Aaron Derstine

Zachary Wontrop

Thomas Kuczun

October 21, 2013

System Design

SEED Quadrotor

ELEE 401

Grove City College

Table of Contents

[Abstract 3](#_Toc370134316)

[List of Figures 3](#_Toc370134317)

[List of Tables 3](#_Toc370134318)

[Introduction 3](#_Toc370134319)

[Overall Design 3](#_Toc370134320)

[System Component Decisions 4](#_Toc370134321)

[Frame 4](#_Toc370134322)

[Motors 5](#_Toc370134323)

[Microcontroller 5](#_Toc370134324)

[IMU 5](#_Toc370134325)

[Batteries 5](#_Toc370134326)

[Power Distribution Board 5](#_Toc370134327)

[ESC’s 5](#_Toc370134328)

[Sub Components 6](#_Toc370134329)

[Power Systems 6](#_Toc370134330)

[Battery 6](#_Toc370134331)

[Power Board 6](#_Toc370134332)

[Control System 7](#_Toc370134333)

[IMU 7](#_Toc370134334)

[Ultrasonic Sensors 8](#_Toc370134335)

[Motors and ESC’s 9](#_Toc370134336)

[Wireless Communication 11](#_Toc370134337)

[System Testing 11](#_Toc370134338)

[Budget 11](#_Toc370134339)

[Schedule 13](#_Toc370134340)

[Task Descriptions 14](#_Toc370134341)

[Conclusion 15](#_Toc370134342)

[References 15](#_Toc370134343)

[Individual Contributions 15](#_Toc370134344)

[Aaron 15](#_Toc370134345)

[Zach 15](#_Toc370134346)

[Tom 16](#_Toc370134347)

# Abstract

This document outlines the system design for our SEED quadrotor project. It contains numerous diagrams of the various sub-components of the system and a description of the choices we made in picking the hardware. It also contains the preliminary schedule, budget, and testing procedures.

# List of Figures

[Figure 1: High Level Block Diagram of System 4](#_Toc370139805)

[Figure 2: Power Distribution System 6](#_Toc370139806)

[Figure 3: High Level Block Diagram of IMU Control System 7](#_Toc370139807)

[Figure 4: IMU Control Loop 8](#_Toc370139808)

[Figure 5: Sensor Control Loop 9](#_Toc370139809)

[Figure 6: Motor and ESC connection to Arduino 10](#_Toc370139810)

[Figure 7: Motor Control Connection Board 10](#_Toc370139811)

[Figure 8: Gantt Chart Showing Preliminary Schedule of Tasks 13](#_Toc370139812)

# List of Tables

[Table 1: List of Expendatures 12](#_Toc370139822)

[Table 2: Description of Tasks 14](#_Toc370139823)

# Introduction

This is the third year that a SEED project has attempted to build a quadrotor. Our strategy is to use some of the components of the previous groups’ projects but to purchase a commercially fabricated frame which we believe will make our project more easily successful. We have spent time researching flight algorithms and reading various blogs and reports on other successful quadrotor projects and assembled a materials list that will allow us to make an autonomous quadrotor capable of sustained, stable flight. We have scheduled adequate time to design, create, and debug each component in an effort to achieve our primary goal of stable hovering at a fixed altitude by the project completion date. This document contains a detailed description of the necessary components to realize this goal.

# Overall Design

Below is the block diagram of how all the systems relate to each other. As can be seen from the diagram the power system provides power for the controller and the propulsion system. The controller gives the propulsion system commands. The controller requests positional values from the IMU and the IMU returns the positional values as it reads them. Finally these three systems are secured to the frame.

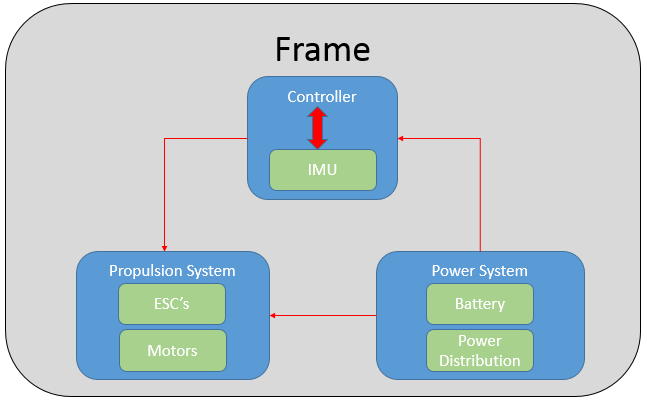


Figure : High Level Block Diagram of System

# System Component Decisions

### Frame

The original frame was designed in-house by the mechanical engineers and aftertime it broke and wasn’t very well balanced. Another problem with the frame involved vibrations. Because the old frame was skeletal, the spinning of the motors produced violent vibrations throughout the frame. This really affected the IMU, because its readings were dramatically being affected by these resonant vibrations. The old frame also did not have much space to mount electronics onto, and almost every electronic device was zip-tied to the frame in some manner. This made things even more unbalanced and created more snags than we were willing to cope with. We compared several different designs by analyzing these aspects. Size, weight, durability, and complexity. It was decided that the SK450 Glass Fiber frame was the lightest and sturdiest for the size that we wanted. It is made of ultra-durable polyamide nylon, and after some research we decided that this is an ideal material. It is slightly bendable, but still very stiff, so the frame should absorb most of the vibrations that were resonating through the old frame. This design also had twice as much space in the center plates for attaching the microcontroller and battery so all of our electronics will be attached securely. In addition to that the arms were flat rather than rectangular which provided more space for wiring and attaching the ESC’s. Another frame that we considered was the 4-Axis HJ600 Plastic Multi-Copter Quadcopter. While this frame was lighter at under 200g it did not have enough areas to mount our boards.

### Motors

This was a simple design decision because we are simply using the motors that the team two years ago used for their quadrotor. We knew that it was unnecessary to do our own research because of the depth of research done by that team. In addition to this several professors told advised us to use those motors.

### Microcontroller

We looked into a few different options for the microcontroller, the two main options being an Arduino and the Raspberry Pi. Doing research and looking into other people’s quadrotors on the internet we found that using the Raspberry Pi has been done but was a bit more difficult since it runs Linux which is not strictly a real-time operating system. Since our entire program will be based on sensor inputs and responding to the incoming data immediately we thought we should just go with the Arduino even though none of us had worked with one before. After looking into a few different models we settled on the Mega 2560 since it had ample I/O pins, I2C capabilities and adequate memory (256kB) and clock rate (16MHz) for our needs.

### IMU

In making the decision on what inertial measurement unit (IMU) we wanted we knew it needed to be simple and fast. We found it would be easier if we bought an IMU that contained the magnetometer, gyroscope, and accelerometer all together with one interface rather than purchasing each separate and wiring them all up manually. A common IMU used in quadrotors and recommended on a number of blogs was the SparkFun 9 Degrees of Freedom Sensor Stick. This had good quality sensors that were fast and all connected internally presenting only a single I2C bus to be connected to our microcontroller. It also is very small and lightweight which should fit into our design nicely. The slowest component is the magnetometer with an output rate of 160Hz which is still fast enough that we can sample every 10ms in our flight algorithm allowing time for processing the data.

### Batteries

Deciding what batteries to use is Zachary’s lab. So he ordered three different batteries. One rated at 2200maH, one at 2700maH, and one at 3000maH. He will be conducting tests to determine how much power is needed for long enough flight time. It may be that we can power the quadrotor for fifteen minutes with the 2200maH battery. If that is the case than a bigger battery is unnecessary. So the flight time and weight of each battery will be calculated to determine the optimal battery. The battery will be fixed to the undercarriage of the quadrotor through a system that will allow its removal but will also keep it firmly attached to the robot. As a safety precaution removing the battery is necessary since we do not want to damage the quadrotor if the battery is over charged.

### Power Distribution Board

This will be built in house and will be a simple board that will connect the battery to the ESC’s and the Arduino. It was decided that this was necessary to keep the quadrotor design streamlined. It would be possible to buy a board that would do this for us. We feel that this would be easier, but by designing and building it ourselves we can gain more experience.

### ESC’s

Electronic speed controllers (ESC’s) take a square wave from the microcontroller and depending on the square wave’s duty cycle, spin the motors faster or slower. The most recent group to work on the quadrotor SEED project purchased new ESC’s for their motors. New developments have since been made in the ESC technology, decreasing size, but we decided to not purchase new ESC’s for our design. Our reasoning for this decision includes that the previous group had them functioning with the motors we are using, and purchasing new ESC’s wouldn’t improve our design as much as it would increase our budget.

# Sub Components

The following section outlines some of the basic designs and work completed in the various sub-components of the quadrotor project.

## Power Systems

### Battery

The battery is a lithium Polymer rechargeable battery and will have an output of somewhere between 2200 and 3000maH. Deciding what output is the best has yet to be determined. This is Zachary’s lab and the parts have only just arrived so he has not had time to evaluate that best choice. The battery will be fixed to the undercarriage of the quadrotor through a system that will allow its removal but will also keep it firmly attached to the robot. As a safety precaution removing the battery is necessary since we do not want to damage the quadrotor if the battery is over charged.

### Power Board

There are five components that need to be powered through the battery. The four motors and the Arduino MEGA. The rest of the sensors will be powered straight through the MEGA. To achieve simple and ergonomic power distribution we have decided to design and build in house a power distribution board. The four motors are connected to an Electronic Speed Controller (ESC) which is then connected to the power distribution board (PDB). This board will have one input and five outputs. The Arduino output will require between 7 and 12V and the motors require 6 to 20 V. The board will be designed so that the four motor outputs are in parallel which each other as well as with the Arduino board.

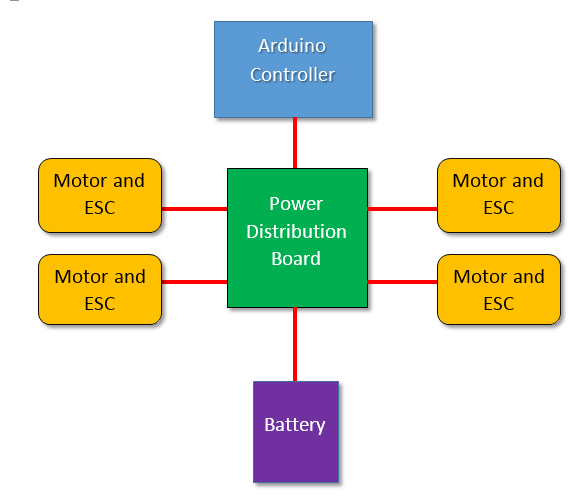


Figure : Power Distribution System

## Control System

The control system for the quad rotor uses the data from the inertial measurement unit (IMU) and ultrasonic sensors mounted on the body to keep the quadrotor stable in flight at a desired altitude as well as to prevent the quadrotor from colliding with walls or other obstacles.

### IMU

The IMU is the basis for achieving flight stability for our quadrotor. Since this is such a crucial piece of the system, we needed to make sure it performed reliably and accurately. The IMU we chose is the SparkFun 9 Degrees of Freedom Sensor Stick. The 9 degrees of freedom comes from the fact that the stick contains as 3 axis accelerometer to measure the gravitational force, a 3 axis gyroscope to measure angular rate, and a 3 axis magnetometer or compass to measure the current heading. As shown below, the data from the IMU will be sent to the Arduino where it will calculate what adjustments need to be made to the motors to maintain stability in flight.

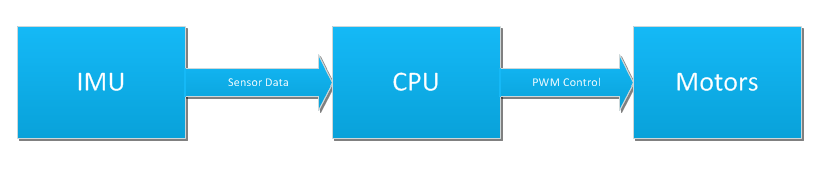


Figure : High Level Block Diagram of IMU Control System

As part of our lab, Aaron will be orienting the IMU in different directions and angles to determine what are positive and negative results according to the IMU as well as the magnitudes of the data returned. This will help with the control algorithm and determining what motors should receive the signal to compensate for a given IMU reading.

The orientation of the craft in the air can be found relatively simply from the sensor data given by the IMU. Since the accelerometer gives the force of gravity which is always pointing down, we can find the angle with trigonometry:

Because the gyroscope returns the angular rate of motion, finding the current angle involves taking the integral of the data given by the gyroscope over the time interval since the last reading.

The extra term D that is subtracted out of the integral is to compensate for the drift in the gyroscope. It must be found experimentally and depends on the temperature.

Theoretically both the gyroscope and the accelerometer should report the same angle since they are both reporting data from the exact same location and orientation. In reality however the data will be different. This is because the accelerometer responds to the “noise” of other forces in addition to the gravity force. As a result, the data from the accelerometer is only reliable in the long-term after these other forces are filtered out. The gyroscope as stated before tends to drift after time as is only reliable in the short-term. To achieve a more accurate angle we found that a complementary filter is often used in quadrotor applications. There are more complicated and computationally expensive filters that can be used, but with the limited processing power of the Arduino we will use the complementary filter. This filter simply multiplies each of the resultant angles from the accelerometer and gyroscope by two constants, the sum of which is equal to 1. Typically the value of the gyroscope multiplier is between 0.9 and 1 and then the multiplier for the accelerometer is the difference.

We will have to find the correct constant for our quadrotor experimentally as well. The IMU control loop is diagramed below.

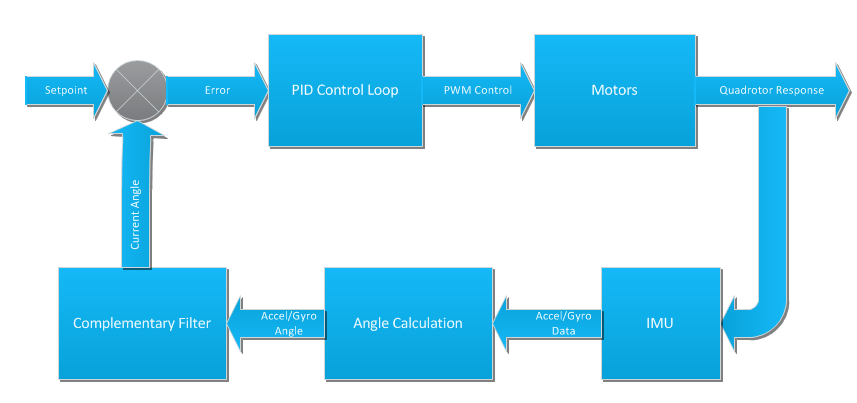


Figure : IMU Control Loop

The proportional-integral-derivative (PID) control loop is executed three times to calculate what adjustments are needed to the motor signals to keep the pitch, roll, and yaw of the quadrotor at its desired orientation. The data from the magnetometer can be used to keep the quadrotor from rotating around the yaw axis. A PID control loop adjusts the output to minimize the error between the set point and the measured output. We will need to find the constants needed for the calculations experimentally.

### Ultrasonic Sensors

The previous groups who worked on the quadrotor project used a Parallax 28015-ND Ping))) sensor to measure the altitude of their craft. We will be reusing this sensor at least initially to measure the altitude of the quadrotor. The control loop for the altitude hold is similar to the control loop for the IMU. The current altitude is read from the Ping))) sensor and then the error is calculated from the desired setpoint. This error is sent to a PID controller where any adjustments are made the motor PWM signals which can then adjust the altitude of the quadrotor.

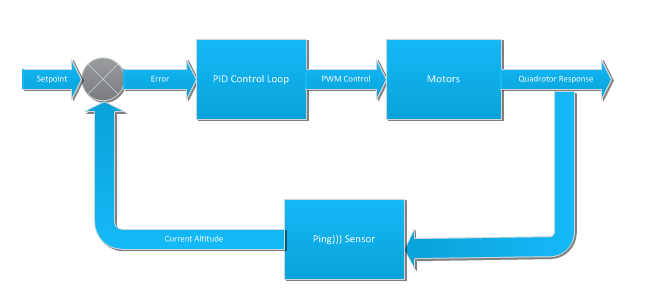


Figure : Sensor Control Loop

If time allows we will begin work on adding collision detection and avoidance functionality to our quadrotor. This would simply read the data from Ping))) sensors mounted on the four sides of the craft and if the data was within a certain threshold the Arduino would adjust the signal sent to the motors to bank the quadrotor in the opposite direction to avoid the collision.

## Motors and ESC’s

The motors are going to be connected to the Arduino controller through ESC’s. Each motor has one ESC connected to it. The ESC’s we have are responsive to a square wave with a frequency of 490 Hz. The higher the duty cycle (closer to 100%) the faster the motors will spin. After some lab testing with our ESC’s and motors, Tom found out that the ESC’s do not spin the motors at all below a duty cycle of 50%. ESC’s are connected like the diagram below. The three wires to the microcontroller are High (5V, red wire), Ground (black wire), and a 490 Hz Signal (duty cycle, blue wire).

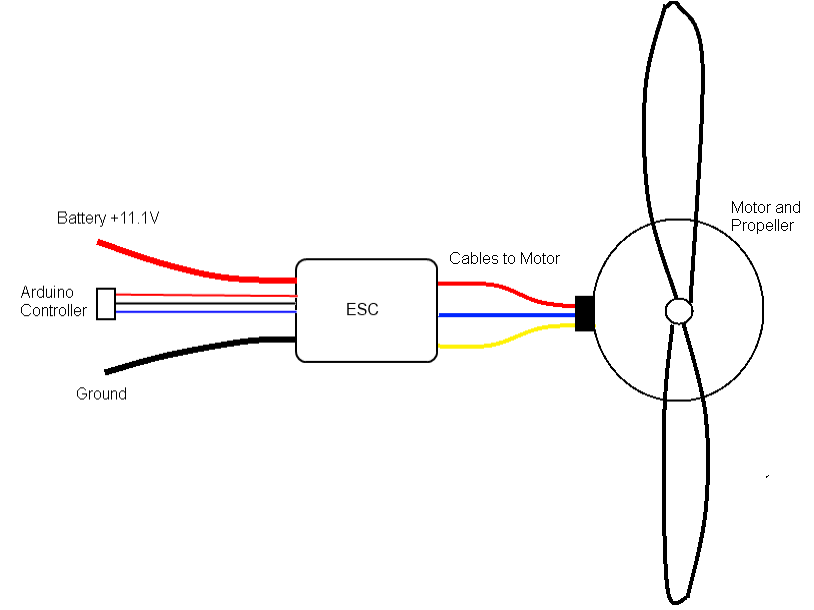


Figure : Motor and ESC connection to Arduino

In order to make our design easier to modify for future changes, we are going to design and build a board designed in figure 5 below. The boxes with wire-ends on the pentagonal circuit board are connectors that will be plugged right onto pins. This allows us to change out a motor or an ESC without worrying about which pins get plugged into the Arduino board. It is just changing connectors. This allows us to easily connect to the Arduino board without exposing any part of the Arduino to possible damages.

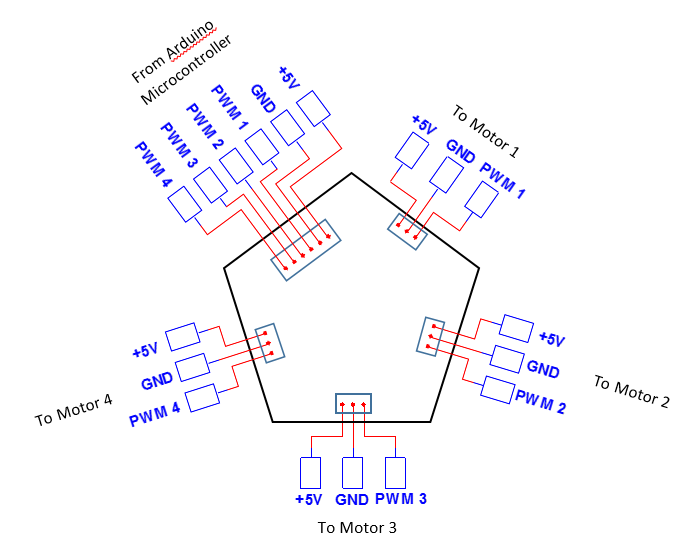


Figure : Motor Control Connection Board

## Wireless Communication

Tom has been looking into connecting wirelessly to our quadrotor in order to give commands, specifically a kill command to shut off the quadrotor. This is a project that he will be pursuing more in the future, with the XBee S6 wireless development kit that the last SEED group purchased. Unfortunately, this hardware is not compatible with Windows 8, so he is pursuing options at Grove City College’s TLC to borrow a computer that has an operating system, Windows 7, which is compatible with these wireless devices.

# System Testing

To make sure that the quadrotor is meeting the various system requirements we will measure the actual flight of the quadrotor using simple devices such as a stop watch and a measuring tape. We will be testing the various components of the quadrotor in our labs as is necessary. This will be done by each team member as they carry out their labs. There are three main requirements that will need to be tested to verify that they have actually been met. These categories are flight stability, flight time, flight performance.

For flight stability we will draw a box on a large sheet of paper that is six inches wider and longer than the quadrotor and set this one on the ground. We then will draw a similar box with four inches of tolerance on another sheet of paper and set this one perpendicular to the other piece of paper. We will then set the quadrotor in these boxes and have it hover there and observe whether it crosses the boundaries or not.

For flight time we will measure that the quadrotor can hold the stable position mentioned above for thirty seconds. This will be done with a stop watch.

For flight performance the requirement is that the quadrotor be able to reach an altitude of eight feet within ten seconds and travel at a horizontal rate of 2 feet per second. To test that these requirements are met the desired distances will be measured out and the quadrotor will be programmed to move from start to finish of this distance. We will then use a stop watch to time the travel distance.

# Budget

Below is a graph showing how much we have spent on the project so far. As of right now we have spent $303.61 which is $1196.39 under our budget. Most of our expenditures have been taken care of so we do not expect money to be an issue for us and will stay well under our budget. The only other major expenditure we expect to encounter is purchasing five ultrasonic sensors to use for navigation. These will only be purchased if we get the quadrotor to have sustained stable flight. Each sensor costs about twenty dollars so that would be an additional $100 to spend.

Table : List of Expendatures

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Item | Description | Price | Quantity | Vendor | Lag Time | Date Needed | Received? |
| Accucel 6 | Balanced Battery Charger | $22.99 | 1 | Hobby King | 19 days | 10/14/2013 | Yes |
| Nano Tech 2200 | LiPo Battery | $18.96 | 1 | Hobby King | 19 days | 10/14/2013 | Yes |
| Nano tech 2700 | LiPo Battery | $31.89 | 1 | Hobby King | 19 days | 10/14/2013 | Yes |
| Nano tech 3000 | LiPo Battery | $20.93 | 1 | Hobby King | 19 days | 10/14/2013 | Yes |
| HXTC 4mm Charging Cable | Connects battery to charger | 4.99 | 1 | ProgressiveRC |  | 10/25/2013 |  |
| XT60 Charging Cable | Connects battery to charger | 4.99 | 1 | ProgressiveRC |  | 10/25/2013 |  |
| Power Supply | For Batter Charger | $9.74 | 1 | Hobby King |  | 10/25/2013 |  |
| XT60 Wire | For connecting batteries w/ XT60 connector to a circuit board | $6.89 | 1 | Hobby King |  | 10/25/2013 |  |
| HXT4mm Wire | For connecting batteries w/ HXT4mm connector to a circuit board | $5.34 | 1 | Hobby King |  | 10/25/2013 |  |
| 9 Degrees of Freedom – Sensor Stick | The Inertial Measurement Unit | $99.95 | 1 | SparkFun Electronics | 12 days | 10/25/2013 | Yes |
| Arduino Mega 2560 R3 Microcontroller | The microcontroller | $58.95 | 1 | SparkFun Electronics | 12 days | 10/25/2013 | Yes |
| HobbyKing SK450 Glass Fiber Quadcopter Frame | Frame Kit | 17.99 | 1 | HobbyKing | 19 days | 10/25/2013 | Yes |
| Sensor Distance Ultrason Ping | Ping Sensor | 29.99 | 6 | DigiKey |  | 3/1/2014 |  |
| TOTAL |  | $483.55 |  |  |  |  |  |

# Schedule

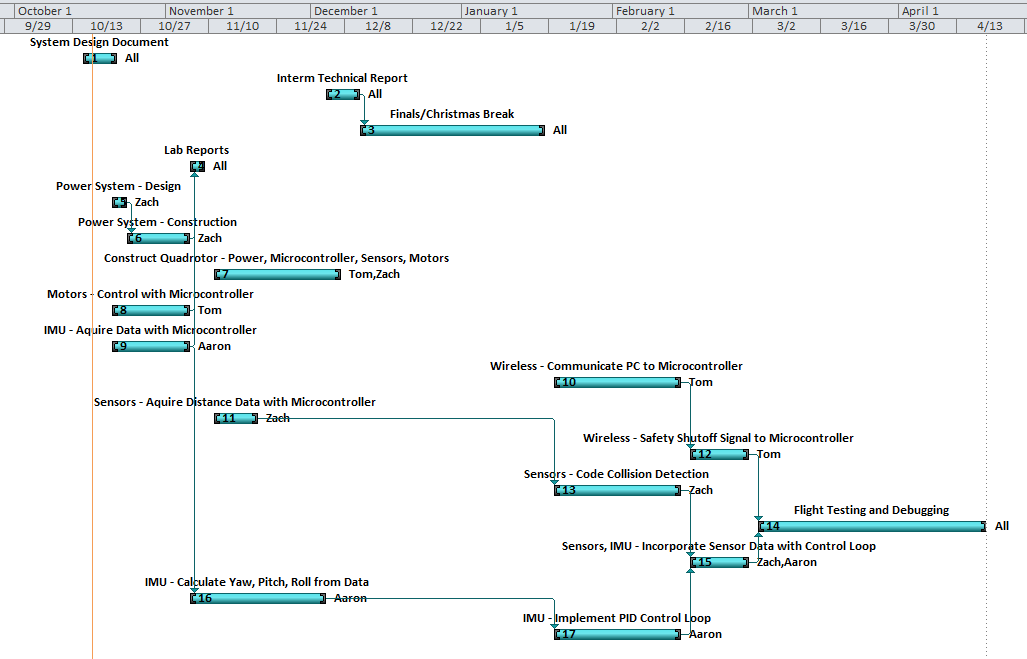


Figure : Gantt Chart Showing Preliminary Schedule of Tasks

## Task Descriptions

Table : Description of Tasks

|  |  |
| --- | --- |
| Task Number | Description |
| 1 | All of us will contribute with the writing of the System Design document. |
| 2 | The Technical Report written at the end of the semester documenting our progress to date will be completed by all of us. |
| 3 | Time off for Christmas! |
| 4 | All of us individually will write our lab reports for our individual tasks. |
| 5 | Determine which battery is best to be used for the quadrotor. This is the main part of Zachary’s lab. A central distribution board will be designed to carry power from the battery to the four ESC’s as well as the Arduino MEGA |
| 6 | Using the machine that can make printed circuit boards build the power distribution system and implement it on the quadrotor. |
| 7 | Install the motors, ESC’s, and Microcontroller on the quadrotor. Determine how to fix the battery and under carriage Ping))) sensor to the quadrotor and install those items |
| 8 | Control Motors with varying speeds with the microcontroller. Determine sensitivity of motors and max thrust. |
| 9 | Write code to acquire the data from the gyroscope, accelerometer, and magnetometer. This is required for Aaron’s lab. |
| 10 | Be able to communicate to the microcontroller wirelessly and send commands to the controller. |
| 11 | Incorporate the Ping))) sensor that we have with the Arduino board. Start reading data from the Ping))) sensor. |
| 12 | Be able to send a “shutdown” command |
| 13 | Write the code necessary to incorporate detecting distances with the Ping))) sensor. |
| 14 | Final flight testing and debugging before the presentation. |
| 15 | Using the distance data from the ultrasonic sensors, incorporate the PID control loop to maintain a fixed altitude and move away from potential collisions. |
| 16 | Using the data acquired from the IMU, calculate the resulting pitch, yaw, and roll angles of the quadrotor at that point in time. This data will then be used in the control loop to keep the heading correct and quadrotor stable. |
| 17 | Actual implementation of the PID control loop in software. Using the angles of pitch, yaw, and roll, adjust the motors to compensate and bring the quadrotor back into stable flight at the desired heading. |

# Conclusion

The two previous groups have had minimal success with the quadrotor SEED project. We feel that with our approach of using some of their research into motors and ESC’s and our use of a simpler frame and microcontroller, we can achieve the success that the previous groups missed out on. By using the basic design outlined in this document, we will spend a lot of time developing the base of the project which is a sturdy frame and stable flight. This will undoubtedly take a lot of time and effort but we are excited to put our skills and talents to good work on this project.

# References

Van De Maele, Pieter-Jan. "Reading a IMU Without Kalman: The Complementary Filter." N.p., 26 Apr. 2013. Web. 17 Oct. 2013. <http://www.pieter-jan.com/node/11>.

# Individual Contributions

## Aaron

As the lead Software Engineer I have been mainly involved with the microcontroller side the project. I did some research into other people’s solutions for a quadrotor and found that Arduino is a common choice for a microcontroller in UAVs. After looking into different choices I presented my pick to the group which is the Arduino Mega 2560. I made this choice because of its ample I/O pins as well as capabilities for I2C to acquire sensor data and PWM to control the motors. I also looked into IMUs as I will be heavily involved in the control algorithm and found that the SparkFun 9 Degrees of Freedom sensor stick will provide usable data for our system with and easy interface. So far I have been able to read the data from the accelerometer and display it on my PC screen using a USB cable connected to the Arduino. With minimal effort I should be able to do the same for the magnetometer and the gyroscope since I wrote the necessary functions and only need to change the register addresses and the initialization data to each of the sensors which will be passed in as parameters.

Also on the software side I found an easy plug-in to Visual Studio for Arduino. This allows us to write code in a more familiar environment and makes uploading the program to the board easy. I also set up a Subversion repository to allow all of us to code and share our changes easily. Zach has experience using Subversion and assisted in getting it set up and working on our machines. The repository is hosted by Assembla which includes additional features one of which is a wiki page. We plan on using the wiki page to document our code and other project tasks so it can be saved as a PDF for future groups to work from.

## Zach

We split up the main components of the design research and I was delegated to research the batteries. The main three components that determine what battery is needed are weight, capacity, and material. I decided upon Lithium Polymer batteries because they were light, inexpensive, and relatively stable. All these rechargeable batteries if not handled correctly can explode or deform. The Lithium Polymer batteries have a long life and do not react in as volatile a manner as most other batteries.

Not knowing the required power consumption of our system I decided to purchase three types of batteries and test them on the quadrotor itself. Fortunately the other parts needed to get the quadrotor running have arrived and so we are almost at the point where I can connect the motor to a battery and time how long it can power the motor for.

I have the most experience with SVN so after Aaron setup the account I have helped resolve conflict issues that we had between each of our checked out versions. I have also spent a lot of time researching different control algorithms and approaches to maintain stable flight. This is a truly complex problem because a quadrotor operates in 9 degrees of motion which means there are a lot of variables to control while trying to maintain stable flight.

## Tom

I have been researching into the complications we are likely to be faced with the physical aspects of acquiring flight. I found out some of the largest issues previous quadrotor SEED groups were related to the physical problems they sounded they were plagued with. Instead of struggling through their problems, I strongly recommended we purchase a manufactured frame that we could assemble and would be balanced and easily replaceable if anything were to get damaged. I proposed several different frames but recommended we purchase the SK450 Glass Fiber frame because it was the best compromise of size, weight, strength, mounting space, and price. The only parts from previous SEED group projects I recommended using were the motors, ESC’s, Ping))) sensor and the wireless capabilities. Using anything else was going to cause more difficulties than they were worth.

My first project with the quadcopter has been with the motors and ESC’s. None of us have any previous experience using ESC’s, and so I researched how they are supposed to be used. We are able to control the speed of the motor by increasing or decreasing the duty cycle of the square wave we connect to the input line. The ESC is controlled by the Arduino board and I will be designing a signal-connection board to allow us to secure the Arduino chip safely and not worry about ease of access. I am also working on connecting to the quadrotor wirelessly for safety reasons as well as for additional features.