

Senior Design Project:

AUVSI SUAS Competition Drone Electronics



UCF

University of Central Florida
Fall 2018 - Spring 2019
Department of Electrical & Computer Engineering

Group 11

Brandon Cuevas (EE & CpE)
Garett Goodale (EE)

Nicholas Omusi (EE & CpE)
Nicholas Peters (EE)

Dr. Chung Yong Chan (Mentor)

Contents

1	EXECUTIVE SUMMARY	1
2	PROJECT DESCRIPTION.....	2
2.1	OBJECTIVES	2
2.1.1	<i>Autonomous Flight</i>	3
2.1.2	<i>Payload Delivery</i>	3
2.1.3	<i>Area Search.....</i>	4
2.2	REQUIREMENTS SPECIFICATIONS.....	4
2.2.1	<i>UAV Flight and Payload Requirements.....</i>	4
2.2.2	<i>UAV Image Processing & Communication Requirements</i>	6
2.3	HOUSE OF QUALITY.....	7
2.4	EXISTING SIMILAR PROJECTS AND PRODUCTS	8
2.4.1	<i>Université de Sherbrooke.....</i>	9
2.5	INITIAL DESIGN ARCHITECTURES AND RELATED DIAGRAMS	10
2.6	OVERALL SYSTEM DIAGRAM	11
3	RELATED STANDARDS AND REALISTIC DESIGN CONSTRAINTS.....	13
3.1	STANDARDS.....	13
3.1.1	<i>Communications Standards</i>	13
3.2	REALISTIC DESIGN CONSTRAINTS	14
3.2.1	<i>Economic and Time.....</i>	14
3.2.2	<i>Environmental, Social, Political</i>	14
3.2.3	<i>Ethical, Health, and Safety</i>	15
3.2.4	<i>Manufacturability and Sustainability</i>	15
3.2.5	<i>Competition Rules</i>	15
3.3	REGULATIONS.....	15
3.3.1	<i>FAA Regulations</i>	16
3.3.2	<i>FCC Regulations</i>	16
3.4	RECEIVED COMPONENTS.....	17

4	BACKGROUND RESEARCH.....	21
4.1	TELEMETRY PLATFORM	21
4.1.1	<i>Context of Platform Requirements</i>	21
4.1.2	<i>Overview of RF Sensitivity and Telemetry Techniques</i>	23
4.1.3	<i>Texas Instruments SimpleLink CC13x0 (Sub-1 GHz)</i>	23
4.1.4	<i>Digi Xbee-PRO 900HP RF Module (Zigbee)</i>	25
4.1.5	<i>Microchip RN2483(LoRa)</i>	26
4.1.6	<i>RFD900 (Sub-1 GHz)</i>	27
4.2	HIGH-SPEED WIRELESS COMMUNICATION.....	28
4.2.1	<i>Device Considerations</i>	28
4.3	WIRELESS COMMUNICATION CONSIDERATIONS	30
4.3.1	<i>Power Considerations</i>	30
4.3.2	<i>Antenna Considerations</i>	31
4.3.3	<i>MIMO</i>	38
4.3.4	<i>Fresnel Zones</i>	38
4.4	WIRELESS COMMUNICATION PROTOCOLS	40
4.4.1	<i>Wi-Fi</i>	40
4.4.2	<i>Bluetooth & BLE</i>	43
4.4.3	<i>LoRaWAN</i>	46
4.4.4	<i>Zigbee & DigiMesh</i>	49
4.5	WIRED COMMUNICATION	51
4.5.1	<i>Parallel vs. Serial</i>	52
4.5.2	<i>I²C</i>	53
4.5.3	<i>I²C isolation</i>	54
4.5.4	<i>SPI</i>	55
4.5.5	<i>Asynchronous Serial Protocol</i>	58
4.6	BATTERY TYPES	61

4.6.1	<i>Lead Acid</i>	61
4.6.2	<i>NiMH/NiCd</i>	62
4.6.3	<i>Lithium-ion</i>	62
4.7	BATTERY MONITORING.....	63
4.7.1	<i>Current sensing</i>	64
4.7.2	<i>Kelvin Sensing</i>	65
4.7.3	<i>Battery Monitoring Requirements</i>	66
4.7.4	<i>Battery Gas Gauge ICs</i>	67
4.7.5	<i>LTC2944</i>	67
4.7.6	<i>TI BQ34110</i>	69
4.7.7	<i>TI BQ34Z100-G1</i>	70
4.8	VOLTAGE REGULATION SYSTEM.....	71
4.8.1	<i>Linear Regulators</i>	71
4.8.2	<i>Switching Regulators</i>	72
4.8.3	<i>Voltage Regulator Components</i>	74
4.9	ELECTRIC MOTORS.....	74
4.9.1	<i>DC Motors</i>	75
4.10	ELECTRONIC SPEED CONTROL.....	77
4.10.1	<i>Brushed Speed Control</i>	77
4.10.2	<i>Brushless Speed Control</i>	79
4.10.3	<i>ESC Comparison</i>	80
4.11	I2C ADDRESS CONVERSION BOARD.....	81
4.11.1	<i>LTC 4316-18</i>	81
4.11.2	<i>TCA9544APWR</i>	82
4.12	FLIGHT CONTROL UNIT CONSIDERATIONS.....	83
4.12.1	<i>BeagleBone Black</i>	84
4.12.2	<i>Pixracer</i>	85

4.12.3	<i>BeagleBone Blue</i>	86
4.13	AUTOPILOT SOFTWARE	87
4.14	GROUND CONTROL SOFTWARE	89
4.15	CAMERA CONSIDERATIONS.....	91
4.15.1	<i>Pixel Ground Size</i>	91
4.15.2	<i>Frame Ground Area</i>	91
4.15.3	<i>Pixel Ground Size and Frame Ground Area Calculations</i>	92
4.16	USB CAMERA OPTIONS	93
4.16.1	<i>IDS UI-3590L</i>	93
4.16.2	<i>e-CAM210_MI230</i>	94
4.16.3	<i>See3CAM 135</i>	94
4.17	ISSUES WITH USB CAMERAS	94
4.18	TRI-CAM SOLUTION.....	95
4.19	CAMERA GIMBAL.....	95
5	HARDWARE AND SOFTWARE DESIGN	96
5.1	HARDWARE DETAIL DESIGN.....	96
5.1.1	<i>Battery Packs</i>	96
5.1.2	<i>Battery Monitoring</i>	97
5.1.3	<i>Voltage Divider Calculation</i>	99
5.1.4	<i>Battery Gas Gauge Schematic and Layout</i>	100
5.1.5	<i>I2C Address Conversion</i>	102
5.1.6	<i>Pull-up Resistor Calculation</i>	103
5.1.7	<i>I2C Address Converter Schematic and Layout</i>	105
5.2	SOFTWARE DESIGN	107
5.2.1	<i>Battery Gas Gauge</i>	107
5.3	JETSON CAMERAS.....	108
5.4	FLIGHT CONTROLLER SETUP	109

5.4.1 *Preparing the BeagleBone* 110

5.4.2 *Installing ArduPilot*..... 112

5.4.3 *Component Connections*..... 115

6 TESTING **117**

6.1 UBIQUITI ROCKET M5 TESTING 117

 6.1.1 *Variable Distance Testing* 117

 6.1.2 *Fixed Distance Testing*..... 119

6.2 TI CC1350 SIMPLE LINK TESTING 120

6.3 HEXACOPTER FLIGHT TESTING 122

6.4 BREADBOARD TESTING 124

7 ADMINISTRATIVE CONTENT **125**

 7.1 PROJECT MILESTONES 125

 7.2 BUDGET AND FINANCE..... 125

8 REFERENCES **A**

9 APPENDIX **F**

Table of Figures

FIGURE 1. HOUSE OF QUALITY ANALYSIS..... 8

FIGURE 2. UNIVERSITÉ DE SHERBROOKE 2018 AUVSI SUAS DRONE 9

FIGURE 3. DIAGRAM OF ELECTRONICS SHOWING RESPONSIBLE PERSONNEL AND STATUS 11

FIGURE 4. OVERALL SYSTEM DIAGRAM 12

FIGURE 5. UBIQUITI ROCKET M5’S AND OTHER MISCELLANEOUS MATERIALS..... 18

FIGURE 6. ELECTRICAL COMPONENTS FOR THE TEST DRONE 19

FIGURE 7. TEST DRONE..... 19

FIGURE 8. NVIDIA JETSON DEV KIT CONNECTED FOR TESTING 20

FIGURE 9. NVIDIA JETSON CLOSE-UP WITH I2C ISOLATION PCB MOUNTED 20

FIGURE 10. TYPICAL RADIATION PATTERNS FOR A MONOPOLE ANTENNA 32

FIGURE 11. TYPICAL RADIATION PATTERNS FOR A DIPOLE ANTENNA 33

FIGURE 12. RADIATION PATTERNS FOR AN AMO-5G13 FROM UBIQUITI..... 34

FIGURE 13. RADIATION PATTERNS FOR THE AM-5G-19-120 BY UBIQUITI..... 35

FIGURE 14. TYPICAL RADIATION PATTERNS FOR A YAGI ANTENNA 36

FIGURE 15. TYPICAL RADIATION PATTERN FOR A CLOVERLEAF ANTENNA..... 37

FIGURE 16. TYPICAL RADIATION PATTERNS OF A HELICAL ANTENNA CONFIGURED TO OPERATE AXIALLY 38

FIGURE 17. 2.4 GHZ WI-FI CHANNELS (GAUTHIER, 2009). THE USE OF THIS IMAGE IS GOVERNED BY THE CC BY-SA 3.0 (CREATIVE COMMONS CORPORATION, 2007). 41

FIGURE 18. DIAGRAM SHOWING THE STRUCTURE OF A MAC-48 NETWORK ADDRESS, EXPLICITLY SHOWING THE POSITIONS OF THE MULTICAST/UNICAST BIT AND THE OUI/LOCAL ADDRESS TYPE BIT (KJU, 2007). THE USE OF THIS IMAGE IS GOVERNED BY THE CC BY-SA 2.5 (CREATIVE COMMONS CORPORATION, 2005). 42

FIGURE 19. BLUETOOTH PROTOCOL STACK (LESZEK CHUCHLA, BLUETOOTH PROTOKOLY, 2010). THE USE OF THIS IMAGE IS GOVERNED BY THE GFDL (FREE SOFTWARE FOUNDATION, 2008). 44

FIGURE 20. BLUETOOTH PACKET STRUCTURE (LESZEK CHUCHLA, BLUETOOTH RAMKA, 2007). THE USE OF THIS IMAGE IS GOVERNED BY THE GFDL (FREE SOFTWARE FOUNDATION, 2008). 44

FIGURE 21. ZIGBEE MESH NETWORK NODES. THE USE OF THIS IMAGE IS GOVERNED BY THE CC BY-SA. 50

FIGURE 22. PARALLEL (TOP) VS. SERIAL (BOTTOM) COMMUNICATION (SPARKFUN ELECTRONICS, 2012). THE USE OF THIS IMAGE IS GOVERNED BY THE CC BY-SA 4.0 (CREATIVE COMMONS CORPORATION, 2013). 52

FIGURE 23. SPI COMMUNICATION (GRUSIN, 2013). THE USE OF THIS IMAGE IS GOVERNED BY THE CC BY-SA 4.0 (CREATIVE COMMONS CORPORATION, 2013). 56

FIGURE 24. SPI COMMUNICATION WITH MULTIPLE SLAVES (GRUSIN, 2013). THE USE OF THIS IMAGE IS GOVERNED BY THE CC BY-SA 4.0 (CREATIVE COMMONS CORPORATION, 2013). 57

FIGURE 25. SPI COMMUNICATION WITH MULTIPLE SLAVES IN DAISY-CHAIN CONFIGURATION (GRUSIN, 2013). THE USE OF THIS IMAGE IS GOVERNED BY THE CC BY-SA 4.0 (CREATIVE COMMONS CORPORATION, 2013)..... 58

FIGURE 26. A FRAME OF ASYNCHRONOUS SERIAL DATA (SPARKFUN ELECTRONICS, 2012). THE USE OF THIS IMAGE IS GOVERNED BY THE CC BY-SA 4.0 (CREATIVE COMMONS CORPORATION, 2013). . 59

FIGURE 27. SIMPLIFIED UART DIAGRAM (SPARKFUN ELECTRONICS, 2012). THE USE OF THIS IMAGE IS GOVERNED BY THE CC BY-SA 4.0 (CREATIVE COMMONS CORPORATION, 2013). 60

FIGURE 28. DIAGRAM FOR A DISCRETE LINEAR REGULATOR 72

FIGURE 29. CIRCUIT FOR A DISCRETE SWITCHING REGULATOR..... 73

FIGURE 30. BRUSHED DC MOTOR VS. BRUSHLESS DC MOTOR. THE USE OF THIS IMAGE IS GOVERNED BY THE CC BY-SA. 76

FIGURE 31. SIMPLE PWM DIAGRAM. THE USE OF THIS IMAGE IS GOVERNED BY THE CC BY-SA. 78

FIGURE 32. SIMPLE HALF H-BRIDGE CIRCUIT. THE USE OF THIS IMAGE IS GOVERNED BY THE CC BY-SA. 79

FIGURE 33. FULL H BRIDGE CIRCUIT. THE USE OF THIS IMAGE IS GOVERNED BY THE CC BY-SA. 79

FIGURE 34. BASIC DIAGRAM FOR A BRUSHLESS MOTOR AND CONTROLLER. THE USE OF THIS IMAGE IS GOVERNED BY THE CC BY-SA. 80

FIGURE 35. BASIC BLOCK DIAGRAM DESCRIBING HOW THE LTC4316 OPERATES (PERMISSION REQUESTED FROM ANALOG DEVICES)..... 82

FIGURE 36. CONTROL REGISTER INPUTS (TABLE COURTESY OF TI) 82

FIGURE 37. FUNCTIONAL DIAGRAM FOR THE TCA9544A (FIGURE COURTESY OF TI) 83

FIGURE 38. IMAGE OF THE BEAGLEBONE BLUE (IMAGE UNDER CREATIVE COMMONS, COURTESY OF BEAGLEBOARD.ORG) 86

FIGURE 39. SCREENSHOT OF BASIC SCREEN VIEW OF MISSION PLANNER (IMAGE UNDER CREATIVE COMMONS, COURTESY OF ARDUPILOT DEV TEAM) 89

FIGURE 40. BATTERY GAS GAUGE SCHEMATIC..... 100

FIGURE 41. BATTERY GAS GAUGE PCB LAYOUT..... 102

FIGURE 42. MAXIMUM PULL-UP RESISTANCE VS. BUS CAPACITANCE. REPRODUCED WITH PERMISSION FROM TI **APPENDIX A**..... 103

FIGURE 43. MINIMUM PULL-UP RESISTANCE VS V_{DPUX} 105

FIGURE 44. I2C ADDRESS CONVERTER SCHEMATIC..... 106

FIGURE 45. TRACE LAYOUT FOR THE I2C ADDRESS CONVERSION BOARD 107

FIGURE 46. SIGNAL QUALITY VS. DISTANCE FOR THE ROCKET M5 TEST ON MEMORY MALL 118

FIGURE 47. CC1350 AVERAGE RSSI VS. DISTANCE..... 122

FIGURE 48. CC1350 FAIL RATE PERCENTAGE VS. DISTANCE..... 122

FIGURE 49. HEXACOPTER READY FOR FLIGHT 123

FIGURE 50. HEXACOPTER UNDERGOING FLIGHT TESTS..... 123

Table of Tables

TABLE 1. ECE CONTRIBUTIONS TO THE SUAS.....	3
TABLE 2. DRONE PAYLOAD AND FLIGHT SYSTEM REQUIREMENTS	5
TABLE 3. DRONE IMAGE PROCESSING AND COMMUNICATION SUBSYSTEM REQUIREMENTS.....	6
TABLE 4. UGV AND GCS SYSTEM REQUIREMENTS	7
TABLE 5. FCC 5GHZ RULES SUMMARIZED. PERMISSION REQUESTED FROM AIR802.	17
TABLE 6. MATERIALS ALREADY OBTAINED.....	18
TABLE 7. ALLOWED RF COMMUNICATION BANDS FOR AUUVSI SUAS COMPETITION	22
TABLE 8. SUMMARY CC13X0 FAMILY CHARACTERISTICS.....	24
TABLE 9. SUMMARY OF XBEE-PRO 900HP CHARACTERISTICS	25
TABLE 10. SUMMARY OF MICROCHIP RN2483 CHARACTERISTICS	26
TABLE 11. SUMMARY OF RFD900 CHARACTERISTICS	27
TABLE 12. COMPARISON OF UBQUITI AIRMAX PRODUCTS.....	30
TABLE 13. RECEIVE POWER SYMBOL DEFINITIONS.....	31
TABLE 14. FRESNEL ZONE SYMBOL DEFINITIONS.....	39
TABLE 15. SYMBOL DEFINITION FOR THE APPROXIMATE FORMULA FOR THE RADIUS AT THE MIDPOINT OF THE FIRST FRESNEL ZONE.	39
TABLE 16. IEEE WI-FI NETWORKING STANDARDS 802.11A AND 802.11B	41
TABLE 17. IEEE WI-FI NETWORKING STANDARDS 802.11G, 802.11N AND 802.11B.....	42
TABLE 18. BLUETOOTH CLASSIC VS. BLE (INFORMATION TAKEN FROM RF WIRELESS WORLD) (RF WIRELESS WORLD, 2012).....	46
TABLE 19. LORAWAN DEVICE CLASSES (INFORMATION TAKEN FROM LORA ALLIANCE)	48
TABLE 20. LORAWAN SPECIFICATION SUMMARY FOR NORTH AMERICA (INFORMATION TAKEN FROM LORA ALLIANCE)	49
TABLE 21. COMPARISON SUMMARY OF ZIGBEE MESH VS. DIGIMESH (INFORMATION TAKEN FROM DIGI INTERNATIONAL)	51
TABLE 22. COMPARISON OF COMMERCIALY AVAILABLE ESCs.....	81
TABLE 23. COMPARISON OF FLIGHT CONTROLLERS.....	87
TABLE 24. SOME EXAMPLES OF PIXEL CALCULATION VALUES FOR POPULAR CAMERAS	93

TABLE 25. BATTERY GAS GAUGE IC PRICING	97
TABLE 26. BATTERY MONITOR BOM (EXCLUDING MINOR COMPONENTS)	99
TABLE 27. COST OF MAJOR COMPONENTS ON THE I2C ADDRESS CONVERSION BOARD	103
TABLE 28. THE RECORDED VALUES FOR DISTANCE, CAPACITY AND QUALITY AS RECORDED IN THE ROCKET M5 TEST ON OCTOBER 12 TH , 2018	118
TABLE 29. RESULTS FROM ROCKET M5 TESTING AT ~600M	120
TABLE 30. LAUNCHXL-CC1350 SIMPLELINK DUAL-BAND PAIR PERFORMANCE SUMMARY.....	121
TABLE 31. BILL OF MATERIALS	126
TABLE 32. SPONSORSHIPS, DISCOUNTS, AND DONATIONS	127

1 Executive Summary

Advances in battery and motor technology combined with lightweight, high power computers and the challenges of navigating on land have led to a tremendous rise in the use of small Unmanned Aerial Systems (sUAS). Popularly known as “drones”, sUAS technology has opened many new applications for robotics to transport objects and search areas in a way that is accessible to a wide range of users. Using sUAS robots to transport objects has allowed for revolutionary new ways of rapidly delivering medical supplies, consumer goods, and even other robots. Using sUAS technology to search areas can improve the ability of rescuers to find people lost in the wilderness, law enforcement to find fugitives, researchers to find rare species, and so on.

AUVSI (Autonomous Unmanned Systems International) is a non-profit organization on a mission to advance unmanned systems and robots. As part of this mission, different branches of AUVSI hold a wide range of student mobile robotics competitions. These include RoboSub, RoboBoat, IGVC (Intelligent Ground Vehicle Competition), and SUAS (Student Unmanned Aerial Systems). RoboSub and RoboBoat are competitions which have students building unmanned, autonomous robots which navigate under and on the water respectively. IGVC is a competition where students build vehicles which navigate a land-based obstacle course. SUAS has students build autonomous sUAS which can complete several tasks in a large outdoor venue. The competition tasks are designed to simulate a sUAS mission which is similar to a rescue or security mission, where the sUAS must fly a set of waypoints to get to the mission area, drop off a UGV payload, then search for a set of objects on the ground before returning to land.

The Robotics Club at UCF is one of UCF’s registered student organizations, and is dedicated to building autonomous robots, most of which compete in AUVSI competitions. This year, the Robotics Club is building a sUAS which will enter the AUVSI SUAS competition. The electronics for the sUAS have requirements which make their development challenging, and a good fit for a senior design team. The electronics will have to provide power to the drone’s flight and logic systems, which have widely different requirements for what kinds of power they require. The logic systems on the sUAS must communicate with the ground station in order to meet the requirements, which requires a wireless communication system which can send the data over a long distance while meeting the throughput requirements of the drone. Additionally, an autonomous UGV must be attached to the drone, which needs power and logic systems of its own.

This document discusses the design, development, and integration of major electronic components that will compose the UAV and auxiliary components of the UGV and ground control system. Research regarding component options and various technology considerations are included for reference.

2 Project Description

The purpose of our project is to develop the electronics required for a relatively large sUAS suitable for entry into the AUVSI SUAS competition. The competition has demanding requirements on the range and payload capacity of a sUAS, which translates to a challenging electronics system. The mechanical components of the drone are being developed by an MAE senior design team, while the software is being developed by the Robotics Club.

The SUAS competition consists of three main stages: waypoints, payload drop, and search. The waypoints stage has the sUAS fly through a series of waypoints with the payload, for approximately 4 miles. The payload drop has the sUAS drop a UGV (Unmanned Ground Vehicle) onto a GPS waypoint. The UGV must carry an 8-ounce water bottle to another GPS waypoint after landing and must weigh less than 48 ounces with payload and landing system. The search stage has the sUAS fly over a 0.125 square mile area, where it must locate a series of ground targets that consist of a colored alphanumeric character on top of a colored shape.

During all 3 phases, the sUAS must contend with a random set of no-fly zones that are uploaded to it from the ground station. The no-fly zones are virtual volumes that the sUAS must navigate around to avoid point losses.

The challenges in the competition closely mirror real world UAS tasks, such as delivering packages to a consumer or finding a lost person in the wilderness, so the systems designed could be applied to these tasks in the future. The competition allows for a large-scale test of these systems that would be difficult to achieve outside it, and provides clear goals and restrictions based on AUVSI expertise.

2.1 Objectives

Our objective is to design and create the electrical systems for the competition entry from the Robotics Club. Our project to do this has 3 major areas of scope: Autonomous Flight, Payload Delivery, and Area Search. The Autonomous Flight area encompasses the power and control systems needed to allow the drone to maintain controlled flight between GPS waypoints, stay powered throughout the entire mission, and communicate with the ground station. The Payload Delivery Area encompasses communication with the UGV and the systems for deploying the UGV from the sUAS. The Area Search encompasses the image acquisition and processing hardware needed to search an area efficiently and quickly. **Table 1.** ECE contributions to the sUAS outlines the responsibilities and contributions that will be made by the ECE team.

Table 1. ECE contributions to the sUAS

Autonomous Flight	Payload Delivery	Area Search
Battery System capable of powering flight motors and auxiliary systems	UGV controller capable of following waypoints	Camera system with gimbal
Battery Monitor reporting battery state information	Control of payload drop system	Integration of computer hardware capable of processing the images
Flight controller capable of following waypoints	Communication system on UGV	Communication system capable of relaying images from sUAS to GCS
Communication system capable of relaying telemetry from sUAS and UGV		

2.1.1 Autonomous Flight

To complete the mission with the given requirements and necessary components, the drone must be relatively large (roughly 40 to 50 pounds) with strong motors. The Robotics Club determined that an octocopter configuration is most desirable given the available components, and such a system will consume 2900 watts while hovering and 6000 watts while at full power. In addition to the motors, power must be supplied to: onboard cameras, a flight controller, onboard computer, speed controllers, a GPS module, RF microcontrollers and more. To meet this demand, the power system must supply enough power to the motors (which can be unregulated) and have voltage regulators with enough power capacity for the other systems.

The mission requires that the drone be able to communicate with a ground station. The communication with the ground station requires that the drone be able to send back telemetry on its position, speed, and status, while also sending back images taken from the cameras and taking commands from controllers.

2.1.2 Payload Delivery

To ensure the ground vehicle completes its mission and to aid development, the ground vehicle must be able to relay important telemetry back to the ground station. The vehicle should be able to maintain communication even if it lands in an area not in line of sight. Our team intends to develop a communication system

for the UGV that can switch between a direct connection and using the drone as a relay. We also intend to develop the electronics required to control the deployment of the UGV.

2.1.3 Area Search

The area search task requires a high-performance image acquisition and processing system. Our group intends to integrate the cameras with a gimbal that can maintain them at an appropriate pointing direction during flight, and with computer hardware that can process the image data in real time. The computer hardware should be selected and integrated well enough to provide a good platform for the Robotics Club CS students to run computer vision algorithms on.

2.2 Requirements Specifications

A wide range of system requirements has been developed in accordance with the various goals, objectives, and rules of AUVSI SUAS Competition rules (AUVSI Seafarer Chapter, 2019) and derived in coordination with other subgroups of the complete competition team.

2.2.1 UAV Flight and Payload Requirements

Table 2 outlines the electronic payload and flight subsystems that will comprise the sUAS. The sUAS payload will consist of an Unmanned Ground Vehicle that will be deployed mid-flight and sent on an independent mission. The drone flight systems are the control and power systems responsible for keeping the drone in the air and under control and will be divided into control and power. Each subsystem is associated with several requirements that must be fulfilled for successful project completion.

Table 2. Drone payload and flight system requirements

Subsystem	Requirements
UGV Payload Systems	<ul style="list-style-type: none"> • Carry a UGV payload up to 3lbs • Follow a waypoint sequence of up to 4 miles • Drop UGV onto a drop zone GPS waypoint • Have enough battery for remaining mission tasks
UGV Payload System Electronics	<ul style="list-style-type: none"> • Apply force to hold payload onto drone during flight • Release the payload at appropriate time • Ensure the payload is undamaged upon landing • Maintain telemetry link between the payload and the ground control station • Maintain command link between ground control station and payload
Flight Controller	<ul style="list-style-type: none"> • Maintain the drone in stable flight • Receive GPS waypoint from main computer • Navigate the drone towards a GPS waypoint • Accept manual override commands directly from suitable radio • Report information via telemetry system • Land safely at a designated point after mission completion
Power System	<ul style="list-style-type: none"> • Supply 6000 peak watts to flight motors at an unregulated 22 volts • Supply 25 watts continuous to the main computer at a protected 12 volts • Supply 20 watts continuous to the flight controller and auxiliary systems at a regulated 5 volts • Supply 8 watts continuous to the high-speed wireless system at 24 volts • Store enough energy to allow the UAS to complete all mission tasks • Monitor battery, current, voltage, charge, and temperature • Report status information to the flight controller or main computer

2.2.2 UAV Image Processing & Communication Requirements

Table 3 outlines several electronic image processing and communication subsystems that will comprise the UAV aka drone. Each subsystem is associated with several requirements that must be fulfilled for successful project completion.

Table 3. *Drone image processing and communication subsystem requirements*

Subsystem	Requirements
Camera System	<ul style="list-style-type: none"> • Image wide area while maintaining a pixel ground size of 2cm per pixel • Maintain pixel ground size for allowable flight altitude
Camera Gimbal System	<ul style="list-style-type: none"> • Keep camera system pointed straight down (within 0.1 degrees from normal) • Maintain position under normal flight conditions • Point cameras off axis if search area is obstructed by no-fly zones
Computer System	<ul style="list-style-type: none"> • Process camera data in real time • Extract target objects in the image
High-Speed Communication System	<ul style="list-style-type: none"> • Send 640x480 images back to the ground station • Send images at a frequency of at least 30 Hz • Transport IP packets • Maintain a reliable connection throughout flight area
Telemetry System	<ul style="list-style-type: none"> • Report state information from flight controller to main computer • Report GPS coordinates at a rate of at least 1 Hz • Maintain reliable link at long range

Auxiliary System Requirements

Table 4. UGV and GCS system requirements outlines the electronic auxiliary systems that will be required to support mission objectives. Each subsystem is associated with several requirements that must be fulfilled for successful project completion.

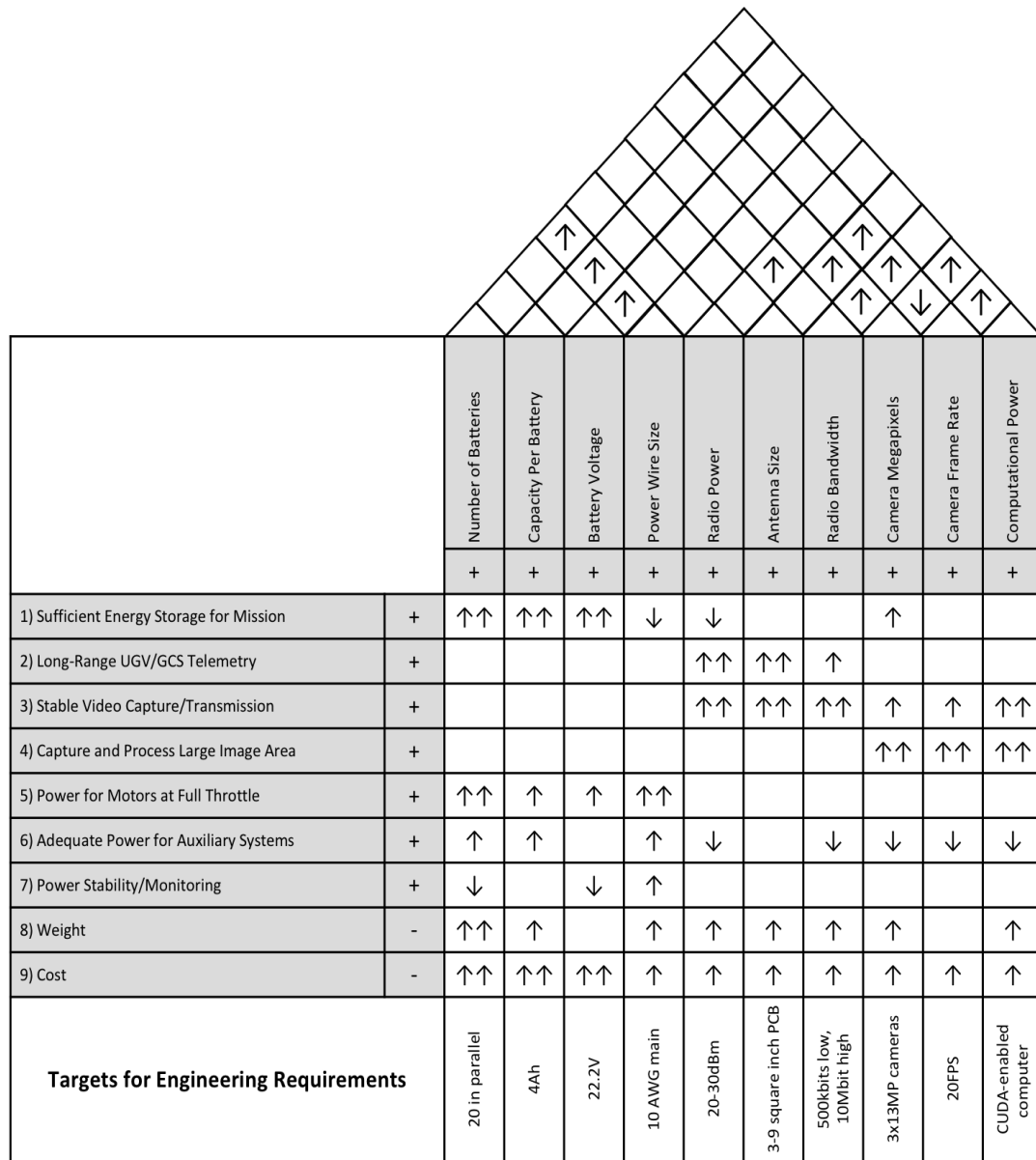
Table 4. UGV and GCS system requirements

Subsystem	Requirements
Unmanned Ground Vehicle (UGV/Payload)	<ul style="list-style-type: none"> • Obtain live GPS position • Determine current vehicle status • Transmit live GPS position and status to the GCS
Ground Control Station (GCS)	<ul style="list-style-type: none"> • Receive data from UAS • Receive data from UGV • Transmit commands to UAS • Transmit commands to UGV • Interface with at least one laptop computer

2.3 House of Quality

Figure 1 provides a house of quality analysis to outline the various relationships between customer requirements and engineering technical specifications that have been derived from the requirements in Requirements Specifications. The house of quality lists customer requirements on the left column, and engineering requirements on the top row, with a + or a – to indicate if something should be maximized or minimized. The arrows indicate if there is a positive or negative correlation between two requirements. A double arrow is used to indicate a stronger correlation. The house of quality also lists engineering requirements on the bottom based on the relations between engineering requirements, customer requirements, and other engineering requirements.

Figure 1. House of quality analysis



2.4 Existing Similar Projects and Products

The AUVSI SUAS competition has been running yearly since 2002, and recent years have had teams publish journal papers with an overview of their designs. These give substantial insight into the state of the art, however the 2019 competition has made large changes, namely in making the payload much bigger and more complex, as well as making the required flight distance much longer. As such the sUAS needs to be much higher performance to meet the objectives.

Another similar project that guided the rover drop system was a project to lower a rover from a drone for volcano research conducted by Tohoku University. They used an Arducopter-piloted hexacopter to lower a 2.5kg rover on a cable. The hexacopter itself weighed roughly 4kg and used a brushless motor as a brake for a winch attached to the hexacopter. This showed that the approach for the rover system was possible using parts available to us, especially considering that our rover is a much smaller fraction of the drone's weight than it was in this project.

2.4.1 Université de Sherbrooke

The Université de Sherbrooke scored the highest at the AUVSI sUAS competition in 2018 with a multi-rotor drone pictured in **Figure 2**. Université de Sherbrooke 2018 AUVSI sUAS drone. The rules for the competition change from year to year, so a drone suited for a competition one year may not be suitable for the competition the next year. In 2018 the drones were not required to fly as long distances and were not required to carry a UGV payload. As such, the drones last year were significantly smaller, however, some systems remain the same. The drone used an Ubiquiti Bullet M5 for transmitting video data and an RFD900 for transmitting GPS data. The drone used Ardu Pilot open source software as the auto-pilot and ran it on the PixHawk platform. These platforms are not directly related to distance and size and as such similar systems could be used in the future. The Université de Sherbrooke drone was relatively physically large with a total diameter of approximately 42 in, and 18 in rotors. The only area this drone is lacking is the battery capacity. The drone uses 32 6s batteries for an approximate flight time of 30 minutes and flight range of 4.35 miles.

Figure 2. Université de Sherbrooke 2018 AUVSI sUAS drone



2.5 Initial Design Architectures and Related Diagrams

The diagram in **Figure 3**. Diagram of electronics showing responsible personnel and status outlines the electrical systems proposed for development as part of the sUAS, the individual contributors of the team responsible for their implementation, and the status of each system. The onboard battery is composed of 20,700 NCR Lithium-ion cells arranged in 6 parallel groups of 20 batteries in parallel with each group connected in series with each other. This gives a battery which can supply up to 320A at 22.2V nominal with a capacity of 80Ah. The battery will have an attached battery monitor and battery protection system. The battery protection system shuts off the battery if there is a risk of explosion or fire caused by overvoltage or overheated cells, or if it detects a probable short circuit. The batteries might also be damaged by under voltage, but as this is less likely to cause a fire the first action will be to land the drone rather than cause a crash by cutting power. The battery monitor is a system which will measure the pack voltage, current, temperature, and charge. The battery monitor does this by connecting a fuel gauge chip to a shunt resistor between the battery and the rest of the system and communicates with the flight controller and computer through I2C and digital signals.

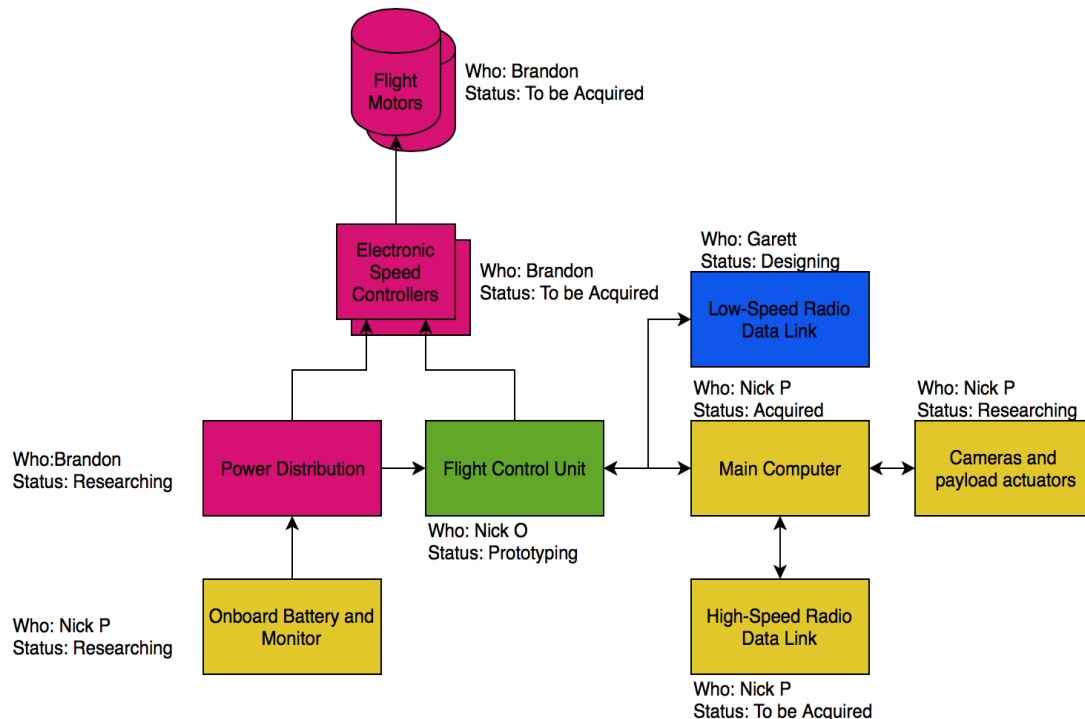
The flight control unit has sensors to monitor the sUAS's state and uses this information to drive the motors to maintain controlled flight. The flight controller runs open source firmware that allows it to be commanded by the main computer to hold position, follow waypoints, land, and perform other flight tasks. It communicates with the speed controllers to set motor output, and has accelerometers, gyroscopes, magnetometers, and a GPS receiver to determine the pose of the drone.

The flight control unit is connected to the main computer with a USB to UART converter, which allows the main computer to send it commands and retrieve information about the vehicle's state. The main computer is an NVIDIA Jetson TX2 on a carrier board connected to the FCU, cameras, and the high-speed data link radio. The TX2 was selected because the large volume of data from the cameras requires substantial processing power, and the CUDA cores on the TX2 are excellent at processing image data quickly. The cameras are a set of 3 4K USB 3.0 cameras with fixed focus lenses. The cameras and lenses were chosen to cover a large ground area while keeping the size of a pixel on the ground near 2 cm per pixel (based on the object size). The cameras must cover a very wide patch of ground so that the number of passes required during the search phase is minimal, which reduces the necessary range and flight time. To cover a large area with high resolution is what produces the high pixel count and data rate requirements. The TX2 also offers excellent Size, Weight, and Power characteristics, making it well suited for a UAS.

The high-speed radio data link consists of an Ubiquiti Rocket m5 modem and its antennas. This was chosen because the Robotics Club already has several units, and they are known to offer good performance in this sort of long range, line of sight point-to-point scenario. The high-speed radio data link allows the TX2 to have an IP network connection to the ground station, which is meant to send the results of the search task to the competition server automatically. Doing this requires sending back images, which requires substantial bandwidth. Extra bandwidth can be used for sending back extra status messages, commands, and images for vehicle development and debugging.

The low speed data link is planned to be a TI SimpleLink Sub-1GHz, which offers more range than the high-speed data link, to be used for critical telemetry and control. The low speed data link is also more lightweight, uses less power, and can communicate with its host over UART instead of Ethernet. This allows it to be used on the payload portion of the system for payload telemetry and control. Having a low speed data link on the drone as well as the ground station and rover opens the possibility of using the drone to relay communications between the two if the rover ends up in an area that is not conducive to a direct connection with the ground station, something not possible with the high-speed data link alone.

Figure 3. Diagram of electronics showing responsible personnel and status



2.6 Overall System Diagram

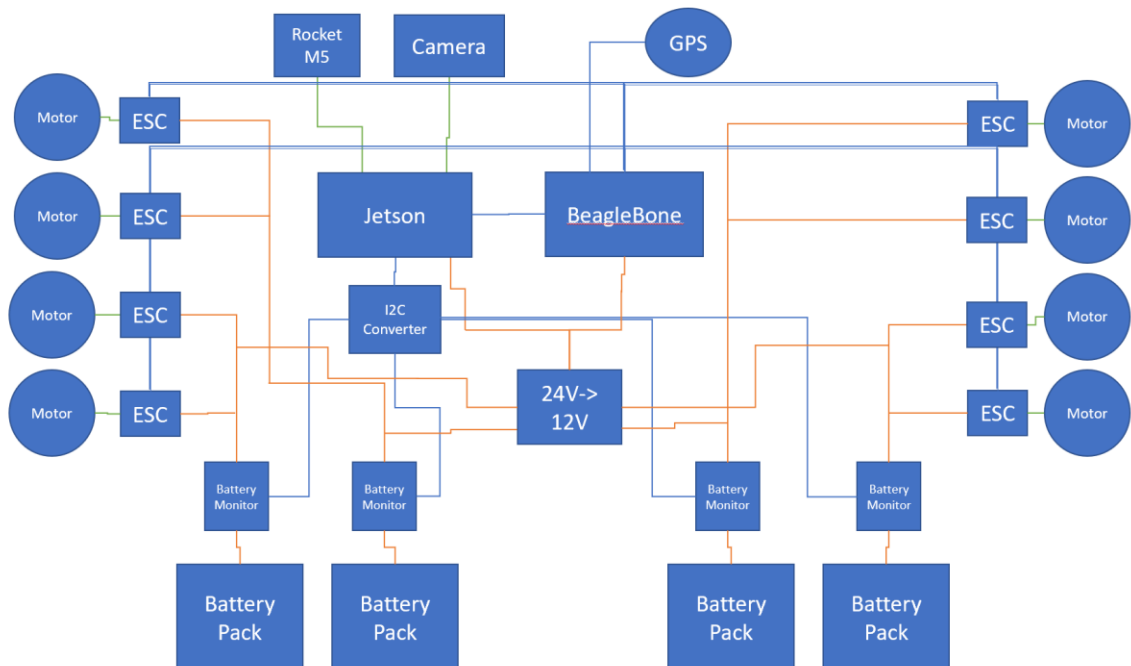
Figure 4. Overall System Diagram outlines the overall system of the drone. Power is shown in red for clarity, data is in blue, and power and data is in green. The

distribution of the power system is a bit unconventional. Each battery pack connects to two motor, on the physical drone these motors will be exactly opposite to each other. The reason for this is because when the drone turns, one motor will power up and the other motor will power down. Since the drone will be turning in all directions the battery packs should be drained relatively evenly.

The 24V to 12V buck converter drives the electronics and connects to all 4 battery packs. This board can select from any of the 4 battery packs and will draw from the one with the highest voltage to help reduce imbalances. Though it is not entirely clear in the diagram, each ESC has an individual connection to the BeagleBone Blue. The BeagleBone Blue can control up to 8 motors with PWM so only one Beagle is required to control the entire drone. The camera system will also have a gimbal which will be controlled over serial from the Jetson; it is not pictured for simplicity.

The NVIDIA Jetson has an ethernet interface to the Rocket M5, which gets power over ethernet, which will be injected on the regulator board. The Jetson also has an interface to the cameras over CSI, which supplies power from the Jetson's regulators.

Figure 4. Overall System Diagram



3 Related Standards and Realistic Design Constraints

Any engineering project will be subject to constraints on the design imposed by a variety of physical or legal constraints, and this project is no exception. Furthermore, additional constraints such as safety and environment constraints may be added to ensure a project is executed in a way that does not do more harm than is necessary. To help a project advance despite all the constraints on it, standards are available which can help ease integration, improve safety, simplify designing for environmental concerns and so on.

3.1 Standards

A few relevant standards are identified for the successful completion of the project. Each standard has been evaluated in terms of possible project implications and potential for use in streamlining the design process.

3.1.1 Communications Standards

Several relevant communications standards are identified for the successful completion of the project. Communication standards are crucial for developing a communication architecture and systems for transmitting telemetry, control, and media data between the ground control station, UAV, and UGV using off-the-shelf products and solutions.

- UART - a way of sending serial data between devices without a clock; this standard is useful for devices that would benefit from extensive cross-compatibility between various components systems
- SPI - a way of sending data between devices with a clock, with provisions for having multiple devices on a bus or in a change; this standard is useful in situations where high speed is required and if there can be a strictly predefined communication structure
- I2C - a protocol that defines a bus which may have a master and up to 127 slave devices, which requires only 2 I/O connections on each device; this standard is useful for wiring many devices to a single master with minimal hardware overhead (i.e. wires), especially if all devices will be used for output
- AES - Advanced Encryption Standard, defines an algorithm for strong symmetric encryption of data
- IEEE 802.11 – also known as Wi-Fi, IEEE 802.11 standard defines a set of physical layer and MAC (media access control) specifications required for the implementation of a computer WLAN (wireless local area network); this standard is useful for communications between ground control station systems and other computers or competition devices
- Bluetooth (Classic) – a common wireless communication standard often used to establish short range connections between devices in a personal

area network; Bluetooth is useful for connecting master devices such as computers and flight controllers to peripherals, but also for establishing point-to-point links between devices

- BLE (Bluetooth Low Energy) – a variant of the Bluetooth standard that uses the same frequencies, encryption, pairing, and authentication architectures and methods as Bluetooth Classic, but operates primarily in sleep mode; BLE is useful for connecting low-power sensors to devices such as computers or UAV flight controllers wirelessly

3.2 Realistic Design Constraints

A few realistic design constraints are identified for the successful completion of the project. Design constraints create potential challenges that may hinder attempts at fulfilling project requirements and objectives and must be considered when creating a design. Design constraints include but are not limited to those pertaining to:

- Finances and economics
- Time limitations
- Environmental impact considerations
- Social impact considerations
- Political impact considerations
- Ethical impacts and design considerations
- Health & safety considerations
- Design manufacturability and project sustainability
- Competition Rules

These design constraints are discussed in further detail.

3.2.1 Economic and Time

- This project must be kept within the budget available from the Robotics Club
- This project must be built in less than 2 semesters

3.2.2 Environmental, Social, Political

- The FAA has restrictions in place on SUAS (small Unmanned Aerial Systems) that affect where and how we can fly, so we must take his into account when testing.
- Flying over natural environments runs the risk of any parts that fall may pollute the environment, so all components must be secured properly to the drone, and there must be some system to assist recovering the drone if it lands or crashes in an obscured location.

3.2.3 Ethical, Health, and Safety

- The rotors of a SUAS can be hazardous if they begin spinning while someone is working near them
 - The flight controller should be programmed to not start the rotors without entering an arming sequence that is difficult to enter accidentally
 - The flight controller should sound an audible alarm before starting the rotors
 - The flight battery should be kept disconnected when not necessary
 - Motor tests should have the drone strapped down or the propellers removed
- Lithium batteries can be hazardous if misused
 - Only charge at a rate that is known to be safe from the manufacturer's data
 - Only charge with a charger designed for the type of battery
 - Visually inspect the battery for damage before use
 - Ensure the battery is protected from crashes and cannot be struck by moving parts
 - Monitor the temperature of the battery during charging and discharging

3.2.4 Manufacturability and Sustainability

- The drone systems should be built using techniques available in the Innovation lab, Robotics Club lab, and Senior design lab. As such, things like high density BGA devices should be avoided.
- The electrical components must fit into a real drone, so they should be checked to be compatible with the mechanical systems. For example, PCBs should be designed in a way that allows them to be mounted to the drone, and the connectors to be inserted without interfering with another part.

3.2.5 Competition Rules

- The drone systems should meet the rules given by the AUVSI SUAS coordinators

3.3 Regulations

Relevant regulations are identified for the successful completion of the project. Project designs must take into consideration the various rules and regulations that will be under affect in the area of operation. The primary governing bodies of the regulations of interest include the Federal Aviation Administration (FAA) and the Federal Communications Commission (FCC). The relevant and associated regulations are discussed in further detail.

3.3.1 FAA Regulations

The FAA regulates all aircraft in the United States. The FAA has recently published guidelines for sUAS which regulate the drone which the proposed systems will be used for.

- The UAS must weigh less than 55 lbs
- The UAS must be flown in line-of-sight *
- The UAS must not be flown near other aircraft or people *
- The UAS must not be flown near airports or in controlled airspace without FAA permission
- The UAS may only be flown in daylight or civil twilight, at or below 400 feet *
- The UAS must not be flown above 100 miles per hour
- The UAS may only be flown in class G airspace

*These rules are subject to waiver, sourced from FAA.gov

These rules apply to hobbyist sUAS flyers which are those flying for non-commercial reasons. Since the drone will not be used to provide any service for monetary gain it can be flown under the hobbyist heading.

3.3.2 FCC Regulations

The Federal Communication Commission has the responsibility of regulating communications on the RF spectrum to make sure it remains usable, rather than allowing unregulated systems to broadcast however they want and produce interference that renders the spectrum unusable.

The FCC does this by reserving certain frequency bands for certain applications and limiting the maximum power. The maximum power is measured as Effective Isotropic Radiated Power (EIRP) which is detailed in the link budget discussion in section 3.1.2. The FCC currently assigns use bands in frequencies between 9kHz and 275GHz. For this project the bands of interest are the 902MHz – 928MHz band and the 5.15GHz – 5.825GHz band. These bands are open ISM bands which means they were originally intended for industrial, scientific and medical purposes, but have since been opened for use by unlicensed devices so long as certain regulations are met. For the 900MHz band the requirements are that the device output cannot exceed 30 dBm (1 Watt) and the EIRP cannot exceed 36 dBm. The rules for the 5 GHz band are similar, but the maximum EIRP varies based on certain conditions. These are summarized in a table from AIR802, which is shown in **Table 5**.

Table 5. FCC 5Ghz rules summarized. Permission requested from AIR802.

BAND	Frequency (GHz)	Permitted Use Location	Maximum Output Power	Maximum EIRP *2
UNII (Low)	5.15-5.25	Outdoor Access Point	1 Watt 30 dBm *1	125mw 21dBm
UNII (Low)	5.15-5.25	Indoor Access Point	1 Watt 30dBm *1	
UNII (Low)	5.15-5.25	Fixed Point to Point Access Points	1 Watt 30dBm *3	
UNII (Low)	5.15-5.25	Client Devices	250mW 24dBm *4	
UNII-2 (Middle)	5.25-5.35	Indoor or Outdoor	200mW 23dBm or 11 dBm + 10 log B, where B is the 26 dB emission bandwidth in megahertz *5	
UNII-2 Extended	5.47-5.725	Indoor or Outdoor	200mW 23dBm or 11 dBm + 10 log B, where B is the 26 dB emission bandwidth in megahertz *5	
UNII-3 (Upper)	5.725-5.825	Typical Outdoor	1 watt 30dBm *6	

(AIR802, 2018)

For this project the 5.725-5.825GHz allows for the least restricted output, which matches the band used by the high-speed radios under consideration. This range still has output power limitations, so the radios chosen will need variable output power to account for the directionality of whatever antenna is chosen.

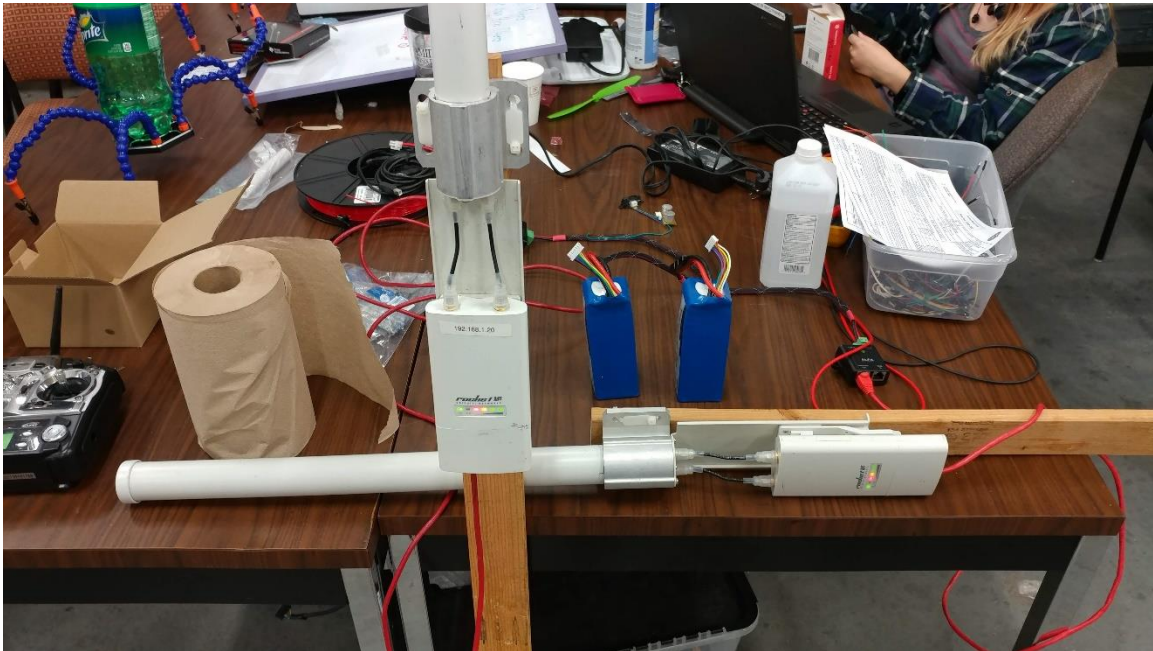
3.4 Received Components

The components are pictured in the form in which they were tested. Many of the components were available at the beginning of the semester thanks to the UCF Robotics club. In addition to the materials obtained specifically for this project, additional materials were also obtained to construct a test drone in order to test the Beagle Bone Blue as a flight controller and to gather pictures for image recognition software. The drone's original flight controller components are not pictured below, but the operational drone can be seen in section 6.3.

Table 6. Materials already obtained

Item	Value
NVIDIA Jetson Dev Kit	\$300
Ubiquiti Rocket M5 (2)	\$180
AMO-5G13 Antennas (2)	\$300
BeagleBone Blue (4)	\$375
CC1350 Modules (4)	\$120
Radio Controller	\$40
ESC's (6)	\$180
GPS (1)	\$90

Figure 5. Ubiquiti Rocket M5's and other miscellaneous materials



The Ubiquiti Rocket M5's along with the AMO-5G13 antennas are pictured above. They were attached to wooden rods for testing purposes. Behind those are two blue LiPo batteries which were used for power sources during testing the Rocket M5's. These batteries are being used for testing the modules since they are already the right voltage and are easily portable to the test location.

Figure 6 shows the electrical components for the test drone that replace the drone's original flight controller. A BeagleBone Blue was connected to the GPS and R/C receiver, then to a regulator. The black PCB functions as a power distribution board and as part of the drone's structure.

Figure 6. Electrical components for the test drone

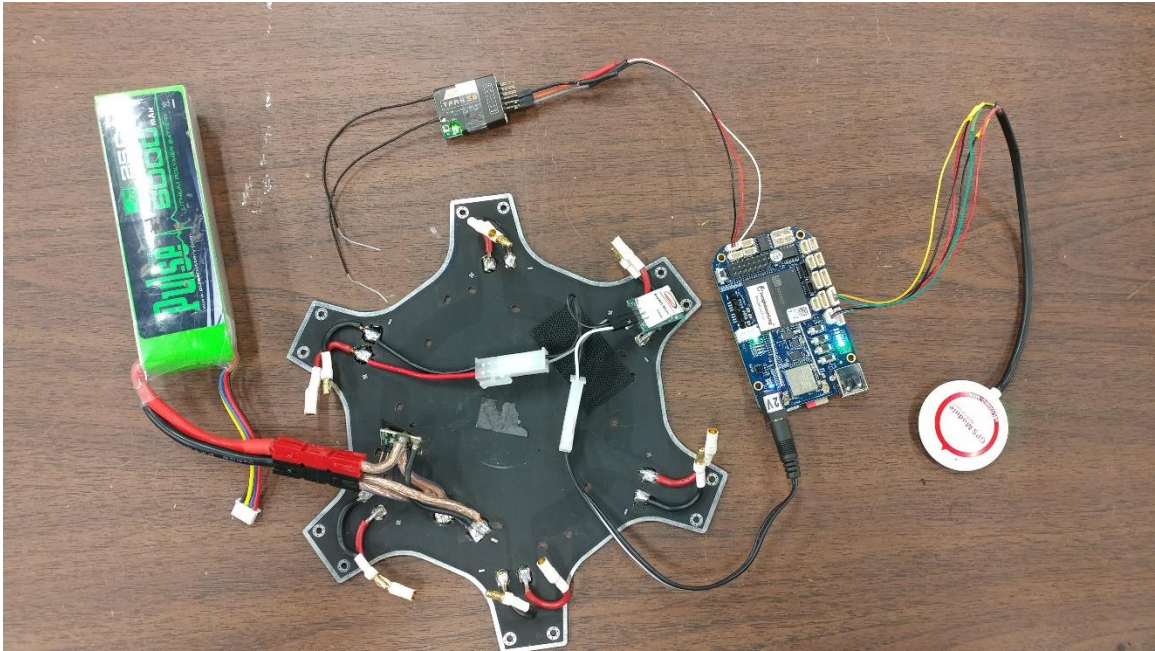
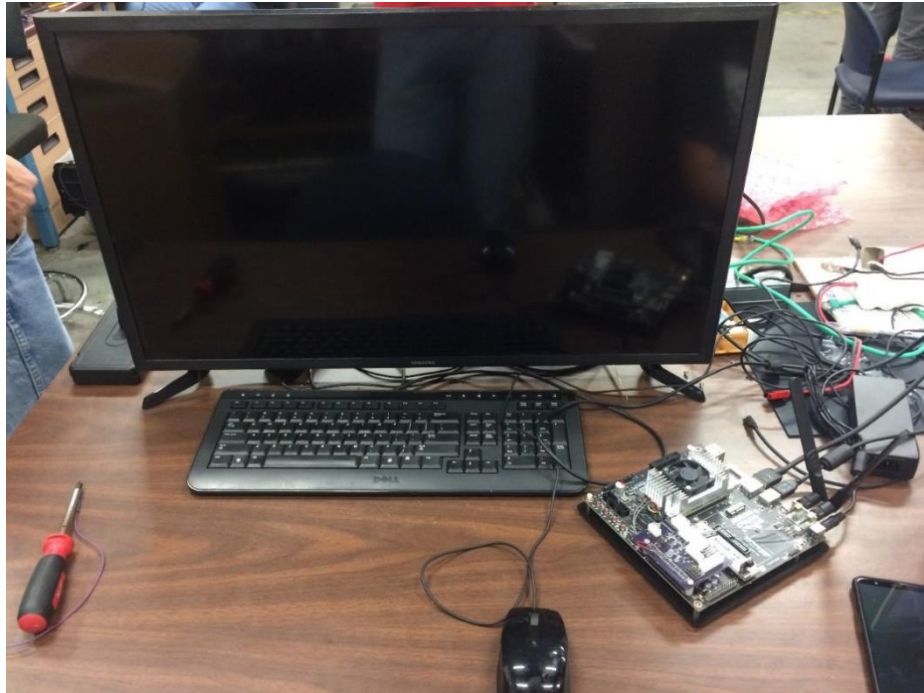


Figure 7 shows the test drone with the new electronics installed. One rotor is missing as the MAE team needed a brushless motor of that size to test part of the rover system.

Figure 7. Test Drone



Figure 8. NVIDIA Jetson Dev Kit connected for testing



Pictured in **Figure 8**, NVIDIA Jetson Dev Kit connected for testing is the Jetson TX2 on the TX2 development board. The Jetson is connected to a keyboard and monitor for programming and testing purposes. An I2C isolation PCB that was built for the Robotics Club’s RoboSub entry is attached and serves as a reference for future PCB designs with the Jetson Dev Kit.

Figure 9. NVIDIA Jetson close-up with I2C isolation PCB mounted



4 Background Research

This project is not done in a vacuum and depends on many existing systems and technologies to make it possible for a group of 4 students. Here we present information on technologies and techniques that are useful or necessary for our project.

4.1 Telemetry Platform

There are several telemetry system requirements and environmental conditions outlined by the AUVSI SUAS competition rules that call for a long-range, efficient, lightweight, and robust telemetry system (AUVSI Seafarer Chapter, 2019). Maintaining a reliable and consistent telemetry connection at distance is critical to meeting performance criterion and fulfilling the objectives necessary to acquire competition points. Communication between a ground station and UAS clearly implies the requirement of wireless communication. Multiple platforms are considered in this section to fulfill this requirement at a satisfactory level.

4.1.1 Context of Platform Requirements

As a prerequisite to multiple critical objectives, the ground station must receive telemetry data from the UAS at a rate of at least 1 Hz. Although the rate and amount of telemetry data that must be received is relatively minimal, even a temporary loss will result in significant point losses. Telemetry data dropouts will prevent the team from gaining points in the major categories of autonomous flight, obstacle avoidance, object detection, classification, and localization, and ground vehicle airdrop. For these categories, the UAS must be capable of sending information such as GPS coordinates, UAS and ground vehicle status, and object classification characteristics. GPS coordinates are required to assess the performance of the UAS in capturing and traveling along multiple waypoint locations, as well as determining whether the UAS remains within a predetermined set of boundary coordinates. In additional emphasis of the importance of maintaining a reliable connection, it should be noted that UAS is required to autonomously return home or land within 30 seconds of communication loss; the UAS is also required to immediately terminate flight after 3 minutes of communication loss. Additional information such as object characteristics will be transmitted as needed to aid in completing the primary objectives (AUVSI Seafarer Chapter, 2019). In general, the reliable transmission of alphanumeric characters and common symbols will be sufficient to communicate the required data.

In contention with the system requirements are several environmental conditions and competition restrictions that must be considered when selecting a telemetry platform. According to the AUVSI SUAS competition rules, no objects taller than 15 feet will be permitted, and the usable RF spectrum will not be managed (AUVSI Seafarer Chapter, 2019). The height requirement ensures that transceivers will remain relatively close to the ground. Although the competition will take place on

an airfield, line-of-sight ranges will not be guaranteed, and the system must be able to compensate for losses due to physical obstructions. Suitable transmission frequency, power, and antenna configurations must be selected with considerations made for the rule restrictions imposed on the ground station and the physical limitations implied by the flight requirements of the UAS.

The lack of RF spectrum management within the field implies an even greater number of variables to consider, especially when adhering to FCC regulations. Because the available RF spectrum will be free for use by any nearby devices, the selected telemetry platform must be resistant to EMI noise, RF signal interference, and similar signals sent by the systems of other teams. Data transmission must be structured to account for external factors and erroneous bits; techniques such as RF filtering frequency active/directional antenna systems, frequency hopping, and dynamic channel selection should also be considered. The AUVSI SUAS competition rules provide some additional notes to consider when selecting a suitable RF communication frequency, as outlined in **Table 7**. Allowed RF communication bands for AUVSI SUAS competition.

Table 7. Allowed RF communication bands for AUVSI SUAS competition

Frequency Band	Notes
72 MHz	Allowed for RC control; highly discouraged
433 MHz	Must use frequency hopping spread spectrum
462.7 MHz	Will be used by judge handheld radios
900 MHz	Allowed
1.08, 1.12, 1.16, 1.20, 1.28, 1.32, 1.36 GHz	Must use frequency hopping spread spectrum
1.2, 1.3 GHz	Analog or digital video systems only
2.4, 5.0 GHz	Allowed
Cellular	Allowed

To account for the possibility of dropped packets due to obstacles and wireless signal interference, a level of redundancy and increased transmission rate must be accounted for. This results in increased throughput requirements for the selected telemetry platform. Non-prohibitive power consumption, physical size, and cost is also a must, particularly when designing for the UAS and ground vehicle systems. Evaluation of each platform will mainly consider line-of-sight performance for comparison, as similar frequency bands and conditions will be utilized.

4.1.2 Overview of RF Sensitivity and Telemetry Techniques

An understanding of the minimum requirements for successful wireless communication is paramount to selecting a suitable telemetry platform. A balanced link budget provides an effective summary of all gains and losses that occur between transmitting and receiving devices (Ray, 2014). A link budget can be summarized as seen in **Equation 1**.

Equation 1. *Link budget*

$$\begin{aligned} \text{Received Power (dBm)} \\ = \text{Transmitted Power (dBm)} + \text{Gains (dB)} - \text{Losses (dB)} \end{aligned}$$

Received power must exceed the receiver sensitivity of the receiving device, often denoted in dBm, to produce an output signal above a signal-to-noise ratio (SNR) (Ray, 2014). This roughly correlate to an acceptable power level for communication to be viable. The dBm is an absolute unit of power, with 0 dBm denoting 1 milliwatt. 10 dBm equates to 10 milliwatts, and -10 dBm equates to 0.1 milliwatts. Gains and (mostly) losses due to propagation, attenuation, connectors, and antennas are also included in relative dB. Intuitively, it would be best to select components and configurations that provide high transmitted power and gains and minimize receiver sensitivity and losses (Ray, 2014).

To design a robust telemetry system, components should be chosen or modified to have adequate shielding and resistance to EMI noise. The other techniques must also be used to compensate for interference and insufficient SNR. To differentiate wanted signals from others, it would be useful to consider unique addressing schemes and encryption or handshake protocols that allow identification of sending sources. RF filtering, frequency hopping, and dynamic channel selection should ideally be built into the transceiver platform of choice, though the latter two features can technically be implemented using additional software. Both frequency hopping and dynamic channel selection would provide significant performance gains in situations where frequency channels of the desired communication frequency bands are being utilized to some capacity.

4.1.3 Texas Instruments SimpleLink CC13x0 (Sub-1 GHz)

Texas Instruments provides a variety of solutions for long-range RF communication. A summary of relevant and evaluated characteristics are included in **Table 8**. Summary CC13x0 family characteristics.

Table 8. Summary CC13x0 family characteristics

Specification	Value	Notes
Frequency Bands	433, 9xx MHz	Programmable
Max Transmit Power	15 dBm	Sub-1 GHz
Receiver Sensitivity	-110 dBm	SimpleLink Sub-1 GHz
LoS Range	200 meters	Field tested
Interface	MCU (UART, I2C, GPIO)	High flexibility due to availability of MCU

Texas Instruments offers a selection of multipurpose and easy-to-use wireless microcontrollers (MCU) built upon their SimpleLink platform (Texas Instruments, 2017). Portable code that can be reused between SimpleLink devices and the built-in support of multiple frequencies and protocols such as Bluetooth, Sub-1 GHz, Wi-Fi on a single device with MCU features are some of the major benefits. For the application of this project, support for Sub-1 GHz wireless communication is the main area of interest due to its superior range and sufficient throughput (in ideal conditions) (Texas Instruments, 2017).

The CC1310 (Texas Instruments, 2018) and CC1350 (Texas Instruments, 2018) and their respective LaunchPad development kits are the primary MCU's under consideration. Though the CC1350 is a newer version of the CC1310 with support for Bluetooth and Wi-Fi frequencies, the requirement for Sub-1 GHz range essentially makes the two devices synonymous for all intents and purposes. The performance characteristics for these two wireless MCU's is essentially identical in the Sub-1 GHz band. The CC13x0 family devices provides robust ultra-low-power functionality with built-in libraries and an almost all-in-one experience with MCU, transceivers, and antennas. The integrated MCU provides many options for expansion and inclusion of sensors and analog or digital input/output. The EasyLink API provides an additional abstraction layer above the CC13x0 family RF Drivers and to support rapid prototyping and development, which is ideal in this situation. Unfortunately, despite these positive characteristics and low cost, the stock performance of the CC13x0 family is not sufficient for this application. Without significant gain increases from amplification and the addition of external antennas, line-of-sight performance was insufficient past 200 meters as tested. Fortunately, Texas Instruments also has a selection of expansion kits and components available to improve the performance of CC13x0 devices including the CC-ANTENNA-DK2 (Texas Instruments) antenna kit and CC1190 850-950 MHz front end module (Texas Instruments, 2010). These additions, however, push the cost of the SimpleLink platform up to be more in line with other options.

The EasyLink API provided by Texas Instruments provides an easy way to configure and extend existing features provided by the CC13x0 family using a portable programming interface. Some RF compliance features such as Listen Before Talk and frequency hopping are not implemented by default but may be added with additional development. Another major benefit of the SimpleLink Platform and EasyLink API is the extensive support and documentation available from Texas Instruments resources (Texas Instruments, 2017).

The CC-ANTENNA-DK2 antenna kit is available from Texas Instruments to boost the wireless performance of the CC13x0 family MCU's. The antenna kit provides an assortment of preconfigured and easy-to-connect antennas that can be connected via the JSC (uSMA) connector available on the CC13x0 family with minor modifications to the MCU board. The variety of antennas allow for thorough field-testing of multiple configurations to determine the optimal antenna for a given application (Texas Instruments).

CC1190 850-950 MHz RF Front End is available from Texas Instruments to improve the transmission power and receiver sensitivity of CC13x0 family MCU's. The CC1190 acts as a range extender, which improves the weakest aspect of the CC13x0 family for this application. The CC1190 effectively increases the link budget of the system by integrating a power amplifier for augmented transmission output, and a low noise amplifier to improve receiver sensitivity (Texas Instruments, 2010).

4.1.4 Digi Xbee-PRO 900HP RF Module (Zigbee)

Digi International provides a variety of solutions for long-range RF communication. A summary of relevant and evaluated characteristics are included in **Table 9**.

Table 9. Summary of XBee-PRO 900HP characteristics

Specification	Value	Notes
Frequency Bands	900 MHz	Pre-certified
Max Transmit Power	24 dBm	Sub-1 GHz
Max Receiver Sensitivity	-101 dBm (200 Kbps) -110dBm (10 Kbps)	Depends on chosen variant
Advertised LoS Range	4 miles (200 Kbps) 9 miles (10 Kbps)	Outdoor/LoS Range
Interface	UART (3V), SPI, GPIO	

Digi advertises best-in-class range with the Digi XBee-PRO 900HP RF module (DIGI International). This embedded module is capable of mesh and multipoint topologies and advanced power-saving sleep modes with their proprietary software and networking protocol. The XBee-Pro 900HP features the DigiMesh networking protocol for connecting multiple nodes across a wide area. Line-of-sight range and data rates can vary depending on the configuration but can be up to 9 miles and 200 Kbps (though not at the same time). Range is of greatest priority, and even 10 Kbps of reliable data transmission would be sufficient for our purposes. With a transmission power of up to 24 dBm standard, the Digi XBee-PRO 900HP should have sufficient transmission capabilities out-of-the-box (DIGI International, 2018).

Like the Texas Instruments SimpleLink platform, the Digi XBee-PRO 900HP RF module provides wireless transmission capabilities over the 900 MHz frequency band. The data interface options for the Digi XBee-PRO 900HP include UART (3V), SPI, and GPIO pins for increased flexibility. A variant of the Digi XBee-PRO 900HP includes 32 KB flash and 2 KB RAM with a 50.33 MHz clock speed CPU for additional variability via programming if necessary (DIGI International, 2018). Supply and voltage levels are at 3 (3.3) volts, which allows for painless compatibility with the Nvidia Jetson. In addition, power consumption is minimal with little impact on system battery life.

The XBee-PRO 900HP offers many benefits in terms of ease-of-use, with minimal programming required and pre-certification for use in the USA. There is also a selection of channel masks for programmable immunity to various forms of interference (DIGI International, 2018).

4.1.5 Microchip RN2483(LoRa)

Microchip provides a relatively new solution for long-range RF communication based on LoRa (Long Range) and LoRaWAN (Long Range Wide Area Network) technology. A summary of relevant and evaluated characteristics are included in **Table 10**.

Table 10. Summary of Microchip RN2483 characteristics

Specification	Value	Notes
Frequency Bands	433, 868 MHz	868 MHz not in competition rules
Max Transmit Power	14 dBm	
Max Receiver Sensitivity	-148 dBm	Relatively low
Advertised LoS Range	-	Not available
Interface	UART, GPIO	

Microchip Technology offers the RN2483 LoRa Transceiver module as a wireless data transmission system that is low-power, long-range, easy-to-use, and fully-certified (Microchip Technology). The RN2483 utilizes the 433 MHz and 868 MHz frequency bands and communicates based on the Low Power Wide Area Network technology also known as LoRaWAN. Using spread spectrum modulation and sub-GHz frequencies, the RN2483 also allows for high network capacity in addition to long range and low power (Microchip Technology).

The RN2483 is available for public and private deployment and include LoRaWAN Class A systems for painless connectivity to various networks, given that they are LoRaWAN-compliant. An ASCII command interface over UART provides for a simple and relatively universal communication interface for simply connectivity and shortened development time. GPIO is included for control, ADC, and status communication, and a MCU with crystal and front end is included (Microchip Technology).

Though the specifications advertise a 14dBm output power rating, the advertised input sensitivity of the module is much greater at -148 dBm (Microchip Technology). A higher power output transmission is desired, but the higher receiver sensitivity may be sufficient to compensate in real-world applications.

4.1.6 RFD900 (Sub-1 GHz)

The RFD900 is a very popular solution for long-range UAV telemetry, and benefits from an extensive track record in the field. A summary of relevant and evaluated characteristics are included in **Table 11**. Summary of RFD900 characteristics.

Table 11. Summary of RFD900 characteristics

Specification	Value	Notes
Frequency Bands	902-928 MHz	
Max Transmit Power	30 dBm	Relatively high output power (1 watt)
Max Receiver Sensitivity	-121 dBm	Good sensitivity
Advertised LoS Range	25 miles	Much higher than other options; more depending on antenna and GCS setup
Interface	TTL UART, GPIO	

The RFD900 series is a widely used and popular telemetry module used in UAV and UAG systems. New variants of the RFD900 are available, such as the RFD900+, but the core attributes remain the same. The RFD900 achieves an

extremely high line-of-sight range of 25 miles, especially when compared to other options. Though somewhat higher in cost, the higher output power and good sensitivity ratings allow for significantly greater wireless communication performance at range. The additional overhead above functional requirements is beneficial in compensating for suboptimal conditions (PX4 Dev Team, 2018).

The RFD900 can also be purchased as a package with antennas and connectors to accelerate development, testing, and prototyping out of the box. The RFD900 includes a transmit low pass filter, low noise amplifier, and built in ESD filtering and protection while remaining relatively compact. Power consumption is higher than other options due to higher transmission power (30 dBm) but is still within reasonable allowances for the project. Multiple UART and over-the-air data transfer rates are available and user selectable via programming (PX4 Dev Team, 2018). All available options should be sufficient for the required telemetry.

The software solution commonly used for the RFD900 is an open source development with main contributors Mike Smith, Andrew Tridgell, and RFDesign called "SiK". A wide variety of parameters are configurable via the software and AT commands such as frequency band, baud rate (for both the UART and over the air), and power levels. SiK also includes features such as frequency hopping spread spectrum (FHSS), point-to-point or multipoint networking, error correction, status reporting, and duty cycle throttling for heat management (PX4 Dev Team, 2018). Many of these features are highly beneficial to compensate for noisy environments and interference that may be present during live operation. The RFD overall makes for a highly robust and well-integrated solution.

4.2 High-Speed Wireless Communication

The intended use for the high-speed wireless communication system is video transmission. The video transmission platform is not required per competition requirements, but it is suggested and allows for easier evaluation of in-flight performance. Streaming video requires a significant amount of bandwidth and consequently a higher transmitting frequency than the telemetry system. Some considerations and solutions for the high-speed system are given in this section.

4.2.1 Device Considerations

The devices chosen to perform the high-speed communication have a lot of realistic design constraints. Due to the high-speed nature of the link a high frequency must be chosen. In the US, the only GHz bands open are the 2.4GHz and 5.8GHz bands. The 2.4GHz band has been in use for quite some time and is rather crowded. Many devices operate in the 2.4GHz band including wi-fi and most remote controllers. Since this aircraft will be flying at an airfield with a large amount of communication, the 2.4GHz band is likely to be too crowded to offer the necessary bandwidth which only leaves the 5.8GHz band. Furthermore, the device should ideally operate with MIMO. Under ideal conditions MIMO doubles the bandwidth and transmission speed but it also provides reliability (as noted below).

Finally, the device needs to have enough output power to be used with a relatively low gain antenna to be able to transmit the required distance. The device ideally allows for external antennas so that antenna gain and size can be optimized. The requirements of 5.8GHz, MIMO, high power and external antennas significantly reduces the devices available. Most of these devices are commercially available for point to point internet access, and the only main player in this arena is Ubiquity. As such, only Ubiquiti devices were considered.

4.2.1.1 Rocket M5

The Ubiquiti Rocket M5 was originally created to be used for point to point wireless networking. These devices could be used to set up a high-speed datalink over many miles. Due to the origin of the device it is not very small or light and has a relatively high-power consumption. However, because the device is meant for a high-speed datalink it has the power, frequency and bandwidth necessary for the video streaming desired. Like the other Ubiquiti Airmax products, the Rocket M5 has R-SMA connectors to allow for external antennas. This allows for antenna optimization as discussed in section 4.3.2.

4.2.1.2 Rocket 5AC Lite

The Rocket 5AC Lite is essentially a new revision of the Rocket M5 and it is better in every way. The most notable improvement between the Rocket M5 and the Rocket 5AC is the bandwidth. The rocket M5 can go up to a 40MHz bandwidth when set to auto ranging mode between 20-40MHz. The 5AC bumps that up to 60MHz which Ubiquity claims leads to a significant increase in throughput. Some other lesser features include weight and size. The 5AC is a little bit lighter and a little bit smaller than the M5. The only drawback is that the 5AC can potentially draw another 0.5W which it reaches its max power draw.

4.2.1.3 Bullet IP67

The Bullet IP67, like the Rocket 5AC Lite is a newer version of an older device. The Bullet IP67 replaces the Bullet M5 which was a compact transceiver that operated in the 5.8GHz band. Unlike the Bullet M5, the Bullet IP67 can transmit and receive in both the 5.8GHz and 2.4GHz band, though not simultaneously. The Bullet IP67 is a physically smaller device than the Rockets which is why it weighs less, but it also gives up features like MIMO and a higher transmit power.

4.2.1.4 Comparison Chart

The **Table 12** provides a comparison of the above devices. From the table the Rocket 5AC Lite has the highest throughput due to the higher bandwidth and MIMO (Ubiquiti claims up to 300 Mbps) (Ubiquiti, 2014). The Rocket 5AC is also lighter than the Rocket M5 even though it is a more powerful rocket with more features. The Bullet is of course the lightest of the three devices and physically the smallest which is a benefit on a drone where every gram counts. However, the drone is relatively large, and the weight savings is not significant enough to merit

the loss in features. That essentially narrow it down to the Rocket M5 and Rocket 5AC Lite. Of course, the Rocket 5AC Lite is better in every way than the Rocket M5 since it is essentially just a newer revision of the M5. Though the 5AC is better in every technical way it is lacking in two major categories; cost and time. The Robotics Club already had access to multiple Rocket M5s which allowed testing to start immediately and helps keep costs low. For this reason, the Rocket M5 was chosen for this project, it sufficiently meets all the requirements and saves both time and money.

Table 12. Comparison of Ubiquiti Airmax products

	Rocket M5	Rocket 5AC Lite	Bullet IP67
Weight (g)	272	250	213
Max Power Consumption (W)	8	8.5	7
Max Receive Sensitivity (dBm)	-96	-96	-93
Max Output Power (dBm)	27	27	21
Max Bandwidth	40MHz	60MHz	40MHz
MIMO	Yes	Yes	No
External Antenna Connectors	Yes	Yes	Yes
Cost (\$)	Donated	135	129

(Ubiquiti, 2011) (Ubiquiti, 2014) (Ubiquiti, 2014)

All the above devices have connectors for external antennas. Both Rockets are MIMO enabled, the Bullet is not.

4.3 Wireless Communication Considerations

The wireless communications system is a large part of the project, and has several aspects which must be considered carefully to produce a well-engineered design.

4.3.1 Power Considerations

There is a relatively simple tradeoff between power and range. When power is increased range increases. This relationship is described in **Equation 2**, with symbol definitions found in **Table 13**.

Equation 2. Approximate equation to calculate the receive power over a distance

$$\text{Receive Power (dBm)} = TP + TAG + RAG - 20\log(\lambda/4\pi R) - M$$

Table 13. Receive power symbol definitions

Symbol	Definition
TP	Transmit Power (dBm)
TAG	Transmitter Antenna Gain (dBi)
RAG	Receiver Antenna Gain (dBi)
λ	Wavelength (m)
R	Range (km)
M	Margin (dBm)

Note that there is a built-in margin in the above equation. The margin is to account for losses not due to free space. If a perfect line of sight (with no Fresnel Zone infractions) maintained, this term can be 0. Practically this is unreasonable, so a typical margin is about 12 dBm. From this equation you can see that the larger the transmit power, the larger the range. However, due to FCC regulated EIRP limits and practical limitations power cannot be increased indefinitely until the desired range is reached. As a result, power must be maximized, but other things like noise and antenna directionality need to be considered to get the best link possible.

4.3.2 Antenna Considerations

With all RF communication platforms antenna design is a major consideration. The antenna gain factor is measured in Decibels Isotropic (dBi) which is essentially the increase in sensitivity of an antenna over a lossless perfectly isotropic antenna. Depending on the design some antennas receive with a higher sensitivity in certain directions. For example, an ideal Omni directional antenna will receive equally well across 360 degrees on a two-dimensional plane. Because the antenna is sensitive to signals in a certain plane it can have a gain over a theoretical isotropic antenna which receives equally well in all planes.

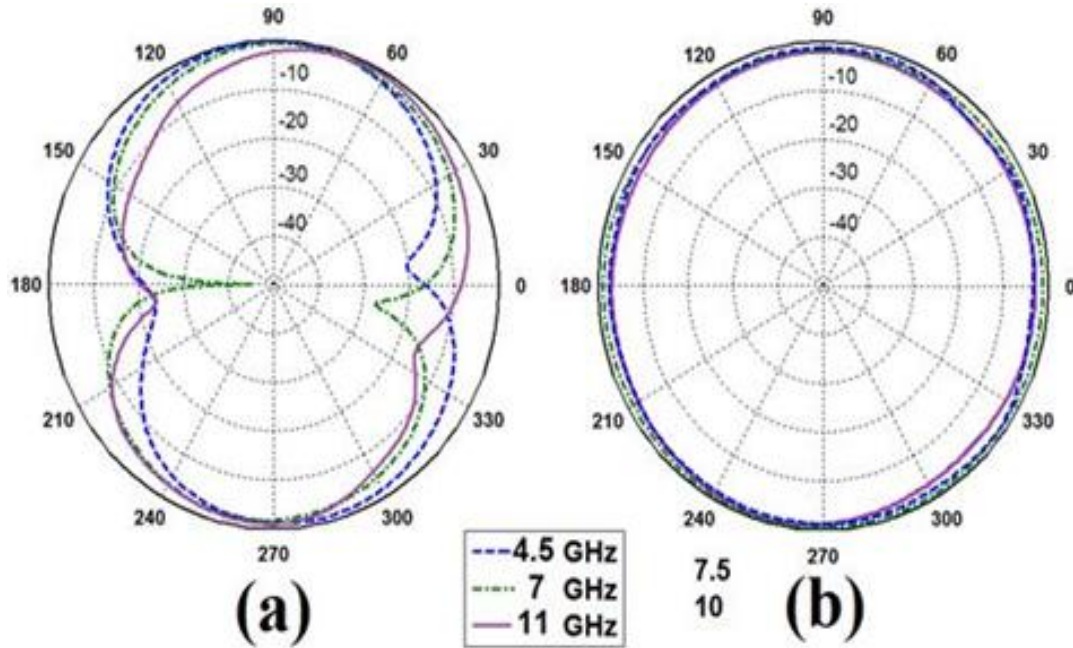
4.3.2.1 Monopole

A monopole antenna is the simplest of antenna, consisting of a single conductive element. The length of the element can be varied to achieve different radiation patterns. The monopole typically radiates close to uniformly in the azimuthal plane and has a relatively wide lobe in the elevation plane. This make the mono-pole antenna good for sending and receiving signals from most directions. The monopole antenna is also very easy to make because any conductive element can be turned into a monopole antenna.

However, because the monopole radiates well in many directions, power is radiated over a larger area and the antenna has a poor gain. Typically, on the order of 2-5dBi. The monopole is also a linear polarizer. On the upside this allows for easy use of MIMO, but it can also make the signal more susceptible to losses. Because the signal is just in one plane, if an object is coincident with that plane the signal will be blocked. Monopole antenna also have the added benefit and detriment of being tunable to many frequencies. This allows monopole antenna to

receive a wider range of frequencies but makes them more susceptible to noise and cross interference. It is also important to note that the number of lobes and the shape of the lobes in the elevation plane can be manipulated by changing the length of the monopole with respect to the wavelength of the received signal.

Figure 10. Typical radiation patterns for a Monopole antenna

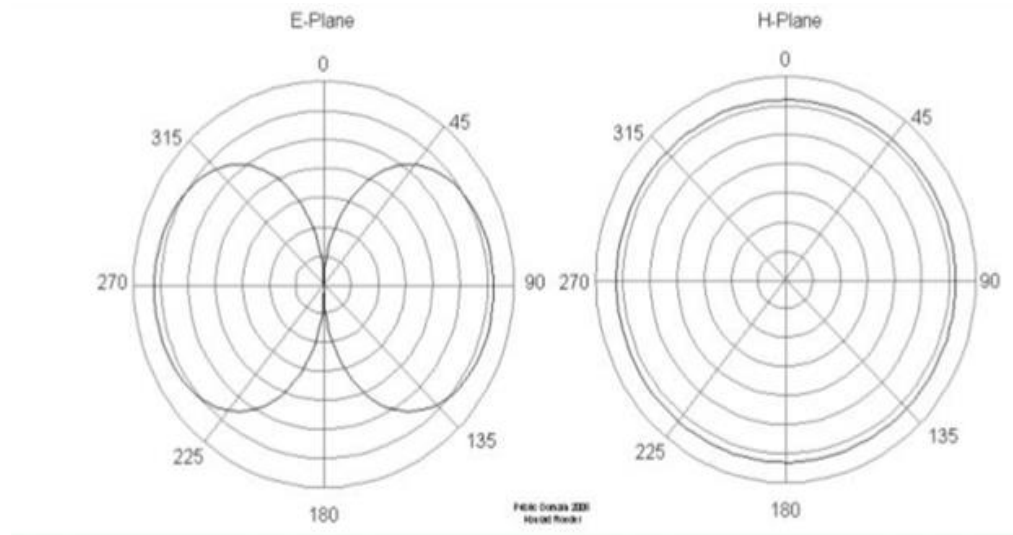


It would be difficult to design a monopole antenna to the requirements of this project. The low gain of a mono-pole antenna makes it less than ideal for long range, high power transmission. The lack of directionality makes the mono-pole susceptible to noise which lowers the potential data rate over long ranges.

4.3.2.2 Dipole

The dipole antenna is like the monopole antenna in many ways. The only real difference between the monopole and dipole is that the dipole has two linear elements instead of one. As expected the dipole is close to omnidirectional in the azimuthal plane (labeled H-plane in the figure below), but it has slightly larger lobes in the elevation plane. This means that a dipole antenna can potentially receive from greater angles than the monopole. The gain of a dipole antenna varies based on the length of the poles relative to the wavelength. For short antennas the gain is somewhere around 1.5 dBi all the way up to 5.2 dBi for an antenna that has poles 1.25 times the wavelength. The standard dipole is a half wave dipole which has a gain of 2.15.

Figure 11. Typical radiation patterns for a dipole antenna



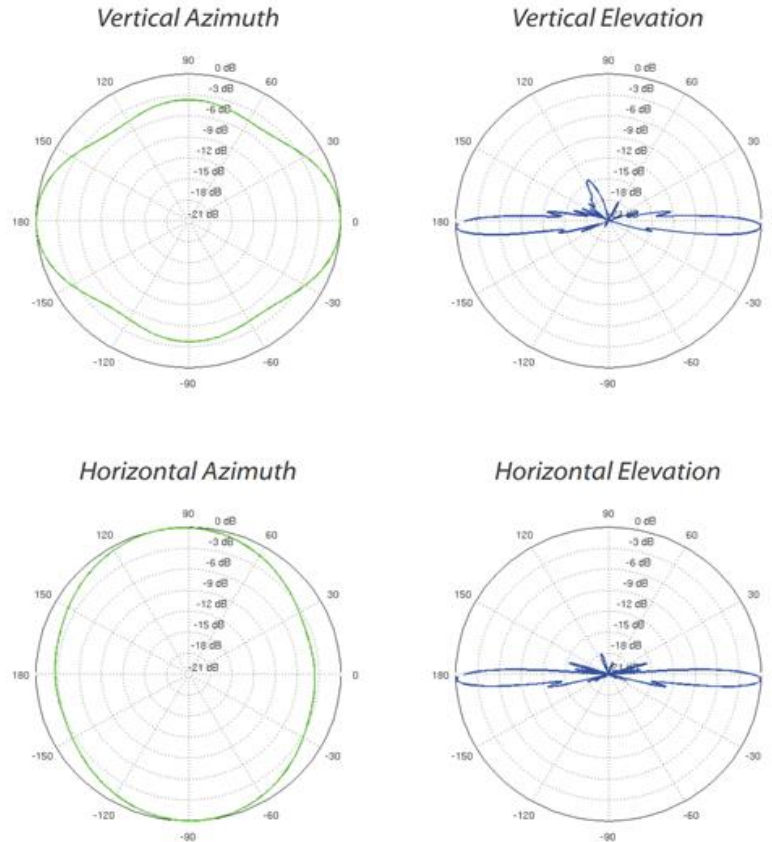
The dipole antenna also emits a linearly polarized signal, and as such is subject to similar limitations of the monopole antenna with respect to object interference. The fact that the dipole can receive from so many angles is beneficial in some ways because that means that if this antenna was mounted on a drone, the drone would be able to receive in most positions. The noise in a dipole antenna would be relatively high due to the wide receiving angles and gain is somewhat low. However, when paired with a proper ground antenna the dipole antenna could potentially be suitable for this project.

4.3.2.3 Omnidirectional

An ideal omnidirectional antenna transmits equally well in all directions in the azimuthal plane, and only the in azimuthal plane. Meaning that these antennas have extremely narrow main lobes in the elevation plane at 90 and 270 degrees or 0 and 180 degrees depending whether the antenna is transmitting in the vertical or horizontal plane. Practically the lobes in the elevation plane have some width and most omnidirectional antennas have multiple minor lobes as well meaning that these antennas do transmit at other elevations though not as well.

Due to the high directivity in the elevation plane, omni directional antennas typically have relatively high gains on the order of 5-15 dBi. Omnidirectional antennas are usually linearly polarized which makes them subject to object interference as noted earlier.

Figure 12. Radiation patterns for an AMO-5G13 from Ubiquiti

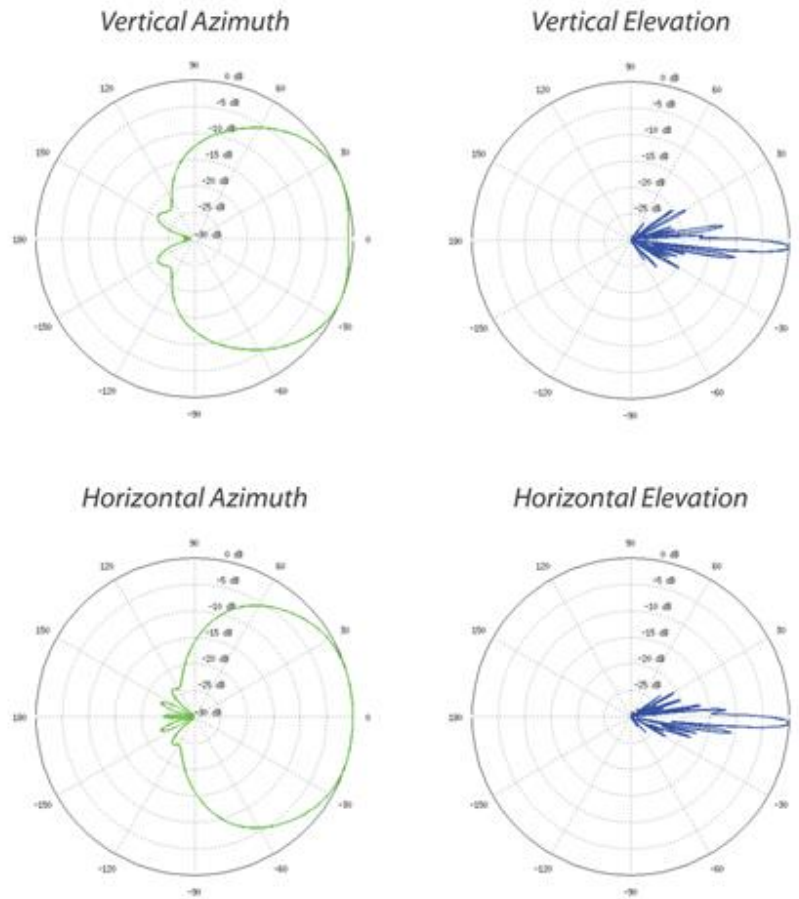


Omnidirectional antennas could potentially be used for this project with certain considerations. The high gain of omnidirectional antennas makes them more ideal for transmitting over long distances. And the lack of directionality allows for movement of the drone in the horizontal plane. However, if omnidirectional antennas are used both on the ground and on the drone, the drone may lose connection at certain altitudes due to the low transmission angles in the elevation plane. This can be compensated for by using a ground antenna that has a large elevation angle.

4.3.2.4 Sector

Sector antennas are the opposite of omnidirectional antennas, instead of transmitting equally well in all directions in the azimuthal plane, they transmit well only through a certain angle. This allows sector antennas to have a higher gain than their omnidirectional counterparts, typically around 15-20 dBi. Because sector antennas radiate through a relatively small angle on the azimuthal plane, they can potentially have larger angles of elevation without sacrificing too much gain. Like all the other antennas discussed up to this point, sector antennas are typically linearly polarized.

Figure 13. Radiation patterns for the AM-5G-19-120 by Ubiquiti

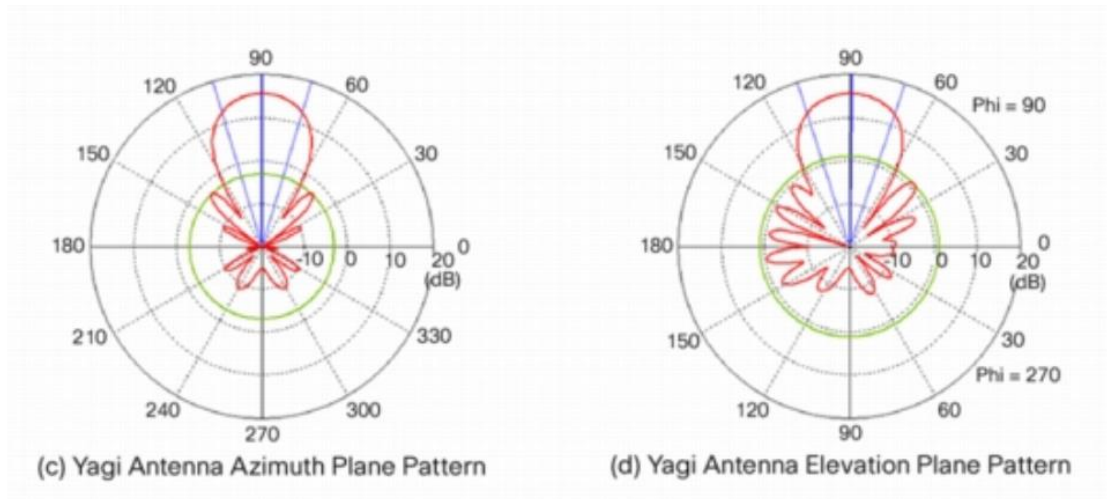


A sector antenna could potentially be a good choice for a ground antenna. The directivity leads to a high gain which is good for long distance transmission, and the directivity will also help reduce noise since the antenna will only receive from the direction of the transmission. Of course, this also presents a problem, if the drone is not flying in the 120-degree angle that the antenna transmits and receives in, connection will be lost. One possible solution is to use an antenna tracker to rotate the antenna and keep it properly oriented. The other choice is to connect multiple antennas to multiple devices in an array to cover all 360 degrees. It is important to note that an antenna with a reasonable elevation angle would have to be chosen to avoid the downfalls mentioned in the Omni-directional section.

4.3.2.5 Yagi

The Yagi antenna can be thought of as the linearly polarized equivalent to a helical antenna. Looking at the radiation patterns it is apparent that the Yagi is relatively directional, casting its radiation in a cone like structure emanating from the end of the antenna. Because a Yagi antenna is an array of elements the spacing and number of elements can be varied to greatly change the radius of the cone to increase or decrease the directionality. The gain of a Yagi antenna varies greatly from 10-30 dBi, again, depending on the number of elements and the spacing.

Figure 14. Typical radiation patterns for a Yagi antenna



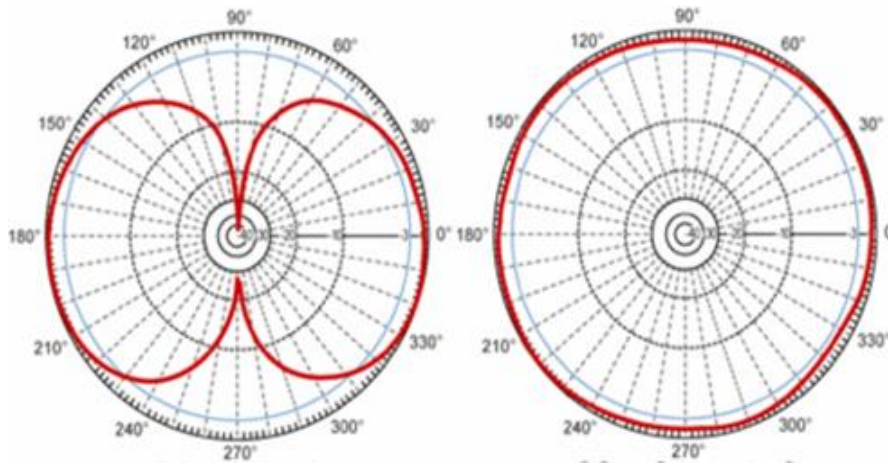
Due to the high directionality, the Yagi antenna is not a good choice for the drone. The drone would have to have some sort of gimbal assembly to point at the ground station. However, having a gimbal assembly on the ground to point the Yagi at the drone is not such a bad idea. The antenna can be configured to have a relatively high beam width (~30 degrees) while still achieving a high gain of 20-25dbi. This will likely be used as the ground station antenna.

4.3.2.6 Clover Leaf

The clover leaf antenna is one of the least directional antennas discussed so far. Most cloverleaf antennas transmit equally well throughout the azimuthal plane and have large lobes in the elevation plane giving them large elevation angles. Of course, low directivity leads to a lower gain and as such clover leaf antennas typically only have a gain of 1-5 dBi.

Clover leaf antennas are not linearly polarized, but circularly polarized. Circular polarization oscillates through more than one cartesian plane and is less susceptible to interference and blockages from objects. Additionally, since most other signals are linearly polarized, using circular polarization leads to lower noise and interference. For these reasons, cloverleaf antennas are popular amount drone users because the drone can be flown in many different orientations and around different objects without losing signal. Circular polarization also makes MIMO slightly harder since two polarizations must be used. Fortunately, circular polarization can be right or left handed allowing for MIMO.

Figure 15. Typical radiation pattern for a cloverleaf antenna



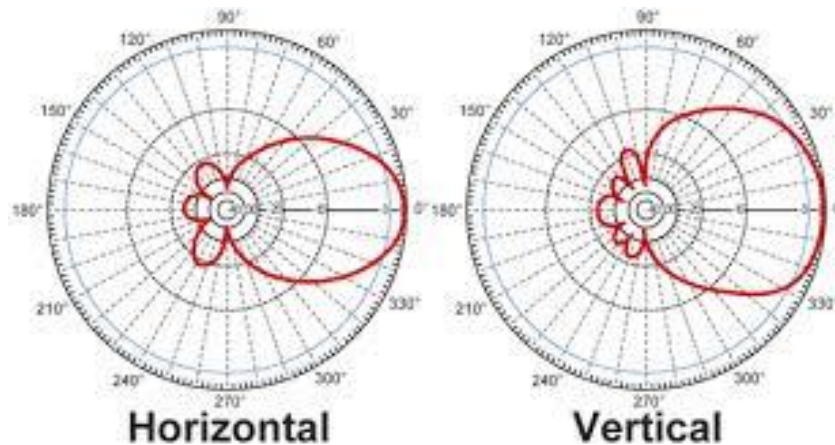
For the reasons listed above the cloverleaf antenna is certainly a contender for use on the drone. The only concern is whether the range will be long enough with the low gain of the antenna. This can be compensated for by using a directive antenna on the ground.

4.3.2.7 Helical

Helical antennas are different than the other antennas discussed so far because they radiate most of their energy in elevation plane rather than in the azimuthal plane. If a helical coil was setup orthogonal to the ground, most of the radiation would be directed vertically above the antenna. Due to the low beam width, this antenna has a very high gain typically on the order of 15-25 dBi.

Helical antennas can be configured in two ways, normal and axial. When configured normally, the helical antenna is basically a curled-up monopole and radiates linearly. When configured axially, the helical antenna becomes a highly directed circularly polarized radiator. Depending on the angle of the coils with respect to the horizontal, the antenna can be configured to emit left-hand or right-hand polarization. Because two polarizations are available the antenna is compatible with MIMO.

Figure 16. Typical radiation patterns of a helical antenna configured to operate axially



This antenna could be paired as a ground antenna with a cloverleaf on the drone. The high gain of the helical antenna would help compensate for the low gain of the cloverleaf and possibly improve the range. The one adaptation that must be made is that antenna tracking would be necessary due to the high directivity of the antenna.

4.3.3 MIMO

Multiple input multiple output (MIMO) is a strategy to boost data rate. The basic concept is that the throughput of a band can be increased by using two polarizations. For example, when using linear polarization, it is possible to use vertical and horizontal polarization to transmit two signals in a single bandwidth. Aside from increasing throughput, MIMO also allows for more reliable connections. If a polarization is blocked it is quite possible that the orthogonal polarization will not be blocked allowing for the connection to continue even if it is at half speed. The two polarizations don't necessarily have to be linear and orthogonal, circular polarization can be used as well with one signal transmitted over the right-hand polarization and the other transmitted over the left-hand polarization.

4.3.4 Fresnel Zones

All electromagnetic wave can be reflected, refracted and absorbed by various materials, often metals. This is typically most obvious with light but is equally applicable to lower frequency electromagnetic waves. The most obvious of these characteristics is absorption which occurs when an object attenuates a signal that passes through it. This is typically what causes signals like Wi-Fi to decrease when the signal passes through a wall or other object. The absorption factor is related to frequency and the higher the frequency, the higher the attenuation factor for most materials. The reflection and refraction of waves can be much more insidious though. Sometime a wave can be reflecting off an object and the reach the receiver with reasonable amplitude and sometime a wave can reflect off multiple objects

and destructively interfere with itself or another wave. This problem becomes worse in some ways when MIMO is used because a reflected wave may change polarizations at interfere with other signals.

The physicist Augustin-Jean Fresnel sought to simplify this problem by creating some formulas for zones in which a signal would be particularly sensitive to obstructions. Over time empirical results have led to the rule of thumb that if the first Fresnel zone is approximately 60% clear, there will be minimal interference to the signal transmitted through that zone.

Equation 3. Approximate equation for calculating the radius of any Fresnel zone at any point P

$$F_n = \sqrt{\frac{n\lambda(d_1)(d_2)}{d_1 + d_2}}$$

Table 14. Fresnel zone symbol definitions

Symbol	Definition
n	Order of Fresnel Zone
λ	Wavelength (m)
d1	Distance from one side to a point P
d2	Distance from the other side to a point P

This formula can be simplified to find the radius at the midpoint of the first zone which is typically the value of interest. The reason why is because obstructions have the biggest impact in the first zone so if the largest radius of this zone is calculated and kept clear of obstructions. The signal is likely to experience very little interference.

Equation 4. Approximate formula for calculating the radius at the midpoint of the first Fresnel zone.

$$F1[m] = 8.656 \sqrt{\frac{D[km]}{f[GHz]}}$$

Table 15. Symbol definition for the approximate formula for the radius at the midpoint of the first Fresnel zone.

Symbol	Definition
D	Distance between transmitter and receiver in km
f	Frequency of the signal in GHz

This formula is useful for calculating the height that the ground antenna should be kept at. Because the first Fresnel zone is so important, and the largest radius occurs at the midpoint, if the ground antenna is kept above this radius there is

unlikely to be infractions on the Fresnel zone from the ground and reduced interference from nearby objects.

$$F1[m] = 8.656 \sqrt{\frac{1}{5.8}} = 3.59 \text{ m}$$

This is the calculation for the maximum radius of the first Fresnel zone of the high-speed wireless connection. Ideally, the antenna should be kept 3.59 m off the ground, which is about 12ft. The competition limits antenna height to 15 ft, so this number fits in nicely. It is important to note that this is an approximation and other factors like the height of the drone and the directionality of the antenna can affect the Fresnel zone and as such the height of the antenna. However, this provides a good rule of thumb.

4.4 Wireless Communication Protocols

There are many wireless protocols available which simplify the design and construction of wireless systems by allowing manufacturers to create modules and products that will interoperate with each other and have known characteristics which simplify the analysis of the wireless design. A selection of wireless protocols useful to this project are presented here.

4.4.1 Wi-Fi

Wi-Fi is one of the most common and widely used forms of wireless communication. Also known as 802.11 networking, in reference to the IEEE standard, Wi-Fi is by businesses and personal consumers alike for connecting computers, mobile devices, and other Wi-Fi enabled devices. Wi-Fi benefits from advantages such as being inexpensive and easily set up. Like all other forms of electronic wireless communication, Wi-Fi makes use of radio waves. In a generic Wi-Fi set up, the general steps taken for data transmission are as follows (Brian, Wilson, & Johnson, 2001):

1. A device's Wi-Fi adapter translates a desired data sequence into a radio frequency signal and transmits the translated data via the air using an antenna
2. A separate Wi-Fi router or adapter in range receives the signal and decodes the information into the original data sequence and sends the information to the target device or over the Internet using a wired connection; a wired and physical Ethernet connection is usually used for this purpose

For the purposes of this project, an implementation of Wi-Fi is considered for telemetry and data transmission where the unmanned aerial vehicle could act as a wireless access point or router between the ground control station and the unmanned ground vehicle. Alternatively, Wi-Fi Direct communication links could be created between the unmanned aerial vehicle and ground control station and

the unmanned aerial vehicle and the unmanned ground vehicle. In this case, no router or wireless access point would be identified as a hub, and systems would communicate directly.

Unlike some other wireless communication protocols such as IR or those that operate in the sub-1 GHz band, Wi-Fi communication typically occurs at frequencies of 2.4 GHz or 5 GHz. Significantly higher than frequencies used by walkie-talkies, televisions, and cell phones, the higher frequencies allow Wi-Fi signals to carry more data. In addition, Wi-Fi radios can typically transmit on one of three frequency bands. Alternatively, they can “frequency hop” rapidly between each band to reduce occurrences of interference while allowing a greater number of devices to simultaneously use the same wireless connection (Brian, Wilson, & Johnson, 2001). This is crucial to avoid communication losses in saturated environments. **Figure 17** depicts how the 2.4 GHz frequency band used by 802.11b, 802.11g, 802.11n, and 802.11 ac is subdivided into multiple channels. The 5 GHz band used by 802.11a, 802.11n, and 802.11ac is used in a similar manner but is regulated at a higher level. Other frequency hopping protocols follow a similar pattern. As previously mentioned, Wi-Fi makes use of IEEE 802.11 networking standards, which are outlined in.

Figure 17. 2.4 GHz Wi-Fi channels (Gauthier, 2009). The use of this image is governed by the CC BY-SA 3.0 (Creative Commons Corporation, 2007).

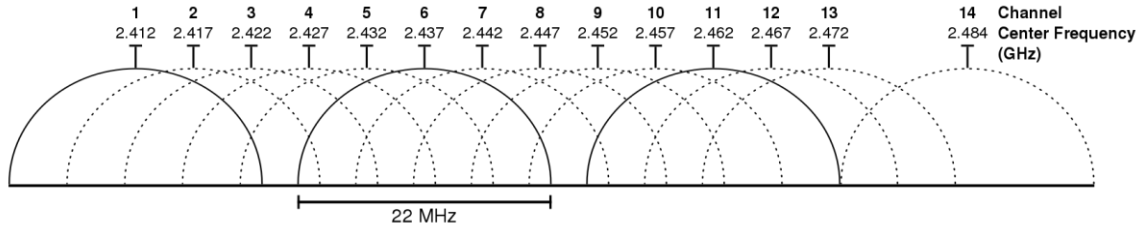


Table 16. IEEE Wi-Fi networking standards 802.11a and 802.11b

Networking Standard	Summary
802.11a	Transmits at frequency of 5 GHz Transmits with an upper limit of 54 megabits per second Uses orthogonal frequency-division multiplexing to split radio signals into sub signals to reduce interference
802.11b	Least expensive, but slowest Transmits at 2.4 GHz Transmits with an upper limit of 11 megabits per second Uses the modulation scheme of CCK (complementary code keying) to improve speeds

Table 17. IEEE Wi-Fi networking standards 802.11g, 802.11n and 802.11b

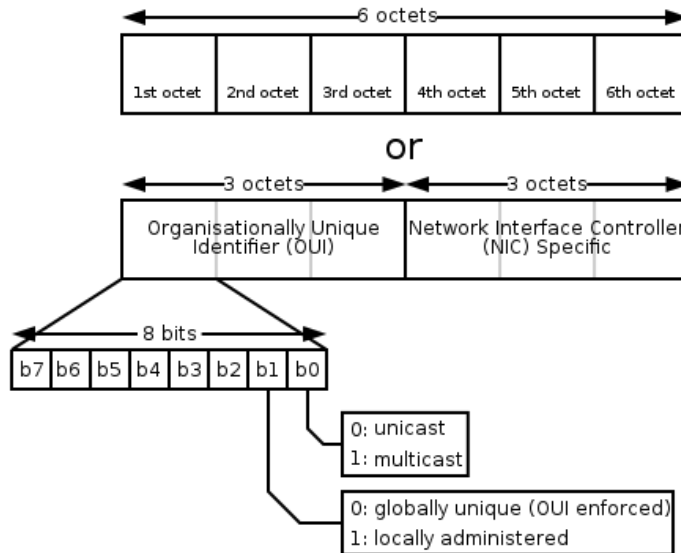
802.11g	<p>Transmits at 2.4 GHz</p> <p>Transmits with upper limit of 54 megabits per second</p> <p>Real-world speeds closer to 24 megabits per second due to congestion of the network</p> <p>Uses the same frequency as 802.11b, but the same modulation scheme as 802.11a for better performance</p>
802.11n	<p>Most common and is widely available</p> <p>Offers backward compatibility with 802.11a, 802.11b, and 802.11g</p> <p>Can achieve transmission speeds of 140~150 megabits per second</p> <p>Up to four simultaneous data streams</p>
802.11ac	<p>Newest standard as of 2013</p> <p>Offers backward compatibility with previous standards</p> <p>Less prone to interference</p> <p>Theoretical limit of 450 megabits per second per stream</p> <p>Up to eight simultaneous data streams</p> <p>Also known as 5G Wi-Fi, Gigabit Wi-Fi or Very High Throughput (VHT)</p>

Although **Table 16** suggests that 802.11ac is an ideal wireless communication standard for all Wi-Fi networks, it is important to note that the 802.11ac standard has not been fully adopted by all devices, especially when compared to 802.11n. The discrepancy is improving, however, as more new devices are released with support for 802.11ac.

As with any wireless connection, security is of paramount importance. Originally, WEP (Wired Equivalency Privacy) was the de facto standard for wireless network security. Improvements on this standard have led to the two much improved Wi-Fi security techniques used today. The first feature available on most Wi-Fi networks is known as WPA2 (Wi-Fi Protected Access version 2) and is recommended for most Wi-Fi networks. WPA2 is based on advanced encryption techniques, especially AES (Advanced Encryption Standard) and password login. MAC (Media Access Control) address filtering is another commonly used technique. Rather than reliance on user authentication via login, MAC address filtering checks each device's MAC address against a list of allowed MAC addresses to establish connections (Brian, Wilson, & Johnson, 2001). An example of a MAC-48 address is shown in **Figure 18**. Each technique has its own benefits and pitfalls.

Figure 18. Diagram showing the structure of a MAC-48 network address, explicitly showing the positions of the multicast/unicast bit and the OUI/local address type

bit (Kju, 2007). The use of this image is governed by the CC BY-SA 2.5 (Creative Commons Corporation, 2005).



Wi-Fi range can depend on many factors, though an average router can provide wireless connectivity for approximately 100 feet radially. Obstacles such as doors or walls, however, can block the signal unless devices such as repeaters or range extenders are used. The relatively short range of Wi-Fi is due in part to the use of higher frequencies, which suffer from greater susceptibility to obstacle interference as a tradeoff from the higher throughput that higher communication frequencies offer (Brian, Wilson, & Johnson, 2001). This characteristic makes standard Wi-Fi unsuitable for drone telemetry, as such systems are expected to maintain stable connections at ranges greater than even multiple hundred feet. However, the high potential data throughput of Wi-Fi makes the wireless communication protocol much better suited for devices that will be communicating with the ground control station systems, which will be in much closer proximity.

4.4.2 Bluetooth & BLE

Bluetooth is very common wireless communication technology that has been adopted as a feature by a large majority of new personal computing devices and mobile computing devices. Bluetooth offers a streamlined and automatic connection process that simplifies the task of connecting multiple devices in specific cases. The complete Bluetooth technology protocol stack is depicted in **Figure 19**. Normally, devices that wish to communicate wirelessly must first determine a predefined way to structure and package data for both reliable transmission and reception such that both devices are able to understand one another. Bluetooth standardizes this agreement on a physical level with a radio frequency communication standard and on a protocol level to determine how many and what type of bits are sent to communicate messages (Franklin & Layton,

2000). **Figure 20** outlines the structure of Bluetooth packets, and the data that is added as a header to data messages transmitted using the Bluetooth protocol stack. Bluetooth also minimizes user intervention and power consumption for higher efficiency. The most common variants are known as Bluetooth (now Bluetooth Classic) and the newer BLE (Bluetooth Low Energy). Though BLE is a more modern standard, each variant has its own strengths and weaknesses and are relevant for different applications.

Figure 19. Bluetooth Protocol Stack (Leszek Chuchla, Bluetooth protokoly, 2010). The use of this image is governed by the GFDL (Free Software Foundation, 2008).

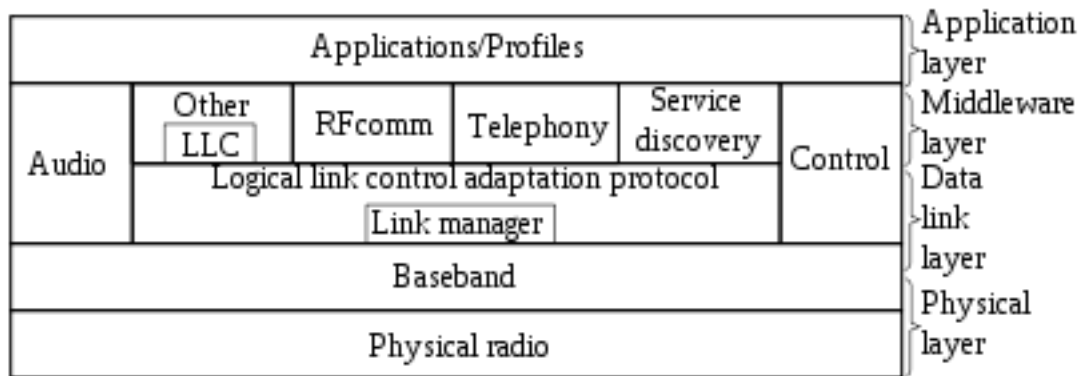
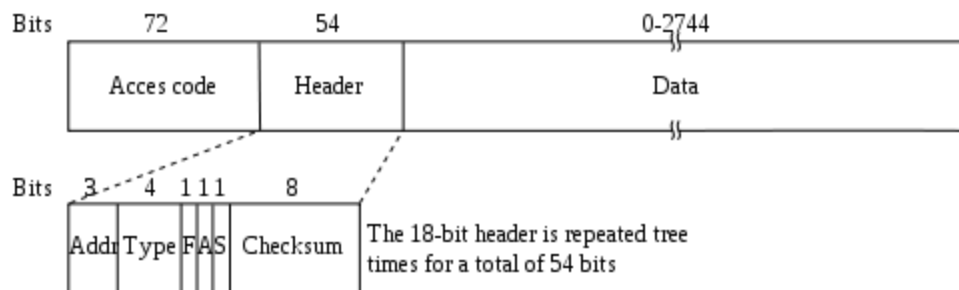


Figure 20. Bluetooth packet structure (Leszek Chuchla, Bluetooth ramka, 2007). The use of this image is governed by the GFDL (Free Software Foundation, 2008).



Bluetooth operates at frequencies centered around 2.45 GHz, close to Wi-Fi 2.4 GHz. To prevent interference with other higher-powered systems operating at a similar frequency, Bluetooth sends comparatively weak signals on the order of a single milliwatt. As a result, the range of Bluetooth is generally around 10 meters, or 32 feet, radially (Franklin & Layton, 2000). This range makes Bluetooth highly unsuitable for drone to ground station telemetry, as such communication requires stable connections at ranges of multiple hundred, and possibly even thousands, of feet or meters.

In contrast, Bluetooth excels at creating relatively short-range personal area networks, such possibly between wireless sensor nodes and a flight computer of an unmanned aerial vehicle. To prevent interference between multiple devices communicating at such close proximity, Bluetooth employs a technique known as

“spread-spectrum frequency hopping”, which allows devices to transmit on and switch between a selection of 79 individual frequency channels automatically (Franklin & Layton, 2000). This technique minimizes collisions and complete disruptions. On top of simplified connectivity, Bluetooth connections also benefit from integrated security.

Though most of the previously described attributes of Bluetooth apply regardless, a new variant of Bluetooth known as BLE (Bluetooth Low Energy) is different enough that a clear comparison should be drawn between the two. As the name implies, BLE is a “low energy” variant of Bluetooth, which can also be called “Bluetooth Classic”. Probably the most significant between Bluetooth Classic and BLE is the fact that devices generally remain inactive in BLE, only activating when data must be transmitted or is received. In contrast with Bluetooth Classic, which keeps both devices active and connected for the duration of the connection, BLE offers a much cheaper and lower-power connectivity solution (RF Wireless World, 2012). In summary, Bluetooth, though unsuitable for long range wireless communication and drone telemetry, is well-suited to connecting peripherals at a personal area network level. Additional differences can between Bluetooth Classic and BLE can be found in

Between Bluetooth Classic and BLE, BLE improves on the older Bluetooth Classic standard in multiple key areas. Although the power consumption of Bluetooth Classic is relatively low compared to other standards, BLE improves on this characteristic for even greater power savings. BLE also offers higher data transfer speed and range, but only in bursts. In real world use, because Bluetooth Classic maintains a constant data stream while BLE does not, the throughput of Bluetooth Classic for applications such as file transfers is considerably higher. Once again, though neither Bluetooth Classic nor BLE support the range necessary for unmanned aerial vehicle to ground station or unmanned ground vehicle telemetry and data transmission, both technologies are well-suited to close-proximity sensor devices.

Table 18. Bluetooth Classic vs. BLE (information taken from RF Wireless World) (RF Wireless World, 2012).

Specifications	Bluetooth Classic	BLE
Network/Topology	Scatternet	Star Bus
Power Consumption	Low (less than 30 mA)	Very Low (less than 15 mA)
Speed	700 Kbps	1 Mbps
Range	<30 m	50 meters (150 meters in open field)
RF Frequency Band	2400 MHz	2400 MHz
Frequency Channels	79 channels from 2.400 GHz to 2.4835 GHz with 1 MHz spacing	40 channels from 2402MHz to 2480 MHz (includes 3 advertising and 37 data channels)
Modulation	GFSK (modulation index 0.35), $\pi/4$ DQPSK, 8DPSK	GFSK (modulation index 0.5)
Data Transfer Latency	Approx. 100 ms	Approx. 3 ms
Spreading	FHSS (1MHz channel)	FHSS (2MHz channel)
Link layer	TDMA	TDMA
Message Size (bytes)	358 (Max)	8 to 47
Error Detection/Correction	8-bit CRC(header), 16-bit CRC, 2/3 FEC(payload), ACKs	24-bit CRC, ACKs
Security	64b/128b, user defined application layer	128 bits AES, user defined application layer
Application Throughput	0.7 to 2.1 Mbps	less than 0.3 Mbps
Nodes/Active Slaves	7	Unlimited

4.4.3 LoRaWAN

Before discussing LoRa and LoRaWAN in detail, it is important to assert that there is a distinct difference between the two. According to the LoRa Alliance, “LoRa (Long Range) is the physical layer or the wireless modulation utilized to create the long range communication link” while “LoRaWAN (Long Range Wide Area Network) defines the communication protocol and system architecture for the network” which is based on LoRa” (LoRa Alliance Technical Marketing Workgroup 1.0, 2015). In contrast with older, more traditional modulation schemes such as frequency shift keying that are used for low-power wireless systems, LoRa is based on a more modern low-power modulation scheme known as chirp spread spectrum modulation; this modulation scheme keeps the low-power benefits of frequency shift keying while significantly extending communication range. Though similar modulation schemes have long been used for military purposes, LoRa

essentially pioneers the use of high link budget technology for commercial usage at low cost (LoRa Alliance Technical Marketing Workgroup 1.0, 2015).

Although Low Power Wide Area Networks (LPWAN) enabled by LoRa and similar technologies exhibit multiple ideal characteristics such as low power, high data rate, and long range, they are not suitable for replacing all network configurations. Based on wide adoption and well-established standards alone, for example, Wi-Fi and Bluetooth are excellent options for high data-rate local and personal area networks. Cellular networks are similarly well suited when power sources are available and long-range communication is required. LPWAN's fit in where low power and long range or inconvenient positioning is required, but data rate is relatively low (LoRa Alliance Technical Marketing Workgroup 1.0, 2015). Wide-area sensor networks and even drone telemetry data links can be considered viable candidates for this type of network communication.

There are a few key characteristics of LoRaWAN networks that should be considered. LoRaWAN network nodes, unlike Wi-Fi or Bluetooth, are not tied to individual gateways. On the contrary, multiple gateways that are in range of the transmitting node will typically receive the message and forward it to some cloud-based network server responsible for handling redundant packets, scheduling, and security checks. In effect, no handover is required (LoRa Alliance Technical Marketing Workgroup 1.0, 2015).

To benefit battery life, LoRaWAN nodes also differ from other types by having asynchronous communication properties. Rather than attempt to synchronize with cellular or similar mesh networks and consume unnecessary power, LoRaWAN nodes only transmit when data is available in an either scheduled or event-driven manner. To optimize network capacity, LoRaWAN employs adaptive data rate and multichannel multi-modem transceivers in gateways to enable parallel and multi-channel reception and provide adaptability to different network conditions. Adaptive data rates also provide power savings and increased network capacity by shortening on-air time. LoRaWAN also includes two layers of security at both the network and application level. Network security ensures node authenticity while application security ensures the privacy of data in-transit (LoRa Alliance Technical Marketing Workgroup 1.0, 2015). Although some of the features built-in to LoRaWAN add additional overhead, they also decrease development and deployment time while augmenting the viability of LoRaWAN for many general-purpose uses.

In the context of this project, an architecture could be considered where the unmanned aerial vehicle, ground control station, and unmanned ground vehicle could each have some combination of characteristics of an end node and gateway for bidirectional data transmission. LoRaWAN standards define differences between various types of devices/nodes as outlined in **Table 19** (LoRa Alliance Technical Marketing Workgroup 1.0, 2015). Each device class identify a level of balance between network downlink communication latency and battery lifetime. All device classes offer bidirectionality of data transmission, ranging from minimal

battery consumption (class A) to continuously open receive windows (class C). Class A devices only open receive windows for a short time after transmission, while class C devices are only closed to receiving data when transmitting (LoRa Alliance Technical Marketing Workgroup 1.0, 2015).

Table 19. LoRaWAN device classes (information taken from LoRa Alliance)

Device Class	Characteristics
Class A (battery-powered sensors)	Most energy efficient Must be supported by all devices Downlink available only after sensor TX
Class B (battery-powered actuators)	Energy efficient with latency-controlled downlink Slotted communication synchronized with a beacon
Class C (main-powered actuators)	Devices which can afford to listen continuously No latency for downlink communication

The long range and low-power characteristics of LoRaWAN networks make them a viable possibility for long range unmanned aerial vehicle to ground control station and unmanned ground vehicle telemetry systems.

Table 20. LoRaWAN specification summary for North America (information taken from LoRa Alliance) **Table 20** provides additional details regarding LoRaWAN usage North America and specifically the United States (LoRa Alliance Technical Marketing Workgroup 1.0, 2015). The frequency bands used by LoRaWAN fall into those acceptable for the competition, and data rates are sufficient for telemetry purposes. A relatively large link budget contributes to impressive range for such low power devices. Power efficiency will also help LoRaWAN modules avoid placing significant additional stress on battery systems. And interference immunity and security will be crucial to protect signal links from unwanted transmissions.

Table 20. LoRaWAN specification summary for North America (information taken from LoRa Alliance)

Specification	Value
Modulation	SS Chirp
Frequency Band	902-928 MHz
Channels	64 + 8 + 8
Channel BW Up	125/500 kHz
Channel BW Down	500 kHz
TX Power Up	+20 dBm typical (+30 dBm allowed)
TX Power Down	+27 dBm
SF Up	7-10
Data Rate	980 bps – 21.9 kbps
Link Budget Up	154 dB
Link Budget Down	157 dB
Power Efficiency	Very High
Interference Immunity	Very High
Security	Yes

4.4.4 Zigbee & DigiMesh

Zigbee and DigiMesh are two popular mesh networking protocols. Mesh networking allows for extended data routing by allowing messages to travel node by node. The multi-point nature of mesh networks also allows for “self-healing” and increased reliability; when nodes or connection links fail, alternative data paths can be created to keep the network online (Digi International, 2018). Although self-healing technology has minimal benefit in a 3-node network such as one between an unmanned aerial vehicle, unmanned ground vehicle, and ground control station, packet routing would be beneficial in cases where line-of-sight between the ground control station and unmanned ground vehicle is obscured; the unmanned aerial vehicle can be used as a router.

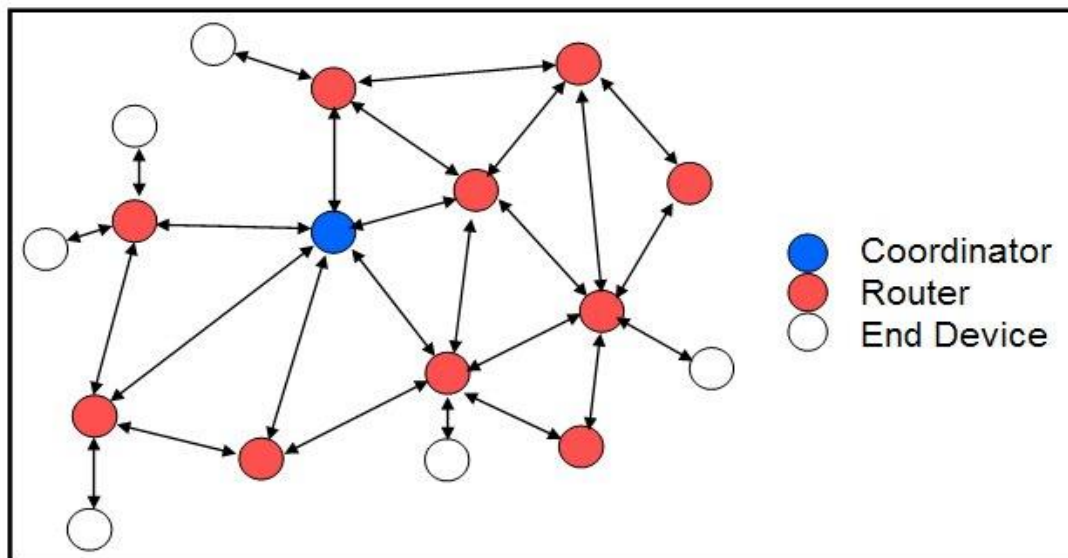
Zigbee, developed by the Zigbee alliance, defines three node types that make up a Zigbee network (Digi International, 2018):

- Coordinator
 - Responsible for network forming and traffic routing
 - One and always one per network
 - Often stores network information
 - Acts as router after network formation

- Router
 - Intermediate node
 - Relays information and data between nodes
 - Can also act as an endpoint
 - Cannot sleep
 - Must be available for relaying messages at all times
- End Device
 - Does not route traffic
 - Does not relay data between nodes (except to coordinator/router)
 - Can be mobile, low-power, battery-driven
 - Allowed to sleep
 - Must have coordinator or router as parent node
 - Up to 20 end devices per parent

An example of a Zigbee mesh network, including all types of Zigbee nodes, is included as **Figure 21**.

Figure 21. Zigbee mesh network nodes. The use of this image is governed by the CC BY-SA.



This Photo by Unknown Author is licensed under CC BY-SA

Overall, Zigbee offers benefits such as (Digi International, 2018):

- Cross-vendor interoperability as an open standard
- Security options
 - Device registration for out-of-band commissioning
 - Join restriction
 - Rotation of network keys
 - Trust center
- Source and many-to-one routing

Unlike standard Zigbee, the proprietary DigiMesh network protocol only has one node type, and thus defines a homogenous mesh network (i.e. every node can route data and is interchangeable with other nodes). In addition, DigiMesh allows for sleeping routers, which reduces overall power consumption considerably. Synchronized cyclic sleep and asynchronous sleep modes enabled by the DigiMesh protocol allow all nodes to sleep and wake in different configurations. In addition to these characteristics, DigiMesh is supported by higher data rate and longer-range sub-GHz platforms. DigiMesh also benefits from simplified addressing, setup, troubleshooting and security (Digi International, 2018). For the purposes of this project, DigiMesh is likely to be preferred over Zigbee for longer range and simplicity, as well as the benefit of homogenous packet relaying. **Table 21.** Comparison summary of Zigbee Mesh vs. DigiMesh (information taken from Digi International) provides a summary of benefits between the two protocols (Digi International, 2018).

Table 21. Comparison summary of Zigbee Mesh vs. DigiMesh (information taken from Digi International)

Zigbee Mesh	DigiMesh
Open standard-based	All nodes can sleep
Cross-vendor interoperability	Simple setup and expansion
Bi-directional communication	Longer range (sub-GHz) options
Sleeping end devices	Broadcast-intense application support

The simplicity, longer range, sub-GHz support, and broadcast-intense application capability are all highly beneficial advantages for the objectives of this project. Telemetry data will be relatively simple but will require long-range support. The benefits of an open standard and cross-vendor interoperability, while useful, will be minimal due to the inherently closed nature of the required systems. Simplicity is also key for development purposes.

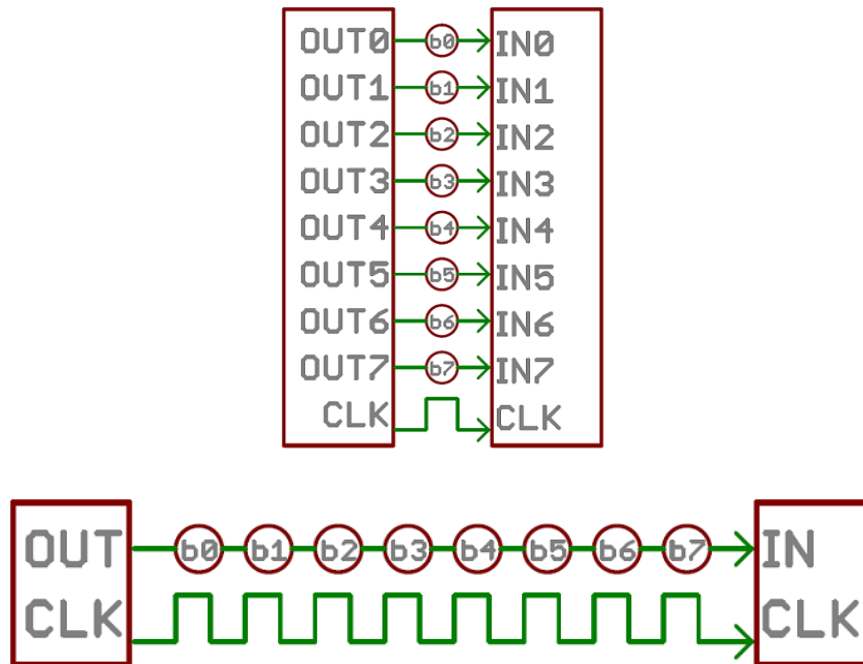
4.5 Wired Communication

Wired communication is essential for connecting and interfacing even the most basic embedded electronics and modules. In general, wired communication is also preferred over wireless communication when convenient for optimal simplicity, robust operation, and throughput performance. Even the simplest wireless protocols require some type of conversion between networking layers and modulation of signals for acceptable transmission ranges. Wireless links also must compensate for many more variables, including obstacles obstructing line of sight, media propagation differences, and interference and noise from multiple other sources. In general, wireless throughput is lower than wired throughput due to a number of these and many other reasons. For all applications that do not require data transmission external to a local system (such as within a UAV), wired communication will be utilized.

4.5.1 Parallel vs. Serial

Though devices may have multiple types of wired communication protocols available, they can be generally be identified as either parallel or serial, as depicted in **Figure 22**. Parallel interfaces allow for multiple bits to be transferred between transmitter and receiver at one time over a bus. A bus of multiple wires allows a large amount of data to be transferred in parallel for extremely high potential throughput compared to serial communication performed over a pair of wires. However, though parallel interfaces are not necessarily complex, they necessarily require a much larger number of input and output lines and more physical overhead. Though parallel interfaces are very useful for sending large amounts of data, serial communication interfaces are often chosen to perform the same duty with lower cost and reduced usage of space, pins, and wires (Sparkfun Electronics, 2012). Parallel communication is used by several components of the Nvidia Jetson, which will act as the primary computer of the UAV.

Figure 22. Parallel (top) vs. serial (bottom) communication (Sparkfun Electronics, 2012). The use of this image is governed by the CC BY-SA 4.0 (Creative Commons Corporation, 2013).



Serial communication can be further divided into asynchronous and synchronous protocols. Synchronous serial interfaces such as SPI and I²C are generally more straightforward and faster but require an additional wire (or more) to share a common clock signal between devices that aim to communicate. This external clock signal is what “synchronizes” two communicating devices during data transfer (Sparkfun Electronics, 2012).

In contrast, serial communication that is performed without the external clock signal is known as asynchronous serial communication. Because no additional wire or pins are required for a separate clock signal, asynchronous serial communication is ideal for compact systems where it is essential to minimize the usage of wires and I/O pins. Asynchronous communication without a clock signal is so common in embedded electronics that most people end up referring to this type of communication as simply “serial” communication (Sparkfun Electronics, 2012). A diverse mix of both parallel and serial communication protocols is required for UAV and ground control systems to operate successfully.

4.5.2 I²C

I²C is a protocol for allowing up to 127 devices to be connected to a simple 2 wire bus with a master. The master can read and write bytes to and from the slave devices on the bus. The bus hardware consists of 2 wires shared by every device: data and clock, each with a pull up resistor. Devices communicate on the bus through open collector outputs and high impedance inputs. This allows devices to drive the bus without worrying about damage to the transceivers if two devices try to drive the bus at the same time, if they can detect the resultant data corruption.

When sending I²C data, the clock line is both driven and monitored by the master. The data line changes to the next bit in the sequence when the clock is low, and the data is read by the recipient when the clock is high. Most of the time the clock is controlled by the master, but slaves may hold the clock low to indicate they are not yet ready for more data (NXP Semiconductors, 2014).

To perform a transaction, the master begins by bringing the data line low while leaving the clock high. Since the data line should not change when the clock is high otherwise, slaves can interpret this as a start condition. The master then sends out 7 bits corresponding to the address of a slave device, and a bit to indicate if it wants to read or write to that device. The device acknowledges by holding the data line low for one clock cycle. The master can then either send a stop condition to complete the transaction or another start condition to read or write a byte (NXP Semiconductors, 2014).

Many I²C devices use a concept of “registers”, where a master can select where the data being read or written will come from or go. A device like the LTC2944 has registers for control and the different readings it produces. To write to a register, the master will start by sending the address with a write bit, then sending the number of the register it wishes to write to, followed by sending the data it wishes to write. Many devices support writing multiple registers by allowing the master to continue writing bytes, incrementing the register number internally after each byte. Reading a byte starts with a similar write transaction that gives the address of the device and the id of the register to be read. The write transaction is then stopped, and a read transaction begins where the master sends the slave address in read mode, then produces a clock while the slave drives the data line to read out the desired byte. Many slaves support reading multiple bytes by the master continuing

to drive the clock after 8 bits and will read out the next register in sequence (NXP Semiconductors, 2014).

I²C devices support several clock rates depending on the device. Most devices will support either 100KHz (standard mode) or 400KHz (fast mode) clocks, while some support up to 1MHz (fast mode +) or 3.2MHz (high speed) (NXP Semiconductors, 2014). The clock speed can vary due to line capacitance and slaves holding the clock to get ready, but this is acceptable since the devices are synchronized by the clock, unlike UART where a drifting baud rate would cause devices to get out of sync and be unable to transfer data.

I²C is very easy to implement since the frequency is not high enough to require sophisticated routing and cabling in most cases. Most embedded processors will support being a master and some support being a slave, and many sensors like battery gas gauges or barometers support being slaves. This makes I²C an excellent choice for applications involving many sensors that don't require large amounts of data, since the master can communicate with many devices using only 2 pins.

The bus does have relatively low bandwidth, so for applications that require rapid readings or large amounts of data another standard may be necessary. Another issue is that some devices have either 1 possible address or a limited number of variable addresses. If multiple devices on a bus have the same address, both will try to respond to it, which makes communicating with one or the other impossible, and can corrupt data. To remedy this, devices are available which can intercept and modify addresses moving from one side of the device to the other, so that slaves on the other side will appear to have a different address to the master. The addresses are typically XORed with a value determined by a pair of voltage dividers connected to the IC. This solves the issue of using multiple devices with fixed addresses while requiring no consideration from the master other than for it to use a different address for the devices behind the translator.

Another option is to simply use a switch that disconnects different parts of the bus depending on which device the master wishes to communicate with, but this requires the master to somehow manage the switch, which may require additional I/O or bus bandwidth.

4.5.3 I²C isolation

In systems which have safety concerns like medical devices or high voltage power, or systems which have issues with different ground potentials or ESD concerns, isolators designed specifically for I²C exist which can transmit the signals across an insulator, ensuring no unwanted currents flow from one side to the other. In this system, the grounds of different parts may differ due to the current sensing arrangement explained later.

TI and Analog Devices offer I²C isolators, the ISO154x and ADuM125x . Both parts operate at I²C speeds up to 1MHz, both can operate on 3.3V or 5V power, and both comply with UL standards for isolating 2500 V_{rms}. They operate by magnetically or capacitively coupling the I²C signals across a barrier. Each device has a variant which has bidirectional sending of both the clock and data signals, and a variant which only has bidirectional sending of the data signal. This is important if the bus has multiple masters or devices which employ clock stretching. The main difference is that the ISO1540 is \$4.314 each in quantity 10, while the ADuM1250 is \$4.95 each in quantity 10.

4.5.4 SPI

Serial Peripheral Interface (SPI) is a synchronous serial interface that is often utilized for sending data between microcontrollers and various small peripherals. These peripherals may include components such as shift registers, SD cards, and various types of sensors (Grusin, 2013). SPI is utilized as a communication protocol by many of the microcontrollers and radio frequency communication devices that are considered for this UAV project. As required by other synchronous protocols, SPI requires a separate clock and data lines. SPI also utilizes a select line to select target communication devices.

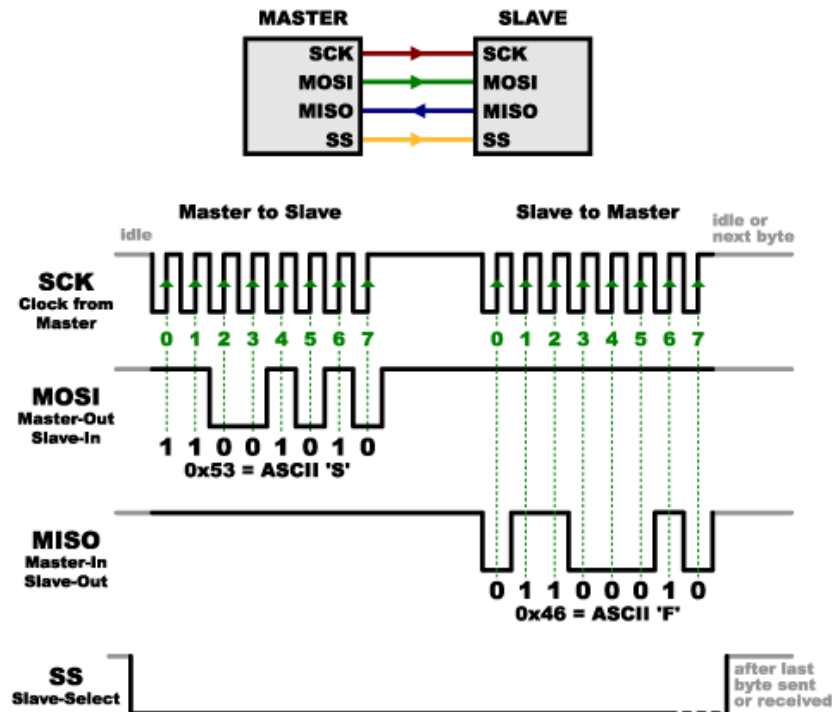
Some of the drawbacks of asynchronous serial communication is the lack of control over when data transmission occurs, and the lack of guarantee on whether both transmitter and receiver are operating at exactly the same rate. Slight differences can exist between devices that are running on separate clock signal sources, and any mismatches can result in garbage data being received by the receiver. This occurs whenever the receiving device samples the transmitted signal at incorrect moments. Though the start and stop bits used in asynchronous serial communications help alleviate the issue to some degree, additional bit overhead is consumed as a result. The hardware required for asynchronous serial communication can also often be quite complex (Grusin, 2013).

In contrast with asynchronous serial communication, SPI uses a dedicated clock signal to ensure that both the transmitting device and receiving device are synchronized. This ensures that data is transmitted and sampled at the correct frequencies on both ends. Although each device may have a maximum transfer speed, the speed of operation does not have to be decided upon beforehand because of the shared clock signal. SPI is very popular, especially in low-cost hardware, because of its simplicity. In fact, the receiving device can be nothing more than a shift register in comparison to the complex Universal Asynchronous Receiver / Transmitter (UART) hardware that is required for asynchronous serial communication (Grusin, 2013). SD cards use the SPI protocol, and are used to store data and even operating systems by devices used in this project.

One-way SPI communication is extremely simple in relative terms. The procedures required for two-way SPI communication, however, is somewhat more complex. In SPI, there may be multiple “slave” devices, but all rely on the clock signal produced

by a single “master” device. The master device is often a microcontroller communicating with multiple “slave” peripheral devices. When the master sends data to a slave device, the bits are transmitted on the MOSI data line. MOSI stands for “Master Out / Slave In”. If a response is expected from the slave, the master generates additional clock cycles and reads the data generated by the slave on a separate data line known as MISO, or “Master In / Slave Out” (Grusin, 2013). A graphical summary of this concept is depicted in **Figure 23**.

Figure 23. SPI communication (Grusin, 2013). The use of this image is governed by the CC BY-SA 4.0 (Creative Commons Corporation, 2013).



The number of additional clock cycles generated by the master must be known beforehand, as the slave is unable to generate clock cycles to match the data. Though this approach may not be as flexible or useful for random data, there are generally few issues when using SPI because data and commands are transferred in a way that is very specific and precise. The number of bytes that are expected to be read and written following each command are always known beforehand so that the appropriate number of clock signals can be generated. Because there are separate data lines for sending and receiving data, SPI is known as a “full duplex” protocol, although only some devices are capable of this feature (Grusin, 2013).

In addition to the clock, MOSI, and MISO lines, SPI has one additional type of line used for communication known as the SS or Slave Select line. As previously mentioned, there may be multiple slaves, but only one master. The master generally one SS per slave, as the SS lines are used to enable one slave at a time for communication (Grusin, 2013).

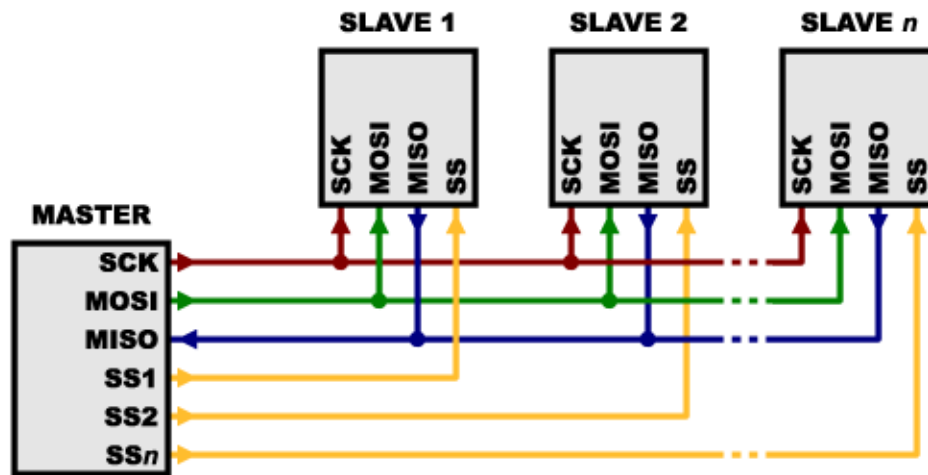
The SS line uses “active low” logic. Essentially, this means that the SS line is kept high by default. While the SS line for a slave is held high, the slave is essentially disconnected, preventing communication. The SS line is only brought low when it is time to begin communications with the selected slave. Before the clock begins generating pulses, and before data is transmitted along the MOSI and MISO lines, the SS for the target slave is brought low. After transmission is complete, the SS line is brought back to the high state (Grusin, 2013).

In summary, the SPI has four primary wires required for communications:

1. Serial Clock (SCK)
2. Slave Select (SS)
3. Master In / Slave Out (MISO)
4. Master Out / Slave In (MOSI)

Multiple slaves can be connected to a single master in two main configurations. Most of the time, each slave is connected to the master via its own SS line. In this configuration, a single slave’s SS line is brought low at a time, and data is transmitted only between the master and the selected slave. All MOSI ports are connected on a single node, and all MISO ports are connected on a single node. This configuration is very straightforward but requires many lines - at least one additional for each slave (Grusin, 2013). This concept is depicted in **Figure 24**.

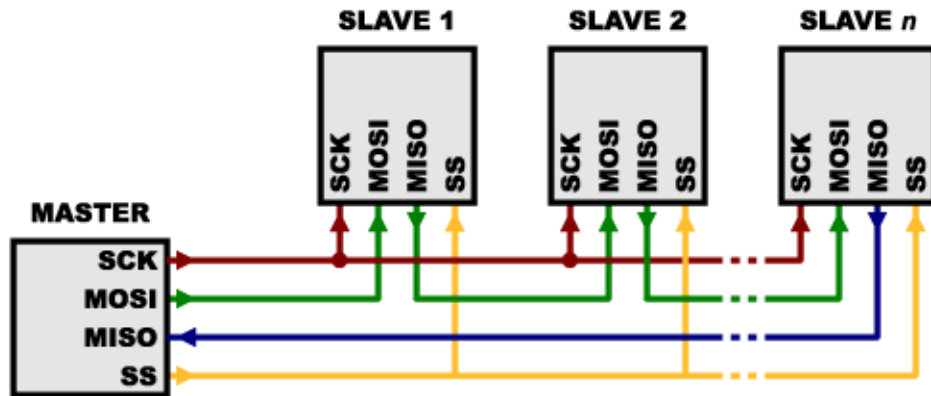
Figure 24. SPI communication with multiple slaves (Grusin, 2013). The use of this image is governed by the CC BY-SA 4.0 (Creative Commons Corporation, 2013).



Multiple slaves can also be connected to a single master device in a daisy-chain configuration. The chain begins by connecting the MOSI port of the master to the MISO port of the first slave. The first slave then connects its MISO line to the MOSI line of the next slave, and the pattern continues until the last slave. The MISO line of the last slave is then connected to the MISO line of the master. All slaves are connected via a single SS line. Data overflows when received by the first slave and from each slave thereafter, so any data that must be transmitted to a single

slave must be transmitted on top of enough data to reach all previous slaves. This configuration is often used in situations where only the output from the master is required. Although the MISO line can still be used, any return data must be transmitted through the daisy chain until the last slave is able to transmit it back to the master (Grusin, 2013). The daisy chain configuration is depicted in **Figure 25**.

Figure 25. SPI communication with multiple slaves in daisy-chain configuration (Grusin, 2013). The use of this image is governed by the CC BY-SA 4.0 (Creative Commons Corporation, 2013).



Compared to asynchronous serial communication, SPI carries the general benefits of faster data transmission, simpler receiving hardware, and the ability for one master to easily communicate with multiple slaves. However, SPI suffers from drawbacks such as the reliance on more signal-carrying wires than other protocols, which is exacerbated for situations that require multiple slaves. In addition, only the master is capable of controlling communication. Slaves are unable to communicate with each other directly, and all communication codes and command structures must be precisely defined beforehand (Grusin, 2013). SPI is still extremely useful for certain devices and peripherals, especially microcontrollers and other embedded electronics on UAV and ground control station systems.

4.5.5 Asynchronous Serial Protocol

Asynchronous serial communication is essential for sending data between many types of components in the UAV, ground control station, and UGV. Asynchronous serial communication is extremely common for sending data between systems, over the air via radio frequency communication modules, and between master and peripheral devices such as the GPS and flight controller. In contrast with synchronous serial communication protocols, asynchronous serial communication protocols transfer data without synchronization based on shared clock signals. Because of this, asynchronous communication requires additional techniques and parameters to ensure that data can be reliably transmitted and received (Sparkfun Electronics, 2012). These mechanisms include the following:

1. Baud rate

2. Parity bits
3. Synchronization bits
4. Data bits

Using a combination of these parameters, (asynchronous) serial communications can be highly configured to fit various applications. The caveat is that both the sending and receiving device must be set to the exact same protocol configurations (Sparkfun Electronics, 2012). An example of the structure of an asynchronous serial data frame including all these parameters is shown in **Figure 26**.

Figure 26. A frame of asynchronous serial data (Sparkfun Electronics, 2012). The use of this image is governed by the CC BY-SA 4.0 (Creative Commons Corporation, 2013).



The baud rate indicates the rate at which data bits are transmitted over a serial line. Baud rate is generally denoted in terms of bits-per-second (bps). The baud rate directly translates to the amount of time a serial line is held in the high or low states by the transmitting device, and the rate at which the receiver samples the data line. 9600 bps is one of the most commonly found baud rates. Though there is technically no limit on baud rate, devices have practical limitations that prevent baud rates from going too high. Baud rates that exceed device capabilities result in garbled data (Sparkfun Electronics, 2012).

In serial communications, data blocks are generally sent in frames of bits composed of a predetermined number of start, parity, and stop bits. Each data frame begins with a start bit, which brings a data line from 1 to 0 (high to low) to indicate the start of transmission. The data frame may end with either 1 or 2 stop bits, which bring the data line back to the idle state of 1 (high). The stop bits and start bits are known together as synchronization bits. One benefit that comes because of these synchronization bits is that even if one frame of data is garbled during transmission, the data line is essentially resynchronized before each data frame (Sparkfun Electronics, 2012).

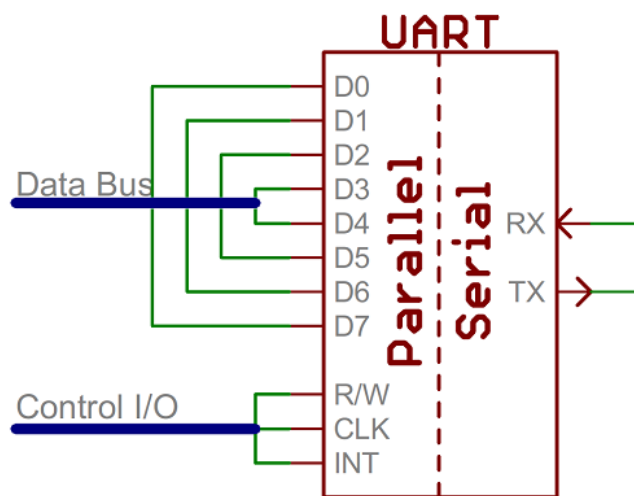
The synchronization bits enclose the data bits (which comprise a data block/chunk) and parity bits. Data chunks are often byte-sized but can vary in the number of bits from 5-9. In addition, data chunks can be sent in either most-significant bit (MSB) or least-significant bit (LSB) format. Least-significant bit first is the most common. Parity bits are optional and can be set to either even or odd parity. Parity bits are generated based on the number 1's and 0's present in the data block. In even parity mode, the parity bit is set such that the total number of 1's (including data

and parity bits) is even. In odd parity mode, the parity bit is set such that the total number of 1's (including data and parity bits) is odd (Sparkfun Electronics, 2012).

TX and RX labels must always be considered in terms of the device itself. For example, the RX pin of a device is connected to that device's receiver, while the TX pin of a device is connected to that device's transmitter. Unlike SPI, where MOSI pins are connected to MOSI pins and MISO pins are connected to MISO pins, the TX pin of one device should be connected to the RX pin of another. Though serial communications may be half-duplex or full-duplex, it is sometimes only necessary for data to flow in one direction. In this case, it is sufficient for a single wire to be connected from one device's TX line to another device's RX. This is identified as a simplex connection (Sparkfun Electronics, 2012).

At the hardware level, asynchronous serial communication can be implemented a multiple number of ways. TTL, or transistor-transistor logic serial is often used by microcontrollers and integrated circuits that are low-level. TTL serial signal levels are often matched to common voltage supply ranges used for microcontrollers, such as 0V to 3.3V or 0V to 5V. In TTL serial, an idle line or stop bit is indicated by a voltage at supply or Vcc level. This level indicates a bit value of 1. A start or data bit of bit value 0 is indicated by a ground signal of 0V. For embedded circuits, TTL is simple to implement but also susceptible to interference, noise, and general losses due to the relatively small voltage range. Standards such as RS-232 or RS-485 help alleviate these issues with alternate voltage magnitudes different start/stop bit level configurations (Sparkfun Electronics, 2012). Many of the simplest and low power devices opt for TTL serial. RS-232 is also commonly found in local systems. RS-485 is often used for more industrial and long-range serial communication applications.

Figure 27. Simplified UART diagram (Sparkfun Electronics, 2012). The use of this image is governed by the CC BY-SA 4.0 (Creative Commons Corporation, 2013).



Universal asynchronous receiver/transmitters (UARTs) can be considered one of the most essential pieces of asynchronous serial communication. Universal

asynchronous receiver/transmitters are the circuit blocks that mediate between serial and parallel interfaces and implement asynchronous serial communication. Universal asynchronous receiver/transmitters are often built into microcontrollers and flight controllers such as those that will be used as components in the UAV, UGV, and ground control station. Software-based universal asynchronous receiver/transmitters are also possible, but operation of such implementations are processor-intensive and often do not perform as well as a dedicated UART (Sparkfun Electronics, 2012) A simplified diagram of a UART is depicted in **Figure 27**.

When using asynchronous serial communication, there are number of common mistakes that, while easily avoidable, prevent successful operation of serial devices (Sparkfun Electronics, 2012):

- Mismatched RX and TX: connecting RX to RX or TX to TX instead of TX to RX
- Mismatched baud rate: devices attempting to communicate set at different baud rates
- Bus contention: multiple transmitters connected to one receiver - not necessarily fatal to operation, but not recommended

Periphery systems such as radio frequency wireless communication modules, global positioning system modules, and radars often make use of asynchronous serial communication for high compatibility and flexibility. Asynchronous serial communication is also useful for terminal debugging and microcontroller-based embedded software development on devices such as the BeagleBone, Texas Instruments LaunchPad devices, and Nvidia Jetson. Multiple systems will require asynchronous serial communication for successful operation.

4.6 Battery Types

There are a wide variety of battery types and chemistries, both rechargeable and non-rechargeable. Given the fact that the drone will be flown until the batteries are empty quite frequently, non-rechargeable batteries cannot be considered due to the enormous cost of having to replace them every flight. There are still many types of battery when limited to rechargeable ones.

4.6.1 Lead Acid

Lead acid batteries are very popular in applications requiring power for starting internal combustion engines, backup power supplies, or some electric vehicles like forklifts. They are available in sealed types which are very safe and robust, are easy to charge, and are very cheap per watt hour. However, the use of lead means the gravimetric energy density is extremely poor, so they are impractical for an aerial vehicle where weight is of extreme concern (Battery University, 2017).

4.6.2 NiMH/NiCd

Nickel-Metal Hydride and Nickel-Cadmium batteries were popular for mobile applications since they offer much better gravimetric energy density than lead acid, at the expense of more complex charging. NiMH offers better energy density and doesn't have toxic Cadmium unlike NiCd, but NiCd is easier to charge and has a higher lifespan (Battery University, 2017). These were used for electric aerial vehicles but have been replaced by Lithium based batteries in many applications.

4.6.3 Lithium-ion

Lithium ion batteries have taken over many applications since they have the best commonly available energy density, long lifespans, and no maintenance requirement. Unlike the other battery types, they are dangerous if not kept within their operating limits, so they must be monitored to ensure they are not overused. Charging is also more difficult, requiring computer-controlled chargers for best results (Battery University, 2017). Despite the higher cost and additional risk, the high energy density is required for this project.

The complexities of Lithium-ion demand that they have a battery monitoring system that can initiate shutdowns if the batteries begin to overheat. In the SUAS however, a loss of power would cause a serious crash, so the battery monitor must be able to warn the flight system in advance of a problem so that it can initiate an emergency landing then shut down. Since the competition occurs entirely over a clear airfield, landing at the current position is acceptable, so a shutdown can happen relatively soon if a battery warning occurs.

There are several types of Lithium-ion batteries, such as Lithium Iron Phosphate or Lithium Cobalt Oxide. There are also several form factors, primarily cylindrical cells or Lithium-polymer soft, flat pouches. Lithium-polymer batteries are the most commonly used in SUAS applications, because they provide a very high-power density, having discharge ratings greater than 20C. However, cylindrical Lithium cells have a somewhat higher gravimetric energy density (~220 Wh/kg) vs Lithium-polymer (~160 Wh/kg) (AA Portable Power Corp, 2018). The high energy density of cylindrical cells allows for a large improvement in range, since the weight of the craft is so important. The lower discharge rate and per-cell capacity must be made up for by using a larger number of cells, so the cells themselves should be as large as possible to reduce the number of wires required, keeping weight down. A common size for cylindrical cells is the 18650, named for being 18mm in diameter and 65mm in length. A newer cell format is the 20700 cell (20mmx70mm). The larger cell has better gravimetric energy density because the metal casing can contain a larger volume of useful material for relatively less surface area, and the larger cells mean fewer are needed, so less extra copper to connect the cells is required. Since these cells are the best commercially available rechargeable energy density, they will be employed for this design.

4.7 Battery Monitoring

Rotary wing aircraft like the multirotor under construction depend entirely on continued power to the rotors to provide lift. This means that if power to the rotors fails landing safely will be difficult or impossible. For sUAS multirotor craft, there is little inertia to keep the rotors spinning and no way to control the craft without being able to use battery power to change the rotors' spin. As such, the drone should have an accurate means of measuring how much energy is available in its batteries so that it can act to avoid a power loss that would lead to a crash. There are multiple methods for estimating how much energy remains in a battery, the most common of which are battery voltage measurements and coulomb counting. The amount of energy remaining in a battery is often referred to as the state of charge or SOC.

Hardware for measuring battery voltage is extremely simple, requiring only an ADC or a voltmeter. However, converting a battery voltage measurement into an estimate of the remaining charge requires knowing the "discharge curve" of a battery. The discharge curve is the relationship between battery voltage and the state of charge, and it can be very complex. For some chemistries like lead acid, the discharge curve gives a quite linear relationship between voltage and SOC. For chemistries commonly used in electric aircraft like Lithium Ion, the discharge curve is considerably more complex, with sharp changes at the ends, and a very flat section in the middle that means the voltage only changes significantly when the battery is either fully charged or discharged. This not only requires a more complex model of the curve, but also means the voltage measurements must be quite accurate to get an accurate reading of the battery SOC. Battery voltage measurements are further hampered by the fact that batteries have an internal impedance, so batteries under different amounts of load will have different voltage readings, even if they are all at the same SOC.

Coulomb counting is another common method of battery SOC measurement and operates by tracking how much electrical charge has entered and exited the battery. The hardware for this is more complex than for voltage measurements, as a means of measuring and integrating the current going into and out of the battery is required. Due to modern demand for battery SOC measurements, integrated circuits are available which perform one or both functions. Coulomb counting generally provides a better estimate of the SOC of a battery since the SOC is directly related to how much charge has gone in and out of the battery, and most batteries are rated in mAh or Ah, both of which are measures of charge. The coulomb counting method is not as affected by the load on the battery, since the internal impedance does not affect the current measurement, but some effect persists since most batteries can not output as much charge under high loads. The other problem with coulomb counting is that it integrates a value, which can integrate error and requires the ability to set a 0 value. Most solutions overcome this by using a dedicated analog integrator that does not require digital sampling or high precision digital numbers, and by setting the charge to 0 when the battery is completely charged, and representing a discharged battery as having a negative charge in it.

The better accuracy of coulomb counting combined with the low-cost availability of ICs for it from Analog Devices and Texas Instruments make it the better choice for this project, since triggering a landing at the wrong time due to voltage readings skewed by different loads would either cause a crash or a failure to complete mission objectives.

4.7.1 Current sensing

ICs available for coulomb counting are available with either integrated or external current sensing. The ICs with internal current sensing are limited to 1A or less due to the limits of the ICs and their packages. As such, we will need to design our own solution for measuring the current going into and out of the batteries. There are two primary methods for measuring current: resistive and magnetic.

Resistive current sensors take advantage of Ohm's law by inserting a small but accurately known resistance in series with the current to be measured. The system measures the voltage across this resistor and applies Ohm's law to determine the current through it. This sort of sensor is relatively cheap, and offers good accuracy, but has the major drawback of being a resistive element taking the full current of the system. This means that a resistive sensor will dissipate I^2R watts, which in high current systems like a sUAS can be considerable.

Magnetic current sensors like a hall effect sensor take advantage of the fact that a current in a wire will produce a proportional magnetic field around the wire. These sensors measure that field and produce a voltage proportional to the current. The main benefit of these sensors is that they do not cause I^2R losses, and inherently isolate the measurement circuitry from the current being measured, which makes designing measurement for high voltage circuits simpler. However, these sensors are much more expensive, and are not as accurate as a resistive shunt.

Another major difference between resistive and magnetic current sensing that is not inherent to either technology is that existing magnetic sensors are not compatible with existing coulomb counter ICs. Coulomb counter ICs are designed to take the differential voltage measurement across a resistor directly, which magnetic sensors do not provide. Many coulomb counter ICs also use the voltage measurements from the resistor to measure the voltage of the battery, which can be used to provide a second means of SOC measurement.

The incompatibility of magnetic sensors means that for this project resistive sensors are necessary. This means that the I^2R losses must be overcome. Since I is known to be at most 300A, picking a low R keeps losses minimal. Putting multiple resistors in parallel is one way to use available low-value resistors to get a small enough R to keep losses down. Doing so causes problems with circuit geometry, where if the resistors are not properly positioned, they may not share current evenly, so measuring one of them could cause false readings. Analog Devices shows that using a simple set of 1-ohm resistors on the sense lines allows averaging the voltage readings across multiple shunt resistors, so that the overall

R can be low while maintaining accuracy (Devices & Eddleman, LTC4218 12V/100A Hot Swap Design for Server Farms).

4.7.2 Kelvin Sensing

Resistive current sensors rely on measuring a voltage across a very small resistance (possibly less than 1 milliohm) to avoid wasting power and lowering the supply voltage. Sense resistors must be connected to the system somehow, and these connections and the wires going to them also have a small resistance. Since the sense resistor itself has a very low resistance, the resistance of the connections and wires can be significant compared to the resistor. If the extra resistance is included in the differential voltage measurement, the measured voltage will be larger than the actual voltage across the resistor, which would make the current measured larger than the actual value. (Keithley, 2012)

To reduce the errors caused by connection resistance, current sensing resistors employ a technique called 4-wire sensing or kelvin sensing. This technique takes advantage of the fact that the current drawn by the device measuring the voltage across the resistor is extremely low, often less than a microamp. A 4-wire connection passes the large current through one connection while measuring the voltage on a separate, dedicated connection. The current will pass unaffected since the resistances are all in series, and since the voltage sense connections have very little current passing through them, the voltage readings will be much closer to the actual value. Precision resistance measurement and precision resistors will often have 4 terminals dedicated for this purpose.

The budget and requirements for this project do not allow for or demand ultra-high precision resistors, so resistors with 2 terminals will be employed. Furthermore, the resistors will be surface-mount devices to save space and weight, as well as reducing cost. A 2 terminal SMD resistor attaches to a PCB with 2 large pads, which makes 4-wire connections difficult. One solution is to split the 2 large pads into 4 pads, with the pads which carry current being larger than the ones carrying the voltage sense. This is achievable because the resistor does not have discrete pins, but simply 2 large surfaces which can be soldered to the PCB. The arrangement of these pads should then avoid sensing the wrong part of the resistor terminal to produce an accurate sensing voltage. Analog Devices published an article detailing tests done on several pad arrangements (Devices & Marcus, Optimize High-Current Sensing Accuracy by Improving Pad Layout of Low-Value Shunt Resistors). They tested several variants of a common configuration, which simply splits each pad down the middle in parallel with the current flow, producing current supplying pads on one side of the resistor and voltage sensing pads on the other (Devices & Marcus, Optimize High-Current Sensing Accuracy by Improving Pad Layout of Low-Value Shunt Resistors). Analog Devices also tested configurations which made a cutout along the center of the current supplying pads and placed the voltage sensing pads there, so the voltage sensing pads were surrounded by the current supplying pads. The pads were connected with vias at different points to a voltmeter and were measured by supplying a current using a

calibrated electronic load, controlling temperature rise by keeping current brief. They found that the configurations that split the pad down the middle into 2 produced ~4% error, while having the voltage sense pad surrounded by the current sense pad produced <2%. 1% error was obtained by sampling the voltage sense pad from the outside edge instead of the middle of the pad. Measuring across the resistor's high current path produced a 22% error, showing that having any of the 4-wire arrangements produces a substantial increase in current sense accuracy (Devices & Marcus, Optimize High-Current Sensing Accuracy by Improving Pad Layout of Low-Value Shunt Resistors).

As per these recommendations, the current sensing part of the battery monitoring system will be built with a 4-wire sense arrangement with separate voltage sense pads in the middle of the current supplying pads. As per the recommendations for using multiple sense resistors, the voltage sense pads will be connected to the voltage sense lines on the inside edge, which improves trace routing and reduces obstruction of the high-current PCB traces without sacrificing current sense accuracy (Devices & Marcus, Optimize High-Current Sensing Accuracy by Improving Pad Layout of Low-Value Shunt Resistors).

4.7.3 Battery Monitoring Requirements

The battery monitoring system fills the monitoring requirement of the power system, and the current monitoring is the weakest link in terms of meeting the overall power requirements. Having a good BMS is also essential to verifying that the battery system will have enough energy to complete the mission.

- Be capable of measuring and withstanding the full voltage, current, and charge of each battery pack.
- Report voltage, current, and charge in 1/10th increments or less.
- Sense and report battery temperature in 1C increments.
- Be able to report readings to a central computing system.
- Be suitable for integration into the mechanical battery pack.
- Support appropriate connectors for power output.
- Have some means of estimating the remaining energy in the batteries

The battery monitoring system will need to have some component in line with the battery output to sense battery current, so it will need to be rated for the full current of the battery pack. This means selecting a suitable set of shunt resistors that produce an acceptable output for the battery gas gauge IC while not dissipating too much power. The selected IC should report values in 1/10th increments or less, since 1/10th increments allow using these measurements to determine if a pack is merely low or is critically low. The battery monitor needs to report readings to a central computer because that computer will also be issuing commands to the flight system, so if the sUAS needs to land due to a critically low battery, the central computer will need to be the part which makes that decision based on the data. For this purpose, I²C is well suited since it is widely supported, and should allow each battery pack to have a monitoring chip attached to the bus without using lots

of I/O. The battery readings are relatively infrequent, so the bandwidth use should be acceptable.

The battery monitor will need to be integrated into the battery pack, so working with the team designing the mechanical systems is important to ensure the design has a usable footprint and set of mounting holes. This will be accomplished during PCB layout and by selecting components which do not have an excessive footprint, although the size of the battery packs makes the maximum size of the PCBs quite large. Since the battery current will need to pass through the monitoring system, the monitoring system will need to have suitable connections for the current input and output.

On the battery side, soldering the battery tabs to the PCB will provide a low-impedance connection, which does not have to be disconnected since the monitor will stay with the battery pack to track charging and lifecycle data. The output connection will need to be removable so that the battery can be disconnected from the sUAS for recharging and maintenance. The battery monitor should have some means of estimating the remaining energy, whether by simple coulomb counts or by a more complex built in algorithm.

4.7.4 Battery Gas Gauge ICs

TI and Analog Devices have a range of battery gas gauge ICs that measure state of charge through both coulomb counting and voltage, as well as temperature. These ICs report the values over some digital serial protocol, and often have extra features like dedicated alarm pins or per-cell voltage measurement. For our purposes, the IC should be able to sense voltages up to 24V to tolerate the full pack voltage, should have a low shunt resistor voltage to minimize I^2R loss, and should communicate over I²C to ease integration with other sensors. The sensors must also be able to sense the 75A maximum pack current and track the maximum pack capacity of 20Ah. The sensors should not require another IC to perform A/D conversions and should operate with a minimum of outside intervention and setup. The sensor should not incorporate features like battery charging, since this dramatically increases part count and is unnecessary, since a high-power external charger will be used.

4.7.5 LTC2944

The LTC2944 is the only Analog Devices part that fits the requirements. It can operate with up to 60V of input, measures voltage, current, and charge, and reports these values directly over I²C. The LTC2944 can use any sense resistor if the voltage across the resistor is no more than 50mV. Current readings have 12 bits of resolution, voltage has 16 bits, and charge has 16 bits. This gives 65535 counts for voltage and charge, and 4095 counts for current. The charge reading defines the 0 point as half-scale, so there are 32767 counts available if the battery starts at half-scale and discharges from there. These resolutions are more than enough for our application, since the resolution is much smaller than the difference

between thresholds. The voltage readings provide 1.08mV per LSB, which is enough to allow voltage to be factored into the SOC equation when allowable and can sense if the battery pack voltage is dropping rapidly, which would indicate the pack is about to run out of energy (Linear Technology, 2017).

The LTC2944 requires very little configuration to operate. The accumulated charge needs to be reset when the battery is charged, and an 8-bit control register needs to be set. The control register has an ADC mode field, a charge prescaler, a field to configure the auxiliary Alert/charge complete (ALCC) pin, and a field to put the chip in shutdown mode to conserve power. For this application, the chip can be kept in the full-power continuous operating mode since the pack will be charged often and is large enough to make the chip's full power usage insignificant. The prescaler bits are set based on the expected discharge time of the battery, which for this application is well-defined since the motors will use a relatively constant amount of power close to the maximum. The prescaler is used to divide the accumulated charge value so that applications that discharge the battery slowly can be measured more accurately. In this application the prescaler will be lowered since the battery will generally be discharged quickly. The last field is the ALCC configuration, which can configure the ALCC pin to act as an Alert output, a Charge reset input, or disable the ALCC pin.

The main factor when using the LTC2944 is the 50mV current sense differential limit. The drone will consume no more than 75A per battery pack, so the resistor should be set to 50mV/75A, or 666 uOhms. Achieving such a low value requires multiple sense resistors in parallel. The power dissipation is then 50mV*75A or 3.75W. This accounts for 0.25% of the drone's total power consumption at full power and will be reduced if the power consumption of the motors is less. If this power consumption is problematic, the resistor can be reduced in value, but the current sensing will have less resolution.

The LTC2944 communicates over I²C, which makes it simple to use with other sensors, and easy to connect to most embedded systems. The ADC and charge values are reported as raw counts, so the values must be converted by a set of simple formulas on the host computer. The voltage conversion is accomplished by multiplying the reported value times the voltage per LSB (1.08mV). The current conversion is accomplished in a similar way, with the current per LSB being 64mV/R_{sense}, but a half-scale value of 0x7FFF must first be subtracted, since the ADC measures both positive and negative current represented as 0 current being 0x7FFF. The accumulated charge count is also converted by subtracting the 0 reference (set by the user) and multiplying by the charge per LSB, which is 0.340mAh*50mV/R_{sense}*Prescaler/4096. Doing these conversions on the host is simple and allows the final values to be represented with whatever data type is desired. The IC has a fixed address, so if multiple LTC2944s are to be used, they must be behind an address translator. The LTC2944 gives 400KHz as the minimum maximum I²C speed, with parts typically able to reach 900KHz. This means the LTC2944 can operate on a fast mode bus, which reduces the chance that the bus will run out of bandwidth.

The LTC2944 has the additional features of temperature measurement and the alert system. The temperature measurement is a simple on-die temperature sensor read to 11 bits which is read by multiplying the reading by 7.78mK (millikelvin) per LSB. This value can then be converted to Celsius or Fahrenheit in the regular way. The LTC2944 alert system allows the user to set high and low limits on all 4 of the measured values. When these values are exceeded, the LTC2944 will assert the ALCC pin if it is set to alert mode, and will respond to I²C address 0x0C, which when read will allow the host to read back the LTC2944's regular address. In systems where several sensors have alert support like this, the host can simply read 0x0C to check if a sensor has an alert and will get the address of the sensor that has the alert if one does. The ALCC pin can also be connected to an interrupt or LED indicator.

4.7.6 TI BQ34110

The TI BQ34110 meets the requirements for a gas gauge as well. It operates like the LTC2944 but has additional features to better estimate the SOC as well as overall battery health. As a result, configuration is more complex, and the IC does not generalize to higher currents and battery capacities as well as the LTC2944 does. Operating at voltages and currents above the standard values is possible but requires additional configuration and programming work.

By default, the TI BQ34110 will measure voltages up to 65535mV, currents up to 32767mA, and capacities up to 29 Ah. Since the drone will use currents up to 75000mA, the BQ34110 must be configured differently. This is accomplished by using an appropriate sense resistor and changing the calibration process such that the "ground truth" current is reported as being scaled down such that the current and capacity values will fit, then multiplying reported capacity and current in the host by the same scaling factor. The largest scaling factor between current and capacity must be used for both. For voltages over the maximum, a voltage divider can be introduced between the battery and voltage sense pins, and then the voltage measurement adjusted similarly to the current measurement.

While the BQ34110 has an expanded +/- 125mV range for the sense resistor input compared to the +/- 50mV input of the LTC2944, the sense resistor voltage must still be kept low to avoid dissipating too much power, especially in the drone (Texas Instruments, 2018).

The BQ34110 has many additional features that the LTC2944 does not. The most important of these is that the BQ34110 supports features for learning the real capacity of the pack and can compensate for the effects of battery load, temperature, and self-discharge to improve the accuracy of the remaining capacity estimate. The capacity estimate fuses both coulomb counting and voltage measurements to improve the results. The BQ34110 can learn the battery capacity by measuring how many mAh the battery could supply during the most recent full discharge and incorporate this with the design capacity. The BQ34110 also has discharge curves programmed in for common battery chemistries that allow it to

estimate the amount of charge a pack has when it is first plugged in, without having to wait for the battery to be fully charged like the LTC2944 does.

The BQ34110 has modes which enable it to control the battery charging and to measure the capacity of batteries which are left to sit for long periods of time (such as battery backups), but these are not needed in this project since the drone will be charged by a dedicated external charger and discharged frequently and deeply.

Another additional feature of the BQ34110 that may be useful is its ability to record statistics about the life and health of the battery. The BQ34110 can be set to record the number of cycles, most extreme voltages, temperatures, and currents seen on the battery pack. This can allow the drone to know if its packs have been abused in the past and estimate if the packs need to be replaced. The BQ34110 can also measure the pack resistance by monitoring it during charging and operating a test discharger to determine the true pack health. This can make better decisions on battery replacement but requires the BQ34110 to manage charging and to be connected to a discharging circuit that can handle 1/20C discharge or more.

The BQ34110 supports an I²C interface that runs at a maximum of 400KHz. This allows it to also be used in fast mode busses but cannot tolerate anything more. The BQ34110 gives another advantage over the LTC2944 in that it supports an external NTC thermistor for battery pack temperature sensing, unlike the LTC2944's on-die temperature sensor. This allows the temperature sensor to be placed inside the pack itself, which can then provide a much more useful reading of the temperature of the hottest part of the battery pack. Measuring the hottest part of the pack allows safer operation, since the sensor will detect overheating more reliably, and can then trigger a response in the drone to reduce its power consumption to reduce battery heating.

4.7.7 TI BQ34Z100-G1

The TI BQ34Z100 (Texas Instruments, 2018) uses a proprietary algorithm to estimate the SOC by comparing the open circuit and loaded voltages of the battery cells. The IC trades some of the charge and discharge control features of the BQ34110 for improved SOC estimation and the ability to control several LEDs that can indicate the battery SOC to a user without needing a host CPU to read it. The BQ34Z100 also supports an external NTC thermistor for temperature reading like the BQ34110 does.

Like the BQ34110, the BQ34Z100 requires calibration to function, since it performs the conversions from ADC and accumulator counts to mA, mAh, and Volts internally. TI provides evaluation software which takes data about voltage dividers and the sense resistor to automatically set the calibration registers in the device. The BQ34Z100 has a similar +/- 125mV sense resistor voltage limit to the BQ34110, but again the values must be kept below this to keep sense resistor power dissipation low.

The BQ34Z100 also supports reading over I2C (Texas Instruments, 2018). This shares pins with the LED outputs, but these are unnecessary for the drone system. The HDQ interface is also supported, but this won't be used since the device will be part of an I2C bus and there is no requirement to use a 1 wire interface.

Another advantage of the BQ34Z100 is that it provides additional features for operating at high currents and voltages. Since the BQ34Z100 expects a voltage divider for all cells higher than 1S, it has better support for setting up a voltage divider on the input in its software. It also has easier to use settings for the current sense resistor as well and has a flag which can be set to indicate to the host that data is being scaled when the design current is above the normal 32-amp limit. The BQ34Z100 has a feature which can disable the battery voltage divider to save current, but seeing as this current is <100 microamps, the voltage divider will be left always connected, since the shutoff requires a pair of ESD-sensitive MOSFETS and extra resistors, for minimal power savings compared to the pack capacity.

4.8 Voltage Regulation System

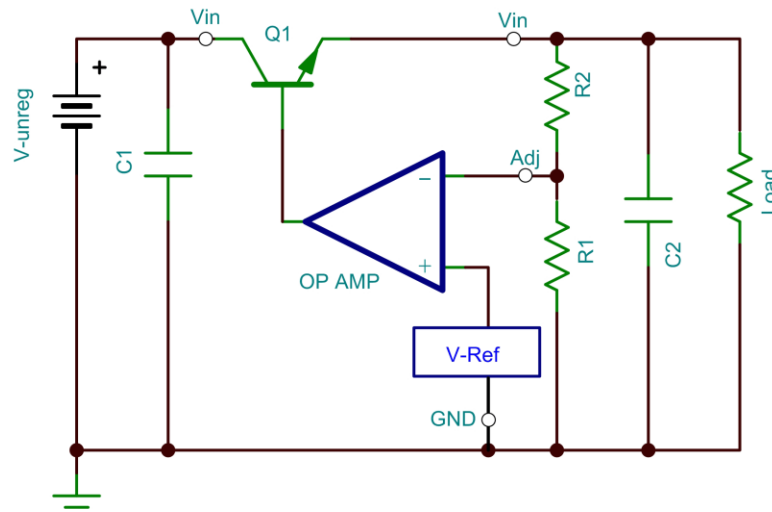
The logic systems on the drone require power that is of a regulated voltage and is protected from spikes and excess noise created by the motors. The requirements for this project call for 25 watts of continuous 12V power, 20 watts of continuous 5V power, and 8 watts of continuous 24V power. The vehicle batteries are the only energy supply on board, so the voltage regulation system will need to accept the 22.2V nominal power from the batteries and convert it to something usable by the other systems. However, the batteries can exceed 24V when fully charged or if the motors slow down quickly and can go lower than 22.2V under heavy load or when somewhat discharged, which complicates the operation of all regulators, especially the 24V regulator.

There are both linear and switching regulator designs available that can accomplish the goal, so the design must be chosen to be efficient, light weight, and practical to assemble and test.

4.8.1 Linear Regulators

A linear voltage regulator is a type of voltage regulator that outputs a steady DC voltage, hence the linear label. Linear voltage regulators use an operational amplifier to create a closed feedback loop to maintain a voltage output reflective of the reference voltage, as long as the input voltage is above a certain threshold. An example of a linear regulator can be seen below.

Figure 28. Diagram for a discrete linear regulator

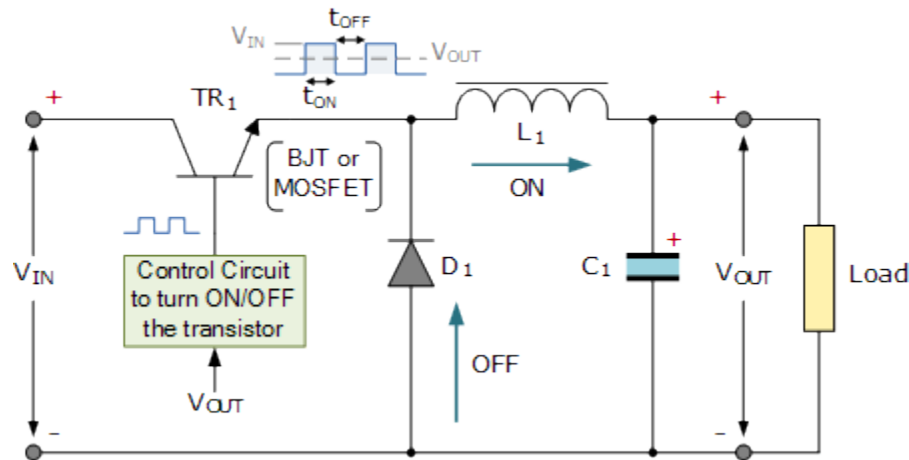


These Linear voltage regulators are generally chosen for their steady DC output and simplicity but using them does come with drawbacks. Linear voltage regulators regulate voltage by dissipating the extra voltage as heat, so they tend to be inefficient and require cooling. The power dissipated is equal to the difference between the input and output voltages times the current, so feeding the regulator from a voltage closer to the output can reduce the power wasted. These linear regulators are step-down regulators, meaning that that the output voltage of the regulator will never exceed its input. By design, linear voltage regulators have a dropout voltage determined by components in the circuit. The dropout voltage must be exceeded by the input to achieve the desired output. As a result, working with these regulators requires accounting for this dropout voltage. Regulators designed to have a low dropout voltage of less than a volt are available, and using them is advantageous to reduce power dissipation in the regulator.

4.8.2 Switching Regulators

Switching regulators are regulators that use inductance to maintain a steady output voltage while the regulator turns on and off, hence the “switching”. An inductor is connected before the output of the regulator, which allows the output to be a steady current despite the frequent switching – as the current flowing through an inductor cannot change instantaneously. The main advantages of the switching regulator over the linear regulator are: Lower voltage drop and increased efficiency.

Figure 29. Circuit for a discrete switching regulator



Due to the switch element being in either the conducting or non-conducting state, switching regulators do not dissipate nearly as much power as linear regulators. Instead of dissipating excess voltage as heat, the voltage is dropped by storing energy in the inductor and allowing it to be released at a known rate. Through this method, switching regulators can reach efficiency ratings of around 90%. A not so great result of the “on and off” working of the switching regulator is output ripple. This ripple is the peak-to-peak difference of the current inside the inductor as the regulator switches from state-to-state. To minimize the ripple current, an inductor with a large impedance is necessary.

While switching regulators are more efficient, they are also more complex and generally require more components than linear regulators. Linear regulators also don't have to deal with ripple current but have more heat and power drop-off. Depending on the use-case, one major benefit of linear regulators is the lack of noise. The rapid switching in switching regulators introduces noise into a system, so a linear regulator may be better on some systems. On the other hand, switching regulators can also be designed as a step-up regulator, meaning the output voltage is larger than the input. Ultimately, both regulators have their uses, and whether or not they should be used is dependent on a number of factors.

4.8.2.1 Buck Regulators

Buck regulators are DC-to-DC switching regulators that regulate the output voltage to a value lower than the input. For this reason, they are also referred to as step-down converters. While the output voltage is decreased, buck converters “boost” the input current of the circuit to a higher value at the output. Buck converters are highly efficient, with peak efficiency ratings being roughly 96%, so one can make the most out of their power supply

4.8.2.2 Buck-boost Regulator

A buck-boost converter is a DC-to-DC switching regulator that output a voltage with a larger magnitude than the input. While standard switching regulators use a

singular inductor, buck-boost converters use 2. One inductor or is for the “buck” mode and the other for the “boost”. Buck-boost regulators use another switching method to charge a capacitor, that when fully charged will have a larger voltage than the input.

4.8.3 Voltage Regulator Components

After going through the two main types of voltage regulators, and looking into specific converters, we decided that the best regulator for our UAV is the Buck Converter. Originally, we were under the impression that our UAV would need a step-down and a step-up converter, so we planned to design both: a buck converter and a buck-boost converter. However, after recalculating the power consumption of our connected devices, we came to the conclusion that only the buck converter was needed.

4.8.3.1 LMR33630 Buck Converter

For our voltage regulation system, the regulator we chose was the LMR33630 from Texas Instruments. Using recommended switching regulator designs as a baseline, we chose this regulator due to its efficiency, which is above 95% in an ideal case. The LMR33630 is a buck converter and can accept a maximum voltage of 36V. The output voltage of the regulator is dependent on what resistors we connect to it, but at max the LMR33630 can step down to 24V. The output current of this regulator is designed for 3A. So, by choosing the correct resistors for the reference voltage of the regulator we have the ability to get the total desired wattage at the output.

4.8.3.2 SM74611 Smart Bypass Diodes

To supply power from multiple battery sources, our voltage regulation system needs a means of “switching” which battery is being used. For switching, our best option is to use diodes that connect in series to our batteries. One concern, however, is the forward voltage of the diode. In most cases diodes have a forward voltage of 0.6 – 0.7V. Using a standard diode would drop our nominal voltage from 22.2V to 21.6V. While this change is not a huge drop-off, our efficiency would decrease, nonetheless. Our solution to limit the drop-off of our system is to use the SM74611 smart bypass diodes from Texas Instruments. These bypass diodes are designed to be “ideal”, meaning the drop-off from using them is near zero. With a forward current of 8A, the SM74611 has a forward voltage of 26mV. These ideal diodes allow our voltage regulator to have a switching system with negligible voltage drop.

4.9 Electric Motors

Electric motors are the primary source of electromechanical excitation for many unmanned aerial vehicle mobility systems, as well as auxiliary actuation systems.

Since the unmanned aerial vehicle for this project will be primarily powered by electricity, electric motors will provide the propellers with the rotational torque necessary to produce upward thrust. The multidirectional mobility of the unmanned aerial vehicle will be enabled by multiple (eight) electric motors. Energizing each motor at different levels causes imbalanced thrust and torque across the chassis of the unmanned aerial vehicle, allowing the unmanned aerial vehicle to tilt, rotate, ascend, descend, and strafe with multiple degrees of freedom.

Electric motors are chosen over combustion engines and compressed gas systems due to a significantly more optimal combination of thrust-to-weight ratio, placement flexibility, and system simplicity. The relative simplicity and robust operation of electric motors also make them more modular; failure of an electric motor can be easily remedied by full replacement and rewiring of the system without significant mechanical rebuild. The primary energy storage of an electric unmanned aerial vehicle can theoretically be placed nearly anywhere on the chassis because power can be transmitted via electric wiring, which is significantly easier to account for in relation to mechanical linkages for combustion engines.

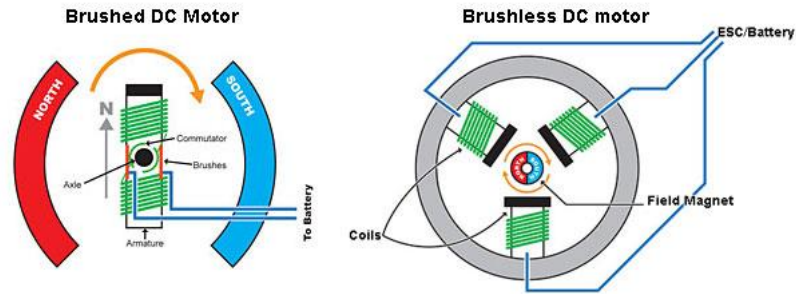
Also, in comparison to combustion and compressed gas systems, electric motors are much easier to control precisely and with minimal latency. These characteristics are crucial for the control of multi-rotor unmanned aerial vehicles. When considering all of these factors, it is also helpful to evaluate different types of electric motors as well as the tradeoffs between each for an unmanned aerial vehicle.

4.9.1 DC Motors

DC motors are generally simpler when compared to AC motors and generally cheaper in initial costs, especially for lower power systems. For these reasons, DC motors were widely used before AC motors, and are still used today, especially for simpler systems. Some of the benefits of DC motors over AC motors include (billd700, 2016):

- Ease of installation
- Wide range of speed, controlled by voltage
- Quick to start
- Quick to stop
- Rapid acceleration and reverse
- High-torque from stop
- Speed-torque curve is linear

Though DC motors are well-suited for low-cost applications, including toy drones, the power and thrust requirements of the unmanned aerial vehicle for this project are not likely to be met with a DC motor solution that can be obtained in a cost-effective way. Though there are many types of DC motors available, they can be separated into two primary categories: brushed and brushless. A diagram of these two types of motors and their construction can be found in **Figure 30**.



[This Photo](#) by Unknown Author is licensed under [CC BY-SA](#)

Figure 30. *Brushed DC motor vs. brushless DC motor. The use of this image is governed by the CC BY-SA.*

4.9.1.1 Brushed

Brushed motors are the more traditional of the two and are often used in lower cost applications with simple control systems. There are multiple types of brushed motors. A selection of some brushed motor types their characteristics are outlined in as follows (billd700, 2016):

- **Series Wound**
 - High starting torque
 - Poor speed control
 - As torque from load on motor increases, speed falls
- **Shunt Wound**
 - Medium starting torque
 - Constant speed
 - Can increase motor current for higher torque without speed reduction
- **Compound Wound**
 - Compound of series and shunt wound motors
 - High starting torque
 - Smooth operation
- **Permanent Magnet**
 - Low torque
 - High precision

When compared to brushless motors, brushed motors tend to have a shorter life span, especially for high-use applications (billd700, 2016). This makes brushed motors less suitable for unmanned aerial vehicles, where electric motor reliability and precise is absolutely critical for safe and successful operation.

4.9.1.2 Brushless

Brushless motors are mechanically much simpler than brushed motors, but are also more difficult to produce, leading to higher initial cost. Many brushless motors use Hall Effect sensors to detect rotor position. The motor controller of a brushless

motor is able to use this information to more accurately control motor speed via adjustment of the current. In addition, brushless motors also have longer lifespans and minimal maintenance requirements, as there are no brushes to wear out. Efficiency is also often greater than 85%. Although initial costs for the brushless motors and their controllers are higher, the greater reliability, ruggedness, and control precision of brushless motors make them far more suited for unmanned aerial vehicle thrust operations (billd700, 2016).

Stepper motors are another type of brushless motor that is often used in position control applications, such as linear actuation and coordinate-based movement (billd700, 2016). Although the use of stepper motors will be minimal for unmanned aerial vehicle thrust operations, they could be useful for auxiliary motions such as the activation of various interlocks or positioning of chassis components.

The combination of characteristics presented by brushless DC motors make them ideal for unmanned aerial vehicle thrust. With the availability of electronic speed controllers to supply precisely-metered power to motors, brushless motors are the clear choice for this application.

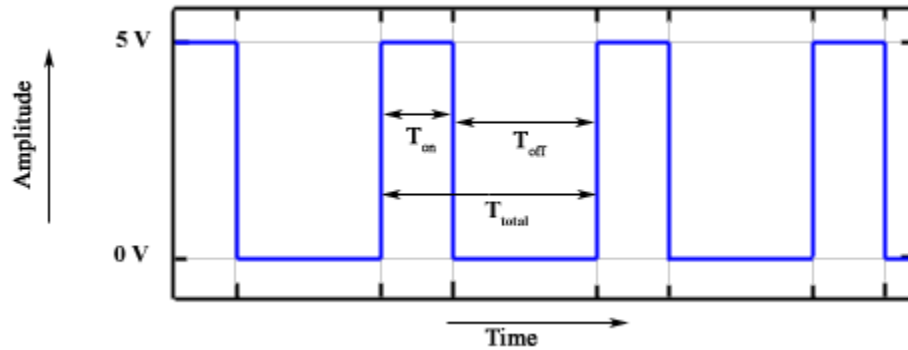
4.10 Electronic Speed Control

A system like the sUAS requires the ability to rapidly and accurately change the speed of its motors based on some control loop. The flight controller uses the orientation to figure out how much thrust is required from each rotor and sets the rotor speeds accordingly. The Electronic Speed Controllers (ESCs) are responsible for making the rotors spin at the commanded speed. There are a couple of basic types of ESC, and different designs of each type can have many different features.

4.10.1 Brushed Speed Control

A brushed speed controller is the simplest type and has different configurations of power switching elements based on what power supplies are available and whether the device needs bidirectional control. In the simplest terms, speed control is just turning the motor on and off at a high rate. If the motor were left on 100% of the time it would turn at a maximum rate. If it were left on for 0% of the time, then it wouldn't turn at all. Speed control comes in when the motor is turned on for some percentage in between. This can be done using Pulse Width Modulation (PWM) which is essentially just a rectangular wave where the percentage of the time that wave is high can be adjusted. The easiest example of this is a square wave where the signal is high 50% of the time and low 50% of the time. In this case the motor would turn at 50% speed. The whole concept driving this is RMS. In a rectangular wave (which PWM is) the RMS value is simple the percentage of the period the signal is high time the maximum value of the signal. For example, if the peak value of a square wave is 12V then its RMS value will be 6V.

Figure 31. Simple PWM diagram. The use of this image is governed by the CC BY-SA.



[This Photo](#) by Unknown Author is licensed under [CC BY-SA-NC](#)

Figure 31. Simple PWM diagramclears PWM up a bit. Using the notation from the figure the duty cycle is defined as follows.

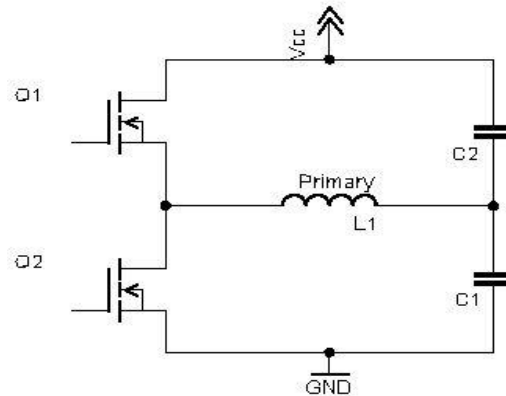
Equation 5. Equation for duty cycle as a percentage

$$Duty\ Cycle\ (\%) = \frac{T_{on}}{T_{total}} \times 100$$

To go back to the square wave example, if T_{on} is half of T_{total} (the definition of a square wave) then it is obvious that the duty cycle would be 50%. Varying the duty cycle varies the RMS voltage and the speed of the motor. Of course, the most devices that generate and control PWM are low power microcontrollers which cannot generate the power necessary to actually drive the motor. To get around this issue the PWM input is sent to some sort of power circuit. The most common circuits are the half H bridge and full H bridge.

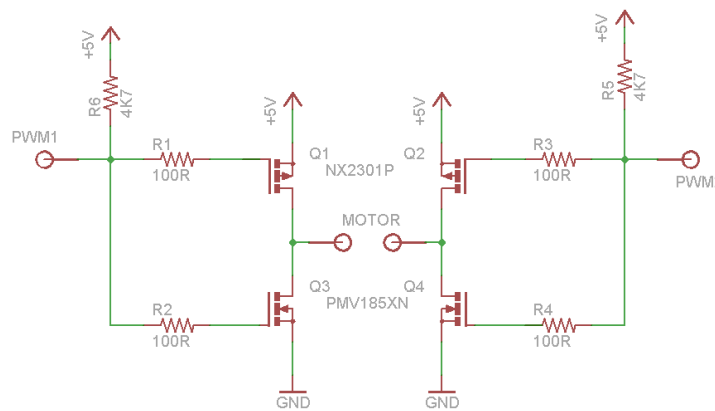
The half H-bridge is a simple two transistor circuit with two inputs. In **Figure 32.** Simple Half H-bridge circui the inductive load in the middle is representative of the inductive load of the motor. A half H-bridge can only be driven in one direction so forward and reverse functionality is desired on the motor a full H-bridge is required.

Figure 32. Simple Half H-bridge circuit. The use of this image is governed by the CC BY-SA.



As seen in **Figure 33**. Full H Bridge circuit. The use of this image is governed by the CC BY-SA., full H-bridge is essentially just two half H-bridge circuits put together. The full H-bridge circuit has the added benefit of being able to be run in reverse. Because it has two PWM inputs the polarity can essentially be switched. In order to reverse the direction of the motor, the PWM1 input gets sent to PWM2 and the PWM2 input gets sent to PWM1. The full H-bridge is also capable of driving more current which makes it necessary for higher current applications.

Figure 33. Full H Bridge circuit. The use of this image is governed by the CC BY-SA.

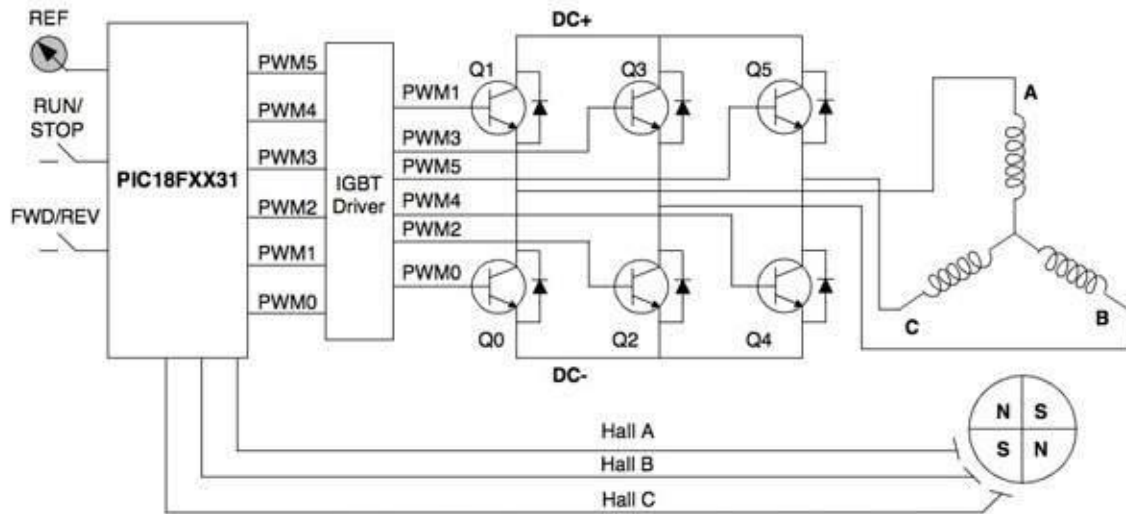


4.10.2 Brushless Speed Control

A brushless speed controller is much more complex, having a more rigid hardware configuration but an even wider variety of software configurations. This is because a brushless motor doesn't have the brushes to control position and phase, so the controller must sense the motor position and phase. This is not mechanically sensed, but by observing the back electromotive force. The only energizes two of the three leads going to the motor at any given time. The third lead is still connected to a coil (or multiple coils) and as such experiences a voltage due to the spin of the

motor. This is known as the back electromotive force and because it is a function of speed it can be used as feedback to control the speed of the motor.

Figure 34. Basic diagram for a brushless motor and controller. The use of this image is governed by the CC BY-SA.



[This Photo](#) by Unknown Author is licensed under [CC BY-SA](#)

The above figure includes Hall sensors to determine motor position. These are not used in most modern ESCs for drones, but the rest of the circuit is approximately correct. The controller uses a network of MOSFETs to effectively create three phase AC power. An oscilloscope output of the PWM will show that it is much more blocky than common sinusoidal AC three phase, but the current does alternate between high and low and the three lines are 120 degrees out of phase. The three lines are 120 degrees out of phase because the physical motor coils are 120 degrees out of phase. The motors that will be used on the drone are lightweight, high output brushless motors so this type of ESC will be used in the project

4.10.3 ESC Comparison

The desired ESC is small, lightweight and can handle 35 amps continuous at the 6S voltage of 22.2V, with the ability to handle higher peak currents up to 40 amps for a short time. Due to recent interest in multirotor drones, these types of ESCs are widely available commercially.

Table 22. Comparison of commercially available ESCs

	Max Burst Current (A)	Max Continuous Current (A)	Accepted Battery Voltages	Weight (g)	Cost (\$)
iPeaka	52	45	3s-6s	4.6	14.75
NIDICI	45	40	3s-6s	8.4	13.75
Wraith 32	45	35	3s-6s		Donated
ReadyTosky	60	40	2s-6s	26	12

All the listed ESCs operate on BLHeli_32 which is the most recent version of BLHeli and one of the most common firmware found on drone ESCs. It is also one of the most feature rich firmware available, as it runs on relatively high performance 32-bit STM32 microcontrollers, so it can support the fastest available communication protocols, and has the most options for things like regenerative braking and improved sensorless operation.

Table 22. Comparison of commercially available ESCs shows many common ESCs. All of them have very similar specifications and are approximately the same cost. Continuous current and price are loosely correlated, but higher power ESCs are not significantly bigger or more expensive. They are also becoming more common as racing drones continue to increase current demands. It is also worth noting the variance in weight. The weight includes the weight of cables which is significant since most ESCs uses 10-gauge wire on the output. The difference in cable lengths largely contributes to the weight. The Wraith 32 meets the necessary requirements because it can handle 35A continuously and 40A for a few seconds. It was offered as a donation and is therefore the first choice for this project.

4.11 I2C Address Conversion Board

This board is necessary because the TI BQ34Z100 battery monitor chips have a preprogrammed I2C address that cannot be changed. Of course, multiple I2C devices with the same address can't be placed on the same bus, and, as mentioned in the earlier I2C section, this issue must be addressed by either using an address converter or some sort of I2C multiplexing. There are multiple chips on the market that provide both types of functionality.

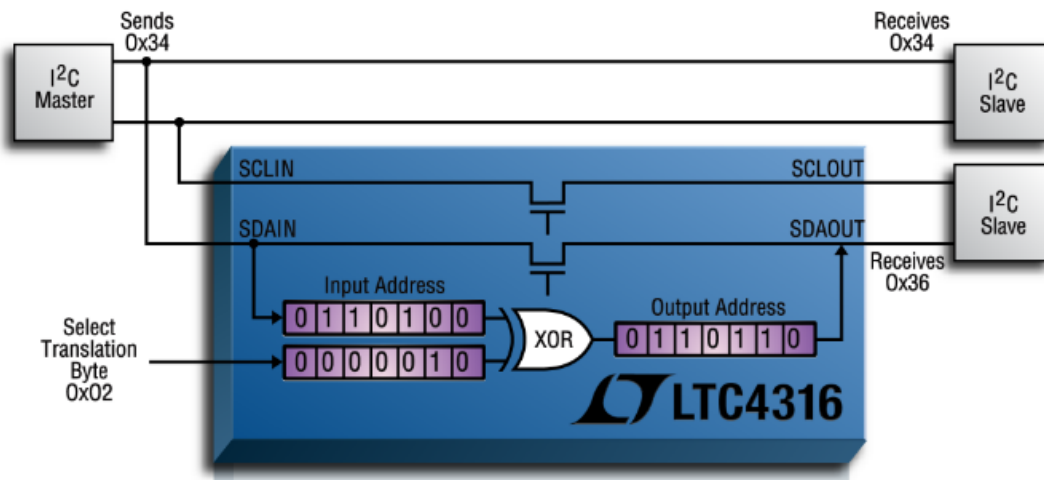
4.11.1 LTC 4316-18

The LTC chips are probably the most straight forward version of address conversion. When an address is transmitted the LTC chip takes the address and converts it to another address by XORing it with an 8-bit binary value called the translation byte. It is important to note that only 7 of those bits have the possibility of changing value since I2C uses 7-bit addresses (typically) with one bit for read or write. Configuring these devices is relatively easy. A random address that is not used needs to be picked, then the translation byte needs to be calculated such

that the generated address XORed with the translation byte yields the desired fixed address.

This solution is very scalable because the translation byte can be any value leading to 127 possible addresses which is the maximum number of addresses that can be handled on 7-bit addressed I2C anyway. The manufacture, Linear, offers this device in three configurations (hence the three part numbers). The LTC4316 is the single package version of the chip, the LTC4317 is Y connected dual, and the 4318 is just two LTC4316's in one package.

Figure 35. Basic block diagram describing how the LTC4316 operates (Permission requested from Analog Devices)



4.11.2 TCA9544APWR

The TCA9544APWR is an I2C bus multiplexer with some additional features. The TCA9544APWR operates like a normal I2C device in the sense that it has an address (0x70 default), and an accessible register that can be written and read. In this “control” register there are four bits for the interrupts and three bits for multiplex selection. Of course, three bits is enough to select up to eight different values, but in this case, it is just four. The selected output of the multiplexer can be decided using the following table.

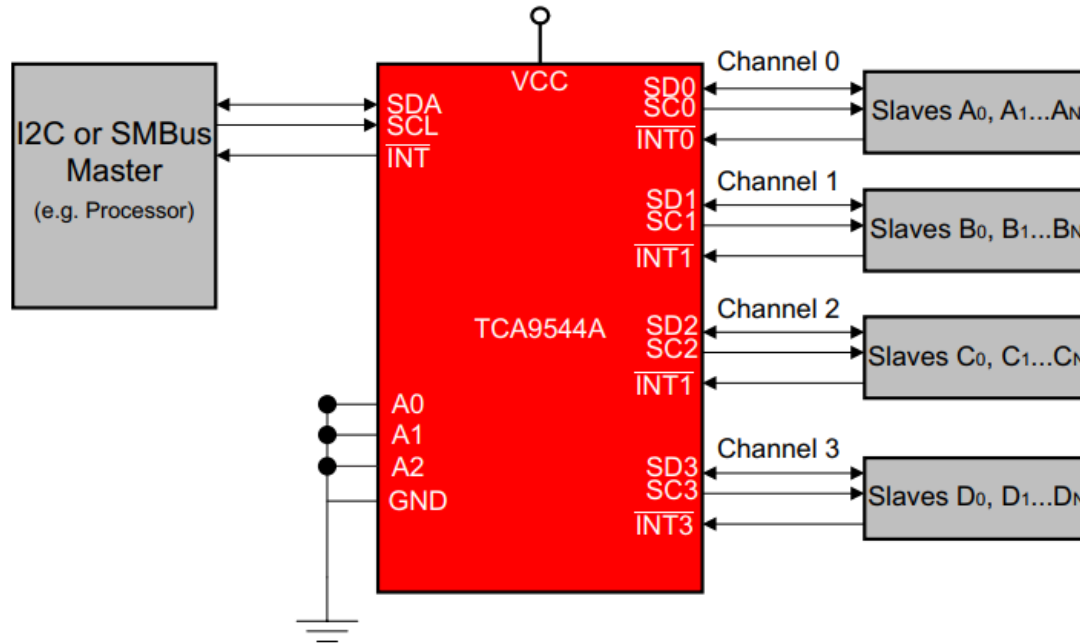
Figure 36. Control register inputs (table courtesy of TI)

INT3	INT2	INT1	INT0	D3	B2	B1	B0	COMMAND
X	X	X	X	X	0	X	X	No channel selected
X	X	X	X	X	1	0	0	Channel 0 enabled
X	X	X	X	X	1	0	1	Channel 1 enabled
X	X	X	X	X	1	1	0	Channel 2 enabled
X	X	X	X	X	1	1	1	Channel 3 enabled
0	0	0	0	0	0	0	0	No channel selected, power-up default state

(1) Only one channel may be selected at a time.

Once these values are known the process is rather easy. Write to the multiplexer control register to select the channel of the device to communicate with. Write the proper values to that register for communication with the desired channel and device. Once the control register has been configured to the desired settings, the connected device can be communicated with normally using its preprogrammed address.

Figure 37. Functional diagram for the TCA9544A (figure courtesy of TI)



The TCA9544A does have a few more features than basic address multiplexing. It provides the ability to vary the address from 0x70 to 0x77 which allows for 8 multiplexers to be on any single bus. With four devices connected to each multiplexer that of course allows for up to 32 identical devices to be connected to a single bus. This chip also offers interrupt functionality which can be used with low power devices to bring them out of sleep mode. In addition to these external features the chip is ESD resistance and also provides voltage isolation between SDA, SCL and SDX, SCX.

These parameters along with the fact that the device accepts 1.65V – 5.5V is why this device was chosen for the project. It is a simple chip that provides the needed functionality as a low cost.

4.12 Flight Control Unit Considerations

For our flight control unit, we chose to go with the BeagleBone Blue. Given the scope of our project the BeagleBone Blue is the ideal computer for satisfying our requirements compared to our other options.

4.12.1 BeagleBone Black

The BeagleBone Black is a potential flight controller and is appealing because of the similarities between it and the BeagleBone Blue – while being a cheaper price. The BeagleBone Black is only \$55, but as a result of it being cheaper, while being manufactured by the same company (Texas Instruments), it doesn't fit our use case as well as the BeagleBone Blue. (beagleboard.org, 2018)

4.12.1.1 Servo Output Pins

The main difference between the two boards for our uses is the lack of dedicated servo pins on the BeagleBone Black. The BeagleBone Blue has 8 dedicated 3-pin servo motor outputs, four DC motor outs and four quadrature encoder inputs, which are enough to support our octocopter without having add anything extra or split connections. On the BeagleBone Black however, there are no dedicated servo output pins. Ultimately, connecting the servos would be possible, but that would require extra wiring, PCBs, and programming – which would ultimately increase the cost of our project timewise and component wise, so the BeagleBone Blue is a better board in this aspect.

4.12.1.2 Wireless Connectivity

Like the BeagleBone Blue, the BeagleBone Black has wi-fi and Bluetooth capabilities. Like the servo connections, the standard BeagleBone Black does not have these wireless connections readily on the board so it would take extra work to get those functions available. To bypass that effort, a version of the BeagleBone Black exists that comes with these connections out of the box. The BeagleBone Black Wireless replaces the ethernet port on the standard BeagleBone Black with a wi-fi module that comes with Bluetooth, but it is also more expensive. At \$76 the BeagleBone Black Wireless is only 4 dollars cheaper than the BeagleBone Blue – which just about removes the advantages of choosing it over the BeagleBone Blue.

4.12.1.3 USB Connections

Both the BeagleBone Black and the BeagleBone Blue have 2 USB connections. Both boards have a USB Type-A connection that can be used to transfer data to and from the board. Both board's Type A connectors are USB 2.0, so they share the same bandwidth – neither is better than the other. Regarding the second USB connection, the BeagleBone Black has a miniUSB port while the BeagleBone Blue has a microUSB port. Both ports are potential connections for power to their respective boards, and each port is USB 2.0 rated for 500 milliamps. With both boards having the same USB capabilities, neither board really edges the other out. However, microUSB is a more popular option over miniUSB because it's smaller.

4.12.1.4 Power Sources

Mentioned in the section above, both the BeagleBone Black and the BeagleBone Blue can be powered by USB ports. Aside from the USB connection, the

BeagleBone Black can be powered from a DC jack or from an external source via expansion headers. The DC Jack on the BeagleBone Black is a 5-volt DC input that pulls 1-amp of current on average. The BeagleBone Black can also be powered via the expansion headers, but this would be inefficient due to the components and time required to make it work (like the potential servo connections on this board). On the BeagleBone Blue, there are three ports for potential power: a microUSB port, a 9-18-volt DC Jack, and a 9-16-volt, 2-cell LiPo battery JST-XH connector. For our project, we will end up using the DC Jack to power our flight controller, but the option for another connection gives us the potential to power sources if that becomes required. The benefit of the BeagleBone Black is that it takes only 5 watts to power but given that our system requires a maximum load of 6 kilowatts, a few watts difference is negligible when designing the battery system. The load from the BeagleBone is more important when designing the logic power regulators but given the space constraints a suitable regulator should be possible for either model.

4.12.1.5 Memory and Storage

Both flight controllers come out of the box with onboard memory and storage. The BeagleBone Black comes with 512 megabytes of DDR3L memory (DDR3L memory is a low-voltage version of DDR3) that cycles at 606 megahertz and has 2 gigabytes of 8-bit eMMC (embedded multimedia card) flash storage that uses NAND chips. For memory, the BeagleBone Blue uses 512 megabytes of standard DDR3 that runs at 800 megahertz. The built-in flash storage for the BeagleBone Blue is twice that of the BeagleBone Black: 4GB of 8-bit eMMC.

4.12.2 Pixracer

One popular option for flight controllers is the Pixracer. In the early stages of our project, our group used a Pixracer as our flight control unit to familiarize ourselves with building and programming a drone. With the Pixracer

Going from a hexacopter to an octocopter created a requirement for 8 servo connections, as opposed to 6, which the Pixracer could not satisfy. Comparatively, the BeagleBone Blue has the 8 necessary ports along with extra features, all while being a cheaper board – (\$100 vs \$80).

4.12.2.1 Processor

The Pixracer uses is a 180 MHz ARM Cortex® M4 with a single precision FPU (Floating Point Unit). In contrast, the BeagleBone Blue comes with an AM335x 1GHz ARM® Cortex-8 processor. (PX4 Dev Team, 2018) Compared to the 180 MHz processor on the Pixracer, the BeagleBone Blue has a 550% increase in clock speed.

4.12.2.2 Memory and Storage

For memory, the Pixracer has 256 kilobytes of SRAM (static RAM: The RAM doesn't have to be refreshed). Choosing the BeagleBone Blue over the Pixracer increases our total memory by a magnitude of 2^{11} , or approximately 2000%. The Pixracer comes with 2 megabytes of flash storage, so there's also a 2000% increase in total flash storage, as the BeagleBone Blue comes with 4 gigabytes.

4.12.2.3 Servo Outputs

On the Pixracer there are 6, 3-pin PWM outputs, which are used for servo/motor control. For the scope of this project, 8 servo outputs are required to communicate with the 8 motors on our octocopter. For this reason, the BeagleBone Blue, with its 8 servo outputs, satisfies the requirements for our drone.

4.12.2.4 Connectivity

On the Pixracer, the only USB port is the microUSB used to power the board. The primary way the Pixracer communicates with outside sources is through wi-fi. The wi-fi module included in the Pixracer is the ESP8266. This module works on the 2.4GHz bandwidth, so it is subject to interference from outside sources. The BeagleBone Blue comes with 2.4GHz wi-fi, Bluetooth, microUSB and a USB Type-A port. These four resources can be used to send data from the flight controller.

4.12.3 BeagleBone Blue

Figure 38. Image of the BeagleBone Blue (Image under creative commons, courtesy of beagleboard.org)



Our flight controller of choice is the BeagleBone Blue from Texas Instruments. Aside from being donated to us, the BeagleBone Blue satisfies our need for a flight controller. The BeagleBone Blue will also be useful for the UGV, which requires a few servo outputs and GPS navigation. Using the same board for both platforms simplifies the configuration of software. The BeagleBone Blue is a Debian Linux-based robotics computer. The board comes with an Octavo OSD3358 SiP (System-in-Package) that's equipped with:

- TI AM335x ARM® Cortex-A8 processor with a 1GHz clock
- 64KB L1 cache (32KB D-cache, 32KB I-cache)
- 256KB L2 cache
- 64KB on-processor L3 RAM
- 512 MB DDR3 RAM
- 2, 32-bit Programmable Real-time Units (PRU)

The board itself comes with:

- Wi-Fi and Bluetooth modules
- 8, 3-pin servo motor outputs
- USB Type-A and a microUSB port
- Sensors (accelerometers, gyros, barometer, etc.)
- JST interfaces for GPS, UARTs, SPI, and more

(beagleboard.org, 2018)

Table 23. Comparison of flight controllers

	Pixracer	BeagleBone Black	BeagleBone Blue
Wi-fi	Yes	Add-on	Yes
Bluetooth	No	Add-on	Yes
Servo Outputs	6	Only with Cape	8
Processor Speed	180 MHz	1 GHz	1 GHz
Price	\$100	\$55	\$80

4.13 Autopilot Software

To fit the scope of this project, our UAV must be able to navigate a course autonomously. To achieve this our flight controller must communicate with an Nvidia Jetson (for computation purposes). During flight the Jetson processes data received from the cameras (e-CAM130_TRICUTX2) connected to it and sends that processed data to the BeagleBone Blue. For the BeagleBone Blue to use this data

meaningfully, it must be loaded with autopilot software. According to the developers, the recommended autopilot software is ArduPilot.

4.13.1.1 ArduPilot

ArduPilot is a collection of open-source autopilot software programs that are used to program unmanned vehicles. The software ArduPilot uses to navigate unmanned vehicles consists of: Radulae, ArduCopter, ArduSub, ArduRover, and AntennaTracker. Aside from navigation software, grouped with ArduPilot are a few ground control software programs, such as: QGroundControl, APM Planner, and Mission Planner. From the ArduPilot suite, we've chosen to use ArduCopter, Mission Planner, and AntennaTracker. Given that our UAV is designed to be an octocopter, the only navigation software included in ArduPilot that fits our requirements is ArduCopter. AntennaTracker serves to point a directional antenna at the drone autonomously, by using a flight controller on the ground attached to an antenna gimbal and GPS data from the gimbal and the drone to compute the right direction to point the antennas for best signal. Having this feature allows our RF links to be much better, since directional antennas will improve receiver sensitivity, which increases the link budget.

4.13.1.2 ArduCopter

ArduCopter is the open-source UAS controller for multicopters and helicopters within ArduPilot. For an octocopter drone, ArduCopter is the ideal choice for multi-rotor flight control applications. This is largely due to the adaptability and compatibility of the software. ArduCopter is flexible and can be adapted to suit many different multirotor configurations. It is also compatible with most of the major flight controllers, making it a favorable choice for this project.

4.13.1.3 ArduRover

ArduRover is the open-source autopilot software for ground vehicles and boats. To complete the mission, our UAV must drop an unmanned ground vehicle onto a landing zone, where this UGV must then travel straight to the destination. As we are already working with a Beaglebone Blue for our flight controller, using it as the controller for our UGV streamlines the setup: we already know how to work with the board, so there's less troubleshooting. To add to this, ArduRover is a part of the ArduPilot software suite that we are already using and goes through the same steps as the ArduCopter setup. By using ArduRover, it means that we can calibrate the ground vehicle using the same Mission Planner program that we are also using for our UAV.

4.13.1.4 Obstacle Avoidance

To complete the mission, the UAS must be able to avoid virtual obstacles uploaded from the competition server. The bulk of the work for this will be done by the Robotics Club, which will produce code that sends GPS waypoints to the flight controller over MAVLink. As such, a MAVLink enabled flight controller is required,

and it must be configured properly to follow the given waypoints in a predictable, accurate fashion.

4.13.1.5 Flight Modes

ArduCopter comes equipped with multiple flight modes that allow PID configuration for the desired style of flight. The recommended flight modes in ArduCopter are:

- Stabilize Mode - PID auto-levels the pitch and roll axis
- Altitude Hold Mode - Yaw, roll, and pitch can be controlled. Drone maintains constant altitude
- Loiter Mode - PID keeps the vehicle at the current location, heading, and altitude
- Return-to-Launch - Copter maneuvers from current position to the home position
- Auto - follows a predetermined mission

4.14 Ground Control Software

The chosen ground control software for our UAV is Mission Planner. Mission Planner is a part of the ArduPilot software suite and is recommended for use with ArduCopter. Included in the Mission Planner software are several features that we use to calibrate our UAV, simulate flight, create a flight plan, and record flight data.

Figure 39. Screenshot of basic screen view of Mission Planner (Image under creative commons, courtesy of ArduPilot Dev Team)



4.14.1.1 Flight Controller Configuration and Calibration

After ArduCopter is loaded onto the BeagleBone Blue, the flight controller must be configured and calibrated before it is ready to fly. With a fully hooked up drone, (frame, motors, speed controller, props, etc.) the flight controller is connected to a computer running Mission Planner. Connecting the drone to Mission Planner takes you through firmware installation before configuring the hardware. Through the Mission Planner interface the frame type is configured, and the accelerometer, radio control, and compass are all calibrated. Aside from calibration, this interface is used to adjust the flight modes and fail safes for the UAV.

4.14.1.2 Simulated Flight

One feature of Mission Planner is the flight simulation. However, the simulation feature in Mission Planner is currently only supported on airplanes. To simulate our copter, we take advantage of another program in the ArduPilot software suite, SITL (Software in The Loop) Simulator. SITL allows for flight simulation without the hardware required. This simulation, along with the flight plan will primarily be done by the Robotics Club as the scope of our tasks focus more on the electronics.

4.14.1.3 Flight Plan

The flight plan feature in Mission Planner grants the ability to setup waypoints and events for an unmanned mission. To path the course, our UAV will get waypoints from the MAVLink connection to the Nvidia Jetson TX2. Using flight planner on a drone running ArduCopter sets the home location at the position the drone was armed, so executing a Return-to-Launch waypoint will return the drone to the ground station; assuming it is armed there. The flight plan commands allow the drone to take on different actions at each waypoint. Flight plan commands include: Takeoff, Loiter, Jump, Land, and Return-to-Launch. As mentioned above, the bulk of the flight planning will be done by the Robotics Club.

4.14.1.4 Flight Data

During the flight, the PID stores flight logs on the flight controller's onboard memory. In ArduCopter these data flash logs start when the drone is armed and can be downloaded via the MAVLink post-flight. These logs can contain all information relevant to the UAV's flight, but the user decides what gets stored. Some data included in these logs are:

- Yaw, roll, and pitch
- Compass, camera, and GPS information
- Battery current and voltage, board voltage
- Event and error messages
- Accelerometer and gyro information
- Flight mode and performance monitoring

The values stored in these data logs can be viewed graphically through Mission Planner's user interface when downloaded onto a computer running the program.

To complete the entire scope of the mission our drone must have cameras that can detect objects of interest over the course. These objects include: a person engaged in an activity of interest, and a colored alphanumeric character painted onto a colored shape. To accurately detect these objects of interest during flight, we must feed high quality (4K ideally) video data from the cameras to our onboard computer - the Nvidia Jetson TX2. In an ideal case, our cameras would connect directly onto the Jetson itself, sending video through the 3.0 USB port on the board. There are several 4K resolution USB cameras on the market, so we considered a few of the following.

4.15 Camera Considerations

In determining what cameras to use, a few things must be considered. The camera's resolution, the focal length of the lens, and the sensor width. To maximize efficiency in our UAV's flight path, our camera should cover as wide an area as possible with as much detail as possible. The object detection is a critical part of the overall mission, meaning the camera we choose heavily impacts our results.

4.15.1 Pixel Ground Size

For our object of interest detection purposes, the most important value is the pixel ground size of the camera. The pixel ground size of a camera details the smallest resolvable value of a pixel on the ground. An object smaller than the pixel ground value will not be detected as a pixel by the camera, so the smaller this value is, the better. The pixel ground size is also important to returning the correct geolocation of these object of interest. Precise object detection creates accurate waypoint readings from the Jetson.

4.15.2 Frame Ground Area

The next most important value for our camera is the frame ground area. The frame ground area is the total viewable area in a frame at a specified altitude. This value is crucial to the search portion of the mission (along with pixel ground size). The larger the area that the drone can see, the sooner it can detect the objects of interest, the faster it can traverse through the course. A large frame ground area will result in an efficient flight path. The drone spends less time in search mode meaning less battery is needed.

4.15.3 Pixel Ground Size and Frame Ground Area Calculations

Before calculating the pixel ground size and the frame ground area, a few values from the camera are needed:

- The camera's resolution split into pixels wide and pixels high.
- Image sensor width
- Lens focal length

Another valued required to calculate the frame ground are is the altitude.

To calculate the pixel ground size, we must first calculate the frame ground width. The frame ground width is a result of the image sensor width divided by the lens focal length multiplied by the altitude, as shown in **Equation 6**. Frame Ground Width in terms of altitude.

Equation 6. *Frame Ground Width in terms of altitude*

$$FGW = (ISW)/(FL * ALT)$$

With the frame ground width calculated, the pixel ground size is calculated by dividing the frame ground width by how many pixels wide the camera is, and then multiplying that value by 100, as shown in **Equation 7**. Pixel Ground Size.

Equation 7. *Pixel Ground Size*

$$PGS = FGW/PW * 100$$

To calculate the frame ground area, we need to multiply the frame ground width and the frame ground height. The frame ground height is found by dividing the frame ground width by the camera's pixels wide multiplied by the number of pixels high, as shown in **Equation 8**. Frame Ground Height.

Equation 8. *Frame Ground Height*

$$FGH = FGW/(PW * PH)$$

Now with the frame ground height, the frame ground area can be calculated, as shown in **Equation 9**. Frame Ground Area.

Equation 9. *Frame Ground Area*

$$FGA = FGW * FGH$$

Table 24 compares a variety of different cameras and their characteristics in conjunction with the same equations and calculations.

Table 24. Some examples of pixel calculation values for popular cameras

Camera	PW	PH	SW (mm)	FL (mm)	ALT (m)	PGS (cm)	FGW (m)	FGH (m)	FGA (m ²)
HawKeye Firefly 8s	4000	3000	6.2	2	100	7.8	310	233	72075
Sony A6000	6000	4000	24	16	100	2.5	150	100	15000
Samsung NX500	6480	4320	24	20	100	1.9	120	80	9600
Sony A5100	6000	4000	24	16	100	2.5	150	100	15000
See3cam 135	4208	3120	4.6	4	100	2.7	115	85	9805
IDS UI-3590L	4912	3684	6.14	8	100	1.6	77	58	4417
Point Grey Blackfly	1288	964	4.8	2.8	100	13.3	171	128	21995
AUVIDEA 41MP	7728	5368	8.08	6.2	100	1.7	130	91	11797
e-CAM210_MI230	5256	3936	6.06	4.62	100	2.5	131	98	12884
Gopro hero 3	1920	1080	6.25	2.98	100	10.9	210	118	24727
LI-IMX274	3864	2196	6.26	5	100	3.2	125	71	8908

4.16 USB Camera Options

4.16.1 IDS UI-3590L

An impressive USB camera is the IDS UI-3590L from IDS Imaging. The IDS UI-3590L is an 18-megapixel camera that streams video through a USB 3.0 interface. At a 100-meter altitude, the IDS UI-3590L has a ground pixel size of 1.6 centimeters and a frame ground area of 4,417.9 square meters. While the pixel ground size is exceptional for this camera, the frame ground area is underwhelming. For our use case, the frame ground area is too low compared to other cameras we have looked at, and the gain from the pixel ground size is not enough to make it a worthwhile choice. Also, the compression of this camera is undisclosed and to discover the price of the camera requires asking for a quote. This would infer that the camera is rather expensive and be out of our budget for this project.

4.16.2 e-CAM210_MI230

The e-CAM210 is 21-megapixel MIPI camera based on the Sony IMX230. This camera uses USB 3.0 for connectivity, and streams raw, uncompressed UYVY video through this connection. At a price of \$59 per individual unit, the e-CAM210 is a cost-efficient camera option. With a low ground pixel size of 2.5 centimeters and a fair frame ground area at 12,884.3 square meters, a single e-CAM210 works well for UAV use. The issue using this camera is that it only streams uncompressed video. Without video compression, the amount of data required to send 4K video from three e-CAM210s easily exceeds the bandwidth capabilities of USB 3.0.

4.16.3 See3CAM 135

The see3CAM 135 from e-con Systems was our camera of choice in the early stages of our project. It is a 13 megapixel, 4K USB camera in a moderate price range (\$140). The see3CAM 135 has two output modes for video data: uncompressed UYVY, and compressed MJPEG. Though it is meant to be used with USB 3.0, the See3CAM 135 is backwards compatible and works with USB 2.0. The one knock to using USB 2.0 with this camera, however, is the loss of frame rate using the uncompressed UYVY mode. At first glance the See3CAM 135 appears a viable camera for our drone as it gives us the ability of streaming compressed 4K footage through USB 3.0 and USB 2.0. At an altitude of 100 meters, the See3CAM 135 has a pixel ground size of 2.7 centimeters and a frame ground area of 9,805 square meters. With a relatively low pixel ground size and a moderate frame ground area, the See3CAM 135 is the most cost-efficient camera that streams compressed 4K video and is known to work with the Jetson TX2.

4.17 Issues with USB Cameras

In our search for the ideal USB camera, our group came across a few issues. The primary issue regarding USB cameras is the data rate. To maximize the quality of footage that the Jetson receives from the cameras, we are required to use the single USB 3.0 port on the board. USB 3.0 can transfer data at speeds of 5 gigabits per second, or 625 megabytes per second. That transfer rate is no problem if the port only uses a single camera, but achieving our objectives requires multiple cameras to share the same port. It is possible to introduce a USB hub and connect all 3 cameras to the hub so they can share the one USB port on the Jetson, but we run into the issue of the high data rate needed by each camera. Three cameras streaming at 4K resolution with uncompressed or even compressed (typically MJPEG) video data would easily put us over the possible limit with USB 3.0. To add to this issue, the Jetson TX2 by default tries to conserve power by turning off “inactive” USB ports. Assuming we were able to get all three 4K cameras under the bandwidth limit, the Jetson would undoubtedly turn at least one of the cameras off during the mission.

4.18 Tri-Cam Solution

After some research, and communication with both the manufacturer and an expert at a company using both Jetson and e-con products, we found a solution to our camera issue. The e-CAM130_TRICUTX2 from e-con Systems. The TRICUTX2 is camera board that was designed to interface specifically with the Jetson TX1/TX2 development board. The TRICUTX2 comes with two modes for streaming: Asynchronous mode, and Synchronous mode. Asynchronous mode is standard streaming with three cameras; each camera can be controlled individually, and stream with higher frames per second compared to Synchronous mode. In Synchronous mode the three cameras work as one unit, so their output is synchronized. In Asynchronous mode the TRICUTX2 can stream 1080p and 72 frames per second, 4K at 30 frames per second, and 13MP at 19 frames per second. In Synchronous mode some of those numbers drop: 1080p streams at 30 frames per second, 4K maintains 30 frames per second, but 13MP streaming is not supported in this mode. The base board for the TRICUTX2 is made to interface with the MIPI CSI (Camera Serial Interface) that is on the Jetson TX2.

4.19 Camera Gimbal

To maintain the orientation of the three cameras during the flight, we are adding a gimbal in conjunction with the TRICUTX2 for stabilized footage throughout the mission. Camera gimbals use a set of motors and a controller to point the camera in a desired direction. In drones, gimbals generally use either R/C servo motors or specially made brushless motors. For this project, we will be using a brushless gimbal since they are lighter weight and have better pointing performance. A 2-axis gimbal will be used to keep the weight and complexity down, since the yaw axis is not required for this system.

The gimbal controller is another critical part of the gimbal system. Gimbal controllers consist of a main board that attaches to the frame of the drone, and a remote IMU that attaches to the camera. This allows the controller to know the orientation of the camera without motor encoders and keeps most of the controller weight off the end of the gimbal. The controller will have a suitable driver for the motor type, typically a pair of brushless motor drivers, specialized for holding the motors in a constant orientation rather than driving them at a continuous high speed. The controller must be easy to interface with the drone's control systems while still being able to drive the gimbal. For our purposes, a good gimbal controller should not require multiple servo signals, since all 8 servo outputs from the BeagleBone are already in use. SBus is an option but would require extra hardware and substantial custom software on the Jetson to work. An ideal setup would be controlled over I²C or UART, since both the Jetson and the BeagleBone have free ports for this. The ArduPilot documentation recommends either the SToRM32 or the SimpleBGC gimbal controller. Both controllers utilize fast 32-bit microcontrollers and provide a UART interface that is compatible with ArduPilot.

The cameras attached to the gimbal are light weight since they do not include batteries or a user interface, so the gimbal itself should be as light weight as possible to keep the overall drone weight down. The lightest gimbal with controller is the “GoolRC 2D”, which weighs only 163 grams with a BGC controller. The BGC controller has better documentation for its serial protocol and multiple examples and sample libraries, so it should be easy to use if custom gimbal control software is required.

Since the gimbal controller uses UART and has a protocol supported by ArduPilot, it can be connected directly to the BeagleBone running ArduPilot. If ArduPilot does not provide enough control over the gimbal, the gimbal controller can also be connected directly to the Jetson to expose all its functionality.

5 Hardware and Software Design

After sufficient research was completed on the necessary systems and potential components, the systems could be designed and integrated. This section details the design decisions as well as software and hardware layouts to be used in this project.

5.1 Hardware Detail Design

5.1.1 Battery Packs

The selected battery configuration is a pack with 20700 Lithium-ion cells configured in 6 series banks of 5 parallel cells. 4 of these batteries will be on the drone, with each one powering 2 motors. This allows the motor current to be evenly split among the packs, making wiring more straight forward since each pack handles much less current.

The 4 6S5P batteries were designed in cooperation with the MAE team who verified that they can deliver enough energy to keep the SUAS flying long enough to complete the mission. Each battery pack will supply power to 2 of the 8 motors, with the 2 motors selected to be opposite each other. Selecting opposing motors should distribute the load evenly, since when one rotor is throttled down, the opposing rotor is throttled up to produce control torque on the drone, keeping the load on each battery somewhat constant. Different batteries may have different amounts of load if the drone is producing a yaw correction, but the drone should not require a net amount of yaw in either direction since it is radially symmetrical.

The batteries will be connected by welding 0.010” thick, 0.5” wide copper tabs onto the battery terminals using a battery tab welder in the Innovation Lab. Welding tabs to batteries is a common means of wiring cylindrical Lithium-Ion cells, so using it on the SUAS is a good means of keeping the risk associated with the batteries down. The physical battery layout and the positioning of the battery monitoring

board will require collaboration with the MAE team to ensure everything fits together.

5.1.2 Battery Monitoring

Several devices, particularly the LTC2944, BQ34110, and BQ34Z100 were considered for battery monitoring solutions. **Table 25** compares the costs associated with each.

Table 25. Battery Gas Gauge IC pricing

Part	Cost
LTC2944	\$7.47
BQ34110	\$3.96
BQ34Z100	\$4.08

The BQ34110, while meeting the requirements is the weakest of the three, since it is as complex to configure as the BQ34Z100 despite having fewer features usable for this project, along with a lesser SOC estimation algorithm. The LTC2944 has substantial value in its simplicity, requiring minimal configuration and has pre-existing code that works with the Robotics Club's software, but is substantially more expensive than the BQ34Z100 despite having no algorithm included for fusing the different measurements into a robust SOC measurement. The BQ34Z100 is then the winner among the 3 devices, having a low cost, built-in SOC estimation that incorporates more than just Coulomb counting, and additional extra features, all of which make up for the increased complexity of configuration and lack of existing software support.

All 3 devices have a common issue that is important to consider in the final design: the parts have fixed I2C addresses which cannot be changed by any means. The sUAS will have 4 independent battery packs, which must be monitored separately. However, due to the fixed address, the translators cannot simply be connected all at once to the I²C bus. The final design will require the use of an address translation chip to allow all the devices to communicate with the host computer. However, the address translator may be moved to a different location. Having the translation done on pack would make debugging more difficult since a pack not on the drone would have a variable address, and when on the drone the address of the pack would not give any information about which part of the drone it has been plugged into. If the translation is done on the drone, then the address of the pack will correspond to the connection it has been plugged into, which can be helpful to determine what course of action to take if a battery pack is failing. Identifying packs by serial number is possible to keep track of the lifecycle statistics reported by the BQ34Z100, which enables us to ensure that every battery pack on the drone is in good condition before taking off. This can indicate to the user that a pack is near the end of its life and has only limited capacity available, so it can be removed

before flight, avoiding an emergency landing. Another consideration is that due to the inclusion of sense resistors in the ground and the high currents involved, an isolator should be used to eliminate issues with ground loops, different ground potentials, and reduce the risk of ESD damage when the batteries are installed or removed.

The BQ34Z100 datasheet shows an example 5S battery configuration, which includes a shift register for driving LEDs to indicate the battery state. For this project, some differences to the example arise. The example shows the voltage divider being controlled by a pair of MOSFETs to save power, but this will be eliminated to simplify the circuit and improve reliability and manufacturability given the boards will be hand assembled. The circuit also shows the use of a MOSFET and a Zener diode as a low-cost pre-regulator when using batteries above 4.5V, but in our design this will be replaced with a COTS linear regulator to reduce component count and ease assembly, since the cost is not so much higher as to justify the additional assembly work and risk of design errors. The sense resistor value will be lowered to ~600 uOhms due to the higher currents required. The voltage divider will also be adjusted according to the given formula for the 6S voltage. The HDQ interface will be omitted entirely, as it is not used in this system and would be redundant with I²C. A more robust protection solution incorporating a proper isolator will replace the Zener diodes and resistors on the I²C lines. The NTC thermistor specified for the battery temperature sensing will be made external to the PCB, so that it can be embedded in the hottest part of the battery pack to ensure no parts of the battery pack overheat.

To achieve the ~600 uOhm shunt resistance, 2mOhm and 1mOhm resistor were put in parallel. The resistors were selected by going to DigiKey, selecting the current sense resistor category, filtering by surface mount resistors, then filtering for 1% resistors with a low temperature coefficient. 50ppm resistors were selected as having reasonable cost for a low coefficient, since the high current means the resistors are very likely to get hot, so the low coefficient keeps error at high steady state current to a minimum. The resistor pad pattern was modified to include the kelvin sense pads in the middle of the current pads. All other resistors and capacitors were entered as generic 1206 size parts since these are much less critical and are much more widely available. The resistors for pull-ups and the current sense filtering will be 5% since these do not influence the measured values much, while the voltage divider for voltage sensing will be 0.1% with a low temperature coefficient since they have a large impact on the final reading (as recommended in the BQ34Z100 TRM (Texas Instruments, 2018)). Capacitors will be rated for twice the design voltage to ensure they do not cause faults. For all integrated circuits the manufacturer's recommendations on decoupling capacitors will be followed, especially to place the capacitors very close to the parts.

The COTS linear regulator selected to replace the discrete regulator is the TPS7B6933, which is a low quiescent current linear regulator from TI, which was selected for its 100ma rating and SOT223-4 package, which is easy to solder and provides reasonable thermal performance in an SMD form factor. The circuit is

expected to use less than 100ma, since the BQ34Z100 uses <1ma at full power, the isolation chip 3mA (on the battery monitor side), the shift register <10uA, and the LEDs 10ma each for 5 LEDs. The LEDs are only on briefly, so the average power consumption is <4mA, while the peak power consumption with all LEDs on is ~54mA. The regulator must drop at most 22V, so at 4mA 0.088 Watts are dissipated. With the LEDs on the regulator must dissipate 1.2 Watts. The regulator's thermal tab will be coupled to the ground plane through vias in the pad, which will allow the heat to be absorbed by the large copper plane on the back of the board. If heating is a problem, the LED display can be configured to be on for less time following a button press in the BQ34Z100's flash memory.

The PCB includes a set of holes suitable for an XT90 connector, which was agreed upon between the mechanical and electrical teams as a connector suitable for both teams' requirements. On the electrical side, the connector is rated for 90 amps, and the split-pin construction provides robust electrical contact. The pins of the connector can be soldered directly into the holes for a low-impedance, mechanically robust connection to the PCB. On the mechanical side, the XT90 has well shielded pins and doesn't have a built-in snap lock, so it is suitable for integration into the battery mount system. To reduce stress on the PCB the battery connector will be epoxied onto the battery case. For connecting the PCB to the batteries, the PCB has a pair of slots that are sized for the battery tabs to go through. The battery tabs will then be soldered directly to the PCB for a solid connection. A bill of materials for the battery monitor solution can be found in **Table 26**.

Table 26. Battery Monitor BOM (Excluding minor components)

Item	Cost/Unit	Qty/board	Total Cost/board
BQ34Z100	\$3.77 (qty 10)	1	\$3.77
SN74HC164D	\$0.338 (qty 10)	1	\$0.34
ADUM1250	\$5.39 (qty 10)	1	\$5.39
0.1% 15ppm Resistor	\$1.02 (qty 1)	2	\$2.04
1% sense resistor	\$1.01 (qty 1)	2	\$2.02
Total			\$11.36

5.1.3 Voltage Divider Calculation

According to the TI BQ34Z100 Technical Reference Manual, the input voltage to the battery voltage sense pin should be no more than 900mV at the maximum battery voltage. To achieve this an external voltage divider is necessary. TI recommends that the bottom leg of the divider be between 15 KOhms and 25

KOhms. The top leg of the divider is given by solving the basic voltage divider equation for the value of the upper resistor, as shown in **Equation 10**. Voltage Divider Calculation.

Equation 10. Voltage Divider Calculation

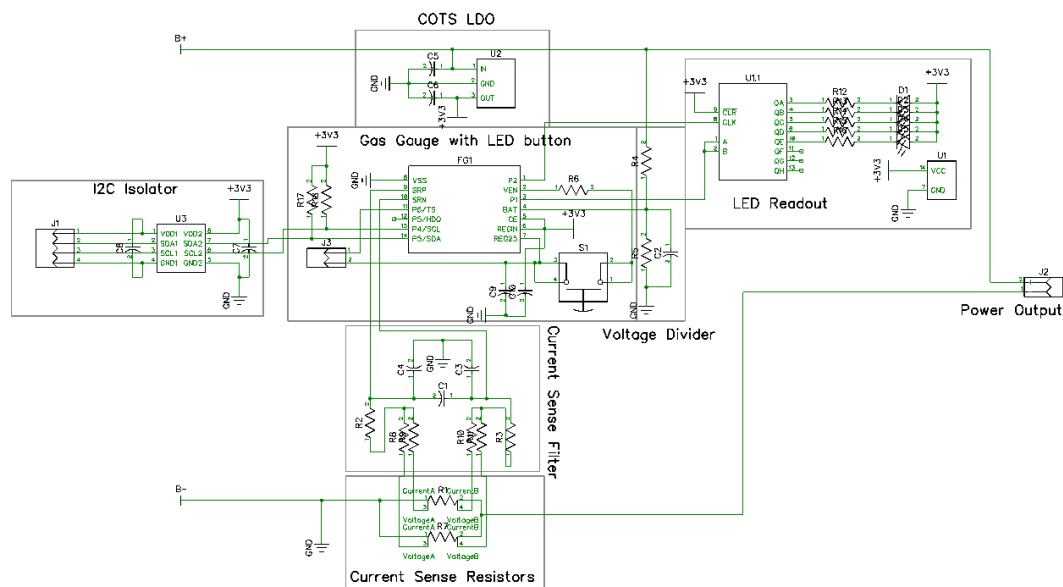
$$R_{upper} = R_{lower}(V_{in} - 900mV)/900mV$$

For a 16.5KOhm lower resistor as given in the TRM example and using the battery full voltage (25.2V) as V_{in} , the upper resistor must be 445.5KOhms. This is not an easily obtainable value, so the value of the lower resistor is varied to find a configuration that uses values close to standard resistors for both sides. The selected configuration is then an 18KOhm lower and 487KOhm upper resistor. The upper resistor is slightly larger than necessary to allow a small margin on the maximum voltage. The chip can tolerate higher voltages than 900mV on the battery voltage pin but cannot measure them.

5.1.4 Battery Gas Gauge Schematic and Layout

The battery gas gauge schematic and PCB layout were designed using a recent version of the DipTrace software, with a full license provided by the robotics club.

Figure 40. Battery Gas Gauge Schematic



Once the schematic was complete, it was used to produce a PCB file. In DipTrace, a PCB file produced from a schematic has the components in the schematic automatically inserted and connects the pins with ratlines. The PCB components were then positioned to make connecting them with actual traces easier, while ensuring components are placed at appropriate distances to each other (like having decoupling capacitors close to the parts, keeping the current sense filter symmetrical, etc.). For most traces, the default 0.33mm trace width was kept, since

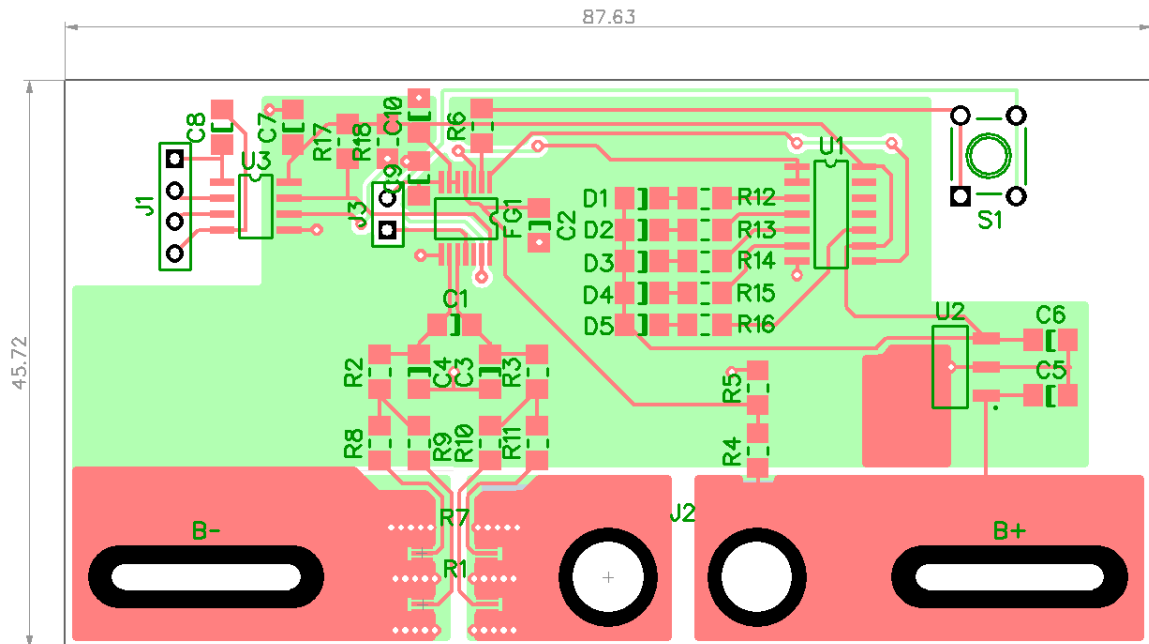
most traces handle very little current. The high current traces between the power connectors and sense resistors were kept as short as was reasonable, and the copper area was defined using copper pours to have the maximum possible width in all parts of the high current connections. To ensure the high current traces do not overheat and to allow for optional additional reinforcement, the solder mask exclusion feature was used to specify no solder mask over the bulk of the high current trace. This improves cooling by bringing the trace into direct contact with air, and if overheating is a problem the trace can have solder added to reinforce it without having to rebuild the boards. The high current traces also take advantage of both sides of the PCB by having the same trace on both sides, then using many vias near the current sensing resistors to allow current from the other side to flow through the resistors. The resistor pads use a custom pattern that allows for the kelvin sensing scheme described by analog devices, with the only difference being that current is sensed from the inside to accommodate the 2-layer PCB with power traces on both layers in that area.

The layout is shown in **Figure 41**. The light green is the bottom layer, the pink is the top layer, the darker green is silkscreen, and the black is a pad. Solder mask is not shown. U2 is the COTS regulator and connected to its heatsink tab is a copper pour to help remove heat through the top of the PCB. S1 is the button to activate the LED readout, and the button was chosen to have a post on top suitable for attaching a 3d printed extension that could go through whatever protective cover or case is attached to the battery.

Future iterations of the battery monitoring PCB may include different connectors or indicators as the mechanical requirements evolve. The battery management PCB design may also be adapted for use on the UGV by recalculating the voltage divider and current shunt for the smaller power on the UGV, and stripping down the isolation or indication since the UGV has a much lower power environment that does not require such a robust system, and is more tolerant of a battery pack being not completely full since it has a limited operation time. The UGV battery monitor could further be simplified by switching to the LTC2944 chip already in use at the robotics club, which can produce a very small battery monitor of less than an inch square with a fuse

The battery monitor PCB was designed with the requirements of the PCBWay PCB manufacturing service in mind, with the goal of being less than 100mm by 100mm and 2 layers to allow us to use the 5 PCBs for \$5 service, which only takes a week to have the PCBs made and delivered. The minimum feature size is 0.006". The PCB will be populated and tested in-house using Robotics Club facilities.

Figure 41. Battery Gas Gauge PCB layout



5.1.5 I2C Address Conversion

As discussed in section 4.11, there are two main techniques for handling I2C address conflicts and many devices to do both tasks. Of course, the TI TCA9544APWR was selected due to its low cost and versatility. The TCA9544APWR needs various pullup resistors and filtering capacitors to support it, so those were added. In addition, it was desired to be able to tell when an I2C port was in use and to provide protection against potential ESD.

Certain consideration are needed when monitors the I2C buses for activity. The first is that whatever connections are made to the bus must not contain significant parasitic capacitance. In fast mode I2C operates at 400 KHz and any significant capacitance can limit the switching frequency. As such, it is desired to keep the parasitic capacitance of the monitoring components under 10pf. The second thing to consider is the active condition of the bus. Both the data and clock bus are normally high. As such, a simple transistor driving an LED connected to the either bus wouldn't work because the LED would be on normally and get slightly dimmer when the bus was in use. To fix this issues, the LED is connected to the clock signal through an inverter. The TI SN74LVC2G14 provides logical inversion with a low input capacitance of 4pf. Since the cost of the SN74LVC2G14 is reasonable for the functionality this component was selected.

In order to protect against ESD TVS diodes must be added to clamp the voltage within a range such that the components won't be damaged. In this case, the I2C buses are operating at 3.3V so a package with diodes to clamp at 3.3V was chosen to provide this protection. The TI TPD4E001 provides this functionality as a low cost and with a very low input capacitance of 2.5pf. The cost of the major

components on this board are summarized below in table **Table 27**. Cost of major components on the I2C address conversion board.

Table 27. Cost of major components on the I2C address conversion board

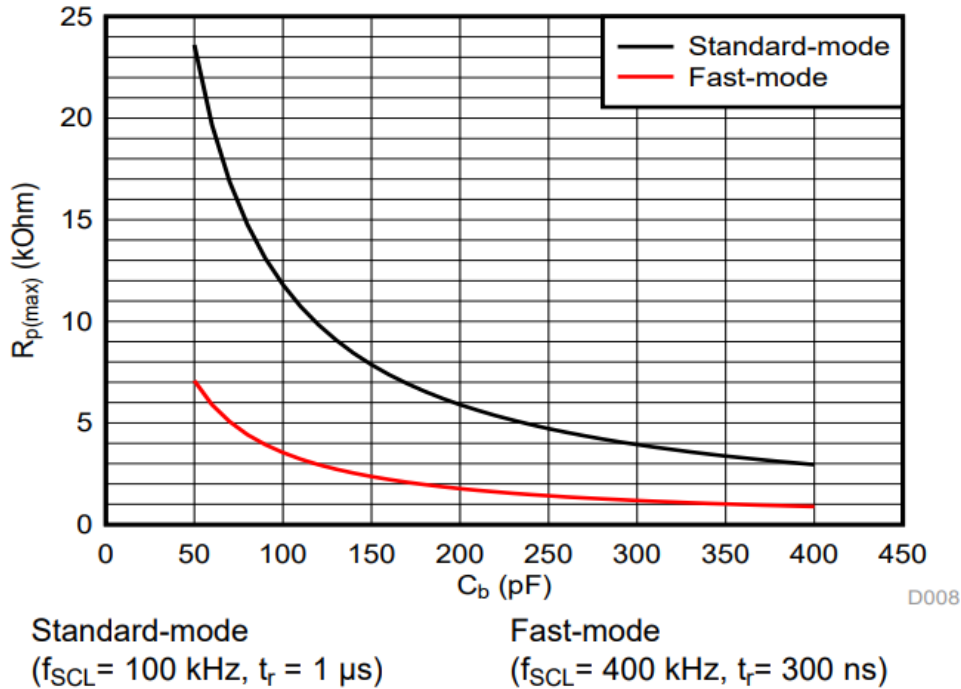
Item	Cost/Unit	Qty/board	Total Cost/board
TPD4E001_DBV_6	\$0.52	3	\$1.56
TCA9544APWR	\$1.10	1	\$1.10
SN74LVC2G14-Q1_DCK_6	\$0.50	2	\$1.00
MF-SMDF050	\$0.48	2	\$0.96
Total			\$4.62

5.1.6 Pull-up Resistor Calculation

The TCA9544APWR requires bus pull-ups resistors in order to communicate properly. Inevitable the bus as capacitance due to the PCB traces and device internals. Whenever a capacitance and resistance are connected in series a time constant results which limits the rate at which the bus can change states (high to low or low to high). An ideal digital signal has an infinite slope, in other words, goes from high to low or low to high instantaneously. Of course, this is never the case in the real world. The time constant limits the minimum time taken to change states. Depending on the frequency the bus is desired to operate at, this time constant must be controlled to ensure that the bus can change states quickly enough.

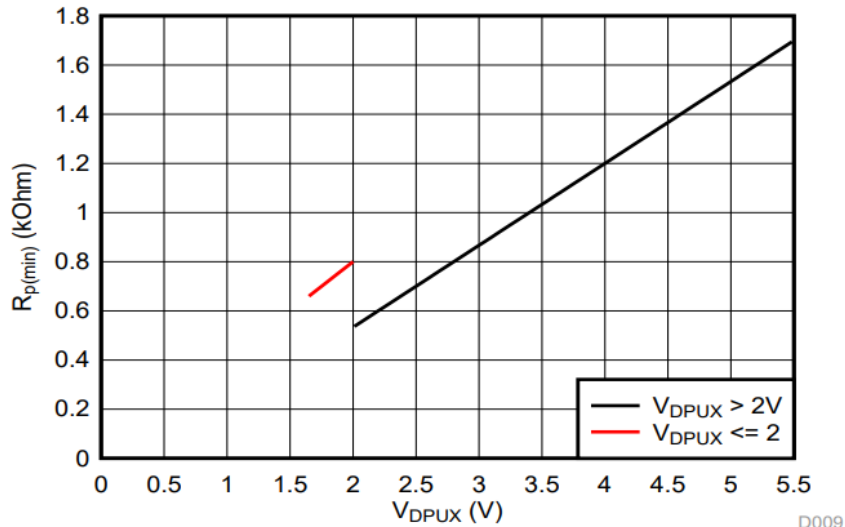
The datasheet for the TCA9544APWR contains tables that show the relationship between the pull-up resistance and the bus capacitance.

Figure 42. Maximum pull-Up resistance vs. bus capacitance. Reproduced with permission from TI **Appendix A**.



The bus is intended to be used in high frequency fast mode which corresponds to a 400KHz clock rate and a rise time of 300ns. Of course, the exact resistance value can only be calculated when the bus capacitance is known. Unfortunately, in order to know the bus capacitance the capacitance of all connected devices must be known as well as the capacitance of all of the PCB traces. These are difficult to calculate so a median value of 200pF is chosen. From **Figure 42**. Maximum pull-Up resistance vs. bus capacitance. Reproduced with permission from TI **Appendix A**. 200pF corresponds to a maximum pull-up resistance of about 2K Ω .

Figure 43. Minimum pull-up resistance vs V_{DPUX}



$$V_{OL} = 0.2 \cdot V_{DPUX}, I_{OL} = 2 \text{ mA when } V_{DPUX} \leq 2 \text{ V}$$

$$V_{OL} = 0.4 \text{ V}, I_{OL} = 3 \text{ mA when } V_{DPUX} > 2 \text{ V}$$

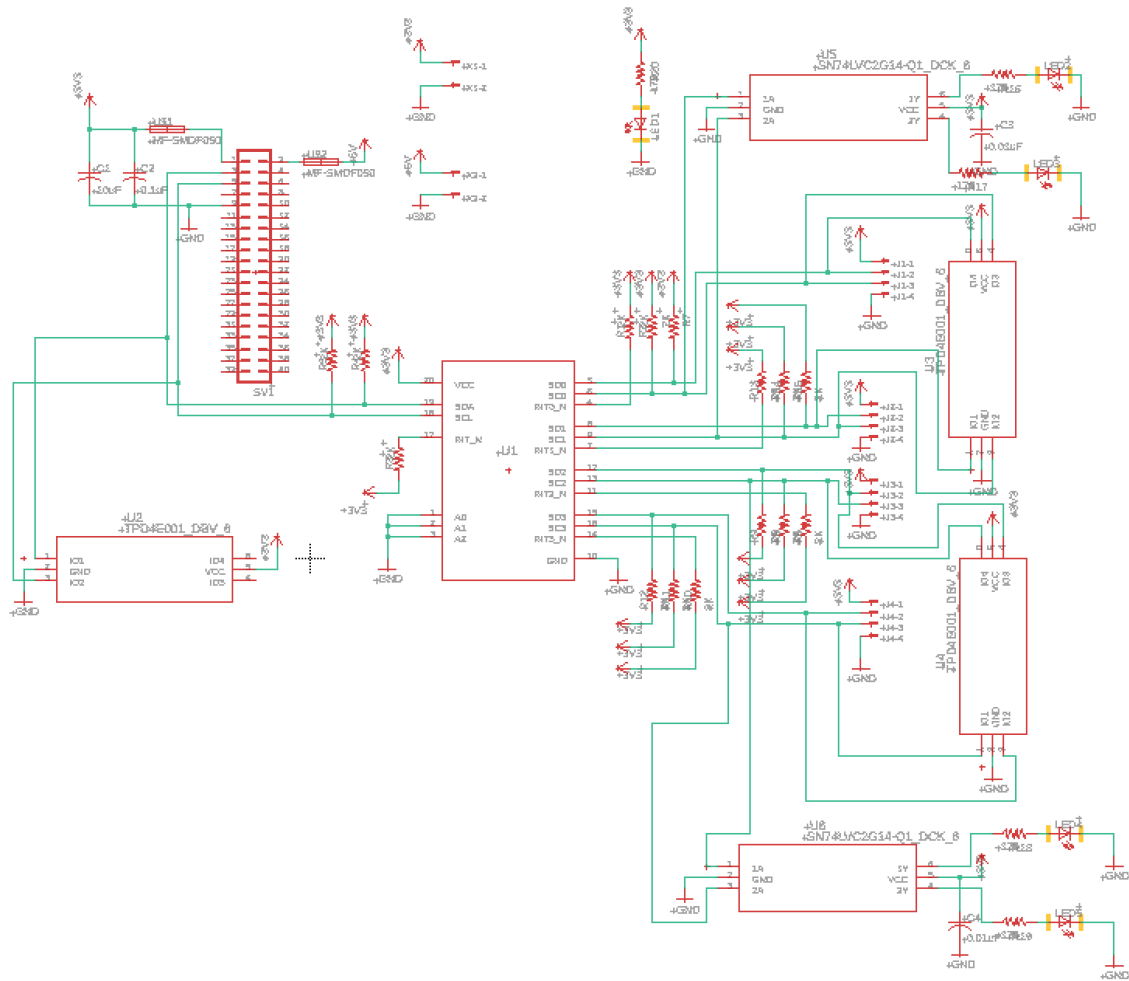
The V_{DPUX} is approximately 3V which corresponds to a minimum pull-up resistance of 0.9KΩ. Due to the potential variance in bus capacitance the pull-up resistance is chosen to be 2kΩ.

5.1.7 I2C Address Converter Schematic and Layout

The battery gas gauge schematic and PCB layout were designed using a recent version of the EagleCad software, with a full license provided courtesy of Autodesk. **Figure 44.** I2C Address Converter Schematic shows the completed circuit

The schematic creation was largely based off the information given in the TCA9544A datasheet and the information given in the NVidia Jetson TX2 adapter board datasheet. The board is meant to sit atop the J21 GPIO connector and provide address independent connections to Molex kk-4 connectors. These connectors go to the battery pack monitoring boards and allow the Jetson to communicate with four of them though not simultaneously. The board uses a 2x20 pinboard so as not to obscure the J21 GPIO pins for other use.

Figure 44. I2C Address Converter Schematic

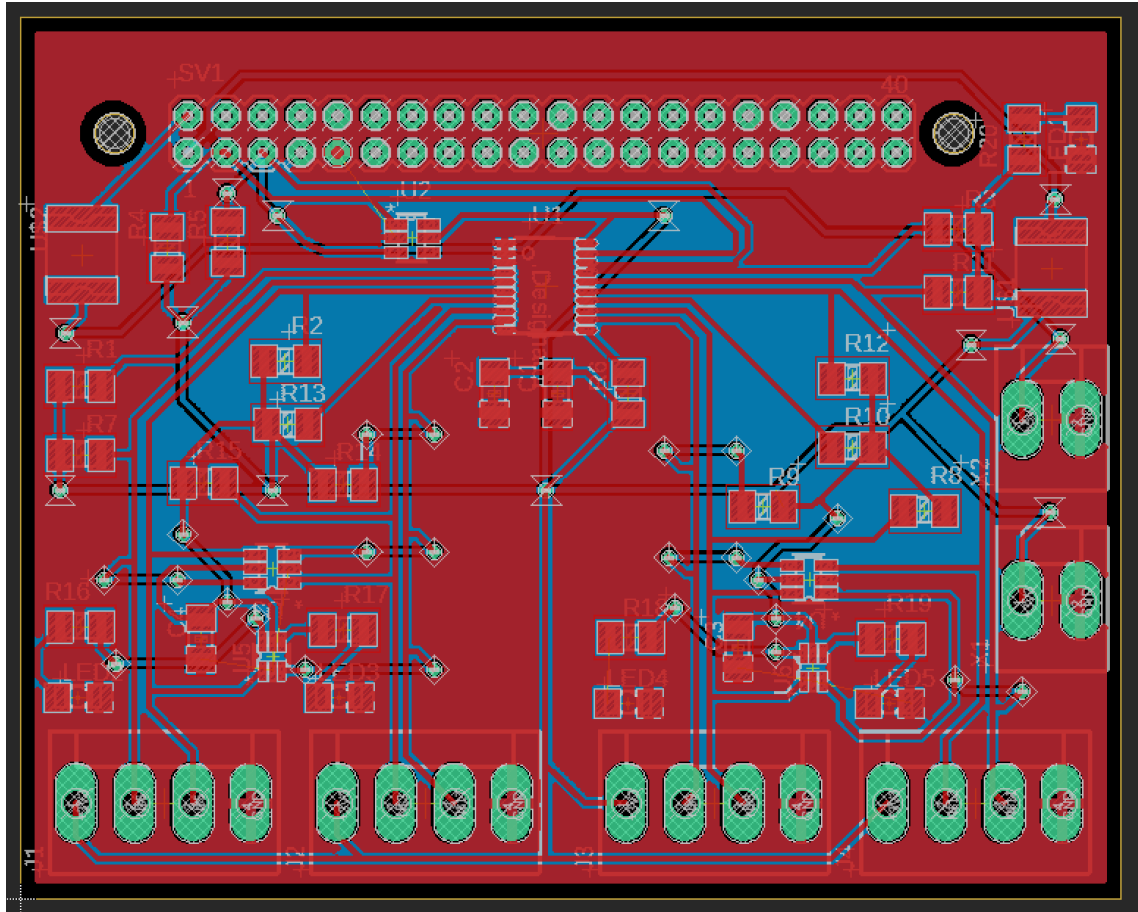


The TI TCA9544A was chosen due for cost, availability and ease of use, as noted in section 4.11.2. The board uses a resettable poly-fuse on the input with a 500ma trip current to help protect the circuit from shorts and unwanted connections. The larger capacitor at the input is used for voltage smoothing and the smaller capacitor is used for decoupling. The pull-up resistors are used to keep the I2C bus high. Their values are important and detailed in section 5.1.6. Pull-up resistors are also required on the input to the TCA9544A because the Jetson does not have internal pull-up resistors. An LED was added to determine when the device is on and additional diodes were added for ESD protection. Two connectors were added for connection to the regulated 3.3V and 5V on the Jetson in case another device required that voltage. The resistors, capacitors and diodes are all 1206 components which are small enough to be space efficient, but large enough to be hand soldered.

The board layout was relatively straight forward. The data traces were kept to the smallest printable size by PCBWay which is 6 mil. These were kept small in order to help keep potential bus capacitance to a minimum. Holes were added for

mounting and screen printing was added for ease of assembly. The board will be fabricated and hand soldered at the UCF Robotics lab.

Figure 45. Trace layout for the I2C address conversion board



5.2 Software design

This project requires software for flight control, resource monitoring and image recognition. Of course, with the amount of open source code available, this project involves more code manipulation and use than raw software development. This section details the software that will be developed as well as how current software will be adapted and used.

5.2.1 Battery Gas Gauge

The sUAS software for the central computer that the battery pack monitors will be connected to is based on the Robot Operating System (ROS). ROS is a framework that allows highly modular pieces of code to communicate with each other, allowing different code to be easily re-used, or plugged into a test system without modification to the actual code under test. The software that bridges between the battery monitoring modules and ROS will be developed by this team since ROS allows us to work only on the parts where we have the best understanding.

Configuring and communicating with the battery monitor requires knowledge of the electrical design, so we are best suited to write this part of the code.

Since the overall system uses ROS, the software for the battery pack monitoring will use ROS as well. Python is the language of choice for this task because it is easy to use, and performance requirements are low for this part of the system. The first stage of battery monitoring software development will be to create a Python class that handles communication with a battery monitor. Python libraries for I²C communication already exist, but substantial work on top of this will be needed to develop code which can determine the correct bytes to send over I²C to both configure and read the battery data. Ideally, the same class should have methods for both configuration and reading so that it can be used standalone for debugging and maintaining the battery packs.

The ROS side of the code will take the class developed for communicating with the battery pack and use it to gather data from all 4 packs on the drone, then send the data to ROS using the ROS-standard BatteryState message. The BatteryState message transmits over ROS important information about a battery, such as its chemistry, overall health, voltage, current, temperature, design and measured capacity, a percentage SOC, cell voltages, and the location and serial number of the battery pack. Most of these values can be read from the battery pack monitors themselves, except for the battery pack location and cell voltages. The cell voltages are not measured by the battery gas gauge IC, so ROS specifies to report them as NaN. The battery pack location will be determined by the address which the gas gauge IC address is translated to by the drone's built-in address translation system.

The measured values once packaged into a BatteryState message can be published to a ROS topic, where higher-level decision-making software can subscribe to them and use them to determine if a landing is needed due to low or malfunctioning batteries.

5.3 Jetson Cameras

One major advantage of the CSI camera connection is that it allows the Jetson TX2 module to use dedicated hardware for the cameras. The Jetson module has 6 2-lane CSI drivers, which can be configured to either write the pixels directly to memory or to write pixels to the Image Signal Processor (ISP). For cameras which require 4 lanes, the CSI drivers can be used in pairs of 2. If the pixels are routed to the ISP, the ISP hardware can perform several common, essential operations like taking statistics, converting from Bayer to RGB/YUV, and correcting color. Compared to a USB camera that does not have Bayer to RGB conversion, this saves a tremendous amount of CPU time. The ISP supports large sensors with very high throughput, so a lot of processing can be offloaded to it. Once an image is processed with the ISP, it can be written directly to anywhere in the Jetson's memory. With a USB solution, the CPU must copy from the USB system to the destination memory, further wasting CPU time.

To take full advantage of the Jetson ISP, new libraries must be used instead of the usual OpenCV image capture. NVIDIA has made an API called Argus, which is designed to enable low level control of camera capture on NVIDIA Jetson devices. Argus allows control of most or all the features of the camera and ISP with support for multiple cameras and the choice between automatic and manual modes for parameters like exposure time or gain. Another feature of Argus is that it allows the use of EGLStreams, which are a system developed by the Khronos group (developer of OpenGL, WebGL, and others) for transferring frames between different pieces of software. When used with Argus, EGLStreams allow the programmer to have the ISP configured to write image data directly to GPU memory. Writing the image data directly to the GPU saves a substantial amount of time (several milliseconds for each frame) that would be spent uploading data if a CPU based capture system was used. In a system where each frame only has 50ms to process 39MP, this much time could easily make or break the success.

To implement this, the ECE team will need to work with the CS team to implement the Argus side of the system to transfer data to the GPU, and from there the code can be passed off to the CS team to implement the CUDA processing of the image.

5.4 Flight Controller Setup

To use the BeagleBone Blue as our flight controller, it must be configured to run the multicopter UAV software, ArduCopter. Running ArduCopter on our BeagleBone Blue allows our flight controller to receive flight data and send the proper signals to the motors, GPS, and other components. Once ArduCopter is loaded onto the BeagleBone, the drone needs to be configured to take in these flight commands – whether manually through an RC transmitter, or autonomously through an onboard computer. For our purposes, this drone receives commands from a Nvidia Jetson TX2 running a ROS MAVLINK node. The communication with the flight controller and the Jetson is outside the scope of this project, but our group is tasked with setting up and connecting the flight controller.

BeagleBone Blues are computers designed for hobbyist robotics. Out of the box, we can install the BeagleBone Blue with the Linux distribution Debian. Having Debian installed simplifies the configuration process of the flight controller because the operating system gives us access to the functions and programs needed to adjust the board. When configuring a robotics computer, the configuration of the board is dependent on what robotic task the BeagleBone is being used for. For this project, our group is using the BeagleBone Blue as a flight controller for an octocopter. To fly, Octocopters rely on 8 independent motors, and the communication from those motors to the flight controller for thrust and balancing purposes. This communication takes place through drone's PID (Proportional Integral Derivative), which processes the data received from multiple sources and sends data based on several factors (flight modes, obstacle avoidance, telemetry information). These PID calculations are made by the UAV software that is running on the drone. For the BeagleBone Blue, the suggested robotics software suite is the open-source ArduPilot. In this group of software is the ArduCopter program,

made for multicopters. ArduCopter is designed to take advantage of flight controllers that are connected to multiple motors and props. It offers support for independent motor control, obstacle avoidance, planned flight, and more. Before running ArduCopter, the BeagleBone Blue needs to be configured to support the features provided by the software. Despite being designed and developed by Texas Instruments, they do not provide detailed guides for BeagleBone configuration. A few reasons for this can be inferred:

- Most robotics software is open-sourced and created by independent developers. There is the option of TI working with these developers, but from a business perspective it makes more sense for TI to build the board and leave the development to those who are more experienced in those areas.
- The setup and configuration of the board varies from person-to-person, project-to-project. How the board is setup and configured is all dependent on the specific needs of each project, and how that person chooses to approach it.
- Ultimately these boards are designed for hobbyist and developers. By not standardizing board setup it allows creates the freedom to design and configure things to their liking.

With the steps for this configuration process not being readily available from Texas Instruments, it took some research to learn how setup the board for ArduCopter. Although they are designed and created by Texas Instruments, BeagleBoards are community-based projects so software development for these boards come from external developers. Regarding ArduPilot programs (ArduCopter, ArduRover, etc.), there are a few guides on how to get those running properly on a BeagleBone Blue. Through our searching we found Imfatant's guide on GitHub for setting up and running ArduPilot on the BeagleBone Blue. The guide goes over the 3 stages it takes to get your drone setup to fly: Preparing the BeagleBone, installing ArduPilot on the board, and connecting components to the board. While it would be redundant to go over the entire guide, our group can speak on how we went about setting up the board.

5.4.1 Preparing the BeagleBone

Before being able to set up the BeagleBone Blue as a flight controller, it needs to be set up first. The BeagleBone Blue is like any computer, it needs an operating system, power, input/out, etc. To power the BeagleBone Blue for setup we use the microUSB port on the board and connect it to a computer. The I/O for setup also happens to interface through that same microUSB port, originally at least. To fully set up our computer we need to install an operating system, then we can configure it the way we want.

5.4.1.1 Installing the Operating System

The first step in configuring our BeagleBone Blue was to install an operating system so the board can run programs. There are multiple options for operating systems on the BeagleBone Blue, Bone-Debian being one of them. Bone-Debian is a modified version of the Debian Linux distribution with BeagleBoards in mind. Bone-Debian simplifies interfacing with the board by allowing us to connect to the board via USB, or Wi-Fi, or even SSH. Running Bone-Debian on our BeagleBone Blue also makes modifying processes and configuration files a simple task.

For storage, the BeagleBone Blue uses a microSD card so we were required to flash an SD card with an image of Bone-Debian to install the operating system on the board. On the GitHub repository, Imfatant provides a link to the bone-Debian image file, which we flashed onto the card using the program rufus. After flashing the SD card, the next time we inserted it into the BeagleBone Blue the board booted into Bone-Debian

5.4.1.2 Wi-fi Setup

The Beaglebone Blue comes with a built-in wifi module, but to use it, we require a means to interface with it. With Bone-Debian now installed on the Beaglebone Blue we gained access to network setting that can be used to connect our board to wi-fi. To login to the board we used ssh (Secure Shell) to connect to the Beaglebone through its local IP address. To run the commands necessary to modify key features of the board, we needed sudo access. Sudo access is super-user access for linux systems, similar to administrator access on windows operating systems. Logged in a sudo, we used the connman (Connection Manager) services that comes pre-installed on Debian systems. With the connman service, we are able to scan nearby wi-fi addresses and connect to the through their unique hash value. Imfatant modifies a configuration file to set the wi-fi for the board, but this method did not work for us as our Beaglebone Blue connects to a public network that requires a network agreement. Still using the connman service, we set up the wifi manually:

Before even trying to use wi-fi on the Beaglebone Blue, the wifi must be enabled. To enable the wifi on the board we ran started connman, then used the command: *enable wifi*. After enabling the wifi module on our board, the first step in connecting to the wifi was using connman to scan nearby access points. This scan stores a list of available wifi connections on the board. To view these scanned wifi points, the connman command *services* is used. Issuing the services command lists all of the scanned SSIDs for nearby wi-fi points, along with their hash values, in the terminal. To establish a connection with one of the listed wi-fi points, we used the *connect* command followed by the hash value of the network we chose. We initially tried to connect to the UCF_WPA2 network, but there was difficulty in verifying the identity. To work around this issue we decided to connect to the public network UCF_GUEST, but we ran into another issue: connecting to UCF_GUEST requires accepting the network agreement before acquiring internet access. To note:

connecting to the Beaglebone Blue to your computer through a USB connection allows one to use the Beaglebone Blue as a wireless network. With that in mind, we were able to troubleshoot our network issue. We first connected to UCF_GUEST through the `connman connect` command. After connecting to UCF_GUEST an odd message would display in the terminal asking the user if they are connected. This message is more than likely the result of not being able to accept the network agreement from UCF_GUEST. Despite this our Beaglebone could display that it was connected to the network, but it did not have internet access. Next, to get an internet connection, we connected to the Beaglebone Blue's wifi through the wireless network interface on our computer. Opening up a webpage after connecting to the Beaglebone Blue's wifi directed us to UCF's network agreement page, and through accepting it, we were able to get internet access to the Beaglebone Blue. One caveat to this solution is that we have to connect to the Beaglebone's wifi and accept the agreement every time we turn the Beaglebone Blue on in order to get a connection. We are currently searching for solutions to fix this issue, as we'll have trouble in the future if we stick with this method. It is also important to note that: with wifi set up on the Beaglebone Blue, we can ssh into the board wirelessly through its IP address. This will come in handy to allow the board to communicate with the other devices that we have on, or near, the drone (Jetson, Ground Station). At the moment, the Beaglebone Blue's "server" is DHCP (Dynamic Host Configuration Protocol), which means our IP address will constantly change. This will be a problem for our communications in the future, so we're currently working on a solution to set a static IP address.

5.4.1.3 Software Updates

With a wifi connection established, our Beaglebone Blue could not use the internet to download and install updates. Compared to the wifi setup, the update process is rather simple. As `sudo`, we used the `apt-get` (Advanced Package Tool) command with the `update` argument to get the update; retrieving the packages from multiple servers. Still using the `apt-get` command, we used the argument `upgrade`. This argument allowed us to upgrade the installed packages that had available updates. The final step was running `apt-get` with the `install` argument, which ultimately installed the newly upgraded packages. Through these steps, we were able to make sure that our Beaglebone Blue was running the most recent software before running ArduPilot software on it.

Having a stable internet connection also allowed us to modify a number of the Beaglebone's features. We needed to update the scripts, modify the kernel, set services and more. After completing all of the steps to install and start ArduCopter, we had to come back to these steps as we unintentionally missed one that required updating a dtb file on the BeagleBone.

5.4.2 Installing ArduPilot

After all the software updates, and a reboot of the Beaglebone Blue, we worked towards getting ArduCopter on the Beaglebone Blue. With our board prepped, our

next step was to create config files for Ardupilot on the Beaglebone Blue. These config files create settings for ArduPilot with the Beaglebone Blue that make sure ArduPilot communicates information through the correct ports and runs the correct services.

5.4.2.1 Telemetry and GPS

The first config file we made was to configure telemetry and GPS communication for ArduPilot with the board. For starters, we created a file in the path “/etc/default/ardupilot” and added three lines to set the telemetry and GPS communications to the specified ports. To specify these ports, ArduPilot uses 6 arguments as identifiers.

The arguments for this config file are:

- -A is a console switch / wifi link. Here in our config file we communicate with the board through a wifi connection sending udp packets.
- -B represents ArduPilot’s GPS port. By default, setting -B will map the serial port to the Beaglebone Blue’s UART2 port with a 57600 baud rate. This port is where we hook up our GPS module with an external compass.
- -C is the first telemetry serial port, which maps to the Beaglebone Blue’s UART1 port with a default baud rate of 57600. This port will allow our drone to communicate to the Ground Control Station and send it telemetry information.
- -D is defaulted as the second telemetry port with a baud rate of 38400. For our uses, we have currently allocated the second telemetry port for wifi – meaning -D is unused. However, we are working on modifying our config file to add support for a third telemetry option.
- -E and -F are unnamed ports defaulted at 38400 and 57600 baud rates respectively. These ports are typically unused, but some potentials uses are for an extra GPS port, or a radio port.

Currently, our config file sets the first telemetry port to UART1 on the Beaglebone Blue and the second telemetry port through the board’s wifi. This setup with wifi as a telemetry option allows our Beaglebone to communicate to Mission Planner wirelessly as long as the IP address of the computer running mission planner is in the config file. Our third configuration is setting the GPS communication of the Beaglebone Blue to the UART2 port on the board, which is conveniently labeled as GPS.

5.4.2.2 Services

After creating our config files, the next step was to add system service files for each ArduPilot software that we are running. The two services that we are running on our UAV are ArduCopter and AntennaTracker. The UGV will solely run ArduRover; it does not need antenna tracker as the UGV will be relatively close to

the Ground Station throughout the duration of the course. These service files are split into three sections: Unit, Service, and Install.

The Unit section defines the unit to ArduPilot and how it should interact with other units on the board. Directly under unit header is a description of the service. For our UAV, we set one files description to be “ArduCopter Service”, and the other file is “AntennaTracker Service”. On the UGV the description is “ArduRover Service”. Underneath the description is the line fore networking service. This line makes sure that networking services turn on with the ArduPilot services. Beneath the network services line is the start limit interval line. This line is set to 0, which allows our Beaglebone Blue to continue trying to start the service if it is not started. The final line of the Unit section lists the conflicts. The conflicts should be labeled with the other ArduPilot services, so they don’t “fight” with each other.

The Service section of this file lists the directories that contain certain files that ArduPilot needs to operate correctly. The environment file is kept in “/etc/default/ardupilot”. The next line in services is set to run *aphw* in the “usr/bin/ardupilot” directory *before* actually starting the service we’re running. The last line in the services section is for what to execute as the service starts. In our groups case, our UAV and UGV execute the telemetry ports along with the GPS port.

Below the Services section, but before the Install section are two lines for setting how the board restarts. The first line is set to restart the services when it fails to run, and the second line sets the delay of the restart to one second. The final line of these service files falls under the Install section. Simply put, this line labels that the Service will start after it reaches a specific runlevel. The purpose for this runlevel is to make sure that all dependencies are running for the service, so it starts properly.

5.4.2.3 Hardware Configuration

Before configuring the hardware, we needed to make the directory “usr/bin/ardupilot” (the directory we’re accessing in the system service file). In this directory, we wrote the *aphw* (ArduPilot Hardware) script that gets run by the system service file. This file allows whatever ArduPilot service is running to turn on specific hardware features, like powering on servo rails and enabling the PRU. After saving this file, we needed to use the *chmod* command to set the permissions for it. Once the permissions were set, we moved on to downloading the service executables on our Beaglebone Blue.

To retrieve these executables, Imfatant linked copies of them in their github repository. These files originally came from Mirko Denecke’s website, bbbmini.org, but were much easier to access from the github page. To store these files on the Beaglebone Blue we used the *wget* command, with the argument being the link to the file. We saved these executables in the same “usr/bin/ardupilot” directory that we created before. On the UAV we saved the ArduCopter executable and will save the ArduRover executable on the UGV. There is the option to compile the

executables, and this is what we attempted originally, but we ran into trouble during the configuration. After storing executables for the services, we were able to start them using the `systemctl` command, which allows us to enable, stop, start, reset, and reload the services. On the UAV we started the `arducopter.service` in order to run ArduCopter on the board at boot. We initially had trouble starting the service but were able to make it work after some troubleshooting.

5.4.2.4 Troubleshooting

The main difficulty we ran into was an inability to get ArduCopter started despite following the steps. When ArduCopter is running on the Beaglebone Blue a red LED is supposed to flash, showing the service is working. Our Beaglebone Blue did not blink flash red and using `systemctl` we found it was not starting. Through the `journalctl` command, we discovered that ArduCopter was not starting due to a file issue. The file of note was a Device Tree Blob file labeled “`am335x-boneblue.dtb`” in the directory “`/boot/dtbs/4.4.113-ti-rt-r149`”. These device tree files describe the Beaglebone Blue’s hardware, so the operating system can use its components. In the mentioned directory, there was already a file with the given name, but it was not set to describe the Beaglebone Blue. To fix our issue, we looked in Imfatant’s github repository, and found the file meant for the Beaglebone Blue. We renamed the original file in the directory, and then downloaded the new file using the `wget` command. After rebooting the board, we finally saw the blinking, red LED, and `systemctl` displayed that ArduCopter was running.

5.4.3 Component Connections

Now, having ArduCopter running on the Beaglebone Blue means that the board now acts as a flight controller. The next steps are to put our flight controller on a frame with the necessary connections (power, receiver, servo outs, gps) and run software to configure the drone. At the moment, the final frame of the drone is not done so our group is doing testing with a frame that was provided by the Robotics Club. When the frame is finished, we will be able to move the components that we need to the new frame and re-calibrate it.

5.4.3.1 Mission Planner Setup

With ArduCopter running on our board, we then worked towards setting up the Ground Control Station software. For our drone, we’re running Mission Planner to take set up and calibrate the drone. To communicate with the drone, Mission Planner gives the option of using a COM link, UDP connection, and TCP. For our UAV, we’ve set it up to communicate through UDP, but we ran into some issues early. Before calibrating the drone, it is required to download and update the firmware for the board. To download the correct firmware, Mission Planner has to receive “heartbeat packets” from the Beaglebone Blue. When trying to connect the Beaglebone Blue, Mission Planner had trouble recognizing the board due to a connectivity issue. We tried using a COM link, UDP connection, and a TCP connection but none of them seemed to fix the issue. One avenue of

troubleshooting was trying to install new drivers. The Beagleboard website lists drivers, but the drivers are not updated frequently so most boards should have the newest drivers. We downloaded the new drivers in an attempt to update them on the board, but the drivers were listed as up-to-date according to windows device manager. As a next step in troubleshooting the board, we looked at the network settings of the board. This involved installing various Debian programs to check open ports and IP addresses. Despite these steps, we still had trouble with Mission Planner connecting to the Beaglebone Blue. Ultimately, with some outside help, we found that the connectivity issue was tied to one of the configuration files we set up previously. When creating the config file for the Telemetry and GPS, it turns out that we used the wrong IP address for our UDP connection. The required IP address for the UDP connection was the IP of the computer hosting Mission Planner, but we stored the network IP address of the Beaglebone Blue. After modifying this IP address, we were able to download the correct firmware. Mission Planner now recognized the Beaglebone Blue, and we are able to calibrate it. As a group, we have started this calibration process, but we must completely assemble the drone before finishing it.

5.4.3.2 GPS Configuration

For the GPS receiver of the drone, we have the uBLOXNEO-M8N – a GPS module with a compass. We went with this GPS module because it was already present in the Robotics Club. Earlier we configured the board to send GPS communication through the UART2 port on the board, so we connected it there. To check if the GPS module is working, we installed i2c tools onto the board. These i2c tools come with the command i2cdetect, which allows the Beaglebone to scan for I2C bus devices. I2C detect uses a mapping feature to display various GPS components that the Beaglebone sees. The barometer, compass, and sensor can all be seen using the i2cdetect command. Running the command in the terminal displays a mapping of the different i2c buses. Through the i2c detect command, we detected all the i2c connections that our GPS module and Beaglebone Blue use. An example of the mapping done by i2cdetect is shown below.

```
$ sudo i2cdetect -r -y 2
0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  68  --  --  --  --  --  --
70:  --  --  --  --  --  --  76  --  --  --  --  --  --  --  --  --
```

These mappings show the three i2c buses on the Beaglebone Blue. The 0c point represents the AKM AK8933, which is the onboard compass. The 68 represents

MPU-9250 sensor from InvenSense, which is also on the board. Finally, the 76 represents the Bosch BMP280 barometer that also comes with the board

5.4.3.3 Receiver Connection

To connect the Beaglebone Blue to our R/C receiver, we needed specific connectors that connect to the SBUS port on the board. These SBUS connectors come with 4 wires that represent ground, power in, signal A and signal B (SBUS), and the same connections on each side. On the other hand, our R/C receiver uses pinout connections, and we only needed three. These three connections are for ground, power out, and SBUS. In order to connect the SBUS connectors from our Beaglebone Blue to the R/C receiver, we had to cut off the other end of the wires and splice them with wires that had female jumper connections. Also making sure to cut off the extra wire that we did not need. This R/C receiver will allow us to use an R/C transmitter to control the drone; which is especially useful during testing, so we can manually fly the drone to observe how it reacts to inputs.

6 Testing

Testing is key to ensuring proper design. All the systems available were tested for functionality and performance. This section covers the performance testing done on the radio systems obtained so far, and the flight tests of the available flight controller and its replacement with a BeagleBone Blue.

6.1 Ubiquiti Rocket M5 Testing

Due to the easy access to the Rocket M5 hardware a test was attempted relatively quickly to determine the range of such devices. Memory Mall is a large open space at UCF in which recreation happens during the day, but is relatively open at night, this seemed like an ideal space for the first test. Upon further consideration a second test was attempted between two parking garages with more desirable results. The data and exact testing procedure are detailed in the following sections.

6.1.1 Variable Distance Testing

Rocket M5 testing took place on October 12th, 2018 along with testing of the TI CC1350 Simple Link Modules. Before evaluating the results, some comments must be made about the nature of this test. This test used the Rocket M5s from Ubiquiti coupled with the AMO-5G13 Omni-directional antennas also from Ubiquiti. One antenna was connected to a laptop in front of the CFE Arena at UCF and set at a picnic table. The antenna was propped up to be approximately vertical. The second antenna was carried semi-vertically across the road and down the sidewalk along memory mall. At approximately every 50m the signal quality, described by **Equation 11**, was measured and recorded.

Equation 11. The equation for signal quality as displayed on the Rocket M5

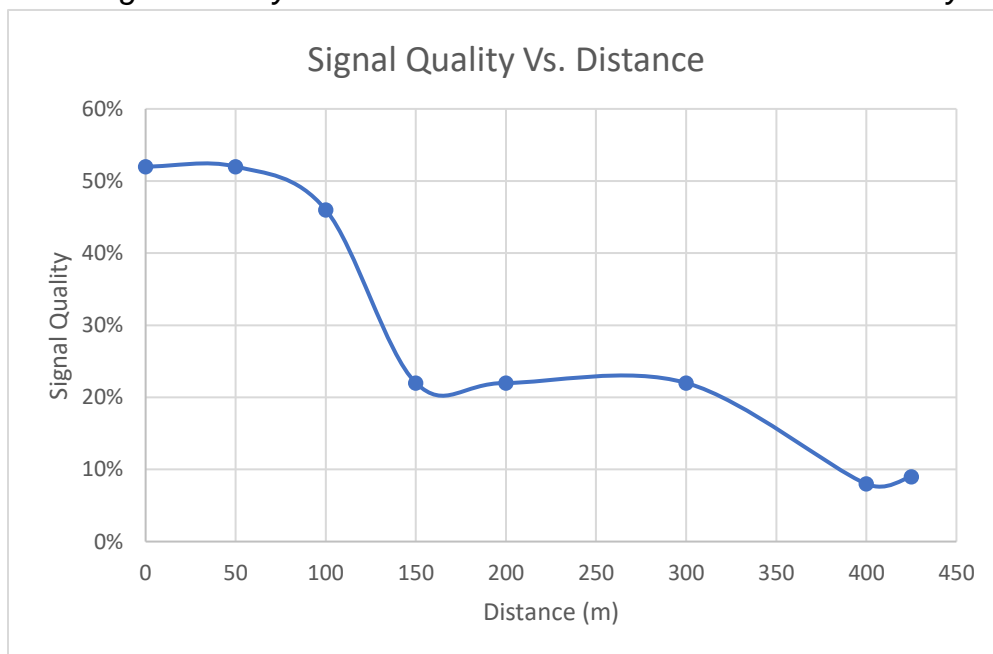
$$\text{Signal Quality \%} = \frac{\text{RSSI}}{\text{Max RSSI}} \times 100$$

Table 28. The recorded values for distance, capacity and quality as recorded in the Rocket M5 test on October 12th, 2018

Distance (m via GPS)	Capacity (%)	Quality (%)
0	88%	52%
50	88%	52%
100	83%	46%
150	73%	22%
200	68%	22%
300	57%	22%
400	40%	8%
425	41%	9%

The results above are the tabulated results of the values as measured during the test. Some important things to note are that 0m measurement was taken when the antennas were very close together but realistically closer to 1m apart, and that the data points were not evenly taken for time savings. The quality versus distance is graphed below for easier interpretation.

Figure 46. Signal Quality vs. Distance for the Rocket M5 test on Memory Mall



The first thing that is immediately apparent from the test is that the quality starts at around 50%. That may seem surprising considering that the antennas were less than a meter apart. In retrospect, this was likely caused by two major flaws in experimentation. The first flaw was that the stationary antenna was propped up on top of a metal picnic table. Since the antenna was directly on top of a metal surface there were probably Fresnel zone infractions. The other problem is that the antenna was just propped up, not held vertically and the antennas are sensitive to orientation.

The next major point that is apparent is the point at 150m where there is a huge drop in the quality to the signal. If the test was taking place in perfect free space the curve should be a smooth slightly exponential curve as the free space losses add up. The sharp drop indicates that the test was not done in an environment like free space and indeed this is the case. As the second antenna was moved further from the first many obstructions came in the way, at first light poles, then cars, and finally trees and brush. The transmitted signal was in the 5.8GHz band of course which is very sensitive to attenuation from objects. Consequently, it is not surprising that the range was a mere 425m (the length of Memory Mall) compared to the several kilometers of range other Ubiquiti users claimed when using similar hardware. After more research into Fresnel zones and radiation patterns a second test was attempted.

6.1.2 Fixed Distance Testing

Rocket M5 testing took place on October 24th, 2018. This test used the Rocket M5s from Ubiquiti coupled with the AMO-5G13 Omni-directional antennas also from Ubiquiti.

There were two major problems with the first test, Fresnel Zone infractions and orientation. Ideally the antennas were to be tested at the greatest distance possible to determine where the signal dropped off to get an approximation of range. The ideal location was a sod farm, but there was not one available in the time frame needed so the test took place between two parking garages.

The parking garages were approximately 600m apart according to Google maps. The path between the two garages was almost perfectly clear except for a few particularly tall pine trees in the line of sight. The garage with antenna 1 on top was approximately 18m and the garage with antenna 2 was approximately 12m tall. To avoid Fresnel zone infractions from the garage both antennas were elevated approximately 1.5m using wooden posts. The equipment used for this test is shown in **Figure 5**.

The antennas were tested at three different orientation to try and simulate best and worst-case scenarios that could occur during drone flight. The best case was when both antennas were vertical and due to the long range between them the aspect ratio was low. This simulated the ideal condition where the drone is relatively far away and flying horizontally such that the antenna is vertical and in the same plane

as the ground antenna. The second scenario was worse than any realistic scenario that would occur during flight. If the drone was flying at distance and was at a significant tilt. In this case, 90 degrees which is not realistic to real drone flight but gives a worst possible scenario. The final situation would never occur but was measured for curiosity. The only time the antennas would be directly pointing toward each other would be when the drone is direction over the ground station. Of course, this would occur at a few 10's of meters instead of 600m. Some connection was still maintained, but it was not reliable enough to get a transmit/receive speed.

Table 29. Results from rocket M5 testing at ~600m

Orientation of Antenna 1	Orientation of Antenna 2	Receiver Sensitivity (dBm)	Approximate Transmit/Receive Speed (Mb/s)
Vertical	Vertical	-64	40
Horizontal	Vertical	-78	15
Horizontal	Horizontal	-93	0

6.2 TI CC1350 Simple Link Testing

To evaluate unmanned aerial vehicle to ground control station and unmanned ground vehicle telemetry options, a pair of two Texas Instruments LAUNCHXL-CC1350 SimpleLink Dual-Band devices were field-tested for range and data rate performance. The devices were set to the following parameters and tested in a near line-of-sight conditions:

- EasyLink_Phy_50kbps2gfsk (-110dBm receive sensitivity) – mode 0
- No external antenna (built-in PCB trace)
- 915 MHz
- Location: UCF Memory Mall
- Date: 10/12/18
- Time: ~5 PM

A few suboptimal test conditions must be considered when evaluating the performance of these modules. These conditions include, but are not limited to:

- Sub-maximum transmission power
 - Power was set to 12 dBm
 - Maximum transmit power possible: 14 dBm
- Sub-optimal line of sight
 - Slight pedestrian and vehicle interference
 - Slight foliage interference
- Ground-level testing
 - Intermittent interference

- Unclear Fresnel zone

Although device performance may be somewhat better in optimal system use, the test results are likely to be predictive of real-world performance. Texas Instruments provides signal quality testing performance between two SimpleLink devices. The results of the tests are shown in **Table 30**. The pair of devices were tested from 0 meters to a range of approximately 300 meters, when signal tests consistently failed to complete. For device testing, a sequence of 30 packets were transmitted by the transmitting device and received by the receiving device. If a packet was missed by the transmitting device, the count of failed packets was increased.

Table 30. LAUNCHXL-CC1350 SimpleLink Dual-Band pair performance summary

Meters	Pass Count	Fail Count	Fail Rate	Average RSSI	Notes
0	30	0	0%	-30.83	
50	25	2	7%	-87.67	*Failed to complete
100	30	0	0%	-76.57	
150	28	1	3%	-81.34	*Failed to complete
200	30	0	0%	-89.13	
300	17	33	66%	-92.76	*Failed to complete

Occasionally, incomplete tests were continued by reinitiating transmission until a larger sample could be measured. In this case, the sum of pass count and fail count could exceed 30. Fail rate percentages were calculated using Equation 12. CC1350 Fail Rate Percentage.

Equation 12. CC1350 Fail Rate Percentage

$$Fail\ Rate\ \% = \frac{Fail\ Count}{Pass\ Count + Fail\ Count} \times 100$$

In addition to fail rate percentage, the average RSSI, or relative signal strength indication, for all transmissions received at a given distance was calculated using **Equation 13**. Though RSSI is not an absolute unit of measurement, RSSI denotes a relative quantization of received signal strength in decibel scale. In other words, a lower value of RSSI denotes a lower magnitude in received relative signal strength. RSSI generally decreased with range, as expected, with an RSSI of 0 indicating no loss in signal power. Notes were made for each communication test range, particularly if tests failed to complete. Some tests failed at shorter distances, likely due to changes in line-of-sight interference. A graphical plot of RSSI and fail rate percentage over distance is included in **Figure 47** and **Figure 48** respectively.

Equation 13. CC1350 Average RSSI at Range

$$\text{Average RSSI} = \sum_{i=0}^{\text{\# of packets}} \text{RSSI}_i$$

Figure 47. CC1350 Average RSSI vs. Distance

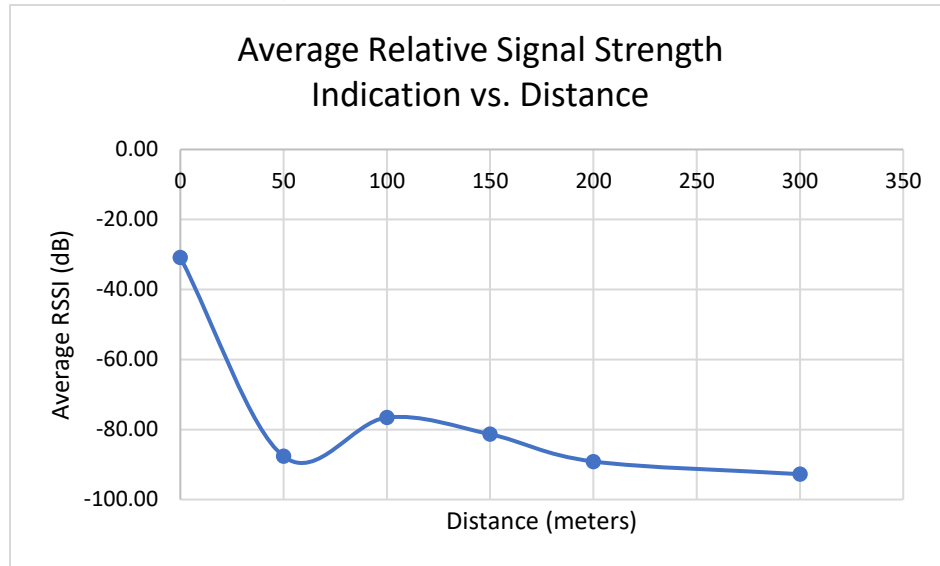
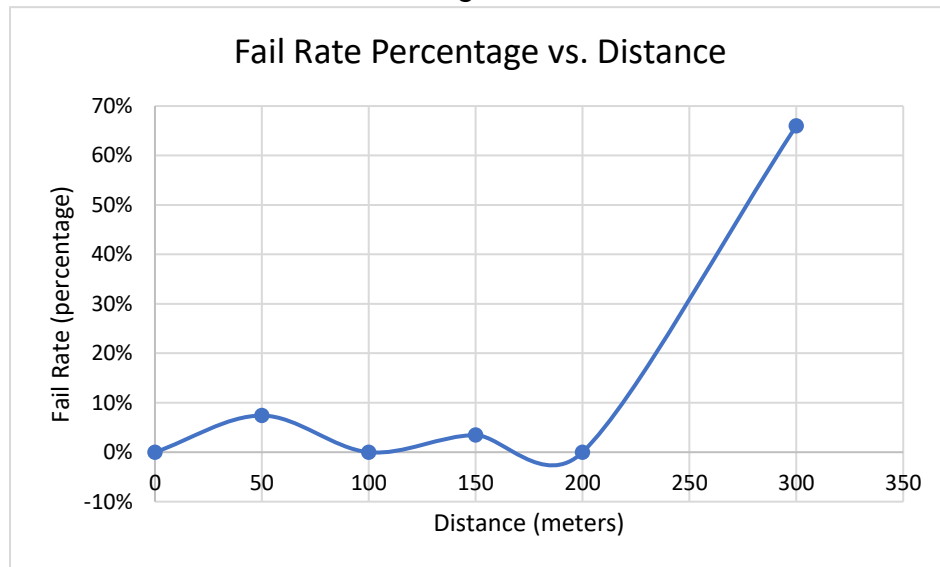


Figure 48. CC1350 Fail Rate Percentage vs. Distance



6.3 Hexacopter Flight Testing

A small hexacopter built during the summer was used to make several short test flights to practice working with sUAS technology. The hexacopter uses a DJI F550 plastic and PCB frame with integrated power distribution, 6 Turnigy Aerodrive 1000kV motors, 35A ESCs, a 3S, 6Ah, 25C Lithium Polymer battery, and a

Pixracer flight controller loaded with the PX4 flight control software. A pre-flight image of the hexacopter is shown in **Figure 49**.

Figure 49. *Hexacopter ready for flight*



The hexacopter parts were chosen based on what was available at the Robotics Club, so only 1 extra motor and the frame had to be obtained. The hexacopter was flown manually for most flights to verify the drive and stability systems worked. The first flights suffered serious stability issues, but these were corrected once the motor control wire connections were reviewed and the motors plugged into the correct ports. After this fix, the hexacopter flew very stably, both in the lab and outdoors. An image of the hexacopter undergoing flight tests is included as **Figure 50**.

Figure 50. *Hexacopter undergoing flight tests*



The next tests added a GPS to the hexacopter, to move from “stabilized” control to one of the semi-autonomous altitude holding modes. Some serious issues arose

with the Pixracer, where the altitude reported would vary in a way that suggested a broken barometer. The next step has been to switch it out with the BeagleBone Blue. Doing so required rewiring the GPS connector, and obtaining a buck-boost regulator, since the battery voltage is 11.1V nominal, but can be somewhat above or below this value. The regulator used is a Dimension Engineering AnyVolt Micro, which had suitable ratings for this test platform and was on hand. The AnyVolt Micro is a buck-boost regulator module with a built-in adjustment potentiometer that can set a wide range of output voltages. It has the same pinout as the 7805 and internal capacitors, so swapping it into the system was simple. The buck-boost topology was required since the BeagleBone Blue requires 12V, while the battery can supply anywhere from 12.6V to 10.5V. **Figure 6** shows the BeagleBone Blue attached to the power distribution board, regulator, GPS, radio, and battery from the hexacopter for testing the function of the power system and the BeagleBone's connection to the GPS and radio.

Once the BeagleBone Blue was set up, ArduPilot was installed for testing the sensors, specifically the integration with the GPS module and the external compass provided by the GPS.

6.4 Breadboard Testing

The PCB's for this project use small surface mount components that are not available in the through hole form for bread board testing. Due to the significant number of pins and nodes it would be unrealistic to test out a whole system on a bread board. Due to the low cost of printing PCB's our team opted to go straight for testing on the PCB's which will arrive late in the semester or early Christmas break.

7 Administrative Content

This section contains necessary administrative content needed to run a project including major milestones as well as the budget. This project is funded through multiple sources which is also detailed in this section.

7.1 Project Milestones

- Fully functional battery pack design
- Power distribution to required electronic systems (Flight Controller, Speed Controllers, Camera, etc.)
- Power monitoring system
- Working flight electronics (Flight Controller, Speed Controller, Camera, RF, Rotors)
- RF communication between the drone, ground vehicle, and ground station
- MAVLink communication for drone control
- Successful manual drone flight
- Autonomous drone flight (Jetson AI module)
- Autonomous image processing

7.2 Budget and Finance

Table 32 shows the value of items that were donated, the value of discounts given for the project, and sponsorship money currently raised for the electrical side of the project. The Student Government sponsorship includes the value given to all parts of the sUAS project, since it was for the overall system. **Table 32** lists materials which have already been obtained, either from sponsors sending them or things the Robotics Club already owns. **Table 31** lists the total bill of materials for the major electronics of the system.

Table 31. Bill of materials

Item	Cost Per Item	Quantity Needed	Total Item Cost
Sanyo NCR 20700 batteries	\$7.25	120+18 spare	\$1000.50
Battery Copper Strip	\$10.00	1	\$10.00
ESCs	\$17.00	8 + 8 spare	\$271.00
BeagleBone Blue	\$93.00	2 + 2 spare	\$375.00
GPS	\$48.00	2	\$96.00
TI SimpleLink sub-1GHz launchpads	\$50.00	3 + 1 spare	\$200.00
Ubiquiti Rocket m5 wireless modems	\$90.00	2	\$180.00
High speed wireless antennas	\$80.00	2	\$160.00
Motors	\$87.00	8 + 2 spare	\$868.00
Gimbal system	\$80.00	1	\$80.00
Cameras	\$200.00	3	\$600.00
NVIDIA TX2	\$300.00	1	\$300.00
Connectors			\$40.00
Battery Measurement System	\$25.00	4	\$100.00
Wire			\$40.00
Total:			\$4320.50

Table 32. Sponsorships, Discounts, and Donations

Source	Item	Value
Student Government	45% of budget	\$5761.28
Texas Instruments	BeagleBone Blue (4)	\$375.00
Texas Instruments	CC1350 and Antennas	\$500
XOAR	30% Discount on motors	\$260.64
imr Batteries	Discount on 20700 cells	\$103.50
Race Day Quads	Discount on ESCs	\$23.98
Total:		\$7024.40

8 References

- [1] AUVSI Seafarer Chapter, "Competition Rules | SUAS 2019," 2019. [Online]. Available: http://www.auvsi-suas.org/static/competitions/2019/auvsi_suas-2019-rules.pdf.
- [2] B. Ray, "RF Sensitivity: What You Need To Know For M2M Communication," December 2014. [Online]. Available: <https://www.link-labs.com/blog/rf-sensitivity-for-m2m>.
- [3] Texas Instruments, "SimpleLink-EasyLink," April 2017. [Online]. Available: <http://processors.wiki.ti.com/index.php/SimpleLink-EasyLink>.
- [4] Texas Instruments, "CC1310 SimpleLink Wireless MCU," July 2018. [Online]. Available: <http://www.ti.com/lit/ds/symlink/cc1310.pdf>.
- [5] Texas Instruments, "CC1350 SimpleLink Wireless MCU," July 2018. [Online]. Available: <http://www.ti.com/lit/ds/symlink/cc1350.pdf>.
- [6] Texas Instruments, "CC-ANTENNA-DK2," [Online]. Available: <http://www.ti.com/tool/CC-ANTENNA-DK2>.
- [7] Texas Instruments, "CC1190 RF Front End," February 2010. [Online]. Available: <http://www.ti.com/lit/ds/symlink/cc1190.pdf>.
- [8] DIGI International, "Digi XBee-PRO 900HP Module Part Numbers," [Online]. Available: <https://www.digi.com/products/xbee-rf-solutions/sub-1-ghz-modules/xbee-pro-900hp#partnumbers>.
- [9] DIGI International, "DIGI XBee-PRO 900HP Wireless Module Datasheet," 2018. [Online]. Available: https://www.digikey.com/en/datasheets/digi-international/digi-international-ds_xbeepro900hp.
- [10] Microchip Technology, "RN2483 LoRa Transceiver Module," [Online]. Available: https://www.mouser.com/new/microchip/microchip-rn2483-module/?gclid=EAlaIQobChMIgpG4iuaa3gIVjITCh3b1gLwEAAYAiAAEgKtlfD_BwE.
- [11] PX4 Dev Team, "RFD900 Long-Range Telemetry," 2018. [Online]. Available: https://docs.px4.io/en/telemetry/rfd900_telemetry.html.
- [12] Ubiquiti, "Rocket 5AC Datasheet," 2014. [Online]. Available: https://dl.ubnt.com/datasheets/RocketAC/Rocket5ac_DS.pdf. [Accessed 15 November 2018].

- [13] Ubiquiti, "dl.ubnt.com," 2011. [Online]. Available: https://dl.ubnt.com/datasheets/rocketm/RocketM_DS.pdf. [Accessed 15 November 2018].
- [14] Ubiquiti, "Bullet IP67 Datasheet," 2014. [Online]. Available: https://dl.ubnt.com/datasheets/bulletm/BulletM_Ti_DS.pdf. [Accessed 15 November 2018].
- [15] M. Brian, T. V. Wilson and B. Johnson, "How WiFi Works," HowStuffWorks, 30 April 2001. [Online]. Available: <https://computer.howstuffworks.com/wireless-network.htm>.
- [16] M. Gauthier, "Wireless Networking in the Developing World," 5 November 2009. [Online]. Available: [https://commons.wikimedia.org/wiki/File:2.4_GHz_Wi-Fi_channels_\(802.11b,g_WLAN\).svg](https://commons.wikimedia.org/wiki/File:2.4_GHz_Wi-Fi_channels_(802.11b,g_WLAN).svg).
- [17] Creative Commons Corporation, "Attribution-ShareAlike 3.0 Unported," Creative Commons Corporation, March 2007. [Online]. Available: <https://creativecommons.org/licenses/by-sa/3.0/legalcode>.
- [18] Kju, "MAC-48 Address," 27 March 2007. [Online]. Available: https://commons.wikimedia.org/wiki/File:MAC-48_Address.svg.
- [19] Creative Commons Corporation, "Attribution-ShareAlike 2.5 Generic," Creative Commons Corporation, June 2005. [Online]. Available: <https://creativecommons.org/licenses/by-sa/2.5/legalcode>.
- [20] C. Franklin and J. Layton, "How Bluetooth Works," HowStuffWorks, 28 June 2000. [Online]. Available: <https://electronics.howstuffworks.com/bluetooth.htm>.
- [21] Z. Leszek Chuchla, "Bluetooth protokoly," 4 August 2010. [Online]. Available: https://commons.wikimedia.org/wiki/File:Bluetooth_protokoly.svg.
- [22] Free Software Foundation, "GNU Free Documentation License," November 2008. [Online]. Available: https://en.wikipedia.org/wiki/GNU_Free_Documentation_License.
- [23] Z. Leszek Chuchla, "Bluetooth ramka," 4 January 2007. [Online]. Available: https://commons.wikimedia.org/wiki/File:Bluetooth_ramka.svg.

- [24] RF Wireless World, "Bluetooth vs BLE-difference between Bluetooth and BLE(Bluetooth Low Energy)," RF Wireless World, 2012. [Online]. Available: <http://www.rfwireless-world.com/Terminology/Bluetooth-vs-BLE.html>.
- [25] LoRa Alliance Technical Marketing Workgroup 1.0, "LoRaWAN What is it? A technical overview of LoRa® and LoRaWAN," November 2015. [Online]. Available: <https://lora-alliance.org/sites/default/files/2018-04/what-is-lorawan.pdf>.
- [26] Digi International, "Wireless Mesh Networking: Zigbee vs. DigiMesh," 2018. [Online]. Available: https://www.digi.com/pdf/wp_zigbeevsdigimesh.pdf.
- [27] Sparkfun Electronics, "Serial Communication," December 2012. [Online]. Available: <https://learn.sparkfun.com/tutorials/serial-communication>.
- [28] Creative Commons Corporation, "Attribution-ShareAlike 4.0 International," 25 November 2013. [Online]. Available: <https://creativecommons.org/licenses/by-sa/4.0/legalcode>.
- [29] NXP Semiconductors, "I2C-bus specification and user manual," 4 April 2014. [Online]. Available: <https://www.nxp.com/docs/en/user-guide/UM10204.pdf>.
- [30] M. Grusin, "Serial Peripheral Interface (SPI)," January 2013. [Online]. Available: <https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi>.
- [31] Battery University, "Whats the Best Battery," March 2017. [Online]. Available: https://batteryuniversity.com/index.php/learn/archive/whats_the_best_battery.
- [32] AA Portable Power Corp, "Category: Li-Ion/Polymer Single Cells," 2018. [Online]. Available: <https://www.batteryspace.com/li-ionsinglecell.aspx>.
- [33] A. Devices and D. Eddleman, "LTC4218 12V/100A Hot Swap Design for Server Farms," [Online]. Available: <https://www.analog.com/en/technical-articles/ltc4218-12v-100a-hot-swap-design-for-server-farms.html>.
- [34] Keithley, "Overview of Two-Wire and Four-Wire (Kelvin) Resistance Measurements," 2012. [Online]. Available: http://download.tek.com/document/2110_2Wire4WireKelvinResistanceAppNote.pdf.
- [35] A. Devices and O. Marcus, "Optimize High-Current Sensing Accuracy by Improving Pad Layout of Low-Value Shunt Resistors," [Online]. Available:

<https://www.analog.com/en/analog-dialogue/articles/optimize-high-current-sensing-accuracy.html>.

- [36] Linear Technology, "LTC2944 60V Battery Gas Gauge," 2017. [Online]. Available: <https://www.analog.com/media/en/technical-documentation/data-sheets/2944fa.pdf>.
- [37] Texas Instruments, "bq34110 Technical Reference Manual," August 2018. [Online]. Available: <http://www.ti.com/lit/ug/sluubf7a/sluubf7a.pdf>.
- [38] Texas Instruments, "bq34z100-G1 Technical Reference," July 2018. [Online]. Available: <http://www.ti.com/lit/ug/sluubw5/sluubw5.pdf>.
- [39] beagleboard.org, "BeagleBone Black," 2018. [Online]. Available: <https://beagleboard.org/black>.
- [40] PX4 Dev Team, "Pixracer," 2018. [Online]. Available: https://docs.px4.io/en/flight_controller/pixracer.html.
- [41] beagleboard.org, "BeagleBone Blue," April 2018. [Online]. Available: <https://beagleboard.org/blue>.
- [42] Federal Communications Commission, "Radio Spectrum Allocation," 2018. [Online]. Available: <https://www.fcc.gov/engineering-technology/policy-and-rules-division/general/radio-spectrum-allocation>.
- [43] AIR802, "FCC Rules and Regulations," 2018. [Online]. Available: <https://www.air802.com/fcc-rules-and-regulations.html>.
- [44] Flite Test, "How to build a 5.8GHz Helical Antenna," June 2012. [Online]. Available: https://www.flitetest.com/articles/How_to_build_a_5_8GHz_Helical_Antenna.
- [45] telos Systementwicklung GmbH, "I2C-Bus," [Online]. Available: <https://www.i2c-bus.org>.
- [46] Texas Instruments, "ISO154x Low-Power Bidirectional I2C Isolators," December 2016. [Online]. Available: <http://www.ti.com/lit/ds/symlink/iso1540.pdf>.
- [47] Analog Devices, "Hot Swappable, Dual I2C Isolators ADuM 125x," July 2015. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/ADUM1250_1251.pdf.

[48] "BQ34Z100-G1 Wide Range Fuel Gauge with Impedance Track Technology," July 2016. [Online]. Available: <http://www.ti.com/lit/ds/symlink/bq34z100-g1.pdf>.

[49] [Online].

[50] Anonymous, "Fresnel Zones," 2018 November 2018. [Online]. Available: https://en.wikipedia.org/wiki/Fresnel_zone.

9 Appendix

Appendix A. Permission to Use TI Materials

EM Easley, Mark <measley@ti.com>
Wed 9/12, 5:50 PM

Yes you have my consent to use the TI materials in your documentation. Just be sure to clearly reference which document it is from and it should have a unique identifier character string in the footer.

For low latency you should look at the EasyLink library first. Basically this gives you the starting point to develop your own stack

<http://processors.wiki.ti.com/index.php/SimpleLink-EasyLink>
<http://dev.ti.com/tirex/#?link=Software%2FSimpleLink%20CC13x0%20SDK%2FDocuments%2FProprietary%20RF%20Examples%20User%20Guide>

If you go into TI resource explorer and the CC13x0 SDK you will find the examples and the SimpleLink Academy training.
<http://dev.ti.com/tirex/#?link=Software%2FSimpleLink%20CC13x0%20SDK%2FSimpleLink%20Academy%2FProprietary%20RF%2FEasyLink%20Network%20Processor%20example>

There is also the TI 15.4 stack, however this would more useful if you wanted to designate one drone as the gateway and do a star network kind of setup. I think EasyLink would be the place to start.

Mark Easley
Texas Instruments
University Marketing Manager – US EAST
MCU Applications Specialist
(M) 919-491-5800
(E) measley@ti.com

BC Brandon Cuevas
Wed 9/12, 12:00 PM

Fadelv Mark <measley@ti.com> Nicholas Peterc Nicholas Omusi Garrett Goodale ✕

Hi, Mark:


I've been looking into the LAUNCHXL-CC1350 and it appears that the SimpleLink platform supports multiple wireless protocols - do you have a subset that you might recommend for our application? Our priorities are ordered as such:

- 1) Latency
- 2) Reliability
- 3) Throughput
- 4) Range

On another note, we require written consent from suppliers to reproduce datasheet snippets and logos (such as to indicate sponsorship) for inclusion in our documentation. Could you please provide us with this consent on behalf of TI? Thanks so much for your continued support.

Kindly,
Brandon

Appendix B. Creative Commons Attribution-Share Alike 3.0 Unported (CC BY-SA 3.0) Deed. Full license available from (Creative Commons Corporation, 2007)

This page is available in the following languages: 



Creative Commons License Deed

Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)



This is a human-readable summary of (and not a substitute for) the [license](#).

You are free to:

Share — copy and redistribute the material in any medium or format

Adapt — remix, transform, and build upon the material

for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:



Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.



ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.


No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Notices:

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation.

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.

Appendix C. Creative Commons Attribution-Share Alike 2.5 Generic (CC BY-SA 2.5) Deed. Full license available from (Creative Commons Corporation, 2005)

This page is available in the following languages: 



Creative Commons License Deed

Attribution-ShareAlike 2.5 Generic (CC BY-SA 2.5)



This is a human-readable summary of (and not a substitute for) the [license](#).

You are free to:

Share — copy and redistribute the material in any medium or format

Adapt — remix, transform, and build upon the material

for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:



Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.



ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.


Notices:

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation.

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.

A [new version](#) of this license is available. You should use it for new works, and you may want to relicense existing works under it. No works are *automatically* put under the new license, however.

Appendix D. Creative Commons Attribution-Share Alike 4.0 International (CC BY-SA 4.0) Deed. Full license available from (Creative Commons Corporation, 2013)

This page is available in the following languages: 



Creative Commons License Deed

Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)



This is a human-readable summary of (and not a substitute for) the [license](#).

You are free to:

Share — copy and redistribute the material in any medium or format

Adapt — remix, transform, and build upon the material

for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:



Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.



ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Notices:

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation.

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.

Appendix E. Request for permissions from Linear Technologies

To: cic.americas@analog.com

Hello,

I am a student at UCF working on a drone for the AUVSI sUAS competition as part of my Senior Design project. For Senior Design we are required to submit some supporting documentation. I would like to request permission to use the below figure from the LTC4316 datasheet in our documentation.

Sincerely,

Garett Goodale

