-Apr-2016. [Online]. Available: https://www.youtube.com/watch?v=g4tv899wn94. [Accessed: 04-Dec-2017].

[3]TheDmel, YouTube, 13-Jan-2011. [Online]. Available: https://www.youtube.com/watch?v=W18Z3UnnS_0. [Accessed: 23-Nov-2017].

[4] M. Simon, “This Robot Tractor Is Ready to Disrupt Construction,” Wired, 18-Oct-2017. [Online]. Available: https://www.wired.com/story/this-robot-tractor-is-ready-to-disrupt-construction/#ampshare=https://www.wired.com/story/this-robot-tractor-is-ready-to-disrupt-construction . [Accessed: 16-Nov-2017].

[5] Tech in Asia - Connecting Asia's startup ecosystem. [Online]. Available: https://www.techinasia.com/talk/6-known-ways-power-a-drone.[Accessed: 23-Nov-2017].

[6] S. Bouchard, “Latest Blog Post,” Robot Vacuum Cup Grippers: Top 5 Problems. [Online]. Available: https://blog.robotiq.com/bid/53128/Vacuum-cups-robot-gripper-Top-5-problems. [Accessed: 23-Nov-2017].

[7] C. Bernier, “Latest Blog Post,” Magnetic Robot End Effector: Top 5 Pros and Cons. [Online]. Available: https://blog.robotiq.com/bid/65794/Magnetic-Robot-End-Effector-Top-5-Pros-and-Cons. [Accessed: 23-Nov-2017].

[8]“Arduino Mini 05,” Arduino . [Online]. Available: https://store.arduino.cc/usa/arduino-mini-05 [Accessed: 17-Nov-2017].

[9] G. Coley, “BeagleBone Black System Reference Manual,” *BeagleBone Black System Reference Manual*, 20-May-2013. [Online]. Available: https://www.bing.com/cr?IG=5482E3C5C7D34F0E93E253693EBC93FE&CID=0361F4901AEA6BF82B06FFDE1BEC6AC7&rd=1&h=gWWXsA1yaWQGQ_RRyIF278jTnlHbGvJXhDnYF7kUDdg&v=1&r=https%3a%2f%2fcdn-shop.adafruit.com%2fdatasheets%2fBBB_SRM.pdf&p=DevEx,5061.1. [Accessed: 03-Dec-2017].

[10] “3DR Pixhawk Mini,” 3DR. [Online]. Available: https://store.3dr.com/products/3dr-pixhawk . [Accessed: 17-Nov-2017].

[11] "Gyroscope," Sparkfun. [Online]. Available: https://learn.sparkfun.com/tutorials/gyroscope/how-a-gyro-works. [Accessed: 04-Dec-2017].

[12] Seiko Epson Corp, "Gyro sensors - How they work and what's ahead | about Gyro sensor | Technical Information | other Information," Epson. [Online]. Available: http://www5.epsondevice.com/en/information/technical_info/gyro/. [Accessed: 04-Dec-2017].

[13] F. Corrigan, "Drone Gyro Stabilization, IMU And Flight Controllers Explained," DroneZon, 05-Jun-2017. [Online]. Available: https://www.dronezon.com/learn-about-drones-quadcopters/three-and-six-axis-gyro-stabilized-drones/. [Accessed: 17-Nov-2017].

[14] Sparkfun Electronics, " Accelerometer Basics," Sparkfun. [Online]. Available: https://learn.sparkfun.com/tutorials/accelerometer-basics . [Accessed: 04-Dec-2017].

[15] P. Jain, "Magnetometers," EngineersGarage, 21-Mar-2017. [Online]. Available: https://www.engineersgarage.com/articles/magnetometer. [Accessed: 04-Dec-2017]

[16] u-blox, "NEO-M8 u-blox M8 concurrent GNSS modules Data Sheet." u-blox, America, 05-Aug-2016. [Online]. Available: https://www.u-blox.com/sites/default/files/NEO-M8-FW3_DataSheet_%28UBX-15031086%29.pdf

[17] u-blox, "LEA-6 u-blox 6 GPS Modules Data Sheet." u-blox, America, 17-Feb-2017. [Online].Available:https://www.u-blox.com/sites/default/files/products/documents/LEA-6_DataSheet_%28UBX-14044797%29.pdf

[18] Trimble , "TRIMBLE MB-ONE Datasheet." Trimble , Sunnyvale, CA. [Online].Available:http://intech.trimble.com/library/DS_MB-ONE_US.pdf

[19] ArduPilot, "LIDAR-Lite Rangefinder," ArduPilot , 2016. [Online]. Available: http://ardupilot.org/copter/docs/common-rangefinder-lidarlite.html. [Accessed: 04-Dec-2017].

[20] GARMIN, "Lidar Lite v3 Operation Manual and Technical Specifications." Garmin, Olathe, Kanas, Sep-2016. [Online]. Available:https://static.garmin.com/pumac/LIDAR_Lite_v3_Operation_Manual_and_Technical_Specifications.pdf

[21] Leddar, “Leddar™One User Guide.” Leddar, Quebec, Canada, 2015.[Online]. Available: http://www.robotshop.com/media/files/pdf2/rb-led-02-04_-_54a0025-2_leddar_one_user_guide_-_2016-09-01.pdf

[22] LeddarTech, “LeddarTech LeddarOne Optical Rangefinder (3.3V UART),” Robot Shop. [Online]. Available: http://www.robotshop.com/en/leddartech-leddarone-optical-rangefinder-33v-uart.html. [Accessed: 04-Dec-2017].

[23] ArduPilot , “Leddar One Lidar,” ArduPilot , 2016. [Online]. Available: http://ardupilot.org/copter/docs/common-leddar-one-lidar.html. [Accessed: 04-Dec-2017].

[24] Sparkfun Electronics, “LIDAR-Lite v3,” SparkFun Electronics. [Online]. Available: https://www.sparkfun.com/products/14032 . [Accessed: 04-Dec-2017].

[25] T. Agarwal, “Introduction To Electronic Speed Control (ESC) Circuits, Working And Applications,” ElProCus - Electronic Projects for Engineering Students, 11-Nov-2016. [Online]. Available: https://www.elprocus.com/electronic-speed-control-esc-working-applications/ . [Accessed: 17-Nov-2017].

[26] “Brushed vs Brushless ESC - What makes them different?,” Hooked on RC Airplanes. [Online]. Available: http://www.hooked-on-rc-airplanes.com/brushed-vs-brushless-esc.html. [Accessed: 17-Nov-2017].

[27] ig.uana.16, “Brushed Vs. Brushless Motors: What's the Difference? | PTR,” Pro Tool Reviews, 27-Dec-2016. [Online]. Available: https://www.protoolreviews.com/news/brushed-vs-brushless-motors/18990/. [Accessed: 23-Nov-2017].

[28] C. McGrady, “Brush or Brushless? Which DC Motor Should You Choose?,” Arrow.com, 07-Feb-2017. [Online]. Available: https://www.arrow.com/en/research-and-events/articles/which-dc-motor-is-best-for-your-application. [Accessed: 23-Nov-2017].

[29] D. Forum, “Propeller Types,” DJI FORUM. [Online]. Available: https://forum.dji.com/thread-40912-1-1.html. [Accessed: 17-Nov-2017].

[30] Sam, “Drone Propellers – Know Everything About Them!,” Top Drones For Sale, 10-Dec-2016. [Online]. Available: http://www.topdronesforsale.org/drone-propellers/ . [Accessed: 17-Nov-2017].

[31] C. Benson, “How to Make a Drone / UAV – Lesson 3: Propulsion,” Robot Shop, 26-Mar-2015. [Online]. Available: http://www.robotshop.com/blog/en/make-uav-lesson-3-propulsion-14785 . [Accessed: 17-Nov-2017].

[32] "Computer Vision with Raspberry Pi and the Camera Pi module," *Open Electronics*. [Online]. Available: https://www.open-electronics.org/computer-vision-with-raspberry-pi-and-the-camera-pi-module/. [Accessed: 03-Dec-2017].

[33] "8MP Sony IMX219 Camera for Raspberry Pi," *Arduino Based Camera*, 20-Oct-2017. [Online]. Available: http://www.arducam.com/8mp-sony-imx219-camera-raspberry-pi/ [Accessed: 03-Dec-2017].

[34] "OpenMV Cam M7," *OpenMV*. [Online]. Available: https://openmv.io/products/openmv-cam-m7. [Accessed: 03-Dec-2017].

[35]"CMUcam5 Pixy," *Overview - CMUcam5 Pixy - CMUcam: Open Source Programmable Embedded Color Vision Sensors*. [Online]. Available: http://cmucam.org/projects/cmucam5. [Accessed: 03-Dec-2017].

[36]"Camera Module V2," *Raspberry Pi*. [Online]. Available: https://www.raspberrypi.org/products/camera-module-v2/. [Accessed: 03-Dec-2017].

[37] "Raspberry Pi 2 & 3 Vs Beaglebone Black," *Uploading*, 10-Jan-2017. [Online]. Available: https://www.engineersgarage.com/blogs/raspberry-pi-2-3-vs-beaglebone-black . [Accessed: 03-Dec-2017].

[38]"Computer Vision vs. Machine Vision," *Vision Online*. [Online]. Available: https://www.visiononline.org/vision-resources-details.cfm/vision-resources/Computer-Vision-vs-Machine-Vision/content_id/4585 . [Accessed: 03-Dec-2017].

[39]"About the BMVA," *What is computer vision?* [Online]. Available: http://www.bmva.org/visionoverview . [Accessed: 03-Dec-2017].
[40]"Sobel Edge Detector," *Feature Detectors - Sobel Edge Detector*. [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm. [Accessed: 03-Dec-2017].

[41]"Canny Edge Detector," *Feature Detectors - Canny Edge Detector*. [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/HIPR2/canny.htm. [Accessed: 03-Dec-2017].

[42]"Divide and conquer algorithms ," *Khan Academy*. [Online]. Available: https://www.khanacademy.org/computing/computer-science/algorithms/merge-sort/a/divide-and-conquer-algorithms . [Accessed: 03-Dec-2017].

[43]UCF Center for Research in Computer Vision, "COCOA," *Center for Research in Computer Vision*. [Online]. Available: http://crcv.ucf.edu/projects/COCOA/ . [Accessed: 03-Dec-2017].

[44] OpenCV: Introduction to SURF (Speeded-Up Robust Features). [Online]. Available: https://docs.opencv.org/3.2.0/df/dd2/tutorial_py_surf_intro.html . [Accessed: 03-Dec-2017].

[45]"FAST," *OpenCV: FAST Algorithm for Corner Detection*. [Online]. Available: https://docs.opencv.org/3.2.0/df/d0c/tutorial_py_fast.html . [Accessed: 03-Dec-2017].

[46]"ORB," *OpenCV: ORB (Oriented FAST and Rotated BRIEF)*. [Online]. Available: https://docs.opencv.org/3.3.0/d1/d89/tutorial_py_orb.html . [Accessed: 03-Dec-2017].

[47] "Tearing an old laptop apart to build a ground control station," Hackaday, 30-Aug-2013. [Online]. Available: https://hackaday.com/2013/08/31/tearing-an-old-laptop-apart-to-build-a-ground-control-station/. [Accessed: 03-Dec-2017].

[48] "API," *Python For Beginners*. [Online]. Available: http://www.pythonforbeginners.com/api/ . [Accessed: 03-Dec-2017].

[49] "Node.JS vs Python: Which is Best Option for Your Startup," *Vizteck*, 29-Aug-2017. [Online]. Available: https://vizteck.com/blog/node-js-vs-python-best-option-startup/. [Accessed: 03-Dec-2017].

[50] "Hypertext Transfer Protocol," *Wikipedia*, 03-Dec-2017. [Online]. Available: https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol . [Accessed: 03-Dec-2017].

[51] T. Point, "Node.js Tutorial," *www.tutorialspoint.com*, 31-Dec-1969. [Online]. Available: https://www.tutorialspoint.com/nodejs/ . [Accessed: 03-Dec-2017].

[52] TechMagic, "NodeJS vs Ruby on Rails comparison 2017. Which is the best for web development?," *Medium*, 03-Feb-2017. [Online]. Available: https://medium.com/@TechMagic/nodejs-vs-ruby-on-rails-comparison-2017-which-is-the-best-for-web-development-9aae7a3f08bf . [Accessed: 03-Dec-2017].

[53] "Guides.rubyonrails.org," *Ruby on Rails Guides*. [Online]. Available: http://guides.rubyonrails.org/v3.2/ . [Accessed: 03-Dec-2017].

[54] "The JSON Data Interchange Format.," *ECMA International. ECMA International*, Oct-2013. [Online]. Available: http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf . [Accessed: 16-Nov-2017].

[55] "Drone No Fly Zone Airspace Map - FAA's Restricted No Drone Zone Areas," Best Drones For Me, 01-Apr-2017. [Online]. Available:

http://bestdronesforme.com/drone-no-fly-zone-airspace-map-faa/ . [Accessed: 03-Dec-2017].

[56]"IEEE IEEE Code of Ethics," IEEE - IEEE Code of Ethics. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html . [Accessed: 03-Dec-2017].

[57] "No Drone Zone," *FAA seal*, 11-Jul-2017. [Online]. Available: https://www.faa.gov/uas/where_to_fly/no_drone_zone/ . [Accessed: 03-Dec-2017].

[58] "IEEE IEEE Code of Ethics," IEEE - IEEE Code of Ethics. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 03-Dec-2017].

[59] "Date and time format - ISO 8601," ISO 8601 Date and time format, 29-Aug-2017. [Online]. Available: https://www.iso.org/iso-8601-date-and-time-format.html . [Accessed: 03-Dec-2017].

[60] "ECMAScript 2016 Language Specification," ECMA International. ECMA International, Jun-2016. [Online]. Available: http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf. [Accessed: 16-Nov-2017].

[61] "Nodejs : Security Vulnerabilities," *Nodejs : Security vulnerabilities*. [Online]. Available: https://www.cvedetails.com/vulnerability-list/vendor_id-12113/Nodejs.html . [Accessed: 03-Dec-2017].

[62] "Global Positioning System Standard Positioning Service Performance Standard.," Gps.gov, Sep-2008. [Online]. Available: https://www.gps.gov/technical/ps/2008-SPS-performance-standard .pdf. [Accessed: Nov-2017].

[63] "DC Systems & Battery Safety Evolution through Codes & Government Regulations," *IEEE Power & Energy Society*. [Online]. Available: https://www.ieee-pes.org/presentations/gm2014/2756.pdf . [Accessed: 03-Dec-2017].

[64] "Interview with Altium! 3/21/17 4:30pm ET @altium @adafruit #adafruit," Adafruit Industries - Makers, hackers, artists, designers and engineers!, 15-Mar-2017. [Online]. Available: https://blog.adafruit.com/2017/03/15/interview-with-altium-32117-430pm-et-altium-adafruit-adafruit/ . [Accessed: 23-Nov-2017].

[65] "Benefits / Advantages of Surface Mount Technology (SMT) Over Through-Hole," Electronics Tutorial | The Best Electronics Tutorial Website, 30-Jul-2015. [Online]. Available: http://www.electronicsandyou.com/blog/benefits-advantages-

of-surface-mount-technology-smt-over-through-hole.html. [Accessed: 23-Nov-2017].

[66] George and R. Schahrer, “Differences Between Tin/Lead and Lead-Free Solder,” Simply Smarter Circuitry Blog, 20-Jun-2014. [Online]. Available: https://www.circuitspecialists.com/blog/differences-between-tinlead-and-lead-free-solder/ . [Accessed: 23-Nov-2017].

[67] ArduPilot , “Communicating with Raspberry Pi via MAVLink¶,” ArduPilot , 2016. [Online]. Available: http://ardupilot.org/dev/docs/raspberry-pi-via-mavlink.html. [Accessed: 04-Dec-2017].

[68] LeddarTech, “Leddar™One Single-Element Sensor Module.” LeddarTech, Quebec, Canada.[Online]. Available: http://www.robotshop.com/media/files/pdf2/leddarone-datasheet.pdf

[69] “Single Engine ESC Wiring,” 2BFly. [Online]. Available: http://2bfly.com/knowledgebase/electronic-speed-controls/single-engine-esc-wiring/ . [Accessed: 17-Nov-2017].

[70] DroneTrest, “How to Attach Your Prop to Your Multicopter Motors,” Unmanned Tech. [Online]. Available: http://www.unmannedtech.co.uk/motors-and-prop-attachment.html. [Accessed: 04-Dec-2017].

[71] “Pixhawk Autopilot,” Pixhawk Autopilot - Pixhawk Flight Controller Hardware Project. [Online]. Available: https://pixhawk.org/modules/pixhawk#specifications . [Accessed: 17-Nov-2017].

[72] “Front End and back End (Flowchart),” *Creately*. [Online]. Available: https://creately.com/diagram/example/hv7ovbfp2/front%20end%20and%20back%20end . [Accessed: 03-Dec-2017].

[73] C. Benson, “How to Make a Drone / UAV – Lesson 3: Propulsion,” Robot Shop, 26-Mar-2015. [Online]. Available: https://www.robotshop.com/blog/en/make-uav-lesson-3-propulsion-14785. [Accessed: 04-Dec-2017].

[74] ArduPilot, “LIDAR-Lite Rangefinder,” ArduPilot , 2016. [Online]. Available: http://ardupilot.org/copter/docs/common-rangefinder-lidarlite.html. [Accessed: 04-Dec-2017].

[75] “Pixhawk Autopilot: Quick Start Guide ,” 3DR. [Online]. Available: https://3dr.com/wp-content/uploads/2017/03/pixhawk-manual-rev7-1.pdf. [Accessed: 17-Nov-2017].

[76] HowToMechatronics. *Ultrasonic Sensor HC-SR04 and Arduino Tutorial*. (2018). [online] Available at:

https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/ [Accessed 13 Apr. 2018].

[77] Micropik.com. (2018). [online] Available at: http://www.micropik.com/PDF/HCSR04.pdf [Accessed 13 Apr. 2018].

10.2 Appendix B - Copyright Permission

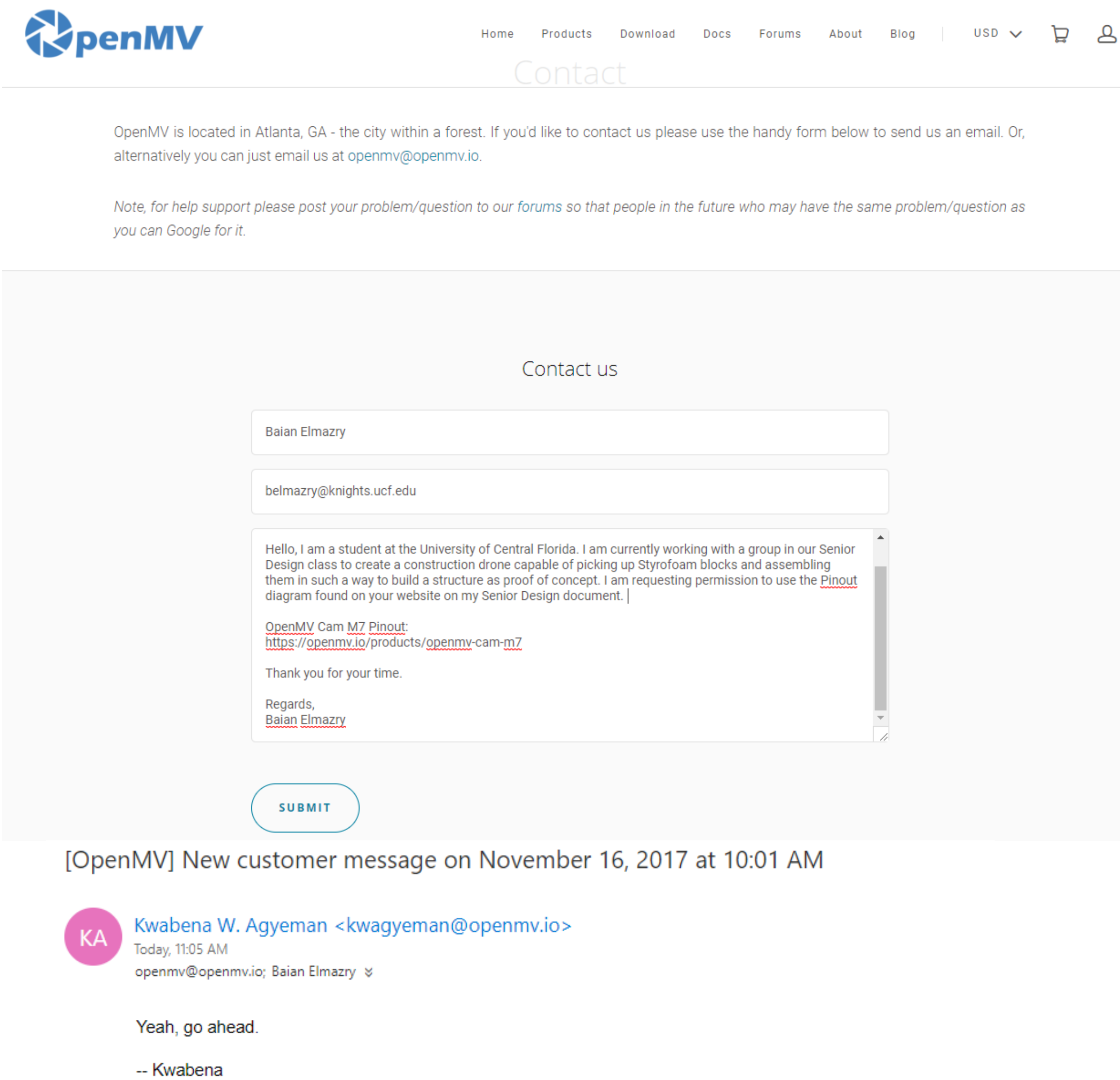


Figure 69: OpenMV Cam Diagram Permission Request /Granted

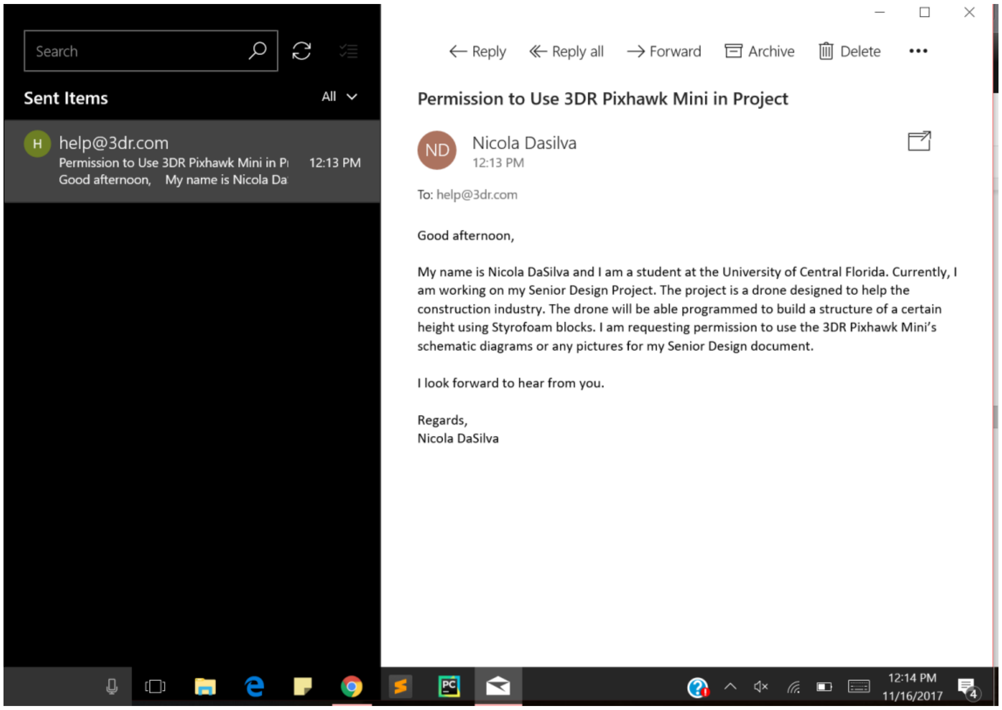


Figure 70: 3DR Pixhawk Mini Permission Request

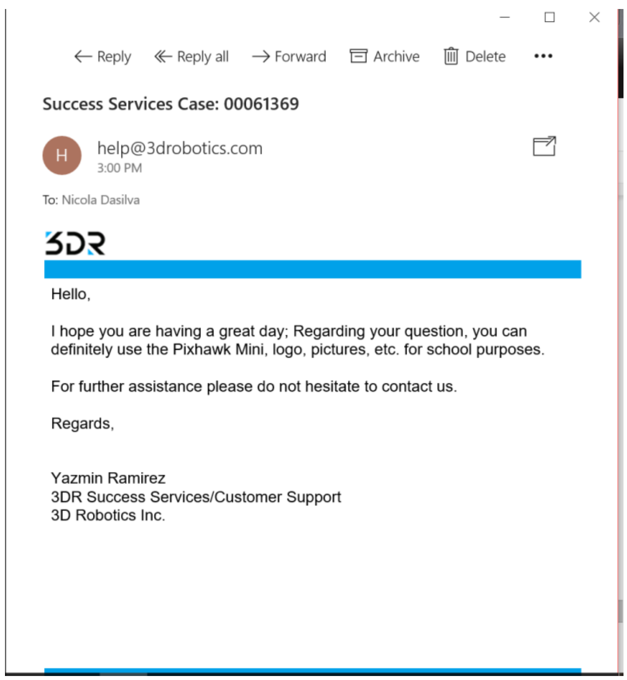


Figure 71: 3DR Pixhawk Response

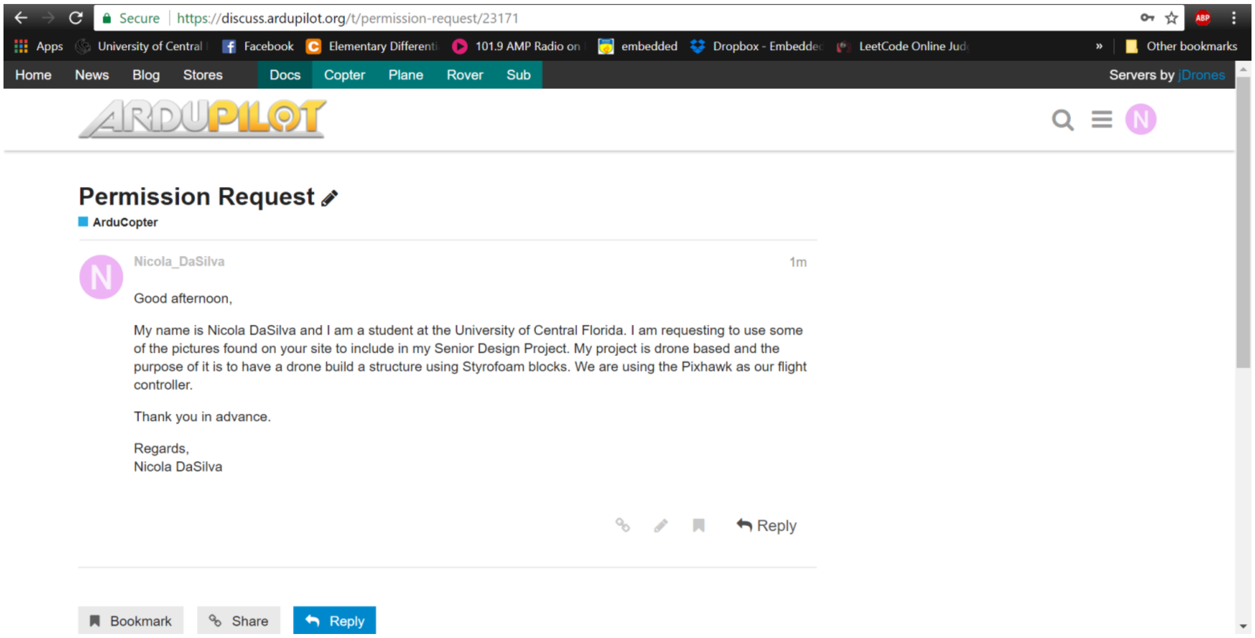


Figure 72: Permission Request to Use Pixhawk Images

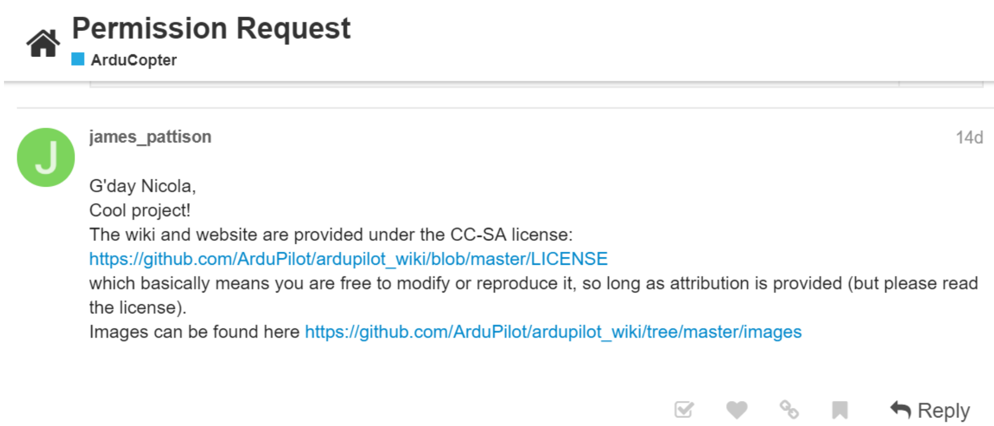
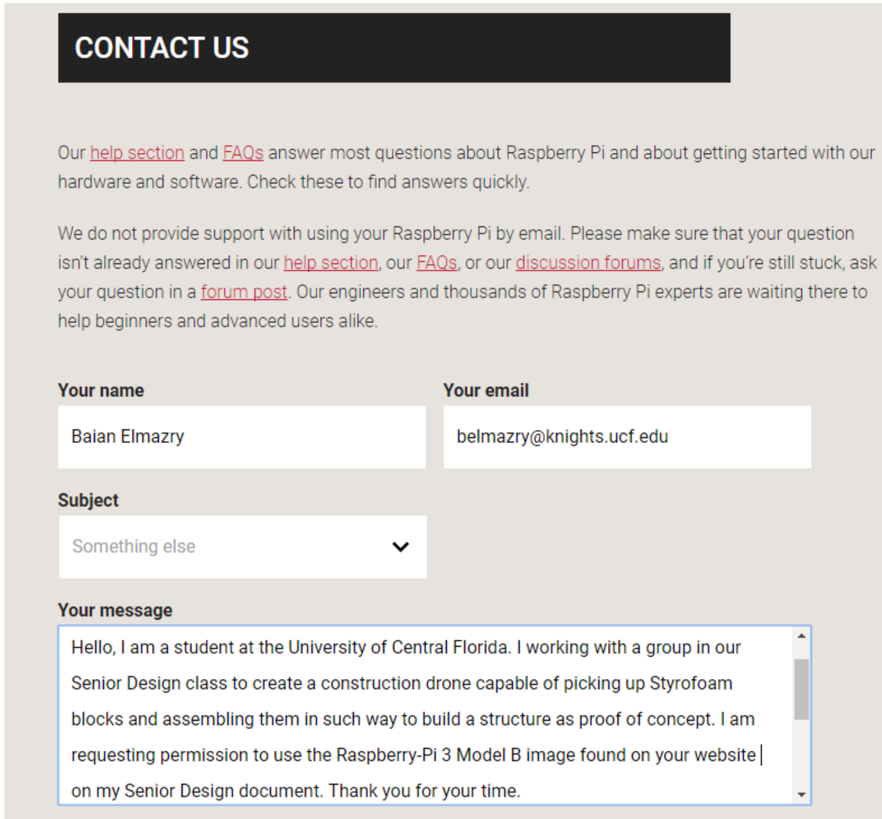


Figure 73: Response to Using Ardupilot Website



CONTACT US

Our help section and FAQs answer most questions about Raspberry Pi and about getting started with our hardware and software. Check these to find answers quickly.

We do not provide support with using your Raspberry Pi by email. Please make sure that your question isn't already answered in our help section, our FAQs, or our discussion forums, and if you're still stuck, ask your question in a forum post. Our engineers and thousands of Raspberry Pi experts are waiting there to help beginners and advanced users alike.

Your name
Baian Elmazry

Your email
belmazry@knights.ucf.edu

Subject
Something else

Your message
Hello, I am a student at the University of Central Florida. I working with a group in our Senior Design class to create a construction drone capable of picking up Styrofoam blocks and assembling them in such way to build a structure as proof of concept. I am requesting permission to use the Raspberry-Pi 3 Model B image found on your website on my Senior Design document. Thank you for your time.

Figure 74: Raspberry-Pi 3 Model B Permission Request

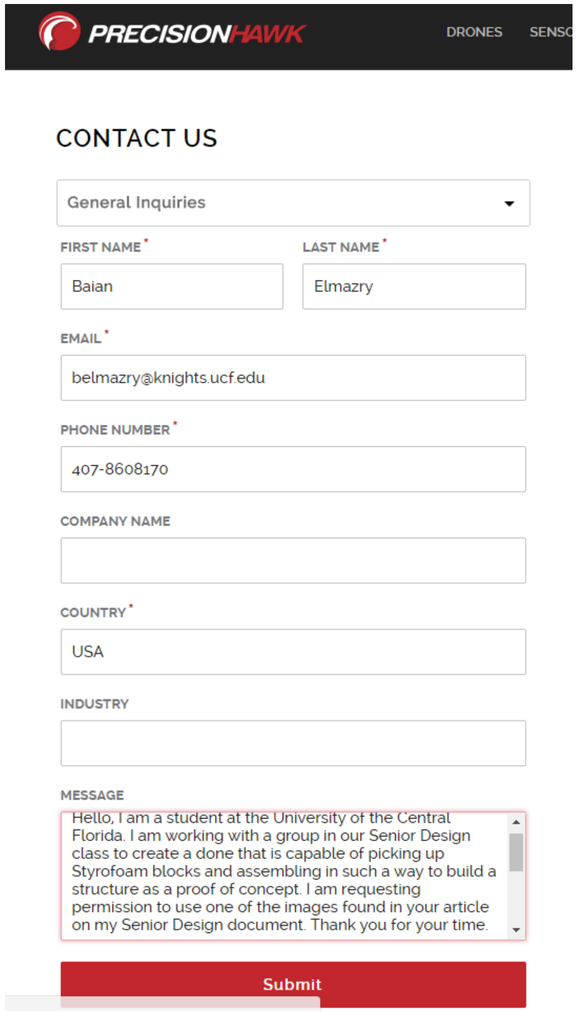
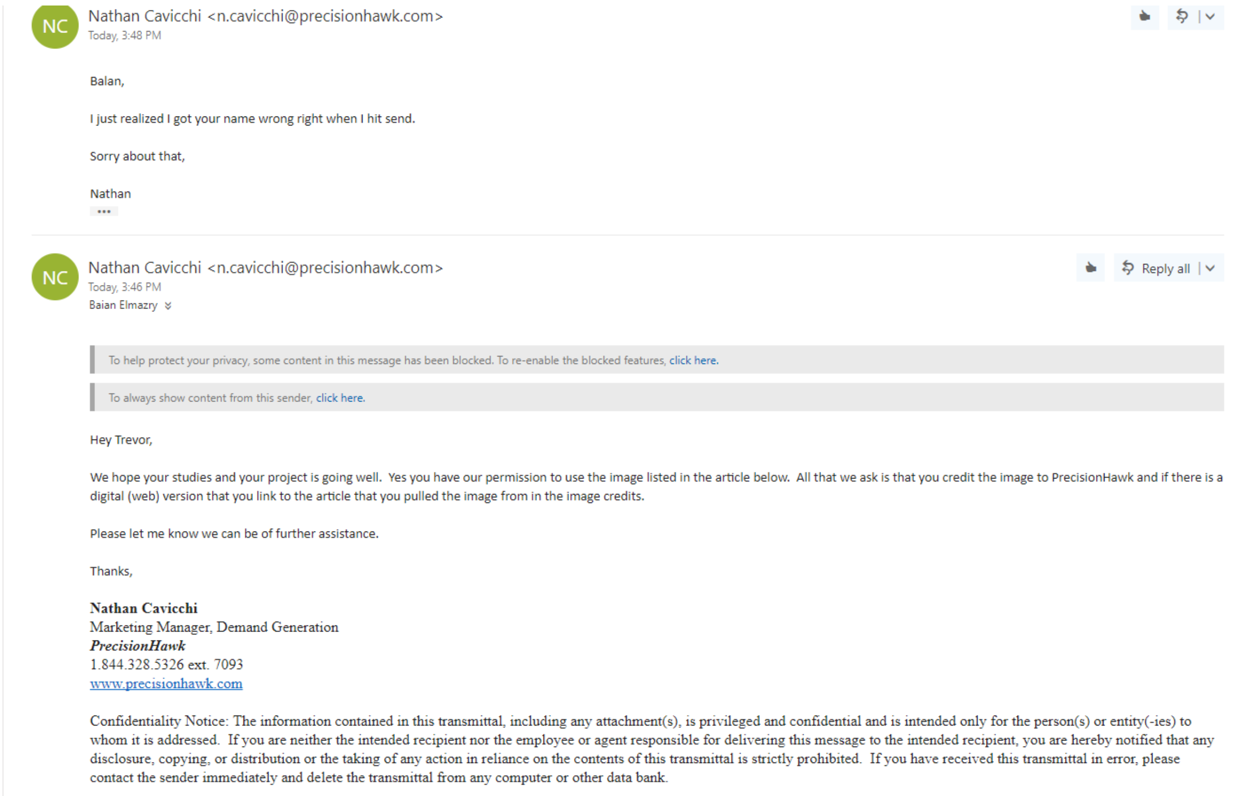


Figure 75 Cover page photo permission request



Nathan Cavicchi <n.cavicchi@precisionhawk.com>
Today, 3:48 PM

Balan,

I just realized I got your name wrong right when I hit send.

Sorry about that,

Nathan

Nathan Cavicchi <n.cavicchi@precisionhawk.com>
Today, 3:46 PM
Baian Elmazry

To help protect your privacy, some content in this message has been blocked. To re-enable the blocked features, click here.

To always show content from this sender, click here.

Hey Trevor,

We hope your studies and your project is going well. Yes you have our permission to use the image listed in the article below. All that we ask is that you credit the image to PrecisionHawk and if there is a digital (web) version that you link to the article that you pulled the image from in the image credits.

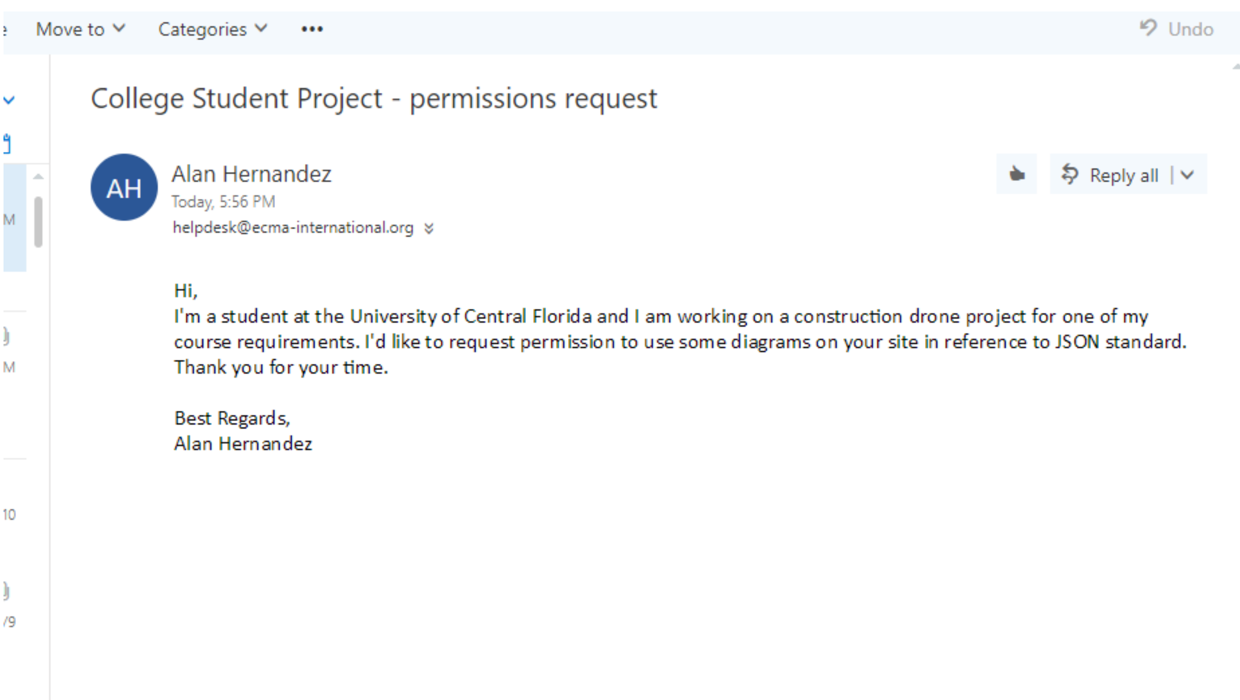
Please let me know we can be of further assistance.

Thanks,

Nathan Cavicchi
Marketing Manager, Demand Generation
PrecisionHawk
1.844.328.5326 ext. 7093
www.precisionhawk.com

Confidentiality Notice: The information contained in this transmittal, including any attachment(s), is privileged and confidential and is intended only for the person(s) or entity(-ies) to whom it is addressed. If you are neither the intended recipient nor the employee or agent responsible for delivering this message to the intended recipient, you are hereby notified that any disclosure, copying, or distribution or the taking of any action in reliance on the contents of this transmittal is strictly prohibited. If you have received this transmittal in error, please contact the sender immediately and delete the transmittal from any computer or other data bank.

Figure 76: Cover photo permission granted.



Move to Categories ••• Undo

College Student Project - permissions request

Alan Hernandez
Today, 5:56 PM
helpdesk@ecma-international.org

Reply all

Hi,
I'm a student at the University of Central Florida and I am working on a construction drone project for one of my course requirements. I'd like to request permission to use some diagrams on your site in reference to JSON standard.
Thank you for your time.

Best Regards,
Alan Hernandez

Figure 77: Diagram permission request

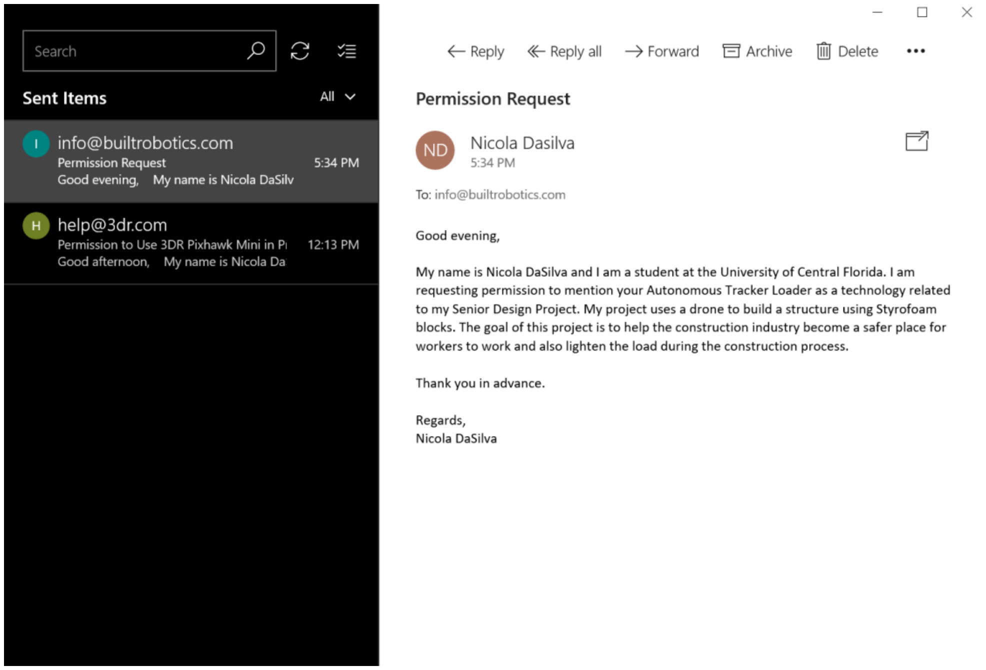


Figure 78: Autonomous Tractor Loader Permission Request

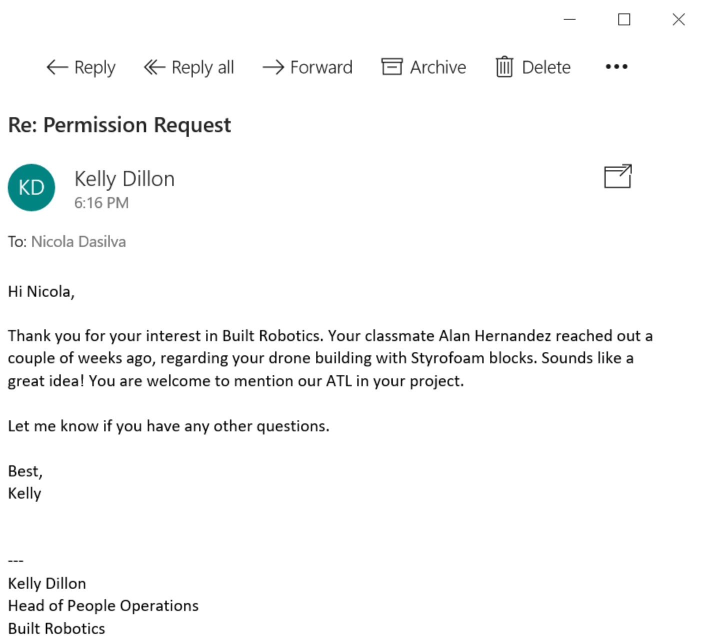


Figure 79: Autonomous Tractor Loader Reply

University Student Project - Content Permission Request

Alan Hernandez
Today, 7:17 AM
editor@hackaday.com

Reply all

Hi,
I'm a student at the University of Central Florida and I am working on a construction drone project for one of my course requirements. I'd like to request permission to an image and reference to the ground control base station article.
https://hackaday.com/2013/08/31/tearing-an-old-laptop-apart-to-build-a-ground-control-station/

Tearing an old laptop apart to build a ground control ...

hackaday.com

Being tired of assembling and disassembling parts/cables every time he went outside to fly his plane, [Elad] figured that he'd be better off building his ...

Figure 80: Ground Station Content Request

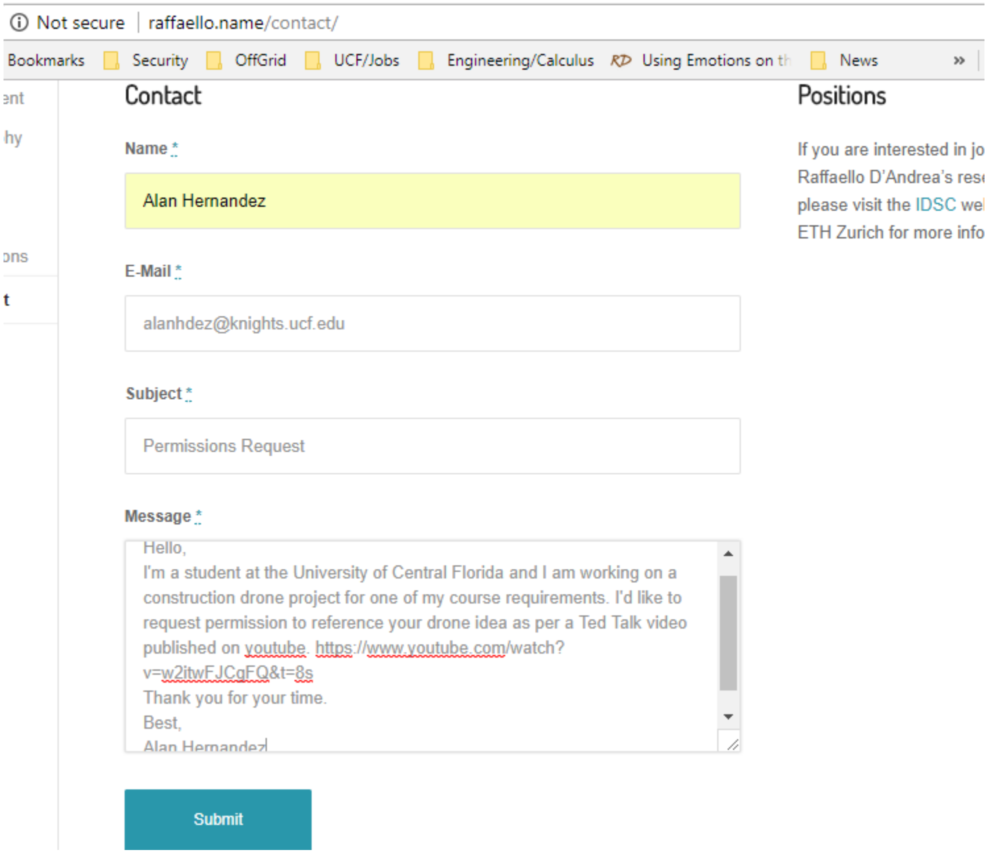


Figure 81: Construction Drone Request

*Special thanks to Precision Hawk for allowing our us to use their image as our cover photo for reference.