2012

Micro unmanned aerial vehicle video surveillance platform quadrocopter aircraft

Michael John Skadan
Eastern Washington University

Recommended Citation

Skadan, Michael John, "Micro unmanned aerial vehicle video surveillance platform quadrocopter aircraft" (2012). *EWU Masters Thesis Collection*. 44.
http://dc.ewu.edu/theses/44

Micro Unmanned Aerial Vehicle
Video Surveillance Platform
Quadrocopter Aircraft

A Thesis
Presented To
Eastern Washington University
Cheney, Washington

In Partial Fulfillment of the Requirements
For the Master of Science Degree in Computer Science

By
Michael John Skadan
Summer 2012

THESIS OF MICHAEL J. SKADAN APPROVED BY

Signature: ____ Date: _______________

Dr. Paul Schimpf, Graduate Study Committee

Signature: ____ Date: _______________

Dr. Claudio Talarico, Graduate Study Committee

Signature: ____ Date: _______________

Dr. Jacqueline Coomes, Graduate Study Committee

MASTER'S THESIS

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Eastern Washington University, I agree that the JFK Library shall make copies freely available for inspection. I further agree that copying of this project in whole or in part is allowable only for scholarly purposes. It is understood, however, that any copying or publication of this thesis for commercial purposes, or for financial gain, shall not be allowed without my written permission.

Signature: ____ **Date:** _______________

Abstract

Unmanned Aerial Vehicle Video Surveillance Platforms are beginning to play a critical role in the military, law enforcement, and search and rescue. The defense department has been the major contributor to this increase in demand. Modern warfare is increasingly fought in urban environments such as towns and cities. Urban warfare is a difficult and dangerous environment to operate in due to the presence of civilians and the complexity of the terrain. Miniature UAVs (MUAVs) offer a new approach to conducting urban combat with reduced exposure to combatant attack.

In order for a MUAV to be able to operate in an urban environment it must be able to take off, land, and navigate in confined spaces. In must also be able to communicate with an operator in order to transmit video, audio and location information in real time in a secure manner. The increased likelihood of downed UAVs in the field requires the design to be low cost and expendable. This project required the integration of mechanical, electrical, and software engineering skills to complete the design. This thesis attempts to design a MUAV that meets the requirements to operate in an urban environment. A quad rotor helicopter (quadrocopter) design was chosen for the design.

The aircraft's physical components were developed using mechanical computer aided design (CAD) software. All physically components of the design were model and assembled virtually in the CAD software prior to actual assembly. This included commercial components off the shelf (COTS) and custom designed components. The airframe's custom designed components were manufacture using a 3D printer and made out of light weight plastic. This enabled the mechanical parts to be made fast, cheap, and reliably.

The aircraft's electrical components were developed using electrical CAD software. All physically components of the design were model and assembled virtually in the CAD software prior to actual assembly. This included COTS and custom design components. The custom designed electrical circuit boards were sent to a printed circuit board (PCB) manufacturer to be produced. This enabled the electrical circuitry to be made fast, cheap, and reliably.

The aircraft's software components were modularly design to make it easier to understand and modify to make improvements. The software was written in two languages C++ and C#. C++ was used to interface at a low level with the electrical circuits. C# was used to implements the high level graphical user interface (GUI). To reduce development time, improve reliably, and aid in identifying requirements, COTS software modules and a spiral design model were used. All software was programmed and debugged in integrated development environments.

The project produced a platform with the required sensors and computing power available for more advanced software to be added later to enable the aircraft to fly missions completely autonomously. The additional software has access the GPS for waypoint navigation and video images for object recognition. The additional software would require no new hardware to be added and should cost little beyond development time to implement.

Table of Contents

List of Figures

List of Tables

List of Equations

List of Acronyms and Abbreviations

ADC – Analog to Digital Converter

AMP – Active Media Products

APC – Advanced Precision Composites

API – Application Programming Interface

BLDC – Brushless Direct Current

BOM – Build of Material

BSP – Board Support Package

CAD – Computer Aided Design

CNC – Computer Numerical Control

COTS – Components off the Shelf

CPU – Central Processing Unit

DDR – Dual Data Rate

DMM – Digital Multi Meter

DOF – Degrees of Freedom

DOM – Disk on Module

DVI – Digital Video Interface

EEPROM – Electronically Erasable Programmable Read Only Memory

ESC – Electronic Speed Controller

FAT – File Allocation Table

FET – Field Effect Transistor

FIR – Finite Input Response

GPIO – General Purpose Input Output

GPS – Global Positioning System

GUI – Graphical User Interface

I^{2}C – Inter Integrated Circuit

IC - Integrated Circuit

IDE – Integrated Drive Electronics

IIR – Infinite Impulse Response

IMU – Inertia Measurement Unit

LED – Light Emitting Diode

LiPo – Lithium Polymer

MCI – Media Control Interface

MCU - Microcontroller

MUAV – Micro Unmanned Aerial Vehicle

OS – Operating System

PATA – Parallel AT Attachment

PCB – Printed Circuit Board

PCIe – Peripheral Component Interconnect Express

PID - Proportional Integral Derivative

PNP – Plug and Play

PPM - Pulse Position Modulate

PWM – Pulse Width Modulation

RAM – Random Access Memory

RDP – Remote Desktop Protocol

SBC – Single Board Computer

SLS - Selective Laser Sintering

UART – Universal Asynchronous Receive Transmit

UAV – Unmanned Aerial Vehicle

USB – Universal Serial Bus

VTOL – Vertical Take Off and Landing

I. Introduction

The unmanned aerial vehicle (UAV) market has grown dramatically in the past decade and will continue to do so for the anticipated future. Most of the growth has been with regards to the defense department. Other areas of growth also include search and rescue, police, border patrol, and hobbyist.

Modern warfare is increasingly fought in urban environments such as towns and cities. Urban warfare is a difficult and dangerous environment to operate in due to the presence of civilians and the complexity of the terrain. In urban combat, civilians may be difficult to tell apart from combatants. Due to the presence of buildings, range of sight is limited, providing combatants with enhanced concealment, and cover as well as ample placement for booby traps and snipers. All of these situations tend to lead to higher numbers of civilian and offensive casualties.

Miniature UAVs (MUAVs) offer a new approach to conducting urban combat. MUAVs enable offensive soldiers an approach to urban combat with reduced exposure to combatant attack. MUAVs enable solders to safely navigate between tall buildings, narrow alleys, inside buildings, sewage tunnels, and subway systems from a safe location. Soldiers can perform reconnaissance for booby traps and snipers without placing themselves in harm's way.

In order for a MUAV to be able to operate in an urban environment it must be able to take off, land, and navigate in confined spaces. In must also be able to communicate with an operator in order to transmit video, audio and location information in real time in a secure manner. It is desirable for the user to communicate with the MUAV through an already familiar device such as smart phone, tablet, or laptop. MUAV's designed for urban environments will have limited flight time and maximum altitude due to their small size. This increases the likelihood of downed UAVs in the field; therefore the systems must be expendable.

II. Background

A. Vertical Take Off & Landing (VTOL) Aircraft

Vertical Take Off & Landing (VTOL) aircraft include helicopters and other rotor powered aircraft as well as fixed wing aircraft. Many approaches have been made to develop aircraft with VTOL capabilities. Some of these aircraft include crafts such as the quad rotor de Bothzat helicopter, the ducted rotor Hiller VZ-1 flying platform, the tilt rotor V-22 Osprey, and the jet powered Harrier.

Most of the early helicopter designs used multiple rotors. In the end the design with a single main rotor and anti-torque tail rotor became recognized around the world as the helicopter. The helicopter differs from most fixed winged aircraft in that it is able to take off and land vertically, hover, and fly forward, backwards, and laterally.

The de Bothzat helicopter was an experimental aircraft designed and built for the United States Army Air Service in the early 1920s. Its design utilized 4 rotating propellers and is one of the first successful VTOL aircraft. This program was later canceled due to the complexity of flight control.

The Hiller VZ-1 flying platform was an experimental aircraft designed and built for the United States Army in the 1950s. Its design utilized a single ducted fan upon which a pilot stood for lift. To control the flight direction the pilot shifted their weight to tilt the platform. Later models utilized conventional helicopter controls. Even though the center of gravity of the platform was high the aircraft was stable. This program was later canceled due to the aircrafts impracticality as a combat vehicle.

The V-22 Osprey is the world's first production tilt rotor aircraft. The aircraft has been flown by the United States military since 2007. Its design utilizes two propeller that tilt up to apply thrust vertically as a helicopter and tilt forward to give thrust horizontally as an airplane. This gives the Osprey the VTOL capabilities of a helicopter and the long range capabilities of an aircraft. The program is currently in operations.

The Harrier is a British designed military jet airplane introduced in the 1960s. It utilizes vector thrusting nozzles designed to force the jet engine thrust down to propel the airplane vertically, giving it VTOL capabilities and designed to force the jet engine thrust back horizontally to give it long range capabilities. The Harrier is currently being phased out for a more modern version, the F-35 Joint Strike Fighter.

B. Helicopter

A helicopter is a type of aircraft that utilizes a type of fan to generate lift and thrust. The modern helicopter has a fan (know as a rotor) mounted on a vertical mast over the top of the aircrafts' center of mass. The rotor is generally made up of two or more air foils (know as blades).

A mechanical apparatus know as a swashplate is connected to the helicopters rotor with hydraulic actuators. The swashplate enables the pilot to control the blades pitch as well as the overall pitch and roll of the rotor system. The ability to control the rotor system in this way allow the pilot to take off and land vertically, hover, and fly forward, backwards, and laterally.

Due to the torque generated by the rotor system spinning above the aircraft, the aircraft will spin relative to the force. To prevent this undesired sight effect a small rotor is mounted perpendicular to the main rotor at a distance from the center of mass known as the tail. This tail rotor is used to generate a force to counteract the torque generated by the main rotor to prevent the helicopter from spinning and allow the pilot to have control over the yaw of the aircraft.

C. Quad Rotor Helicopter (Quadrocopter)

A quadrocopter is a type of aircraft that utilizes 4 fans to generate lift and thrust. The fans (also known as rotors) are mounted on a plane normal of earth's gravity symmetrically around the aircrafts center of mass.

As with other helicopters the rotors are generally made up of two or more air foils (blades). The velocity of each rotor is controlled independently. This enables the pilot to take off and land vertically, hover, and fly forward, backwards, and laterally. In order to counteract the torque generated by the spinning rotors, a quadrocopter spins two of the rotors clockwise and the other two rotors counter clockwise, producing a net torque of zero. The pilot controls the aircraft's yaw by adjusting the angular velocity of the rotors spinning clockwise relative to the velocity of the rotors spinning counter clockwise.

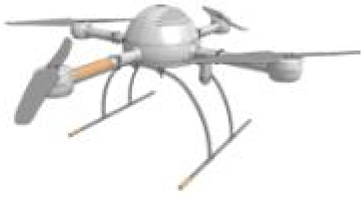
D. Advantages of the Quadrocopter Helicopter

Quadrocopters' have several advantages when compared to helicopters. Quadrocopters' utilize a simpler drive system that does not require mechanical linkage to adjust the pitch of blades and rotors. They only require controlling the rotors angular speed of rotation. The drive system thus has less mechanical parts to maintain. Quadrocopters' also divide the total thrust required for flight over all 4 rotors. This allows the quadrocopter to utilize smaller rotors. These smaller rotors possess less kinetic energy while flying than a single large rotor on a helicopter. Therefore, in the event of a collision between a rotor and an object, the overall damage will be less. Also, by utilizing multiple rotors, all independently controlled, the quadrocopter employs the possibility to maintain flight in the event of an engine failure.

E. Quadrocopter Applications

Quadrocopters have the ability to perform similar tasks as helicopters with their ability to take off and land vertically, hover for extended periods of time, and operate at low air speeds. Applications include aerial reconnaissance and surveillance, search and rescue, transportation, aerial photography, and much more.

F. Existing Quadrocopter Designs

 **Figure 1: Microdrone**	 **Figure 2: AR Drone**
 **Figure 3: Cyber Technology**	 **Figure 4: Dragonfly**

The images shown here are just a few examples of quadrocopters commercially available.

G. Project Goal

This thesis outlines a quadrocopter design that leverages components off the shelf (COTS) to reduce cost and time to completion. Due to this being a student funded project, cost must be kept to a minimum without sacrificing performance. An attempt will be made to design and build as much of the overall system as time and financing permits.

A quadrocopter aircraft design will be implement that will be capable of operating in an urban environment as required for many warfare scenarios. The device shall use concepts developed in commercial and research platforms. The quadrocopter design shall enable VTOL and confined space navigation.

The minimum goal to be achieved is integration of the subsystems necessary for the aircraft to take off, hover, travel to a destination, and land. This goal involves the design of an air frame and integration of several subsystems; a drive system, an inertia measurement system, a flight stabilization control system, a wireless communication system, a global positioning system, and a video system.

III. Design

The complexity of this project requires several components to be designed and built in parallel. The jobs include mechanical engineering, electrical engineering, software engineering, test engineering, writing and editing

Mechanical engineering is required to design and build the frame and analyze the aerodynamics.

Electrical engineering is required to design and build the power supply and control circuitry.

Software engineering is required to control the circuitry.

Systems engineering is required to integrate the components.

Test engineering is required to verify that the mechanical, electrical, and software engineering meet the specifications set out in thesis.

IV. Component Prioritization

The components of this project must be identified and sorted into the order that they will be designed and implemented. The project components will be prioritized based on the need to meet the end goal of this project.

	Component
1	Airframe
2	Drive System
3	Flight Control Circuitry
4	Wireless Communication
5	Video
6	Global Positioning (GPS)
7	Flight Stability Control Algorithm
8	Hovering Flight
9	Collision Avoidance
10	Flight to Destination

Table 1: Project Component Prioritization

The airframe is the first component designed. To design the air frame and because of the use of COTS, all the components physical dimensions will be identified in advance in order to drive the airframe design. In actuality the airframe design is an iterative process requiring adjustments to be made as the subsystems' designs are refined

The drive system will be selected based on the maximum weight of the aircraft. I have chosen to set the maximum weight to be 1500 grams in order to keep the aircraft small. The drive system must be powerful enough and light enough to efficiently lift the aircraft while remaining within the weight budget. Details of the weight budget can be seen in section X. Mechanical Design, D. Weight Budget.

The flight control circuitry will be capable of making measurements from inertia sensors, controlling the drive system for flight, wireless communicating with the pilot, capturing video, receiving location information from GPS, and detecting proximity to surrounding objects. Once the interface to the inertia measurement sensors, the drive system, the wireless communications, the video, and the GPS are chosen, the circuitry can be designed accordingly.

In order for the aircraft to fly stably, flight control software algorithms will be developed. Once the aircraft's hardware components are assembled, the algorithms will be tuned via the wireless communications. Once the algorithms are tuned the aircraft can begin to take flight and stably hover. The algorithms can then be augmented to include collision avoidance by detecting the aircrafts proximity to surrounding objects. Finally the aircraft will fly.

V. Risk Management

The complexity of this project requires a broad range of in depth skills to design and build such a device. All this coupled with limited time and resources means that proper risk management must be employed from the start.

A risk is the probability of damage occurring due to error over a given period of time. The amount of risk can be statistically expressed as the product of the probability of errors occurring multiplied by the severity of the damage. A Risk Matrix can be utilized to evaluate the quantified levels of danger associated with the likelihood of errors occurring relative to the severity of their damage. The matrix enables the display of a graphical correlation of the risk for assist management decision making.

This project has quantified the severity of the damage into the following levels;

Critical, Marginal, and Negligible. Furthermore the probability of the error occurring has been quantified into the following levels; Likely, Possible, and Unlikely.

The levels of damage are defined as follows.

"**Critical damage**" is to key multiple components & can cause entire project to fail.

"**Marginal damage**" is to one or more components & may delay project completion.

"**Negligible damage**" is to a component that has little impact on success of the project.

Based on this breakdown the risk matrix can be generated.

The combination of probability and level of damage result in the quantification of a particular risk into three levels; Low, Moderate, High.

	Negligible	Marginal	Critical
Likely	Low	High	High
Possible	Low	Moderate	High
Unlikely	Low	Low	Moderate

Table 2: Risk Management Matrix

The risk management process of this project is broken down into three fundamentals parts; Identify, Analyze, and Plan.

A. Identify

The first part of risk management for this project is to identify the components of the project that will have risk that impact the odds of meeting the end goals set out in this project. The items have been identified below in order of importance to the final product.

	Components with Risk
1	Time
2	Funding
3	Airframe
4	Drive System
5	Circuitry
6	Wireless Communication
7	Flight Stability Control Software Algorithm
8	Collision Avoidance
9	Video
10	Global Positioning
11	Changing Requirements

Table 3: Project Components with Risk

Time and funding are limited resources available to this project. Therefore these are identified as risks.

The airframe is identified as a risk due to lack of experience in airframe and mechanical engineering. In order for the aircraft to fly, the frame must be light, rigid and impact resistant.

The drive system is identified as a risk due to lack of experience with propulsion systems and mechanical engineering. The drive system is what gives this product its ability to fly. In order for the aircraft to fly the drive system must be capable of producing enough force to lift the aircraft off of the ground. It must also do so in an efficient manner to provide sufficient air time to be of use.

The circuitry is identified as a risk due to its central role in the overall functionality of the system. The circuitry is the control system that links all the electrical subsystems together in a useful way.

The communication system is identified as a risk due to its central role in enabling the designer to program the software for the avionics and upon completion of the project, enabling the operator to control the flight of the aircraft.

The flight stability control software algorithm is identified as a risk due to limited prior experience in signal processing and feedback control systems.

The aircraft's ability to avoid collision is identified as a risk due to the danger the aircraft posses to itself, other objects, and pedestrians.

The aircraft's ability to capture and transmit video in real time is identified as a risk due to the aircraft's capability of flying away from the operator and possibly out of sight.

The aircraft's ability to track its location is identified as a risk due to the aircraft's capability of flying away from the operator and possibly out of sight.

Changes in the project requirements are identified as a risk due to the complexity of this project.

B. Analyze

The second part of risk management for this project is to analyze the components identified in the first step to obtain an estimation of the probability the risk will become an issue and the severity of the impact that risk will have on the project. With the probability of the error occurring and the severity of the impact determined, using the Risk Matrix the level of risk can be determined for each component of the project.

1. Time

Meeting the minimal goals stated in II. Background, G. Project Goals is sufficient to deem the project successful. If the goals are not met, project success will be dependent on a qualitative evaluation by the graduate advisor, and additional time to complete will be granted. Complexity results in a probability of extra time, beyond 2 years, as “possible”. The impact of needing extra time to complete is “Marginal”, and thus the level of risk due to time factors is “Moderate”.

2. Funding

The likelihood of this impacting the end product of this project is “Possible.” Completing the goal is desirable to complete in full. The initial proposal takes into account estimated overall cost of all foreseeable components required for the completion of the end product with an extra percentage to buffer any unforeseen purchases. Therefore the overall impact shall be classified as “Marginal.” The calculated level of risk for this component of the project is “Moderate.”

3. Airframe

Airframe design for quadrocopters is fairly well understood thus the probability of this impacting the end product of this project is “Possible”. If the airframe fails to meet the requirements needed to fly this device, the end product of this project is of little use. The airframe is the center most piece of the entire hardware design. It is what holds all sub components in place and gives the product its overall structure. Therefore the overall impact shall be classified as “Critical.” The calculated level of risk for this component of the project is therefore “High”.

4. Drive System

The likelihood of this impacting the end product of this project is “Possible” due to lack of experience with aircraft propulsion systems. If the drive system fails to meet the requirements needed to fly this device, the end product of this project is of little use. Therefore the overall impact shall be classified as “Critical.” The calculated level of risk for this component of the project is “High”.

5. Circuitry

If the circuitry system fails to meet the requirements needed to fly the device, the end product of this project is of little use. Even with having prior electrical engineering design experience, the likelihood of this impacting the end product of this project is “Likely.” The circuitry must be able to properly interface with all subsystems to enable those systems to function. Therefore the overall impact shall be classified as “Critical.” The calculated level of risk for this component of the project is “High”.

6. Communications

Having prior electrical engineering, software design experience, and through the use of components off the shelf (COTS) the likelihood of this impacting the end product of this project is “Unlikely”. If the communications system fails to meet the requirements needed to program and fly the device, the end product of this project is of little use. Therefore the overall impact shall be classified as “Critical.” The calculated level of risk for this component of the project is “Moderate”.

7. Flight Stability Control Software Algorithm

The likelihood of this impacting the end product of this project is “Possible”. The flight stability control software algorithm enables the aircraft to automatically stabilize during flight. Without the control algorithms, the aircraft is capable of flight, but due to its natural instability such flight poses serious danger to the aircraft itself and people within its vicinity. Therefore the overall impact shall be classified as “Critical.” The calculated level of risk for this component of the project is “High.”

8. Collision Avoidance

The likelihood of this impacting the end product of this project is “Possible.” The aircraft shall fly in all directions through the air and as long as it flies in wide open skies collisions are unlikely. The aircraft is likely to cause damage to itself or obstacles around it when flown in confined spaces. Therefore the overall impact shall be classified as “Marginal.” The calculated level of risk for this component of the project is “Moderate.”

9. Video

Having prior electrical engineering experience, software design experience, video capturing experience, and through the use of COTS the likelihood of this impacting the end product of this project is “Unlikely”. Without on-board video the operator’s control over the aircraft, when not in visual range, is unpredictable and dangerous. The main concern with the video component is being able to process and transmit the video at a fast enough rate to be useful to the operator. Therefore the overall impact shall be classified as “Negligible”. The calculated level of risk for this component of the project is “Low”.

10. Global Positioning

Having prior electrical engineering experience, software design experience, and through the use of COTS the likelihood of this impacting the end product of this project is “Unlikely.” The Global Positioning System (GPS) is useful in identifying a way to return to the operator after being flown out of sight into unfamiliar territory. Therefore the overall impact shall be classified as “Marginal.” The calculated level of risk for this component of the project is “Low.”

11. Requirements Change

The lack of prior experience designing similar devices, thus leading to unforeseen changes impacting the design are “Likely”. The overall design must be flexible enough to accommodate any unforeseen changes. Therefore the overall impact shall be classified as “Critical.” The calculated level of risk for this component of the project is “High”.

The table below is a summarization of the items identified and their associated risks.

	Item	**Risk**
1	Time	Moderate
2	Funding	Moderate
3	Airframe	High
4	Drive System	High
5	Circuitry	High
6	Wireless Communication	Moderate
7	Flight Stability Control Software Algorithm	High
8	Collision Avoidance	Moderate
9	Video	Low
10	Global Positioning	Low
11	Changing Requirements	High

Table 4: Project Components and Risks

C. Plan

The next step in the Risk management process is to plan. This involves identifying alternatives and mitigating approaches for the “High” risk components of this project.

1. Time

As a full time student and employed part time, time to dedicate to this project is limited. The “Moderate” risk of this component of the project makes it of utmost concern. Possible alternatives for dealing with this constraint include bringing on other students to work on the project or making the most of the time I have available. Since there are a very limited number of students at my university interested in working on this project for their thesis, I have decided to focus on managing my time as best as possible. This means getting started early on the project. Therefore I made the decisions necessary to start working on the project prior to beginning the degree. With a plan of attack decided upon, I was able to start working on this thesis on day one.

2. Funding

As a full time student and employed part time, funding is another limited resource. The "Moderate" risk of this component of the project makes it of utmost concern. The goal of this project has been to gain as much experience in a diverse range of skills. To achieve this as much of the project as can be managed and funded will be designed rather than purchased. The custom made components will cost more than COTS part available on the market. Therefore to mitigate the finical cost COTS parts will be purchased when funding prohibits.

3. Air Frame

The airframe is a crucial component of the project with a "High" risk. With the desire to gain further experience designing mechanical aeronautical components, the initial plan is to design a custom frame. To mitigate some of the risk the design will be based upon other proven designs on the market. In the case that this approach appears too difficult to implement, the alternative is to purchase a COTS airframe. Using the alternative approach, the likelihood of this impacting the end product of this project is "Unlikely." The overall impact shall remain as classified as "Critical." The new calculated level of risk for this component of the project based on the alternative is now lowered to "Moderate".

4. Drive System

The drive system is also a crucial component of the project with a "High" risk. With a wide range of COTS parts sold, the plan is to purchase the standard parts. To mitigate some of the risk the purchase will be based upon other proven designs on the market. Using the alternative approach, the likelihood of this impacting the end product of this project is "Unlikely." The overall impact shall remain classified as "Critical." The new calculated level of risk for this component of the project is now lowered to "Moderate".

5. Circuitry

The avionics system is also a crucial component of the project with a “High” risk. With my major focusing on computer engineering and the desire to gain more experience designing avionics equipment, the initial plan is to design custom hardware and software. To mitigate some of the risk, the design will be based upon other proven designs on the market. In the case that this approach appears too difficult to implement, the alternative is to purchase COTS circuitry. Using the alternative approach, the likelihood of this impacting the end product of this project is “Unlikely.” The overall impact shall remain classified as “Critical.” The new calculated level of risk for this component of the project based on the alternative is now lowered to “Moderate”.

6. Communication

The communications system is also a crucial component of the project with a “Moderate” risk. COTS will be the approach taken and therefore there will not be an alternative approach, the likelihood of this impacting the end product of this project remains “Unlikely”, the overall impact shall remain classified as “Critical”, and the level of risk for this component of the project will remain the same.

7. Flight Stability Control Software Algorithm

The flight stability control software algorithm is an important component of the project with a “High” risk. To mitigate some of the risk the design will be based upon other proven designs on the market. In the case that this approach appears too difficult to implement, the alternative is to purchase COTS electronics with flight stability control software algorithm. Using the alternative approach, the likelihood of this impacting the end product of this project is “Unlikely.” The overall impact shall remain classified as “Critical.” The new calculated level of risk for this component of the project based on the alternative is now lowered to “Moderate.”

8. Collision Avoidance

The collision avoidance system is an extra feature with a "Moderate" risk. The aircraft can fly with or without this. This system has few example products to base its design upon. The risk was accepted as is, without alternatives to mitigate the likelihood of this system failing.

9. Video

The video system is not a crucial component of the project, with a "Low" risk. With a wide range of COTS parts sold, the plan is to purchase the standard parts. To mitigate some of the risk the purchase will be based upon other proven designs on the market. The risk was accepted as is, without alternatives to mitigate the likelihood of this system failing.

10. Global Positioning

The GPS is not a crucial component of the project, with a "Low" risk. With a wide range of COTS parts sold, the plan is to purchase the standard parts. To mitigate some of the risk the purchase will be based upon other proven designs on the market. The risk was accepted as is, without alternatives to mitigate the likelihood of this system failing.

11. Requirements Change

The requirements are "Certain" to change and planning for this is a crucial component of the project with a "High" risk. To minimize the impact that these changes will have, well defined standard hardware and software interfaces shall be utilized to simplify the modifications. With a plan of attack decided upon, the project began immediately to allow sufficient time to deal with these changes and the risk was accepted as is.

The table below is a summarization of the items identified and their new associated risks.

	Item	**Risk**
1	Time	Moderate
2	Funding	Moderate
3	Airframe	Moderate
4	Drive System	Moderate
5	Circuitry	Moderate
6	Wireless Communication	Moderate
7	Flight Stability Control Software Algorithm	Moderate
8	Collision Avoidance	Moderate
9	Video	Low
10	Global Positioning	Low
11	Changing Requirements	High

Table 5: Project Components and their Alternative Risk

VI. Aeronautical Design

This section deals with the dynamics associated flight and the design approach taken in this thesis. Thrust generated by the drive system is first analyzed and then the rigid body dynamics acting upon a quadrocopter. Finally the requirements for stable flight and aerodynamics are analyzed.

A. Thrust

Thrust is the force developed by the blades of a rotating propeller measured perpendicular to the rotor. The quadrocopter utilizes thrust to counteract the force of gravity to fly vertically and accelerate the aircraft forwards. In order for the quadrocopter to fly at a minimum the total thrust generated by the propellers must be greater than the total force of gravity acting upon the quadrocopter. Ideally the thrust must be much greater than the force of gravity to overcome possible force created by wind and other conditions. The quadrocopter's ability to maneuver is increased by having greater total thrust available. The efficiency of electric motors is not optimal when operated at full power. Therefore the thrust to weight ratio shall be chosen so that the thrust generated by the motors is equal to the weight of the quadrocopter when operated at their most efficient power level. Also the total thrust of the propellers, when operated at full power, shall be 3 times the weight of the quadrocopter.

B. Flight Mechanics (Rigid Body Dynamics)

Rigid bodies occupy space and have geometrical properties such as center of mass and moment of inertia. This characterizes motion into six degrees of freedom, three directions of translation, and three directions of rotation.

Figure 5 depicts the translational forces acting upon a quadrocopter in flight. Two forces act upon the quadrocopter while hovering. The force of gravity pulls the quadrocopter to towards earth and lift generated by the propellers pushes the quadrocopter up against gravity. When in forward flight two more forces act upon the quadrocopter, the forward thrust generated by the propellers pushing the quadrocopter forward and drag caused by air resistance.

Figure 6 depicts two of the rotational forces acting upon the quadrocopter. The spinning motors create torque. Rotors ‘A’ and ‘D’ spin in one direction, while rotors ‘B’ and ‘C’ spin in the opposite direction, yielding opposing torques. The torque generated by each motor is proportional to the angular velocity of the rotor. The net rotation of the quadrocopter due to any imbalance in the torque is referred to as ‘yaw’.

Figure 7, depicts the two other rotational forces ‘roll’ and ‘pitch’. Roll is rotation around the forward axis and pitch rotation around the side axis. These rotations are produced by differences in total thrust between pairs of rotors. Differences between front and rear cause a change in pitch and differences between starboard and port cause a change in roll.

Flight Mechanics (Rigid Body Dynamics)

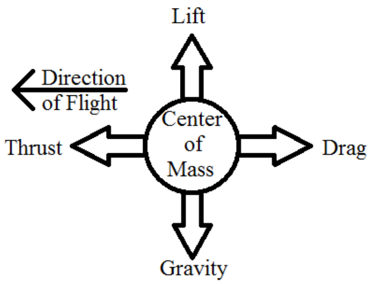


Figure 5: Translational Forces

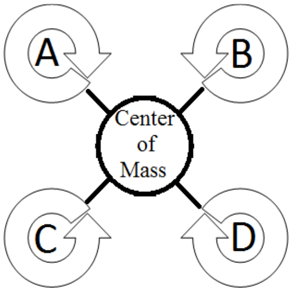


Figure 6: Horizontal Rotational Forces

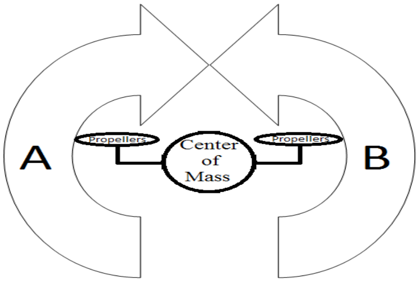


Figure 7: Vertical Rotational Forces

C. Stability

The stability of the quadrocopter is largely dependent on its center of mass location. Ideally the center of mass should be located directly between all four propellers. This gives the quadrocopter a mechanical equilibrium at that point that requires no added force to stay balanced. Any deviation from this will require some propellers to generate additional thrust to stabilize the aircraft.

D. Aerodynamics

The ability of the quadrocopter to fly efficiently depends on its drag. Drag generates force that acts against the quadrocopter's vertical lift and forward flight. Drag is generated due to the body of the quadrocopter passing through air. The drag force is dependent on the velocity of the airframe relative to the air and the surface area perpendicular to the direction of the air flow. In order to reduce the effects of drag, surfaces will be slopped in the direction of airflow as much as possible.

VII. Motor Thrust Testing

In order for this aircraft to fly, the drive system must be capable of producing sufficient thrust to lift the craft off the ground. This section describes the tests that were conducted to verify the selected drive system capabilities.

A. Thrust

To test the thrust an apparatus was built as depicted in figure 8. This device allowed the thrust of different motor and propeller and combinations to be measured. While conducting the thrust measurement, current measurements were also made. Based on these results the most efficient combination was chosen.

The test apparatus shown utilizes a right angle lever with a pivot at the corner. The angle has equal length legs with one leg vertical and the other leg horizontal, perpendicular to the ground. The vertical leg has a motor and propeller attach to it 9 inches from the pivot. The horizontal leg has a screw protruding from it 9 inches from the pivot. Underneath the screw on the horizontal leg is a digital scale. When the motor is powered the propeller spins generating thrust. The thrust causes the lever to rotate around the pivot. This rotation causes the screw to apply a force on the scale. Because the motor and screw are equal distances away from the pivot, the force the screw applies to the scale is equal to the force the motor and propeller generate. This enables the thrust of the motor and propeller to be measured.

The speed of the motor was tested originally with a servo tester with a rotating knob to adjust the speed. A digital circuit was later used to more preciously control the speed of the motor. A power analyzer was connected between the speed control and the battery to monitor the current while the motor was driven. Also, while driving the motor a tachometer was used to measure the revolutions per minute of the propeller, Figure 10.

The values in table 2 are the results from the tests of the most efficient motor propeller combination. The combination is a Turnigy L2210 1650KV 250W brushless DC motor with an APC 9”x4.7” Slow Fly propeller. See VIII Electrical Design, E Motors for more details on the motors.

Figure 8: Thrust Measurement Setup

Figure 9: APC 9x4.7 SF Propeller

Figure 10: Tachometer

Speed%	0	10	20	30	40	50	60	70	80	90	100
Thrust (g)	0	42	126	220	305	410	550	676	792	898	990
Current	0	0.44	1.61	3.15	4.75	7.15	10.6	14.5	18.6	23.2	26.7
Power (W)	0	4.9	18.1	35	53	78	115	155	197	240	274
RPM	0	2190	3720	4800	5640	6600	7530	8450	9150	9720	10230

Table 6: Motor Propeller Measurements

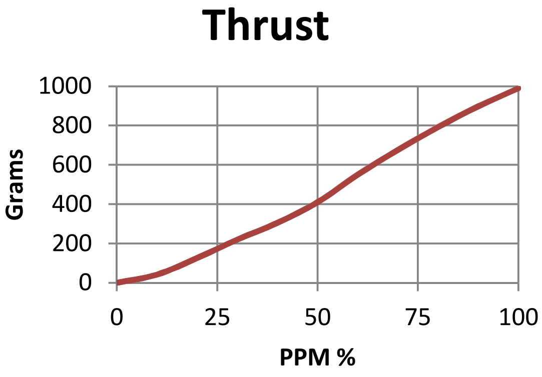


Figure 11: Motor Propeller Thrust Plot

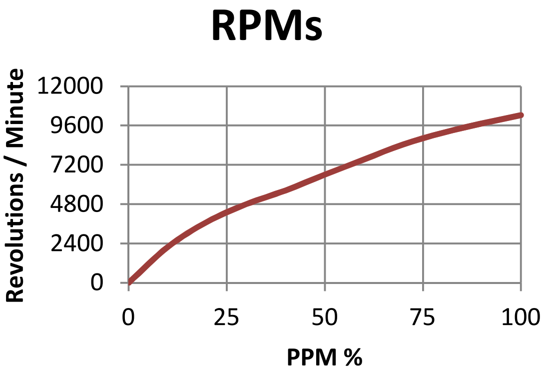


Figure 12: Motor Propeller RPM Plot

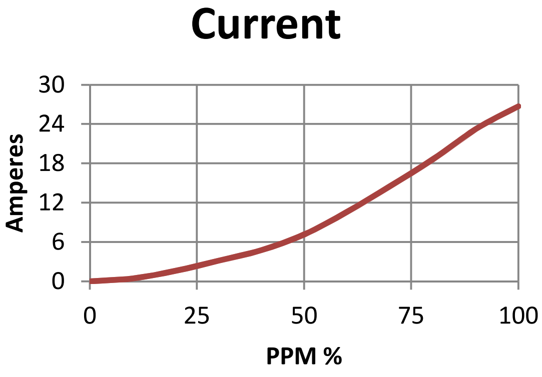


Figure 13: Motor Propeller Amps

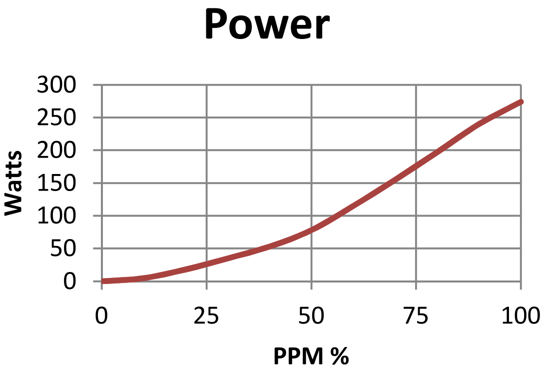


Figure 14: Motor Propeller Watts

VIII. Electrical Design

The electrical system for this aircraft is broken down into 3 main components; the power system, the drive system, and the avionic flight control system.

The power system supplies the power to fuel the aircraft. The power system is a battery. The battery must be capable of supplying a large amount of current for as long as possible while remaining light weight. The type of battery chosen is a lithium battery, specifically a Lithium Polymer (LiPo) battery.

The drive system supplies thrust to give the aircraft lift and the ability to fly. The drive system's electrical components are the motors and speed control circuitry. The motors must be capable of rotating the propellers fast enough to give enough thrust to lift the aircraft off the ground and maneuver in the air. The motor requires a special circuit known as an electronic speed controller (ESC) to operate. The ESC must be able to supply sufficient power to the motor.

The avionics system is responsible for giving the aircraft the ability to function. The avionics system is broken down into two levels; high level functionality and low level functionality. The low level functionality is controlled by a microcontroller (MCU) and embedded software running on bare metal without an operating system (OS). The high level functionality is controlled by an X86 32bit CPU running .NET software on a Windows 8 OS.

The low level functionality is responsible for flight control. The MCU mainly manages sensor readings and motor control while receiving guidance from the high level avionics. The MCU must be fast enough to execute these requirements efficiently enough to fly. The MCU must also be able to interface with all the sensors, motor controllers and high level avionics circuitry.

The high level functionality is responsible for controllable flight. The CPU manages guidance, user interfacing, video, GPS, and security. The CPU must be able to run Microsoft Windows 8 OS for the .NET software. The CPU must be fast enough to render the display graphics, capture video, capture GPS position, and compress and encrypt video and control signals sent between the user's control device and the avionics.

A. Block Diagram

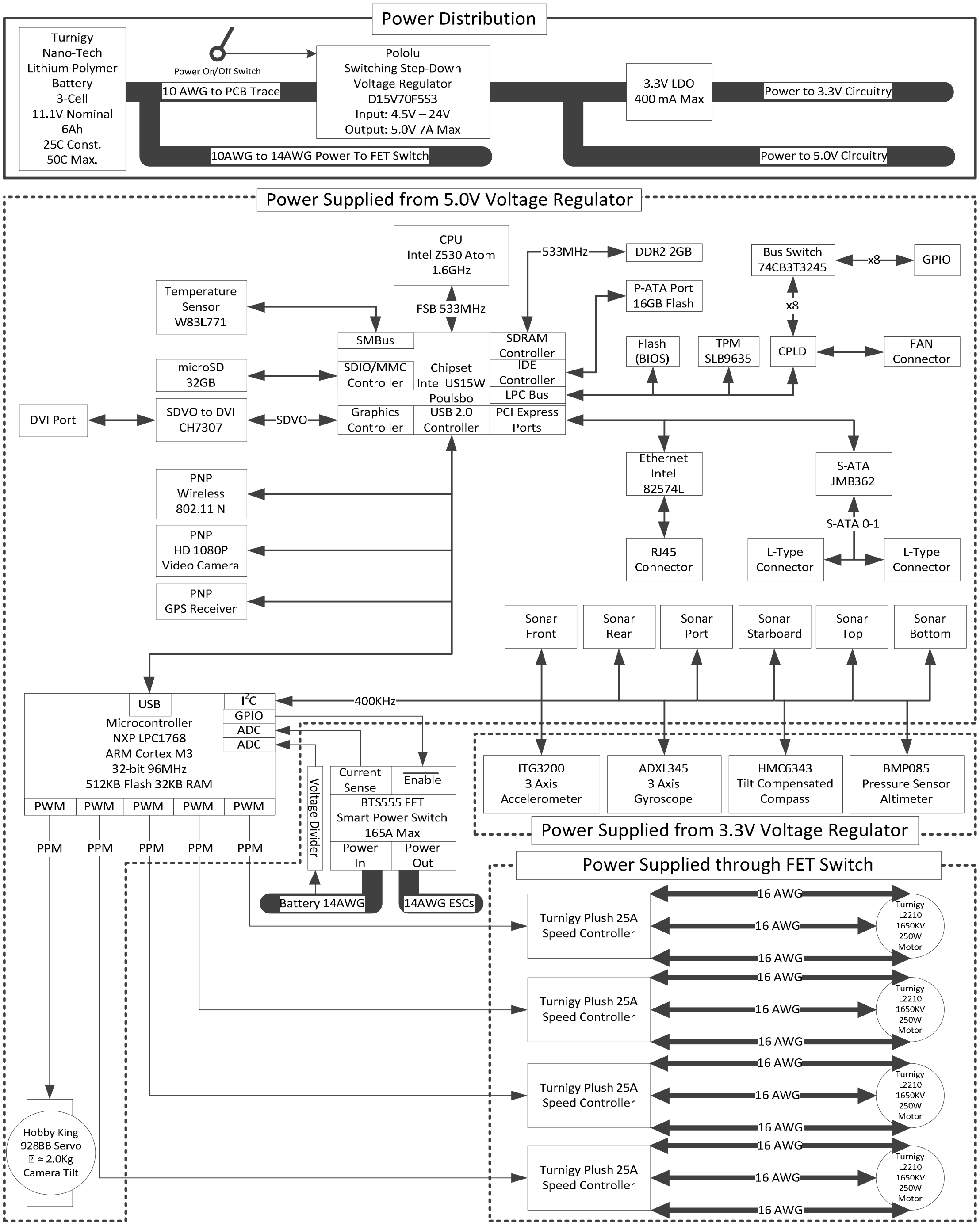


Figure 15: Circuitry Block Diagram

B. Battery

Figure 16: Lithium Polymer Battery

The battery was chosen based on the most demanding power consuming component on the aircraft, the motors, along with the weight limit requirement budgeted for the design. The battery is a Turnigy Nano-Tech Lithium Polymer (LiPo) battery. Relative to their weight LiPo batteries have high discharge capability, and high storage capacity. The cells are sealed in a flexible polymer laminated case causing them to be lighter than their equivalent hard cell with equivalent storage. The voltage of each LiPo cell is 3.7V nominal, 3.0V at its minimum charge, and 4.2V at it maximum charge.

The battery pack selected has 3 cells connected in series, giving it a total nominal voltage of 11.1V. This voltage was chosen to accommodate the motors. The battery pack used for testing this aircraft has a total capacity of 6.0Ah, although the aircraft can accommodate a battery up to 8.4Ah in capacity. The battery has a continuous discharge capability of 25C and a peak discharge capability of 50C. This means that the battery can supply 25 times the capacity continuously and 50 times the capacity peak, 6A * 25 =150A continuously current or 150A * 11.1V = 1665W continuously power and 6A * 50C = 300A peak current or 300A * 11.1V = 3330W peak power. The battery also has a high charge rate of 5C enabling the battery to be recharged fast, 1/5 hour = 12 minutes charged at 6A * 5 = 30A. The battery will be run well below this level during flight.

C. Voltage Regulation

Figure 17: Voltage Regulator

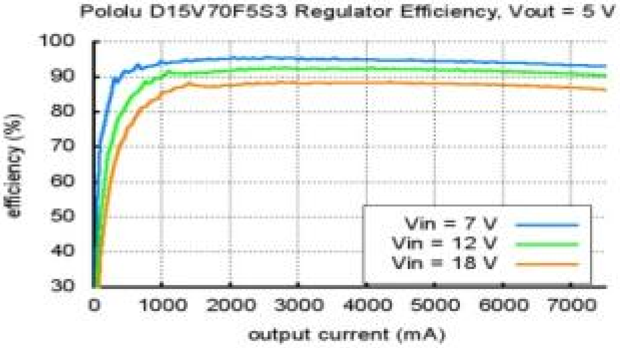


Figure 18: Voltage Regulator Output Efficiency

The circuitry is not able to operate at the voltage supplied by the battery. Therefore a voltage regulator is used to step the voltage down to 5.0V for the circuits. A Pololu switching regulator was chosen to save power due to the large voltage drop required and high current consumed by the circuitry. The regulator is capable of accepting an input voltage between 4.5V and 24V and outputting up to 7A of current.

D. Electronic Speed Controller (ESC)

Figure 19: Electronic Speed Controller

Figure 20: Electronic Speed Controller Programmer

The Turnigy Plush ESC was chosen to be capable of accepting input at a high speed and supplying the power required to operate the motor at its full speed with power to spare. It is capable of sourcing up to 25A of constant current, with a 30A burst, and receiving control input at a rate of up to 400 times per second. Power is supplied directly from the battery. The input control signal is a Pulse Position Modulate (PPM) signal sent using Pulse Width Modulation (PWM). A PPM signal is a PWM signal with a specific frequency. In the project the frequency is 400Hz, the maximum input rate at which the controller can operate. The PPM signal has a limited range of duty cycle that is valid. In this project the pulse width high time must be between 1ms and 2ms. The speed of the motor is controlled by this timing. The speed is at 0% when the pulse width equals 1ms and the speed increases as the pulse width increases up to 100% at 2ms.

A Turnigy programmer shown in figure 20 was used to setup the ESC. The ESC is configured to run on a 3 cell LiPo battery with a power cutoff voltage of 9.0V. The power cutoff is set to help prevent over discharging the battery. The cutoff is set to ramp down the power slowly so that the aircraft will not fall out of flight.

E. Motors

Figure 21: Brushless DC Motor

The motor is a brushless direct current (BLDC) out runner bell style motor with a maximum power rating of 250W. It is designed to be powered by a LiPo battery with 3 cells in series. It is a 3 phase motor that is controlled by the ESC. They are Turnigy L2210 1650KV 250W motors.

F. Single Board Computer (SBC)

Figure 22: PITX-SP (SBC)

The SBC was chosen to be capable of providing sufficient on board processing power to run Windows 8 OS, the application software GUI, and have plenty of USB interfaces for the required peripheral devices. The SBC is a Kontron PITX-SP.

G. Central Processing Unit (CPU), Chipset & RAM

The SBC contains a CPU, Chipset, and RAM.

Figure 23: Central Processing Unit

Figure 24: Chipset

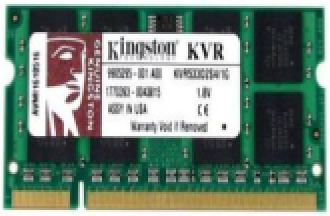


Figure 25: RAM

The processor was chosen based on the operating system requirements. It is an Intel Atom Z530 1.6GHz processor. The chip set is based on the CPU requirements. It is an Intel US15W. It has onboard Graphics, provides interface for RAM, IDE, PCIe, USB, and more. The RAM was chosen based on the chipset. The chipset supports up to 2Gb of DDR2 533MHz RAM.

H. IDE Flash Drive & MicroSD Flash Drive

The SBC is connected to an IDE Flash Drive to store the software, and MicroSD Flash Drive to data.

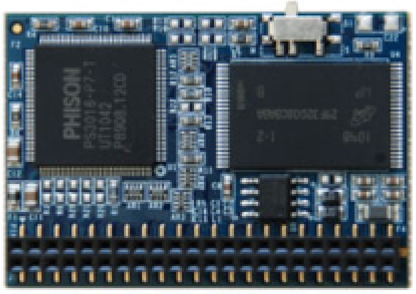


Figure 26: Flash P-ATA 16G

Figure 27: Flash

The IDE Flash Drive was chosen based on the SBC hardware interface and the OS and software requirements. The drive is an Active Media Products (AMP) Disk on Module (DOM) Type 4 with 16GB, enough to hold Windows 8 and the executable.

The micro SD flash chip is used to store captured video and can store up to 32GB worth of video. This is where the platform can store its entire mission's data, navigation coordinates, video captured, etc... The device can then be swapped in and out for different missions. The mission data can be edited and analyzed on a personal computer with a micro SD card reader.

I. Wireless Communications

Figure 28: Wireless Transceiver

The wireless communications are used to allow remote control of the aircraft. A USB Plug and Play (PNP) wireless device was chosen to enable the wireless protocol to be swapped out with minimal system impact. The wireless device used is a Buffalo 802.11N Wi-Fi module.

J. Video Camera

Figure 29: Video Camera

The video camera is used to allow the remote control user to see live video from the quadrocopter. A USB PNP video camera was chosen to enable the camera to be swapped out with minimal system impact. The camera is a Microsoft LifeCam capable of 1080p high definition video.

K. Global Positioning System (GPS)

Figure 30: GPS

The GPS is used to allow the remote control user and the quadrocopter to identify the quadrocopter's location. A USB PNP GPS was chosen to enable the GPS to be swapped out with minimal system impact. Currently the aircraft is using an ND-100s GPS module.

L. Microcontroller Unit (MCU)

Figure 31: Microcontroller

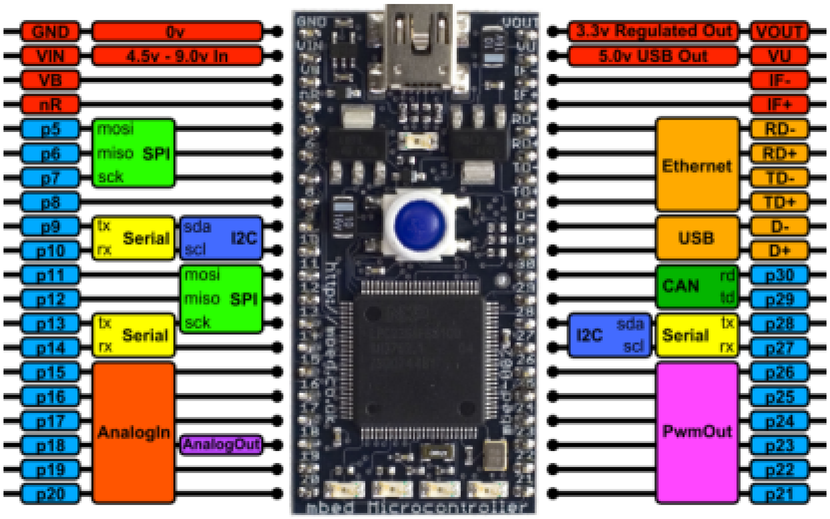


Figure 32: MBED

The MCU is used to control all low level hardware devices used for flight. The MCU is an ARM Cortex M3 32bit core running at 96MHz with 32kB of RAM, and 512kB of Flash. A MBED development board was used to control the motors, accelerometers, gyroscopes, pressure sensor, compass, and sonar. The MBED interfaces with the Intel Atom CPU through the chip set over USB 2.0.

M. Inertia Measurement Unit (IMU)

The IMU is composed of 2 modules responsible for sensing the 6 degrees of freedom (6DOF) the aircraft is capable of moving. The IMU functionality utilizes a 3 axis accelerometer IC for sensing the angle of the aircraft relative to earth's gravity and a 3 axis gyroscope IC for sensing the aircrafts rate of rotation. The IMU is also augmented with 2 more circuits, a 3axis magneto-resistive element IC for sensing the aircrafts angle relative to magnetic north and a pressure sensor IC for sensing the aircrafts altitude relative to sea level.

Figure 33: Accelerometer

Figure 34: Gyroscope

Figure 35: Altimeter

Figure 36: Compass

The Analog Devices ADXL345 accelerometer has sensors on 3 axes. It is capable of sampling at 800 samples per second with 13bit resolution. It has a selectable measurement range up to ±16g while maintaining a 4mg/LSB scale factor. It communicates with the MCU over the I^{2}C bus at 400 kHz. This sensor is used to sense the angle of the aircraft to give feedback to the control algorithm for stable flight. The accelerometer measurements are in Gs, see the XII Software Design, F Embedded Software Modules, 6 Accelerometer for details on how Gs are converted to angle.

The Invensense ITG-3200 gyroscope has sensors on 3 axes. It is capable of sampling at 8,000 samples per second internally filtered down to 400 samples per second externally with a sensitivity of 14.375 LSBs per °/sec and a full scale range of ±2000°/sec. It communicates with the MCU over the I^{2}C bus at 400 kHz. This sensor is used to sense the angular speed of the aircraft to give feedback to the control algorithm for stable flight.

The Bosch BMP085 pressure sensor is capable of operating in the pressure range of 300 — 1100hPa (+9000m — -500m above sea level). It has selectable measurement resolution of 16-19 bits. Typically it has a resolution of 0.01hPa (~8.5cm), accuracy ≤±1.5hPa, and a sample time of 17ms. It communicates with the MCU over the I^{2}C bus at 400kHz. This sensor is used to sense the altitude of the aircraft to give feedback to the control algorithm for stable flight.

The Honeywell HMC6343 compass is capable of sampling at 10 samples per second with 0.1° resolution. It communicates with the MCU over the I^{2}C bus at 400kHz. This sensor is used to determine the direction of magnetic north to give feedback to the control algorithm for stable flight.

N. Sonar

Figure 37: Sonar Module

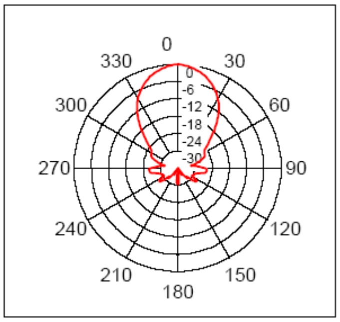


Figure 38: Sonar Manufacturers Beam Pattern

The aircraft has 6 Devantech SRF02 sonar modules, one for sensing proximity of objects is each direction; forward, backward, port, starboard, downward, and upward. The sonar is capable of detecting the distance to objects from 15cm to 249cm away with centimeter resolution. The sonar transducer has a wide beam as depicted in figure 38. It communicates with the MCU over the I^2C bus at 400 kHz. Range readings require 66ms to complete. This sensor is used to detect objects near the aircraft to give feedback to the control algorithm for collision free flight.

O. Servo

Figure 39: Servo

The servo was chosen to be strong enough to tilt the camera while remaining as light weight as possible. The servo is a HobbyKing 929MG Metal Gear Servo with 2.2kg of stall torque. The aircraft will turn to pan the camera.

P. Computer Aided Design & Manufacturing

Dip Trace PCB design software was utilized to generate Gerber files to send to the PCB manufacture. A 4 layer PCB for the inertia measurement sensors, power regulation, motor control and micro processing was designed. Figure 40 is a screen shot of the top layer of this board. The Gerber files were sent to Advanced Circuits to be manufactured.

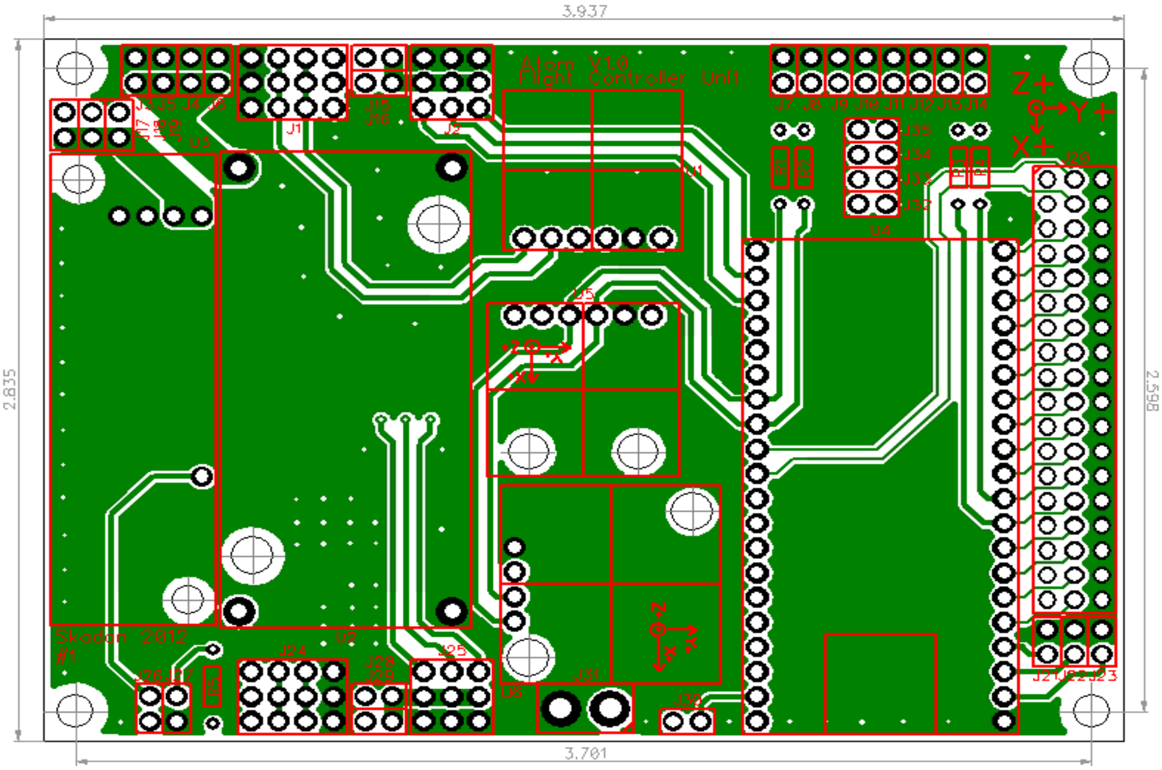


Figure 40: Flight Controller Printed Circuit Board

Q. Power Budget

During the initial design phase a power budget was produced to verify that the battery and power regulator would be capable of powering the circuitry.

Circuitry Power Break Down				
Part	Voltage	Current (Typical)	Watts (Typical)	Notes
VREG	Battery	45mA	540mW	
SBC	5.0V	1000mA	1000mW	
Wi-Fi	5.0V	100mA	500mW	
Camera	5.0V	100mA	500mW	
GPS	5.0V	55mA	275mW	
MBED	5.0V	100mA	500mW	
Accelerometer	3.3V	140uA	0.462mW	
Gyroscope	3.3V	6.5mA	21.45mW	
Altimeter	3.3V	12uA	0.0396mW	
Compass	3.3V	4.5mA	14.85mW	
Sonar	5.0V	4mA x 6 = 24mA	120mW	
Servo	5.0V	240mA	1200mW	
Total	5.0V	1.619A	4.095W	
Total	3.3V	11.152mA	36.8016mW	
Total	Battery	<2A	4.1318016W	

Table 7: Circuitry Power Break Down

The Flight Time is in minutes, the Hover Power is in Watts, and the Battery Power is in Watt Hours. Based on the data collected in the thrust test, circuitry power break down, and designing the aircraft not to exceed 1500g— The Hover Power is approximately 200W.

Running on a battery with 4.0Ah and nominally 11.1V giving a total 44.4Wh.

$$\therefore FlightTime = \frac{BatteryPower}{HoverPower} \times 60minutes = \frac{44.4Wh}{200W} \times 60minutes \cong\ 13mins$$

Equation 1: Flight Time w/ 4.0Ah Battery

Running on a battery with 6.0Ah and nominally 11.1V giving a total 66.6Wh.

$$\therefore FlightTime = \frac{BatteryPower}{HoverPower} \times 60minutes = \frac{66.6Wh}{200W} \times 60minutes \cong\ 20mins$$

Equation 2: Flight Time w/ 6.0Ah Battery

IX. Electrical Testing

After assembling the aircraft's circuitry, electrical testing was performed to verify that all over all the electrical circuitry functioned within the specifications set.

A. Power Supply

The first electrical test was to verify the power supply's functionality. The main components involved in this part of the design are the battery and the voltage regulator.

1. Battery

The battery was tested to verify that it will fully power the four motors with propellers attached. A watt meter specially designed to measure power from LiPo batteries powering motors was utilized. The device in figure 41 allowed the measurement to be made, to show that the battery had no issue sourcing up to 800W.

Figure 41: Power Analyzer

2. Voltage Regulation

The voltage regulator was tested to verify that it was able to supply the required voltage and current over the possible voltage input range. The minimum current required by the circuitry shall not go below 250mA and the maximum current required shall not exceed 4A. The regulator was loaded to source 250mA while the output voltage was monitored with a digital multi-meter (DMM). Then the input voltage was test at its minimum value the battery will supply 9V. Next the input voltage was tested at the maximum value the battery will supply 12.6V. This same test was then repeated with the regulator loaded to source 4A. The test showed that output would remain within 5% of the required output voltage, 5.0V.

B. Drive System

The second electrical test was to verify the drive system's functionality. The main components involved in this part of the design are the motors and the electronic speed controllers.

1. Motors

The motors were tested to verify that they could operate up to their maximum power rating level, 200W, using the same watt meter as used for the battery test.

2. Electronic Speed Controllers

The ESCs where tested to make sure that they could supply the power required by the motors with the same watt meter.

C. Avionics

The third electrical test was to verify the avionics functionality. The components involved in this part of the design are the SBC, wireless transceiver, video, GPS receiver, MCU, accelerometers, gyroscopes, altimeter, compass, and sonar.

1. SBC (CPU, Chipset, RAM, Flash, and Graphics)

The CPU, Chipset, RAM, Flash, and Graphics were tested by loading an Operating System (OS) "Windows 8"onto the Flash and booting the OS. The OS booting and video was displayed verifying that they were functioning correctly.

2. Wireless

With the OS loaded the wireless module was plugged into the USB and its functionality was verified. Windows automatically detected the device. The internet was then accessed and data transfer rates were viewed in Windows Task Manager.

3. Video

With the OS loaded the video camera was plugged into the USB. Its functionality was verified using the software provided by the manufacture.

4. GPS

With the OS loaded the GPS was plugged into the USB. Its functionality was verified using the software provided by the manufacture.

5. MCU

With the OS loaded the MCU was connected to the USB. Its connectivity was verified using Windows Device Manager. The device shows up as a virtual COM port.

6. Accelerometer

The accelerometer was tested over the I^2C bus. A request for its ID was sent from the MCU and a reply was received with the expected ID number.

7. Gyroscope

The gyroscope was tested over the I^2C bus. A request for its ID was sent from the MCU and a reply was received with the expected ID number.

8. Pressure Sensor (altimeter)

The pressure sensor was tested over the I^2C bus. A request for its ID was sent from the MCU and a reply was received with the expected ID number.

9. Compass

The compass was tested over the I^2C bus. A request for its ID was sent from the MCU and a reply was received with the expected ID number.

10. Sonar

The sonars were tested over the I^2C bus. Request for their IDs were sent from the MCU and replies were received with the expected ID numbers.

X. Mechanical Design

Utilizing the aeronautical and electrical design, requirements mechanical designs were developed using computer aided design (CAD) software. The CAD enabled the weight of all parts to be tracked and the overall weight of the entire system to be totaled. It also allowed all parts to be fitted together virtually to assure that there would be no parts interfering with each other during actual assembly.

A. Computer Aided Design Drawings

The follow are just a few of the CAD Drawings generated during the design of this project.

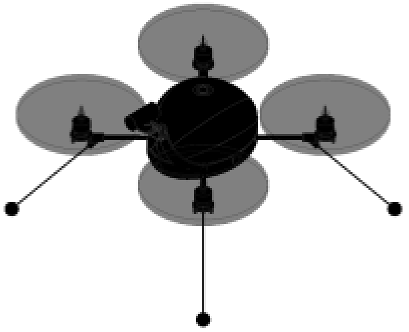


Figure 42: CAD Isometric View of Aircraft

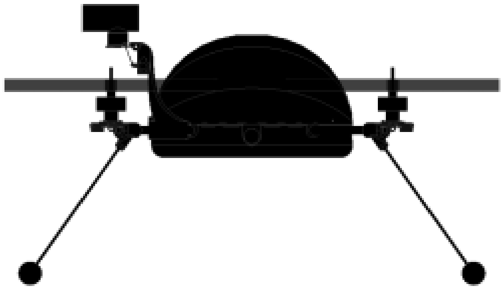


Figure 43: CAD Side View of Aircraft

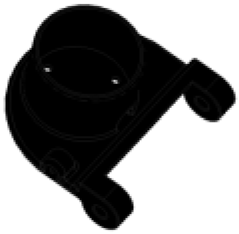


Figure 44: CAD Camera Mount

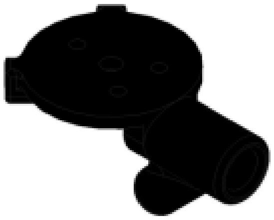


Figure 45: CAD Motor Mount

B. Materials

The airframe is built out of custom made nylon parts and standard carbon fiber tubes. The nylon parts are a durable plastic printed from a polyamide (nylon) powder. The color is white and somewhat flexible depending on the size of the geometry. It is porous and has a slightly textured finish.

C. Manufacturing

The nylon parts were designed in Alibre and made on a Selective Laser Sintering (SLS) 3D printing machine. This machine uses a laser beam to fuse small particles of plastic together, layer by layer. A thin layer of fine plastic powder is spread over the bed of the machine. A laser passes over the powder selectively fusing it together based on a cross section of the 3D model. Once the first layer is complete, the bed lowers and a new layer of powder is spread over the work. The process is repeated until the parts have been printed. The un-sintered powder acts as the support material and once the part is complete it is removed using air. All stereo lithography files were sent to Ponoko to be made into nylon parts.

D. Weight Budget

Component Weight Break Down		
Part	**Weight (grams)**	**Notes**
Air Frame	**280**	All Plastic and Carbon Fiber Parts
Motors	**200**	4 motors w/ wire
ESCs	**88**	4 ESCs w/ wire
Battery	**481**	1 LiPo 6.0Ah w/ wire
SBC	**260**	Computer w/ RAM & Flash Drive
Flight Controller Circuit Board	**50**	PCB w/ all components attached
Wi-Fi	**8**	USB Dongle
Camera	**40**	Camera w/ cable
GPS	**20**	USB Dongle w/ cable
Propellers	**40**	4 propellers
Servo	**9**	1 Servo
Wire	**20**	9 ft 14AWG
Total	**1496**	

Table 8: Component Weight Break Down

A weight budget was created to keep track of the overall weight of the aircraft to make sure that it remained under the limit of 1500g for flight analysis. The weight budget is shown in table 4.

XI. Mechanical Testing

The purpose of the mechanical testing was to verify that the aircraft components met the strength and weight requirements for flight.

A. Deflection

The carbon fiber tubes were tested by the manufacturer. A 3-point bend test was conducted on the tubes. The test consisted of a 26" span with a 2.0lb mass suspended from the center. The deflection was then measured. The 0.375" outside diameter tube deflected 0.065". The 0.22" outside diameter tube deflected 0.419". The larger tubes are used as the arms to attach the motors. While in hover the larger tubes will experience a static load equal to ½ the weight of the aircraft. Therefore the larger tube will experience a static load of ~1.65lbs and bend less than 0.065". The aircraft is not design for aerobatics and the dynamic loads will be kept to a minimum. The smaller tubes are used as the legs (landing gear). The legs are used to dampen the impact of the landing and therefore are more flexible.

B. Weight

The final weight of the aircraft fully assembled came to a total of 1498 grams.

XII. Software Design

This section outlines the architecture of the software designed to control the aircraft. First the requirements will be examined followed by an in depth analysis of the software architectural components of the design.

A. Requirements

The software architecture consists of code that runs on two processors, the ARM and the Atom. Therefore the architecture has been divided into two sections, one for each processor.

Section one is the embedded code developed to run on the ARM MCU and written in C++ code. The section must be capable of communicating and controlling the sensors (accelerometers, gyroscopes, compass, altimeter, and sonar) and motors while also communicating to the second section.

Section two is .Net code developed to run in Windows on the Atom CPU written in C# code. The section must be capable of displaying the GUI, interfacing with the video camera, the GPS receiver, the wireless transceiver, and section one, the embedded code. The GUI uses the on board graphics hardware and sends the display to the host over RDP. By setting up the graphics in this way the pilot's control device, regardless of its OS, can view the GUI, without having to have specifically developed custom code for each platform.

The following data flow diagram shows the sources and destinations of the data in the system. It also shows the relationship between the embedded software, the .Net code and the pilot's control device.

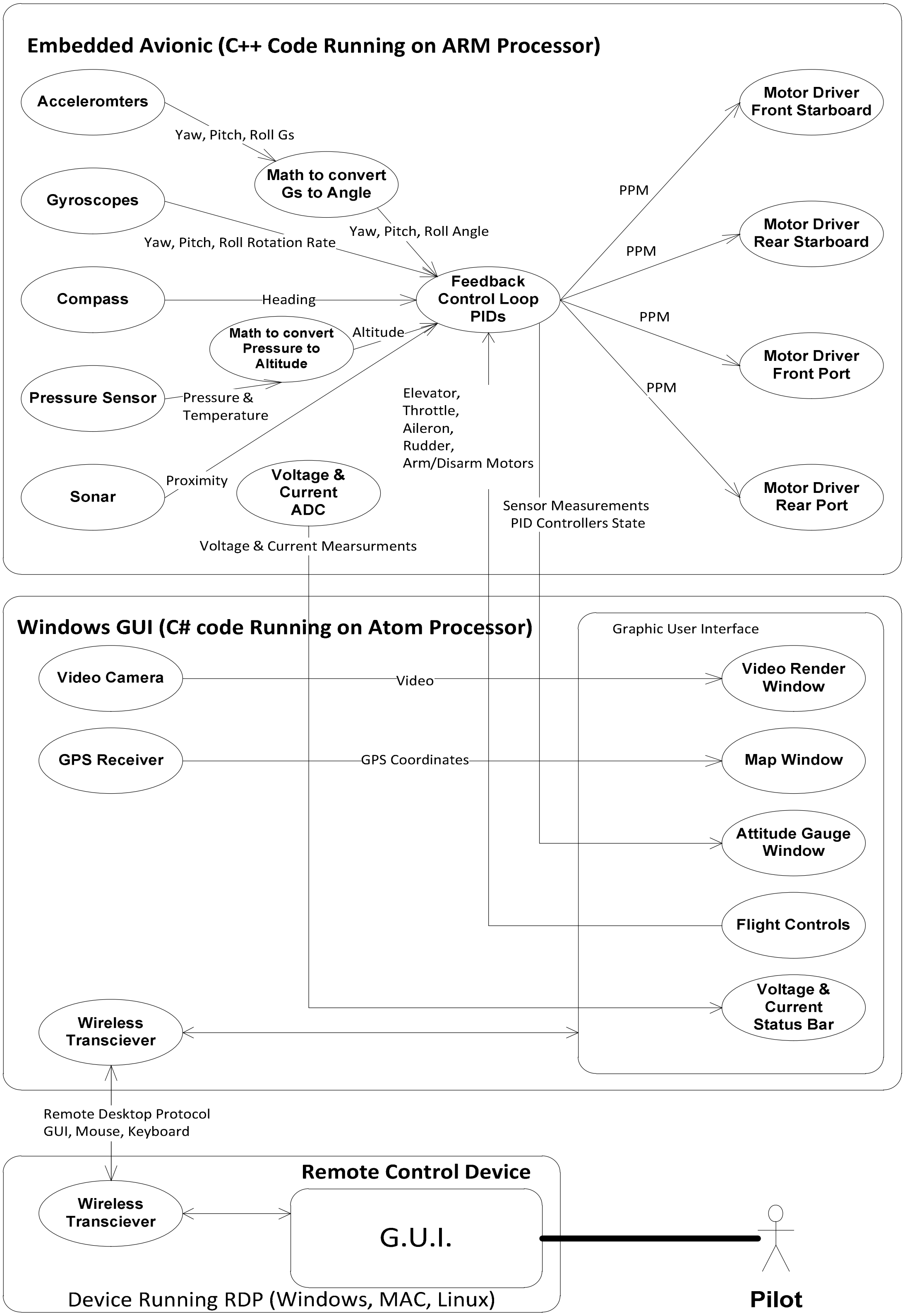


Figure 46: Software Design Data Flow Diagram

B. Embedded Software Source File Relationship

This diagram illustrates the relationship between the C++ software files utilized to implement the embedded functionality. The arrows indicate the direction of function calls between the files. Methods and attributes with the "+" sign are accessible in all files and those with "-" are only accessible within the file they are declared.

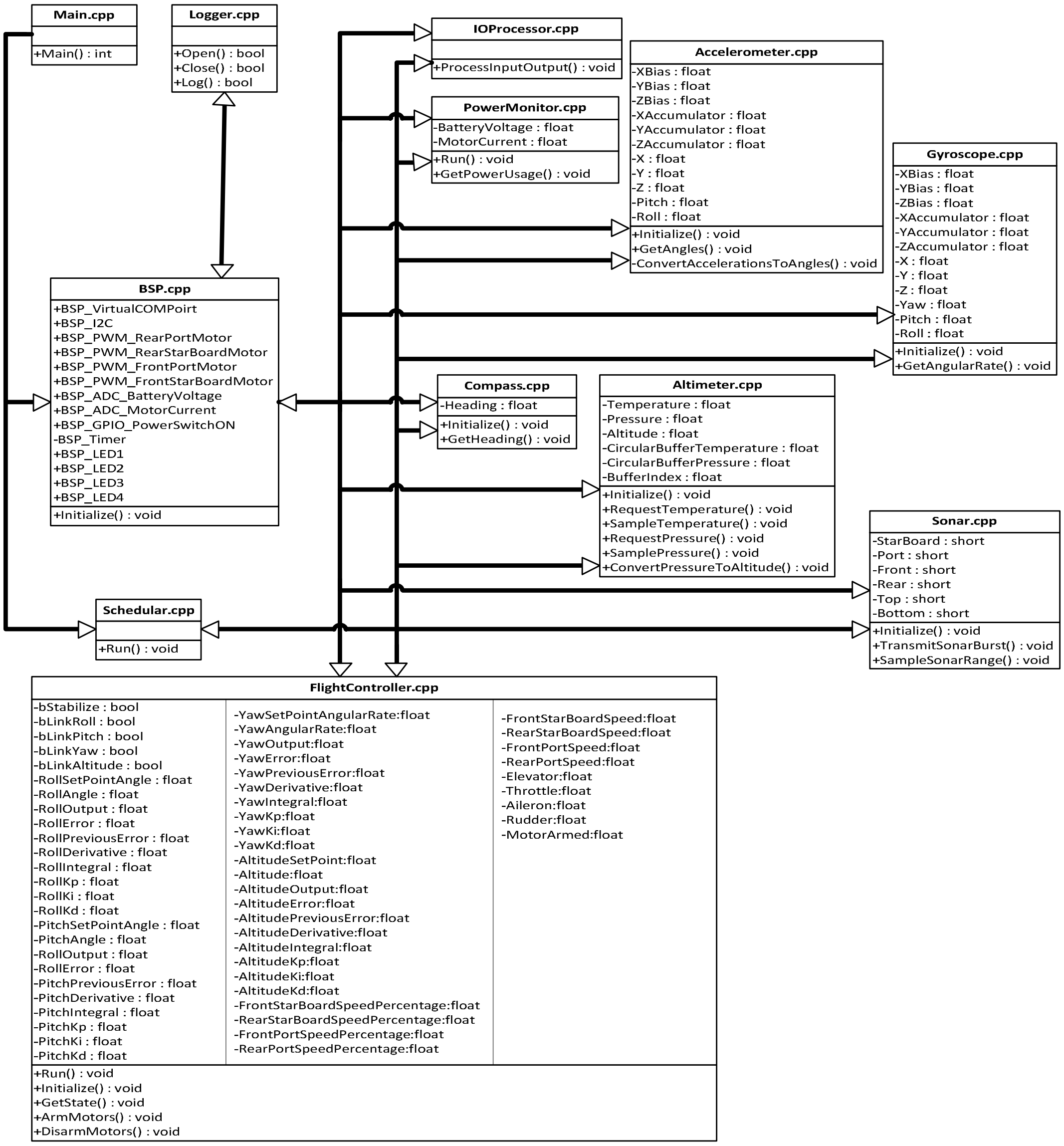


Figure 47: Embedded Software Source File Relationship Diagram

1. Main

The embedded software begins executing in the main module when power is supplied to the MCU. The Main module is responsible for properly setting up the board support package (BSP) and starting the scheduler.

2. Board Support Package (BSP)

The BSP initializes the circuitry and abstracts the printed circuit board configuration details away from the software design. The BSP sets up abstract interfaces for managing the signals, busses, and devices on the circuit board. The processor's pins are configured for their required functionality (input/output, PWM, ADC, I^2C, Frequency, etc…) and peripheral circuitry is initialized.

First the data Logger is configured. Then a UART bus is set up to connect as a virtual COM port over the USB hardware to the Application software. An I^2C bus is set up to connect to the sensors. The PWM hardware is configured to interface with the motor's speed controllers and servo. Then two ADC channels are configured to interface to the power monitoring circuitry. Finally two GPIOs are configured to control the heartbeat LED and the FET power switch used for safety and power reduction to supply power to the motors only when armed.

The UART bus baud rate is configured to operate at 921600. The I^2C bus frequency is configured to run at 400 kHz. The PWM interfaces that control the brushless motor driver circuitry are configured to produce PPM signals with the required period of 2.5ms and pulse width between 1ms and 2ms.

Once complete the system is ready to begin executing the scheduler.

3. Scheduler

The scheduler is in charge of managing the timing of the entire system.

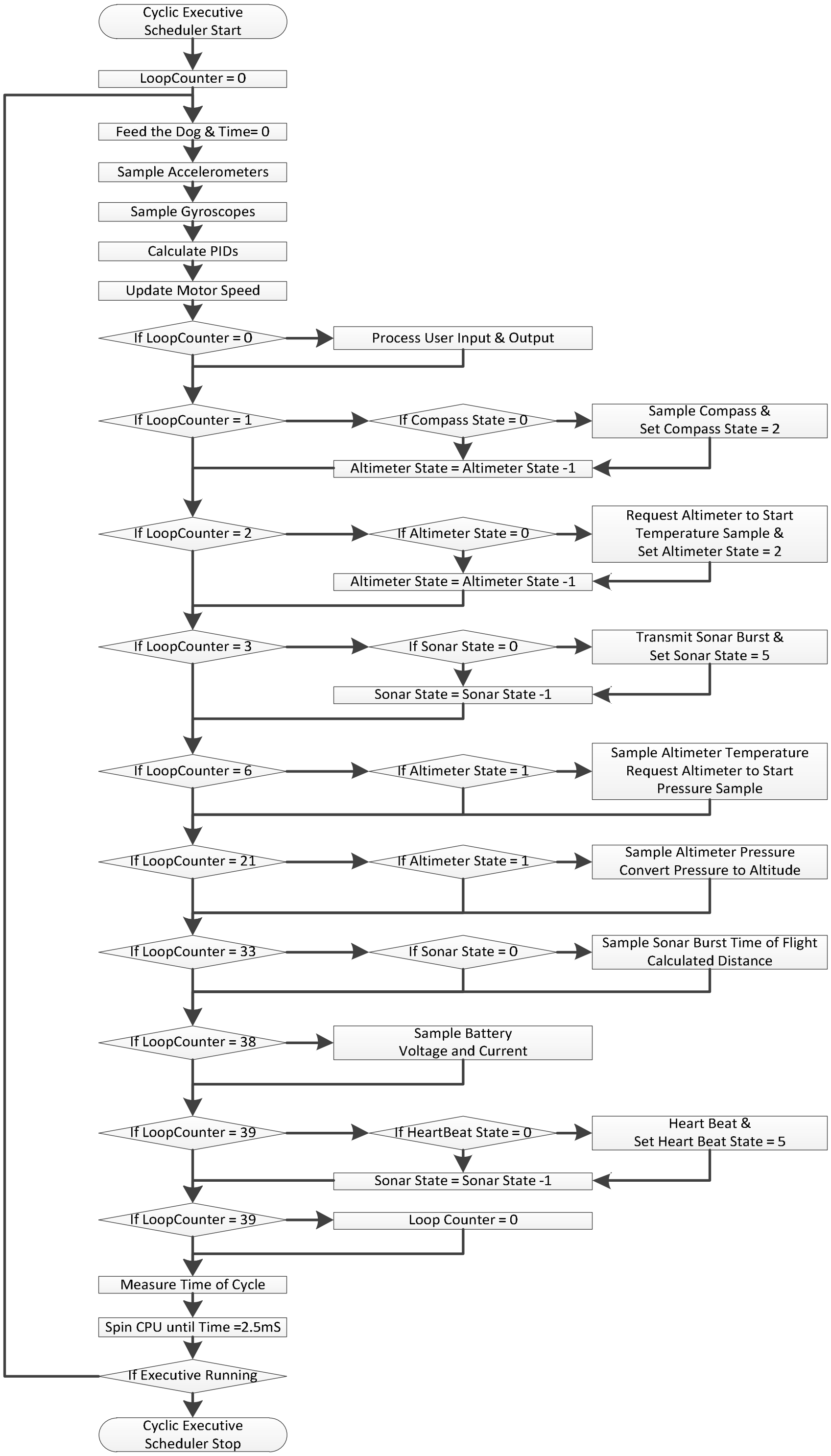


Figure 48: Embedded Software Flow Diagram

The scheduler executes specific sections of code at predetermined hard coded times in the order required and operates as a cyclic executive. It manages the execution as illustrated in figure 48. The scheduler consists of a loop that is designed to repeat every 2.5ms.

All iterations of the loop execute code to feed the watch dog timer, sample the accelerometers, gyroscopes, calculate new feedback control values, updated the motors' speeds, and idle the processor so that the loop cycles once every 2.5ms. By doing so the motor's speeds are updated at a rate of 1S/2.5ms=400Hz.

If the dog is not fed for an extended period of time, meaning that the Idle task has not been executed in an extended period of time, then the dog will die. This is managed with the hardware watch dog timer. The timer is counting down to zero and feeding the dog adds to the amount of time before the counter reaches zero. If the counter reaches zero, a task has failed to execute properly, and the MCU will be automatically restarted. This feature is utilized to allow the system to recover from unexpected software execution failure and possibly prevent the aircraft from crashing.

All hardware sensors are not capable of or required to respond at the rate of 400 times per second. A loop counter is used to divide the frequency by 40, to give a frequency of 400Hz/40=10Hz.

A LED is toggled every half a second. This is referred to as a heartbeat and gives the developer a visual indication that the systems is properly running. A variable counter is used to further divide the frequency down so the heartbeat will toggle at the required rate.

The compass is only capable of supplying new samples every 200ms. A variable counter is used so the sample compass code only executes every 200ms.

The pressure sensor requires several commands to be sent to the circuit. After each command a delay must be implemented before the data is ready to be sampled. The pressure sensor uses a counter variable to implement these timing requirements.

The sonar modules function in a similar way. A command is sent to the modules to make distances measurements. The modules transmit ultrasonic bursts of sound, count the time until they receive echoes back, and then calculate the distances to the objects the sounds reflected off of by dividing the time of flight by two times the speed of sound in air. A counter variable is used to wait a sufficient amount of time for the process to complete and then code is executed to retrieve the measurements that were made.

4. Logger

The Logger module is utilized for debugging the hardware and software. First a FAT32 files system is set up in Flash memory. A hardware timer is configured to allow the logger to time stamp any data it records. The Logger creates a file named "Log.txt" in the file systems. All modules in the embedded software can use the Logger to log events. The file can be viewed in a text editor.

5. Accelerometer

The Accelerometer module is responsible for interfacing with the accelerometer circuit. It sends commands to the hardware and receives measurements for each of the 3 axis. It repeats this process at a rate of 400 samples per second. The sensor measurements are relative to gravity, measured in grams, and must be converted to angles for pitch and roll.

A simplified example of how to calculate the tilt angle with 1 axis measurement for the X axis is shown below in figure 49 and equation 3.

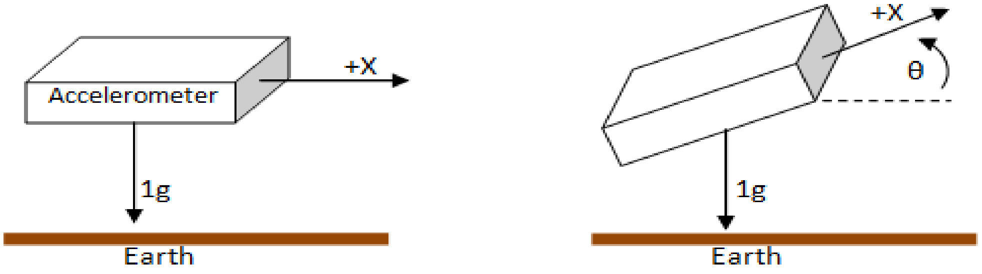


Figure 49: Single Axis Accelerometer Measurement to Angle

Using basic trigonometry, the projection of the gravity vector on the x-axis produces acceleration equal to the Sine of the angle between the accelerometer's x-axis and the plane orthogonal to the gravity vector. Based on this ideal situation, the X-axis acceleration is

$$A_x[g] = 1g \times \sin\theta$$

Equation 3: Acceleration on Axis of Tilted Accelerometer

Conversion from acceleration to angle in radians then only requires using the inverse sine function. Conversion to degrees is done by multiplying the resulting angle in radians by (180/π).

$$\theta = \sin^{-1}\left(\frac{A_x[g]}{1g}\right)$$

Equation 4: Angle of Axis of Tilted Accelerometer

By combining this type of transformation across the 3 axis of acceleration measured, the angles for pitch and roll can be calculated. The following equations were utilized to calculate the angles for pitch and roll using the 3 measurements from the accelerometer.

$$\rho = tan^{-1}\left(\frac{A_X}{\sqrt{(A_Y^2 + A_Z^2)}}\right)$$

Equation 5: Pitch Acceleration to Angle

$$\Phi = tan^{-1}\left(\frac{A_Y}{\sqrt{(A_X^2 + A_Z^2)}}\right)$$

Equation 6: Roll Acceleration to Angle

ρ defines the angle of the X-axis relative to ground. Φ defines the angle of the Y-axis relative to ground. A_X, A_Y, and A_Z are the axis measurements from the accelerometer multiplied by the acceleration of gravity 9.81m/s^2. The accelerometer is mounted in the aircraft with the positive Y axis pointing towards the front. Therefore **ρ** equals the aircraft's roll and Φ equals the aircraft's pitch. For more detail on the mathematics of sensing accelerometer inclination, refer to the appendix on "Using an Accelerometer for Inclination Sensing."

These calculations are based on the assumption that the accelerometer is at rest and not experiencing any other acceleration besides that of gravity. While the aircraft is in flight other acceleration forces will be experienced. Researches of other projects that have implemented similar functionality have not indicated that these forces adversely affect the performance of these calculations's ability to calculate the angle of the accelerometer.

6. Gyroscope

The Gyroscope module is responsible for interfacing with the gyroscope circuit. It passes commands to the hardware and receives measurements from it. It repeats this process at a rate of 400 samples per second. The measurements are received in units of degree per second for the rate of rotation. These values are used as is, without any further conversions. The measurements are used to sense the aircrafts Yaw, the rate of rotation around the aircrafts vertical axis. Internally the gyroscope IC samples at 8000 samples per second and filters out noise in the measurements using an Infinite Impulse Response (IIR) filter.

7. Compass

The Compass module is responsible for interfacing with the compass circuit. It sends commands to the hardware and receives heading measurements in degrees. These measurements identify the aircraft's heading relative to magnetic north. It repeats this process 5 times per second. Internally the compass IC filters out noise using and IIR filter.

8. Altimeter

The Altimeter module is responsible for interfacing with the pressure sensor circuit. It sends commands to the hardware and receives measurements from it. The measurements are in units of Pascal. It repeats this process once every second and filters out noise in the measurements using a moving average Finite Impulse Response (FIR) filter. See section 12 below for more on the Moving Average Filter.

With the measured pressure $\mathbf{p}$ and the pressure at sea level $\mathbf{p_0}$ the altitude can be calculated with the international barometric formula.

$$altitude = 44330 \times \left(1 - \left(\frac{p}{p_0}\right)^{\frac{1}{5.255}}\right)$$

Equation 7: International Barometric Formula

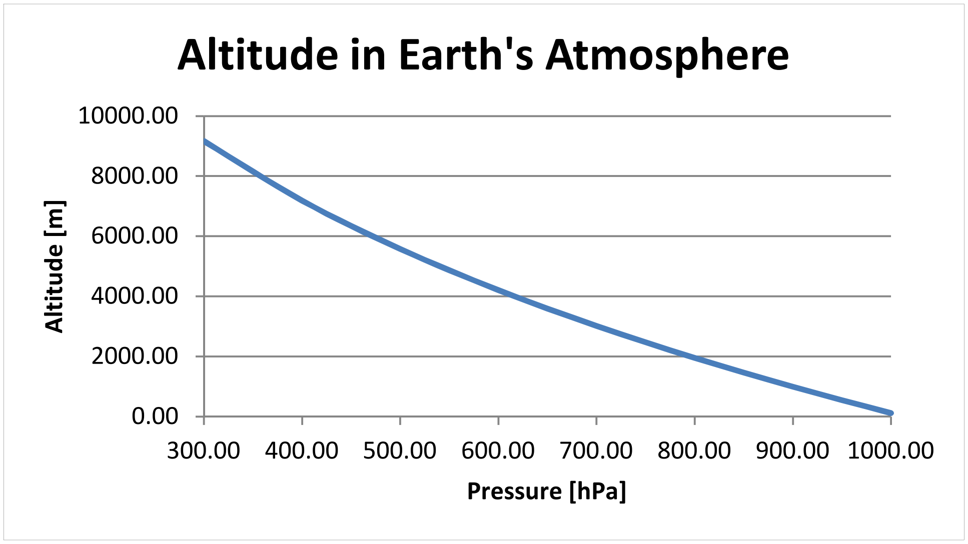


Figure 50: Plot of Pressure to Altitude

As shown in figure 50, the pressure $\mathbf{p_0}$ at sea level is 1013.25hPa.

9. Sonar

The Sonar software module is responsible for interfacing with the sonar hardware modules. The software module sends commands to the hardware modules and receives measurements in centimeters, of the distance to objects in front of the sonar hardware modules. The calculations to convert the echo delays to distances are performed by the sonar hardware modules. This process is repeated 2 times per second. During flight the Flight Controller module calls into the Sonar software module to get proximity measurements.

10. Flight Controller

The Flight Controller module is responsible for interfacing with the motors' speed controllers, accelerometer module, gyroscope module, compass module, altimeter module, sonar module, and the user input output processor module to enable stable controlled flight. Figure 51 illustrates the flow of data as it applies to the Flight Controller.

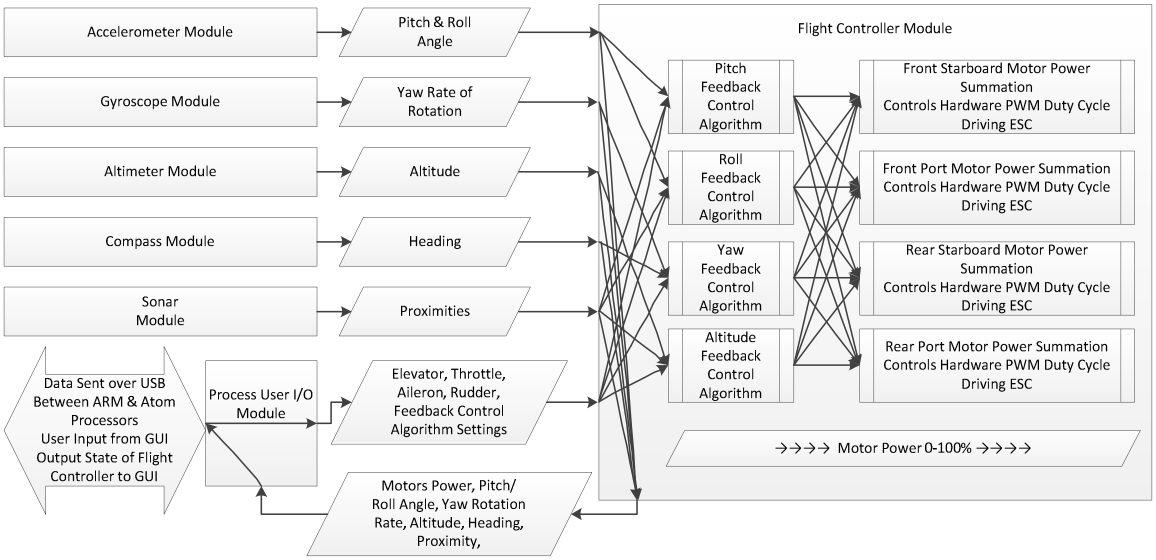


Figure 51: Flight Control Data Flow Diagram

As depicted in the diagram, the data from the accelerometer module, gyroscope module, compass module, altimeter module, sonar module, and the user input output processor module are utilized by the Flight Control module's 4 separate internal feedback control loop algorithms. Refer to section 16 below on Proportional Integral Derivative (PID) Controllers for details on the control loop algorithms.

The pitch and roll feedback control algorithms use the data from the accelerometer module to determine the aircraft's angle. The user input is used to enable the pilot to control the angle the aircraft will orientate too, so that the pilot can steer the aircraft. Data from the sonar module is also used in a similar way as the user input. It controls the angle the aircraft will orientate too, so that when objects are in close proximity to the aircraft, the algorithm can steer the aircraft away in the opposite direction. The throttle data controls the pitch and the aileron data controls the roll.

The stable angle is 0°, no pitch, no roll. The pilot's desired angle is then added. After that the angle caused by the sonars detecting objects in close proximity is added. The algorithm then calculates the required motor speeds to change the aircrafts' pitch and roll to this angle.

The yaw feedback control algorithm uses data from the gyroscope module to determine the aircraft's rate of rotation. It also uses data from the compass module to fix on a specific orientation relative to magnetic north. The user input is used to enable the pilot to control the orientation of the aircraft relative to magnetic north, so that the pilot can steer the aircraft. The rudder data controls the aircraft's yaw.

The algorithm uses the gyroscope data to control the rate of the aircraft's rotation and the compass data to give a fixed point of reference to eliminate the error that can accumulate in the gyroscope measurements. The stable rotation rate is 0°/second. The pilot's desired yaw rotation is then added. After that the rotation required to align the compass with the set orientation is added. The algorithm then calculates the required motor speeds to change the aircrafts' yaw to this rotation. When the pilot stops sending the rudder command to the aircraft the algorithm uses the current heading of the compass module as its new set point.

The altitude feedback control algorithm uses data from the altimeter module and data from the upward and downward facing sonar modules for altitude control. The user input is used to enable the pilot to orientate the aircraft at a specific altitude, so that the pilot can steer the aircraft. The elevator data controls the aircraft's altitude. Data from the upward and downward facing sonar module is also used in a similar way to the user input. It controls the elevator so that when the aircraft is near the ground or an object above is in close proximity, the algorithm can steer the aircraft away in the opposite direction. The algorithm then calculates the required motor speeds to change the aircrafts' altitude. When the pilot stops sending the elevator command to the aircraft the algorithm uses the current altitude of altimeter module as its new set point.

11. User Input Output Processor

The User Input Output Processor module reads input and writes output on the virtual serial port over the USB. This input is used to control the embedded software and flight of the aircraft. The pilot interfaces with the GUI elevator, throttle, aileron, rudder, and camera tilt controls to fly the aircraft. The application software sends messages over USB to the embedded software based on the changes the pilot makes to the GUI. The embedded system sends messages over USB to the application software to display the state of the flight controller module and power system. The messages are used to update the GUI with the current measurements from the accelerometer, gyroscope, altimeter, compass, sonar, motor power, and battery power.

12. Moving Average Filter

A Moving Average Finite Impulse Response (FIR) filter is used to filter out noise in the pressure sensor data. A FIR filter is a type of filter that contains no feedback so that its response to any finite length input is of finite length. It uses the mathematical process of convolution to remove unwanted parts of a signal. Below is a block diagram of a discrete time FIR filter of order N. The top portion is an N stage delay line with n+1 taps. Each delay is represented by Z^{-1} in Z-transform notation. The input is the measurement from the sensor and the output is a weighted sum of the current measurement and n previous values of the measurement. The operation can be described with the following equation relating the output signal y in terms of the input signal x.

$$y[n] = b_0X[n] + b_1X[n-1] + \cdots + b_NX[n-N] = \sum_{i=0}^{N} b_iX[n-i]$$

Equation 8: FIR Input Output Relation

Where b_i are the filter coefficients, known as the tap weights, and n is the order of the filter.

The FIR filter is implemented as a moving average. The coefficients are $b_i = \frac{1}{N+1}$.

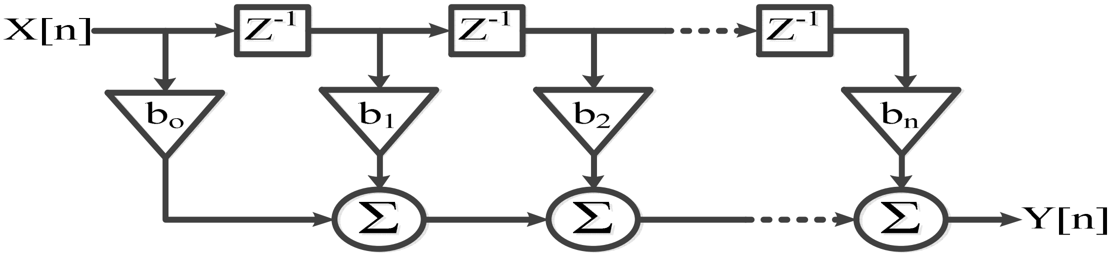


Figure 52: Finite Impulse Response (FIR) Filter

13. Proportional Integral Derivative (PID) Controllers

The 4 feedback control loop algorithms used in the flight controller for pitch, roll, yaw, and altitude are PID controllers. PID controllers utilize a feedback loop based on an error that is the difference between a measured process variable and a desired set point. The error is used to control the process in a way that reduces the error. The algorithm uses three parameters, the proportional, the integral, and the derivative values, referred to as P, I, and D. P is the value of the present error, I is the accumulation of past errors, and D is the current rate of change in error. These three parameters are multiplied by weights and summed to generate the process control signal. The following 4 diagrams illustrate how the 4 PID controllers used function.

The diagram in figure 53 illustrates how the pitch PID controller is setup. The controller uses the accelerometer's Y axis measurement as the process variable to calculate the error of the pitch angle relative to the desired set point of 0°. The user input from the throttle control plus the front and rear sonar inputs are added to the set point to control the forward and reverse directions of flight.

The diagram in figure 54 illustrates how the roll PID controller is set up. The controller uses the accelerometers X axis measurement as the process variable to calculate the error of the roll angle relative to the desired set point of 0°. The user input from the aileron control plus the port and starboard sonar inputs are added to a set point to control the side to side directions of flight.

The diagram in figure 55 illustrates how the yaw PID controller is set up. The controller uses the gyroscope's Z axis measurement as the process variable to calculate the error of the Yaw rotation rate relative to the desired set point of 0°/second. The user input from the rudder control plus the compass heading minus the desired heading are added to the set point to control the heading of flight.

The diagram in figure 56 illustrates the how the altitude PID controller is setup. The controller uses the altimeter's altitude measurement as the process variable to calculate the error of the altitude relative to the desired altitude. The user input from the elevator control plus the downward and upward facing sonar inputs are added to the set point to control the vertical direction of flight.

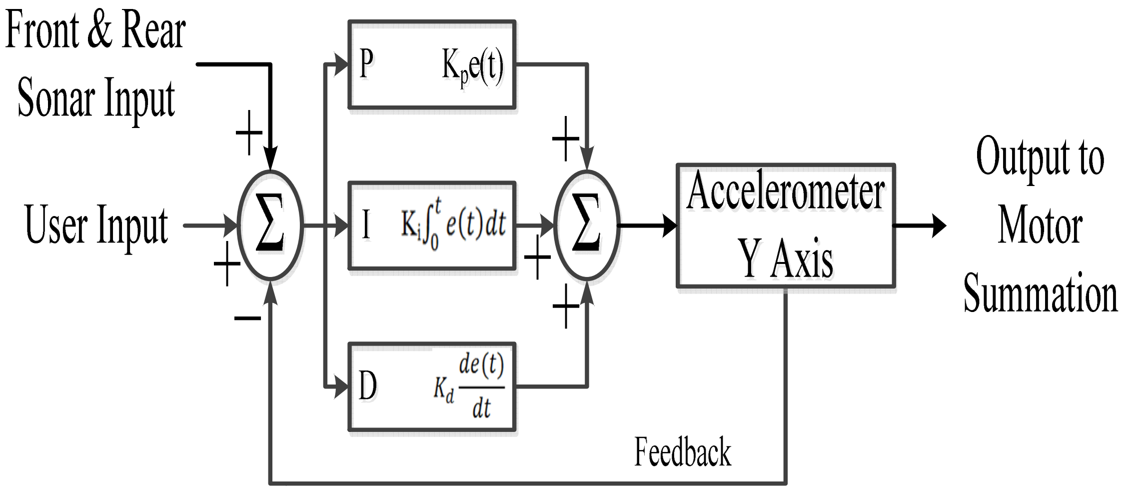


Figure 53: PID Pitch Controller

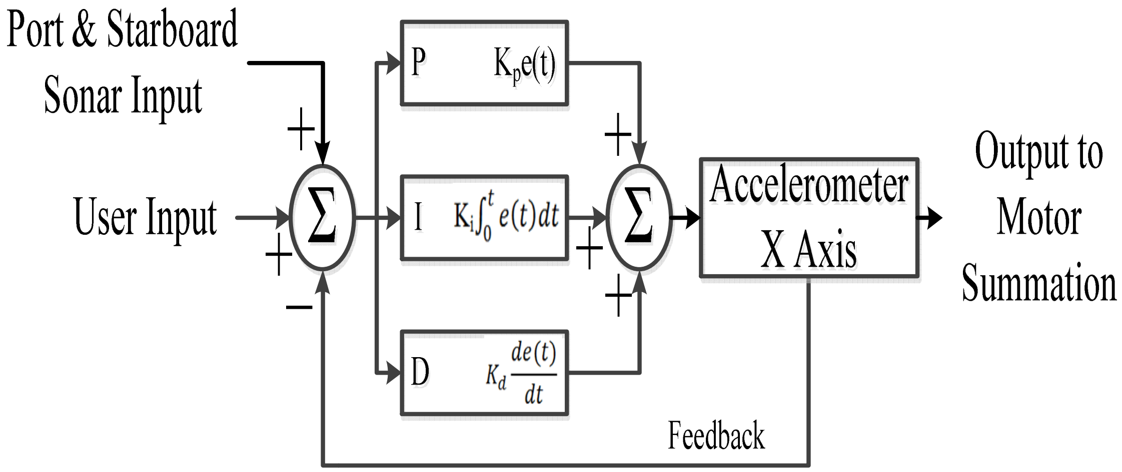


Figure 54: PID Roll Controller

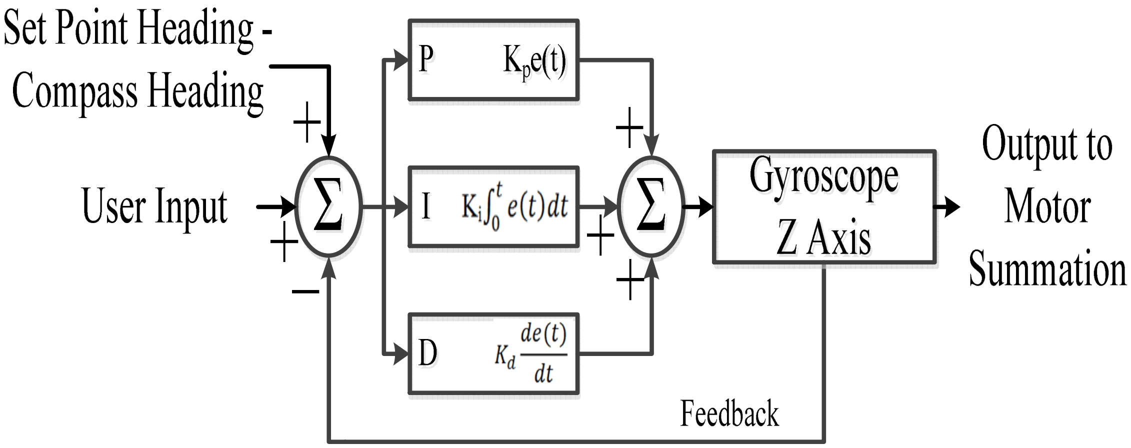


Figure 55: PID Yaw Controller

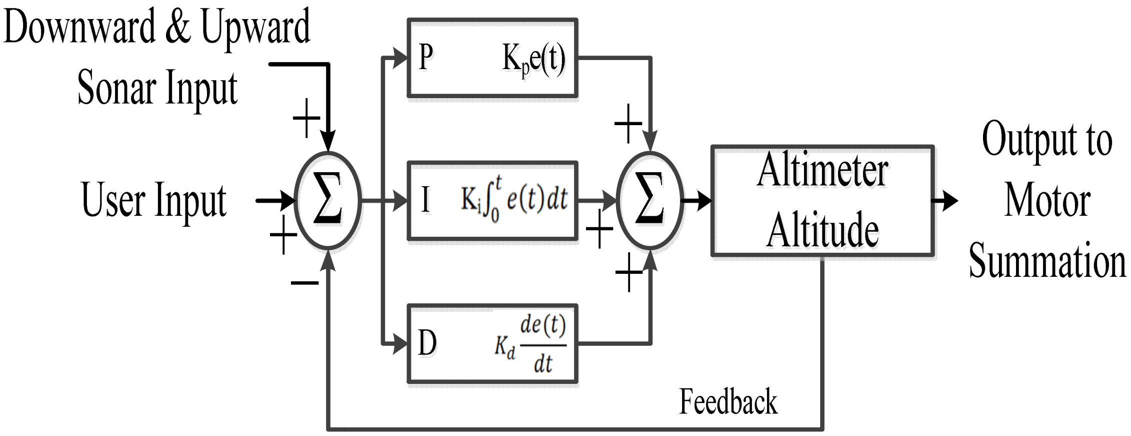


Figure 56: PID Altitude Controller

C. MCU Utilization

This table and chart are based on measurements made on an oscilloscope showing how much of the overall processing power is required to execute specific code functionality.

Code Functionality	CPU % Utilized	CPU Time / Execution	Frequency / Execution
Accelerometer	11.2	0.00028	400
Gyroscope	27.2	0.00068	400
Compass	0.371	0.000742	5
Altimeter	0.04	0.0004	1
Flight Controller	2.4	0.00006	400
Collision Avoidance	0.912	0.00152	6
User IO Processor	0.42	0.00042	10
Power Monitor	0.01	0.00005	2
% MCU Time Idle	57.447		
% MCU Time Utilized	42.553		

Table 9: Task MCU Processing Time Utilization

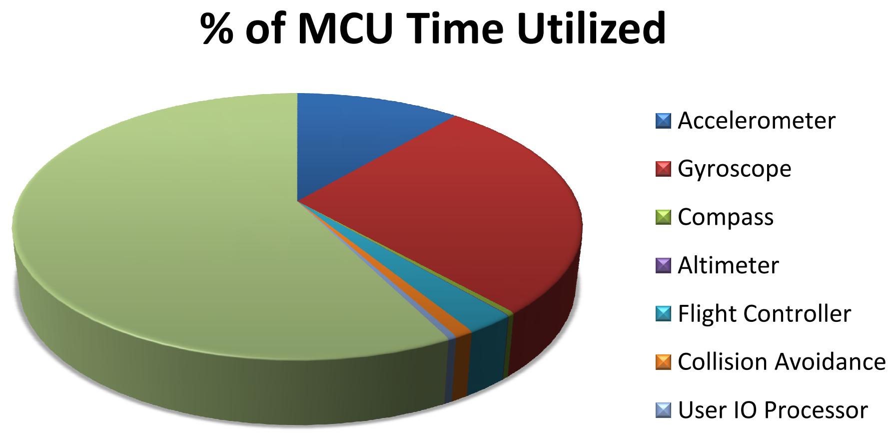


Figure 57: MCU Processing Time Utilization

D. Application Software Component Diagram

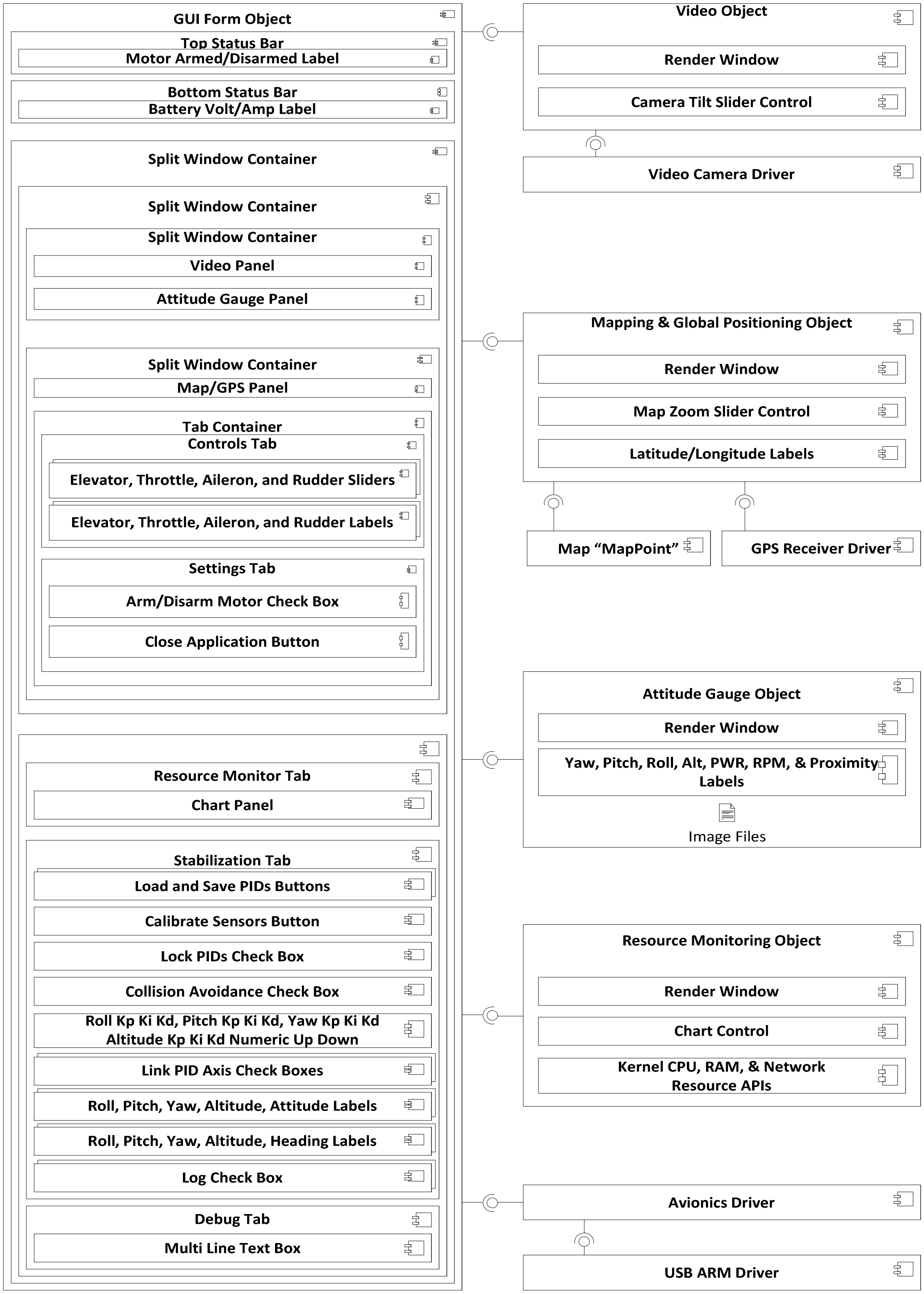


Figure 58: Application Software Component Diagram

E. Operating System (OS)

The OS chosen for the aircraft is windows 8. The windows 8 operating system enables the C# .Net code to be executed on the atom processor. C# .net was chosen because of prior experience utilizing it for GUI development, the wide range of commercially available software components, and its support of a wide range of plug and play devices. The OS also supports Remote Desktop Protocol (RDP).

F. Remote Desktop Protocol (RDP)

RDP is a Microsoft proprietary protocol for providing a remote networked computer with a graphical interface to another computer. RDP client software exists for most modern operating systems including, Windows, Linux, UNIX, Mac OS X, and Android. RDP can use 128-bit encryption to encrypt most data transmissions in both the client-to-sever and server-to-client directions. Windows 8 supports RemoteFX, a technology that is designed to improve the visual experience of RDP. RemoteFX is capable of preserving a high-fidelity video, audio, and text experience.

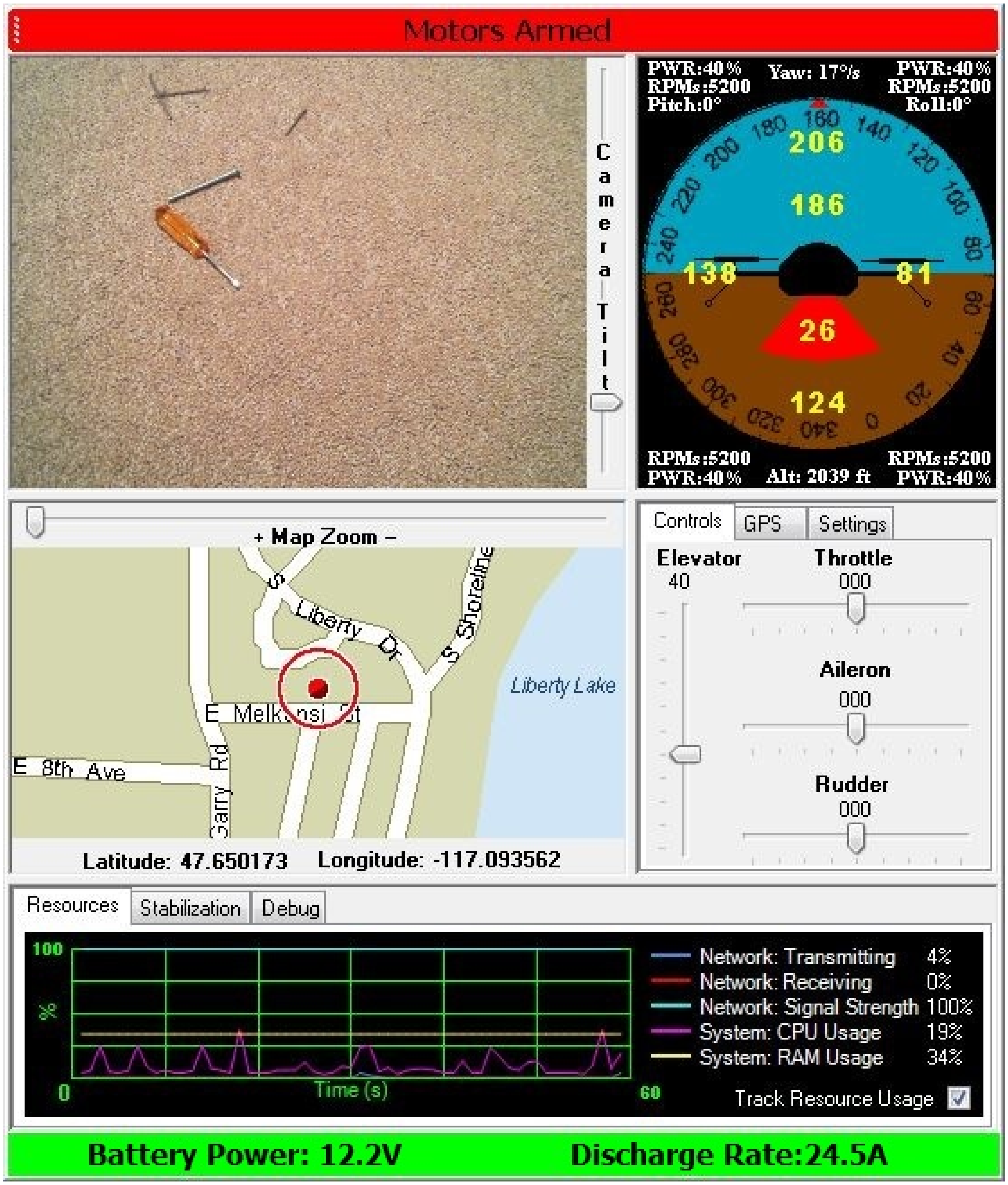
This aircraft utilizes RDP to allow it to be remotely controlled from a wirelessly networked computer running a modern OS with RDP client software installed. By utilizing RDP transmitted data, video, and control signals can be encrypted.

G. Application Software Objects

The C# .Net application software architecture is composed of several objects. Each object is designed to control specific functionality of the system.

1. Graphic User Interface (GUI)

The GUI is designed as a Form Object that gives a structure to the graphics and also manages the interaction of the other objects.



Motors Armed

PWR:40%
RPMs:5200
Pitch:0°

Yaw: 17°/s

PWR:40%
RPMs:5200
Roll:0°

RPMs:5200
PWR:40%

Alt: 2039 ft

RPMs:5200
PWR:40%

Latitude: 47.650173 Longitude: -117.093562

Elevator 40

Throttle 000

Aileron 000

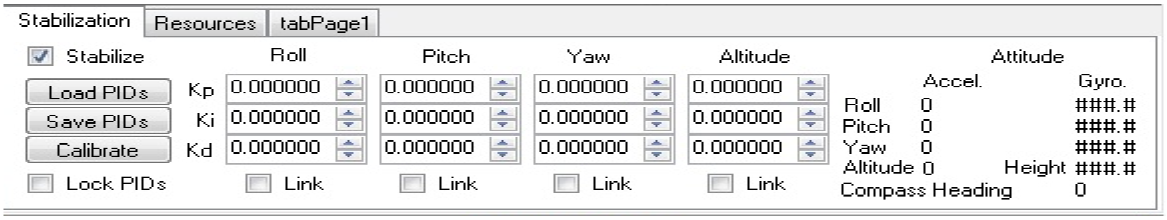
Rudder 000

Network: Transmitting 4%
Network: Receiving 0%
Network: Signal Strength 100%
System: CPU Usage 19%
System: RAM Usage 34%

Track Resource Usage ☑

Battery Power: 12.2V Discharge Rate:24.5A

Figure 59: Screen Shot of User Interface



☑ Stabilize		Roll	Pitch	Yaw	Altitude
Load PIDs	Kp	0.000000	0.000000	0.000000	0.000000
Save PIDs	Ki	0.000000	0.000000	0.000000	0.000000
Calibrate	Kd	0.000000	0.000000	0.000000	0.000000
☐ Lock PIDs		☐ Link	☐ Link	☐ Link	☐ Link

Attitude

	Accel.		Gyro.
Roll	0		###.#
Pitch	0		###.#
Yaw	0		###.#
Altitude	0	Height	###.#

Compass Heading 0

Figure 60: PID Setting

2. Video

The Video Object encapsulates the functionality of interfacing with the camera and displaying the captured images in the GUI. This object encapsulates Windows' API calls to the Media Control Interface (MCI) to control the camera.

3. Map and GPS

The Map Object encapsulates the functionality of interfacing with the GPS receiver and displaying the world map and aircraft's current position. This object encapsulates calls made to the MapPoint API.

4. Attitude Gauge

The Attitude Gauge Object encapsulates the functionality of displaying the aircraft's yaw, pitch, roll, altitude, and sonar measurements graphically on a single gauge. This object encapsulates image loading and rendering code to display the gauge.

5. Avionics Driver

The Avionics Driver Object encapsulates the functionality to interfaces with the ARM processors and embedded software. This object encapsulates calls made to the USB virtual serial port and the commands sent to interface with the avionics.

6. Resource Monitor

The Resource Monitor Object encapsulates the functionality of tracking the current usage of resources (Network, CPU, RAM). This object encapsulates Win32 API calls.

XIII. Software Testing

Software testing was conducted at every stage of the project. Testing started with the design of the subsystem units, followed by the integration of subsystem units, and the fully assembled system.

A. Unit Testing

The unit testing was conducted on each subcomponent unit of software to verify that the unit performed as expected.

1. GUI Testing

The GUI was the initial software unit designed and the first section to begin testing. C# code was written and tested to verify that a GUI could be displayed to the user. Initial testing utilized a video display monitor connected to the SBC's DVI interface. A blank Windows .Net Form was displayed on the monitor.

2. Wireless Remote Control Testing

Wireless remote control of the GUI was the next unit to begin testing. Microsoft's RDP was utilized to implement this functionality. The initial testing of this unit involved connecting a laptop computer to the SBC wirelessly over 802.11 and verifying that the GUI running on the SBC could be viewed and controlled from the laptop.

3. Video Testing

Video capture testing then followed. A C# application was developed to interface with a camera and display video. This unit of code was tested to verify that video images could be captured and displayed in real time with a high resolution and frame rate. The unit was also tested to work with several different cameras. The initial test was executed on a laptop computer before testing on the SBC.

4. GPS Testing

GPS testing was conducted on C# code that was developed utilizing Microsoft's MapPoint API. The testing was conducted to verify that the GPS receiver could be communicated with and that the data received could be plotted on a map of the world. The unit was tested to verify that the plotted position would be updated while moving on a vehicle. The initial test was executed on a laptop computer before testing on the SBC.

5. Resource Monitoring Testing

A resource monitoring unit was designed and tested. The unit was tested to verify that it could plot the utilization percentage of the CPU, RAM, and wireless network as well as the signal strength of the wireless connection. The initial test was executed on a laptop computer before testing on the SBC.

6. Microcontroller Testing

The MCU was the first part of the embedded system to be tested. This unit was tested to verify that the MCU could be programmed and that the program could be executed. This involved C++ code written to flash an LED on the embedded system board.

7. MCU and SBC USB Testing

The USB communications between the SBC and the MCU were tested next. C++ code was written to enable the MCU to send and receive data over a USB connection. At the same time C# code was written to enable the SBC to send and receive data over a USB connection. The 2 units were tested in parallel to verify that they could both send and receive data.

8. Embedded Flash File I/O Testing

The embedded Flash file I/O unit was then tested. C++ code was written to create a text file, then read and write data to it. The file was then opened on a laptop computer using Microsoft's Notepad text editor to verify that the file was written to properly.

9. Accelerometer Testing

The accelerometer was the first sensor of the embedded system to be tested. C++ code was written to communicate with the sensor over I^2C. The test code first requested the ID of the component to verify that the MCU was able to communicate with the accelerometer. Then the code requested measurements from the sensor. A simple GUI application was developed to display the measurements. The accelerometer was then tilted to various known angles and the measurements were compared to verify that the circuit was functioning properly.

10. Gyroscope Testing

The gyroscope was the next sensor of the embedded system to be tested. C++ code was written to communicate with the sensor over I^2C. The test code first requested the ID of the component to verify that the MCU was able to communicate with the gyroscope. Then the code requested measurements from the sensor. A simple GUI application was developed to display the measurements. The gyroscope was then rotated at various known angles and the measurements were compared to verify that the circuit was functioning properly.

11. Compass Testing

Then the compass sensor of the embedded system was tested. C++ code was written to communicate with the sensor over I^2C. The test code first requested the ID of the component to verify that the MCU was able to communicate with the compass. Then the code requested measurements from the sensor. A simple GUI application was developed to display the measurements. The compass was directed to known headings and the measurements were compared to verify that the circuit was functioning properly.

12. Pressure Sensor Testing

Then the pressure sensor of the embedded system was tested. C++ code was written to communicate with the sensor over I^2C. The test code first requested the ID of the component to verify that the MCU was able to communicate with the pressure sensor. Then the code requested measurements from the sensor. A simple GUI application was developed to display the measurements. The pressure sensor was placed at a known altitude and the measurements were compared to verify that the circuit was functioning properly.

13. Sonar Testing

Then the sonar sensors of the embedded system were tested. C++ code was written to communicate with the sensor over I^2C. Code was written to update the EEPROM I^2C address of each sonar module. The test code first requested the ID of the component to verify that the MCU was able to communicate with the sonar sensors. Then the code requested measurements from the sensors. A simple GUI application was developed to display the measurements. Objects were placed at known distances in front of the sonar modules and the measurements were compared to verify that the circuits were functioning properly.

B. System Testing

After all the software was developed for the subcomponents, the integration and system testing began. Subcomponent by subcomponent was integrated together and tested to verify that all functionality remained as expected. Once all components had been integrated together the system testing began. With all software, electrical and mechanical systems integrated and functioning properly, flight control tuning and testing could begin.

XIV. Flight Control Tuning & Testing

The fully assembled aircraft is shown below. At this point further testing and tuning must be done to verify that the entire system is able to function. Specifically the PID controller must be tested and tuned.

Figure 61: Fully Assembled Aircraft

A. Testing Rig

In order to tune the control loop variables the system was placed in a controlled environment. A testing rig was designed and built to fix the aircrafts rotation to a single axis. By only allowing the aircraft to rotate around a single axis at a time the PID controller can be tuned one axis at a time. Shown below is an image of the test rig.

The rig utilizes a metal rod on to which the aircraft mounts and rotates around, thus fixing it to a single axis of rotation for tuning pitch and roll. The rig utilizes a cable which the aircraft mounts and rotates around, thus fixing it to a single axis of rotation for tuning yaw. The rig was designed in Alibre and Manufacture by Ponoko using Computer Numerical Control (CNC).

Figure 62: Tethered Rig

B. Tethered Testing

The tethered test involved applying a rotational force to the aircraft around the free axis of rotation and allowing it to self-stabilize. The force was applied by pushing down on one side of the aircraft and then releasing. The PID gain control variables were tuned to attain the desires response. Both pitch and roll axes were tuned one at a time in this manner.

Then the aircraft was suspended with a cable above and below it to only allow it to rotate in the yaw direction. Due to the torque generated by the different rates of spinning propellers the aircraft will spin around in the yaw direction. The PID gain control variables for the yaw axis were tuned to equalize the clockwise and counter clockwise torques and eliminate the aircrafts rotational spin.

Figure 63: Pitch & Roll PID Tuning

Figure 64: Yaw PID Tuning

C. Flight Test

Once the tethered testing had been completed and the PID controls were properly tuned for stability, it was time for the first flight test to be conducted. The testing was conducted indoors to ensure that weather conditions would not impact the testing.

The hover stability flight test was run under human control. The user moved the elevator control to increase the thrust of the propellers and the aircraft was allowed to rise into the air a foot, then hover for several seconds, and finally land.

Figure 65: Hover Test

XV. Budget & Build of Materials (BOM)

Funding for this project was entirely financed by the student out of pocket. The maximum funding allocated to purchase new components for the project was $1000.00. Most of the components used to build this project had to be purchased new. Many of the components used in the project were already owned and therefore did not need to be purchased. Most of the components were ordered from overseas vendors. Tax and shipping cost were not included in the components prices. The Tax and shipping cost added an additional amount of approximately $100.00.

The following table documents the build of materials (BOM). The BOM includes a description of the part, whether or not it was purchased new or previously owned, the quantity of the part, and the total cost of the parts.

ID	Part Description	Notes	Qty	Total $
1	*PITX-SP Single Board Computer*	*Already Owned*	*1*	*373.00*
2	*MBED Microcontroller Development Board*	*Already Owned*	*1*	*59.95*
3	*HMC6343 Compass*	*Purchased New*	*1*	*143.95*
4	*ADXL345 Accelerometer*	*Purchased New*	*1*	*14.95*
5	*ITG-3200 Gyroscope*	*Purchased New*	*1*	*24.95*
6	*BMP-085 Pressure Sensor*	*Purchased New*	*1*	*8.95*
7	*SRF02 Sonar Range Finder*	*Purchased New*	*6*	*145.32*
8	*Plush Electronic Speed Controller*	*Purchased New*	*4*	*49.92*
9	*BLDC Motors*	*Purchased New*	*4*	*35.44*
10	*Lithium Polymer Battery 3S 6.0Ah*	*Purchased New*	*1*	*38.59*
11	*5.0V Regulator*	*Purchased New*	*1*	*24.95*
12	*Variable High Voltage Regulator*	*Purchased New*	*1*	*20.00*
13	*BTS555 FET Power Switch*	*Purchased New*	*1*	*9.97*
14	*Battery Charger*	*Already Owned*	*1*	*39.99*
15	*Carbon Fiber Tubes*	*Purchased New*	*1*	*16.00*
16	*Plastic Parts*	*Purchased New*	*1*	*250.00*
17	*Web Camera*	*Already Owned*	*1*	*49.99*
18	*Servo*	*Purchased New*	*1*	*2.67*
19	*GPS Module*	*Purchased New*	*1*	*30.00*
20	*Wi-Fi Dongle*	*Already Owned*	*1*	*20.00*
21	*Flash Drive*	*Already Owned*	*1*	*60.00*
22	*RAM Module*	*Already Owned*	*1*	*32.52*
23	*Microsoft Windows Operating System*	*Already Owned*	*1*	*120.00*
24	*Microsoft Map Point*	*Already Owned*	*1*	*290.00*
25	*Printed Circuit Board*	*Purchased New*	*1*	*66.00*
26	*Training Stand*	Purchased New	*1*	70.00
27	*Steel Rod*	Purchased New	*1*	5.00
28	*Tachometer*	Purchased New	*1*	9.99
29	*Power Analyzer*	Purchased New	*1*	23.95
	Total Spent			***$990.60***
	Total Cost to Purchase all Materials New			**$2036.05**

Table 10: Build of Materials

XVI. Conclusion

The goal of this project was to produce a micro unmanned aerial vehicle video surveillance platform. Aeronautical, mechanical, electrical, and software engineering skills were required to bring the project to completion. The total cost of the hardware required to build a new aircraft is approximately $1450.00. Currently the platform is not fully autonomous. It has the sensors and computing power required to run software algorithms to enable fully autonomous flight missions to be executed.

Beyond the aircraft's hardware and propulsion system design its ability to fly is mainly governed by means of software. The majority of the development time was devoted to the software engineering. The spiral model was utilized for software development. This methodology allowed unforeseen requirements to be integrated into the system. The strength of the spiral method was the continual testing performed at each iteration of the development process. After software prototypes for each of the subsystems' components were complete, combining them into a single software architecture went smoothly without any unexpected behaviors. The modularity of the software design enabled the code to be more manageable and easier to understand.

Improvements could be made to all aspects of this project, the mechanics, the electronics, and the software. Now that they have been tested, the electronics can be reduced in size by removing many of the debugging interfaces. With fewer and or smaller electronic circuits the airframe can also be reduced in size and weight. This would increase the overall flight time of the aircraft. More software could also be added to enable the aircraft to fly missions completely autonomously. The additional software could access the GPS for waypoint navigation and video images for object recognition. The additional software would require no new hardware to be added and should cost little beyond development time to implement.

I will continue to work on this project after completing my degree. Now that the airframe and electronics assembled are tested. I will focus on designing more advanced algorithms.

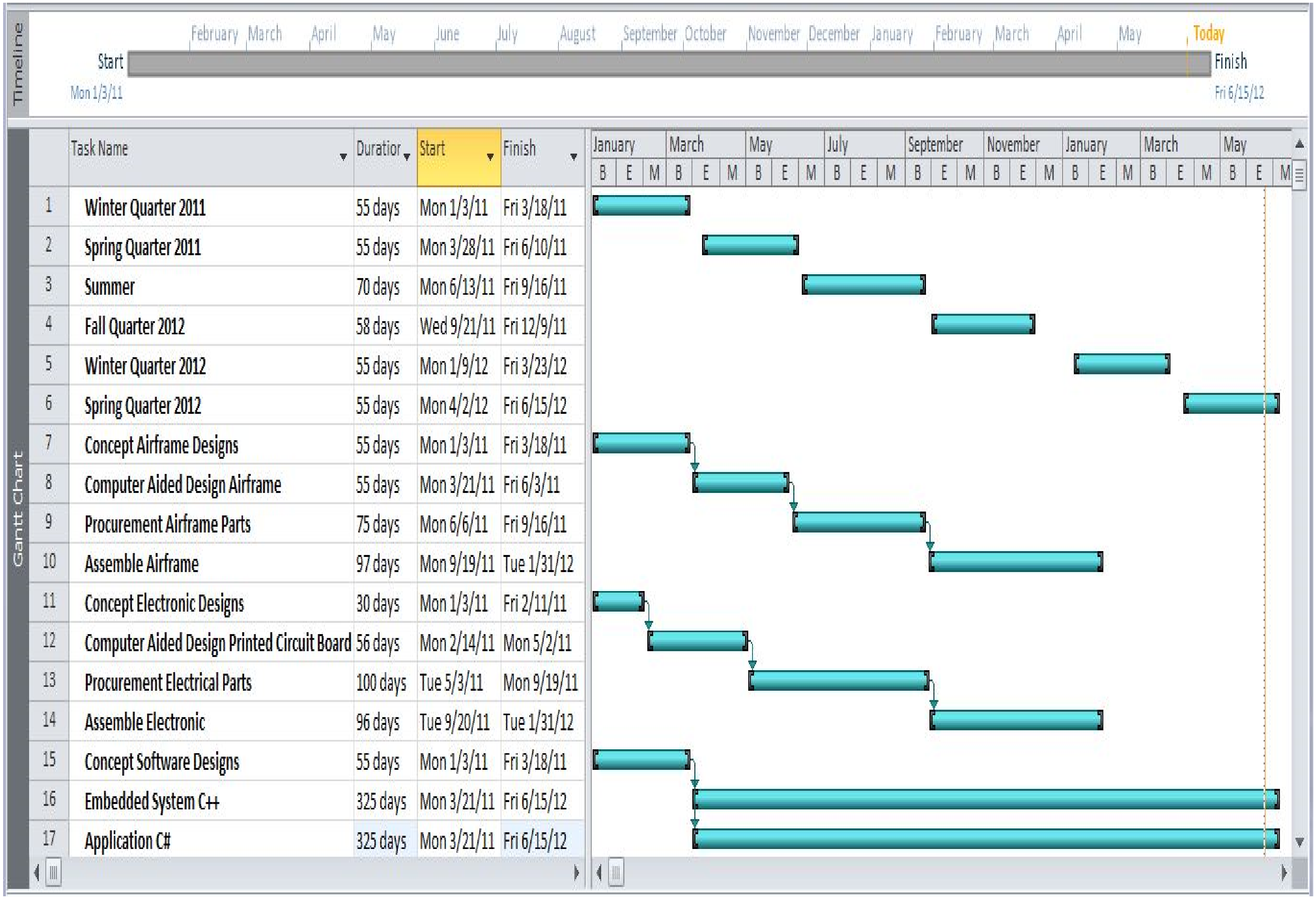
XVII. Project History

Once the topic of an unmanned aerial vehicle was chosen and approved by my advisor, work began immediately. Time was a major limiting factor, single-handedly working on such a complex project, while completing my degree full time and working part time.

The project design involved engineering the tightly coupled mechanical, electrical, and software components in parallel. Concept designs were made and from those designs one was selected to build. At that point the required components were identified and ordered. While waiting, mechanical parts were designed to accommodate the first set of parts ordered. Then the mechanical parts were ordered from the manufacture. As the parts started to arrive, they were assembled and test fixtures and code were developed to verify that the components would perform as required. Eventually the whole aircraft was assembled and operational.

The Gantt chart in figure 65 outlines the history of the project. The top of the chart breaks the projects history up into school quarters, beginning with winter quarter of 2011 and finishing with spring quarter of 2012. Below that are outlines of the project history for the mechanical, electrical, and software components.

The last quarter of my degree was dedicated entirely to software testing, control loop tuning, and flight testing. The last couple of weeks of the final quarter were dedicated to the creation of the final presentation. The documentation was finalized and the presentation was rehearsed.



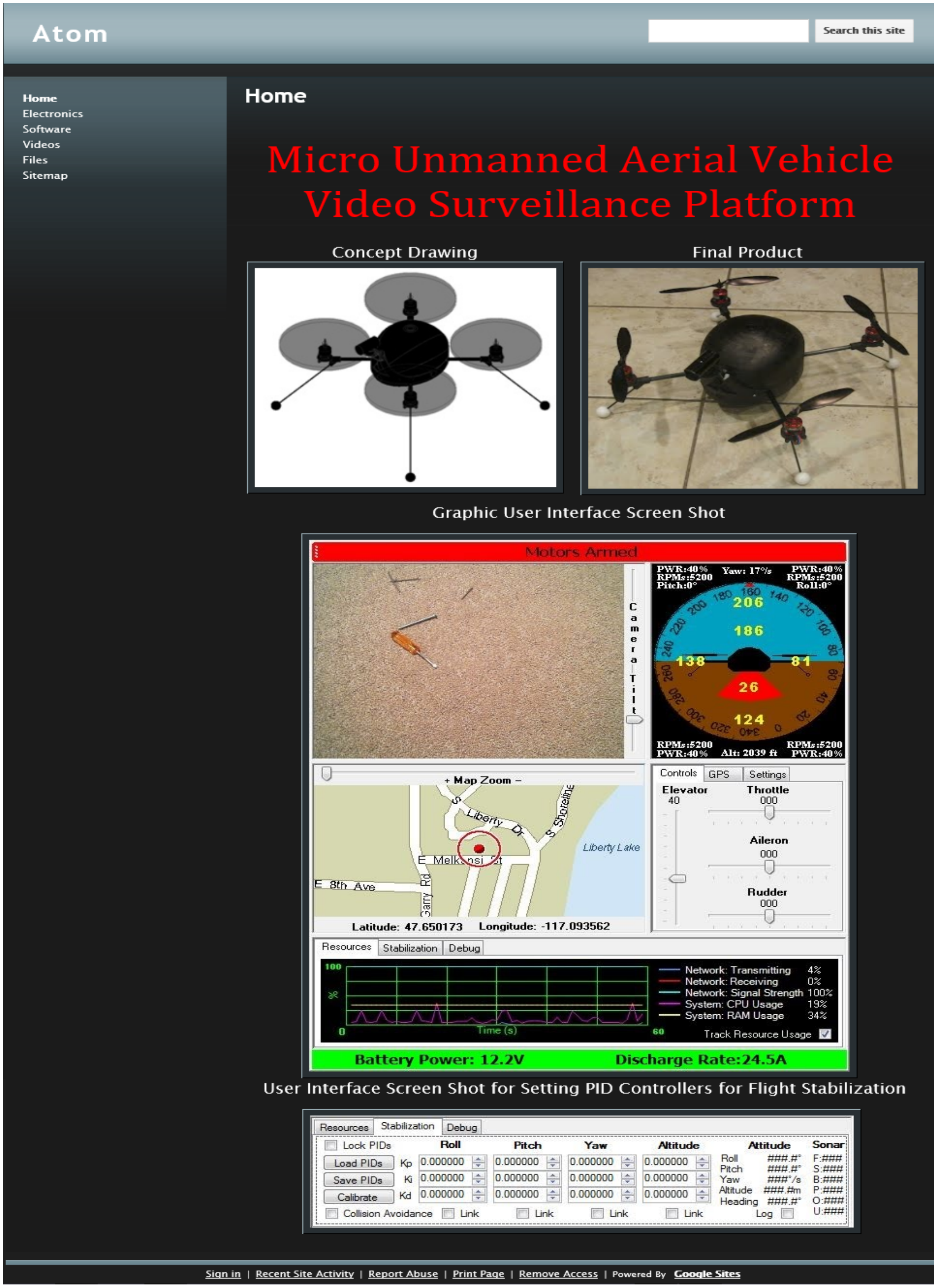
	Task Name	Duration	Start	Finish
1	Winter Quarter 2011	55 days	Mon 1/3/11	Fri 3/18/11
2	Spring Quarter 2011	55 days	Mon 3/28/11	Fri 6/10/11
3	Summer	70 days	Mon 6/13/11	Fri 9/16/11
4	Fall Quarter 2012	58 days	Wed 9/21/11	Fri 12/9/11
5	Winter Quarter 2012	55 days	Mon 1/9/12	Fri 3/23/12
6	Spring Quarter 2012	55 days	Mon 4/2/12	Fri 6/15/12
7	Concept Airframe Designs	55 days	Mon 1/3/11	Fri 3/18/11
8	Computer Aided Design Airframe	55 days	Mon 3/21/11	Fri 6/3/11
9	Procurement Airframe Parts	75 days	Mon 6/6/11	Fri 9/16/11
10	Assemble Airframe	97 days	Mon 9/19/11	Tue 1/31/12
11	Concept Electronic Designs	30 days	Mon 1/3/11	Fri 2/11/11
12	Computer Aided Design Printed Circuit Board	56 days	Mon 2/14/11	Mon 5/2/11
13	Procurement Electrical Parts	100 days	Tue 5/3/11	Mon 9/19/11
14	Assemble Electronic	96 days	Tue 9/20/11	Tue 1/31/12
15	Concept Software Designs	55 days	Mon 1/3/11	Fri 3/18/11
16	Embedded System C++	325 days	Mon 3/21/11	Fri 6/15/12
17	Application C#	325 days	Mon 3/21/11	Fri 6/15/12

Figure 66: Project Gantt Chart

XVIII. Web Site

Along with this project I also created a website to include in my resume as a way to give prospective employers a closer look at my work. I used Google to create and host the site. The site includes pictures of the aircraft, the aircraft's GUI, diagrams illustrating how the design is implemented, and videos of the aircraft. This site has enabled my resume to include much more information than in a plain text document. The home page of the website can be seen in figure 66 below. The website's address is https://sites.google.com/site/atomuav/ .

Along with the tools that Google provided to create and host this website. Google also provides tools to track and monitor statistical information about visitors to the site. The image in figure 67 shows a screenshot of some of this information. In this screenshot the location of the people visiting my website can be seen on a map. This has enabled me to identify what companies are closely reviewing my resume. It also allows me to see what they are specifically looking at in regards to my project's website.



Home

Micro Unmanned Aerial Vehicle Video Surveillance Platform

Concept Drawing

Final Product

Graphic User Interface Screen Shot

Motors Armed

PWR:40%
RPMs:5200
Pitch:0°

Yaw: 17°/s

PWR:40%
RPMs:5200
Roll:0°

RPMs:5200
PWR:40%

Alt: 2039 ft

RPMs:5200
PWR:40%

Latitude: 47.650173 Longitude: -117.093562

Elevator 40

Throttle 000

Aileron 000

Rudder 000

Network: Transmitting 4%
Network: Receiving 0%
Network: Signal Strength 100%
System: CPU Usage 19%
System: RAM Usage 34%

Track Resource Usage ☑

Battery Power: 12.2V Discharge Rate:24.5A

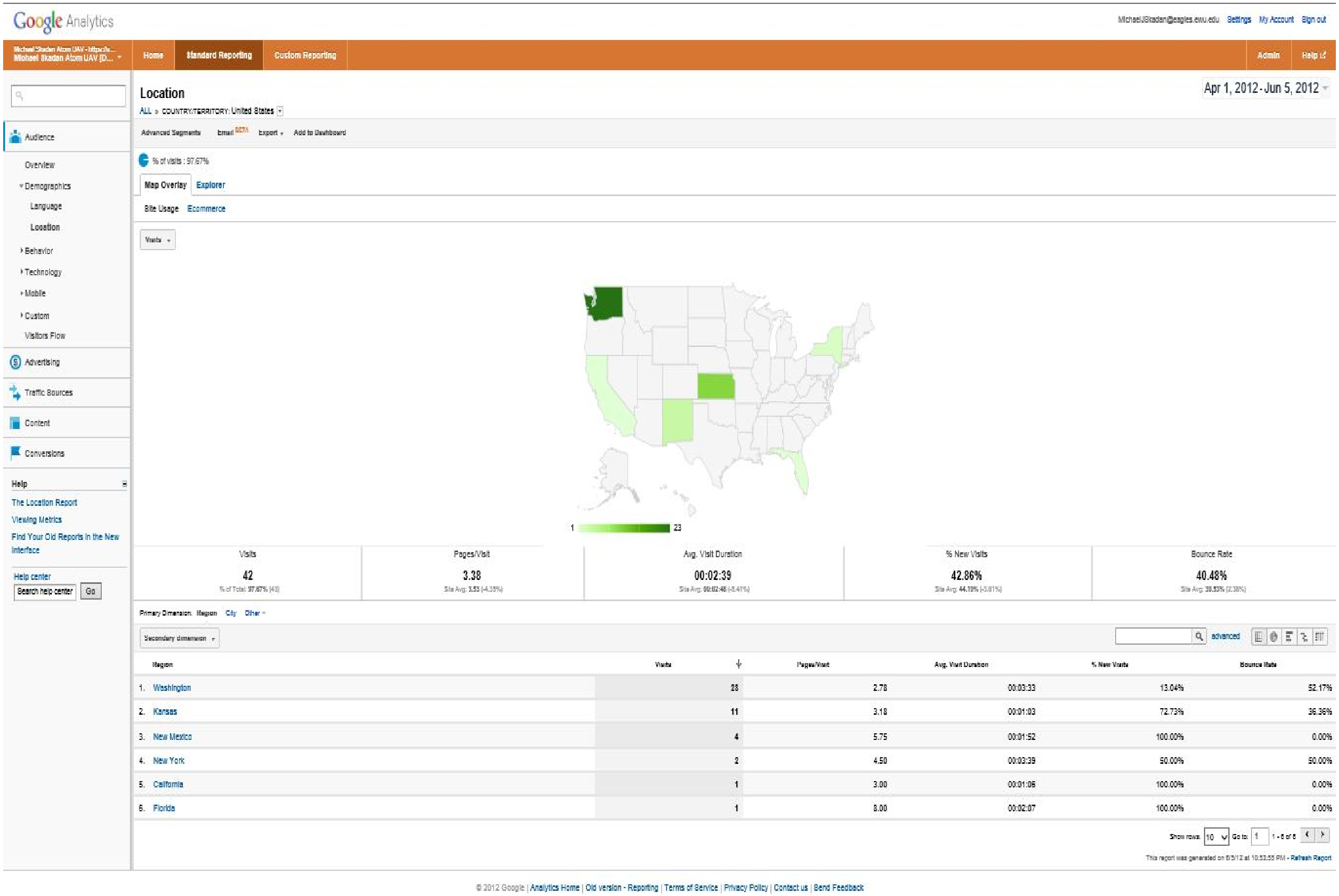
User Interface Screen Shot for Setting PID Controllers for Flight Stabilization

☐ Lock PIDs	Roll	Pitch	Yaw	Altitude
Kp	0.000000	0.000000	0.000000	0.000000
Ki	0.000000	0.000000	0.000000	0.000000
Kd	0.000000	0.000000	0.000000	0.000000
☐ Collision Avoidance	☐ Link	☐ Link	☐ Link	☐ Link

Attitude
Roll ###.#°
Pitch ###.#°
Yaw ###°/s
Altitude ###.#m
Heading ###.#°
Log ☐

Sonar
F:###
S:###
B:###
P:###
O:###
U:###

Figure 67: Web Site Home Page



Location

ALL » COUNTRY/TERRITORY: United States

Apr 1, 2012 - Jun 5, 2012

% of visits : 97.67%

Visits	Pages/Visit	Avg. Visit Duration	% New Visits	Bounce Rate
42	3.38	00:02:39	42.86%	40.48%
% of Total: 97.67% (43)	Site Avg: 3.53 (-4.35%)	Site Avg: 00:02:48 (-5.41%)	Site Avg: 44.19% (-3.01%)	Site Avg: 39.53% (2.38%)

Region	Visits	Pages/Visit	Avg. Visit Duration	% New Visits	Bounce Rate
1. Washington	23	2.78	00:03:33	13.04%	52.17%
2. Kansas	11	3.18	00:01:03	72.73%	36.36%
3. New Mexico	4	5.75	00:01:52	100.00%	0.00%
4. New York	2	4.50	00:03:39	50.00%	50.00%
5. California	1	3.00	00:01:06	100.00%	0.00%
6. Florida	1	8.00	00:02:07	100.00%	0.00%

This report was generated on 6/5/12 at 10:53:55 PM

Figure 68: Google Analytics Location

XIX. Appendix

Advanced Circuits
http://www.4pcb.com/

Alibre CAM/CAD software
http://www.alibre.com/

Active Media Products Disk on Module
http://activemp.com/DOM/IDE-PATA-DOM-Disk-on-Module.htm

Analog Devices 3 Axis Accelerometer ADXL345 Datasheet
http://www.analog.com/static/imported-files/data_sheets/ADXL345.pdf

Advanced Precision Composites Propellers
http://www.apcprop.com/v/index.html

Bosch Pressure Sensor BMP085 Datasheet
http://www.bosch-sensortec.com/content/language1/downloads/BST-BMP085-DS000-05.pdf

Buffalo USB Wi-Fi Transceiver
http://www.buffalotech.com/products/wireless/client-adapters/airstation-n150-ultra-compact-usb-20-wireless-adapter-wli-uc-gnm/

Cyber Technology Quadrocopter UAV
http://www.cybertechuav.com.au/-Overview,85-.html

Devantech SRF02 Sonar Module
http://www.robot-electronics.co.uk/htm/srf02tech.htm

DIPTrace Printer Circuit Board Design Software
http://www.diptrace.com/

Dragonfly Innovation Inc Quadrocopter UAV
http://www.draganfly.com/

GlobalSat ND-100s GPS Receiver
http://www.usglobalsat.com/p-590-nd-100s.aspx#/images/product/large/590.jpg

HobbyKing Servo
http://www.hobbyking.com/hobbyking/store/__11856__HobbyKing_929MG_Metal_Gear_Servo_2_2kg_12_5g_0_10sec.html

Honeywell Tilt Compensated Compass HMC6343 Datasheet

http://www51.honeywell.com/aero/common/documents/myaerospacecatalog-documents/Missiles-Munitions/HMC6343.pdf

Invensense 3 Axis Gyroscope ITG-3200 Datasheet
http://invensense.com/mems/gyro/documents/PS-ITG-3200-00-01.4.pdf

Kontron PITX-SP Single Board Computer
http://us.kontron.com/products/boards+and+mezzanines/embedded+sbc/pitx+25+sbc/pitxsp.html

MBED ARM Development Board
http://mbed.org/

Microdrones GmbH Quadrocopter UAV
http://www.microdrones.com/index.php

Microsoft LifeCam Web Camera
http://www.microsoft.com/hardware/en-us/webcams

Microsoft MapPoint
http://www.microsoft.com/mappoint/en-us/home.aspx

Microsoft Remote Desktop Protocol Encryption
http://support.microsoft.com/kb/275727

Microsoft Visual Studios
http://www.microsoft.com/visualstudio/en-us

Parrot AR Drone Quadrocopter UAV
http://ardrone.parrot.com/parrot-ar-drone/usa/

Pololu Voltage Regulator
http://www.pololu.com/catalog/product/2111

Ponoko (Selective Laser Sintering & Computer Numerical Control Manufacturer)
http://www.ponoko.com/

Turnigy L2210 Motor
http://www.hobbyking.com/hobbyking/store/uh_viewItem.asp?idProduct=18545

Turnigy Nano-Tech Lipo 6.0Ah 25-50C
http://www.hobbyking.com/hobbyking/store/__20681__Turnigy_nano_tech_6000mah_3S_25_50C_Lipo_Pack_USA_Warehouse_.html

Turnigy Plush 25A Electronic Speed Controller http://www.hobbyking.com/hobbyking/store/__2163__TURNIGY_Plush_25amp_Speed_Controller.html

Turnigy Tachometer
http://www.hobbyking.com/hobbyking/store/uh_viewItem.asp?idProduct=10783

Turnigy Watt Meter and Precision Power Analyzer
http://www.hobbyking.com/hobbyking/store/uh_viewItem.asp?idProduct=10080

Using an Accelerometer for Inclination Sensing Analog Devices,
Application Note, AN-1057
http://www.analog.com/static/imported-files/application_notes/AN-1057.pdf

Windows 8
http://windows.microsoft.com/en-US/windows-8/release-preview

XX. Vita

Author: Michael J. Skadan

Place of Birth: Albuquerque, New Mexico

Undergraduate Schools Attended: DeVry University

Degrees Awarded: Bachelor of Science, 2007, DeVry University

Honors and Awards: Graduated Magna Cum Laude, DeVry University, 2007

Professional Experience: Hardware Design Engineer, Xbox Console, Redmond, WA 2007-2010