FXUAV
Senior Design II
Report Group 7

Group Members

Adam Kutchak
Luis Brum
Jamie Peck
Greg Kelso

Summer 2016

Sponsor: FLIR Systems

Table of Contents

1 Executive Summary

The Fire Extinguishing Unmanned Aerial Vehicle was an idea created between the group members after several hours of brainstorming of ideas using UAVs and infrared technology. Thus, we came up with an idea that incorporates both desired technologies, and a fire extinguishing drone, or FXUAV for short. The reason for the decision was based on the group members' desires for interacting with recent, advanced technology that would challenge and force each group member to learn something new about computer and electrical engineering, as well as learn about themselves and what their abilities are when it comes to research, development, and design. Additionally, ties with FLIR Systems also helped enforce the decision to use infrared technology, as it would help provide funding for our project, as well as require each member to research and learn subjects beyond the scope of our specified academic major but remain relevant to the scope of our general academic major, which is engineering.

Beyond our reasoning for the topic of our project is the description of our project, which also assists in defining how the project will actually challenge each group member. Essentially, the Fire Extinguishing UAV will be an autonomous unmanned aerial vehicle that detects the presence of a flame and suppresses it as such. The UAV used in this project can be defined as a quadcopter, which has four separate motor and propeller systems to navigate and produce other airborne activities. Attached to the quadcopter will be a Logitech camera to be used for image processing. This thermal imaging camera will be used to detect the thermal energy from the flame and will therefore be the means of the project of flame detection. There will be a ground unit that process the images from the Logitech camera and extracts the data from the raw image in order to locate the flame. This ground unit will also communicate to the flight control unit that controls the motors and propellers. By doing so, this will allow the ground unit to position the quadcopter directly over the flame. The last subsystem of the design is the fire suppressant release. Once the quadcopter is hovering directly above the flame, a signal will be sent to a servo from a specially design controller which will cause the servo to open or retract and release the chemical fire suppressant. At this point the suppressant will be released and will extinguish the fire.

Tasks were divided up into four components (one for each group member) and coincided with the main concerns and subsystems within the project. These items are listed as follows:

- Image processing to identify fire presence
- Design of Power Distribution
- Fire Suppression
- Mission Software

Currently, each member is researching and designing a solution to each concern. All parts of since been purchased and prototyping has begun. Part assembly has

also begun along with minimal testing. Some other minor aspects of the project that will be taken on by all other group members include the following

- Flight Stabilization
- Object Avoidance

2 Project Description

Group 7's project also known as FXUAV is a multirotor unmanned aerial fire suppressing system. This system is composed of three major parts and are listed as follows

- Quadcopter
- Ground Unit

These two units will work together to provide a successful termination of a predefined and controlled fire. The quadcopter will carry the camera and send images to the ground unit. The quadcopter is responsible for locating the fire and extinguishing the fire. Attached to the quadcopter is a webcam that will be used to detect the contour of a flame. Once a gathering of red pixels is detected, the servo signal will output high and open the door to the container which will release the chemical suppressant onto the fire.

Attached to the quadcopter will also be a companion computer that will run all of the scripts that control the quadcopter and release the payload. The companion computer is responsible for most of the onboard processing, versus our previously mentioned ground station doing all of the processing. Using MAVLink commands via serial connect, the companion computer will communicate with the Pixhawk flight controller.

The ground unit is responsible for executing the grid searching algorithm. It is also used to execute the fire detection algorithm, which then calls the fire extinguisher scripts to activate the servo.

2.1 Project Motivation and Goals

The objective of the Fire Extinguishing UAV is to aid the group in proving the concept that unmanned aerial vehicles can assist first responders for fire-safety and other life-saving fields. By detecting flame presence and aiding in flame suppressant, we seek to equip ourselves with design experience and knowledge, teamwork experience and abilities, and research and development skills, as well as design a system that could potentially be helpful in the future. If this concept can be proven, it could lead to less injuries, deaths, and losses caused by fires.
The motivation behind the FXUAV lies in the staggering amount of deaths, injuries, and financial losses caused by fires. According to the U.S. Fire Administration, within the first two months of 2016, there has been eight fire-related fatalities. Fire incidents were responsible for over 20,000 deaths and injuries and over $11 billion in damages in the United States in 2011, alone. In order to help drive the number

of deaths, injuries and financial burdens down, the FXUAV concept is our solution. By aiding first respondents in suppressing and controlling fires, the number of deaths due to structure collapse, smoke inhalation, and other associated injuries and deaths can be effectively reduced, and that is the motivation behind this project.

2.2 Project Requirements and Specifications

2.2.1 Functional Requirements

2.2.1.1 Quadcopter

In today's market there are hundreds of variations of flying drones that can be designed and implemented. Since the main goal for the FUAX system is to prove than an autonomous system can locate and extinguish a fire, group 7 has decided to build a flying copter with the following requirements:

- Be able to lift a payload up to 5 pounds, with the total weight not exceeding 10 pounds.
- Have a flight time of at least 10 minutes with a payload of at least 1 pound attached to the system.
- Capable of wirelessly interfacing with a ground control system that will control its flight system with a distance of up to 500ft.
- Autonomously hover through a 1000 square feet area to identify a fire that is at least 1 square inch in size within 5 minutes.
- Process images using an infrared camera system to locate and identify a fire.
- Ability to avoid objects in its flight path using ultrasonic proximity sensors.
- Integrate two single board computer units for flight and image processing.
- Wirelessly transfer images to a ground unit to process the image data from the infrared camera with a distance of 500 feet.
- Transfer telemetry of its flight data to a ground unit within 500 feet.
- Use a power distribution board to provide power to all the necessary equipment on board the Copter from a single source.
- Have a failsafe mechanism that will ground the unit in case of loss of power.
- Use a GPS system to identify its location with an accuracy of 4 meters.

Figure: 2.2.1.1A on the following page shows the diagram of the FXUAV basic components that will be needed to fully meet specifications.

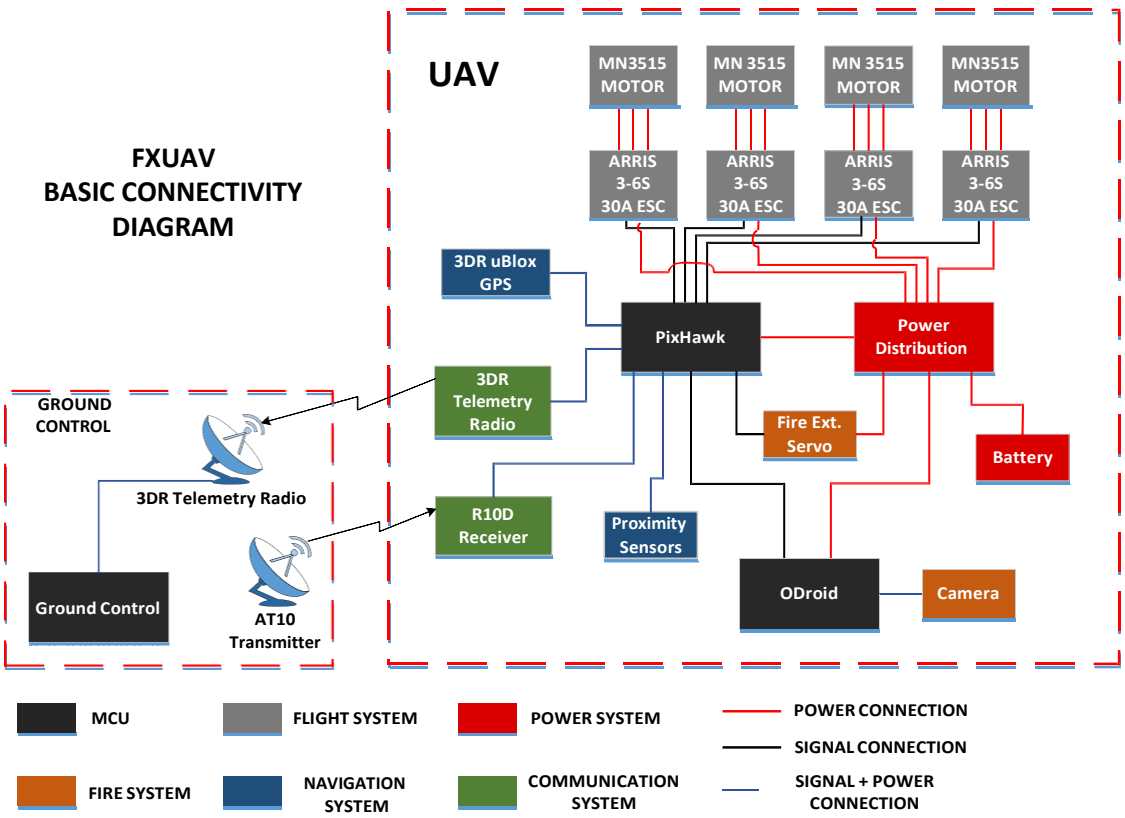


Figure 2.2.1.1A: Basic Functional Diagram of the FUAX system.

2.2.1.2 Fire Extinguishing System

The fire extinguishing system will be controlled by a servo. The servo will be controlled by a custom self-designed printed circuit board. Once the quadcopter receives the data that it is directly above the target, a signal will be sent to the custom printed circuit board that indicates to activate the servo. Once the servo receives the signal, it will retract, opening the contents of the container that is affixed to the quadcopter. The contents inside the container will be a chemical fire suppressant used in many fire extinguishers. The weight of the contents will be approximately 1lb so as to meet our requirements of extinguishing the fire, and also, keeping the quadcopter light enough to still fly. The material within the container will be Sodium Bicarbonate (baking soda). This is used in many BC fire extinguishers, and is the only dry chemical used in large scale for kitchens. Its properties are mostly effective towards grease fires, and small electrical fires. When the chemical heats up, it starts to foam, and effectively smothers the fire. The functional requirements of the fire extinguishing system are listed as follows:

- The system shall be able to communicate with the quadcopter's custom PCB
- The system shall be able to hold at least one pound of Sodium carbonate
- The system shall contain an electrically operated opening/closing device
- The system shall continuously hold its contents unless a release signal is received by the quadcopter

- The system shall remain affixed to the quadcopter at all times
- The system shall release the contents upon receiving a signal from the custom PCB

2.2.1.3 Image Processing and Fire Detection

A high quality camera is mounted on the UAV frame, more specifically, on the underside of the frame in order to ensure the most accurate alignment amongst the FXUAV and the tracked object. The camera is a Logitech C270 webcam that exhibits data streaming capabilities that is used to send data from the FXUAV to the ground control unit. The camera is suitable to provide accurate real time processing via a certain number of frames per second. The image processing module, which is located within the ground control unit, receives the streams of data outputted from the camera.

The image processing module is responsible for implementing both fire detection algorithms and fire tracking algorithms. When the image processing module first receives data from the camera via the onboard microprocessor, it implements the object detection algorithms to determine if the object is currently within the field of view or not. It then sends this data back to the onboard micro processing controller which then communicates with the flight controller.

Once the image processing unit detects the object within the cameras field of view, it then commences implementation of the module's tracking algorithms. The image processing unit then sends this data back to the onboard microprocessor which communicates with the flight controller in order to manipulate the position of the UAV in accordance with the location of the object. The image processing unit signals the onboard microprocessor once FXUAV is overtop the object; the object must be in the center of the camera's field of view which is calculated via the image processing algorithms. Once the image processing module signals the onboard microcontroller, the processor sends a signal to the fire extinguishing unit, in effect, releasing the payload.

2.2.2 Non Functional Requirements

2.2.2.1 Quadcopter

The drone system itself has several non-functional requirements that will serve to aid the team in developing a complete product, maintaining a goal vision that produces quality work, and allows room for creativity while designing, building, and testing the product. The drone system's non-functional requirements are listed as follows:

- The drone system shall be able to handle operating during a full day of testing without any parts overheating the system.
- The drone system shall maintain user controls easy enough to fly near flames during a full day of testing and sustain no damage to the system and its parts.

- The drone system shall be light enough and include motors powerful enough that several days of testing do not introduce significant wear and tear on the drone, causing emergency replacements to be needed in the middle of the development process.
- The drone system shall transmit data at a rate that is acceptable for a pilot to safely maneuver the drone around any obstacles, including fire, without damage to the system and its parts. The latency between the image source and the data processing should also allow for autonomous control of the system to maneuver without sustaining any damage to the system.
- The placement of the thermal imaging camera on the drone, in combination with the available field of view with the camera, will allow for the drone system to identify fire from a reasonable distance away, to allow for the system to have a realistic application outside of a black box testing environment.

The communications between the drone system and the operations center responsible for data processing is secure, in order to ensure that testing, demonstration, and general use of the drone system is possible without having a risk of outside interference.

2.2.2.2 Fire Extinguishing System

While the fire extinguishing system itself is a functional requirement of our entire system, there are some features of the fire extinguishing system that are need functional requirements in order for the entire system to operate as planned. These nonfunctional requirements are not essential in completing our goal, but will aid us in doing so. In other words, these nonfunctional requirements are for lack of a better term, “wish list items”. The nonfunctional requirements are listed as follows:

- It would help if the fire extinguishing system could hold 2 or more pounds of Sodium carbonate. This would help ensure that our goal of fire suppression is met
- It would be beneficial if the entire system without the contents inside weighed less than 5 pounds. This helps take load and stress off of the motors and allows battery availability for longer periods of time.
- It would be beneficial if system were placed top dead center over the flame before releasing the contents. This makes releasing the contents much easier without having to compensate with calculations.
- It would be beneficial for safety if there were a fail-safe or redundant servo

2.2.3 Quadcopter Specifications

The specifications of the fire extinguishing system explain the plan to meet the requirements. For each functional requirement, there is a plan to implement it. These plans or specifications are listed as follows:

- The motors chosen for this drone system, the Tiger RC MT3515 KV400, are capable of providing the required 10lbs of lift for a sustained 10 minutes,

while operating at far less than 100% throttle. Additional loads and longer flight times would be accomplished easily, but will begin to introduce power consumption tradeoffs.

- The drone will use RC transmitters to communicate data, which operate in the 2.4GHz bandwidth. This will allow for up to 1km of communications, which easily meets the requirement of 500ft. The telemetry communications will operate at 915MHz, and the video communications will operate at 5.8GHz, allowing the drone to communicate at all distances that will be required during development.
- The Mission Planning software that ships with the Pixhawk flight controller offers several pre-built functions to aid with flight control, in addition to having easy to use customization options. The addition of using Python scripts to customize the flight control will allow the drone system to move towards fires, hover at a required distance long enough to extinguish the fire, and any other maneuvers required during testing.
- The Logitech HD C270 camera with image detection software will allow for easy identification of fire anywhere inside the lens' field of view.
- HC-SR04 Ultrasonic sensors will be used to aid the drone system in understanding the environment around it. The sensor data will be used to program the drone so that it is able to autonomously avoid obstacles and ensure flight that does not sustain damage to the drone.
- The 3DR uBlox GPS will allow accurate positioning up to +/- 2.5m while providing a very low amount of extra weight and power consumption to the drone system.

A 6000mAh LiPo battery is used, and the power allows the drone system to sustain long flight times.

2.2.4 Fire Extinguishing Specifications

The specifications of the fire extinguishing system explain the plan to meet the requirements. For each functional requirement, there is a plan to implement it. These plans or specifications are listed as follows:

- The system will communicate with the quadcopter's custom secondary PCB. A servo will be physically attached to the controller. When the controller sends the signal the servo will retract.
- In order to hold at least a half-pound of sodium carbonate, the container will be made of plastic and have a volume of 50 in^3. ½ teaspoon of baking soda weighs 3 grams. There are roughly 454 grams in one pound. 454 divided by two gives us 227 grams for one half pound of baking soda. Dividing 227 by 3 gives us 76, ½ teaspoons. Dividing 76 by two gives us 38 teaspoons, which translates to roughly .8 US cups, which will fit in an 11.5 cubic inch storage container.
- In order to control the system electronically, a servo will be used to open and close the container. The container will be PTO, or power to open. Once the servo receives the signal, it will open the container.

The entire system will be attached to the quadcopter by some means of adhesive or a substantial amount of zip ties.

2.2.5 Realistic Constraints

As with any project, there are constraints. These constraints range from time, money, environment, ethics and plenty of other resources. Resource constraints typically play a major role in projects. These resources include time, money, real estate, and even man-power.

2.2.5.1 Time and Economic Restraints

Time and economic restraints are of the most important for this particular project. These constraints have greatly impacted our design decisions and have thus, limited us as such. When taking time into consideration, we have only a total of approximately six and a half months to come up with our idea, research solutions and implementation ideas, prototype, document, implement, test, and deploy our system. Most of which will be spent researching, prototyping, and documenting. This leaves a small window for actual implementation testing, and deployment. With that in mind, we have developed our project and assigned tasks appropriate for each group member to complete with sufficient time, while we chose to purchase some components outright in order to save time versus building every component ourselves. Not to mention, mechanical and structural engineering are not our concerns, so we decided it would be a good idea to buy the quadcopter frame versus building one.

Economic restraints also come into play here, and can someone snowball when it comes to time constraints. With limited time, we must find the quickest and most efficient way to complete tasks. Often times, the result is purchasing components that are already assembled. Case in point, the quadcopter flight controller. However, we must be careful with which we choose to purchase, because buying all of the components will not only leave us with nothing to actually design, it will leave our pockets empty. Luckily our sponsor FLIR was generous enough to fund our project, up to a certain financial figure. There is a balance between time and money and one comes at the expense of the other when dealing with projects. One key decision we made as a result of time and money constraints is our decision to eliminate a camera gimbal. A gimbal would likely add complication to our project and take up more time that doesn't contribute much to the design aspect. Additionally, the gimbal can cost up to several hundred dollars. Therefore, we decided we were better off without one and adjusted our design accordingly.

2.2.5.2 Environmental and Political Constraints

When designing our quadcopter and its implementation, we need to take into account environmental factors. Factors such as rain, wind, and the sun can all greatly impact the direction we choose to take with our project. Rain is never a good element to introduce when dealing with electronics. However, indoor testing becomes difficult when trying to receive a GPS signal.

Wind can also affect how our system behaves during testing and deployment. Strong wind will compromise our chemical release function such that, the wind will disperse the chemical until it is no longer affective against our target.

Suns radiation can also affect our project when considering thermal imaging. When there are shiny metal objects in range of the camera, they tend to reflect the suns thermal signature and can trick the camera into false detections. Luckily, fire has a unique thermal signature which we can differentiate from the suns energy.

Elements can really cause problems when designing our system, therefore we've decided to shoot for outdoor deployment, without a requirement that the system must operate in all weather conditions. This allows us to perform our deployment and demonstration in close to ideal weather conditions.

Certain political constraints can have adverse effects on our design choices as well. These constraints include regulations put in place by the FAA. Such as, restrictions on legal locations to fly quadcopters, as well as size limits. Public UAVs must be within a certain size limit, otherwise it cannot legally be handled publically. Additionally, the International Traffic in Arms Regulations made it difficult to get our hands on the FLIR Tau modules. There are some models that are considered ITAR items, including the 30Hz FLIR Tau camera. Since the 9 Hz option is less common, we found it difficult getting the module in time, which resulted in us turning to the Logitech C270 webcam.

2.2.5.3 Ethical and Safety Constraints

Equally important as the previously mentioned constraints, is ethical constraints. Ethical constraints can be defined as a guideline to direct "normal" society behavior. For instance, if there were another project similar to the FXUAV, and we claimed it as our own, that would be considered unethical. Sometimes ethical issues can evolve into legal issues, other times, they are considered rude. Once realistic ethical constraint we have encountered is privacy. There aren't any laws or regulations in place that prohibit UAVs from entering public airspace, even if the space is located above an individual's home. This is where legal and ethical differ. It is ethical to respect privacy of others, but there are no legally binding documents stating we are in violation for flying our project over someone's home.

Safety constraints are also a big concern with UAVs. UAV incidents can be dangerous when not handled properly around groups of people or animals. The carbon fiber propellers, although light, are still very strong and will be rotating up to 6500 RPM, fast and strong enough to physically sever extremities. Therefore, testing will need to be done carefully in a completely controlled environment.

2.2.6 Standards

2.2.6.1 IEEE Standards and Regulations

An organization within IEEE, The Institute of Electrical and Electronics Engineers Standards Association (IEEE-SA), is responsible for cultivating a robust set of standards and regulations applicable to a broad range of technical industries. Their resulting widely accepted consensus is what drives the proficiency and functionality of worldwide technologies. Below is the list of IEEE standards that provide the specifications and procedures that facilitated the development of the design, functionality, and usage of the FXUAV system.

1. 802.15.4j-2013 - IEEE Standard for Local and metropolitan area networks - Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 4: Alternative Physical Layer Extension to Support Medical Body Area Network (MBAN) Services Operating in the 2360 MHz – 2400 MHz Band

2. 802.1AB-2016 - IEEE Standard for Local and metropolitan area networks - Station and Media Access Control Connectivity Discovery

3. 802.11b/Cor 1-2001 - IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher Speed Physical Layer (PHY) Extension in the 2.4 GHz Band - Corrigendum 1

4. 802.11g-2003 - IEEE Standard for Information technology-- Local and metropolitan area networks-- Specific requirements-- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Further Higher Data Rate Extension in the 2.4 GHz Band

5. 802.11n-2009 - IEEE Standard for Information technology-- Local and metropolitan area networks-- Specific requirements-- Part 11: Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput
6. 802.15.4-2011 - IEEE Standard for Local and metropolitan area networks-- Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)

7. 172-1983 - IEEE Standard Definitions of Navigation Aid Terms

8. 8130.34B - FAA Order Airworthiness Certification of Unmanned Aircraft Systems and Optionally Piloted Aircraft

9. 1453.1-2012 - IEEE Guide--Adoption of IEC/TR 61000-3-7:2008, Electromagnetic compatibility (EMC)--Limits--Assessment of emission limits for the connection of fluctuating installations to MV, HV and EHV power systems

10. C95.4-2002 - IEEE Recommended Practice for Determining Safe Distances From Radio Frequency Transmitting Antennas When Using Electric Blasting Caps During Explosive Operations

11. 610.4-1990 - IEEE Standard Glossary of Image Processing and Pattern Recognition Terminology

12. 730.1-1995 - IEEE Guide for Software Quality Assurance Planning

13. 1667-2015 - IEEE Standard for Discovery, Authentication, and Authorization in Host Attachments of Storage Devices

14. C37.111-2013 - IEEE/IEC Measuring relays and protection equipment – Part 24: Common format for transient data exchange (COMTRADE) for power systems

15. 1521-2003 - IEEE Standard for Measurement of Video Jitter and Wander

16. 900.1a-2012 - IEEE Standard Definitions and Concepts for Dynamic Spectrum Access: Terminology Relating to Emerging Wireless Networks, System Functionality, and Spectrum Management Amendment 1: Addition of New Terms and Associated Definitions

17. 1900.6a-2014 - IEEE Standard for Spectrum Sensing Interfaces and Data Structures for Dynamic Spectrum Access and Other Advanced Radio Communication Systems - Amendment 1: Procedures, Protocols, and Data Archive Enhanced Interfaces.

2.2.6.2 FCC Standards and Regulations

The FCC, Federal Communications Commission is a government association responsible for regulating a multitude of communication technologies and efforts. This regulation includes radio communications which has affected the design and development of the FXUAV system in several areas including our ground system communication with the BeagleBone Black microcontroller and the Pixhawk Flight Controller.

Title 47: Telecommunications

Chapter 1—Federal Communications Commission

Part 2—Frequency Allocations and Radio Treaty Matters; General Rulers and Regulation

1. **Subpart B—Allocation, Assignment, and Use of Radio Frequencies**
    a. §2.101 Frequency and wavelength bands
2. **Subpart C—Emissions**
    a. §2.201 Emission, modulation, and transmission characteristics
3. **Subpart I—Marketing of Radio-frequency Devices**
    a. §2.803 Marketing of radio frequency devices prior to equipment authorization
    b. §2.805 Operation of radio frequency devices prior to equipment authorization
    c. §2.815 External radio frequency power amplifiers

Title 47: Telecommunications

Chapter 1—Federal Communications Commission

Subchapter A—General

Part 15—Radio Frequency Devices

1. **Subpart A—General**
    a. §15.5 General conditions of operation
    b. §15.9 Prohibition against eavesdropping
    c. §15.15 General technical requirements
    d. §15.17 Susceptibility to interference
    e. §15.23 Home-built devices
    f. §15.25 Kits
    g. §15.27 Special accessories
    h. §15.32 Test procedures for CPU boards and computer power supplies
2. **Subpart B—Unintentional Radiators**
    a. §15.109 Radiated emission limits
    b. §15.111 Antenna power conduction limits for receivers
    c. §15.121 Scanning receivers and frequency converters used with scanning receivers
3. **Subpart C—Intentional Radiators**

a. §15.205 Restricted bands of operation
b. §15.212 Modular transmitters

4. **Subpart C—Unlicensed Personal Communications Service Devices**
    a. §15.305 Equipment authorization requirement
    b. §15.317 Antenna requirement
    c. §15.319 General technical requirements
5. **Subpart F—Ultra-Wideband Operations**
    a. §15.511 Technical requirements for surveillance systems

§15.525 Coordination requirements

3 Research

3.1 Research of Similar Projects and Existing System

There are several similar ideas to the FXUAV that are already currently in existence or that are in early to late stages of product development. Scientific Systems Company, Inc. (SSCI) for example, is collaborating with several colleges in an effort to demonstrate the capabilities of unmanned, aerial vehicles (UAVs) to autonomously track and monitor forest fires from an aerial perspective. The company envisions being the country's leading provider for technologies that will aid support of hazardous and threatening operations. This industry-university team including Massachusetts Institute of Technology (MIT) and Olin College, announced that they have successfully executed multiple, preliminary flight tests which have substantiated not only autonomous monitoring tracking technologies, but such technologies designed specifically for real-time applications of forest fire investigations. The system developed by SSCI, MIT, and Olin College was designed to extract three pieces of key information from its aerial fire surveillance efforts. One of the initial flight tests that were designed, was created to determine how successful the autonomous system was in extracting these three pieces of vital information—fire location, its rate of speed, and its extent. In support of this information the system was also responsible for extracting wind intensity and direction data, terrain data, and vegetation data, all details that affect the main priority of the systems wildfire monitoring and tracking duties.

SSCI's, MIT's, and Olin College's vision is to create a system of integrated UAVs that will collaborate in the effort to provide a fire commander with the necessary details and information about the wildfire, of which are vital to the fire department's success in locating, and exterminating the fire in safe, but timely manner. Already developed, by this highly skilled team of developers and engineers, is the software implementation package. This package includes both the planning and control interface code for the system's ground station. Furthermore, the team has rendered, successfully, the development of the onboard software responsible for attributing to the system its autonomous features and capabilities as well as the system's coordination software for the use of multiple UAVs. The success of this project is due primarily to the development of "machine-learning" (Scientific Systems Company, Inc.) technologies for fire spread rate projections as well as a

series of algorithms for autonomous cooperation among multiply UAVs for fire search missions, geo-location, feedback-control surveillance, and edge tracking abilities in real time as well as when under certain contingencies. That being said, the technologies that have stemmed from the collaborative work of SSCIs, MIT and Olin College in this UAV effort, have been said to be both the brain and nervous system of most leading edge technology in the areas of UAV autonomous operation today.

Similarly, ELIMCO'S E300 with FENIX is a fire tracking UAV designed specifically for night missions. In an effort to maximize the prestige and usability of the proposed system, a key feature incorporated by engineers during the design phase was the development of its electrical, low-noise propulsion. The system can be launched remotely and venture as far as 62 miles from its point of departure. FENIX is the system's planning and monitoring software system that is responsible for real-time transmission of geo-tagged images and video from the E300 to the company's command headquarters. Then, in real time using a mapping service to decode the geo-tagged data that was received via the FENIX software system, the operator is notified of the fire's location and address. This system was created for the purpose of improving wildland firefighting during night time hours in an effort to stop the discontinuing efforts of firefighters at night which allows for the quick expansion of wildfires.

The E300 scans the contaminated area by flying directly over the fire. It therefore implements its monitoring and tracking efforts via a downward looking camera. Their camera of choice can record both video and thermal images which it then geo-tags for special reference and transmits them the mobile ground unit.

L3 Communications developed a similar system as well, known as the Viking 400-S. The Viking 400-S is an Autonomous Unmanned Aircraft Systems designed specifically to carry a large payload to support the detection of hazmat crises. These Autonomous Unmanned Aircraft Systems can hold up to 100 pounds of lucrative detection technology ranging from nuclear to biological detectors. The system has the capability to send chemical, biological, radiological, and nuclear (CBRN) data wirelessly to CBRN Operations Specialists to minimize the exposure of hazardous materials to first responders.

How ours is different:

The above, defined systems, together, provide a basis or foundation for the FXUAV. Below is a list of specific aspects that exist in all or one of the above described systems that provide a strong baseline for the FXUAV, preliminary design implementations that the FXUAV will employ.

- Autonomous Aerial Monitoring
- Autonomous Aerial Tracking
- Autonomous Aerial Detection of Fire
- Ground Communication Station
- Wireless Feedback
- Geo-Tagged Locations
- Real Time Feedback

- Thermal Imaging Camera
- Situational Awareness

Although these systems provide a robust foundation for the idea of the FXUAV, further implementation needs to be done in order to accomplish the task of extinguishing the hazard. With that being said, one of the main aspects of the FXUAV design is in response to the research finds above or the lack thereof, if you will. The FXUAV will not only incorporate every feature in the above list, but it will have the insurmountable ability to also diminish or terminate the hazard in which it is detecting and tracking. Neither of the above systems has implemented an extinguishing mechanism to mitigate the hazard in which it is deployed to survey and therefore the FXUAV came about.

3.2 Research of Possible Solutions

3.2.1 Quadcopter

3.2.1.1 Frame

One of the primary decision to make when building a multirotor copter is what kind of frame to use. The frame configuration will contribute to multiple factors of the design of the multirotor copter, such as maneuverability, total payload, numbers of motors to use, flight software configuration, and many more. Frames can be made from various types of materials such as, Fiberglass, aluminum, plastic, wood, and etc. Since the total weight of the multirotor copter is a big constraint when building a multirotor, fiberglass is usually the most popular material used in the construction of a multirotor frame. Another key factor of the frame used for the multirotor is the base size of the frame itself, as shown in Figure: 3.2.1.1A the base size is measured from one motor to the motor directly across from it. The base size of the motor in most multirotor configuration will put a constraint on the propeller size and the horizontal clearance needed for the multirotor to fly through an area.

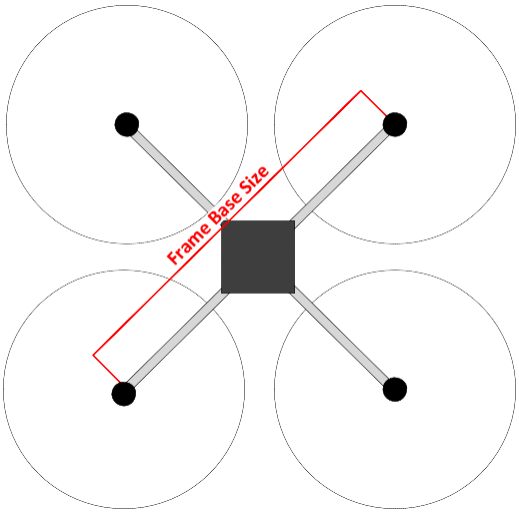


Figure: 3.2.1.1A Quad X Frame Base Size.

Multirotor Copters are classified by the number of motors it uses, for example: Quadcopters uses four motors, HexaCopters uses six motors, OctoCopters uses eight motors, and etc. Frames in the other hand are classified their shape, and how many motors it supports. Different shapes all have their benefits and downsides, for this project the best three configurations are the Quad X, the HEX6 X and the V8 configuration. Figure 3.2.1.1B shows a few of the different configurations researched for the design of our multirotor copter.

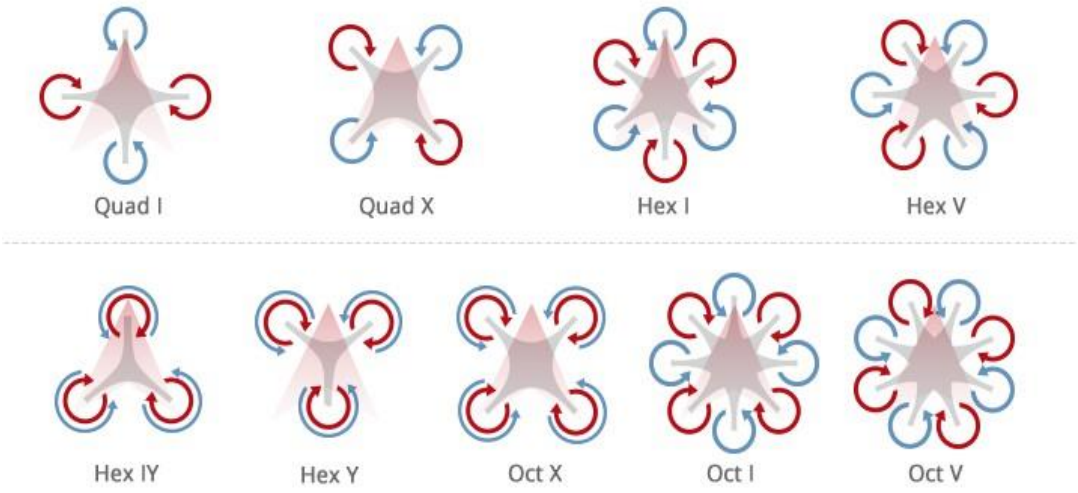


Figure: 3.2.1.1B Basic multirotor frame configurations. Reprinted with permission from DJI. (Pending)

The Quadcopter X is one of the most popular frame design used to build multirotor copters for consumer use. Its configuration allows users to carry a small payload, and achieve a flight time of anywhere between 10 minutes to 30 minutes without increasing the cost to build too much. To increase the capable payload with this design larger propellers and slower turning motors are needed, which decreases the top achievable speed of the multirotor. Since the Hexacopter and Octocopter are capable of carrying bigger payloads with more efficiency they are more popular for commercial applications, but because they require more hardware, their cost of built is much greater than a Quadcopter. Because with each additional motor you increase your thrust to weight factor, the payload and time of flight increases tremendously when compared to a Quadcopter design. They are also better for fail safe designs because unlike the Quadcopter, they can still fly if an engine failure was to occur during flight. Since this project is only to prove an autonomous vehicle can be designed to locate and extinguish a fire, a pre-fabricated quadcopter frame was chosen for the design due to budget constraints.

3.2.1.2 Motor

Brushless electric motors are what is used in the industry for the majority of multirotor designs, due to their basically frictionless design they are very efficient and reliable for the strain they will be put through. One of the most important factor

when choosing a motor to use in the design of a multirotor, is the overall weight of the system. As specified in the requirements for this project the total weight of the system will be 10 pounds, this requirement limits the options of motors available for the design of the multirotor.

There are several rating factors to an electric brushless motor for a multirotor system, the KV rating, diameter height, maximum operating temperature, and the magnets in the rotor and stator of the motor. These factors determine the total thrust power, efficiency, the power needed, and the maximum propeller size that should be used with the motor. The main rating factor to look at when choosing the right motor and propeller combination is the thrust to weight ratio, in general you to throttle your engine at 50% of the maximum thrust per engine of the multirotor.

When shopping for motors, there are data tables provided from the supplier with test data results of the specific motor with different propellers and power supplied. These data tests are the most reliable source of information when purchasing a motor and this is what was used when choosing the motors needed for the FUAX. After comparing numerous motor test data, the final three choices for the motor used in the FXUAV are shown on the Table 3.2.1.2A below.

FXUAV MOTOR COMPARISON							
MOTOR	VOLTS (V)	THROTTLE	PROPELLER	AMPS (A)	WATTS (W)	THRUST (G)	RPM
TIGER RC MT3515 KV400	22.2	50%	T-MOTOR 15*5CF	4.3	87	970	4200
TIGER RC MT3515 KV600	22.2	50%	T-MOTOR 13*4.4CF	8.7	193	1200	7000
EMAX MT3515 KV 650	14.8	50%	APC 15*4CF	12	177.6	1270	5710

Table: 3.2.1.2A FXUAV Motor Comparison

When making the final decision for the FXUAV the power consumption was the deciding factor. As shown on Table 3.2.1.2A, the Tiger RC MT3515 KV400 was most efficient out of the three options only using 87 Watts per engine at 50% throttle. By choosing the most efficient motor, the flight time of the FXUAV can be maximized to achieve the required flight time of at least 10 minutes while carry the maximum payload calculated per the specification. Since the test data for the motor is provide with the specific propeller combination, the propeller used for the FXUAV was chosen according to the test data. The full test data and specification for the motor selected for the FXUAV is shown on Figure 3.2.1.2B below:

Specifications:

KV	400
Configu ration	12N14P
Stator Diameter	35mm
Stator Length	15mm
Shaft Diameter	5mm
Motor Dimensions(Dia.*Len)	Φ42.5×37.5mm
Weight (g)	188g
Idle current(10)@10v(A)	0.3A
No.of Cells(Lipo)	3-8S
Max Continuous current(A)180S	30A
Max Continuous Power(W)180S	900W
Max. efficiency current	(10-18A)>80%
internal resistance	116mΩ

Item No.	Volts (V)	Prop	Throttle	Amps (A)	Watts (W)	Thrust (g)	RPM	Efficiency (g/W)	Operating temperature(℃)
MT3515 KV400	22.2	T-MOTOR 14*4.8CF	50%	3.4	76	780	4300	10.26	45
			65%	6.3	140	1180	5400	8.43	
			75%	8.5	192	1480	5900	7.71	
			85%	11.4	255	1800	6500	7.06	
			100%	13.7	301	2010	6950	6.68	
		T-MOTOR 15*5CF	50%	4.3	87	970	4200	11.15	50
			65%	7.9	177	1470	5200	8.31	
			75%	11.6	254	1880	5700	7.40	
			85%	14.5	322	2200	6200	6.83	
			100%	17.2	376	2460	6500	6.54	
		T-MOTOR 16*5.4CF	50%	5.2	117	1180	4050	10.09	59
			65%	9.4	210	1730	4850	8.24	
			75%	13	288	2100	5400	7.29	
			85%	16.9	372	2630	5850	7.07	
			100%	20	437	2830	6250	6.48	

Notes:The test condition of temperature is motor surface temperature in 100% throttle while the motor run 10 min.

Figure 3.2.1.2B Tiger RC MT3515 KV400 Motor Specification. Permission from T-Motor. (Pending)

3.2.1.3 Propeller

Believe it or not, propellers are not all made the same. In fact, in order to stabilize the yaw rotation two propellers must rotate in opposite directions. However, when purchasing a set, usually they come with the appropriate propellers. Where they really start to differ is the material. There are several different materials when it comes to props, but the main ones discussed are usually carbon fiber, plastic, and wooden propellers.

Carbon Fiber props are very popular and generally very successful when it comes to quadcopters. Since carbon fiber is much stronger than plastic, it gives confidence to the person flying it. However, that can turn into a negative in some cases. If the quadcopter were to crash, since the propellers are less likely to break, more stress will be put on the motor bearing. Additionally, carbon fiber is much lighter in weight which translates to less inertia and can prove to respond the motor speed changes quicker. Carbon fiber props also produce less noise and vibration which is important when image processing is a major aspect of the quadcopter project. Another issue is that carbon fiber props tend to be more expensive than the other materials. Another thing to note is that generally, carbon fiber props use less voltage, current, input power and battery capacity when compared to plastic propellers.

Wooden props are just too heavy, and not as popular as carbon fiber or plastic. Since weight is a large component to the project, wooden props will most likely not be a good idea for this application.

3.2.1.4 Flight Controller

There are dozens of different flight controller boards available for multirotor copters, the type of flight controller that is best for the multirotor highly depends on the platform being used and what the multirotor will be used for. The first deciding factor is what type of platform that will be used for controlling the device, and for this project we chose to use the Ardupilot platform. The Ardupilot platform is based on the Arduino open-source electronics and it provides open source hardware, firmware, and software that can fully control multirotor copters. Other factors to

take into consideration is the number of propellers being used, additional interfaces needed, total power output, flight sensors, and the processing capability. Most of these factors are determined by the purpose that the drone will be built for, and physical restraints of the multirotor. Within our budget constraint, Table 3.2.1.4A shows the different specifications we looked at for the FXUAV Flight Controller in terms of processor, memory and sensors.

Flight Controller Comparison Table Processor Memory and Sensor			
Component	**Pixhawk**	**PixRacer**	**NAVIO+**
Processor	32-bit ARM Cortex M4 core with FPU	Core: ARM® 32-bit Cortex®-M4 CPU with FPU	900Mhz quad-core ARM Cortext-A7 CPU
Memory	256 KB RAM, 2MB Flash	2 MB of Flash, 256+4 KB of SRAM including 64-KB of CCM, 256-Kbit	1GB RAM
Sensors	MPU6000, ST Micro 16-bit gyroscope, ST Micro 14-bit accelerometer/compass, MEAS barometer	MPU9250 Accel / Gyro / Mag (4 KHz), ICM-20608 Accel / Gyro (4 KHz), MS5611 Barometer, Honeywell HMC5983 Compass	MPU9250 as main accel, gyro and compass, MS5611 barometer, U-Blox M8N GPS

Table: 3.2.1.4A Flight Controller Processor, Memory and Sensor Comparison

After analyzing the different options available for the design of the FXUAV, the Pixhawk was the best choice for the different options we needed. This is due to the fact that it contains the more serial interfaces than any other flight controller, which will be needed to add the sensors necessary to make the FXUAV be able to avoid obstacles as required per the specification of the project. Since it is an open-hardware and open-hardware flight controller, it can be easily modified to achieve the necessary requirements of the FXUAV.

3.2.1.5 Power Distribution

Because of the way Brushless motors are designed, electronic Speed controllers are key components necessary to make brushless motors work properly. Since brushless motors work using electromagnetic charge to spin the rotor or the stator. The Pixhawk Flight Controller board needs an electronic controller that switches the DC signal supplied by the battery, and changes it to an AC signal that uses a feedback system to time, increase, or decrease the different charges accordingly.

The first and main priority when selecting an ESC is the maximum current draw and the total voltage each unit can withstand. For the RC Tiger MT3515 KV400 motor we have selected, the maximum current draw is 30 Amps with 22.2 nominal

Volts applied. Other features to consider while buying any ESC units are the firmware it uses, the refresh rate of the processor, and the possible connection interfaces for programming the unit. Table 3.2.1.5A shows the different Electronic Speed Controllers that were evaluated for this project.

ESC Comparison Table				
Model Name	Maximum Current		Voltage Input	Motor Rotation
	Contiuous	Burst		
ZTW Mantis	45A	65A	2S-6S	yes
ZTW Gecko	45A	65A	2S-6S	yes
XRotor PRO	40A	60A	3S-6S	no
Harrier-eXTrem	45	65A	2S-6S	yes

Table 3.2.1.5A: ESC Comparison Table

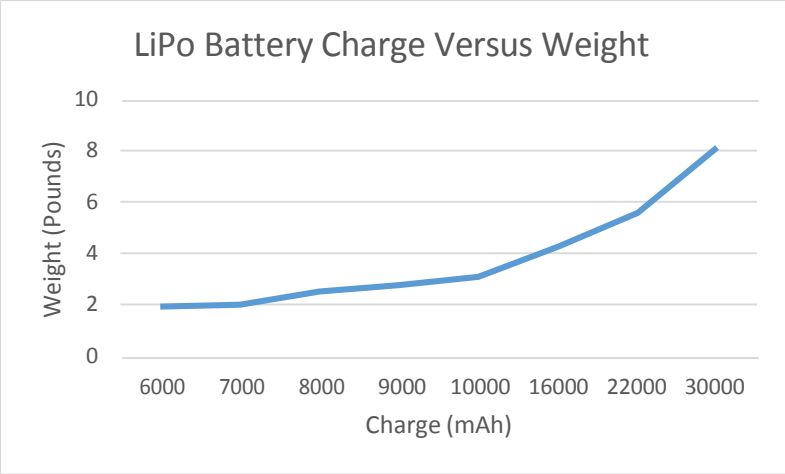
The electronic speed controller that was chosen for the FXUAV project was the XRotor PRO. After analyzing the four different electronic speed controllers that will work for the MT3515 KV400, the XRotor PRO will meet all the minimum necessary requirements and will fit within the project budget.

Since multirotor copters are very inefficient aerodynamically, it takes a great amount power to get them to fly and remain in the air. In fact power supply is one of the biggest problem for commercial multirotor systems, and in average the maximum amount of flight time is anywhere from 5 to 30 minutes in the consumer grade multirotor copters. The standard battery technology used for multirotor copter is the Lithium-Ion Polymer (LiPo) batteries.

The first main reason why LiPo batteries are popular in multirotor copters, is due to their high Gravimetric Energy Density. The Gravimetric Energy Density is a ratio of the battery's Watthours per Kilograms, which in other words means how much power per weight can the battery hold. Another important feature of the LiPo battery is that the cell in the batteries are Solid State cells, meaning that they do not need any liquid in them to produce power. Because LiPo batteries do not use any liquids or acids, it is a lot easier and lighter to package the cells together safely. The cells used in the LiPo battery can be produced in many different shapes or sizes, and this is another great feature of the LiPo batteries.

The cells used in the LiPo batteries that are used in multirotor copters are square and each cell holds a nominal voltage of 3.7V. The individual cells are connected in series, so the number of cell in series in the battery depends on the voltage needed by the system. Since the MT3515 KV400 motors require 22.2 nominal Volts, the FXUAV needs a six cell battery connected in series. The battery voltage is either displayed on the battery by its nominal voltage, or by the number of cells connected in series which in this case the battery would show a 6S on its packaging.

The second important factor of a battery, is the total amperage charge it holds, and how fast it can discharge it. LiPo batteries can come with a total charge of anywhere from 1000mAh to 30,000mAh, which determines the total power the battery can output. Because of the fact that the weight of the battery is linearly proportionate to the amount of charge it holds, the right size should be considered depending on the purpose of the multirotor. Graph 3.2.1.5B below shows different sizes of batteries versus their weight. The data for this chart was acquired from various manufacturers. The weight of each different total charge capacity may vary a total of +/- 20%.



Graph 3.2.1.5B: Lipo Charge versus Weight.

After calculating the total amount of charge is needed for this project, the next part is to identify the total amount of amps it can discharge. This discharge rating represented by a letter C, is a measurement included by manufactures that identifies the total rate of power the battery can safely discharge. The total safe discharge rate can be calculated by multiplying the C rating and the total charge. The rate of discharge depends on the total charge drawn from all of the loads attached to the battery, and for the FXUAV all the different systems. Table 3.2.1.5C below shows a comparison in the different LiPo battery total charge, discharge rating, weight, and price for 6S Lipo batteries that would at least meet minimum requirements for the FXUAV.

This also lead us into purchasing 2 separate batteries. Two 6,000 mAH batteries for testing and one 10,000 mAH battery for the final presentation. This was decided to ensure that we could have enough power supply if we needed to make multiple runs.

FXUAV Battery Comparison					
Lipo 6s Battery	Total Charge (mAh)	Discharge Rating (C)	Total Discharge (Amps)	Weight (Pounds)	Price (Dollars)
TATTU	6000	35	210	1.92	$150.92
TATTU	16000	15	240	4.25	$340.00
TATTU	22000	25	550	5.60	$461.89
QuadroPower	6000	35	210	1.91	$174.99
QuadroPower	10000	25	250	2.99	$174.99
QuadroPower	16000	15	240	4.28	$295.00
Lumenier	8000	25	200	2.37	$139.99
Lumenier	10000	25	250	2.74	$179.99
Lumenier	12000	20	240	3.10	$209.99

Table 3.2.1.5C: FXUAV Battery Comparison

After analyzing the different options of batteries, the battery chosen for the FXUAV was the QuadroPower 6S 10,000 mAh. This battery will meet all of the necessary requirements of this project, and will fit within the budget constraints of the project.

Multirotor copter systems have many different electrical components that have various different power requirements. There are multiple ways to deal with this problem such as provide each component its own power source, or have a power source that can provide power to multiple equipment. Because of the strict weight restrictions of multirotor systems, Power distribution boards can be a very important part of a multirotor copter system. With a power distribution board, we can provide power to different equipment with different power requirements. There are there are power distribution boards available in the market for most of the consumer multirotor copter design requirements. Because of requirements for this project, the FXUAV will have a power distribution board that is designed by Group 7 member Luis Brum. Table 3.2.1.5D below, shows a preliminary power requirement from the equipment that will be used by the FXUAV.

FXUAV Preliminary Power Requirements		
Equipment	Voltage (Volts)	Current (Amps)
Flight Control	5.7	1
Battery Sensor	20	.05
ESC	22.2	30
ESC	22.2	30
ESC	22.2	30
ESC	22.2	30
MCU	5.7	1
Video Transmitter	9	0.2
Telemetry Radio	5	0.1
Camera	5	1
Fire Control System	9	1

Table 3.2.1.5D: FXUAV Preliminary Power Requirements

The Power distribution board will be built using Texas instruments adjustable voltage regulators. The decision to use Texas instrument regulator was because of the use of Texas instruments components in the Electronics laboratories. Since there are a lot of free schematics and support by Texas instruments for their product, this will help simplify the design and assure the design will work as needed. Figure 3.2.1.5E below shows a basic schematic of a linear voltage regulator circuit that can be used to regulate the voltage from a single source to different loads.

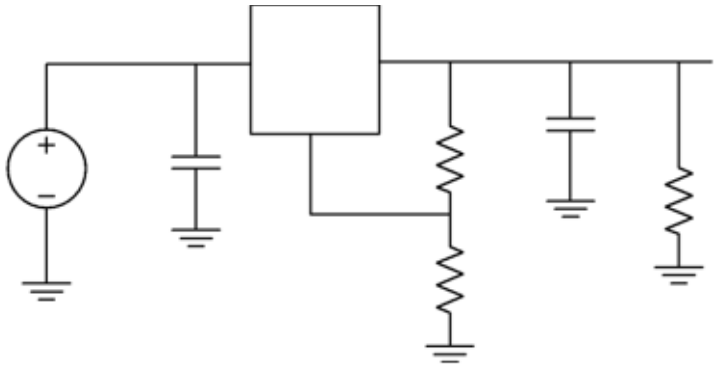


Figure 3.2.1.5E: Basic Voltage Regulator Circuit Schematic.

The power distribution board will also have a circuit built into it that will monitor the battery level, the sensor will alarm a buzzer and signal the battery status to the flight controller. This will allow the FXUAV to operate safely, and make sure no components and the battery are damaged. The low voltage sensor will allow the FXUAV to have 1 minute to land after the battery the alarm signal has been activated, this should provide enough time for the FXUAV to safely land. This safety feature can be accomplished by using a voltage comparator circuit similar to the ones studied in Electronics 2 laboratory experiments. Figure 3.2.1.5F: below shows a basic comparator circuit can drive voltage to a buzzer and signal the microprocessor when the battery's voltage drops below an acceptable magnitude.

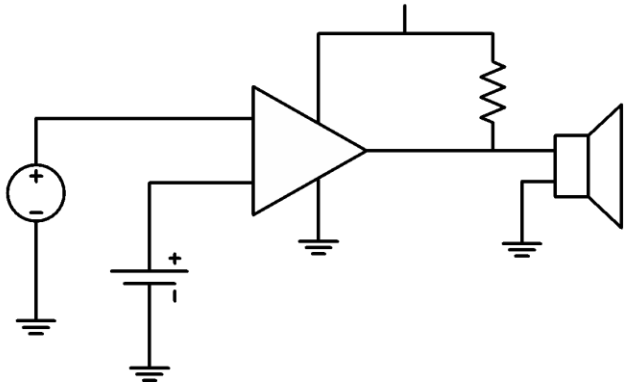


Figure 3.2.1.5F: Basic Comparator Circuit Schematic.

The Power distribution board will incorporate all of these circuits in a single Printed Circuit Board (PCB), which will be designed using Cadsoft Eagle PCB software. This board will be the centralized power manager for the different equipment that utilize power on the drone, and will add safety measures that are part of the FXUAV design requirements. Because of the fact that the final design of the PCB will be dependent of the final equipment used on the FXUAV, further design parameters for it will be discussed in **Section 4.**

3.2.1.6 Camera

A camera will act as the eyes of the FXUAV during both the system's detection and tracking mode. For this reason, camera choice is a vital aspect of the design and development process for the FXUAU. The camera will be mounted to the underbelly of the quadcopter frame and situated in such a way to direct its field of view straight towards the ground, ensuring no angular offset. The chosen camera must be able to communicate with an onboard microcontroller unit in order to transmit individual frames of data or images. The camera must also be able to provide adequate data, both in terms of rate and level of quality, so that a fire can successfully be distinguished from its surroundings. Furthermore, the quality of the cameras output is also important because it must be high enough to allow the object to be detected from a height of about 10ft. optionally, however preferred, the camera must incorporate a stabilization module to compensate for both climate

conditions such a wind, and vibrations stemming from the body of the quadcopter of which its attached to.

Thorough camera research has minimized the list of suitable cameras down to two possibilities. These modules are GoPro's HERO4 Black camera, the TAU Thermal Imaging camera from FLIR Systems, and the Logitech HD C270 webcam.

GoPro HERO4 Black

This camera was a key candidate for FXUAV detection and tracking system due to the many features it incorporates. The HERO4 Black provides top quality images and its built in processor is two times faster and more powerful than previous GoPro HERO modules. It has ultra-high resolution and two times the deliverable frame rate than its forerunner. Furthermore, the GoPro HERO4 incorporates both Wi-Fi and Bluetooth capabilities which is especially advantageous per communication requirements with the ground control unit. The GoPro HERO4 camera produces both color video feedback or color photo feedback. The decision to either use this camera or another camera will be largely based on the factors listed below. Table 3.2.1.6A and 3.2.1.6B show some of the video attributes of the GoPro HERO4 camera.

Advantages:

- 5PM photo resolution, 2560X1920 screen resolution
- Refer to Table 3.2.1.6A Video and Photo Capture, to review Video recording and time lapse photo capabilities and Table 3.2.1.6B Advanced Video Settings for superview settings
- Continuous photo capture, Night photo capture
- Lithium-ion rechargeable battery
- Wi-Fi communication capabilities
- Up to 64 GB storage capacity

Disadvantages:

- Medium Field Of View at 5MP
- Weighs 5.4oz (152g)
- Costs US$499.00

Video Resolution	Video Frames per Second	Video Field Of View
1440p	25, 24	Ultra Wide
1080p	30, 25, 24	Ultra-Wide, Medium, Narrow
720p	60, 50, 30, 25	Ultra-Wide, Medium, Narrow

Table 3.2.1.6A Video and Photo Capture

SuperView Mode	Video Resolution	Field Of View
4K SuperView	3840 x 2160	Wide Angle
2.7K SuperView	2704 x 1520	Wide Angle
1080p SuperView	1920 x 1080	Wide Angle
720p SuperView	1280 x 720	Wide Angle

Table 3.2.1.6B Advanced Video Settings

Logitech HD C270 Webcam

The Logitech C270 was a cheap alternative to the GoPro that offered impressive quality for a significant cost difference. It also featured a 1080x720 resolution, the same as the GoPro. However, it did not have any thermal imaging capabilities, which made it inferior in every aspect to the FLIR Tau 2. In addition, the Logitech camera is lighter than all other cameras available at 4.48oz. Another benefit is that the camera does not need to be connected serially. The Tau does, because it transmits analog data, which may be more accurate, but takes more effort to troubleshoot connection issues. The Logitech camera connects directly to the single-board computer with a USB connection, and is accessible through SSH or a VNC server.

FLIR Tau 2 Thermal Imaging Camera

The Tau 2 camera was specifically designed to incorporate features that make them well suited for autonomous aerial vehicles. It is loaded with features such as image processing modules, radiometry, analytics, and telemetry temperature readings, increased sensitivity, imagers and much, much more. This camera is a thermal imaging camera and it therefore also includes such features as isotherms colorizing and edge sharpening. The Tau 2 camera is a United States Department of Commerce export-controlled device. Below are a list of the top advantages and disadvantages that will be examined when decided whether to use the Tau 2 camera for the FXUAV System. Table 3.2.1.6C shows the types of infrared radiation used in the Tau2.

Advantages:

- Thermal Imaging Infrared Radiation (See Table 3.2.1.6C: Types of Infrared Radiation for a complete overview)
- No added cost—camera provided by FLIR Systems Inc.
- Uncooled VOx Microbolometer imager for low power consumption
- Multiple Lens Attachments (See Table 3.2.1.6D: Focal Length vs. FOV)
- Specifically Designed Image Processing Modes
- Adjustable isotherm thresholds
- Middle to High Resolution (see Table 3.2.1.6E: Tau Overview below)
- 40,000ft Operational Altitude

Disadvantages:

- Added weight from cooling module
- Potential ‘blindness’ from its own radiation—decreased sensitivity
- Resolution caped at 640X480 (Usable)

Division Name	Abbreviation	Wavelength
Near-infrared	NIR	0.75 − 1.4μm
Short-wavelength Infrared	SWIR	1.4 − 3μm
Mid-wavelength Infrared	MWIR	3 − 8μm
Long-wavelength Infrared	LWIF	8 − 15μm
Far Infrared	FIR	15 − 1000μm

Table 3.2.1.6C: Types of Infrared Radiation

Determining which camera, amongst infrared cameras, was best suitable option was first narrowed down by the type of infrared radiation the camera was sensitive to. This is particularly important when dealing with infrared because the type of infrared determines the temperature and illumination source requirements of the system's operation environment. That being said, Long wavelength infrared radiation is most desirable for the FXUAV detection module because this region allows infrared sensors to attain fully passive images. Furthermore, this infrared region does not require the sensors which are operating within it, to have an illumination source in comparison to electro-optical cameras which do require visible light illumination. As a result, it was decided that a long wavelength infrared (thermal imaging) camera would be the best type of infrared camera if in fact, an infrared camera is the ultimate decision for the detection unit on the FXUAV system. As a result, several versions of FLIR Systems Tau2 thermal imaging camera were compared, as seen in Table 3.2.1.6E: Tau Overview, below.

Module	Tau 640	Tau 336	Tau 324
Digital Video	640x512	336x256	324x256
Analog Video (NTSC)	640x480	640x480	640x480
Pixel Pitch	17μm	17μm	25μm
Frame Rate	30Hz	30/60Hz	30/60Hz
Time to Image	<5.0 sec	<4.0sec	<4.0sec
Power Consumption	~1.2W	<=1W	<=1W
Weight	72g	72g	72g

Table 3.2.1.6E: Tau Overview

Of the three infrared modules above, the Tau 640 is the most power hungry however the increase in power consumption between this module and the other two is very small and therefore does not have much effect on narrowing the decision between the three modules. Because price is not a factor if a Tau camera is chosen, the better of the three options would be the Tau 640 module due to its higher digital video resolution output and its support for larger UAVs rather than the Tau 336 or the Tau 324 which are better suited for smaller UAVs.

The final decision regarding the infrared module would be the size lens that is attached to the camera. Because the FXUAV System is designed in such a way

that the detection unit would be mounted in a fixed position with relatively no degrees of freedom, the field of view is a critical factor in the detection success rate. The field of view is dependent upon the focal length; a small focal length will harvest a large field of view and a large focal length with harvest a small field of view. Consequently, a small focal length will cause objects in the image to appear very small and a large focal length will cause objects in the image to appear quite large. That being said, there exists a tradeoff between the size of the field of view and the success rate of detection. Table 3.2.1.6D: Focal Length vs FOV below, depicts this tradeoff for the Tau2 640 infrared camera.

Lens	FOV-Horizontal (deg)	Pixel Size	Analog Display-Horz	Analog Display-Vert	Detector Size	Focal Length
WIDE FOV						
7.5mm	90	1.70E-05	640	480	0.01088	0.006926423
9mm	69	1.70E-05	640	480	0.01088	0.009034465
13mm	45	1.70E-05	640	480	0.01088	0.013852846
19mm	32	1.70E-05	640	480	0.01088	0.019480565
NARROW FOV						
25mm	25	0.000017	640	480	0.01088	0.024935123
35mm	28	1.70E-05	640	480	0.01088	0.022263503
50mm	12.4	1.70E-05	640	480	0.01088	0.050272426
100mm	6.2	1.70E-05	640	480	0.01088	0.100544852

Table 3.2.1.6D: Focal Length vs. FOV

Based on the above calculations, the acceptable focal length for the Tau2 640 is 9mm or 13mm. This was determined based on the conclusion that a field of view of at least 30 -35 degrees is desired in order to obtain the best possible object size within a fairly large field of view. Because the detection unknit will be mounted in a fixed position on the underside of the quadcopter frame, no pitch, row, or yaw will have to be account for and we can therefore use the 9mm lens to obtain a slightly large field of view given the fact that the FXUAV system will be approximately 10ft above the ground and each pixel is 17µm for the Tau2 640 camera. Please refer to **Focal Length and Perceived Object Size Calculations** for focal length and perceived object size calculations.

Post technical feature analysis of both the infrared modules and the color module proved that an infrared camera module would be the best choice, in effect, the Tau2 640 unit. Although both cameras, the Tau2 640 and the GoPro HERO4 Black, are suitable for the FXUAV, the Tau 2 camera has several desired advantages over the GoPro HERO4 Black camera such as video and image output format and the capability to colorize temperatures via distinct isotherm thresholds.

In addition to these slight technical advantages, the Tau 2 thermal imaging camera also holds a significant pricing and accessibility advantage given that FLIR Systems, the manufacturer of the Tau 2 thermal imaging camera is the sponsor of the FXUAV project and has stated that they will obtain a thermal imaging camera for the FXUAV system at no cost.

Unfortunately, due unforeseeable shipping issues in conjunction with time constraints, the Tau2 640 was not used on the FXUAV. Instead, the engineers choose the Logitech HD C270 Webcam as the alternative, based on the research above.

3.2.1.7 Image Processing

The Intel Open Computer Vision Library, OpenCV, offers an assortment of algorithms related to both computer vision and video capturing. The libraries offered, provide a means to streamline the process of extracting and reading images from a specified video camera. With respect to similar computer vision libraries, OpenCV is far superior in terms of speed and emphasis on real time computing. Moreover, despite its focus on real time application and algorithms, OpenCV maintains an extensive diversity of applicable computing. That being said, OpenCV is very suitable and desirable among many programmers for both generating and testing a wide range of computer vision algorithms. OpenCV is foreseeably as close to a real time repository as a global library can achieve. Its library is constantly under development to include the most up to date versions of each computer vision algorithm it holds.

OpenCV is compatible with a specific list of microcontroller units. The Beagle Bone is on the list of compatible microcontroller devices due to OpenCV's support of the Linux operating system. Once the compliant MCU has access to the specified data source, OpenCV can execute its algorithms on that data.

OpenCV's open source framework supports a select few programming language interfaces; Python, C, C++, and Java. In regards to image processing algorithms specifically, OpenCV offers two different methods for object detection; Latent SVM and Cascade Classifier. Latent SVM can only be implemented via C and C++ code. This algorithm approach is based very similarly on the Dalal-Triggs detector. It provides the programmer with convenient features such as storing images, and a framework for using filters on histogram of oriented gradients (HOG) descriptors. The second image processing algorithm provided by OpenCV, Cascade Classifier, is a two stage process. It involved the training stage and the detection stage. Amongst the two applications within OpenCV, which are used for classifier training, both HAAR-like features and LBP features are supported. Both negative and positive samples are needed when using OpenCV's Cascade Classifier Training method for object detection. Cascade Classifier Training can be implemented in all supported interfaces of OpenCV; that is, Python, Java, C, or C++.

That being said, the FXUAV image processing unit will implement OpenCV's Cascade Classifier Training algorithms to allow for more versatility when it comes to implementation methods. A list of classifiers will be trained to detect an object

specified by a specific set of training materials. The classifiers are then used to determine, when given an image, if the specified object occurs in any vicinity within that image. An object is said to be detected, if or when a vicinity, within the image of which was passed to the classifier(s), passes all classifiers successfully.

Two common errors that are seen when Cascade Classifier Training is implemented are, false positives and false negatives. A false positive error occurs when a vicinity with an image passes each classifier in indication that the object was detected when in fact, the object does not occur within that vicinity. Secondly, a false negative occurs when a vicinity within an image that does contain the specified object, does not pass all the classifiers, which again, indicates that the object does not exist in that vicinity.

In a perfect world, one-hundred percent of the true positives should be deemed as true by each of the classifiers. In retrospect however, it is a fact that not every true positive can be found and each classifier is therefore trained to catch as many true positives as possible. That being said, the main issue with this method of classifier training, lies principally in the fact that the number of false positives reported by a single classifier is, on average, about fifty percent. In attempt to surmount such a high error rate, it is of practice to cascade a number of classifiers amongst the input data. Previously as mentioned, a classifier in itself has roughly a 50% false positive error rate. Therefore, by cascading several classifiers, each with an approximate 50% error rate, the *overall* error rate will be less; depletion occurring as the number of cascaded classifiers increases. In effort to do so, each subsequent classifier, in the cascaded line, should be trained to anticipate false positive reports from the preceding classifier and mark it as such. The AdaBoost algorithm provides the foundation of which such classifier training is a derivative of. The technique implicates each classifier based on the number of image regions reported as positive, in descending order; more explicitly, each classifier is cascaded in a decreasing manner, with respect to the number of positive image vicinities it identifies. This systematic technique allows for the number of false positives to decrease as the data gets forwarded further down the line of cascaded classifiers in a liquidating fashion. With redistribution rights for the following figure still pending from Jeff Bier, the founder of Embedded Vision Alliance, Figure 3.2.1.7A illustrates the cascade of classifiers of which the systematic technique executes.

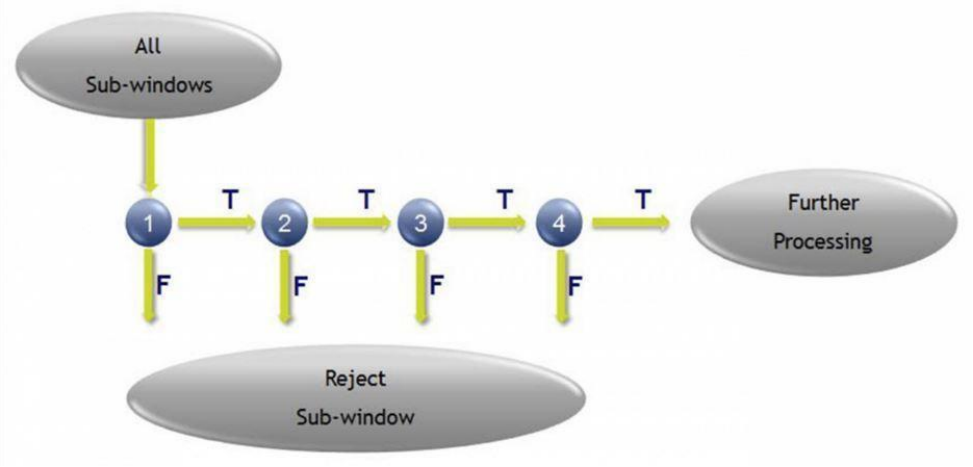


Figure 3.2.1.7A: The Cascade of Classifiers

Each classifier is said to take in a certain number and type of feature based on the algorithm used in implementation. The basic classifier input consists of edge features, line features, and center-surround features. This set of classifier inputs are known as Haar-like features and can be seen in Figure 3.2.1.7B below (redistribution rights granted from OpenCV User Guide documentation). The white areas within each feature signify positive pixel weights and similarly, the black areas signify negative pixel weights. Responses are calculated via a simple arithmetic difference equation. The weighted sum of pixels covering the black spaces of the feature is subtracted from the weighted sum of pixels covering the whole feature. The resulting integer is then used to classify the image as either a positive or a negative, as it pertains to containing the object or not containing the object, respectively. The threshold defining a positive versus a negative response is determined by the programmer in accordance with the preferred false positive rate as well as the desire detection rate.

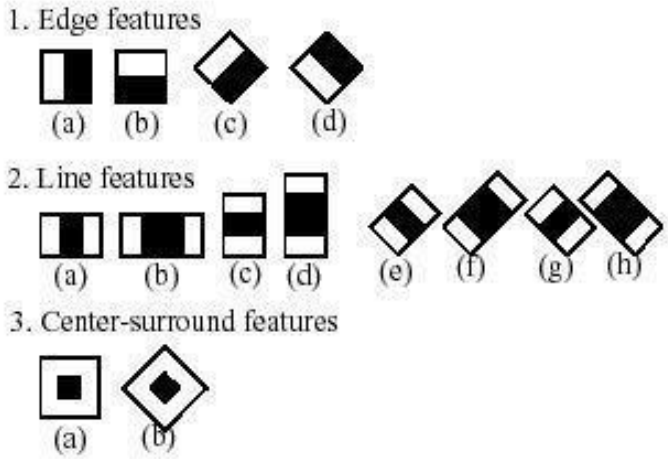


Figure 3.2.1.7B: Feature Prototypes of Haar-like Features

The pixel values that are used in the response classification process are calculated via integral images at a very rapid speed. The intrinsic functions within the OpenCV library that calculate the integral of an image, take in certain parameters from a particular set. The differing particulars are dependent upon the programming language interface being used to execute the integral function. This particular set includes and image parameter, a sum parameter, a sqsum parameter, a tilted parameter, and finally an sdepth parameter. The integral function is responsible for calculating three differing integral images for a given source image using a series of three equations (see Figure 3.2.1.7C: Equations for Integral Image Calculation—copyright permission granted from OpenCV User Guide documentation).

$$\mathtt{sum}(X,Y) = \sum_{x<X,y<Y} \mathtt{image}(x,y)$$

$$\mathtt{sqsum}(X,Y) = \sum_{x<X,y<Y} \mathtt{image}(x,y)^2$$

$$\mathtt{tilted}(X,Y) = \sum_{y<Y,\mathtt{abs}(x-X+1)\leq Y-y-1} \mathtt{image}(x,y)$$

Figure 3.2.1.7C: Equations for Integral Image Calculation

The equations above clarifies how each pixel value is calculated; it is equal to the sum of the value of the source image's corresponding pixel, the value of the pixels above the corresponding pixel, as well as the values of the pixels to the left of the corresponding pixel. The underlying effort of such rapid calculation of weighted pixel sums is in response to the integral image calculations above which make it feasible to compute any rectangular sum with merely four single array calls.

The appropriate set, or cascade, of classifiers that is optimal for a high detection rate and a low false positive rate is specific to the object being detected. During the classifier selection process, there are two different data sets that should be used during testing. The first should be a database comprised only of images that *do* contain the object being detected. The second should be a database comprised only of images that *do not* contain the object. The results of these two tests in conjunction with chosen threshold values for both detection and false positive rates will help determine which classifiers are optimal for the object detection task at hand. After the classifier training process has been completed, the object detection algorithm can be executed.

3.2.1.8 Communication Interface

Multirotor copters usually use anywhere from 1 to 3 communication interfaces with the user, the number of communication interfaces needed on a multirotor depends on what purpose the multirotor copter will be used for. The first and main communication interface, is for the multirotor is the RC transmitter and receiver.

For the most part, RC transmitters and receivers are used for the user to fully control the multirotor vehicle, but since the FXUAV will be fully autonomous, the RC transmitter and receiver will be heavily used to set-up and calibrate the multirotor copter's motor properly. Once the initial set-up and calibrations are completed the autopilot will communicate with the multirotor using the Telemetry communication interface.

The second and necessary communication interface for autonomous flying multirotor is the telemetry radio. This communication interface will allow the Ground Station read data from the Pixhawk inertial measuring units, and uses that data to control the multirotor flight. The Pixhawk come with four mounted to the MCU, the MPU6000 as main accelerometer and gyroscope, the ST Micro 16-bit gyroscope, the ST Micro 14-bit accelerometer/compass (magnetometer), and the MEAS barometer. These sensors combined with a GPS unit and a compass, is what the Ground Station relies on to make sure the multirotor copter is navigating and functioning correct.

The third and final communication interface is the video receiver and transmitter. This communication interface is used to provide a live video link to a camera attached to the multirotor copter. It is very common to not see this third transmitter on a multirotor because there are many cameras with built in Wi-Fi available in the market, these cameras can help save on overall cost and total power usage. Since the FXUAV is required per specifications to send live infrared image to the ground station, there will be a need for the third communication interface for video processing.

The three most popular RC communication interfaces today are Bluetooth, Wi-Fi, and radio frequency transmitters. Since the radio frequency transmitter and receiver can provide the most reliability and the longest range, the FXUAV will be using the Radio Frequency transmitter and receiver. Most of the RC transmitters and receivers operates in the 2.4GHz bandwidth due to FCC regulations, this bandwidth provides a range in average of up to 1Km in free space. The minimum of 4 channels is required in the transmitter to control the different functions of the multirotor copter, each channel allows one individual function to be controlled on the aircraft. The four basic functions to fly a multirotor copter are the pitch, yaw, roll and throttle. Since the FXUAV will need to support other functions or different flight mode, it is ideally to have at least 8 channels to use for different functions.

In the receiver side, it is important to know what the compatible connections in the flight controller are to be able to receive the signal. Since the FXUAV will be using the Pixhawk flight controller, the interfaces available for the receiver are the Spektrum DSM/DSM2/DSM-X Satellite input, Futaba S.BUS input and the PPM sum signal. It is also important to verify the compatibility of the receiver and the transmitter because even though they operate in the same bandwidth they have different modulation, frequency hopping and spread spectrum technology. There are also different models of receivers that allows user to increase their antenna gain with a gain amplifier. Since the FCC regulates the total transmission power, it

is important to not use a gain that can risk any violations. In Table 3.2.1.8A below, we compared different transmitter and receiver combinations that would satisfy the FXUAV communication requirements.

RC transmitter and receiver Comparison							
Transmitter	Frequency	Modulation	S. Spectrum	Channels	CH. Bw	Receiver	Interface
ARRS AT9	2.4GHz	QPSK	DSSS	9	5MHz	ARRIS R9D	S.BUS
Futaba 8JH	2.4GHz	PWM	FHSS	8	5MHz	R2008SB	S.BUS
Spektrum DX8G2	2.4GHz	DSMX	DSSS	8	5MHz	AR8000	S.BUS
TARANIS X9D	2.4GHz	N/A	ACCST	16	5MHz	X8R	S.BUS

Table 3.2.1.8A: RC transmitter and receiver Comparison

Telemetry systems are used by the Flight controller computer board to send flight data to the ground station, then the ground station uses the data to fly the multirotor within the required mission specification. There are currently two popular ways to transmit this data, through a Bluetooth connection, or through radio frequency. Bluetooth is a short range system, so the best option available is to transmit the telemetry data through radio frequency. Since the FXUAV is required to be fully autonomous, the telemetry data is crucial for a successful design. With this in mind, it is best to use different bandwidth to transport the signal from the multirotor to the ground station. Most of the telemetry radios in the market operate at the allowed frequency of 900 MHz, which would be more than sufficient to meet the communication requirements of the FXUAV. Because we are using the Pixhawk as a flight controller, the main requirement for the telemetry radio is that the Ground Station firmware can be loaded onto it. Fortunately, there are many different options in the market for an open-software radio system, and since all of the 915MHz radios will meet the FXUAV requirements, we chose a telemetry radio system that best fit the project budget. After much research and comparing different options, the most viable option for this project is the HKPilot telemetry radio sets. The full features and specifications for the HKPilot radio are as following:

Features:

- Interchangeable air and ground modules
- Micro-USB port
- 6-position DF13 connector
- 100mW maximum output power (adjustable) -117dBm receive sensitivity
- Based on HopeRF's HM-TRP module
- SMA connector
- 2-way full-duplex communication through adaptive TDM
- UART interface
- Transparent serial link
- MAVLink protocol framing
- Frequency Hopping Spread Spectrum (FHSS)

- Configurable duty cycle
- Error correction corrects up to 25% of bit errors
- Open-source SIK firmware
- Configurable through *Mission Planner and *APM Planner

Specs:

- Supply voltage: 3.7-6 VDC (from USB or DF13 connector)
- Transmit current: 100 mA at 20 dBm
- Receive current: 25 mA
- Serial interface: 3.3 V UART
- Size: 25.5x 53x11 mm (without antenna)
- Weight: 11.5g (without antenna)

The third and final communication interface used by the FXUAV will be the video transmitter, this transmitter is required to be able to transmit live video from the FXUAV to the ground station as per requirements. The two most popular methods used in multirotor copters are Wi-Fi and 5.8GHz radio frequency. There are numerous Wi-Fi capable video cameras available in the market which is a good option for short range video feed. Since the requirement for this project is to use an infrared camera for fire detection, the Wi-Fi capable camera will not be an option. Because of this constraint, the only other option available for the FXUAV to meet the necessary requirements is a 5.8GHz video transmitter and receiver. A 5.8 GHz transmitter using 200mW of power to transmit can reach a total transmission distance of 300 meters with no interference when transmitting in free space. With a wide range of video transmitter and receiver available to meet the necessary requirements for the FXUAV, the best option found within our budget constraints was the Boscam 5.8GHz TS351 TX and RC305 RX.

RC305 5.8 GHz Receiver Specifications:

- Channels: 8
- Receiver operating voltage: 5VDC
- Receiving sensitivity:-90dBm
- Operating temp: -10~85C
- Video bandwidth: 0~8.0MHz
- Audio frequency: 6.5MHz
- Video input level: 0.8~1.2Vp-p
- Video input impedance: 75Ohm
- Audio input level: 0.5~2.0Vp-p
- Audio input impedance: 10K/Ohm
- Power supply voltage: 5-12V
- Supply current: 150mA
- Double lines AV output: analog AV signal output
- Frequency control: built-in frequency and phase lock loop
- Gross weight:120g

- Antenna connector: RP-SMA (Inside the needle)
- Dimensions: 61x52x13mm(aluminum alloy extrusion profiles)

Channels

5.8 GHz: 5705, 5865, 5665, 5645, 5885, 5905, 5925, 5945MHZ

TS351 5.8 GHz Transmitter Specifications:

- Video Format: Pal/NTSC
- Output impedance: 50Ohm
- Output power: 21~23dBm
- Channels: 8
- Transmitter operating voltage: 7~15VDC / 150mA
- Video bandwidth: 0~8.0MHz
- Audio frequency: 6.5MHz
- Video input level: 0.8~1.2Vp-p
- Video input impedance: 75Ohm
- Audio input level: 0.5~2.0Vp-p
- Audio input impedance: 10K/Ohm
- Weight: 25g
- Antenna connector: RP-SMA
- Dimensions: 55x26x17mm

Channels

5.8GHz: 5705, 5865, 5665, 5645, 5885, 5905, 5925, 5945 MHz

The Global positioning system is the last important module required to fly the FXUAV autonomously, this is the only way that the ground station can know the geo-location of the multirotor system at any given time. GPS has become an important part in autonomous navigational systems, and without it there's no way for the FXUAV to fly using an autopilot software. There are hundreds of different types of GPS modules commercially available in the market, some are made for general purposes, while other are made for a specific application. For multirotor copters, GPS module must be light and use as little power as possible without affecting its performance. The main features analyzed for a GPS module for this project were the refresh rate, accuracy, power consumption, and interface. There are also GPS modules for multirotor that includes other internal measurement units, but since the Pixhawk flight controller board already contains all necessary IMUs except a compass, a GPS and compass combination module is the best option to help keep the cost within our project budget. Table3.2.1.8B: shows a comparison between the top 5 models available that fit the FXUAV requirements.

GPS Module Comparison Table					
Model	Refresh rate	Weight	Accuracy	Power Comsumption	Interface Compatible
3DR u-blox	1-10Hz	16.8g	+/- 2.5m	51mW	yes
NAZA-M V2	1-10Hz	21.6g	+/- 2.5m	N/A	yes
Adafruit	1-10Hz	8.5g	<3m	100mW	yes
Copernicus II	1Hz	1.7g	<3m	93.9mW	yes

Table 3.2.1.8B: GPS Module Comparison

After analyzing the options available in the market for GPS modules that ware compatible with the Pixhawk flight controller board, the 3DR U-Blox GPS is the most viable option for this project. Since it is made by the same manufacture of the flight controller and it is fairly similar to the other models, it is the best option available that fits the FXUAV budget constraints while still meeting the necessary requirements.

3.2.1.9 Microcontroller Unit

The initial FXUAV design was to incorporate several microcontroller units in the system design. The flight controller and wireless communication would run through one microcontroller unit while the image processing would run through a second microcontroller unit. However, after extensive research, cost analysis, and risk assessment, it was determined that a single microcontroller unit would be best to run all three components; the flight controller, wireless communication, and image processing.

After much research of several microcontroller units, the candidates were narrowed down to four big contenders. These contenders, being the ATMega2560, the BeagleBone Black, the Odroid XU4, and the Raspberry Pi, each provided several distinct capabilities that proved helpful to the vital operational needs of the FXUAV. That being said, a detailed comparison between the two microcontroller units was carefully mapped out to include the key characteristics of these two units in terms of what aspects were advantageous to the FXUAV design. Figure 3.2.1.9A Assessment and Evaluation of Microcontroller Units, effectively illustrates the similarities and differences of the two microprocessor contenders.

	Raspberry Pi 2 B	BeagleBone Black	Odroid XU4	Arduino ATMega2560
Cost	$45.00	$55.00	$75	$30.00
Universal Serial Bus	4x2.0	1x2.0	2x3.0	N/A
RAM	1 GB	512 MB	2 GB	8 KB
Speed	900 MHz	1 GHz	2 GHz	16 MHz
Processor	ARM Cortex -A7	TI Sitara AM3359 ARM Cortex A8	Samsung Exynos5422 Cortex™-A15 2Ghz and Cortex™-A7 Octa core	ATMega2560
GPIO (pwm)	Yes	Yes	No	Yes
Serial Comm. To Flight Controller	Yes	No	Yes	Yes
Operating System	Android, Linux	Android, Linux, Windows CE, RISC OS	Android, Linux	N/A

Figure 3.2.1.9A Assessment and Evaluation of Microcontroller Units

Because the flight controller and the wireless communication tasks of the FXUAV are of more versatility in terms of their communication capabilities, the image processing task held most priority when determining which microcontroller unit would be best. Our image processing research yielded several open-source code modules that are written in both C and Python programming languages which, in turn, placed the Paspberry Pi in better standing than the Arduino ATMega2560 right off the bat. Furthermore, we were able to receive the Raspberry Pi at no charge from FLIR, the sponsors of the FXUAV system. However, in addition to these advantages, on the technical side of things, the Raspberry Pi, as seen from the above evaluation, is operating at a much faster speed than the Arduino ATMega2560 and provided both serial communication with the flight controller and pwm-capable pin outputs. Furthermore, microprocessor speed holds a very high priority in the MCU decision matrix, observed below in Figure 3.2.1.9B MCU Decision Matrix.

Weighted Priority	1	4	3	2	2	12	
	8%	33%	25%	20%	17%	100%	
Option	Cost	Speed	Programming Language	IDE	RAM	Score	
Raspberry	100	80	90	70	50	80	
Arduino ATMega2560	50	50	70	100	80	74	

Figure 3.2.1.9B MCU Decision Matrix

Aforementioned, the image processing task is most dependent on the MCU decision, in terms of how easy/hard it will be to develop the necessary image processing detection code and not only how fast the image processing system can detect the desired target, but also how fast these detections can be communicated to the flight controller for corrected navigation. Therefore, after careful consideration of both the dependencies of the image processing subsystem and the contending microcontroller units, it was determined that the Raspberry Pi proved to be the best choice for the FXUAV.

3.2.1.10 Ground Station

The ground station is the software that controls and is able to autopilot the multirotor after the mission settings and waypoints have been planned by the user. It communicates with the multirotor wirelessly from a computer on the ground, and it uses the telemetry data from the multirotor to command the unit properly. For this design, the FXUAV needs a ground station that is capable of fully controlling the multirotor autonomously. Table 3.2.1.10A below shows three of the different ground station options that were considered for the FXUAV.

FXUAV Ground Station Comparison		
Name	Platform	Main Features
Tower	Android	Mission Planning, Follow Me Mode, field Surveing, and 3D mapping
Mission Planner	Windows	Point and click configuration, Acro, Stabilize, Loiter, Return to launchpoint, Follow me, GeoFence, Failsafe programming, and Auto(runs phyton scripts during mission.)
APM Planner 2	Windows, Mac OS X, Linux	Same as Mission planner.

Table 3.2.1.10A FXUAV Ground Station Comparison

After comparing the different Ground Station options, the Mission Planner software was the best choice for the FXUAV. This is an open-source software that gives the user ability to run python scripts within the mission which should facilitate the objective of finding a fire on the ground with the script created by Jamie Peck.

3.2.2 Fire Extinguisher

3.2.2.1 Concept of Design

The design of the fire extinguishing system will need to meet all specifications defined in section **2.2.4 Fire Extinguishing Specifications**. This is in order to ensure all requirements are met and that the fire extinguisher is effective at suppressing a fire. The design concept will use a servo that is electronically controlled. Once the servo receives a voltage signal, it will retract, and pull open one side of the container. Once the container is open, gravity will pull its contents down towards the ground. How do we verify we are hovering over the flame? By using the FLIR camera to detect a flame, we will process each image until the flame that is detected is in the center of the frame. At this point the onboard controller will receive a message from the ground unit that is doing the entire image processing, to send a signal to the servo on the fire extinguisher. The onboard controller sends the signal to a custom PCB and the servo retracts, and everything that was mentioned previously, takes place. Fig 3.2.2.1A and 3.2.2.1B show some sample designs.

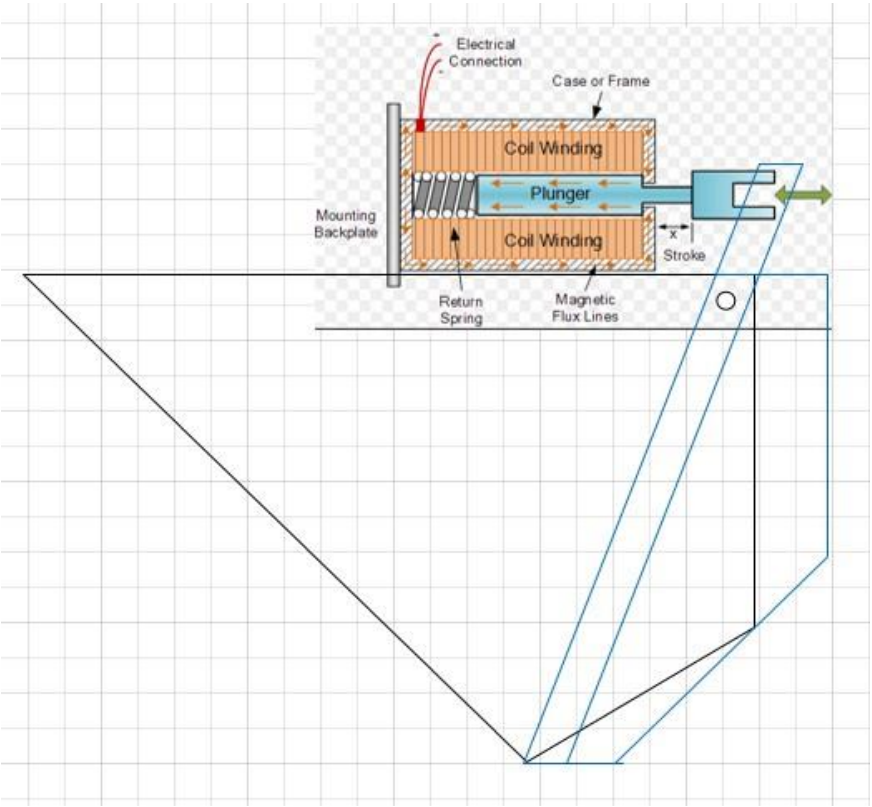


Fig 3.2.2.1A: Level Release Illustration

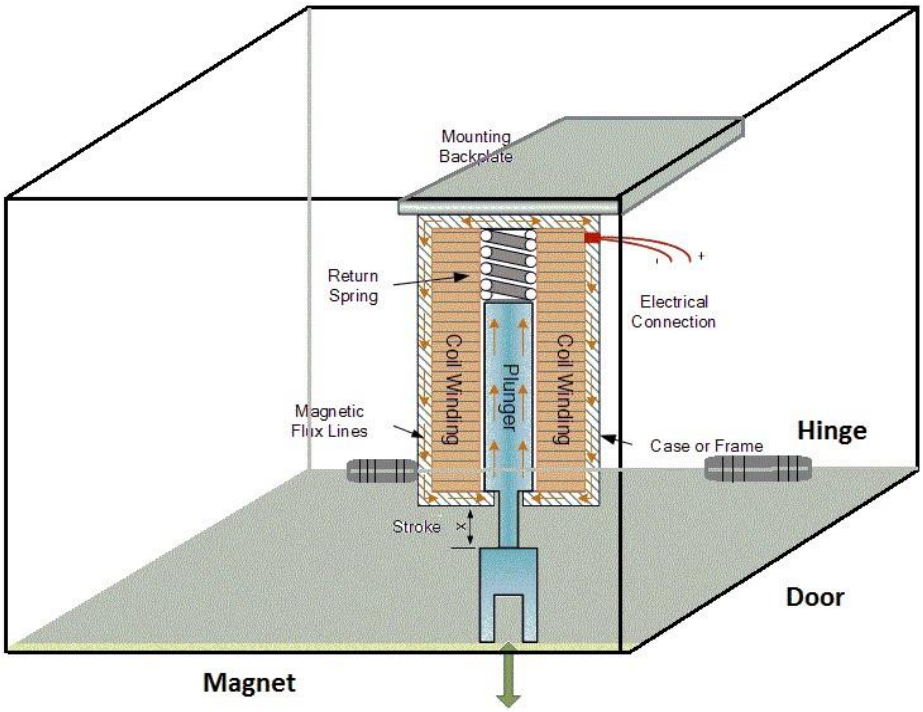


Fig 3.2.2.1B: Trap Door Illustration

Figure 3.2.2.1A describes a system wherein the servo is used as a latch and is retracted to pull the latch and thus, pull the wall of the container, allowing the contents to be released.

Figure 3.2.2.1B describes a system wherein there is a trap door that is held together using a magnet. The servo strength will be used to overcome the strength of the magnetic bonding and create a trap door type of release.

Also required, in order to meet our requirements, is we must create a custom printed circuit board in order for the servo to be controlled. Below in figure 3.2.2.1C a circuit schematic is shown for a possible solution for a minimal circuit board using an ATMEGA 328 microcontroller, also found in the Arduino UNO

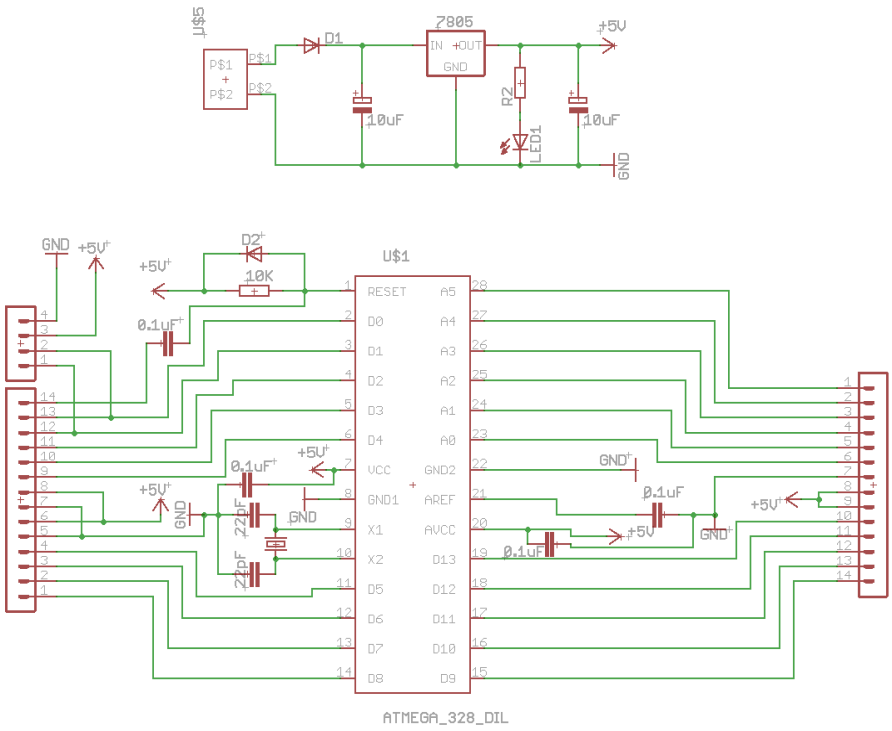


Figure 3.2.2.1C: Sample PCB Design Schematic

Once this schematic has been drawn it can then be laid out onto the board for the board design process. Using EAGLE this board layout is easy to develop once the schematic is made. All that is needed for the transition is to follow the schematic onto the board file. Figure 3.2.2.1D shows an example of the board layout for the schematic shown above.

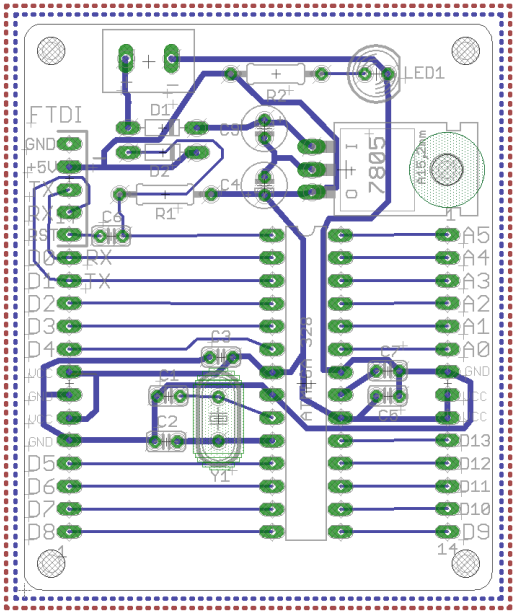


Figure 3.2.2.1D Sample PCB Design Layout

3.2.2.2 Contents of Extinguisher

There are multiple methods to extinguishing fires. Some fires require certain materials in order to effectively put them out. For example, grease fires are normal not fazed by water. This is because water is denser than grease. Therefore the grease rests on top of the water and the fire continues to burn. One thing that all fire suppressants have in common is the approach to extinguishing the fire; that is, suffocating, or oxygen deprivation. As long as the grease sits on top of the water, the water is not suffocating the fire. Most commercial fire extinguishers use powder, or chemical suppressants.

Types of fire suppressants

- Water - Water is an obvious choice for battling fires, and is often the first choice for most individuals. However, water isn't suitable in all cases. As discussed above, water is dense, and will therefore be ineffective when it comes to grease fires. A better choice for such a situation is a dry chemical. Although easily available and probably the most low cost out of the bunch, water is also heavy to transport. For a quadcopter, weight is a major concern. It is unlikely that a battery powered quadcopter will be able to carry a necessary amount of water to suppress anything other than a candle flame. For these, reasons we've decided that water may not be our best option, and more research was required.
- Sodium bicarbonate - Sodium Bicarbonate has many other names. For instance, baking soda. Sodium Bicarbonate has a tendency to create a cloud of carbon dioxide in the heat of fire. The carbon dioxide smothers the fire by driving away the oxygen. Baking soda is also very accessible and inexpensive. Additionally, baking soda is much lighter than water and can therefore be transported more efficiently, with more baking soda being transported per trip.
- Potassium bicarbonate - Potassium bicarbonate like sodium bicarbonate, is effective as putting out grease and oil fires. In fact, it's twice as effective as sodium bicarbonate. However, it is much more expensive and slightly heavier than sodium bicarbonate. For the purpose of this project, and due to budget constraints, this may not be as good of a choice as sodium bicarbonate.
- Foam - Foam is just as effective, if not more effective at suppressing fires as dry chemicals. This is because the foam blankets the fire and seals over it, refusing to let in oxygen. Foam is very lightweight and can be transported in large quantities, with much ease. Also, its weight class serves as an advantage when extinguishing grease or oil fires, for the same reason discussed about water vs dry chemicals. However, what makes foam special is that since it isn't dry, the foam doesn't suffer from flashbacks; that is, since gas created by cry chemicals disperses quickly, the fire can sometimes start up again. One drawback is the delivery of foam. The only way to effectively disperse foam over a fire is by means of compressed air. Without it, the foam is likely to fall slowly and inaccurately. Foam fire extinguishers only come in compressed containers.

3.2.2.3 Release System

Not only are the chemical and physical properties of the suppressant itself important, but so is the delivery of said contents. Imagine if there were a fire growing rapidly, and the only way to suppress it was to dump a chemical powder on it. It would be advantageous to the firefighter if there were some way to quickly release the powder onto the fire versus dumping it manually. This way, the firefighter can extinguish the fire faster than it can spread and will eventually lead to its demise. Discussed below are several options and combinations of how to each property suits this project, and why some are more advantageous than others. Also, discussed will be the drawbacks and concerns for each material.

Types of Delivery

- Compressed air - Compressed air containers are generally more effective because it allows the user to disperse the contents faster, and with more efficiency and accuracy. However, they typically require two hands for operation. One hand to aim, and the other to activate the release valve. Since our quadcopter will be airborne when it releases its contents, this may serve as a bit of a problem. Not to mention, the kickback from the pressure once the release valve is pulled. This can cause the quadcopter to shift unpredictably.
- Gravity - Letting gravity do the work isn't as effective as using compressed air. The pressure of gravity vs the pressure of a compressed fire extinguisher are worlds apart, and thus letting gravity do the work is much slower. However, by choosing to drop the fire suppression versus "shoot" it, we are able to control the flight of the quadcopter predictably. Also, budget constraints favor gravity, since it is a free resource. Although, dropping the contents will also require a material that is not easily dispersed. Otherwise, the contents will be ineffective by the time it reaches the flame.

4 Design

4.1 Software Design

4.1.1 Quadcopter

4.1.1.1 Mission Planner

One of the requirements of the drone is autonomous flight. This requires software that will track the drone system's location, read sensor input, and use those inputs to control the motors in a way that allows the drone to fly successfully. By using the Pixhawk flight controller, we will have access to the ArduPilot Mission Planner software. This software runs on a ground station computer that communicates with the drone system, features an easy to use interface, and controls the drone autonomously.

The Mission Planner software is a Windows-based application that features Google Maps-integrated waypoint navigation, with the ability to save and reload missions onto the Pixhawk. The Mission Planner software not only provides real-time data on the drone system's status while flying, but it also stores telemetry logs which will provide useful data when testing the system. The Mission Planner software was developed by Michael Osbourne for the open-source APM autopilot project.

Fig. 4.1.1.1A Mission Planner Interface

By using the Mission Planner software, we will be able to send data to the BeagleBone Black, which will fully integrate the flight controller within the system. The Mission Planner application also provides firmware that can be installed on the Pixhawk flight controller. The biggest benefit of the Mission Planner software is its extensive API and its ability to run Python scripts through MAVLink packets, which significantly extends the customization and capability of the software, allowing it to meet our system's requirements.

4.1.1.2 Image Processing

The FXUAV contains an image processing module which serves the purpose of analyzing data to look for a specified object. The object, in the case of the FXUAV, is a grease fire. The image processing module will receive data, or frames of images, from the thermal imaging camera via the Raspberry Pi micro processing unit; the Raspberry Pi controller will send these frames to the ground control unit via a wireless communications transmitter. After processing these images, the ground station will report back to the Raspberry Pi with a certain command, of

which is dependent upon the results of the previous image processing job. Figure 4.1.1.2A Image Processing Module below, illustrates this progression more in detail.

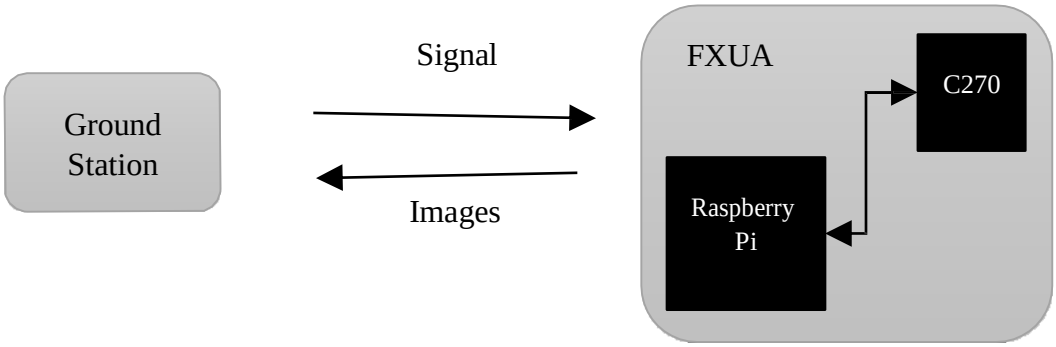


Figure 4.1.1.2A Image Processing Module

OpenCV is required for the image processing capabilities of the FXUAV. Once the C270 sends the images to the Raspberry Pi and the Raspberry Pi then wirelessly transmits this data to the ground station, OpenCV is used to analyze each image and detect if the fire is present in the field of view or not. If the fire is not detected within the current image, it continues extracting frames from the video stream, otherwise, if the object was detected it sends a course over ground back to the Raspberry Pi which would then send this to the flight controller. The image processing system then continues extracting frames from the stream and as long as the fire is detected in the images, then it continues to report a heading back to the Raspberry Pi.

The analysis of the images extracted from the video feed is done by way of defined classes and functions within the OpenCV library. OpenCV has a specific Python application designed precisely for object detection modules. The python application borrows functions and classes from the cv2 library as well as from the numpy library and although the application's algorithms are designed specifically for face detection functionality, the algorithms can easily be altered in accordance to the FXUAV fire detection requirement. The OpenCV algorithm has a method attributed to cascade classifying (refer to section 3.2.1.7 Image Processing, for a detailed understanding of cascade classifier training) of which imports a specific xml file responsible for delineating what constitutes as a face and what does not. Therefore, modifying the OpenCV face detection algorithm is as easy as importing your own xml file which contains the necessary classifiers for the specified object's detection, in this case, an xml file will be uploaded of which contains classifiers that will dictate what constitutes a fire and what does not.

There are two different methods that OpenCV uses in order to train classifiers. They are opencv_haartraining and opencv_traincascade. Opencv_haartraining is

no longer widely used in the realm of object detection because opencv_traincascade is an updated and faster version which supports the use of two types of features; haar features and local binary pattern features. Opencv_traincascade will therefore be the version that is used for the FXUAV object detection training.

The way in which the opencv_traincascade application works, first requires data to be prepped or created for use. Both positive and negative image samples are needed, in which the negative are manually created images that do not contain instances of fire and the positive are algorithmically created samples that do contain fire objects. The negative samples will be chosen from sets of subjective imagery and will be numbered uniformly and stored in a specific text file. OpenCV has a special utility called opencv_createsamples and this is the tool that will be used to help generate a set of positive samples for the FXUAV fire detection from a series of distinct images each containing differing view of what a fire may/could look like. The OpenCV utility will used a combination of approaches such as random rotating, intensity, background placement, etc. in order to generate a varied and large set of positive samples. Once the application finishes running, it outputs a file containing a set number (specified by the user) of random positive samples. The larger the pool of positive samples, the higher the accuracy of the detection module because the classifiers will, in turn, be of higher performance.

Once both positive and negative image samples have been created and stored in the proper manner, opencv_traincascasde can be run to generate the classifiers and store those in an xml file.

The process of creating sample image sets and training the classifiers is a task that can be done from any computer and at any time prior to the inauguration of the FXUAV system. The xml will then need to be transferred to whatever machine, or processor that is responsible for running the detection and tracking algorithms. For the FXUAV system, the xml file will need to be transferred to the ground communication system where the detection algorithm can be run as many times as need be until a fire is detected.

Once the ground station has done some processing and detected a fire, it will send a wireless communications signal to the Raspberry Pi processor to signify the object has been detected. The Raspberry Pi will then send a flag to the flight controller via a serial data bus connection, in order to interrupt the quadcopters current directional course with the new calculated heading. This heading is calculated by the image processing unit by calculating the slope and distance between the center pixel of the image and the center pixel of the square region containing the fire. The Raspberry Pi receives this new heading from the ground station and sends it out to the flight controller. Once the quadcopter has centered the fire in its field of vision it will then release the payload and extinguish the fire. This continuous cycle can be seen below in Figure 4.1.1.2B: Image Processing and Object Detection.

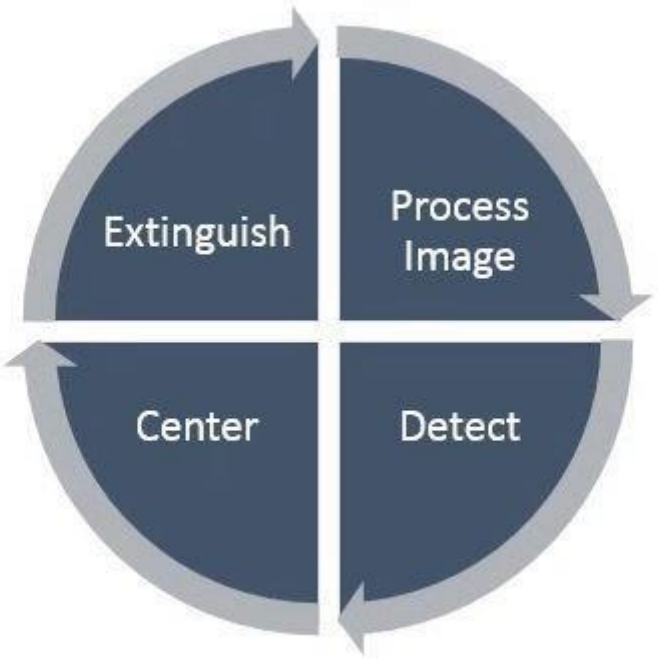


Figure 4.1.1.2B: Image Processing and Object Detection

4.1.1.3 Waypoint Navigation

Waypoint navigation will be handled through Python scripts, in coordination with the Mission Planner software explained in section 4.1.1.1. The Mission Planner application features Google Maps integration, and allows the user to set waypoints at any GPS coordinate on the map. This allows the team to create custom waypoint maps for testing, with the ability to save and reuse the maps. In addition to this, the Mission Planner application also has a built-in Auto Grid function, which has the drone system fly over a certain area in a grid pattern, which will make finding the fire extremely simple.

Fig. 4.1.1.3A Mission Planner Waypoints

The process of searching for a fire, identifying a fire, moving towards the fire, and hovering over the fire while the fire suppression is released, requires a well-defined state machine so that the drone system is able to complete the missions regardless of different conditions. A diagram of the state machine and descriptions of each state are as follows:

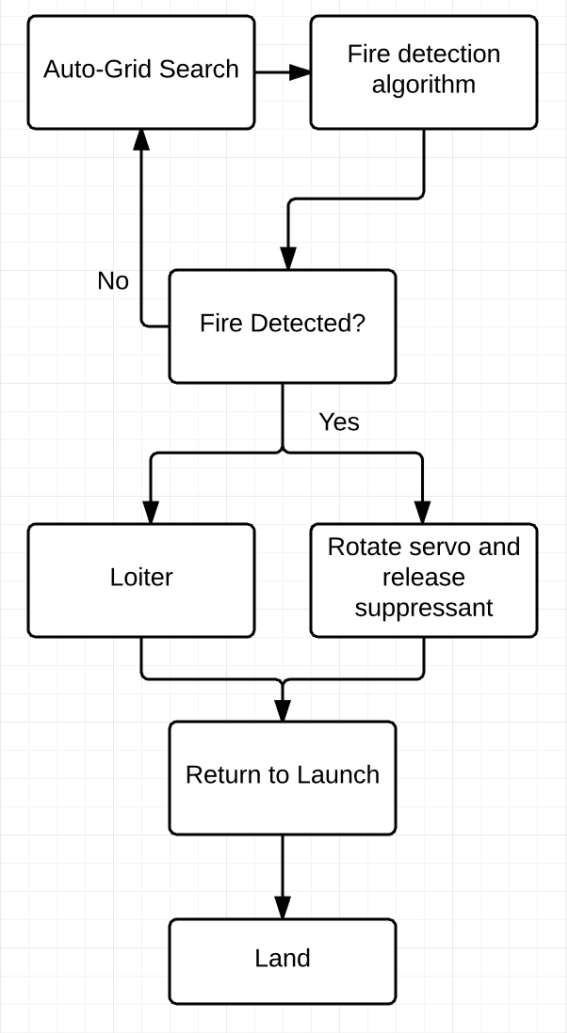


Fig. 4.1.1.3B Waypoint Navigation State Machine

The Auto-Grid Search is a function available within the Mission Planner application that will allow the drone system to search around a specified GPS coordinate in a grid-like pattern. This state will exit once the Logitech HD C270 camera identifies a fire, which will be communicated to the Pixhawk through the Raspberry Pi. Otherwise, the drone system will recursively search around the GPS coordinate until a fire is found. Further testing revealed that addition scripting was necessary to get this to work as intended, which will be explained in Section

Moving towards the center of the fire will occur once the fire is within the camera's vision. This first change that occurs is changing the state of the drone system from Auto mode to Guided mode. In this mode, the drone system is given a GPS coordinate that it moves toward, and once at the coordinate, hovers indefinitely. The Raspberry Pi will determine when the fire is within the camera's field of view. Then it will send data to the Mission Planner, which will change the state for the drone system to LOITER mode, while simultaneously running the script for the suppressant system. The waypoints require the data sent as latitude and longitude,

and also takes in the length of time to spent hovering at the current waypoint before moving to the next state.

LOITER mode requires an input deciding the length of time it will spend hovering at the height it enters Loiter mode at. The Raspberry Pi will send the current location to the Mission Planner, and will then run a script that releases the fire suppressant. Once the suppressant has been released, the drone exits the LOITER mode, and switches to the Return to Launch mode. During Return to Launch, the original GPS location of the drone when it is initialized and armed is sent as a new waypoint, which the flight controller then moves the drone towards. Conveniently, the altitude for the location is initially set to be the same altitude of the drone that is called when Return to Launch is called. This way, the drone only moves to the original location in terms of (x, y) coordinates, and then awaits further instructions. At this point, the mode is switched from Return to Launch to LAND, where the drone lowers itself until it is on the ground, then disarms the motors and powers down.

This will complete the mission and the test or demonstration will be complete.

4.1.2 Fire Extinguisher

4.1.2.1 Fire Suppression Release Function

The software design of the fire suppression release was implemented by using pulse width modulation, or PWM. PWM commands are made up of two main components, the duty cycle and the time. The duty cycle refers to the amount of time the output signal is high per cycle. See figure 4.1.2.1A that describes how duty cycles work.

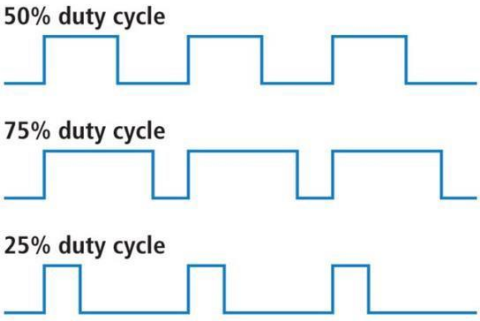


Figure 4.1.2.1A: Duty Cycle

The second component is the time. The time component dictates the angle of rotation. From 1 to 2 ms is the time range from 0 to 180 degrees. For example, a time of 1.5 ms will rotate 90 degrees. Once the first is the detected, the software will send a PWM command to rotate 180 degrees, sleep, and then rotate back to close the container.

4.1.3 Ground Control Station Development

The Ground Control Station will serve as the main hub with which all components of the FXUAV system communicate. The main components of the Ground Control Station include Mission Planner and communication with the Pixhawk on the drone, and the virtual network computing for communication with the Raspberry Pi. The communication between the Pixhawk and the Raspberry Pi, in addition to communication between the Logitech HD C270 and the Raspberry Pi, are done through wires on the drone itself. The Pixhawk and Raspberry Pi communicate through serial communications, and the Logitech HD C270 is plugged into a USB port on the Raspberry Pi. At the same time, there will also be one line of communication done via Wi-Fi between the Ground Control Station and the Pixhawk, for running Mission Planner, in order to use a remote control to override the Raspberry Pi's MAVLink scripts.

4.1.3.1 Implementation

The Ground Control Station is able to be run on any computer, ideally a Laptop, that can run APM/Mission Planner software, and a way to connect to the Raspberry Pi via secure shell (SSH) and virtual network computing (VNC). With Group 7's implementation of the FXUAV, a laptop running Windows 10 with an Ubuntu Virtual Machine was chosen as the Ground Control Station. Mission Planner is first run in order to get connected with the drone, and to load preset missions into the remote control, such as LAND. These missions are to be used in case of emergency, where the code does not work and the drone needs to safely land immediately. To communicate with the Raspberry Pi, a virtual machine for an Ubuntu Operating System was setup with Virtual box. A WAN would be setup and distributed from the Ground Control Station with a Pineapple Nano, and a miniature USB Wi-Fi module plugged into the Raspberry Pi would allow it to pick up the WAN. Within Ubuntu, the Ground Control Station would then SSH in, and setup a VNC server from the Raspberry Pi with the TightVNC software package. After starting the server, the SSH session would be disconnected, the virtual machine powered down, and a VNC viewer would be started on the Ground Control Station. After connecting to the Raspberry Pi's VNC server on the correct IP address and port number, the Ground Control Station would have full access to the Raspberry Pi's interface and operating system, from which all programming scripts would be run.

4.1.3.2 High-Level System Design

The overall software architecture of the ground station can vary greatly depending not only on the type of implementation of the application with the overall system, but also with the intricacies of how the data will be handled and managed by the application. Designing the software requires thought being put into the application at all levels, from high-level decisions such as a software architecture design pattern, to a low-level decision, such as creating a UML class diagram. Before beginning to work on an application, it is essential for a good software engineer to create a high-level design, and then compare it with other design patterns to identify the best practice for the project. There are several options for a high-level

design pattern to be used; these can include Model-View-Controller, Model-View-Viewmodel, Model-View-Presenter, and Layered Abstraction.

The Model-View-Controller (MVC) pattern has data models that store and organize data, controllers that manipulate, manage, and transfer that data, and views that describe the presentation of the user interface. The Model-View-Viewmodel (MVVM) design pattern is a derivation of the MVC design, and abstracts the public properties of the Model, in the View model stage, to prepare for presentation in the View stage. We do not need any abstraction, so we will not use the MVVM design. The Model-View-Presenter (MVP) design is another MVC derivation, however the difference is it assumes the developer does not have access to controlling the model, and instead assumes it works properly and controls it in the Presenter stage. This is usually used when developing an application around data retrieved from a third party. We do have control over our model state, so we will not use the MVP design. The Layered Abstraction pattern has four layers in its design: Applications, Application Framework, Libraries, and the Linux Kernel. All components work in the same layer of abstraction, and the layer below provides functionality for the layer above. This is an excellent approach for large projects, but far too much overhead for our needs.

The Ground Control Station we are designing should take in raw data from the camera and the sensors, organize and control that data, and then display it to the user. With these needs, the Model-View-Controller design pattern fits the best and is what we will go with. There will be models that hold data from sensor inputs, and models that hold raw image data from the Logitech HD C270 camera. There will be controllers that manage each model, and also a higher-level overall sensor controller that extends each of the individual sensor controllers. This will make it easy to add or remove sensors from the design without impacting the rest of the application. Figure 4.1.3.2A illustrated the high level design for the application.

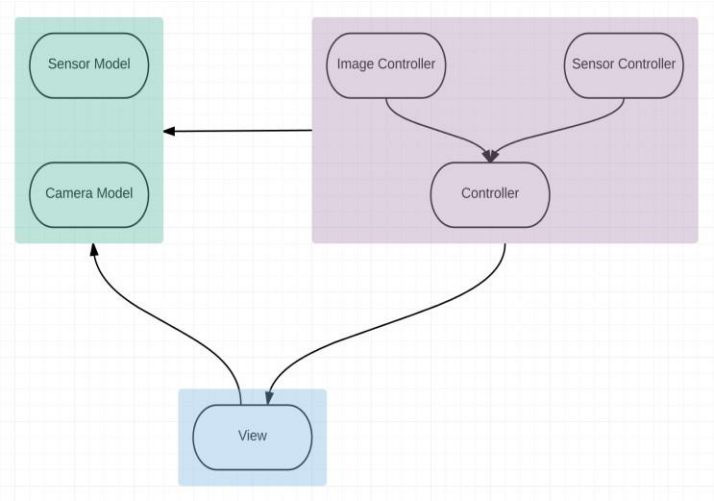


Fig. 4.1.3.2A High Level Design

Table 4.1.3.2B shows which aspect of MVC will handle which portion of the application

Interface	Description
View	Formats and sends the data to the user's view in the Android application's Graphical User Interface (GUI)
Camera Model	Holds the current frame of data sent by the Tau2 infrared camera
Sensor Model	Holds data from the sensors on the drone, organized into an object-oriented class hierarchy
Controller	Serves as an interface between the individual controllers (Image, Sensor1, Sensor 2, Sensor N, etc.), and the Model/View. All individual controllers are extended from this one
Image Controller	Makes any necessary adjustments to the image before sending to the view, manages the framerate, and offers potential debug options
Sensor Controllers	Manages, organizes, and presents the relevant data from the Sensor models to the View for display on the Android application

Table 4.1.3.2B: Relation between MVC components and their functions

By choosing to go with an MVC architecture, and with our specific implementation, we will allow many aspects of this application to be reusable. By having a high-level Controller class we can use the same classes and methods to create as many sensor or camera controllers as we would like. In addition to this, while the drone is currently being design according to specific sensor, camera, and other hardware requirements, the object-oriented nature of the design will allow for an ease in adapting to potentially different hardware interfaces. With each component being encapsulated, and our requirements being clearly defined, testing individual parts will be very easy. Unit testing will be done throughout the project and several integration tests will be completed near the end of the project. In terms of performance, many of the future performance issues are difficult to predict, and likely to occur when streaming real-time thermal imaging video from a drone to a smartphone. Developing, testing, and optimizing on a three-year old smartphone will allow the team to get a good idea of the minimum standards in which the application will run.

4.2 Hardware Design

4.2.1 Quadcopter

The hardware design for the quadcopter isn't too intensive when dealing with parts the quadcopter alone. For example, most all quadcopters include a frame, motors, electronic speed controllers, batteries, and some sort of flight control. The following sections breaks down all of the hardware components used to design our quadcopter, why we chose each component, and what it does for our project.

4.2.1.1 Frame

The UAV frame is one of the largest and most important design components of the FXUAV system due to its underlying effect on the lift capacity of the system, maneuverability attempts versus successes, number of motors needed, thrust, etc. That being said a prefabricated quadcopter frame, the Tarot 650 Iron Man 4-axis Folding 3k Carbon Fiber Aluminum Tubes RC Quadcopter Frame Kit, was of choice for the FXUAV.

The Tarot frame is specifically designed to incorporate a GoPro camera for first person view however please refer to sections 3.2.1.6 and 6.1.7.1 for further details concerning the FXUAV camera and camera attachment, respectively.

The Tarot Iron Man 650 Rack is based on Toray carbon fibers. It has adopted the Toray carbon fiber baseline which is known for its high quality, stable composite properties, uniformity, and reliability. More specifically, the Tarot 650 Iron Man Rack uses the Toray 3K carbon fiber cloth woven carbon fiber board as well as the Toray 3K hollow twill carbon fiber tube.

The frame design allows for complete folding without disassembly which helps support transferability requirements. Its 4 axes are configured to form the Quadcopter X configuration which helps support the multirotor copter design. Its center chassis holds a prefabricated camera mount that holds the FXUAV camera responsible for detecting the fire.

The FXUAV frame is designed to incorporate proximity sensors for easy of flight navigation and maneuverability. Although the exact location of each sensor is yet to be determined, they placed on the frame of the quadcopter in order to achieve the best possible obstacle detection and avoidance reactions as possible.

4.2.1.2 Motor

The brushless motor purchased for the FXUAV, was the MT3515 KV400 manufactured by Tiger Motors. The MT3515 KV400 is an Outrunner motor, meaning that the rotor in spinning outside of the Stator. Because of this configuration the magnetic force is applied to the shaft from a greater distance, giving it the ability to produce more torque. The stationary stator of this motor is 35mm in diameter and 15mm in length with a 5mm shaft, while the rotating rotor is 42.5mm in diameter and 37.5mm diameter in length. Because the number of coils on each of the Stator stack is unknown, it is not possible to calculate the total force that is applied on the rotor. Fortunately, the manufacture provides a data table with the parameters of the motor under certain conditions. Figure 4.2.1.2A: below shows the overall hardware design of the MT3515 KV400 Motor.

Figure: 4.2.1.2A: Overall hardware design of the MT3515 KV400. Reprinted with permission from T-Motor. (Pending)

According to its specification, the MT3515 KV400 motor with a zero load has the maximum of 8,000 rpm. According to the manufacture's datasheet, because of the 15*5" propellers that is being used in the FXUAV the maximum rpm the motor is able to generate a maximum rpm of 6500 with 22.2 Volts applied using 376 Watts of power. The main important parameter of this motor and propeller configuration, is the amount of thrust it is able to generate to lift the multirotor copter. According to the manufacture's datasheet provided this motor and propeller configuration each outputs a total of 0.97Kg of thrust at 50% of throttle to 2.46 Kg of thrust at 100% throttle. With this design, the FXUAV is able to hover with the throttle of the motor just over 50% only using 87 Watts of power per motor. This motor and propeller combination was chosen to maximize agility and flight time of the FXUAV so it can meet of the necessary flight requirements efficiently.

The efficiency of a brushless motor is important to maximize the flight time of the multirotor copter while not overstraining the motors. The efficiency of a brushless motors is specified the total power outputted by the motor versus the power provided to the motor. According to the data sheet of this motor and propeller design, the efficiency of the MT3515 KV400 motor with the 15*5" T-Motor propeller is 11.5 g/W at 50% throttle and 6.52 g/W at 100% throttle. Another way to measure the efficiency of the MT3515 KV400 motor is using Equation: below.

$$\text{Efficiency} = \frac{\text{Power Out}}{\text{Power In}}.$$

Where the power out of the motor is the power into the motor minus the power loss to heat and the power loss to the electromagnetic field. The power loos to heat is represented by the square of the current provided to the motor time the internal resistance of the motor, and the magnetic loss is the voltage provided to the motor times the current needed for the motor to run when it's not carrying a load. The manufacture does not provide the current needed for the unloaded motor, but according to the manufacture's data sheet the internal resistance of this motor is

0.115 Ohms. The final power efficiency of the motor can be calculated using Equation: below.

$$\text{Efficiency} = \frac{VI-(I_2 \quad R+VI_{u\square\square\square\square\square})}{VI}$$

This final efficiency value was calculated during the testing phase of the FXUAV hardware in Section: But according to the datasheet, this motor and propeller combination provided enough thrust to lift over 10 pounds, with payload efficiently enough to have a flight time of at least 10 minutes which meets all of the requirements for the FXUAV. The power provide to each of the MT3515 KV400 motor is delivered by the electronic speed controller circuits (ESC). Each motor is connected to its ESC with 3 16AWG wires that plugs into the ESC 3.5mm gold connectors.

4.2.1.3 Electronic Speed Controllers

The FXUAV needs to power the DC brushless motors using an Electronic Speed Controller (ESC), the choice of ESC for this project was the Hobbywing XRotor 40A ESC. Since a DC motor uses electromagnetic force to rotate, it needs an electronic speed controller that controsl the movement of the motor by applying electricity to the electromagnets in the motor. The coils in the electromagnets are composed of three wires that are connected to the Electronic speed controller. As shown in Figure 4.2.1.3A, in the inside of the motor they are terminated either using a WYE WIND or a DELTA WIND, which is why brushless motors is also known as tri-phase motors.

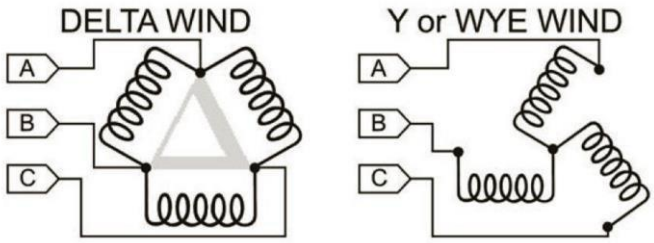


Figure 4.2.1.3A: Brushless DC motor Coil Wire Configuration Reprinted with permission from Greg Covey. (Pending)

When the ESC unit provides power to any one of the phases it creates an electromagnetic force that is used to either push or pull the pole of the motor in a given direction. Since there are three phases in the motor, the ESC unit applies a total of six different power combinations or winding phases for each pole to move the motor in one full rotation. Since the MT3515 has 12 poles that needs to be controlled by the ESC with a maximum of 6,500 rpm for our multirotor configuration, the ESC provides 234,000 electrical cycles to the motor per minute for it to operate correctly. ESC units control these cycles using Pulse Width

Modulation, and the faster it modulates the signal the higher the throttle of the motor is.

Modern ESC units are designed using a microprocessor, and MOSFETs that provide the necessary power to the right coil at a given time. Figure4.2.1.3B shows a basic diagram of an ESC unit, which contains a microprocessor, a rotor position circuitry, a MOSFET drive circuitry, and motor position circuitry.

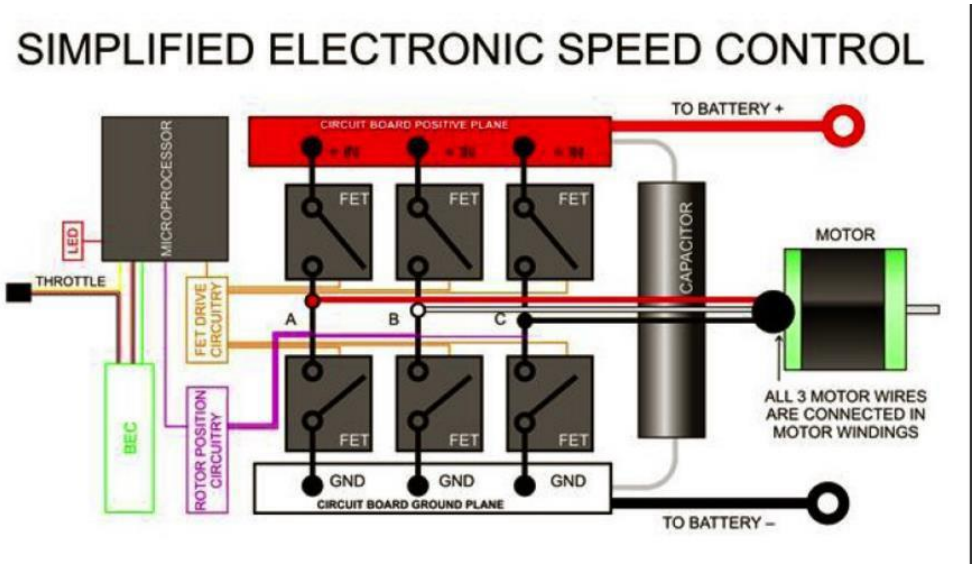


Figure 4.2.1.3B: Simplified Electronic Speed Control Unit. Reprinted with permission from Greg Covey. (Pending)

The microprocessor in the Hobbywing XRotor 40A ESC has a proprietary firmware that it uses to calculate the outputs signals according to the input it receives. The microprocessor processes the input signal from the Pixhawk flight controller unit, from the MOSFET drive circuitry, and the rotor position circuitry, and calculates the signal required to output to the motor. The output signal then provides the voltage necessary to the MOSFET gate that needs to be turned on or off for the operation to continue. Figure 4.2.1.3C below shows a picture of the Hobbywing XRotor 40A pro ESC that was used in the FXUAV.

Figure 4.2.1.3C: Hobbywing XRotor 40A pro ESC. Reprinted with permission from Hobbywing. (Pending)

The three 3.5mm female connectors was connected to the motor, the 16AWG black and red wires was connected to the power distribution board, and the white and black wires are going to be connected to the servo outputs of the Pixhawk flight controller. The ESC is 73.5mm in length, 21.8mm in width, and 11mm in height. The circuit is protected by an aluminum aired heat sink to keep the micro controller from overheating, and two 220 micro Farad capacitors that stabilizes the voltage from the battery and protected it from the ripple effect.

4.2.1.4 Flight Control

The main and most important component of the FXUAV is the flight controller unit. Because of the popularity of multirotor copter systems, there are dozens of different flight controller available in the market. Group 7 decided to purchase the Pixhawk open hardware and software system as a flight controller for the FXUAV. The Pixhawk flight controller uses the Ardupilot open firmware system with an open software ground station controlling its flight operations. The full specifications of the Pixhawk is listed below.

Processor:

- 32bit STM32F427 Cortex M4 core with FPU
- 168 MHz
- 256 KB RAM
- 2 MB Flash
- 32 bit STM32F103 failsafe co-processor

Sensors:

- ST Micro L3GD20H 16 bit gyroscope
- ST Micro LSM303D 14 bit accelerometer / magnetometer
- Invensense MPU 6000 3-axis accelerometer/gyroscope
- MEAS MS5611 barometer

Power System and Protection:

- Ideal diode controller with automatic failover
- Servo rail high-power (max. 10V) and high-current (10A+) ready
- All peripheral outputs over-current protected, all inputs ESD protected
- Power Supply

Normal Operation Maximum Ratings:

- Power module input (4.1V to 5.7V)
- Servo rail input (4.1V to 5.7V)
- USB power input (4.1V to 5.7V)

Absolute Maximum Ratings

- Power module input (0V to 20V)
- Servo rail input (0V to 20V)
- USB power input (0V to 6V)

The Pixhawk was mounted on the center of the Tarot 650 frame using dampers to help minimize vibrations effect of the multirotor copter on the microcontroller.It receives input signals form the Mission Planner software and the Raspberry Pi through its I2C receiver which control its flight path. The Raspberry Pi receives inputs from the Logitech web camera, and proximity sensors that creates interrupts when necessary so it can succeed in its mission of detecting and putting out a fire autonomously. Figure 4.2.1.4A: shows the layout of the Pixhawk interfaces.

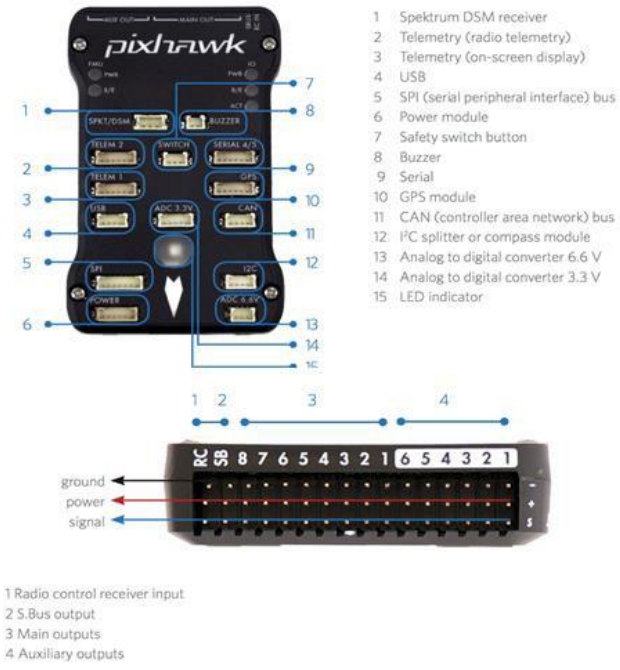


Figure 4.2.1.4A: Pixhawk Interfaces. Reprinted with permission from Pixhawk.

Final interface connectivity between the Pixhawk and its components will be further described in Section 6: Hardware Installation.

4.2.1.5 Power Distribution

The power distribution board controls the voltage required by the 3 different loads with a single source input. The total voltage that is supplied by the battery is 22.2 nominal Volts and the maximum drain current is 90 Amps. This design was partially made using Texas Instruments WebBench. The converter used is the LM3150 which supplie power to the Pixhawk, RaspberryPi, and the servo motor of the fire extinguisher system. The second part of the power distribution board is a voltage and current sense module. This module which was designed by Attopilot provides the Pixhawak the total current being used by the system, and the voltage on the battery. The power distribution board also provides power to the four different ESC units without any regulators. Figure 4.2.1.5A below shows the overall block diagram of the power distribution board.

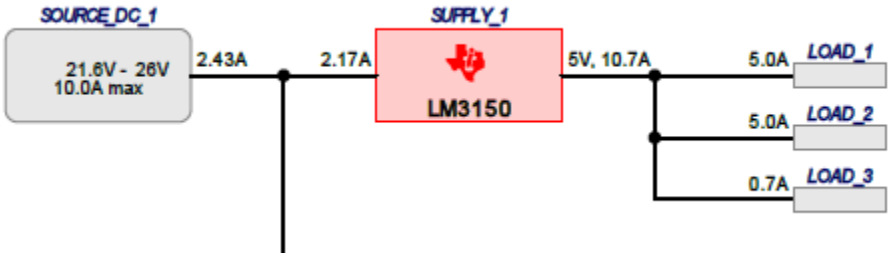


Figure 4.2.1.5A, Power Distribution Block Diagram. Courtesy of Texas Instruments.

Figure 4.2.1.5B below shows the schematic diagram for the LM3150 which is used to support the Pixhawk, RaspberryPi, and the Fire Extinguisher system servo motor.

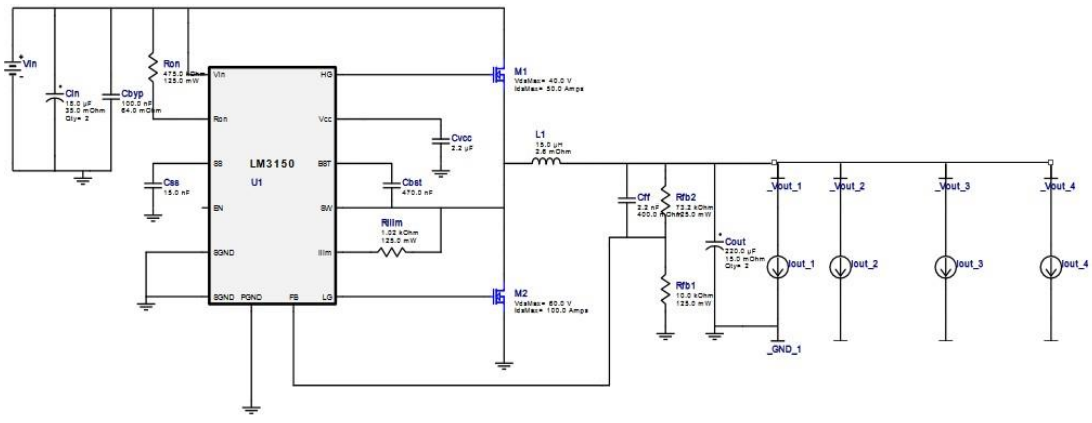


Figure 4.2.1.5B: LM3150 Schematic Design. Courtesy of Texas Instruments

The current and voltage sense which is the second part of the power distribution board, uses the Texas instrument INA 169 current shunt sensor monitor. This monitor uses a differential input op-amp and provides the pixhaw with a reading for the total current being used by the system, while also providing the total voltage level of the LiPo battery being used by the quadcopter. Figure 4.2.1.5C shows a basic schematic of the shunt current monitor.

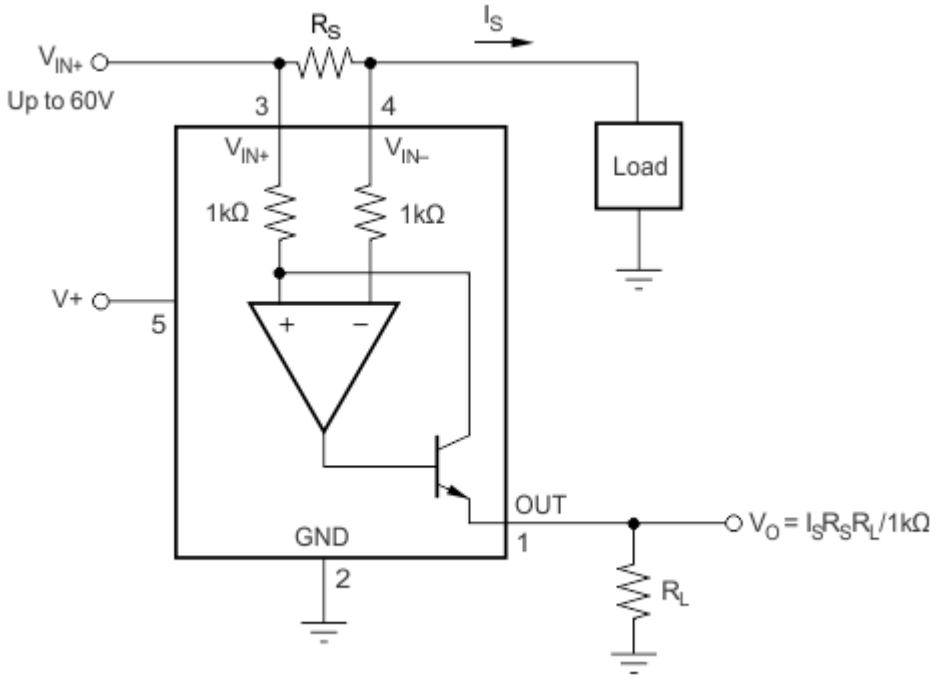


Figure 4.2.1.5C: Current Shunt Monitor. Courtesy of Texas Instruments.

The two different schematics of the power distribution board, was integrated into a single printed circuit board that was be mounted to the FXUAV unit. The final PCB design was be created using CADSoft Eagle Freeware, and the schametic provided by Texas Instruments. Since the PCB handles over 80 amps of power, the bread board file was manually created by Luis Brum. The required trace width was acquired using the 4PCB.com online calculator, and the minimum required trace That provides current to the three different loads is 15mm with a .4mm copper pour. This is a two layer circuit board that is able to support the maximum thermal requirements of the high current drain from the ESC units. Finally it was also important to keep the output fairly separated from the switching regulator due to the noise that the regulator makes. This noise was reduced by using a Coilcraft inductor that help clean out the ripple effect of the regulator. Figure 4.2.1.5D shows the two different boards that was integrated in a single PCB unit.

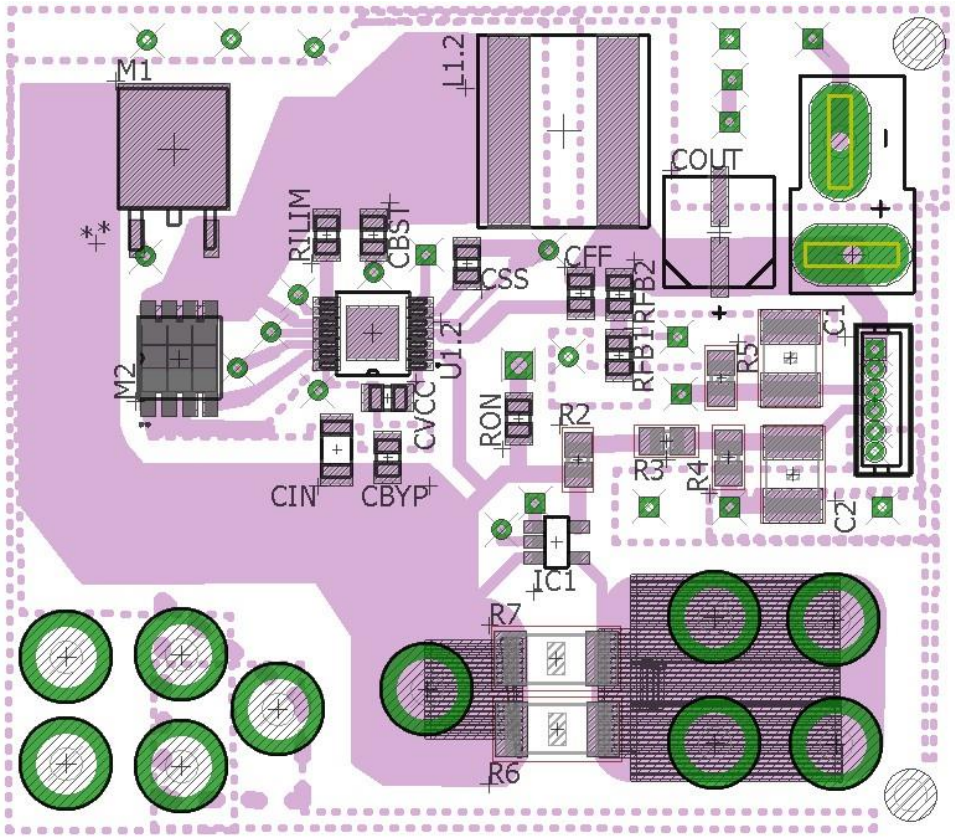


Figure 4.2.1.5D: PCB layout.

4.2.1.6 GPS

The final component necessary for the FXUAV to fly autonomously is the 3DR uBlox GPS unit. The GPS is composed with the NEO-7 GNSS module, HMC5888L digital compass, Taoglass 4mm thick GPS Patch Antenna, MAX2659 GPS/GNSS Low-Noise Amplifier, and 2 indicator LEDs. Figure 4.2.1.6A shows the PCB board with all the components attached

Figure 4.2.1.6A: 3DR Ublox GPS with Compass unit. Reprinted with permission from 3DR.

Features and Specifications:

- u-Blox NEO-7 module
- 5 Hz update rate
- 25 x 25 x 4 mm ceramic patch antenna
- LNA and SAW filter
- Rechargeable 3V lithium backup battery
- Low noise 3.3V regulator
- I2C EEPROM for configuration storage
- Power and fix indicator LEDs
- Protective case
- APM compatible 6-pin DF13 connector
- Exposed RX, TX, 5V and GND pad
- 38 x 38 x 8.5 mm total size, 16.8 grams.

The GPS was mounted on a mast, which provides enough distance from the battery and motors. This prevented electromagnetic interference that are created from the power sources that can corrupt the compass data. It connected to the Pixhawk using two different cables, a five-position-to-six-position that connects to the GPS module interface, and a four-position cable that connects to the I2C interface. Once connected and calibrated properly, the GPS LED indicator has a solid red light to show it is receiving power, and a flashing blue LED that signal the GPS has acquired a lock. The calibration of the GPS unit was done through the Mission Planner software, and further instructions on the initial set up is given in section: Hardware installation and section: Software installation.

4.2.1.7 MCU

The FXUAV has three large, distinct processing tasks that largely contribute the either the success or failure of the system. These three distinct processing jobs lie in image processing, flight path processing, and proximity sensor processing. Image processing is the only task that is completed on ground level at the ground communication station. Both flight path processing and proximity sensor processing are both done onboard the FXUAV. Furthermore, both tasks are controlled via one, onboard processor, the Rasberry Pi onboard computer. That being said however, the image processing module and the Rasberry Pi communicate via wireless communication signals. The Pixhawk flight controller was connected to the Raspberry Pi via a serial data bus and the processor contains a sensor hub in order to control each and every proximity sensors that is located on the FXUAV frame. The camera mounted on the FXUAV is also controlled via the Raspberry Pi. The Raspberry Pi receives video streams from the Logitech web-camera via a USB port and it sends these streams of data wirelessly to the image processing unit located at the ground station.

Once the object is detected the Raspberry Pi signals the Pixhawk Flight Controller that the waypoints it receives from the mission planner software need to be ignored and instead it must navigate the quadcopter to the fire that was detected. In order to establish this connection between the Pixhawk flight controller and the

Raspberry Pi, the I^{2}C serial interface protocol was used. i2c0, i2c1, and i2c2 are the available I^{2}C interface busses on the Raspberry Pi. The second bus, i2c1, was used for serial communication between the flight controller and the Raspberry Pi for the sole reason that the expansion header does not expose the i2c0 bus. The Raspberry Pi has several connectivity ports (Figure 4.2.1.7A) including a USB host, USB client (for power and communications), Ethernet, HDMI, and 2x46 pin headers. Figure 4.2.1.7A: Raspberry Pi Pin Headers below, shows the 2x46 pin headers and their locations. The i2c1 serial interface bus utilizes pins located on the P9 header; pins 17 and 18.

Pins 17 and 18 can serve multiple functionalities and therefore have to be configured as I2C1_SCL and I2c1_SDA, respectively. I2C1_SCL pin controls the I^{2}C bus serial clock and I2c1_SDA pin is responsible for controlling the serial line data. Upon startup, the device name may not be physically mapped directly and therefore the I^{2}C serial bus interface protocols may be named incorrectly. If this is the case, each buss can only be identifiable via their memory address (see Table 4.2.1.8B: I^{2}C Bus Address

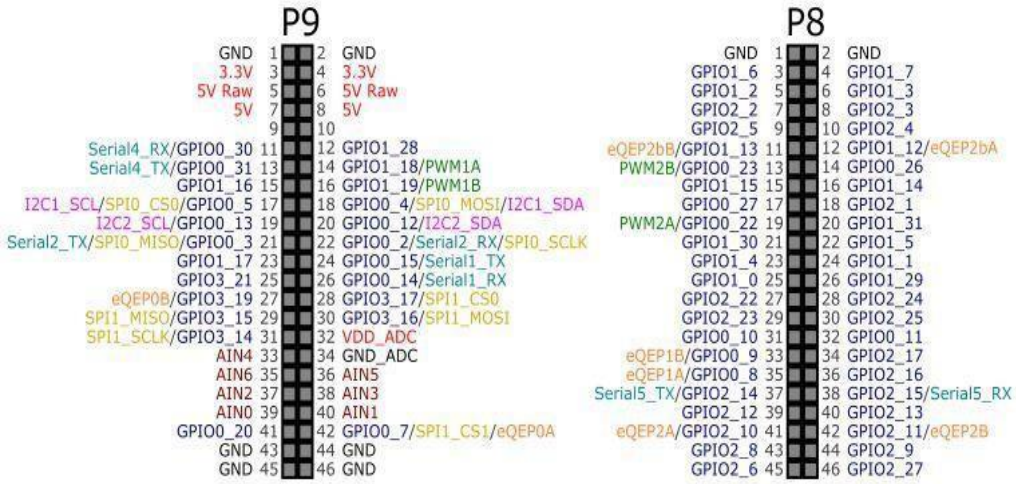


Figure 4.2.1.7A: Raspberry Pi Pin Headers

i2c0	**0x44E0B000**
i2c1	0x4802A000
i2c2	0x4819C000

Table 4.2.1.7B: I^{2}C Bus Address

The Raspberry Pi development platform runs on a 16MHz processor and operates at 3.3V. The Pixhawk Flight Controller operates at the same value of

3.3V and therefore no power conversion or level shifting is needed between both devices.

4.2.1.8 Focal Length and Perceived Object Size Calculations

The given focal length is the distance from the image sensor to the lens when a given target is in focus. Figure 4.2.1.8A illustrates this dimensions and the relative positioning of the image sensor and lens with respect to the focal length.

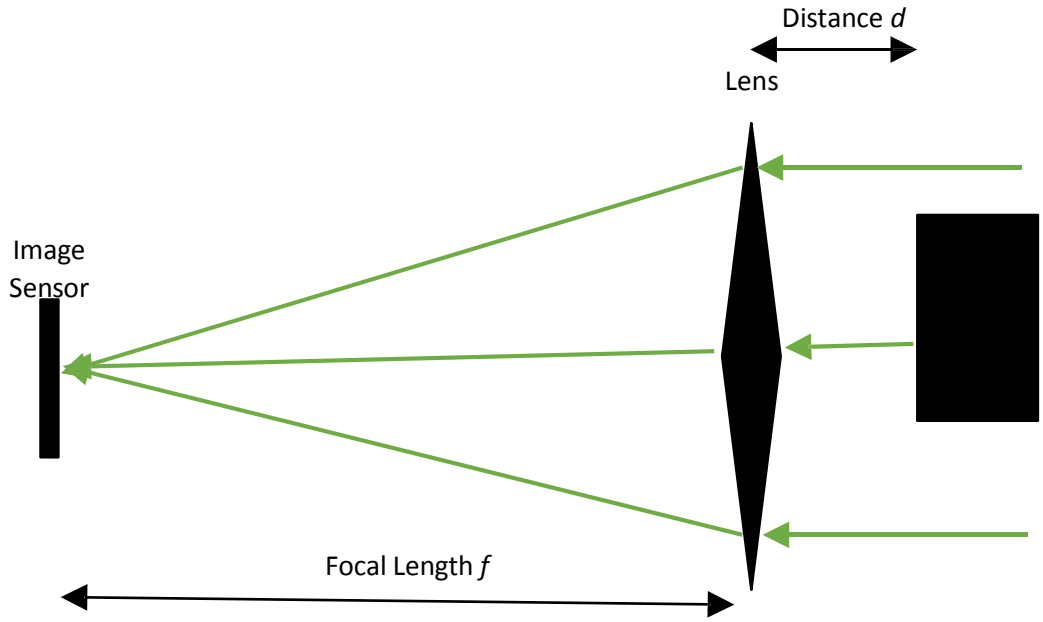


Figure 4.2.1.8A: Focal length illustration with respect to image sensor and target

The respective relationship between the focal length *f*, distance *d*, and object size *s* is demonstrated in the following equation. It equates the projected size of the object in focus.

$$\frac{s}{d} = \frac{s_p}{f}$$

In order to find the size of the projected object in terms of the number of pixels it occupies, the following equation is used, where s_p is the same variable as defined above, a_{tot} total image sensor size, and p_{ob} is the number of pixels that make up the object.

$$\frac{s_p}{a_{tot}} * p_{tot} = p_{ob}$$

To simplify the calculations, we can combine both equations above to obtain the resulting equation:

$$\frac{\text{object size} \times \text{sensor size}}{\text{object distance} \times \text{focal length}} = \text{pixels in object}$$

As mentioned in section **3.2.1.6**, the focal length has a direct effect on the size of the perceived object. Furthermore, as can be seen from the above equation, if the focal length were doubled the resulting effect on the perceived object size would be the same as if the distance between object and the lens was decreased by 50 percent.

To determine the size of the projected object given a certain focal length, a few vital pieces of information need to be known such as height, camera position, etc. It was determined that the FXUAV should fly at an altitude of approximately 10ft (3.048m). The detection unit, the Logitech HD C270, was mounted flush to the underside of the quadcopter frame, hence pointing a full 90 degrees downwards. The below calculations are done assuming a 9mm lens was mounted on the C270. Also, assume the height of the grease fire is 1ft (304.8mm) and situated in the center of the camera's field of view. The resulting distance between the camera and the grease fire would therefore be 10ft or 3048mm.

From Table 3.2.1.6D Focal Length vs. FOV, for the Tau2 640 as it has been decided, it is seen that the pixel size is 17µm and the image sensor is therefore 17µm * 512 pixels = 8.7mm. If the fire is 6in x 6 in when looking from above and P_{tot}, the total number of pixels in the vertical direction of the image, is 512 then the resulting perceived size of the object's height is equal to

$$\frac{9\,mm \times 152.4\,mm}{3048\,mm \times 8.7\,mm} \times 512 = 26.48 \sim 26\,pix$$

4.2.2 Fire Extinguishing Unit

4.2.2.1 Release Function

The hardware aspect of the fire extinguisher release function revolves around a hinge, a servo motor, and spring loaded door. This will create a door effect for the container in which the contents will released out of. In order to keep the contents within the enclosure, an electrically controlled device is needed to keep the door like figure closed, and only open when triggered. Shown below in Fig 4.2.2.1A is a drawing of the servo door hinge concept. In this figure, once the servo rotates, it will pull the door open, pause then rotate back using the spring pressure to close it once again.

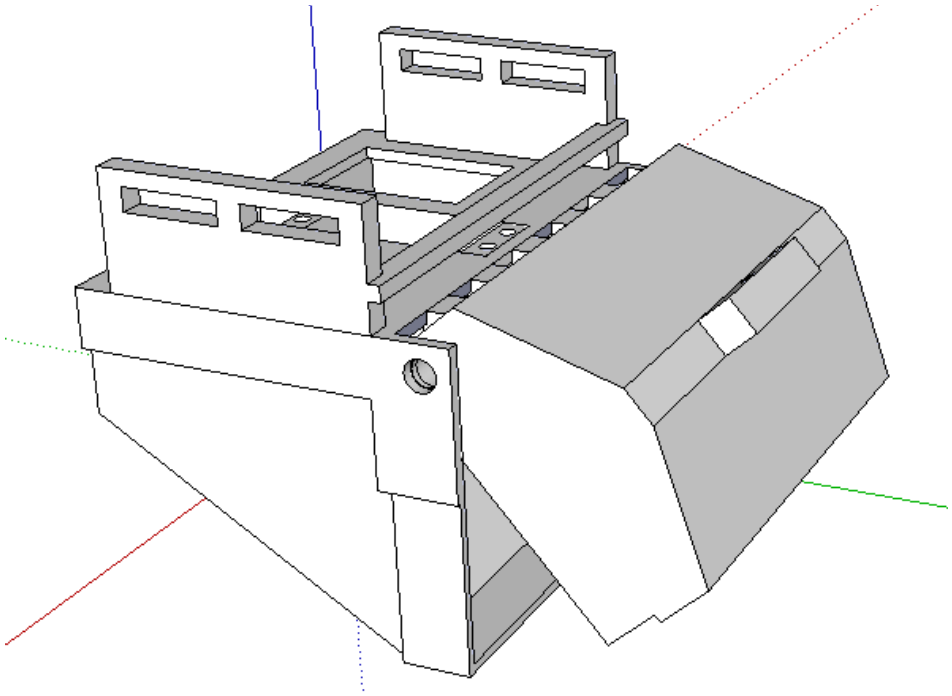


Fig 4.2.2.1A: Trap Door Illustration

4.2.2.2 Dimensions

The dimensions of the container are designed to fit at least one half pound of baking soda. The calculations in section 2.2.4 explain the necessary volume of the container, 11.5 cubic inches. The dimensions however can vary to better suit our needs. This will depend on the shape of the container. By making the container a wedge shape, shown in figure 4.2.2.2A, and having the pointed end face down towards the ground, it would prove more efficient at dispensing all of the contents since it will all be forcing it's way downward due to the shape of the container. However, one drawback with the wedge shape is volume. The smaller most efficient shape for volume for the drone is a square. This way we do not need to compensate for the decreasing volume in the point of the wedge by extending the length. With a square or rectangle, the volume can be easily judged and fabricated.

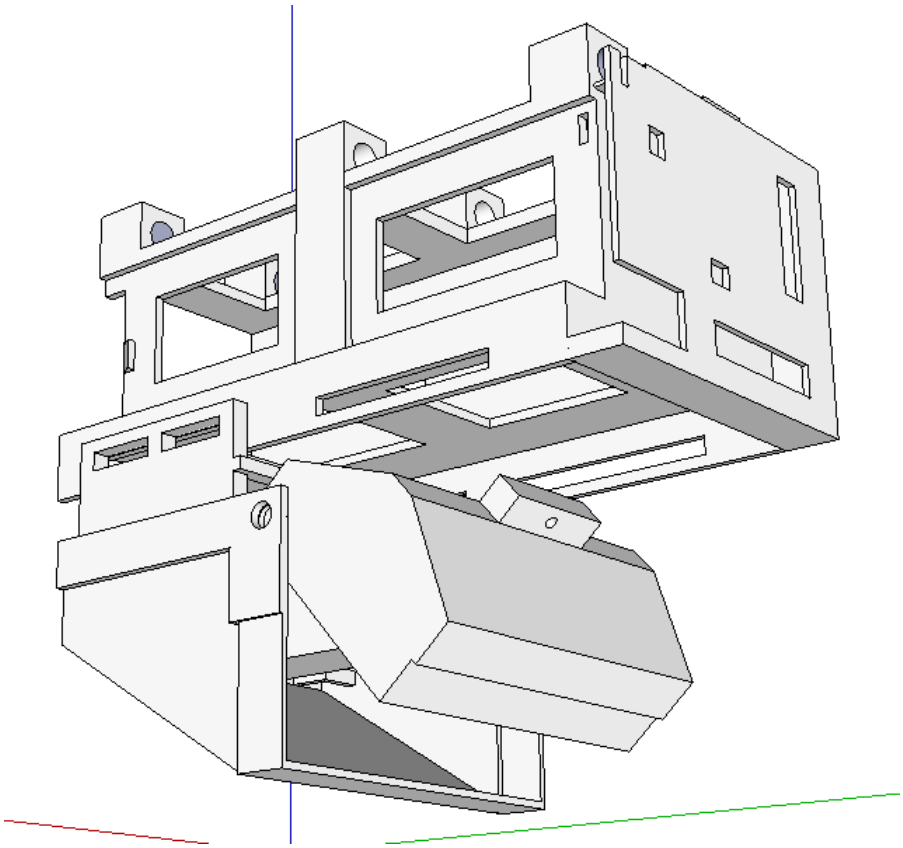


Figure 4.2.2.2A: Wedge Container Underneath the Battery Container

To calculate volume for a figure like on shown in Figure 4.2.2.2A, we need the volume formula for a wedge, which is $V = \frac{1}{2} * * h$ where a, b, and h are the respective sides of the wedge. A sample of dimensions to meet 11.5 cubic inches would be a = 2.875", b = 2", and h = 4". However, this became a bit of a problem when trying to load the cartridge with baking soda. We created a hole in the top but the battery case seemed to be in the way. Nice complex design but we had to fill it up, upside down in order to be the most efficient with filling our payload.

4.2.2.3 Capacity

Determining capacity was much more difficult than originally thought. There are many constraints to consider when deciding the capacity of the payload. Some key aspects that contribute to our decision are maximum thrust, delivery distance and size of target. Since this project will be scaled down for sake of demonstration, our target will be much smaller than a real fire. Realistic scenarios can be anywhere from grease fires, to forest fires, to large scale brush fires. However, as previously mentioned, for the sake of presentation, time, and financial constraints, the target will be that of a tea candle size.

Additionally, maximum thrust was a major component in determining our payload capacity. Too much volume will not allow the quadcopter to even lift off the ground. Too little and the chemical will be dispersed into the air such that it will not affect

the flame. At 100 percent, our motors can produce is 2.4kg per motor. This translates to roughly 5.2 pounds, per motor. However, we don't want to operate at 100 percent thrust 100 percent of the time, otherwise our flight time is decreased dramatically. At 50 percent thrust, each motor is capable of 970 grams, which translates to about 2.1 pounds. In order to conserve energy and err on the side of caution, we will assume we will be operating with 8 pounds of thrust. Given that amount a half pound payload seems substantial enough to hit the target, and leave enough room for comfortable lift off and maneuverability.

Finally, distance delivery comes into play. If we deliver the payload from 20 ft., our accuracy decreases along with the likelihood that the chemical will remain intact enough to sufficiently suppress the flame. Dropping the payload too low can cause issues with the quadcopter being too close to the flame, as well as extinguishing the flame from the propeller wash. Therefore, we've decided that 10 feet will suffice for such an application. This is assuming a half pound capacity of chemicals.

4.2.2.4 Contents

As discussed in section 3, Research, There are many types of fire extinguishers: dry, wet, even sound. But which of this will produce our desired result. We found Sodium Bicarbonate (baking soda) was the best choice for this project. The ease of access, the performance in fire suppression, and the weight properties all line up for our intended purposes. Baking soda is a crystalline powder that is used for many things including, cleaning, odor neutralization, and even fire extinguishers.

Baking soda releases carbon dioxide when heated. Carbon dioxide, while heavier than air, helps in assisting the smother the fire. Thus, making it a rather effective fire suppressant.

While some chemicals are better at extinguishers different types of fires, baking soda has proved to perform well for our intended tea candle or small controlled grease fire. The fire by design will be small enough and of the correct nature, such that our payload of baking soda will possess absolute ability to extinguish the fire.

4.2.2.5 PCB for the Fire Extinguisher Release

The design for the Extinguisher release requires a separate PCB and MCU in order to trigger the release function. Once again using EAGLE software helped in designing a barebones PCB that can control the release. Shown below in Figure 4.2.2.5A is the schematic for the PCB that is also using the ATMEGA 328 similar to the schematic shown in Figure 3.2.2.1C. This schematic is a very barebones layout which should suffice for the servo controller. It is necessary to keep things as simple as possible in order to reduce space, and therefore reduce cost.

Figure 4.2.2.5A: PCB Design Schematic and Figure 4.2.2.5B, the board layout of the schematic shown in 4.2.2.5A is shown. This board layout is not 100% efficient as layout design is subjective. However, this board will prove to be within our size and budget and no more optimization is necessary.

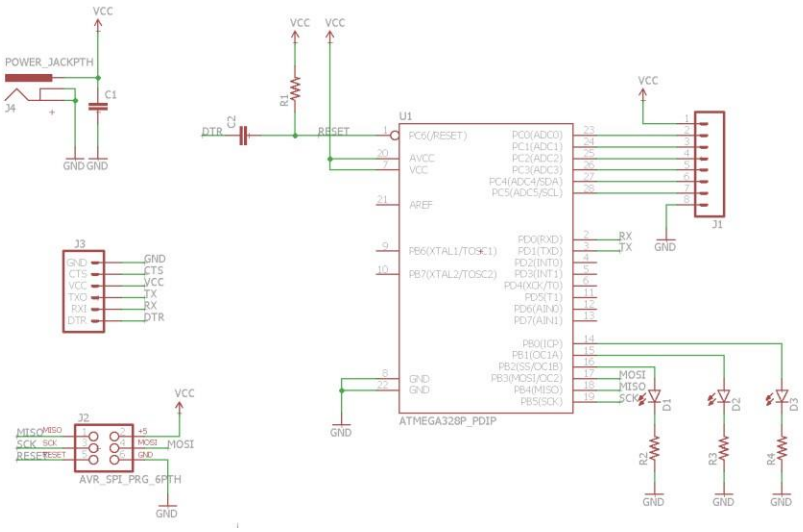


Figure 4.2.2.5A: PCB Design Schematic

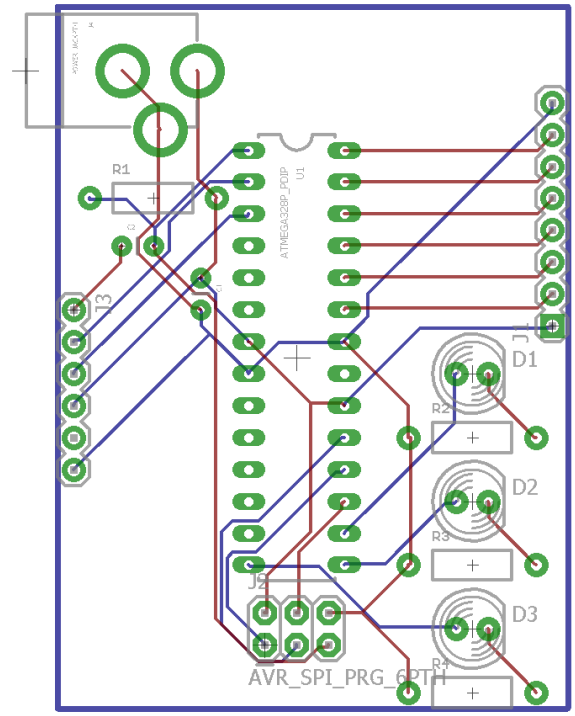


Figure 4.2.2.5B: The PCB Design Layout

Due to our selection changes of the companion computer, we no longer required the use of a PCB servo controller. The Raspberry Pi is capable of PWM output through one of the several GPIO pins.

The last step in creating the PCB is to add copper pours and silk screening to add finishing touches and customization. Figure 4.2.2.5C: The Adamuino is shown below and is ready to be generated into a gerber file and sent to be printed.

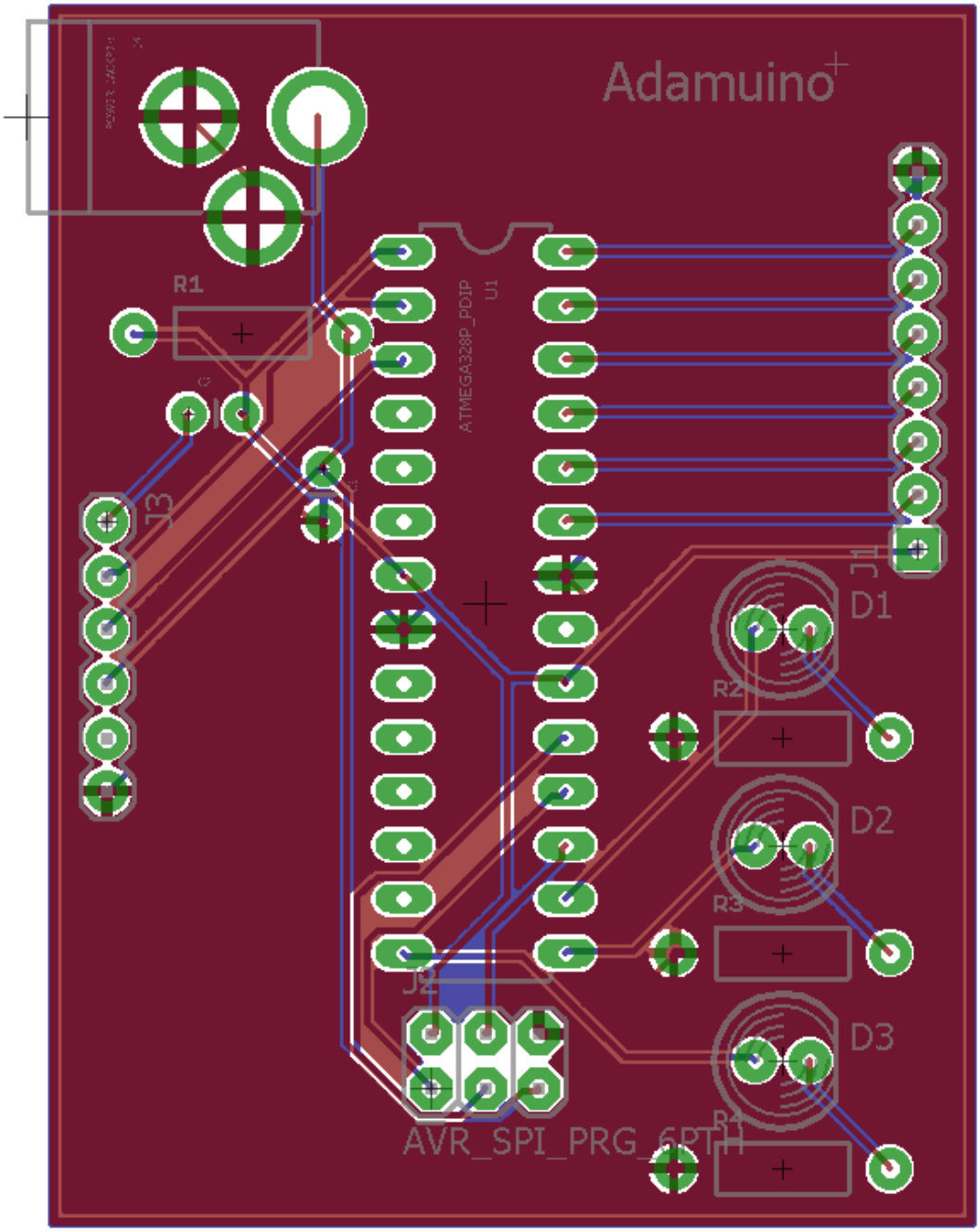


Figure 4.2.2.5: The Adamuino Finished PCB Product

4.2.3 Wireless Communication

The FXUAV has three different wireless communication systems on board, one for telemetry data, one for live video data, and the third one for the control signals. These systems transmits data simultaneously between the FXUAV and the ground station so the FXUAV can autonomously navigate and put out the fire successfully.

Radio Control Communication

The radio control system is the main wireless communication system that sends signals from the ground control to the FXUAV. For this project, the FXUAV uses is the RadioLink AT10 transmitter and the R10D receiver.The transmission operates on the 2.4GHz frequency spectrum with 10 channels using Frequency Hopping Spread Spectrum technology.

Transmitter Specifications:

- Encoding PCM 10 channels and 8 channel PPM(2.4Ghz)
- 8 internal model memories
- 167x34mm LCD 8 lines, 22 characters with adjustable contrast
- Double deflections (D / R) and a timer
- 11.1v 1700mAh LiPo battery

Receiver Specifications:

- 2.4GHz 10 channels PPM
- Dipole single wire antenna
- Operating Voltage 4.5V-6V

The Radiolink AT10 2.4GHz radio manual states the maximum range for the receiver to be 1000 meters in the air. This range is well above the necessary range of 500 feet as stated in the requirements for the FXUAV. The receiver was mounted on the TAROT 650 frame of the FXUAV, and the antenna was placed in the back of the frame to maximize the range. It is connected to the Pixhawk through the RC input pins that is located on the side of the flight controller.

Telemetry Radio

The second wireless communication system for the FXUAV is the 3DR Telemetry Radio V2. This system is used for transferring the flight data from the multirotor copter to the ground station. The ground station uses the telemetry data to calculate the multirotor's position and then uses that data to provide the correct signal using the autopilot software. This radio is an open source hardware and open firmware system.

Telemetry Radio Specifications:

- 100 mW maximum output power (adjustable)
- -117 dBm receive sensitivity
- 2-way full-duplex communication through adaptive TDM
- Frequency Hopping Spread Spectrum (FHSS)
- Error correction corrects up to 25% of bit errors
- Supply voltage: 3.7-6 VDC (from USB or DF13)
- Transmit current: 100 mA at 30 dBm
- Receive current: 25 mA
- Serial interface: 3.3 V UART
- 915 or 433 MHz
- 6-position DF13 connector

The FXUAV uses the radio using the 915 MHz frequency, on the ground station it will connect using its USB port while on the multirotor it connects to the telemetry interface of the Pixhawk using the 6-position DF13 connector. Figure 4.2.3A below shows the different interfaces and physical characteristics of the 3DR telemetry radio.

Figure 4.1.3A: 3DR Radio V2. Reprinted with permission from 3DR.

The final communication system was implemented in the FXUAV is the 8 channel 5.8GHz Boscam TS-351 transmitter and the Boscam RC305 video transmission system. The transmitter was mounted on the FXUAV and receives a National Television System Committee (NTSC) analog video format from the TAU2 camera. The power connection uses a Japan Solderless Terminal (JST) Female connector that was directly connected to the power distribution board system. Figure 4.2.3B below shows the transmitter Pin-out description of the Boscam TS-351Transmitter.

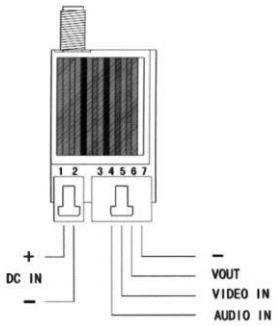


Figure 4.2.3B Boscam TS-351 Pin Layout. Reprinted with permission from Boscam. (Pending)

Transmitter Specifications:

- 7V to 12V operating voltage
- 150 mA operating current
- 22 dBm output power
- 8 MHz video bandwidth
- 6.5 MHz audio carrier frequency
- RP-SMA antenna connector
- 0.8 to 1.2 Vp-p video input level
- 0.5 to 2 Vp-p audio input level

The 8 Channel 5.8 GHz receiver was connected to the ground station using a 3.5mm male to RCA female cable for the video and audio signal, and it receives power from a battery using a JST connector. There are also Dip switches for frequency pairing between the receiver and transmitter, the proper configuration of the transmission will be done in Section: Hardware testing and it was set to the channel that provides the best transmission quality

Receiver Specifications:

- -90dBm sensitivity typical.
- 50 ohm RF port impedance.
- RP-SMA RF connector.
- NTSC / PAL compatible, standard 1Vp-p video (75Ω).
- Line Level Audio (10KΩ).
- Dual Audio-Video Output Jacks.
- 4.8 V to 5.2 V operating voltage
- 200mA operating current
- 1.3mm x 3.5mm micro barrel type power Inlet.
- JST/BEC power cable.

Figure 4.2.3C below shows the Dip switches transmission channel settings that was used to test the transmission quality of the 8 channel 5.8GHz Boscam TS-351 transmitter and the Boscam RC305 video transmission system.

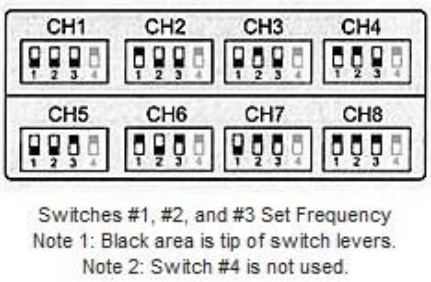


Figure 4.2.3C: Boscam RC 305 receiver Dip switch configuration. Reprinted with permission from Boscam. (Pending)

The video transmission was used so the operator of the FXUAV can oversee the process of the FXUAV while putting out the fire. The data from the Logitech camera is simultaneously transmitted to the Raspberry Pi which uses the data to navigate and control the FXUAV flight system when putting out the fire.

5 Integration and Design Summary

All integration hasn't been completed for the FXUAV. Although some software, hardware, and wiring diagrams, state machine diagrams, and other UML diagrams have been completed, everything has not been 100% documented. Various software and hardware aspects are still being considered, and therefore haven't been included in final implementation documentation.

The integration description is as follows. A quadcopter built from scratch will follow all hardware components to classify it as such, and then some. The frame will be equipped with the tested motors, esc's, custom designed power distribution board, battery, an onboard image processing MCU, a custom design PCB, and a fire suppressant release system, and of course a flight controller and FLIR infrared camera.

The basics of a quadcopter require four motors, a battery, some sort of flight controller, and power distribution for the battery and motors. The onboard MCU will process the images that are taken from the FLIR IR camera. The images will be processed such that the flame will have a distinct signature. The FLIR camera possess a feature known as isotherms that allows manipulation of a specific section of infrared footprint. The idea is that using OpenCV we will highlight the unique IR signature of the flame to stand out from the rest of the image. This will allow us to target the flame from everything else in the image. A classifier will then be trained to look for this distinction in each image and be able to locate the flame. At this point, we will command the quadcopter to maneuver until the flame is in the center of the frame. Once our target is in position, we will descend using an ultrasonic sensor to set the quadcopter 10 ft. above the ground, and release the payload.

The fire suppression release system will consist of a simple servo controlled by the GPIO pins from the Raspberry Pi and will receive a high input to release the chemical fire extinguishing agent. The system will also consist of a container to hold the payload, which will remain closed, until a high signal is received from the GPIO. This container will be attached to the quadcopter to deliver a complete package.

The motors will be controlled by our custom designed power distribution board, and the Pixhawk flight controller, and powered by a 10,000maH, 22.2V battery. Each motor will be connected to an ESC and individually wired to the power distribution board, which will then receive its power from the battery. The propellers will be attached to each motor, counter-clockwise, and clockwise in order to produce flight.

The flight control by Pixhawk will be responsible for stabilizing flight and will be powered by the battery as well. Additional sensors, ultrasonic sensors will also assist in object avoidance and will also be powered in the same manner. The ultrasonic sensors will detect a distance and decide, once within a certain threshold are we too close to an object or if we are free to move.

The Logitech C270 camera is responsible to imagining. The camera will be powered by the single battery and will be connected to the Raspberry Pi. The webcam will take video and the frames will be processed by the Raspberry Pi to detect the red contours.

The GPS unit will be used to navigate waypoints for the mission planner and will aid in locating geographically where the quadcopter is and where its destination is.

6 Prototype Construction and Code

6.1 Quadcopter Parts Acquisition and Assembly

6.1.1 Frame

The Tarot 650 Iron Man 4-axis Folding 3k Carbon Fiber Aluminum Tubes RC Quadcopter Frame Kit was used for the FXUAV system. It was purchased from Amazon.com and it cost a total of $95.00. The kit included a 4axis quadcopter which is designed to hang a camera device for first person view.

The Tarot 650 Iron Man 4-axis Folding 3k Carbon Fiber Aluminum Tubes RC Quadcopter Frame Kit was ordered by the FXUAV sponsor, FLIR Systems and therefore did not cost the FXUAV design team any money. It was delivered on Monday, March 28th, 2016 to FLIR Systems, who elected to pay for the Frame Kit as per our sponsor agreement. The Frame assembly is pictured in Figure 6.1.1A below.

Figure 6.1.1A Tarot Iran Man 650 Frame Assembly

6.1.2 Motor

The Tiger RC MT3515 KV400 were purchase from fpvmodel.com online website, the motors come with four M3 frame mounting screws, one propeller adapter, three propeller adapter screws and one motor mount. Figure 6.1.2A below shows all the parts included with each motor.

Figure 6.1.2A: MT3515 KV400 parts. Reprinted with permission from T-Motor. (Pending)

The motors was mount directly to the Tarot 650 frame using the M3 four frame mounting screws with the 3 wires that connects directly into the ESC facing the direction pointed by the arrow as shown in Figure 6.1.2B below.

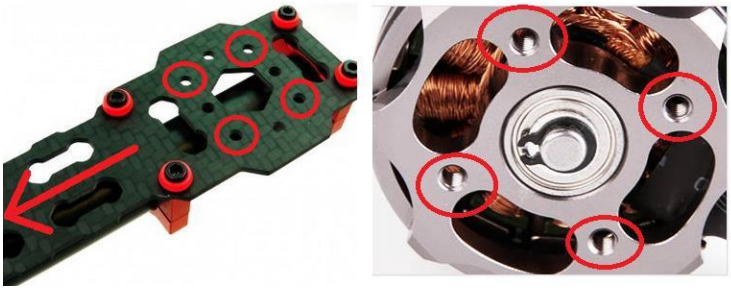


Figure 6.1.2B: MT3515 KV400 Mounting. Reprinted with permission from T-Motor. (Pending)

The propeller adapter was mounted to the top of the motor using three M3 propeller mounting screws as shown in Figure 6.1.2C below.

Figure 6.1.2C: MT3515 Propeller Adapter Mount. Reprinted with permission from T-Motor. (Pending)

Figure 4.1.2D: below shows the top view of the Tiger RC MT3515 KV400motor connected to the Hobbywing XRotor 40A ESC unit. After the motor and the ESC unit is secured to the frame, connect the bottom, middle, and top wire from the motor to the bottom, middle, and top female connector of the ESC respectively. Once the motor is connected to the ESC unit the excessive wires were secured to the frame using zip-ties.

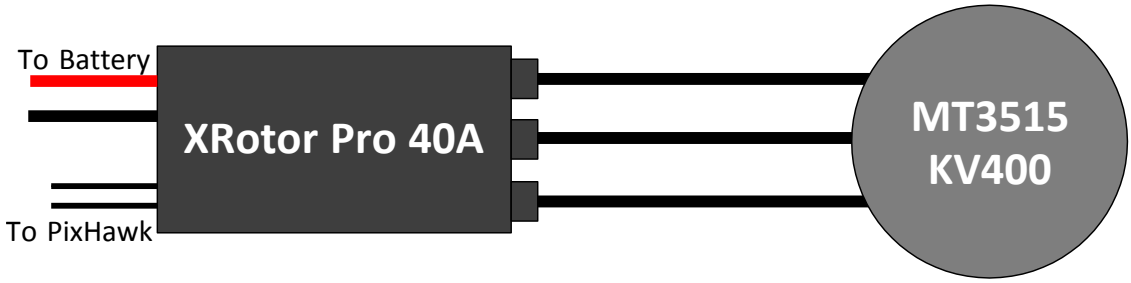


Figure 4.1.2D: MT3515 KV400 Wiring Diagram

6.1.3 Propeller

The IFLIGHT 15x5.5 T-Style Carbon Fiber propellers were used for the FXUAV system. They were purchased from gothelirc.com for $25 per pair. Each pair comes with a clockwise and counter-clockwise carbon fiber propeller, each weighing 25 grams. Four sets of propellers were bought, which allowed for 4 spare propellers in case of crashes that damage or break the propellers. The propellers were attached to the Tiger RC MT3515 KV400 motors as shown in Figure 6.1.2C. Figure 6.1.3A shows the propellers and their specifications.

Figure 6.1.3A: Propellers for the Drone

6.1.4 Power Source

For testing purposes, we've acquired multiple batteries. This helps in case one battery goes out and we do not want to wait on a part to order. Therefore we've acquired 4 batteries. One 10,000 mAh battery, two 6,000 mAh batteries, and one 5,000 mAh battery that was donated from a previous project. The idea is that the smaller batteries was used for initial testing so we do not burn up the more expensive one. However, there were some testing done with the larger battery to ensure our weight requirements and thrust requirements are still met.

6.1.5 Flight Control and Communication Modules

The Pixhawk flight controller that was used in the FXUAV was purchased from the 3DR online store. The Pixhawk flight controller was be mounted to the Tarot 650 frame of the FXUAV using the mounting foam that is provided from the manufacture. The package includes the following items: Buzzer, safety switch button, 3DR power module, 6 position connector cable, micro USB cable, SD card and reader, 3-wire servo cable, and I2C splitter module with cable. Figure 6.1.5A below shows the components that are included in the 3DR Pixhawk flight controller Kit.

Figure 6.1.5A: Pixhawk Components Reprinted with permission from Pixhawk.

Before all the components can be connected to the Pixhawk flight controller, it must be loaded with the ArduCopter firmware. Follow the following steps to load the firmware into the Pixhawk:

1. Download the Mission Planner software Version 1.3.37 to a computer.
2. Connect the buzzer to the buzzer port of the Pixhawk.
3. Connect the safety switch to the switch port of the Pixhawk.
4. Open the Mission Planner software on your computer.
5. Connect the Pixhawk to the computer using a micro USB to USB cable.
6. Choose the communication port that the USB is plugged into the computer.
7. Click on the initial setup option on the top menu bar of the Mission Planner software.
8. Click on the ArduCopter V3.3.3 Copter Quad firmware icon.
9. Follow the prompts that appears on the Mission Planner software.

After the initial setup has been completed and the Pixhawk has the right firmware installed, secure the Pixhawk to the center of the Tarot 650 frame using the double sided 3M anti-vibration foam. Once attached to the frame, connect the different components to its corresponding port on the Pixhawk flight controller. There should be a minimum amount of force required when connecting the cables to their corresponding ports. If the cable does not fit the port, it is because it is not facing the correct direction, flip the cable 180 degrees and try again. Follow the following steps to connect all the components to the board:

1. Connect the buzzer to the buzzer port.
2. Connect the safety switch to the switch port.
3. Connect the 6-position cable to the power port.
4. Connect the 3DR Telemetry Radio V2 to the Telemetry 1 port.
5. Connect the I2C splitter using the 4 wire cable to the I2C port.
6. Connect the 3DR uBlox GPS to the GPS port and the I2C splitter port.
7. Connect the RC receiver to the RC in port.
8. Connect the 4 XRotor PRO 40A ESC units signal inputs to the main output signals 1-4.

9. Connect the Beagle Bone to the I2C break
10. Insert the micro-SD card to the micro-SD card reader.

Figure 6.1.5B below shows the label ports on the Pixhawk flight controller according to the steps listed above.

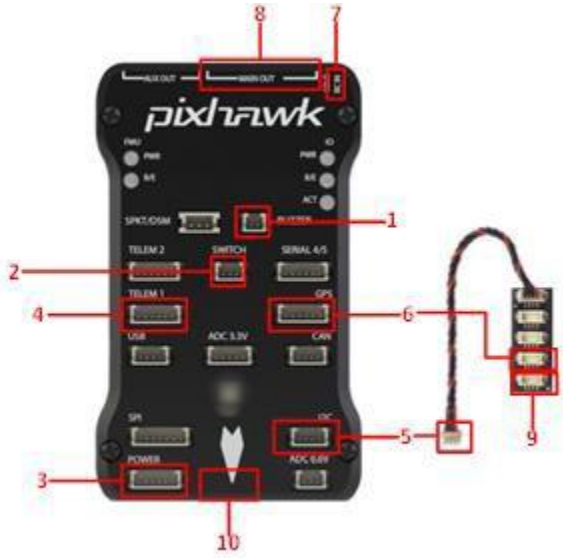


Figure 6.1.5B Pixhawk Connection Assembly Reprinted with permission from Pixhawk.

After connecting all the components to the Pixhawk, it is necessary to secure all of the components using the 3M double sided foam to the Tarot 650 frame accordingly. It is important to make sure the weight of the individual components is evenly distributed on the FXUAV so the center of gravity of the multirotor system is not compromised. Once all the components have been secured to the Tarot 650 frame, fasten any loose wires to the Tarot 650 frame of the FXUAV using zip-ties. Figure6.1.5C below shows the front view of the Tarot 650 frame with the components attached to it.

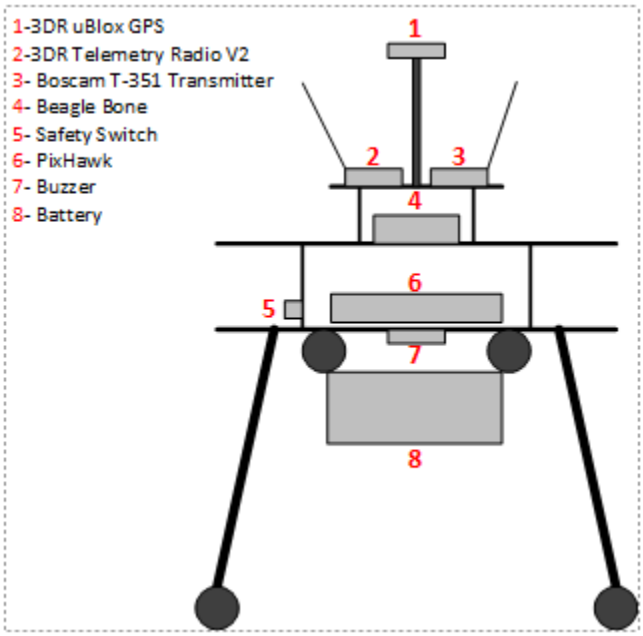


Figure 6.1.5C Tarot 650 Component Layout Front View

6.1.6 Sensors

The DJI iOSD Mini on-screen display was used for the FXUAV system. It was purchased from quadrocopter.com for $50. The iOSD Mini weighs 14g and has a PAL/NTSC port to communicate with the camera. The iOSD Mini overlays the camera feed with useful data such as voltage drain, current velocity, number of satellites being communicated with, distance to origin, direction to origin, and many others. Figure 6.1.6A shows an image of the on screen display module.

Fig. 6.1.6A DJI iOSD Mini

The 3DR uBlox GPS was used for the FXUAV system. It was purchased from 3dr.com for $89.99. The 3DR GPS weighs 16.8 grams, and provides GPS and Compass data at a rate of 5Hz. The 3DR GPS is also designed to work well with the Pixhawk flight controller and is shown in Figure 6.1.6B.

Fig. 6.1.6B 3DR uBlox GPS

The HC-SR04 Ultrasonic Module was used for the FXUAV system. Five ultrasonic sensors were purchased from gearbest.com for $2.50 each for a total of $12.50. Each HC-SR04 ultrasonic sensor weights 10 grams. The ultrasonic sensor sends out a sonar wave at 40 kHz, records the time taken for the wave to travel back to the sensor, and then calculates the distance to the object from that time. The output of the sensor was the sensor's high signal (5V for the HC-SR04) for a length of time correlated to the distance of the object. The maximum performance of the HC-SR04 ultrasonic sensor is within a 30 degree angle, so 5 sensors are chosen to maximize the coverage around the drone. A depiction of the ultrasonic sensor model is shown below in Figure 6.1.6C.

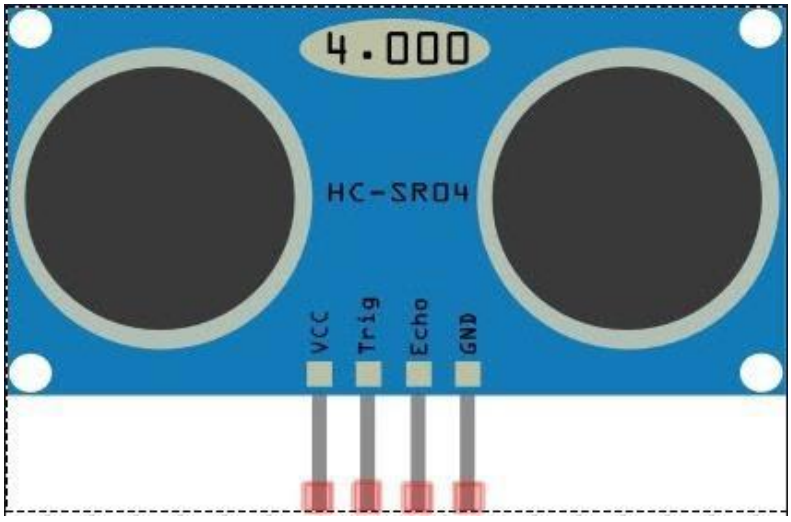


Fig. 6.1.6C: HC-SR04 Ultrasonic Module

Figure 6.1.6D illustrated the output signal of the ultrasonic sensor. The signal is initiated and waits for an echo back. The echo back was used to determine the distance of the object in question. That is how we used these sensors to detect how close the quadcopter is to an object in a given direction and it adjusted the flight path accordingly.

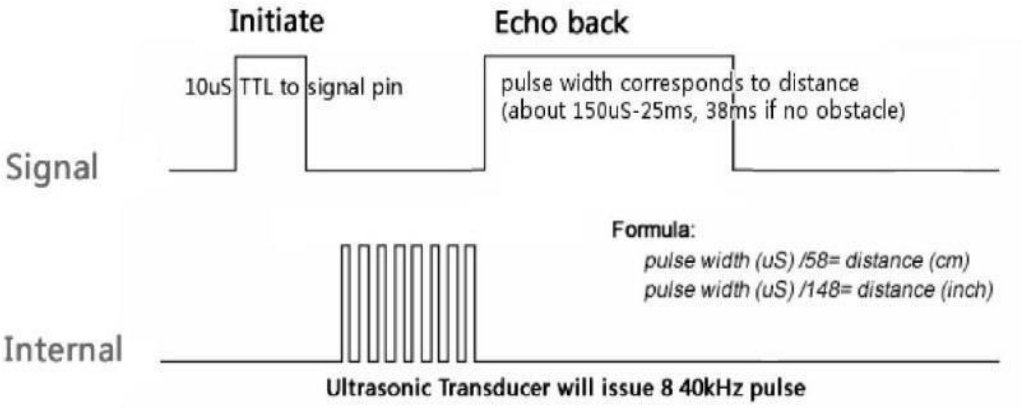


Fig. 6.1.6D HC-SR04 Output Signal (pending permission from Cytron Technologies)

6.1.7 Camera

The Logitech HD C270 camera was acquired from FLIR Systems, the company sponsoring the FXUAV. The Logitech camera therefore, did not require added cost to the projected FXUAV project estimate. It came with the desired lens size (refer to section 3.2.1.6 for lens size calculations and evaluations).

The Logitech camera was mounted directly underneath the frame of the quadcopter using the prefabricated mount on the Tarot 650 Iron Man. The prefabricated mount however was intentionally made for the GoPro camera and therefore difficulties arose upon installation of the Logitech HD C270 onto the quadcopter frame, several adhesive approaches were used. The camera was mounted at a 90-degree angle with the quadcopter frame to acquire a completely vertical field of view to the ground.

As mentioned above, difficulties arose upon assembly of the camera to the quadcopter's prefabricated mount and therefore a layer of plastic was secured to the mount and the Logitech was then secured using several strips of Velcro. The plastic panel secured between the Logitech camera and the quadcopter also helps account for any camera instability caused by the vibrations of the moving quadcopter. The Logitech camera attachment design can be seen in the Figure 6.1.7A below.

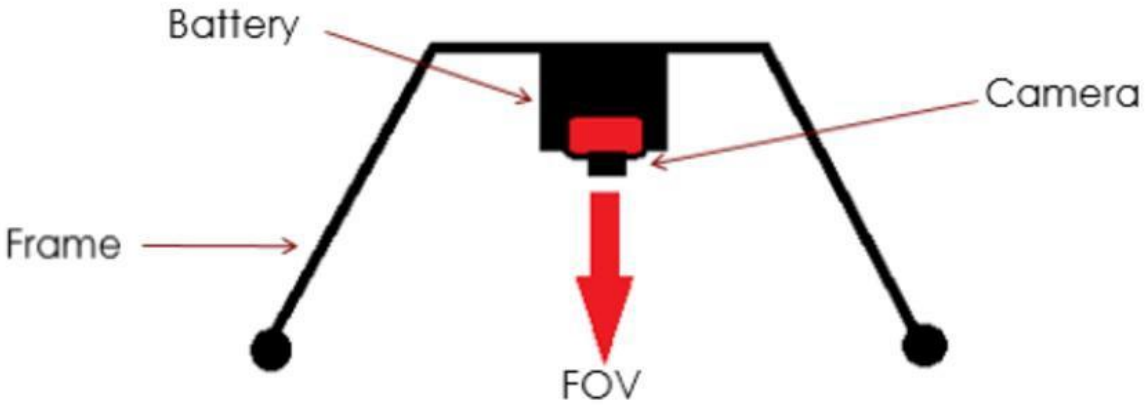


Figure 6.1.7A: Camera Attachment

6.2 Microcontroller and Flight Control

6.2.1 Flight Path Processing

The goal of the drone is to be able to autonomously make its way towards a fire, and use its payload to extinguish the fire. Once the camera recognizes pixels signifying a fire, and the processor confirms that the threshold is high enough, then the next step in the state machine is to move to the fire from wherever the drone is. Once a location is estimated, the drone needs to move towards that location. The method of moving an object from point A to point B is called pathfinding, and there are two popular algorithms for accomplishing the task. In both cases, the software needs to estimate a two-dimensional grid of nodes for the drone to move through. It can be estimated that one node is the size of a one-meter square. The drone then identifies through the GPS which node it currently resides on, which node it needs to move towards, and then follows and algorithm that moves along a path towards the destination node.

Dijkstra's pathfinding algorithm expands outwards from the origin location, looking at every single node along the way. Once the destination is found, all nodes are then evaluated and the absolute shortest path to the destination is found. Dijkstra is an impressive pathfinding algorithm because it always finds the shortest possible route to its destination. The main drawback of Dijkstra's algorithm is that it always searches all nodes around it until it reaches the destination node. For small graphs, this is not an issue. However the processing time it takes grows extremely quickly the further away the destination is, and as a result Dijkstra's can be slow compared to other algorithms. Figure 6.2.1A illustrates Dijkstra's algorithm in action and is shown below.

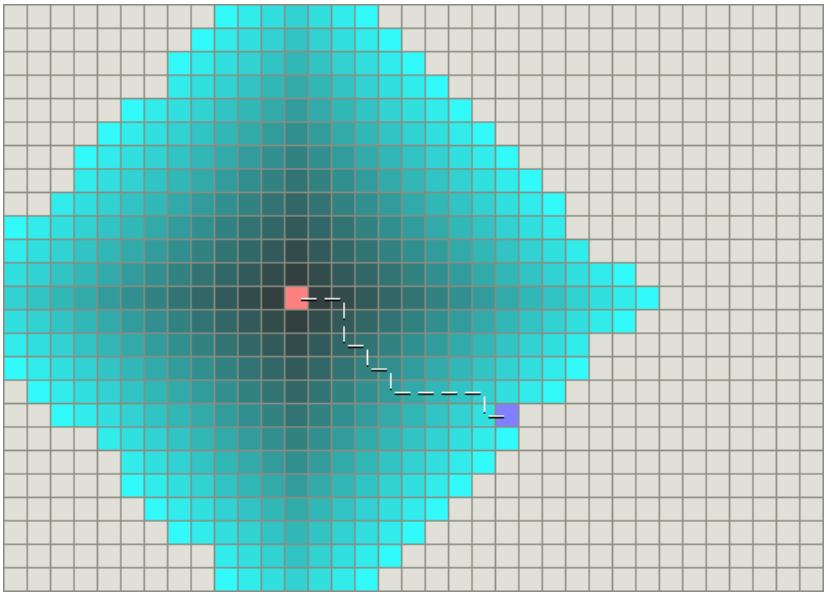


Figure 6.2.1A Pink square finds shortest distance to Blue Square with Dijkstra's (Permission pending)

By contrast, a Greedy Best-First-Search (BFS) algorithm only cares about nodes that put it closer to the destination node. A heuristic is used to determine the distance from the current node to the destination node, and then picks the adjacent node that puts it the closest. Instead of searching all nodes, it only searches the ones immediately adjacent to it at each iteration, allowing it to run much more quickly than the time-intensive Dijkstra's algorithm. Unfortunately, this can become an issue if there are obstacles in the way. Figure 6.2.1B illustrated best first search algorithm in action.

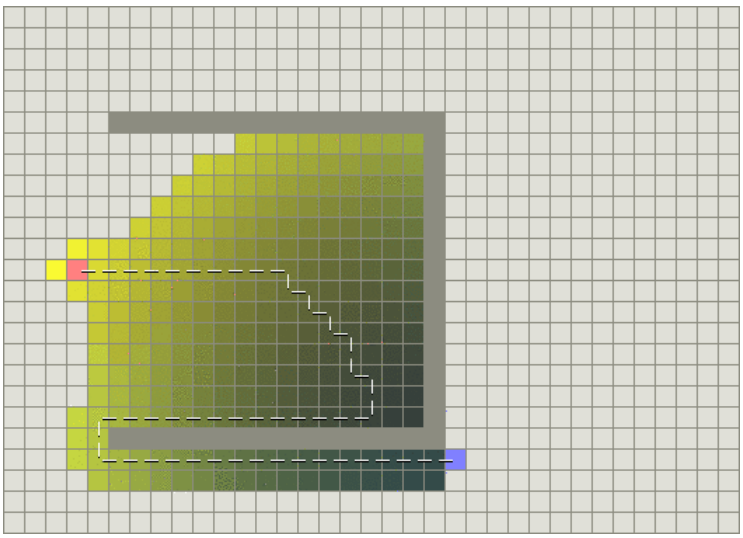


Figure 6.2.1B: Greedy Best-First-Search gets slowed down by obstacles (Permission Pending)

The A* algorithm is a combination of Dijkstra's and the Greedy Best-First-Search. It looks for both the shortest distance to the goal, and also uses the distance from the origin to dynamically search the nodes for an optimal path. Depending on the weights given to the two parameters of the algorithm, A* has the capability to be both fast, and effective, at searching for the proper path to the destination. The algorithm can also be customized to weigh one parameter more or less than the other, making it more or less like Dijkstra's. Figure 6.2.1C illustrates the A* algorithm in action.

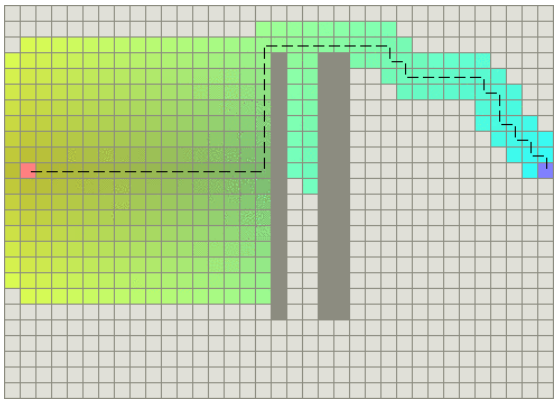


Fig. 6.2.1C The A* algorithm combining Dijkstra's and Greedy BFS (Permission Pending)

The A* algorithm is the ideal choice for a flight controller that updates its path frequently while searching for obstacles, and still wants to find the shortest route possible to minimize energy consumption for time spent flying. Thus the autonomous flight path of the FXUAV was controlled with the A* algorithm.

6.2.2 Sensor Processing

Collision avoidance for a small drone flying through the air can be difficult to achieve. The drone has to be able to not only sense objects around it, but also be able to either divert or completely stop movement in time. This requires that the drone not only be able to change its velocity and acceleration very quickly, but also have good sensors that can detect objects from far away. The solution for the FXUAV is to use five HC-SR04 ultrasonic sensors, one for every dimension of the drone, except for the top. The HC-SR04 sensors are able to detect up to 4m out, and they return a value correlating to how far away the object is.

Once an object is detected, a decision has to be made about what to do. After further testing, the team was able to decide what a safe stopping distance is, and form an equation that relates it to the speed the drone is traveling at. Once these numbers are known, a stopping threshold can be established. After establishing the stopping threshold, the next step is to determine what to do about objects that were colliding with the drone, when the drone is going too fast to stop in time.

First a state machine needed to be established, and correspond with three different levels of distance away from the FXUAV: a dangerous zone, a close

zone, and a safe zone. The safe zone would be any obstacle that does not meet any dangerous thresholds and can easily be avoided without stopping or slowing down. The close zone would indicate obstacles that are close enough to require slowing down or stopping. The dangerous zone indicates obstacles that are too close to slow down or stop, and must be quickly avoided. Figure 6.2.2A illustrates the three zones for UAV obstacle avoidance.

Fig. 6.2.2A the three zones of obstacle detection for the FXUAV (pending permission from Sergio Montenegro)

The state machine has conditional statements that check which zone the obstacle is in and react accordingly.

- State 0 is a state where collision avoidance is turned off, or there are no readings from the ultrasonic sensors.
- State 1 is when an obstacle is nearby but can be easily avoided without slowing down. The conditional statement checks for (objectDistance > (a + b)).
- State 2 is when an object has entered the close zone and requires slowing down or stopping in order to be avoided. The conditional statement checks for (a > objectDistance > (a + b)).
- State 3 is when an object has entered the danger zone and needs to be immediately avoided. The condition statement checks for (a < objectDistance).

Figure 6.2.2B shows the state machine for collision avoidance.

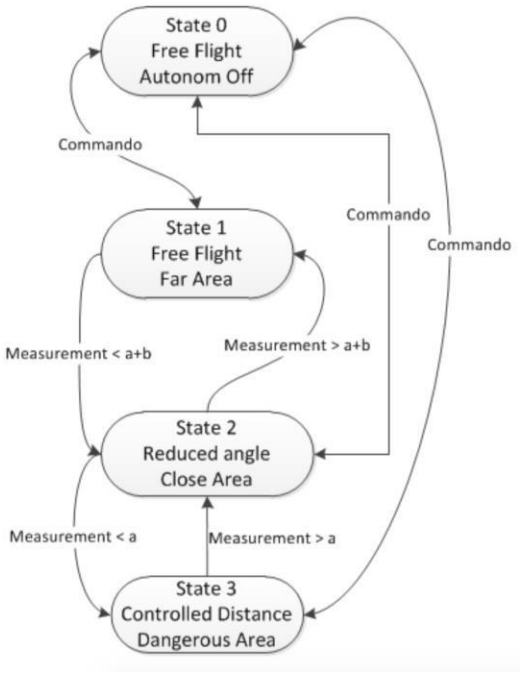


Figure 6.2.2B: State Machine for collision avoidance (pending permission from Sergio Montenegro)

In addition to using the ultrasonic sensors to aid with collision avoidance, the DJI iOSD Mini was used to obtain helpful flight path data that aided with testing, recording, and debugging.

- The number of satellites currently connected to helped troubleshoot the source of errors. If the drone malfunctions and crashes, the first step would be to check the logs and see if there was a sudden drop in communications. If not, then the next steps would be looked at.
- By using the distance from origin, the user was easily be able to set up maximum boundaries for the FXUAV to autonomously stay within, even there is a fire too far away. This decision can either be user preference, or determined after identifying the average flight speed combined with the length of time the drone is able to stay in the air. Then an out-and-back maximum distance can be calculated, keeping the drone from running out of battery while in air.

The current speed can be kept track of and logged, which is a variable that aided with many tests that are needed to further optimize the efficiency and functionality of the FXUAV.

6.3 Ground Control Station Development

6.3.1 Windows/Linux

The Ground Control Station is an essential component of the FXUAV system. It is the interface with which the firefighters will use and monitor the FXUAV while it is performing missions. The Ground Control Station is able to initiate, abort, and report progress on given firefighting missions. The Ground Control Station also is able to provide a livestream to the camera on the FXUAV, in addition to the overlay data provided by the image processing software. Finally, the Ground Control Station is able to provide telemetry and diagnostic data from the FXUAV at all points during the mission via command line, so the user is able to know whether to make adjustments or not, and to be able to react to potentially hazardous situations.

Development of the Ground Control Station also included writing programs in both C++ and Python. C++ was used for the image detection software, and Python was used for both flight path algorithms, and control of the fire suppression system. Development in Python was completed with both IntelliJ's PyCharm editor, in addition to development with VIM. On the following page is an example of source code from the flight path script. Development for the image recognition software was done using Microsoft's Visual Studio IDE (Integrated Development Environment). Execution of all of the code was done on the Raspberry Pi. The initial flight path Python script would be called from the command line, and it would then make system calls to trigger other scripts (such as servo control) from within the code. A Python-based API called DroneKit was used to write Python code which would then be converted into MAVLink packets, which would be sent to the

Pixhawk to control the drone's heading, velocity, destination, and many other parameters. The structure of a MAVLink packet is also found on the following page.

These are the system's requirements:

Ground Control Station Requirements

1. The Ground Control Station shall be able to initiate, modify, and abort firefighting missions.
2. Monitoring of mission progress shall be available at all times.
3. The livestream camera shall be available at all times.
4. The Ground Control Station shall communicate with FXUAV subsystems at all times.

```
        time.sleep(1)

#Arm and take of to altitude of 3 meters
arm_and_takeoff(2)

# Gets LocationGlobal at distance N&E of a given location
def get_location_metres(original_location, dNorth, dEast):

    earth_radius = 6378137.0 #Radius of "spherical" earth
    #Coordinate offsets in radians
    dLat = dNorth/earth_radius
    dLon = dEast/(earth_radius*math.cos(math.pi*original_location.lat/180))

    #New position in decimal degrees
    newlat = original_location.lat + (dLat * 180/math.pi)
    newlon = original_location.lon + (dLon * 180/math.pi)
    if type(original_location) is LocationGlobal:
        targetlocation=LocationGlobal(newlat, newlon,original_location.alt)
    elif type(original_location) is LocationGlobalRelative:
        targetlocation=LocationGlobalRelative(newlat, newlon,original_location.alt)
    else:
        raise Exception("Invalid Location object passed")

    return targetlocation;

# Get distance between two objects in meters
def get_distance_metres(aLocation1, aLocation2):
    dlat = aLocation2.lat - aLocation1.lat
    dlong = aLocation2.lon - aLocation1.lon
    return math.sqrt((dlat*dlat) + (dlong*dlong)) * 1.113195e5

# Moves vehicle to position dNorth&dEast meters N&E of current position
def goto(dNorth, dEast, gotoFunction=vehicle.simple_goto):
    currentLocation = vehicle.location.global_relative_frame
    targetLocation = get_location_metres(currentLocation, dNorth, dEast)
    targetDistance = get_distance_metres(currentLocation, targetLocation)
    gotoFunction(targetLocation)

    #print "DEBUG: targetLocation: %s" % targetLocation
    #print "DEBUG: targetLocation: %s" % targetDistance

    while vehicle.mode.name=="GUIDED": #Stop action if we are no longer in guided mode.
        #print "DEBUG: mode: %s" % vehicle.mode.name
        remainingDistance=get_distance_metres(vehicle.location.global_relative_frame, target
Location)
        print "Distance to target: ", remainingDistance
        if remainingDistance<=targetDistance*0.1: #Just below target, in case of undershoot.
            print "Reached target"
            break;
        time.sleep(2)
```

Figure 6.3.1A: Quadcopter Flight Path Code

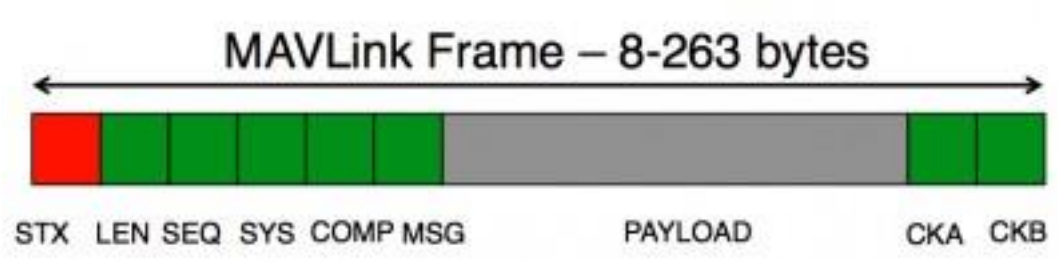


Figure 6.3.1B: MAVLink Packet Structure

6.3.1.1 perational Features

The following are Required Features that must be met for the Ground Control Station to be fully functional. Following are features that will greatly enhance the usability and efficiency of the project, but are not necessary for a fully functional Ground Control Station to interact with the FXUAV.

Required Features

1. System runs on a Laptop with the Windows 7 or OSX Operating Systems.
2. TCP/IP based wireless communication with the Ground Control Staton and the Raspberry Pi using 802.11 b/g/n Wi-Fi
3. USB or Serial communication between the desired camera and the Raspberry Pi
4. Override options for initiating and aborting missions
5. Text based status report of a mission in progress
6. Text based report of quadcopter diagnostics and telemetry

Optional Features

1. Application runs on *any* modern Operating System, with a modern Ubuntu virtual machine
2. Application runs on *any* modern Operating System, *without* a virtual machine
3. FXUAV firefighting mission live stream is viewable by multiple concurrent devices
4. Aesthetically pleasing and easy to use user interfaces designed in one application
5. Enhanced graphical display of mission status report
6. Enhanced graphical display of quadcopter diagnostics and telemetry

7 Prototype Testing

7.1 Hardware Testing

7.1.1 Environment

When testing hardware, ideal conditions are always a just that, ideal. For initial testing is smart to test indoors before we try to combat the elements that can contribute heavily to component failure. The following sections will describe our direction with testing and the environment in which we choose to test.

7.1.1.1 Quadcopter

The FXUAV is designed for outdoor missions, however at the start of the testing phase, the quadcopter was tested indoors. All initial testing is to be done within closed quarters to minimize outside factors, such as wind, rain, people, etc., in anticipation of maintainable flight stability. In addition, the quadcopter's initial flight tests consisted of restricted area coverage and height. This was accomplished by utilizing an originally designed and developed testing unit which can be seen below in Figure 7.1.1.1A Quadcopter Test Environment.

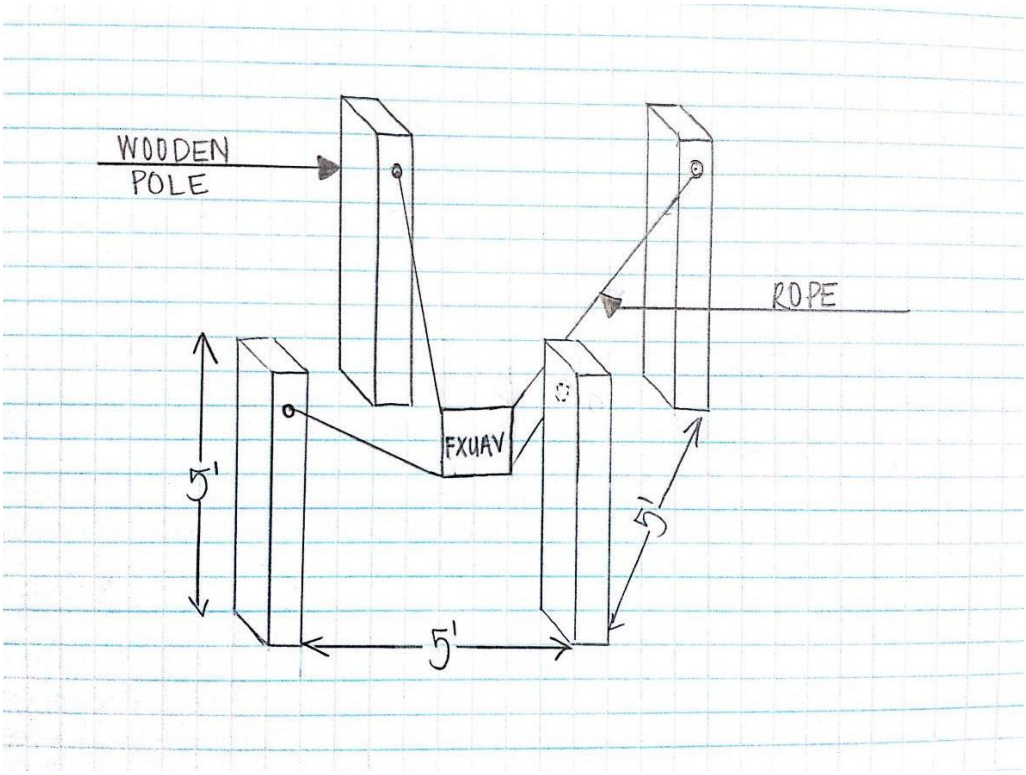


Figure 7.1.1.1A: Quadcopter Test Environment.

Once the drone has achieved stable flight maneuver and hover, additional testing was completed outdoors. The outdoor testing environment consisted of variable atmospheric conditions in order to minimize risk when it comes to releasing the deliverable for flight demonstration. These environments included hot days, cool days (weather permitting), sunny conditions and slightly windy conditions. The testing environment for the quadcopter does not include rainy climates or extreme

windy conditions due to the projects lack of sufficient funds appropriated to weather related damages resulting in the need of replacement parts.

7.1.1.2 Fire Suppression Release

The hardware test environment for the fire suppression release was tested indoors initially, in order to test the functionality of the release system, such that it fulfills all functional requirements listed in section 2.2.1.2 of Functional Requirements. In an attempt to capture the environmental scenarios for the simple initial testing, a fan was used to replicate the propeller wash from the quadcopter system. This gave an idea of how the fire suppression contents dispersed when released from the system. Each test was also conducted from a similar height at which the drone was when the contents of the system was released to simulate a real test once all systems are integrated together. Once proven indoors, the system was tested in a similar fashion outdoors, including the simulated propeller wash from the fan and the height from which the payload was released. All of these conditions are necessary in order to capture realistic behavior of the system once everything is integrated.

Once the servo was proven to work, we wrote a testing script where we placed the fire directly under the quadcopter and just loitered and searched for the fire. This differs from the searching algorithm. We ran this test script just to test that our scripts can run concurrently, and that the quadcopter would still fly and activate the servo. This helped by reducing the complexity of the testing.

7.1.2 Test Cases

7.1.2.1 Quadcopter

Once all of the equipment are installed and configured according to their manuals, the initial testing of the FXUAV flight operations was tested. The first test of the FXUAV flight stability, which determined any further calibrations are necessary. The second test was of the total flight time with the maximum payload attached. The third test was of the maximum range between the communication interfaces of the FXUAV. The fourth and final test was of the autopilot system through the Mission Planner software.

Flight Stability Test

The flight stability test was first conducted in a controlled environment, the best location of this test was in an indoor location where the weather environment was not a factor. The FXUAV was secured to four poles that are 5 feet tall that are 5 feet apart from each other using 4 feet long bungee cords. This set up limited the vertical and horizontal movement in case the FXUAV loses stability. Once the FXUAV had properly demonstrated stable flight while secured to the poles, the testing was moved to an outside location where the stability of the flight system was tested in different limited weather conditions. During the outside testing, the FXUAV was limited to a flight no higher than 10 feet, and a horizontal distance no

longer than 20 feet. Once the FXUAV passed the flight stability tests, the payload and flight time testing begun.

Flight Time Testing

The purpose of the total flight time test, is to calculate the total possible flight time of the FXUAV with the maximum payload attached to it. This test was done indoors using the same procedures as the flight stability test to secure the FXUAV. The FXUAV was weighted using a calibrated digital scale to ensure the maximum payload is attached to the system. Once the right payload is attached to it, the FXUAV was set to a stationary hover for at least 10 minutes as required. If it passes the 10 minutes of flight time, it was tested again with one more minute added to the flight time. This procedure was repeated until the low battery warning signals are activated. After the indoor flight time testing is completed, we repeated the same testing procedure in an outside area where the FXUAV was affected by the environment conditions. After both testing procedures are completed, the total flight time was the lowest time that FXUAV was able to sustain flight.

Results:

After testing the quadcopter flight time during a sample of over 20 test runs, it was noted that the quadcopter could sustain over 15 minutes of flight time using the 10,000mAh LiPo with maximum payload attached to it.

Wireless Communication Testing

The wireless testing was conducted to analyze the greatest distance the FXUAV can fly from the ground station without critical signal interruption. Out of the three wireless systems in the FXUAV, only the telemetry signal and RC signal are critical wireless system to operate the FXUAV. Since the video transmission system is only providing video to the ground station operator, a certain loss of signal was tolerated as long as any other function of the FXUAV is not compromised. The first test was conducted in an outside field where there was a clear line of site between the FXUAV and the ground station. The first clear line of site testing was conducted by a group member holding the multirotor. For safety precautions, the propellers were detached from the motors in the first part of this testing. The Mission planner software shows a signal strength in the mission screen for the RC signal, which was the first metric noted. The telemetry data is also displayed on the Mission Planner screen, and the signal quality was acceptable as long as the data is showing. A person held the FXUAV 100 feet away from the ground station and proceed to move 10 feet away with every acceptable measurement. Once either the telemetry signal or the RC signal is lost, the previous measurement taken was considered the acceptable length.

The second wireless communication test was conducted when the FXUAV does not have a clear line of site to the ground station. The same steps taken during the

clear line of site was taken to measure the greatest distance the first critical wireless system is lost. Once this value is noted, the third wireless communication system testing was conducted.

The third wireless communication system was conducted while the FXUAV is flying while being controlled by the Radio Link AT10 radio control transmitter. For this test, the team tried to acquire a second Radio Link radio control transmitter that is paired to the same frequency as the primary radio control system. This secondary unit was used to take over the flight of the FXUAV if the radio control signal is completely lost, and the FXUAV becomes unresponsive. The FXUAV flew at a height of 50 feet from the ground, and stared its flight 100 feet away from the ground station. Then it proceeded to move forward at a speed no greater than 5 mph, once any of the two critical systems signal is lost the distance was annotated. This flight test was first done with a clear line of site from the ground station, and the second test was done without a clear line of site.

The final wireless signal range was the shortest distance measured out of all of the testing conducted of the two critical flight communication system. This assured that at no point the FXUAV lost communication with the ground control during regular operations. During all of the testing procedure, the distance of the video signal was also annotated. This helped the operator of the ground station know what to expect during normal operations, and not ground the FXUAV just because of video signal transmission has been lost.

Results: After all the wireless communication tests is was noted that the telemetry system was able to reach the required range of over 500 square feet as necessary. Unfortunately, the radio control system and the wireless video system did not respond the same. Group 7 lost contact with the first quadcopter built due to loss of Radio control communication at about a 300 feet range and had to rebuild the quadcopter since it was not able to located the lost drone. After rebuilding the drone, the flight was limited to a 200 feet range so no loss of communication would happen again. Also during our final testing, the quacopter lost contact with the Wifi module in the Raspberry Pi at a range of about 200 feet, and due to time constraint further testing were limited to 100 feet.

Autopilot system testing.

The final test of the wireless communication system of the FXUAV was conducted while the unit is being fully controlled by the autopilot system. This test helped measure the full capabilities of the autopilot system of the FXUAV without an operator present. The first autopilot system test was conducted on an open field, where the behavior of the FXUAV can be observed at all times. For this test, the autopilot had way points set through the Mission Planner autopilot software. Group 7 put ground markers where the way points of the flight plan has been set, this helped determine if the FXUAV is reaching the way point assigned in the flight plan. After the autopilot has been tested in an open field, the testing was then moved to an area where there was no clear line of site between the ground unit

and the FXUAV. For this part of the testing, the group members were located where they were able to verify that the FXUAV has successfully reached the way points set in the flight plan. For both of parts of the autopilot testing, the radius of the FXUAV was expanded with each successful testing done. The autopilot system was determined acceptable once a reasonable range of at least 500 feet in radius has been completed as per mission requirements.

Results: Since the telemetry system passed all the range testing as required, the Pixhawk also passed the required range of communication while in autopilot mode. Unfortunately, due to design changes the total range of the system was reduced to no more than 100 feet so it would not lose communication with the Wi-Fi module.

7.1.2.2 Fire Suppression Release

The fire suppression release system's hardware needed to be tested before the entire system is integrated into the quadcopter system. That is, the servo's functionality alone will need to be verified, as well as the entire system once all the pieces are assembled and the system is ready for integration. The procedures for each test case are listed below.

Procedure A (Servo)
Purpose: verify the hardware of the servo is free of defects.
Materials: Servo
PCB
Laptop

1. Connect the servo to a Raspberry Pi using the diagram shown in figure 7.1.2.2A

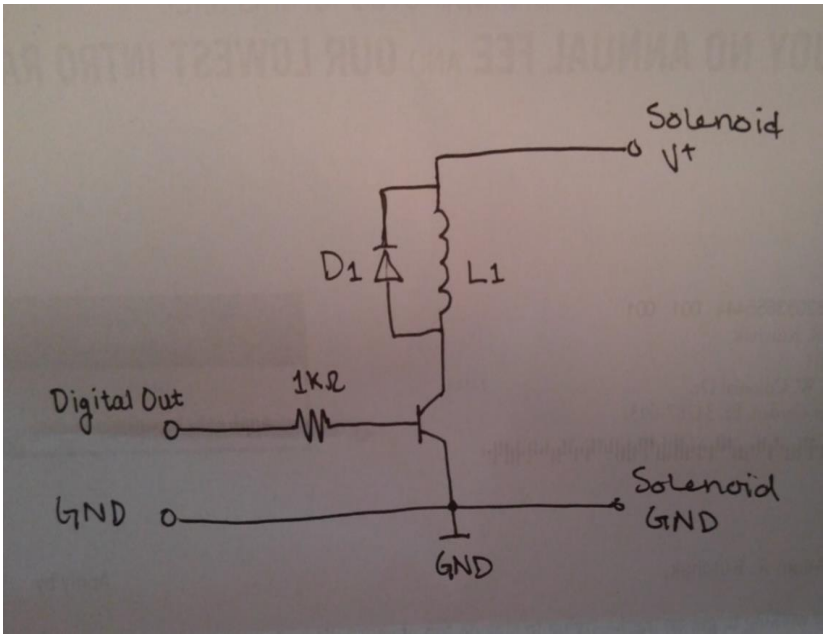


Figure 7.1.2.2A: Schematic for connecting a simple servo

2. Connect the servo to a larger power supply (9V-24V).

3. The connections should look like the figure shown in Figure 7.1.2.2B

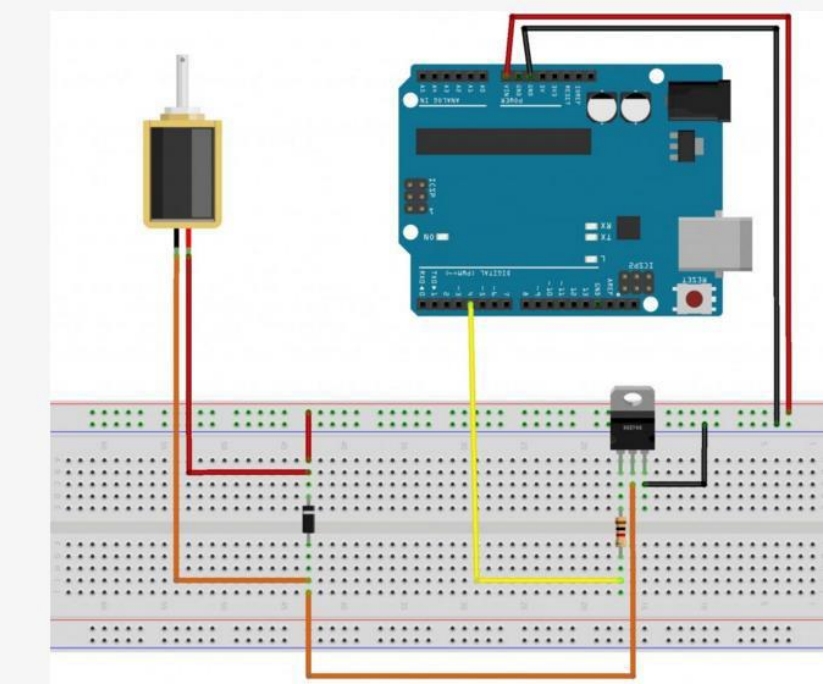


Figure 7.1.2.2B: Connection diagram for simple servo (made using Fritzing)

4. Run test code to test the servo. Set the servo pin high and low in a loop.
5. Verify the servo retracts/extends as necessary according to the code.

Procedure B
Purpose: to verify the release system will work now. The steps for verification are shown below.
Materials: Servo
PCB
Assembled systems including container, contents etc.

1. Once the release system has been assembled, it is now time to test the door will open upon activation.
2. Run similar test code to procedure A, just not in a loop.
3. When the servo pin is set to high, verify that the release door is opened.
4. Repeat steps 2 and 3 with contents inside the container, and verify the contents are dispersed.
5. Verify all contents of the container have been emptied once the release door has been activated.

7.2 Software Testing

7.2.1 Environment

7.2.1.1 Quadcopter

While the quadcopter is being tested initially to iron out flight stabilization and everything necessary for it to fly autonomously, all tests was conducted indoors.

Testing indoors can help supply cooler air for the electrical components to operate efficiently. In addition, while the flight software is being tested, each axis of the quadcopter was affixed to a non-moving element with substantial slack to allow of hovering. This helps with multiple aspects of software flight testing. That is, being affixed on all axes limits the quadcopter's range in all directions so that it cannot fly away and damage itself or anything else. However, the slack allows the quadcopter some freedom in all directions in order to give it real automation and let the controller and software actually control the quadcopter. If one of the attachments is taught while others have slack, this indicates that the quadcopter flight code isn't exactly where we want it. Therefore, attaching all axes with some slack benefited software testing in several different ways.

Once to quadcopter flight code is stable, we must test the camera's abilities indoors to detect thermal energy. Due to the nature of thermal detection, the testing must be conducted away from any shiny metal surfaces that reflect the thermal energy. This could cause major problems in the detection of the flame, therefore we are eliminating that threat from the problem all together. Testing indoors initially provided a good base to start since the environment was relatively dark when compared to sunlight which helps with thermal imaging as well. The darker the better. Once tested inside, the camera was then tested outside in order to simulate realistic conditions.

After all systems have been tested independently, the camera and quadcopter was integrated together, and tested indoors initially. The reasons are stated previously. In short, initial testing indoors, allows the tester to control the environment and determine if the system is ready to be tested outdoors. Once outdoors, the quadcopter can essentially be considered free and is no longer in our control. This environment of testing can prove detrimental to the project and is why we choose to test outdoors last after we believe all kinks have been worked out and are confident enough to let the quadcopter have full reign of itself.

7.2.1.2 Android Application

Android Studio 2.1 is based off of the powerful IntelliJ IDEA IDE. The IDE easily structures source code and resource files into standardized, easy to understand modules that allow for a great amount of user customization. Android Studio uses the Gradle build automation system to turn the developer's project into a packaged, usable application. Gradle will compile, package, and automatically test the application once built. Android Studio also has several other useful development features such as inline debugging, a memory and CPU monitor while the app is running, viewing of the Java heap dump, a memory allocation tracker, and detailed code inspections supplementing IntelliJ IDEA's already-existing feature. Finally, Android Studio has built-in version control integration with version control systems such as Git, Mercurial, Subversion, and others. Shown in Figure 7.2.1.2A is Android Studio's memory monitor.

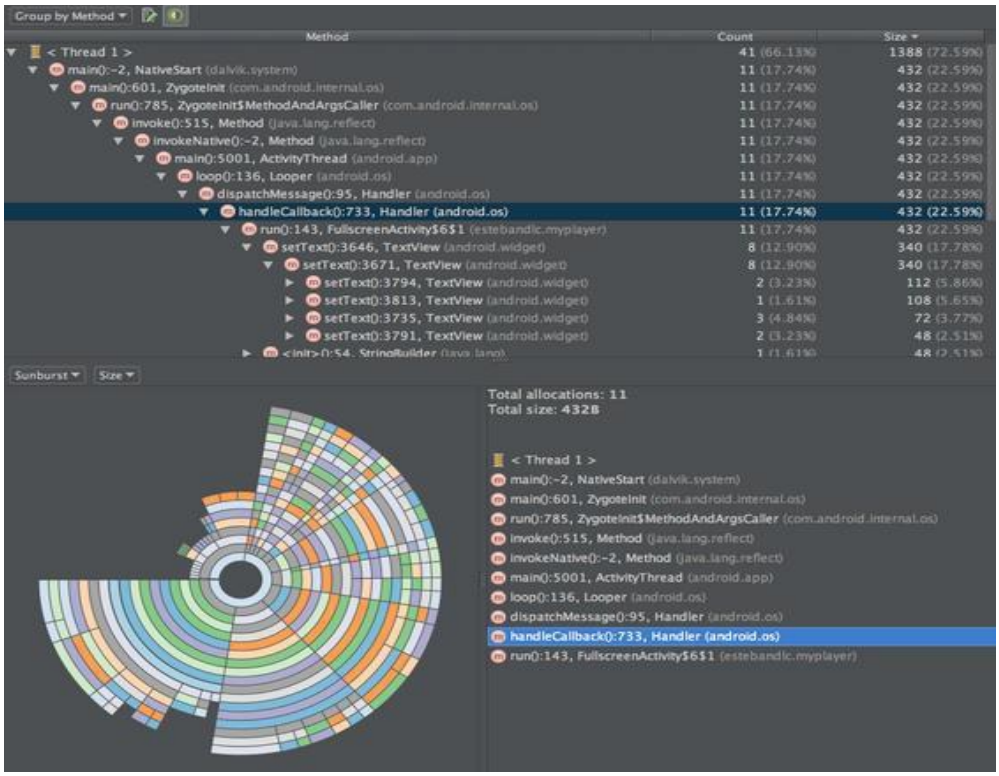


Fig. 7.2.1.2A: Android Studio's Memory Monitor

The current version of the Android SDK is Revision 25.0.0. The Android SDK contains Application Program Interfaces (APIs) for all of the versions of the Android Operating System; these range from Android 1.0 to Android 6.0 (Marshmallow). The Android development documentation lists each major version of the Android OS and correlates it to the percentage of Android owners that are able to run the specified Operating System. The documentation suggests setting a minimum SDK requirement of API 15, correlating to Android 4.0.3 (IcecreamSandwich), which will be usable by 94.8% of all Android owners. The team's smartphones are able to run this operating system, and the target development/testing environment (Samsung Galaxy S4), was released with Android 4.2.2 and can be updated to Android 5.0.1. In addition, by choosing the Android Operating System with the minimum SDK of Android 4.0.3, the team maximizes the possibility of the application being usable by firefighting services. With approximately 70% of Americans owning smartphones, and with Android phones making up >50% of the smartphone market, combined with the 94.3% chance of Android users being able to run the application, there is a very low possibility of the application being unusable to the customers. Figure 7.2.1.2B shows the Android API usage for the different operating system versions of Android.

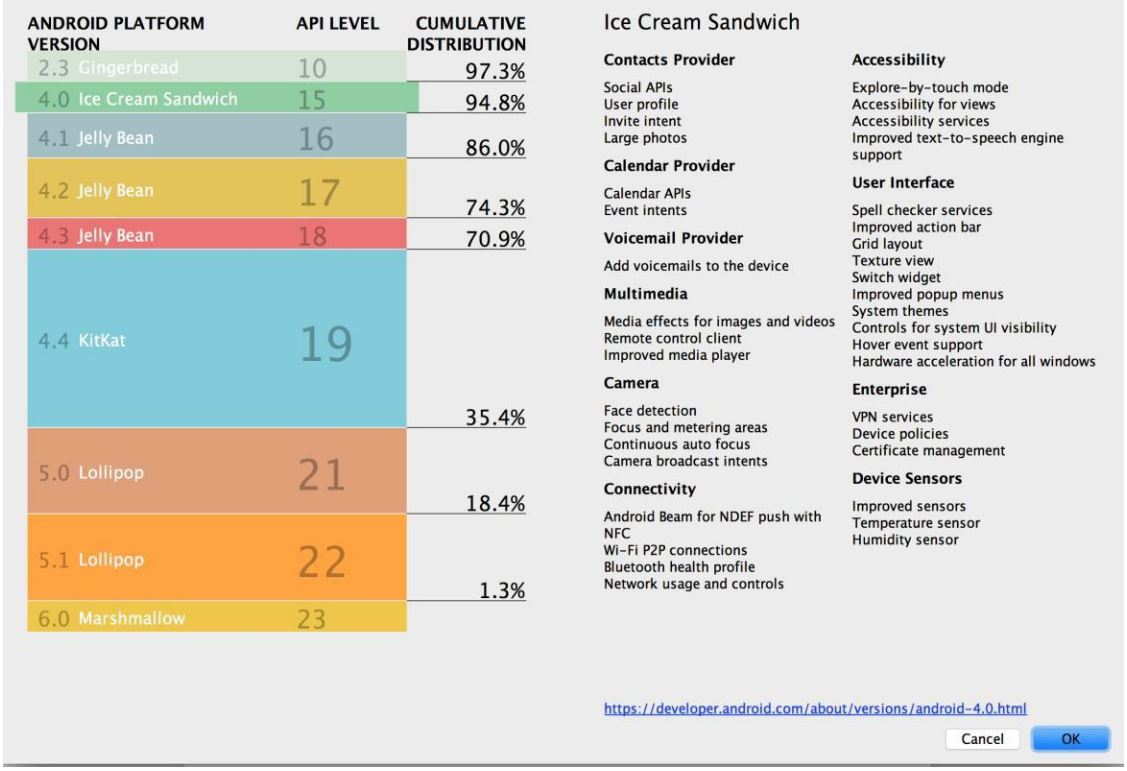


Fig. 7.2.1.2B Android API Usage

Android applications are also built using the Java programming language. This was also a factor in choosing an environment in which to build the mobile application. The team had to deal with the tradeoff of picking a language familiar to all members, in order to reduce development time and spend more time solving other problems, or lose hours for everyone to learn a new technology. The decision was ultimately made to go with Android. All of the members of the team that are Computer Engineering majors have extensive use with Java, which is the main programming language used to develop Android applications. This will allow more time from each of the members to spend working on other aspects of the drone's completion due to the time saved from the potential learning curve.

Managing the project will be a major challenge, which is why it will be important to choose a Version Control System (VCS) that is robust and easy to use. Thankfully Android Studio has built-in functionality to work with all of the major current VCSs such as Git, Mercurial, SVN, and others. The team members have experience with all three VCSs, and in the end it was decided to host the project on a private Github repository while being managed with Git, due to the ease in setting up and maintaining a Github repository.

7.2.1.3 Fire Suppression System

The testing environment for the fire suppression release system was initially conducted indoors just as the hardware test was. This again is largely due to the fact that we can control the indoor environment to cater to specific behaviors. For

example, overheating issues can be remedied with proper cooling. This is something more easily provided indoors rather than outdoors. The software required many changes, and revisits, much like a trial and error type of strategy. When the software didn't behave how we intended, we revisited the code and tried to analyze the behavior. We made some changes, and retested. The process was repeated until the software and hardware behaved as intended. Until then, all testing was conducted inside. It was much more convenient to test software, and edit, and reload firmware into a device while inside, versus constantly having to reconnect a device to a laptop and edit the code and reload it onto the device. Instead, leaving it plugged in, indoors saved time, and sweat. The fire suppression software was tested in two different cases. Did the servo work as intended? Does the entire system work as intended. As stated previously, each system will be tested in an indoor environment where conditions can be controlled. Once proven worthy, each component will be tested in a similar fashion outdoors where conditions cannot be controlled as easily.

7.2.2 Test Cases

7.2.2.1 Quadcopter

In order to fully test the functionality of the quadcopter itself, its components were put through a series of test cases. These test cases provided the details we needed to know in order to determine whether our system is performing to our standards. Motors need to be working and free of defects. The Raspberry Pi single board computer must be capable of running a simple program. The batteries must be free of defects and supply the power as advertised. The ultrasonic sensors must be free of defects and operate up to the advertised distance.

Procedure A:

Purpose: To test each motor and verify its working, and doesn't have a bad connection or a short

Supplies: Motor
Multimeter
ESC
Power Supply

Steps

1. Verify the resistance across all combinations of the three wires is relatively equal. Otherwise, a short may occur.
2. Attach motor to ESC
3. Power on the motor
4. Measure the resistance of the connectors. High resistance will lead to heat which will cause failure.
5. Verify that the motor is spinning once received power.

Results:

After connecting all the motors according to the design specifications, the motors showed proper operation. Unfortunately, we were not able to test the resistance of the motors using a multimeter. After further analyzing multiple quadcopter flights no motor failure occurred, and the quadcopter was able to fly smoothly with the maximum payload attached to the system.

Procedure B:
Purpose: To test the functionality of the onboard microcontroller and verify that the software is working correctly.
Supplies: Raspberry Pi
SD Card with Ubuntu Flasher
Laptop
USB Cable
Steps

1. Insert SD card into Raspberry Pi slot.
2. Hold down the boot button while plugging the Raspberry Pi into the laptop.
3. Let Linux install onto the Raspberry Pi.
4. Verify no LED's are flashing.
5. Login to Ubuntu
6. Run a test program to test the software is running correctly and is capable of doing so.

Procedure C:
Purpose: To test if the batteries are functional and not defective
Supplies: Battery
Multimeter
Steps

1. Connect positive lead of multimeter to positive terminal of battery.
2. Connect negative lead of multimeter to negative terminal of battery.
3. Verify the voltage output is nearly 22.2V for all batteries.
4. Repeat steps 1-3 for all batteries.

Results:
After testing all batteries that were purchased for the project, we concluded that all batteries were fully operational. At no point during testing there was loss of power to the drone due to its power source.

Procedure D:
Purpose: Verify the ultrasonic sensors are working and not defective
Supplies: Ultrasonic sensors
PCB
Laptop
Steps

1. Connect the ultrasonic sensors to the PCB.
2. Power the PCB by connecting it to the laptop

3. Assign output pins and run a test code and verify the ultrasonic sensor is working and is accurate as advertised.
4. Repeat steps 1-3 for all sensors used in the project.

7.2.2.2 Camera and Image Processing

This section encompasses the distinct tests cases that shall be completed on the FXUAV image processing module. Included in this section, is the expected result from each test case and constitutes as passing standards.

Image Processing Object Detection Testing

- Purpose: To verify the image processing module functionality once it has been connected to the Raspberry Pi

- Materials
  - ➢ Raspberry Pi
  - ➢ Transmitter- BosCam 5.8 GHz RC305
  - ➢ Receiver- BosCam 5.8 GHz RC305
  - ➢ Logitech HD C270 Camera
  - ➢ 10000MAH 25C 6S Battery

- Test Procedure 1: Create a set of classifiers and while indoors, test OpenCV's application for object detection on the target.

- Expected Results: The system should return a false positive rate of no more than 1.5% and the returned detection rate should therefore be no less than 98.5%.

- Test Procedure 2: Create a second set of classifiers and while outdoors, test OpenCV's application for object detection on the target.

- Expected Results: The system should return a false positive rate of no more than 1.5% and the returned detection rate should therefore be no less than 98.5%.

- Test Procedure 3: Create a third set of classifiers and while outdoors, test OpenCV's application for object detection on the target while both the C270 as well as the Raspberry Pi are properly mounted onto the quadcopter and the quadcopter is hovering over the object at an altitude of about 10ft.

- Expected Results: The system should return a false positive rate of no more than 1.5% and the returned detection rate should therefore be no less than 98.5%.

- Test Procedure 4: Using the same set of classifiers as used in Test Procedure 3 and while outdoors, test OpenCV's application for heading interrupt while the Pixhawk is properly mounted onto the quadcopter,

- Expected Results: The system should return a false positive rate of no more than 1.5% and the returned detection rate should therefore be no less than 98.5% and the quadcopter should change course to reposition itself over the fire once it is detected.

- Conditional Requirements: The above test cases all require subsets of positive and negative imagery in relation to the target object. Section **4.1.1.2**: Image Processing, demonstrates how to obtain these positive and negative images and how these raw images are transformed into sample data sets for classifier training. For data accuracy and detection, at least 1500 positive image samples and 1500 negative image samples should be obtained.

Image Processing Module Integration Testing

- Purpose: The test and verify the image processing unit function after integrations with the Pixhawk flight controller, to verify waypoint interruption and redirect.

- Materials:
  - Raspberry Pi
  - Transmitter- BosCam 5.8 GHz RC305
  - Receiver- BosCam 5.8 GHz RC305
  - Logitech HD C270 Camera
  - 10000MAH 25C 6S Battery
  - Pixhawk Flight Controller

- Test Procedure 1: This test includes the integration of the quadcopter's flight controller to the image processing module. The quadcopter should commence its course from a base location outdoors. It shall travel to a specified waypoint determined by the Mission Planner software. On route to the waypoint, the system should pass over a fire.

Expected Results:

Once the image processing module detects the fire, the FXUAV copter should halt its current course and hover over the targeted object. It should then adjust its position to approximately 2 feet above the fire and release the payload.

7.2.2.3 Flight Control Processing

Flight Control Test

- Purpose: To verify the flight controller works, and that all of the internal sensors were calibrated successfully and can execute flight.
- Materials:
  - Pixhawk
  - Assembled quadcopter
  - RC
  - Mission planner
- Test Procedure 1:
  - Give the quadcopter a simple test plan. Use takeoff to 5 meters, loiter for 15 seconds and fly to an arbitrary waypoint.
  - Then set the quad copter to return to launch and land.
- Expected Results: The drone should take off to an altitude of 5 meters, and hover for 15 seconds. It should then take off to the designated waypoint and return to where the mission started, then land itself. Ensure the drone behaves properly, without any excessive swaying, dipping, or rising, as this may indicate the compass, GPS, or accelerometer have not been calibrated correctly.

Pixhawk Serial Communications

- Purpose: To verify that the Pixhawk can successfully receive and transmit waypoint data without loss of data or corrupted bits.

- Materials:
  - ➢ Raspberry Pi
  - ➢ Logitech HD C270 Camera
  - ➢ 10000MAH 25C 6S Battery
  - ➢ Pixhawk Flight Controller

- Test Procedure 1: This test involves the integration of the Pixhawk flight controller with the Raspberry Pi. The FXUAV copter should commence its course from a base location outdoors. The mission planner should specify a waypoint of which the copter must navigate to. The copter must fly over a fire.
- Expected Results: The Raspberry Pi should successfully send the data it receives from the image processing module, after the fire has been detected, to the flight controller. This should halt the FXUAV copter to remain hovering over the detected object. It is imperative that the Raspberry Pi relay this information with 100% accuracy to the Pixhawk flight controller in order to ensure that the copter is positioned directly over the detected fire.

- Test Procedure 2: This test involves the Pixhawk integration with the Raspberry Pi. Have the Pixhawk send GPS location data, i.e. coordinates, to the Raspberry Pi.

- Expected Results: The Raspberry Pi should receive the GPS Coordinates of the FXUAV copter from the Pixhawk flight controller with 100% accuracy.

Mission Planner Integration Test

- Purpose: To verify communication between the Pixhawk Flight Controller and the Mission Planner software running from the ground communication station.

- Materials:
  - ➢ Raspberry Pi
  - ➢ Logitech HD C270 Camera
  - ➢ 10000MAH 25C 6S Battery
  - ➢ Pixhawk Flight Controller

- Test Procedure 1: The Mission Planner software should be configured with a user defined GPS coordinate Map with a series of waypoints set at each desired coordinate. The FXUAV should then be launched from a base location outdoors.

Expected results: The FXUAV quadcopter should autonomously navigate according to the waypoint map that was configured in the Mission Planner software. The FXUAV should remain in Auto mode while doing so.

Mission Script Integration Test

- Purpose: To test the final mission script, the fully autonomous software that will operate the drone and maneuver it without interference from outside sources. This will test our autopilot searching software
- Materials:
  - Raspberry Pi
  - Pixhawk Controller
  - Assembled Drone
- Procedure:
  - Load the mission scripts onto the Raspberry Pi and mount the Pi onboard the drone.

    - SSH into the Raspberry Pi via another computer.
    - Execute the scripts and observe the drone's behavior.
    - Be sure to have a failsafe mode programmed into the RC in case of emergency.
- Expected Results: The results of this should be that once the scripts are run, the quad copter should arm itself, take off and start moving east, then south, then west, then north. It should repeat this circular pattern will a minor decrease in diameter with every revolution. This would ensure that we are taking an outside in approach to locate our target. Once we complete the search and reach the very middle of the search grid, the drone should land.

Results: After testing the original design of the quadcopter, we did not have success using the BeagleBone Black and the Tau 2 camera. The Beagle Bone Black was not fast enough to process all the data necessary so we decided to make a design change. Group 7 first replaced the BagleBone with an Odroid XU4 system, which proved to be much faster in processing all the necessary data. Unfortunately, the Odroid had issues communicating with the Pixhawk Flight controller, and did not have a PWM output for the servo motor. So Group 7 then replaced the Odroid with a Raspberry Pi. After further testing, the Raspberry Pi showed sufficient processing speed to process all the required data while communicating with the Pixhawk successfully. Group 7 also had to substitute the Tau 2 camera with a Logitech web camera since the supplier was no able to get the Tau 2 in an adequately timely manner.

7.2.2.4 Additional Sensor Processing

The additional sensors that we're using to test for object avoidance are extremely important. These tests will help to let us know that our drone can effectively maneuver its own way without colliding with any object. This will help protect itself and whatever it may collide with, people or fragile objects. Below are the following procedures that are used to test our ultrasonic sensors to ensure a safe flight.

Procedure A:
Purpose: to determine whether or not the sensors work.
Materials: Ultrasonic Sensor
PCB (Arduino or any other)
Laptop
Steps:

1. Connect the appropriate pins of the sensor to the Arduino
2. Write a simple test code that triggers the ultrasonic sensor
3. Take measurements with the sensor and double check them with a ruler
4. Determine the sensors maximum and minimum working distances

Conclusion and Results
The expected results are that the sensor works and you've found approximately what distances the sensor no longer works. This will help with object avoidance

Procedure B:
Purpose: To verify if the quadcopter will indeed respond and avoid collisions
Materials: Drone (assembled with sensors)
Laptop
Steps:

1. Once the quadcopter has been assembled and the sensor(s) are in place, write a test code that adjusts the quadcopters flight path once reaching within 50 cm of an object
2. In a controlled environment with the quadcopter constrained, introduce an object within the threshold and verify the quadcopter reacts accordingly
3. Repeat steps 1 and 2 until all sensors have been accounted for in all directions.

Conclusion and Results:
The expected results are that once and object is introduced to the sensors within a 50 cm threshold, the quadcopter will react quickly and position it safe within a safe distance of the object.

7.2.2.5 Fire Suppression Release

In order to validate that the fire suppression system will indeed do its job correctly, all individual components must be put through a series of tests. In addition, the system as whole will need to be assembled then tested again to verify its performance as satisfactory. The procedures are listed below.

Procedure A:
Purpose: Verify the software release function works as intended, and the servo and PCB are both capable of handling the written test code that may simulate the application. The state diagram is shown below in Fig 7.2.2.5A

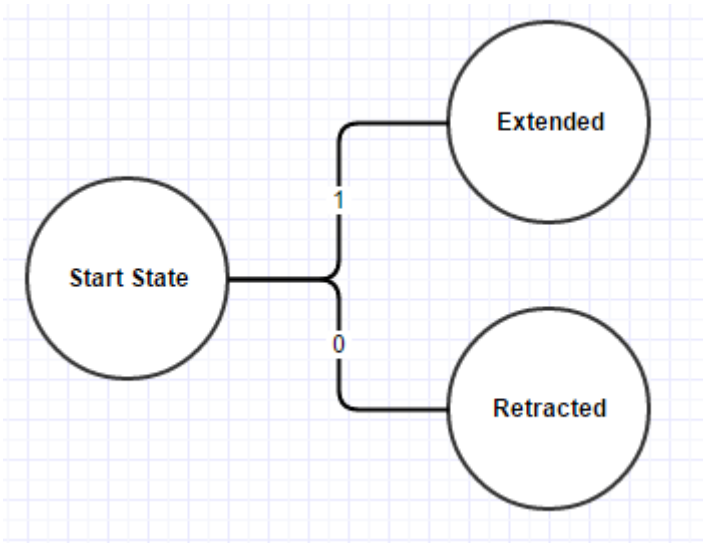


Fig. 7.2.2.5A: State diagram for servo actions

Materials: Servo
Raspberry Pi
Laptop

Steps:

1. Write test code to extend and retract the servo in a loop every 20 seconds.
2. Connect servo and Raspberry Pi.
3. Run test code
4. Verify the servo extends and retracts every 20 seconds.
5. Write test code to extend and retract servo in a loop every 10 seconds.
6. Run test code
7. Verify the servo extends and retracts every 10 seconds.

Conclusion:
Shown below in Table 7.2.2.5B is a table of functionality for each bit-sequence

Bit Sequence	Function
0	Retract
1	Extend

Table 7.2.2.5B: State table corresponding to Figure 7.2.2.5A

Procedure B:
Purpose: Once the software has been proven, we must now write the code to extend and release upon the intended time frame. That is, once the quadcopter has detected the flame, and has positioned itself accordingly, the payload must be delivered. This is assuming the image processing software has been written and is ideal. The state diagram is shown below in Fig. 7.2.2.5C

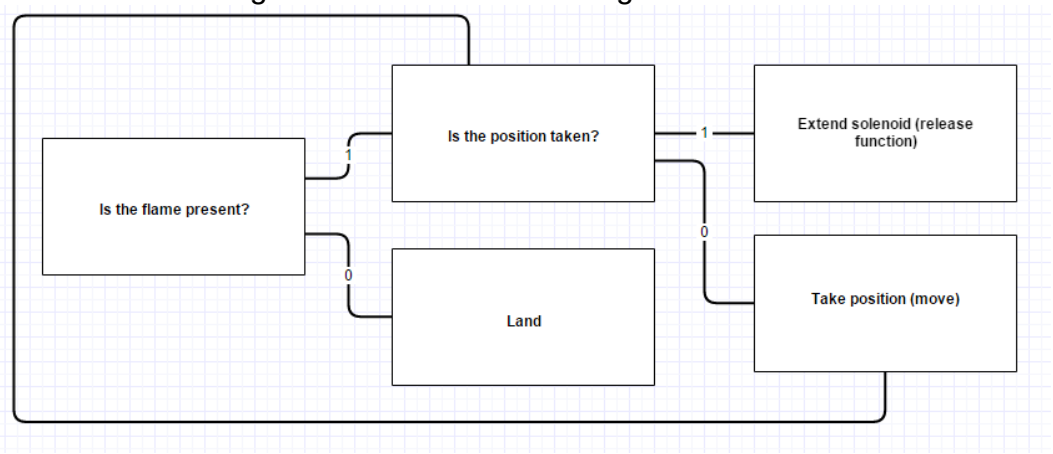


Fig 7.2.2.5C: State diagram for quadcopter actions

Materials: Assembled Quadcopter (including payload release system)
Steps:

1. Load test code onto Raspberry Pi.
2. Run the quadcopter's flight path and flame detection software.
3. Verify the quadcopter positions itself over the target.
4. Verify once the position is taken, the servo retracts/extends.
5. Verify the contents of the container are dispensed onto the target.

Conclusion:

As indicated in the state diagram, a two-bit sequence is used to determine functionality. Table 7.2.2.5D illustrates the functionality of each bit sequence.

Bit Sequence	Function
00	Land
01	Land
10	Continue moving until positioned correctly
11	Release chemical suppressant

Table 7.2.2.5D: State table corresponding to Figure 7.2.2.5C

8 Administrative Content

The administrative content to follow this section is composed of all of our planned testing, our bill of materials, credit to our sponsors, and work distribution. This will attempt to explain any differences between our plans and our executions. For example, our projected budget and our actual budget. The two ended up being quite similar in absolute value, but very different in terms of contents.

8.1 Milestones

Below in Figure 8.1A shows the milestones for Senior Design I which concludes our research phase and will begin our design and prototype phase.

Phase	Start Date	End Date	Duration (weeks)
Research (Senior Design 1)	**2/1/2016**	**3/13/2016**	**6**
1.1 Brainstorming	2/1/2016	2/7/2016	1
1.2 Research Drones	2/8/2016	2/14/2016	1
1.3 Drone Research Cont. (Flight Control)	2/15/2016	2/21/2016	1
1.4 Research Cameras And sensors	2/22/2016	2/28/2016	1
1.5 Research Fire Extinguishing Deployment.	2/29/2016	3/6/2016	1
1.6 Research Communication Methods	3/7/2016	3/13/2016	1
Design (Senior Design 1)	**3/14/2016**	**4/28/2016**	**7**
2.1 Design Drone	3/14/2016	3/20/2016	1
2.2 Design Comm systems	3/21/2016	3/27/2016	1
2.3 Design Flight System	3/28/2016	4/3/2016	1
2.4 Design Fire Extinguishing System	4/4/2016	4/10/2016	1
2.5 Design imaging system	4/11/2016	4/17/2016	1
2.6 Purchase Parts & Testing	4/18/2016	4/24/2016	1
2.7 Finalize Design Paper & Testing	4/25/2016	4/28/2016	1

Fig 8.1A: Senior Design I Milestones

Fig 8.1B below, shows the milestones for Senior Design II which begins our implement phase and ends with our final deployment.

Implementation & Integration (Senior Design 2)	**5/16/2016**	**7/10/2016**	**5**
3.1 Implement Flight System & Navigation	5/16/2016	5/22/2016	1
3.2 Implement Flight System & Navigation	5/23/2016	5/29/2016	1
3.3 Implement Flight System & Navigation	5/30/2016	6/5/2016	1
3.4 Implement Flight System & Extinguisher	6/6/2016	6/12/2016	1
3.5 Implement Extinguisher	6/13/2016	6/19/2016	1
3.6 Implement Imaging System	6/20/2016	7/10/2016	3
3.7 Implement Communication System	6/20/2016	6/26/2016	1
3.8 Integration & Testing	6/27/2016	7/3/2016	1
3.9 Integration & Testing	7/4/2016	7/10/2016	1
Finalize (Senior Design 2)	**7/11/2016**	**7/31/2016**	**3**
4.1 Verify Design Functionality	7/11/2016	7/17/2016	1
4.2 Finalize Design and Documentation	7/18/2016	7/24/2016	1
4.3 Prepare Presentation Materials	7/25/2016	7/31/2016	1

Fig 8.1B: Senior Design II Milestones

8.2 Budget and Financing

Group 7 has received so far $2000 from our sponsor FLIR Systems Inc. This budget includes all parts necessary to build the drone with extra parts in case parts fail, break, or are incompatible with other parts. This amount should prove to be more than enough for all of the hardware components for our quadcopter. Additionally, FLIR will be providing a thermal camera for us the use. Since the quadcopter will be built from scratch all parts will contribute to the cost, such as motors, propellers, frame, camera, flight controller, transmitters, batteries, battery charger, GPS and sensors to name a few. This is all in addition to the hardware that will be used to fabricate the fire suppression release system. That will include some basic plastic, a controller and a servo, and possibly some magnets.

8.3 Bill of Materials

Shown below in Figure 8.3A is the bill of materials for this quadcopter portion.

Line	Part	Model Name	Projected QTY	Actual QTY	Unit Price	Projected Cost	Total Cost
1	FRAME	TAROT 650	1	2	$129.99	$129.99	$259.98
2	MOTORS	TIGER RC MT3515 400KV	4	4	$49.95	$199.80	$199.80
3	MOTORS	TIGER RC MN3515 400KV	0	4	$49.95	0	$199.80
4	PROPELLERS	Iflight 15x5.5 T-style Carbon	4	10	$25.00	$100.00	$250.00
5	FLIGHT CONTROL	3DR PIXHAWK	1	2	$200.00	$200.00	$400.00
6	ESC	Arris 30 A	0	4	$12.50	$0.00	$50.00
7	ESC	Hobbywing Pro 40A	4	4	$25.00	$100.00	$100.00
8	BATTERY	10000MAH 25C 6S	1	1	$199.99	$199.99	$199.99
9	MINIMOSD	APM MinimOSD	1	1	$49.99	$49.99	$49.99
10	TX/RX	BOSCAM 5.8GHZ RC305	1	1	$50.00	$50.00	$50.00
11	TELEMETRY	3DR TELEMETRY RADIO	1	2	$65.99	$65.99	$131.98
12	CAMERA	TAU	1	0	$900.00	$900.00	$0.00
13	GPS	3DR uBlox	1	2	$89.99	$89.99	$179.98
14	SENSORS	HC-SR04 Ultrasonic Module	5	5	$2.50	$12.50	$12.50
15	CHARGER	Hitec X1+	1	1	$75.00	$75.00	$75.00
16	MCU	BEAGLEBBONE BLACK	1	1	$55.00	$55.00	$55.00
17	PDB	OWN DESIGN	1	1	$150.00	$150.00	$150.00
18	FIRE EXTINGUISHER	OWN DESIGN	1	1	$50.00	$50.00	$50.00
19	CABLES		1	1	$30.00	$30.00	$30.00
20	CAMERA	Logitech HD C270	0	1	$10.00	$0.00	$10.00
21	Single Board	Rasberry Pi	0	1	$0.00	$0.00	$35.00
22	MCU	Odroid XU4	0	1	$75.00	$0.00	$75.00
23	Wifi Module	Realtek RTL 8188 CUS	0	1	$0.00	$0.00	$10.00
24	BATTERY	6000 MAH 25C 6S	1	1	$150.00	$150.00	$150.00
25	MISC		1	1	$100.00	$100.00	$100.00
				SUBTOTAL		$2,708.25	$2,824.02
				TAX		$189.58	$180.81
				EST. S&H		$270.83	$258.30

Figure 8.3A: Bill of Materials for the quadcopter

The bill of materials has changed considerably over the course of the project. Due to some very drastic changes, we made some alterations to the bill of materials as needed. During testing, a quadcopter lost signal with the RC and was determined lost after 11 man hours of searching. Therefore, we had to purchase an entirely new system. Which nearly doubled the price of our project. Additionally, since our BeagleBone Black was no longer useful, we decided to purchase a different companion computer, the Odroid XU4, which costed us an extra $100, only to find out the Raspberry Pi suited us best. However, one financial factor that did help in our favor was the availability of the FLIR Tau camera. Normally the camera can be purchase for several thousand dollars, but through our sponsor would only cost $900. Unfortunately, due to the unavailability of certain items and the international regulations placed on some items, we were not able to obtain this camera either. Therefore, that saved our project $900.

Table 8.3B shows the bill of materials for the power distribution board for this project. The power distribution board will be our own design and requires its own BOM.

POWER DISTRIBUTION			
Line	BOARD REF	P/N	VALUE
1	CIN	GRM32ER7YA106KA12L	CAP CER 10UF 35V X7R 1210
2	CBYP	CC0805KRX7R9BB153	CAP CER 0.015UF 50V X7R 0805
3	M1	FDD8647L	MOSFET N-CH 40V 14A DPAK
4	M2	CSD18532Q5B	MOSFET N-CH 60V 23A 8VSON
5	L1.2	XAL1010-562MEB	XAL1010 High Current 5.6 uH 20 % 21.2 A
6	U1.2	LM3150MH/NOPB	IC REG CTRLR BUCK 14TSSOP
7	RON	ERJ-6NF2493V	RES SMD 249K OHM 1% 1/8W 0805
8	CVCC	MK212B7225KG-T	CAP CER 2.2UF 16V X7R 0805
9	RILIM	ERJ-6ENF1001V	RES SMD 1K OHM 1% 1/8W 0805
10	CBST	EMK212B7474KD-T	CAP CER 0.47UF 16V X7R 0805
11	CFF	08055C122KAT2A	CAP CER 1200PF 50V X7R 0805
12	RFB2	ERJ-6ENF7322V	RES SMD 73.2K OHM 1% 1/8W 0805
13	RFB1	P10.0KCCT-ND	RES SMD 10K OHM 1% 1/8W 0805
14	COUT	6SVPE220M	CAP ALUM POLY 220UF 20% 6.3V SMD
15	CSS	CC0805KRX7R9BB153	CAP CER 0.015UF 50V X7R 0805

Table 8.3B: Power Distribution Board PCB parts

8.4 Credit to Sponsors

Group 7 (Adam, Luis, Jamie, and Greg) would like to give a big thanks to FLIR Systems Inc. Without their mentorship, creativity, and financial support, this project would not be possible. Heavily because getting our hands on a capable FLIR camera isn't all that simple for us. Not to mention the cost of the project excluding the FLIR camera would also be a difficult to swallow without the help from FLIR. However, the financial assistance isn't the largest contribution FLIR has made. FLIR has provided insight, ideas, and drive that is worth far more than anything money can purchase. The knowledge that has been passed down thus far from those helping us from FLIR has given us the confidence to take on such a project and have no concept of failure in mind. From the very beginning the FLIR staff has been very welcoming and very helpful throughout the entire process, from creating the ideas, coming up with solutions, mentoring with calculations that we are not otherwise familiar with, to ordering components, explaining the team design dynamic, and giving advice and insight from their personal experiences and relating them to us. As stated previously, we owe a large portion of our success up

to this point to FLIR, because just without their, mentoring, this project would be far more difficult. Add the financial responsibility into the mix, and the outcome becomes even more improbable, and for that, we would like to give FLIR Systems Inc. a special thanks.

8.5 Distribution and Scope of Work

The Distribution of work was split up among different systems for the entire project. There have been 4 main pieces to the project and each group member was mainly responsible for their section, but is not limited to their section. The distribution is shown below in Table 8.5A.

	Adam	Luis	Jamie	Greg
Mission Software	✖		✖	✖
Power Distribution Board	✖	✖		
Image Processing			✖	✖
Quadcopter hardware		✖		

✖ Primary

✖ Secondary

Table 8.5A: Distribution Table

As the table shows, Adam is responsible mostly for the mission script software, with contribution in the power distribution board design. The mission script software includes the grid searching algorithm and the servo release software. Luis is responsible mostly for the power distribution, with contribution in the hardware design. Hardware design being battery and fire extinguishing container. Greg was mostly responsible for mission script software and image processing. Jamie was responsible for image processing with light contribution for fire suppression release software. During integration, all group members will inevitably work together in order to ensure, their piece is working as intended and is fulfilling all functional requirements. In addition to the responsibility chart, Adam is leading the team through the project and is the established group leader. The distribution of work was dictated according to each group member's interest in the portion of the project, as well as their skillset. For example, since Luis is the only electrical engineer, in the group, it makes sense to put him in charge of the power distribution board.

The scope of work was decided that only software and hardware that each member is capable of learning and completing within the given period of time will be necessary. Other parts and pieces will be purchased given time, resource, and knowledge constraints. Surely, it isn't expected of the group to build our own frame for the quadcopter from scratch. There isn't any doubt that this task could be completed, however, this task doesn't apply to the problem at hand when taking into account the purpose and the focus of the project. Therefore, purchasing a built frame is wise than wasting time and resources trying to build one. Factoring all of

this in, the scope has been narrowed all the way down to circuit level, where Luis will be designing a printed circuit board for our power distribution. Additionally, higher level software will be in the scope of work in order to process the imaging and such, as well as low level in order to integrate all hardware systems together. The scope of work will thus range, from very low level to very high level in order to demonstrate the skills learned throughout the various courses in our degree curriculum. Shown below is the table of contents with each group member's contribution for the document and overall research. The legend for the document and research contribution is shown below in Figure 8.5C

Multiple Member's Section

Luis's Sections

Adam's Sections

Jamie's Sections

Greg's Sections

Figure 8.5C: Member Section Contribution Color Code

As illustrated in Figure 8.5C, Luis was responsible for sections in red, Adam was responsible for sections in blue, Jamie for purple, and Greg for green.

8.6 Final Conclusions and Results

To conclude our project, our goal was to simply implement and demonstrate a concept that unmanned aerial vehicles can assist first responders in the field and can effectively help reduce the risk of lives and damages. Fires are responsible for an unnecessary amount of injuries, deaths, and financial damages and we're seeking to prove a concept that can help reduce these factors greatly. For a large portion of the testing, we seemed to continue to develop problems and hit road blocks. Losing a drone was a major loss.

Virtually an entire week of work was lost due to searching, and waiting for new parts to arrive. Not to mention, after nearly almost every failed test where the drone fell out of the air, it required hours of labor to piece back together. Resources such as money, time, and even brain power began to grow scarce near the end. However, once things started picking up with little time to go, we eventually pulled off a successful project following the testing guidelines we set in the previous semester. Testing the systems individually and gradually integrating and testing again helped us to define the root of failures when they did occur. There were some minor and major setbacks that led us to making the choices we did. Some led to changing major parts of the project, such as scrapping our android application, and scrapping our obstacle avoidance section. The obstacle avoidance became too unreliable because of the sensors that were chosen. This was such a minor section of the project that we had no time to devote to it when we were nearing the end of the project. Choosing the correct companion computer also proved to be somewhat difficult, but after trying 3, we found a good match. Overall a success, but one that required much hard work, and many late nights.

9 Appendices

9.1 References and Citations

1) Felzenszwalb, P. F. and Girshick, R. B. and McAllester, D. and Ramanan, D. *Object Detection with Discriminatively Trained Part Based Models*. PAMI, vol. 32, no. 9, pp. 1627-1645, September 2010
2) Rainer Lienhart and Jochen Maydt. *An Extended Set of Haar-like Features for Rapid Object Detection*. Submitted to ICIP2002.
3) Shengcai Liao, Xiangxin Zhu, Zhen Lei, Lun Zhang and Stan Z. Li.*Learning Multi-scale Block Local Binary Patterns for Face Recognition*. International Conference on Biometrics (ICB), 2007, pp. 828-837.
4) Paul Viola, Michael Jones. *Rapid Object Detection using a Boosted Cascade of Simple Features*. Conference on Computer Vision and Pattern Recognition (CVPR), 2001, pp. 511-518.
5) "SSCI, MIT and Olin College Demonstrate Autonomous Aerial Monitoring and Tracking Capability for Forest Fires Using Multiple Unmanned, Aerial Vehicles (UAVs)." *Scientific Systems Company, Inc.* 26 Oct. 2015. Web. 27 Mar. 2016.
6) "5 Drone Technologies for Firefighting | Fire Chief." *Fire Chief*. Ed. Mary Rose Roberts. 20 Mar. 2014. Web. 27 Mar. 2016.
7) "BoneScript Library." *BoneScript*. Web. 25 Apr. 2016.
8) "PyBBIO Archives - Gray Cat Labs." *Gray Cat Labs PyBBIO Tag*. Web. 25 Apr. 2016.
9) "OpenCV: Cascade Classifier Training." *OpenCV: Cascade Classifier Training*. 18 Dec. 2015. Web. 25 Apr. 2016.
10) "Beagleboard: BeagleBoneBlack." *ELinux.org*. Web. 25 Apr. 2016.
11) "U.S. Fire Statistics." *U.S. Fire Administration*. Web. 27 Apr. 2016. https://www.usfa.fema.gov/data/statistics/
12) "A Servo Tutorial." *Arduino Playground*. Web. 27 Apr. 2016. http://playground.arduino.cc/Learning/ServoTutorial
13) "University of Missouri - College of Engineering." *Engineering*. Web. 27 Apr. 2016. http://engineering.missouri.edu/mae/files/realistic_constraints.pdf
14) "Publications of Sergio Montenegro." *Publications of Sergio Montenegroe*. Web. 27 Apr. 2016.
15) "Introduction to A* From Amit's Thoughts on Pathfinding." *Introduction to A**. Web. 27 Apr. 2016.
16) "HC-SR04 User's_Manual." *Google Docs*. Web. 27 Apr. 2016.
17) "Baking Soda." *Density in 285 Measurement Units*. Web. 27 Apr. 2016.
18) "Advanced MultiCopter Design" ArduPilot. 25 April 2016. Web. 20 March 2016. http://ardupilot.org/copter/docs/advanced-multicopter-design.html
19) Brian Schneider. "A Guide to Understanding LiPo Batteries" Rogers Hobby Center. May 2012. Web. 20 March 2016.

20) http://www.rogershobbycenter.com/lipoguide/
21) Sam. “Brushless motors - how they work and what the numbers mean” Dronetest. October 2014. Web. Mar 25 2016
22) http://www.dronetrest.com/t/brushless-motors- how-they- work-and-what-the- numbers-mean/564
23) Clym Montgomery. “Multi-Rotors, First-Person View, And The Hardware You Need” Tom’s Hardware. June 2014. Web. March 25 2016.
24) http://www.tomshardware.com/reviews/multi-rotor-quadcopter-fpv,3828- 6.html
25) “Rechargeable Lithium Batteries” Battery and Energy Technologies. May 2015. Web. March 25 2016.
26) http://www.mpoweruk.com/lithiumS.htm#polymer
27) Ian Poole. “Frequency Modulation Bandwidth, Spectrum & Sidebands” Radio-Electronics. April 2014. Web. March 25 2016.
28) http://www.radio-electronics.com/info/rf-technology-design/fm-frequency-modulation/spectrum-bandwidth-sidebands.php
29) Greg Covey. “Greg Covey's Amp'd Issue 21: Brushless Basics” RC Universe. May 2011. Web. March 10 2016
30) http://www.rcuniverse.com/magazine/article_display.cfm?article_id=1344
31) Lorenz Meier, Dominik Honegger and Marc Pollefeys. “PX4: A Node-Based Multithreaded Open Source Robotics Framework for Deeply Embedded Platforms” ICRA (Int. Conf. on Robotics and Automation) 2015. Web. March 10 2016.
32) https://Pixhawk.org/modules/Pixhawk
33) Bernard Chevalier. “Turnigy 9x 2.4GHz radio TGY.” Personal-Drones. September 2010. Web. April 1 2016.
34) http://www.personal-drones.net/wp- content/uploads/2013/08/Turnigy-9x.pdf

9.2 Copyright Permissions

Figure 3.2.1.7A: “The Cascade of Classifier” by Jeff Bier, the Founder of Embedded Vision Alliance

Figure 3.2.1.7B: “Feature Prototypes of Haar-like Features” from OpenCV User Guide Documentation

Figure 3.2.1.7C: “Equations for Integral Image Calculation” from OpenCV User Guide Documentation

Re:
To: Jamie Peck

Hello!

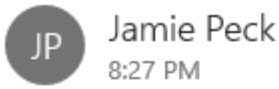
You are free to use it as long as you mention the origin of the figures

Regards,
Vadim

27 марта 2016 г., в 5:46, Jamie Peck <jamiepeck@knights.ucf.edu> написал(а):

> Hello,
>
> I am a Computer Engineering student at the University of Central Florida. Currently, as a senior, I am working on a design project that involves image processing and object detection via cascade classifier training algorithms. Per design documentation efforts, my team and I would like to request permission to use some of the figures listed in the OpenCV User Guide documentation. We greatly appreciate any help you can provide regarding this issue!
>
> Regards,
> Jamie Peck

Figure 4.2.1.7A: BeagleBone Black Pin Headers



BeagleBone Pin Header Diagram
To: alex@graycat.io

Hello,

I am a Computer Engineering student at the University of Central Florida. Currently, as a senior, I am working on a design project that involves the BeagleBone development board and its I2C protocols. Per design documentation efforts, my team and I would like to request permission to use the pinout diagram you have posted on your Grey Cat Labs Blog. I greatly appreciate any help you can provide regarding this issue!

Regards,
Jamie Peck

Figure 4.2.1.7B: BeagleBone Black Development Board

Permission of image use

LB Luis Brum

To: admin@dji.com;

Reply all | 7:44 PM

From: Luis Brum
Sent: Mon 4/25/2016 7:44 PM
To: admin@dji.com;

Dear Sir/Ma'am,

My name is Luis Brum and I am a current Electrical Engineer student at University of Central Florida. I am currently working on a senior design project of a fire extinguishing quadcopter system. I would like to ask permission use the image of the types of multirotor supported from your website in my research paper. This project will only be used for academic purposes, and will not be used for economic reasons. I greatly appreciate your help in this matter, and hope DJI will approve the use of this Image.

Thank you,

Luis Brum

Permission of image use

LB Luis Brum

To: admin@dji.com;

Reply all | 7:44 PM

Dear Sir/Ma'am,

My name is Luis Brum and I am a current Electrical Engineer student at University of Central Florida. I am currently working on a senior design project of a fire extinguishing quadcopter system. I would like to ask permission use the image of the types of multirotor supported from your website in my research paper. This project will only be used for academic purposes, and will not be used for economic reasons. I greatly appreciate your help in this matter, and hope DJI will approve the use of this Image.

Thank you,

Luis Brum

Request of image use.

LB Luis Brum

To: ☐ service02@rctigermotor.com;

Reply all | 7:59 PM

From: Luis Brum
Sent: Mon 4/25/2016 7:59 PM
To: ☐ service02@rctigermotor.com;

Dear Sir/Ma'am,

My name is Luis Brum and I am a current Electrical Engineer student at University of Central Florida. I am currently working on a senior design project of a fire extinguishing quadcopter system. The quadcopter I am designing will be using the MT3515 KV400 motors manufactured by Tiger RC. I would like to ask permission use the images of the MT3515 KV400 motor provided on your website for my research paper. This project will only be used for academic purposes, and will not be used for economic reasons. I greatly appreciate your help in this matter, and hope DJI will approve the use of this Image.

Thank you,

Luis Brum

From: Luis Brum
Sent: Mon 4/25/2016 8:12 PM
To: ☐ greg@rcuniverse.com;

Mr . Covey,

My name is Luis Brum and I am a current Electrical Engineer student at University of Central Florida. I am currently working on a senior design project of a fire extinguishing quadcopter system. I would like to ask permission use the image of the Simplified Electronic Speed Controller the "RCU Review: Greg Covey's Amp'd Issue 21: Brushless Basics" in my research paper. This project will only be used for academic purposes, and will not be used for economic reasons. I greatly appreciate your help in this matter, and hope you will approve the use of this Image.

Thank you,

Luis Brum

From: Luis Brum
Sent: Mon 4/25/2016 8:21 PM
To: ☐ cs@hobbywing.com;

Miss Guo,

My name is Luis Brum and I am a current Electrical Engineer student at University of Central Florida. I am currently working on a senior design project of a fire extinguishing quadcopter system. The quadcopter I am designing will be using the XRotor Pro 40A ESC manufactured by Hobbywing. I would like to ask permission use the data table and specifications and images related to the Xrotor Pro 40A ESC provided on your website for my research paper. This project will only be used for academic purposes, and will not be used for economic reasons. I greatly appreciate your help in this matter, and hope Hobbywing will approve the use of this image.

Thank you,

Luis Brum

Figure 4.2.1.2A: Pixhawk Interfaces. CC BY-SA 3.0

Figure 6.1.5B: Pixhawk Connection Assembly Reprinted with permission from Pixhawk. CC BY-SA 3.0

Figure 6.1.5A: Pixhawk Components Reprinted with permission from Pixhawk. CC BY-SA 3.0

Figure 4.2.1.6A: 3DR Ublox GPS with Compass unit. Reprinted with permission from 3DR CC BY-SA 4.0

Figure4.2.3A: 3DR Radio V2. Reprinted with permission from 3DR. CC BY-SA 4.0

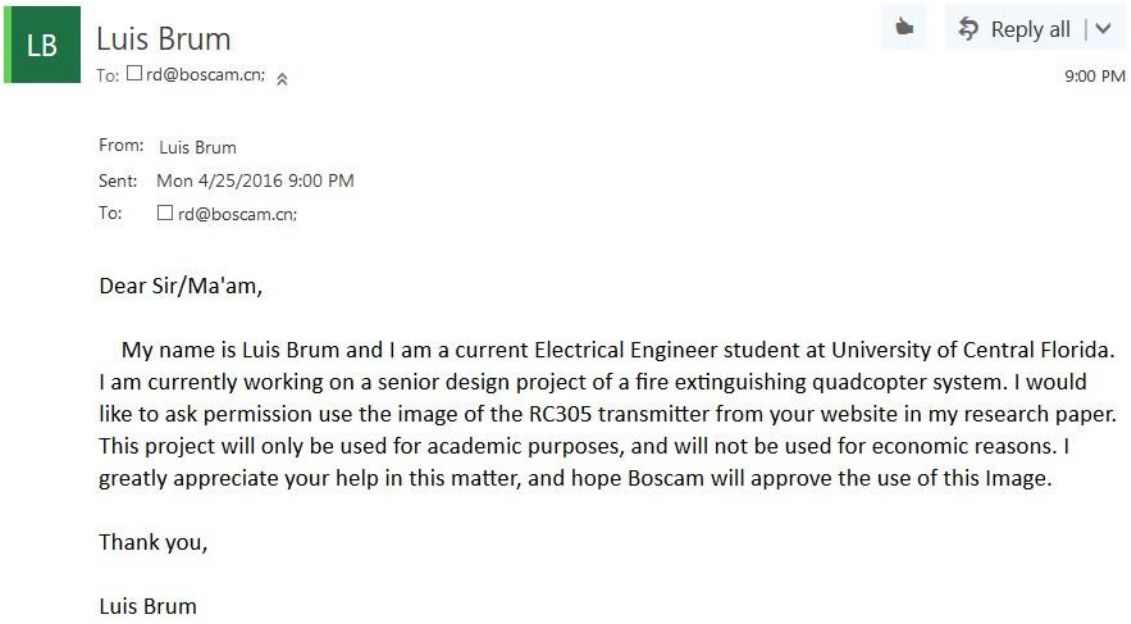
From: Luis Brum
Sent: Mon 4/25/2016 8:56 PM
To: ☐ rd@boscam.cn;

Dear Sir/Ma'am,

My name is Luis Brum and I am a current Electrical Engineer student at University of Central Florida. I am currently working on a senior design project of a fire extinguishing quadcopter system. I would like to ask permission use the image of the TS351 transmitter from your website in my research paper. This project will only be used for academic purposes, and will not be used for economic reasons. I greatly appreciate your help in this matter, and hope Boscam will approve the use of this Image.

Thank you,

Luis Brum



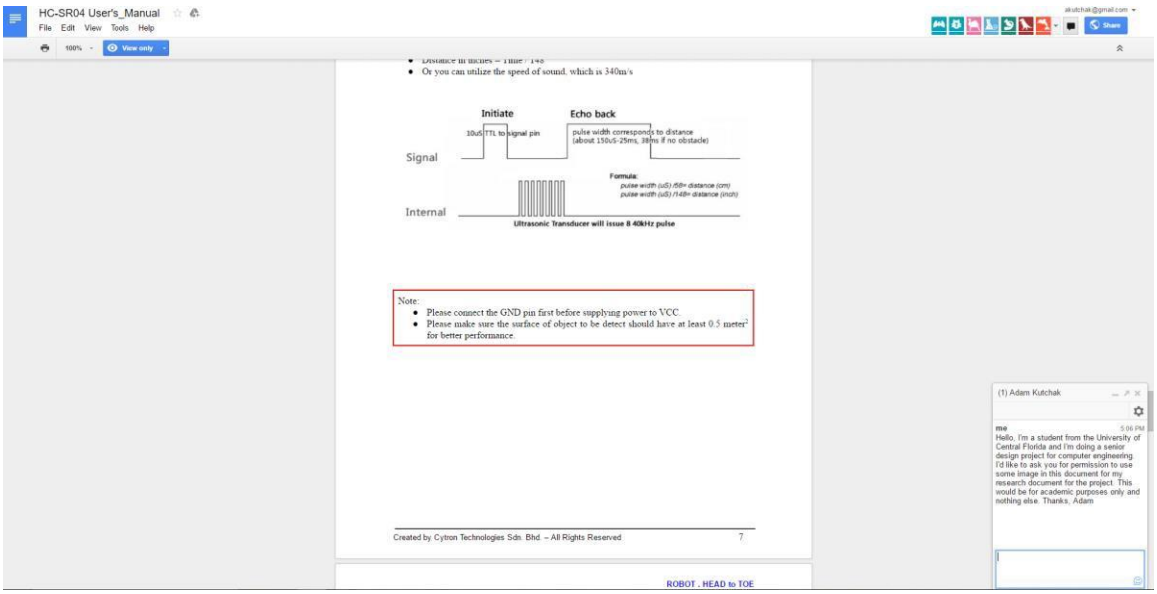
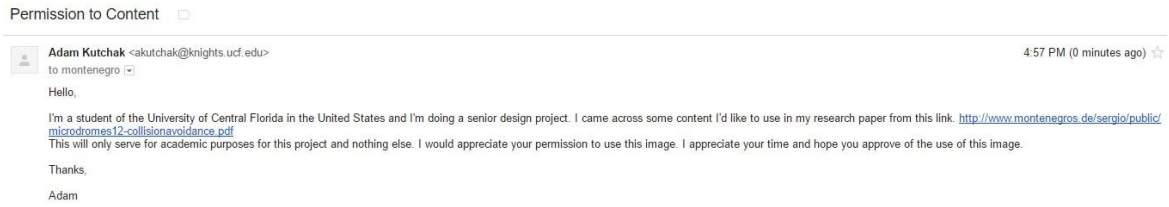


Figure 6.1.6D

Figures 6.2.1A, 6.2.1B, and 6.2.1C



Figures 6.2.2A and 6.2.2B