BRADLEY
University

Vision Based Autonomous Control of a Quadcopter

Zach Engstrom, Caleb Gill, Jeff Deeds, and Zack Woods

Advisors: Drs. Yufeng Lu, Jing Wang, and In Soo Ahn

Department of Electrical and Computer Engineering
Bradley University

December 6, 2016

Table of Contents

1. Introduction

In this project, an autonomous vision-based control system for a quadcopter is designed. A quadcopter from *3D Robotics* autonomously takes off and executes mission plans under the guidance of system vision input. An industry-standard advanced autopilot, *Pixhawk*, is used to interface airborne sensors: accelerometer, gyroscope, magnetometer, and GPS sensors [1]. The *Pixhawk* is also used to control the quadcopter's motors. The overall system has 3 different modes: diagnostic mode, manual mode, and mission mode. In diagnostic mode, the *Pixhawk* communicates with the radio-control (R/C) transmitter from the ground station. System diagnostics are performed for mission preparation. In manual mode, the *Pixhawk* accepts manual commands from the R/C transmitter. This mode is mainly used for emergency or testing purposes. In mission mode, the *Pixhawk* communicates with an airborne embedded system. The airborne system performs a video processing algorithm to track a target. To localize the target, *AprilTags*, a type of two-dimensional bar coding system, is used [2-3]. The mission plan of the quadcopter is to autonomously take off, fly to a waypoint with a predefined GPS coordinate, locate a target equipped with an *AprilTag*, and land near the target.

2. Review of Literature and Prior Work

In March 2016, a team from Queensland University of Technology developed a computer-vision based guidance system for UAV [4]. Their design is very similar to our project in the context of vision-based quadcopter control. Both use an on-board camera to guide the quadcopter for mission plans. Both use *Micro Air Vehicle Link* (MavLink*)* commands for autopilot communication, where *MavLink* is a protocol for communicating with a small unmanned vehicle. However, there are a couple of key differences between these two projects. First, the system from Queensland University utilizes a large red cloth as a target. The computer vision process mainly focuses on color matching. It becomes problematic in real-life applications. A practical approach with a different identifier (i.e. *AprilTag*) is proposed in our project. *AprilTag* is a visual fiducial system developed at the University of Michigan for robot tracking. Visual tracking algorithms are developed to detect an *AprilTag* identifier placed on the ground. As a result, relative distance and orientation of the *AprilTag* are tracked. Second, the system from Queensland University uses a python script to transmit commands to the *Pixhawk* autopilot system. In our project, the communication commands are defined in C++ header files and transmitted via serial communication to the *Pixhawk* autopilot system. The vision control system runs in the OpenCV platform on a Linux environment.

Research has also been done on *AprilTag* literature in order to understand how *AprilTag* detection functions. This system is based off of *ARTags*, which stands for augmented reality tags, but has a better detection rate and has gained popularity in robotics visual detection. The mechanism of *AprilTag* detection in a captured video frame is summarized into the nine steps below [5].

1. The color-based frame is first converted into a equal weighted grayscale format, then a gaussian blur filter is applied to smooth the sharpness from edges.
2. Local gradients are calculated at each pixel. It saves the information of magnitude and direction. In addition, a threshold is applied to reduce false positives.

3. Pixels with the same gradient direction are grouped together. It generates all edges in the frame.
4. For those pixels in a line perpendicular to their gradient, they are clustered together to thin the edges. It results in a cluster of points outlining an object's edges.
5. A line segment is fit to the the cluster of points creating a set of single pixel edges.
6. The line segments are connected end to end. The calculated gradient direction is saved as a small perpendicular line on each line segment.
7. Loops are detected, which are any connected line segments whose gradients are directed to the center of the loop. Only loops with 4 line segments are saved and are called quads.
8. Quads are decoded for *AprilTag* information. If there is no data payload detected then the quad is considered a false positive and thrown out.
9. Scenarios of overlapped tags or multi-detection tags are considered. The detected tag with the lowest hamming distance for each location is saved.

A relative distance, yaw, pitch, and roll can be calculated from the detected tag, given the information of object ID.

3. Standards and Patents Applicable to the Project

Standards

If the quadcopter is over 0.55 pounds, it must be registered with the Federal Aviation Administration (FAA). The *3D Robotics Iris* quadcopter is used in this project. It is over the weight limit and must be registered with the FAA. In addition, FAA rules related to fly a quadcopter are quoted as follows.

- The quadcopter must fly below 400 feet.
- The quadcopter must fly within visual line of sight of the user at all times.
- The quadcopter must not fly directly over people.
- The quadcopter must not fly over stadiums and sporting events.
- The quadcopter must not fly near emergency response efforts.
- The quadcopter must not fly near wildfire firefighting operations.
- The quadcopter must not fly near aircraft, especially near airports.
- The quadcopter must follow all airspace restrictions and requirements.
- The user is not allowed to fly while under the influence of drugs or alcohol.

Patents

There is no any patents pursuited with this project. The project work will not result in patented material. All work done in this project does not infringe on any existing patents. However, the *AprilTag* C++ library included in the project is licensed under GNU LGPL v2.1 software license. The library can be utilized as a part of the project without being considered a derivative work. If the library is modified, it may require the software of this project to also be filed under this same software license.

4. System Block Diagram, Specifications and Subsystems

The system block diagram is shown in Figure 1. The complete system includes the quadcopter, peripherals, and flight control embedded systems.

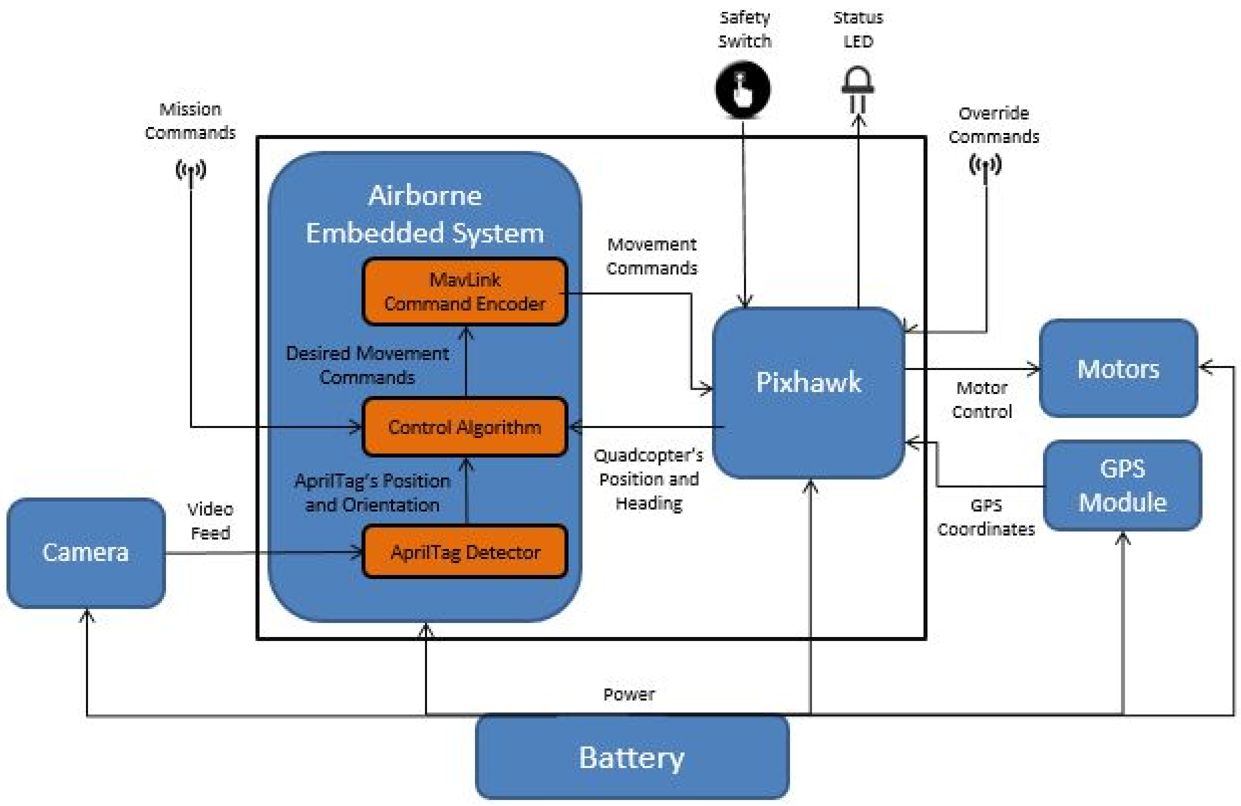


Figure 1: System Block Diagram

The system specifications are listed below.

System Specifications

Quadcopter:

- Motors : AC2830-358 850 kV Brushless Motors (4)
- Propellers : 10" x 4.7" SF (2), 10" x 4.7" SFP (2)
- Communication Range : 300 meters
- Battery : Venom 8C 3S 5100mAh 11.1V LiPo
- Flight Time : 10 minutes

Peripherals:

- GPS Module : uBlox with integrated magnetometer
- Camera : 8 Megapixels, 1080p Resolution
- Telemetry/Radio Control Frequency : 433 MHz, 915 MHz

Flight Control Embedded Systems:

- *Pixhawk* Processor : 32-bit Cortex M4
- Airborne Embedded System : Raspberry Pi 3

The system inputs/outputs are listed below.

<u>System Inputs / Outputs</u>

Pixhawk (advanced autopilot system on the quadcopter):

Inputs:

- Override Commands
- Movement Commands
- GPS Coordinates
- Power
- Safety Switch

Outputs:

- Quadcopter Position and Heading
- Motor Control
- Status LED

Airborne Embedded System:

Inputs:

- Video Feed
- Quadcopter Position and Heading
- Mission Commands
- Power

Outputs:

- Movement Commands

The overall system can operate on diagnostic mode, manual mode, and mission mode. They are described as follows.

<u>Modes of Operation</u>

<u>Diagnostic mode</u>

Pixhawk:

- The diagnostic mode is activated when the system is powered on.
- The system performs diagnostics including checking the availability of GPS, battery level, and radio-control connectivity.

- The system sends diagnostic codes to the LED.
- Once the diagnostics have passed, the manual mode is activated.

Airborne Embedded System:

- The diagnostic mode is activated when the system is powered on.
- The system performs diagnostics including checking the validity of the received video feed and quadcopter position and heading.
- The system activates an audible alarm based on the results of the diagnostic tests.
- Once the diagnostics have passed, the manual mode is activated.

Manual mode

Pixhawk:

- The system accepts movement commands from the ground station.
- The system controls the speed of the quadcopters motors.

Airborne Embedded System:

- The system does not perform operations during the manual mode.

Mission Mode

Pixhawk:

- The system receives movement commands from the airborne embedded system.
- The system provides the quadcopter's position and heading information to the embedded system.
- The system controls the speed of the quadcopter's motors.

Airborne Embedded System:

- The system receives the quadcopter's position and heading information from the *Pixhawk*.
- The system receives mission commands from the ground station.
- The system receives a video feed from an external camera and runs a real-time video processing algorithm to locate a target equipped with an *AprilTag*.
- The system sends out movement commands to the *Pixhawk*.

There are three subsystems in the project. They are *AprilTag* detector, control algorithm subsystem, and MavLink command encoder. The function of subsystems is described as follows.

Subsystems

AprilTag Detector

- The subsystem receives a video feed from a camera that is external to the quadcopter.
- The subsystem uses the video feed to detect the desired *AprilTag* and determine its position and orientation, with respect to the camera's position and orientation.
- The subsystem sends the *AprilTag*'s position and orientation to the control algorithm subsystem.

Control Algorithm Subsystem

- The subsystem receives a position and orientation of a detected *AprilTag* from the *AprilTag* detector subsystem.
- The subsystem receives mission commands from the ground station.
- The subsystem receives the quadcopter's position and heading from the *Pixhawk*.
- The subsystem uses the mission commands, *AprilTag*'s position and orientation, and quadcopter's position and heading to determine the movements that need to be made by the quadcopter to close in on the desired target's location.
- The subsystem sends desired movement commands to the MavLink command encoder subsystem.

MavLink Command Encoder

- The subsystem receives desired movement commands from the control algorithm subsystem.
- The subsystem uses the desired movement commands to convert them into MavLink commands, so that the *Pixhawk* accepts them, and the quadcopter performs the specified movements.
- The subsystem sends movement commands, now in MavLink format, to the *Pixhawk*.
- The subsystem handles serial I/O to and from the airborne embedded system at a baud rate of 57,600 bits/s.

Software Flow Charts

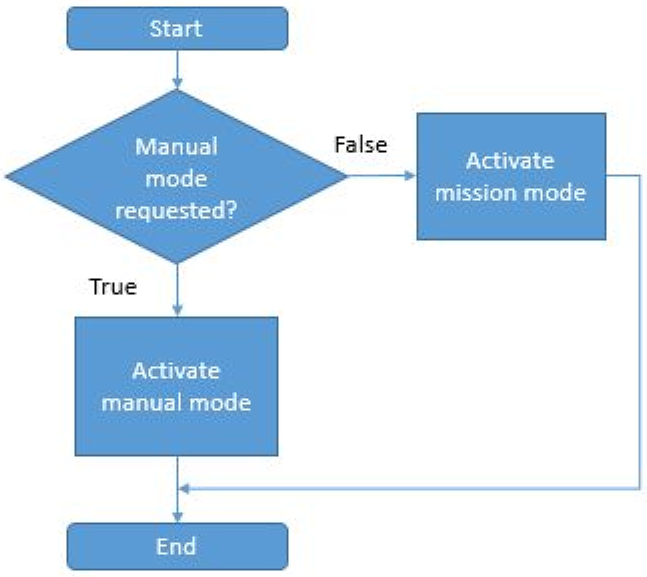


Figure 2: Interrupt Service Routine Flowchart

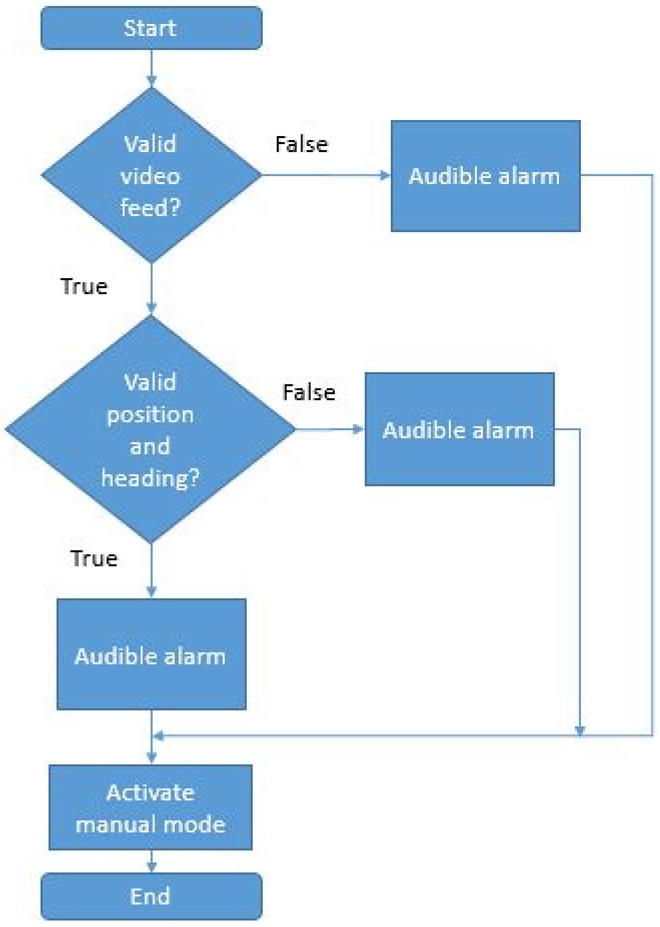


Figure 3: Diagnostic Mode Flowchart

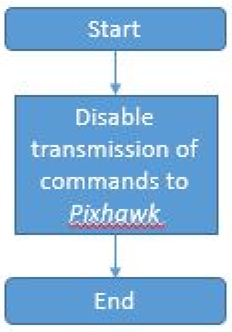


Figure 4: Manual Mode Flowchart

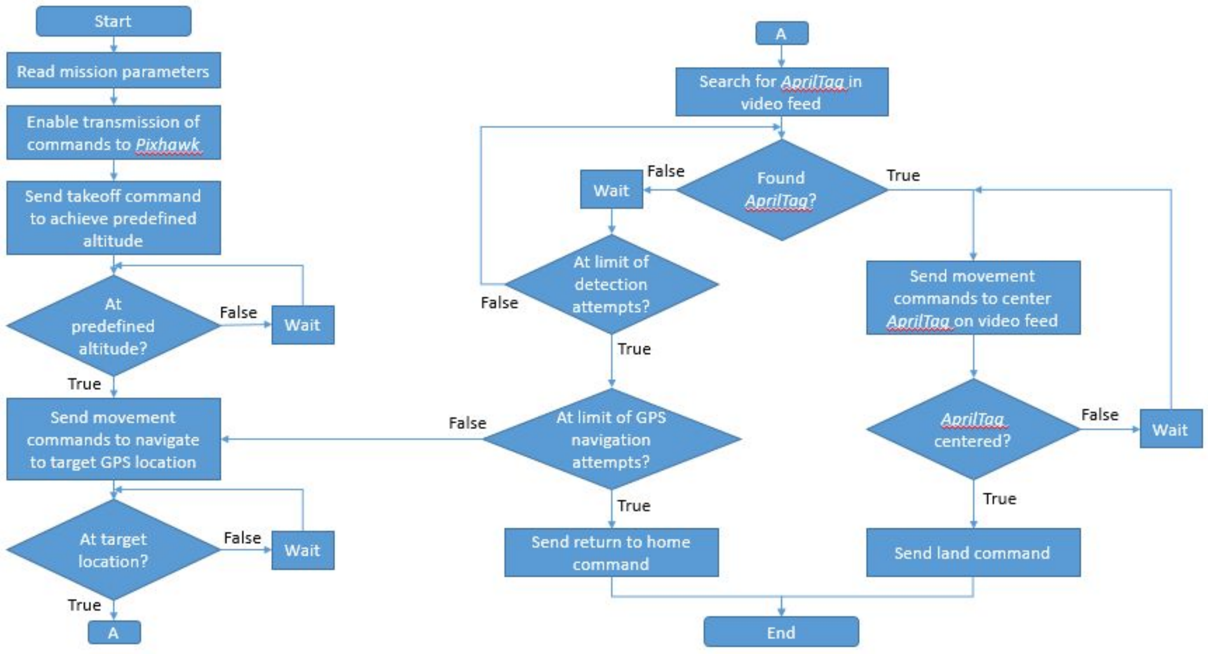
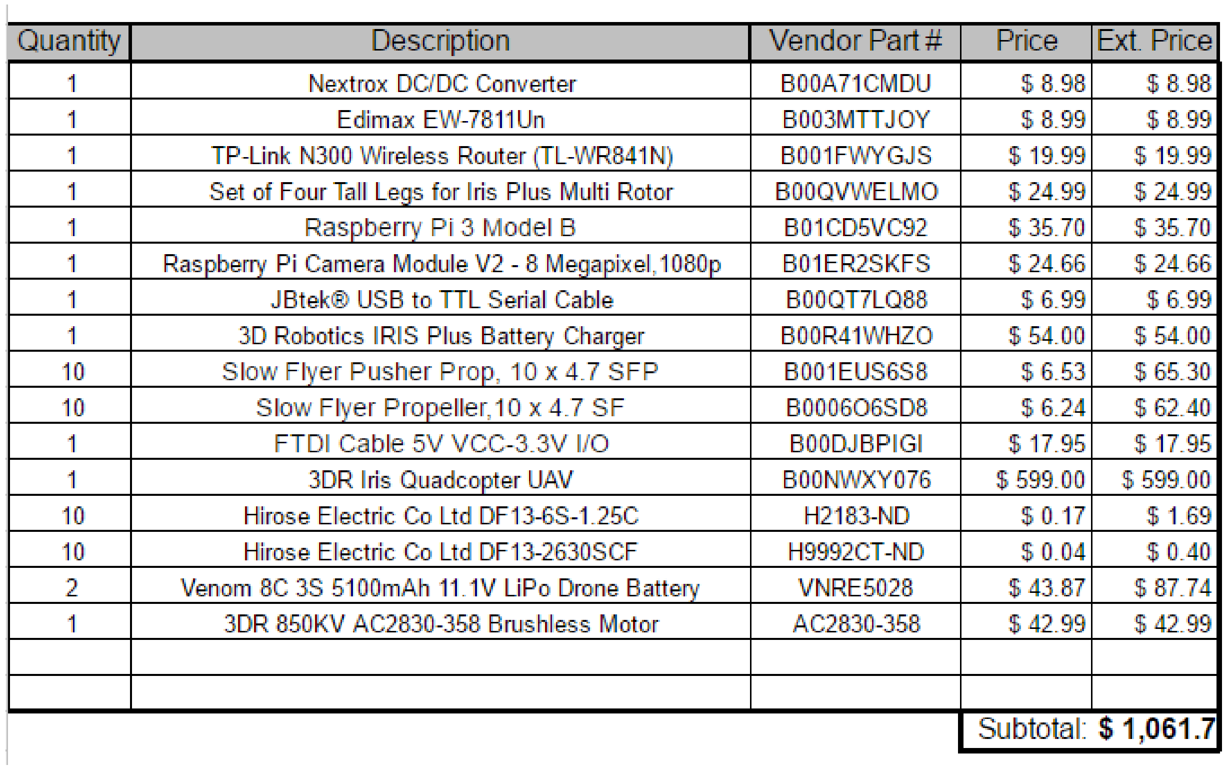

Start
Read mission parameters
Enable transmission of commands to Pixhawk
Send takeoff command to achieve predefined altitude
At predefined altitude?
False
Wait
True
Send movement commands to navigate to target GPS location
At target location?
False
Wait
True
A
A
Search for AprilTag in video feed
Found AprilTag?
False
Wait
At limit of detection attempts?
False
True
At limit of GPS navigation attempts?
False
True
Send return to home command
True
Send movement commands to center AprilTag on video feed
AprilTag centered?
False
Wait
True
Send land command
End

Figure 5: Mission Mode Flowchart

5. Engineering Efforts Completed to Date

- Researched previous work from Queensland University of Technology.
- Researched *AprilTag* detection algorithms.
- Acquired and installed autonomous flight software.
- Updated and calibrated the firmware of the quadcopter.
- Tested the quadcopter's autonomous flight mode and various flight modes included on the onboard autopilot.
- Tested and determined the quadcopter's communication range to be approximately 300 meters, which is well below the advertised range of 1 km.
- Created a Debian virtual machine and installed the necessary packages for development without the embedded device.
- Loaded the embedded device with a Linux distribution image and added necessary packages for operation.
- Configured WiFi settings on the embedded device and set up a WiFi network for remote access of the embedded device.
- Tested an *AprilTag* demo on a PC, a virtual machine, and the embedded device.
- Configured the camera module on the embedded device with various resolutions, to test framerate, delay, and detection accuracy.
- Determined which camera resolution, *AprilTag* family, and quadcopter hovering distance to use throughout the project.
- Determined the field of view of the camera module, the quadcopter's viewing area size when flying at an altitude of 5 meters, and the detectable speed of a moving *AprilTag* when the quadcopter is flying at an altitude of 5 meters.

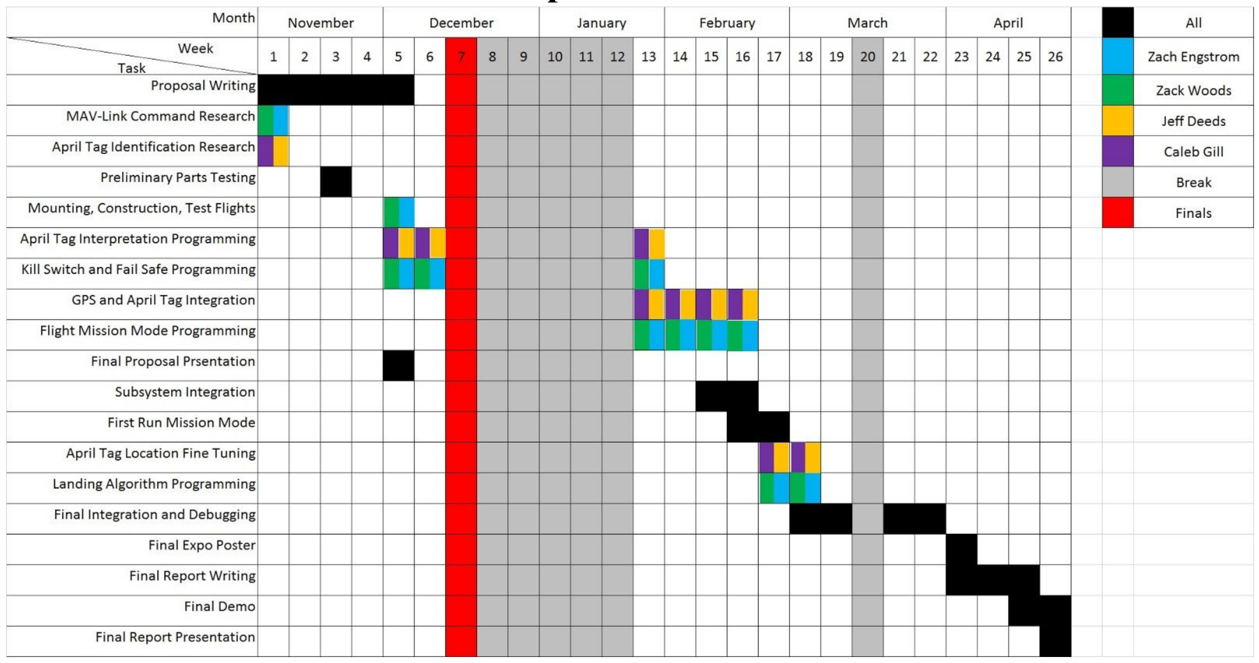
6. Parts List



Quantity	Description	Vendor Part #	Price	Ext. Price
1	Nextrox DC/DC Converter	B00A71CMDU	$ 8.98	$ 8.98
1	Edimax EW-7811Un	B003MTTJOY	$ 8.99	$ 8.99
1	TP-Link N300 Wireless Router (TL-WR841N)	B001FWYGJS	$ 19.99	$ 19.99
1	Set of Four Tall Legs for Iris Plus Multi Rotor	B00QVWELMO	$ 24.99	$ 24.99
1	Raspberry Pi 3 Model B	B01CD5VC92	$ 35.70	$ 35.70
1	Raspberry Pi Camera Module V2 - 8 Megapixel,1080p	B01ER2SKFS	$ 24.66	$ 24.66
1	JBtek® USB to TTL Serial Cable	B00QT7LQ88	$ 6.99	$ 6.99
1	3D Robotics IRIS Plus Battery Charger	B00R41WHZO	$ 54.00	$ 54.00
10	Slow Flyer Pusher Prop, 10 x 4.7 SFP	B001EUS6S8	$ 6.53	$ 65.30
10	Slow Flyer Propeller,10 x 4.7 SF	B0006O6SD8	$ 6.24	$ 62.40
1	FTDI Cable 5V VCC-3.3V I/O	B00DJBPIGI	$ 17.95	$ 17.95
1	3DR Iris Quadcopter UAV	B00NWXY076	$ 599.00	$ 599.00
10	Hirose Electric Co Ltd DF13-6S-1.25C	H2183-ND	$ 0.17	$ 1.69
10	Hirose Electric Co Ltd DF13-2630SCF	H9992CT-ND	$ 0.04	$ 0.40
2	Venom 8C 3S 5100mAh 11.1V LiPo Drone Battery	VNRE5028	$ 43.87	$ 87.74
1	3DR 850KV AC2830-358 Brushless Motor	AC2830-358	$ 42.99	$ 42.99
			Subtotal:	**$ 1,061.7**

Figure 6: Bill of Materials

7. Deliverables and Dates for Completion



Month	November				December					January				February				March					April			
Week / Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Proposal Writing	⬛	⬛	⬛	⬛	⬛		🟥	▨	▨	▨	▨	▨								▨						
MAV-Link Command Research	🟩🟦						🟥	▨	▨	▨	▨	▨								▨						
April Tag Identification Research	🟪🟨						🟥	▨	▨	▨	▨	▨								▨						
Preliminary Parts Testing			⬛				🟥	▨	▨	▨	▨	▨								▨						
Mounting, Construction, Test Flights					🟩🟦		🟥	▨	▨	▨	▨	▨								▨						
April Tag Interpretation Programming					🟪🟨	🟪🟨	🟥	▨	▨	▨	▨	▨	🟪🟨							▨						
Kill Switch and Fail Safe Programming					🟩🟦	🟩🟦	🟥	▨	▨	▨	▨	▨	🟩🟦							▨						
GPS and April Tag Integration							🟥	▨	▨	▨	▨	▨	🟪🟨	🟪🟨	🟪🟨	🟪🟨				▨						
Flight Mission Mode Programming							🟥	▨	▨	▨	▨	▨	🟩🟦	🟩🟦	🟩🟦	🟩🟦				▨						
Final Proposal Prsentation					⬛		🟥	▨	▨	▨	▨	▨								▨						
Subsystem Integration							🟥	▨	▨	▨	▨	▨			⬛	⬛				▨						
First Run Mission Mode							🟥	▨	▨	▨	▨	▨				⬛	⬛			▨						
April Tag Location Fine Tuning							🟥	▨	▨	▨	▨	▨					🟪🟨	🟪🟨		▨						
Landing Algorithm Programming							🟥	▨	▨	▨	▨	▨					🟩🟦	🟩🟦		▨						
Final Integration and Debugging							🟥	▨	▨	▨	▨	▨						⬛	⬛	▨	⬛	⬛				
Final Expo Poster							🟥	▨	▨	▨	▨	▨								▨			⬛			
Final Report Writing							🟥	▨	▨	▨	▨	▨								▨			⬛	⬛	⬛	
Final Demo							🟥	▨	▨	▨	▨	▨								▨					⬛	⬛
Final Report Presentation							🟥	▨	▨	▨	▨	▨								▨						⬛

⬛	All
🟦	Zach Engstrom
🟩	Zack Woods
🟨	Jeff Deeds
🟪	Caleb Gill
▨	Break
🟥	Finals

Figure 7: Schedule of Work

8. References

[1] Iris Operation Manual, V2. (2013). 3DR Robotics.

[2] AprilTags C++ Library. (n.d.). Retrieved November 26, 2016, from http://people.csail.mit.edu/kaess/apriltags/

[3] AprilTags. (n.d.). Retrieved November 26, 2016, from https://april.eecs.umich.edu/wiki/AprilTags

[4] Choi, Hyunwoong, Geeves, Mitchell, Alsalam, Bilal, & Gonzalez, Luis F. (2016). Open source computer-vision based guidance system for UAVs onboard decision making. In 2016 IEEE Aerospace Conference, 5-12 March 2016, Yellowstone Conference Center, Big Sky, Montana. http://eprints.qut.edu.au/93430/

[5] E. Olson, "AprilTag: A robust and flexible visual fiducial system," *2011 IEEE International Conference on Robotics and Automation*, Shanghai, 2011, pp. 3400-3407. doi: 10.1109/ICRA.2011.5979561