Quad-Copter

Autonomous Surveillance Robot

EEL 4914 Senior Design Documentation
under Dr. Samuel Richie

Group # 3
David Malgoza
Engers F Davance Mercedes
Stephen Smith
Joshua West

Summer-Fall 2010

Contents

1 Introduction

1.1 Executive Summary

The quad-copter was intended to be a UAV (Unmanned Aerial Vehicle) with autonomous subsystems. The plan was to equip it with a video transmitting subsystem so its purpose could be for surveillance; however, due to cost and other priorities, the video system was not implemented. It does, however, have a small payload capacity that could be applied to the emergency delivery of low weight supplies to remote locations. Perhaps the most significant purpose of the quad-copter, however, is simply the exercise in engineering design that it was intended to be.

The motivations to choose the quad-copter as a senior design project are varied among the group members. At some point a consensus arose to build a robot, then a robot with sensors and autonomous capabilities, and finally the consensus shifted from a mobile ground robot to a flyer. At first, there were concerns about the technical difficulties involved in designing and building a flyer however, the idea of a flyer generated enough excitement that ultimately the challenge was accepted.

As the quad-copter concept has developed, its characteristics and capabilities became more clearly defined. The quad-copter was intended to be a small lightweight hover-capable vehicle that could be controlled over a custom wireless system. The custom wireless protocol, developed by the group specifically for the quad-copter, was thoroughly designed but was not implemented due to soldering defects in the custom system. A premade Xbee module was substituted for the custom system at a late stage in development. The quad-copter has a robust sensor suite so that it can also operate in a more autonomous mode. The autonomous mode includes subsystems such as a GPS module so that the quad-copter, once given a GPS target location, can make its own way to the target coordinates without further human control. This flight mode requires additional subsystems such as ultrasonic proximity sensors, so that the robot can detect and avoid obstacles (including the ground) and a digital compass, so that its direction can be ascertained and corrected. The autonomous systems were not wholly integrated due to a developmental bottleneck in attaining steady flight. All of the sensors send a lot of data to the MCU, the brain of the quad-copter, which must process the information according to its algorithms and prompt the appropriate subsystems to action. An especially complex task assigned to the MCU is to maintain level flight by varying the speed of individual motors based upon the calculation of data received from the IMU (Inertial Measurement Unit). The IMU combines data from a triple-axis accelerometer and a dual-axis gyroscope using a sensor fusion algorithm. The subsystems of the quad-copter are highly interdependent, linked by the MCU, the physical frame, and the power system. Power comes at a premium in an aerial vehicle where flight duration varies directly with its total weight. The frame must be designed strong and rigid enough to support all the other systems yet, light enough to so as to prolong flight duration to acceptable levels.

1.2 Motivation

The initial motivation of this project was to create a flyer for surveillance purposes. In deciding on the quad-copter, the group measured the differences between this project and a variation on the A.R.M.O.R.D. project (a ground vehicle) from last semester. The group had to evaluate the feasibility of a flyer versus a ground vehicle in terms of the group's resources such as time, technical knowledge and finances. The group had and still operates under the assumption of not receiving third-party financing for the project. No one in the group was a member of the robotics club; and only one member had built a basic robot, which was a variation of the Rhoomba bot. The group consisted of two electrical engineers and two computer engineers which should constitute enough combined technical knowledge to design the control system for either vehicle. The quad-copter was ultimately chosen based upon the realization of feasibility, its possible future applications, and the enjoyment of building, testing, and implementing a flyer.

The group members knew the quad-copter was a possible idea. During the decision process, the group members researched other groups which had attempted to build a similar type of quad-copter. The research yielded some video successes on YouTube, as well as a few failures which questioned the groups' ability to construct and test the quad-copter. To determine whether or not the group should press forward, it was decided to create an archetype by prototyping a very basic two-rotor copter. Within two weeks, the bi-copter was constructed out of PBC piping, two brushless DC out-runners, an Atmel processor, two C-batteries, and hook-up wire connected to a car battery. The initial test flight, which was done in June of this year, was successful in terms of expected power to weight ratios. The successful test was significant in that it substantiated the feasibility of the quad-copter thereby crystallizing the group's resolve to build an aerial vehicle rather than a ground vehicle.

Initially, what the group wanted to do was something that would benefit a user in terms of application. One of the long-term upgrades to be considered was the possibility of using the flyer for carrying lightweight payloads to unreachable areas. In the military, this would mean a syringe of penicillin to a patient in a dilapidated building or mountain range. This same application would also be used in civilian areas which are prone to natural disasters, such as areas affected by earthquakes and tornados. Such disasters can destroy part of a building, and can leave the rest of the building structurally unsound. This would also make it difficult for firefighters and EMTs to reach areas which require basic medical attention. The on-board camera would also allow the user to get a visual perspective of the area being traversed. This would allow users, such as firefighters, alternative routes to be able to reach the people, as well as determine to an approximation the structural damage to construct a time table before a building collapses.

The group wanted to build a quad-copter over the A.R.M.O.R.D. upgrade because the group wanted to build a flyer. It was the first experience most members would have in building a robot, and the experience in building something for each member to test and fly is remarkable. All the members of the group have a love for flying planes and copters in simulation and games, so this is a means of testing how to fly with a unique flyer. The design would also allow for group members to test interaction devices, which are the ultrasonic sensors and the GPS modules, to create a flyer which could use object avoidance and establish a flight path, respectively and concurrently. Because the quad-copter shows great vertical stability, it would be ideal design for testing the paradigm in an aeronautical situation. For other members, the project pushes to design a communications protocol and see it in action. This allowed for certain design challenges due to the motors magnetic effects, as well as determining the issue of interfering with the on-board digital compass. But the most important learning aspect, the group gets to learn the basics of flight in a live scenario, and make adjustments to the quad-copter to simplify the interface for the applications discussed above, as well as for a regular consumer interested in flying this quad-copter.

1.3 Goals and Objectives

The main goal of this project was to create lightweight flyer that can maintain a steady altitude in flight, fly in accordance to a pre-set flight path, has a simple interface for controlling, and has the capability to carry a payload. The quad-copter is to be used by any user capable of responsibly using the flyer. Therefore, a design and controller based on a beginner-level pilot's skills are essential to the project.

The first goal was software implementation and testing. The two processors will implement a high-level language to process commands and for fast debugging. The computer interface would be a GUI, or graphical user interface, developed on a high-level language. PWM signals are to be implemented and tested on the ATMega2560. A means of controlling the signal in small steps would be used for tilt measurements. A means of error-correction would be made to compensate for tilting for forward, reverse, and turning by measuring accelerometer and gyroscope measurements. Checking for altitude would be accomplished by both an ultrasonic sensor and by measuring and estimating pressure and temperature data using a pre-calculated array. The software would take into consideration the sampling rate of all analog inputs from their respected ADCs to confirm the given data.

The second goal was to setup a remote communications device to communicate with the controller. The main controlling device for testing the device would be done through a computer interface, with the possibility of adding a separate controller. The communications would be done using a customized system of ZigBee, 6LowPan protocols, and point-to-point communications between each transceiver. These protocols would be communicated from a microwave antenna, to a dedicated processor, to be sent via UART to the main processor. The main processor would then interpret the command to determine user-instructed commands to determine whether the command was either a non-autonomous, real-time flight controlled by the user; or an autonomous command preset by the computer for a demonstration or surveillance flight.

The third goal involved getting the quad-copter in the air in stable flight. This was defined as being able to lift from the ground to a steady-state position with almost no lateral drift and no spinning. Simply put, what was assigned as the front stays facing the same direction until the user sends the command to move. This would be first used in an environment which allows for more accurate readings under ideal conditions, which were a controlled temperature in an enclosed space, preferably a garage or indoor basketball court. To reach this goal, tests were done on the motors to determine the types of signals needed to stabilize flight. This meant testing the signals to the accelerometer, gyroscope, and the digital compass to determine direction and tilt. These tests were handled by manipulating the PWM signals being sent to each of the motors controlled via computer connection. This will, at first, be done by connecting the frame with a prototype board onto an apparatus to allow for the quad-copter to tilt significantly without crashing the copter.

The fourth goal was to get real-time flight telemetry. This would be done by testing the flight controller, which would confirm the PWM signals sent to each motor. The altimeter, which was a pressure sensor, was used to test the overall height which can be reached within tolerance of the transceiver. The transceiver’s range from the base unit, which was a MacBook, would be tested in terms of indoor and outdoor communication limitations. The accelerometer and gyroscope controllers were tested in outdoor areas to determine stability under light turbulence, or light breeze interactions, as well as testing forward/reverse movement and turning and how it affects stable flight. The ultrasonic sensors would test the quad-copters auto-correction system to allow for autonomous flight control in areas where the user has little or no direct control. The mounted camera system was to be tested under airborne conditions to determine possible vibrations and jittering effects, which will be compensated by a foam buffer. Finally, flight time will be monitored for the quad-copter thus, determining the amount of practical application time and usage. Further goals and objectives are bulleted below.

- To create a light weight, wireless-controlled quad copter with a mounted camera.

- To build a quad-copter which can hover and maintain an altitude at a relatively stable position
- The copter must have navigational capabilities for tracking and positioning.
- The copter must be able to be controlled via computer interface via a separate controller.
- The copter must be able to receive and interpret a preset flight path.
- The copter needs to use object sensing to avoid collision and modify flight path
- The copter needs to return gyroscope and altimeter signals for current flight information (telemetry).
- The copter must give power feedback, from the engines and the processor, via wireless connection (telemetry).
- The quad-copter must have the capability to take video or pictures.

1.4 Requirements and Specifications

The requirements and specifications for specialized subsystems of the quad-copter are addressed individually in the appropriate research subsections of this document. General requirements and specifications for the quad-copter are bulleted below:

- Must be able to lift 2 kg of mass, including the mass of the unit.
- Navigation must be accurate to within 10 ft. (X and Y coordinates).
- A 2.4 GHz signal will be used to transmit telemetry and for direct user control.
- The quad-copter must maintain flight for a minimum of 5 minutes.
- The frame must be lightweight, preferably less than 800g total.
- The quad-copter should have a radius of less than 20 in.
- The quad-copter must be able to detect objects (including the ground) within a range of 1 to 6 ft. minimum.
- The copter must be able to move in 5 basic directions forward, rotate left, rotate right, rise and descend.
- The quad-copter should achieve a minimum speed of 3 mph.
- The copter must be able to be controlled via wireless computer connection using keyboard commands for direct control within a 100 ft. radius.

1.5 Risks

The quad-copter was a complex project with multiple potential points of failure. The assumption that a known risk is preferable to an unknown risk justifies further analysis of the risks associated with the quad-copter. According to a text on software engineering, risks can be quantified by equating risk exposure with the product of risk probability and risk impact (Pfleeger, ch3.6). This approach is sufficient as a basis for assessing risks and ultimately avoiding negative consequences pertaining to the project. The risks involved in such a project can come in many forms ranging from issues of personal safety, a high impact risk, all the way down to losing a letter grade, a lower impact, higher probability risk.

There are issues of personal safety involved with the quad-copter such as the potential combustion of the LiPo (Lithium Polymer) battery if charged incorrectly. This risk can be overcome through researching safety precautions, implementing the precautions, and by buying the most suitable equipment without being overly swayed by price, i.e. by not buying a cheap charger. Another, lower impact, risk associated with the LiPo battery is that if the battery is drained too much then the battery could become un-rechargeable and therefore a time/money risk. This risk can be addressed through power regulation and emergency shutdown procedures but, it could still be considered as having a significant degree of probability and thus, having a significant total modified risk exposure. Other personal safety concerns include working with the substantial current of the LiPo batteries and potential injury from the propellers. Also to be considered is that if the quad-copter were to get out of control and cause injury to the public then liability would be a factor. Another risk that should be mentioned is that some motor/propeller testing was undertaken using a car battery as a power source, which has a potentially fatal level of amperage. There are other risks more uniquely associated with the quad-copter as the choice of a project. Previous senior design groups have had great difficulty in achieving flight stability with a quad-copter. From their mistakes it was learned that early prototyping should be undertaken in order to reduce this risk. If the quad-copter cannot maintain a steady hover in a time frame compatible with the milestone chart then the risk of not having a successful project increases. Another high probability project completion risk is that the wireless communication system, for direct control of the quad-copter, is to be an original, custom designed system undertaken by a group member with substantially more enthusiasm than experience on the subject. Again, previous senior design groups have had difficulties with similar systems. Although this custom wireless system will undoubtedly be rigorously attempted, there remains the risk of failure. This risk could be partially mitigated by maintaining a backup plan of substituting a predesigned system at the last minute.

There are numerous high probability project completion type risks associated with the quad-copter: MCU code development, parsing I2C serial data, power distribution and regulation, Aeronautics, and the list goes on. The quantity multiplied by the probability multiplied by the impact level of these project completion risks would therefore seem to generate a substantial level of risk exposure according to the risk assessment rubric. In conclusion, while high impact risks should not be underrated, lower impact risks can accumulate to threaten a project. The quad-copter is a technically demanding endeavor that will require all members of the group to function on a steep learning curve.

Looking back on the project it can be said that the above risk assessment identified several areas in which the project did suffer setbacks. For instance, the flight stability system was very difficult to perfect which led to other systems being neglected. Also, the custom wireless system did not work properly due to soldering defects. Furthermore, a test flight went awry towards the end of the development life cycle, which caused some scrambling to repair the quad-copter in time for presentation, as well as scrambling out of the way!

2 Research

2.1 Aeronautics

2.1.1 Flight Basics

A few basics will need to be discussed regarding flight capabilities. Aeronautics utilizes concepts of air pressure and exploits them to create a lift force. Thus, it would be beneficial to review a few core concepts of pressure and make some assumptions.

The first concept was the air pressure. Deep sea divers are known for living under larger degrees of water pressure for days at a time. The deeper they go, the more pressure they experience. This concept is similar to people who live in the mountains. The higher they went, the less pressure they'd experience. The observation was made that aeronautical engineers, when designing a system for flight, refer to a pressure ratio rather than simply to pressure. The air pressure ratio related the current pressure with the pressure at sea level, thereby establishing a relation between air pressure and altitude, but not a direct equation. Table 1 provided a basic set of data for relating the air pressure to altitude.

Altitude Measurements		Temperature Measurements		Atmospheric Pressure
Feet	Meters	Fahrenheit	Celsius	kPa
-2000	-610	66	19	108.8
-1500	-458	64	18	106.9
-1000	-305	63	17	105.0
-500	-153	61	16	103.1
0	0	59	15	101.33
500	153	57	14	99.49
1000	305	55	13	97.63
1500	458	54	12	95.91
2000	610	52	11	94.19

Table 1: Barometric Pressure versus Altitude
Source: http://www.sablesys.com/baro-altitude.html From Sable Systems.

These are the calculations found at sea level. For residents in Florida, the temperature further in land will be considerable higher and should be considered for possible higher ground in other regions. Notice how air pressure doesn't drop to below half until 6km above sea level.

The next basic concept was temperature, which holds a direct relationship with pressure. Notice the temperature level placed at approximately 15 C, or 59 F. For most Florida temperatures, this would not come close to what was expected on a normal day. For the demonstration, because the initial demonstration occurred within a few weeks of the new season, temperatures will drop to a relatively close range to this temperature. Like pressure, designers prefer using ratios.

Density, another basic concept, also factored into the aeronautics of the Quad-Copter. Understanding air pressure facilitates a grasp of tangential speed moving through the airfoil, which in turn is needed to calculate lift. Density, in terms of air, was given in the following equation:

Density equation: $$\rho = \frac{P}{RT}$$

Where, R is the gas constant, P is pressure and T is temperature. Thus, a similar relationship exists for the density ratio. The calculations for it on the table are referred to as "sigma". The pressure and temperature ratios allow for the determination, within a degree of error, the approximate value of the current air density for analysis.

The final basic concept to consider in choosing wing and craft design is air viscosity. This was the air friction caused by a craft while flying through the air. This constant becomes apparent at speeds close to and exceeding the speed of sound, which is approximately 343 m/s. At slower speeds, viscosity is present, but negligible in the presence of calmer winds. Therefore, ideal conditions for viscosity will be assumed in the design of the quad-copter. It will also be assumed that the pressure is relatively equal to sea level, meaning the pressure ratio is at 0 km altitude (no significant changes to pressure occur for at about 2 km above sea level). The table will act as a reference for calculating air pressure at the daily temperature. This will need to be checked every day of prototyping and testing.

Also to be considered were the effects of pressure from airspeed at a cross-section, or the continuity and Bernoulli's equations. By the continuity equation presented below, the pressure of a fluid passing through and area at a velocity remained constant throughout the system:

Continuity equation $$pAV = k$$

Where, p is static pressure, A is the cross-sectional area measured, and V is the velocity of the fluid. Bernoulli applied this knowledge to understand the effects of dynamic pressure, which relates air density to air speed, given as:

Dynamic Pressure equation: $q = \frac{1}{2}\rho V^2$

Where, ρ is the air density and V is the speed of the air. This equation facilitates calculation of the required tangential airspeed travelling through an airfoil that provides the required lift. By this relation, lift is related to angular speed and centrifugal force.

2.1.2 Lift and flight stabilization

To facilitate flight stability, two forces must be considered: Lift force and the Magnus effect. The most important force pertaining to flight was the lift force. Lift is based on the upward vertical component acting on an airfoil's aerodynamic center. The force which reacted to the thrust of the airfoil moves in a perpendicular to the airfoil, which was represented as a vertical and horizontal force. The horizontal vector is known as the drag force (as the name implies, it moves in opposition to the thrust). Lift, which will need to be related to air density, is given by the following equation:

Lift equation: $L = C_d qS$

Where, C_d is the lift coefficient, q is the dynamic pressure and S is the planform area, or the area of the blade or airfoil. For most airfoils, this was measured at the center of pressure, which is usually found at the center of the chord of the blade. This concept will hold true for most airfoils found on standard aircraft.

In terms of measuring the lift for a helicopter, the propeller speed was considered rather than the thrust. This was due to the fact that the propeller, which is with few exceptions virtually vertical, was now operating on a horizontal plane. The centrifugal force used to move a propeller plane forward was now the primary lift force of the Quad-Copter. This calculation facilitated optimization of the motors in relation to the propellers at hand. Propeller design also factored in as the blade turns outward from the center. This resulted in a change of the angle of attack to its optimum angle at the tips of each propeller blade. This would be a disadvantage to users who are more experienced with flight stunts for helicopters. In terms of using the copter as a surveillance device, it would be ideal for the most static picture from an aerial perspective.

The other force to consider for was the lateral forces being induced. Normally, there was no drag forces found on the helicopter in hover. This was due to the dual blade propeller moving at such high speeds. The torque generated from the motor created a rotational force, causing the entire craft to spin, hence the second blade to counteract the forces of the torque (called the anti-torque blade).

For the quad-copter, the torques of the counter rotating blades cancelled each other out thus, protecting the system from drift. Another design consideration was the Magnus effect. This assumed a cylinder of infinite length which rotates. What was found was the air would have a tendency to induce a lift if the rotation of the cylinder was moving relatively upwards. This is meant to explain why a baseball pitcher can curve a ball. For the project, the force was considered negligible in still air at hover but, when turbulence occurred, it was possible for this force to become imbalanced and create a drift force. This drift force was also something to be considered when in motion. Initial take-off circumstances were also be taken into account. At start-up, the capability of a slow increase in the propellers' speed to avoid a fast take-off was desirable.

Fast take-offs had long term stress and shearing effects on the internal bolts of the craft. Grounding effects was also a leading factor in faster take-off due to more air being pushed back into the airflow of the blades. This would lead to an initial cyclonic effect surrounding the blades, and created further stress on the frame and the propellers.

2.1.3 Forward Motion

Forward motion was still an issue involving propellers for standard helicopters. Normally, blades on the main rotor can tilt in a way to create forward flight. In the quad-copter design, the propeller blades remained relatively static to the center piece. To compensate and create forward flight, the whole copter needed to be tilted by reducing the lift speed of at least one motor. Timing of this was crucial for maintaining altitude. To maintain altitude, the optimum angle the copter had to maintain, with a maximum forward thrust, was no more than 10-15 degrees. There was also the placement of the sensors for feedback to the processor. The accelerometer and the gyroscope were required to assist in this faculty. For optimal response, the accelerometer handled lateral and vertical feedback, and a gyroscope handled all rotation feedback. The orientation of each device was considered for controlling flight. Further testing in the signal processing allowed the user to observe the required statistics needed to command forward flight.

As a basis for finding the optimum orientation for forward flight, the helicopter model was useful. For the typical copter, the AOA (Angle Of Attack) for each blade was assumed to be optimum for lift at 15 degrees. Since forward flight meant a change in angle of the copter, it also meant a change in the AOA, which would cause a drop in the altitude. To compensate for this, the above equation, involving the centrifugal force, was modified by setting the lift used in hovering as the forward thrust. The new lift was the vertical force upward, which had to equate to the lift in hover before forward motion to maintain altitude. The new equations generated from forward thrust were as follows:

Lift equation (Revised): $L_F = L_{hover} \times \cos\zeta$

Forward Thrust equation: $T_F = L_{hover} \times \sin\zeta$

Where L_{hover} was the lift in hover before forward motion, which was calculated by centrifugal force, and ζ was the tilt angle forward. For small angles (below 2 degrees) it will only appear to be a drift, and testing all angles between 5-15 degrees for any clear results was advised. The minimum value was given to distinguish from the effects of vibration and drift to the forward thrust, and the maximum is considered 15 since airfoils which are angled at more than 15 degrees begin to stall. These same considerations were also given for reverse flight, to allow for halting forward motion, as well as a reverse function for the copter.

Once the copter was in a constant forward motion, the net forces were considered to be zero, and the velocity of the copter was regarded in a linear perspective. This can be said if it was assumed that the plane, in which the copter's propellers are rotating, was a solid and the mass is centered. This allows for a steady-state velocity to be established, which can be determined by the linear momentum equation. Next, turning was also an issue, since forward flight attitude requires more power. For turning, the solution was the same relation with forward flight as was discussed before. Equations 1.5 and 1.6 were the same solutions of tilt required, including more thrust required for forward flight. Since this would require additional power, the revised equations from above were as follows:

Lift equation (rev. 2): $L_{FT} = L_{hover} \times \cos\zeta \times \cos\theta$

Thrust equation (rev. 2): $T_{FT} = L_{hover} \times \sin\zeta \times \sin\theta$

Where theta was the angle of the pitch to be determined, again, the range of angles theta must be kept was between 5-15 degrees, as discussed above.

2.1.4 Frame

In considering the frame, the first consideration was the material to be used. It must be lightweight, sturdy, and affordable. The forces which act on the aircraft primarily will be gravity and air pressure. Gravity allowed for construction under the guidance of a limited mass to allow for structural stability on the ground, as well as control of the copter in the air. Air pressure, which is used to determine the airspeed, will affect the quad-copter's stress on the screws at higher altitudes. The higher the altitude, the lighter the air, the smaller the forces against the frame, which implies the copter's frame, is being stretched. This is what is kept in mind when considering for the base material for our aircraft. For the project, three materials are possibilities due to their popularity in the RC World: aluminum, wood, and carbon fiber tubing.

Wood was a very popular choice for many RC designers due to its low cost and to its soft nature. However, issues begin to arise in an entirely wooden craft, especially in Florida summer heat and humidity which can lead to further complications in propeller and frame design. Furthermore, wood, without much reinforcement, has a tendency to bend over a short period of use such that the design would quickly go out of tolerance. Reinforcing the wood is a possibility however; it would necessitate more mass being added to the system. This would be better suited for an airplane system, but not for a helicopter. Therefore, a purely wooden system would not be the most prudent approach.

Aluminum was the next best choice, due to its tolerance to Florida's environment. With aluminum, test flights were performed repeatedly without requiring reinforcement from another source. Furthermore, due to its increased strength to stress, aluminum was less likely to bend due to take-off or stable flight; also, it carried a stronger stability to the frame. This would mean a longer life for the frame in regard to its basic structure. Problems arise concerning the weight of the aluminum beams to meet the minimum quad-copter requirements. It is advisable to use a minimal amount of aluminum, due to its more expensive nature, relative to wood. Aluminum could also act as a plate which can stabilize the main board, as well as reinforce the frame.

Carbon fiber tubing was found to be best option for the quad-copter due to two of its strengths: it can handle stretching better than wood, and it is more lightweight than aluminum. Wood operates better when it's being compressed, which makes it perfect for internal structure of a larger version of the craft. It would be less effective for the quad-copter regarding long-term and maintenance issues. Aluminum would be a solution to these issues of structure, since most metals (especially aluminum) can handle external stretching on the structure. Further research concluded carbon fiber tubing would be exceptionally costly for replacement. It was advised to use this as a production version for commercial application, for its cleaner look. However, for testing and demonstration, it would not be the best choice.

2.1.5 Propellers

Regarding the propellers used for this project, a choice emerged as to purchase premade blades, or to design them from scratch. Designing them had one major advantage, namely, any size and pitch blades could be fashioned without constraint. In addition, an attachment to the rotor could be created, which would mimic a helicopter's structure. The major disadvantages were that the only materials conducive to this type of experimentation would be wood, and the fact that no group members had ever designed or balanced an RC propeller blade from scratch. Purchasing premade blades allowed the use of stronger, lighter plastic as well as be a more efficient use of the group's time. It was for these reasons that premade propeller blades will be bought.

Most of the available propeller blades have a maximum chord length, the length from the front tip to the back tip of the propeller blade, of 1 inch. By using the assumptions above, the best type of blade for the quad-copter can be determined. To provide enough room for the centerpiece, which would house the batteries and main controlling unit, and to prevent interfere among the blades, it was decided to choose blades which were at the radial length. For an 18 in. radius frame, as high as 9 in. radius blades would be acceptable. Larger blades would mean a possible greater risk to the inexperienced pilot, the engineers involved in testing, as well as any nearby spectators. At the current state of analysis, a 12 in. radius for the frame seemed the best candidate for the quad-copter, being less costly and easier to maneuver than with a larger blade. This smaller design allowed for more versatility in mobility, and can be an implementation for future development. Therefore, 6 in. radius blades would be largest blade for this size of copter.

Finally, there was an issue with the material for the propellers. Four materials were very popular for use: plastic, wood, fiberglass, and carbon fiber. Fiberglass and carbon fiber would do for the purposes of a demonstration however, these materials are more expensive to acquire. Wood was still popular with most RC flyers, and can be used for stunts. Wood, as an RC propeller is prone to breaking very easily at high speeds. Wood would needed to be glued onto the rotor shaft using a type of plastic wedge. This made replacing a broken propeller difficult and time-inefficient. Plastic, like fiberglass and carbon fiber, doesn't suffer from the use of glue. A simple washer and screw held the blade in place. Furthermore, replacement of the blade was quick and simple. Therefore, a plastic blade was the best choice for the quad-copter.

2.2 Power Source

One of the most essential items considered in designing the quad-copter was the power source; which must provide a significant current to accommodate the four motors. An auxiliary, low-Watt battery system was used for the main board, sensors, wireless communication, and video systems. Specifications and requirements for the primary power source include:

- High power RC battery – between 10.5V to 12V at operational amperes (varies from motor to motor).
- No more than 800g of mass dedicated towards power.
- Must be able to sustain a flight time of 10 minutes or more.

Regarding the types of batteries for the particular motors researched, there was a trend found of requiring a minimal voltage of 10.5 V for operation. This was due to the effects of the motor which, when read from a multi meter, left a drop in overall voltage, which was due to the motor. The motor was a three-phase load, and thus follows Lens's Law of inductive energy. This would ideally create an equivalent reverse force on the battery, assuming no impedances from the battery or the inductive load. The equivalent series network between the battery and the motor created a loss in this force, or BEMF (Back EMF), and allowed for a drop in voltage.

For the initial test motor, the TowerPro 2410-09L Brushless Outrunner, the voltage dropped by approximately 2.5V. By then dropping the voltage, the determination was made that 10.5 V was the minimum required voltage. Once the power supply dropped to below 8.0V, the engine had stopped. One member read the actual voltage from the multimeter after disconnecting the motor from the system, and it read 10.48V.This was also measured at 20 percent duty cycle from the controller. This made the choice official: for the motor to run optimally, the required source should have a voltage no smaller than 10.5V. The maximum of 12 V can be used however; anything higher would come with a considerably greater financial cost and with the added risk of burning out the motors and the system.

The next issue was the current, which was the key for calculating the approximate flight time. Most batteries weren't rated by their current, but their charge in terms of current-time (usually mAH or AH in RC power supplies). Since flight time was assumed as a linear function of the battery, then the maximum time was easy to calculate as shown in equation 2.2.1:

Motor Operation equation:

$$\frac{Q_{charge}(in\ A \times hr)}{I_{mo}(in\ A)} \times 60 = t_{motor}(in\ min)$$

Where Q_{charge} is the charge held by the battery, I_{mo} is the motor's standard (or maximum efficiency) operational current, 60 is the conversion factor from hours to minutes, and t_{motor} is the time the maximum time the motor will remain on. This formula held for the maximum current output in considering a worst case scenario.

Since there are 4 motors operating, three options needed to be considered as follows (with the operational motor time being t_m for each calculation):

- Option 1 – each motor has its own battery source. In which case, the formula above would apply to each motor. The downside was that the added mass would be unacceptable due to the possibility that the batteries would likely consume up to 1 kg of weight (well over the specification tolerance). Thus, $t_m = t_{motor}$.
- Option 2 – one battery for every two motors. In essence, half the power would go to each ESC and motor. The weight would be closer to 500g, and within specification. Issues may arise in designing the power supplied to the main board. Therefore, $t_m = t_{motor}/2$.
- Option 3 – one battery for all motors. This was the ideal design in considering a battery to power the motors and the processor due to simplicity. The issue of using a single power source for all the components would pose a problem for batteries near shutdown. If a voltage detector

was to be used, the microprocessor will need time to shut down power to the other subsystems to protect the system. A separate power module was required for this configuration to work. Thus, $t_m = t_{motor}/4$, approximately.

Finally, there was the mass to consider. Since it was ideal to keep the quad-copter at a low mass, having power efficient batteries with a fast recharge time, reliability in battery life-longevity, and practical application of power be considered in design. A perfect example would be powering the main board, which house the processor and sensor control functions, to be turned on and off by either a main battery, or a separate battery and allow for a dedicated power supply to the motors. This was considered more in the design section of the documentation.

2.2.1 Batteries

There are five battery types used by all RC beginners, hobbyists, and enthusiasts:

1. Alkaline
2. Nickel Metal-Hydride (NiMH)
3. Nickel Cadmium (NiCad)
4. Nickel Zinc (NiZn)
5. Lithium Polymer(LiPo)

For most electromechanical systems, such as RC helicopters, batteries were required for engine start-up and signal management. Since all cars run off of a DC source, the electric network was required to convert the DC energy to AC by means of relays and inverter networks. From an analysis perspective, this requires both DC and AC analysis, in terms of voltage, current, and power. This also effected DC power consumption, as well as thermal effects that can result in component shutdown (the motors especially).

The argument was simple: “Which was more important: motors, the processor, the sensors, or the video system? When was there an exception?” At this point, it was assumed the video system has its own power source. Therefore, we only need to concern ourselves with the motors, the processor, and the sensors. The sensors themselves were auxiliary systems which operate with the processor, and are not entirely necessary for flying the craft. However, it does give the user essential data regarding flight data and analysis for control. For this section, the sensors and processor were considered as the main board.

Most of the components on the main board were low-power devices, operating in current ratings as high as the mA to as low as the nA range. This made most power dissipation from the source minimal, at best. However, it was an issue in regards to initial start-up, and at shut-down. The effects of current and voltage spikes due to an under-damped system, and the reverse current on regulators were an issue regarding the life of the components. With the exception of the GPS and the compass module all the sensors are designed as analog parts. These parts were ICs which can easily overheat due to unregulated current and voltage. Furthermore, the digital components, which were the processor, the GPS, and the digital compass; could not suffer from under-damped DC effects (overshooting nominal operating, standing wave effect, electromagnetic interference due to signal leakage or improper shielding).

The parts themselves may not necessarily have the same voltage and power needs. The processors which would be optimal for the application range in the nW range, with an operational voltage of 3V to 5V (operating with mA!). The sensors, however, will require less power to give the required feedback signals to the processor and to the user. This would mean, without any hard values, to calculate the overall power dissipation of the board to anticipate temperature changes and power consumption. There was also the issue of thermal effects from the parts, which will be able to operate at 25 C n with no effort. For our final demonstration, which was assumed to be in an outdoor environment, the thermal equations must be taken into account. For Florida, the ambient temperature is between 60 to 110F year round. As a solution to most of these issues, voltage regulators will be required for distributing power to the main board. This will be discussed further in the Voltage Regulator section.

The following gives the recommended specifics for each battery and their applications to which they may be the best fit. Please note that all batteries related to the motors will use a C-cell configuration (with certain exception in the LiPo cell batteries). All of these batteries have massive overlap in their applications. It is important to consider effects of each batteries weight, overall charge, and best application. Thus, the batteries themselves were considered in their life-longevity, feasibility, and reusability.

Alkaline Batteries

The most basic of all of these batteries was the most commonly used battery in the world, alkaline batteries. In using a system which was entirely alkaline, the quad-copter can get an overall life based on a non-recharging source, which was optimal for environments where battery charging was not an option (urban areas with poor infrastructure, rural areas with no electric power, jungles, deserts, and other uninhabitable or inhospitable terrain). Furthermore, they're cheap. They can be acquired in almost any store in the US and in most first world stores. The biggest problem with alkaline batteries was their low charge. Most batteries carry about 1.5V with a current of 700mA or more (depending on the load). This was due to battery architecture and personal safety issues. Note: the minimum current required for stopping a human heart inside the human body is between 100-500mA. For designing a power source for the motors, the amount of batteries required to operate the motors was considerable.

This example was used for demonstrating the ideal conditions for dealing with the load. Let's say there was a motor which requires 10.5V at 8A. For arguments sake, assume a source of up to 10.5V composed of brand new AA Energizer Max E19BP-16H batteries, (rated with an advertised charge of 2850mAH). Furthermore, assume ideal conditions for the battery. Thus:

$$\frac{10.5\,V}{1.5\,V/_{battery}} = 7 \text{ batteries}$$

Using the formula from above, all 4 motors could run for approximately 5 minutes under standard operating current. This was not an unreasonable flight time for the given specifications above. However, there is a cost issue of using too many disposable batteries. After a trial of ten flights with this battery setup, over 70 batteries have been used, which would retail at about $40. This didn't mean the alkaline was not without use in this project. Because it doesn't have much power to draw, the overall power supplied was a perfect match for the main board. This would allow for a dedicated power supply to allow for the maximum flight time of the copter. Plus, since DC batteries are not prone to the same issues of recharging, they can be replaced without effecting significant extra overhead.

Nickel Metal-Hydride (NiMH) Batteries

The next alternative, NiMH batteries, were common in most cordless phones. This was one of the first rechargeable battery sources discovered and is still considered as one of the most reliable sources for longevity and reliability. This is still a very common source to use in some RC toys and basic models of RC. These batteries were more expensive than their alkaline equivalents, and cannot hold a charge for as long. What they lack in charge, they make up for in reusability. With these batteries, there are two issues regarding power. First, the highest you typically see these batteries rated was at 9.6V. This posed an issue with a voltage limited ESC and motor, since there may be a voltage drop of 2.5V due to back EMF. As a solution, two batteries can be configured in series to handle the same task. This caused issues involving current and heating due to EMF effects. This was leading to an issue of weight from the batteries. Even if the batteries will allow for the voltage with no effects, the weight was still exceeding the maximum specified tolerance of 800g. And second, recharge time was considerably longer than any other popular rechargeable device. This would mean longer time between flight tests, possibly one a day.

Like with alkaline, the NiMH batteries were used for powering smaller devices with ease. As stated before, these batteries were used on cordless phones, which can be used constantly for approximately 2-4 hours, depending on the phone. Most NiMH batteries were rated higher than their alkaline counterparts, being ranked from 1200mAH to 2000mAH of capacitance for most C and D class batteries. This was a good recommendation for a rechargeable source. The typical voltage for AA and sub-C batteries used was rated at 1.2V, which would mean more batteries required if higher voltages are a necessity. This would be solved by the application of a boost regulator if the design requires.

Nickel Cadmium Batteries

The more popular brand of rechargeable batteries for RC Designers was still the Nickel Cadmium batteries as they are slightly cheaper than NiMH, and having a faster recharge time. These batteries were once the power supply of most laptop computers during its earlier years. They were chosen for their fast charge time, which was ideal for earlier laptop users, such as the military for field analysis and computer communication.

Regarding the Quad-Copter, most prepackaged batteries come in 6.0V (5 cell), 7.2V (6 cell), and 9.6V (8 cell) packs, usually rated at approximately 2200mAH per battery (the Sub-C battery from Turnigy was considered as a basis, though there are packs with capacitance as high as 4800mAH). Like NiMH batteries, these batteries hold a smaller potential of 1.2V, opposed to the alkaline with approximately 1.5V. Therefore, 9 batteries in series would be required to generate the minimum required voltage for the motors under consideration. This could be done by setting up batteries in series with one another to the desired amount. However, this meant recharging must be done individually, or more practically using a combination of a balancer and a charger to keep an even charge among the batteries.

NiCad batteries lost a lot of popularity due to problems involved in charging and lifetime performance such as:

- Memory Issues in charging
- Dendrite formations
- Heavy Metal Poisoning
- Reverse Current

These issues were discussed in further detail in the Battery Risks section. Despite these issues, people still use NiCad batteries to this day. Because they are cheaper and recharge faster than the NiMH batteries, they were still popular with most cordless phone designers. As such, it was also an excellent choice for the main board as a separate source. Like the NiMH, they hold a 1.2V, typically, and the same design consideration of the boost regulator was a viable option if batteries become too heavy.

Nickel Zinc (NiZn) Batteries

A brand of batteries now making a comeback was the NiZn batteries. These batteries have been found in applications for electric bikes and vehicles, and were now being scaled for smaller applications such as cordless phones. The sizes of the batteries are similar to that of the NiMH or NiCad batteries, except they were currently distributed very popularly in AA and sub-C sizes. The voltage on a single AA or sub-C battery is approximately 1.6V, a 33% gain in voltage over its Nickel-based counterparts.

According to Powergenix, one of the main developers of NiZn batteries, these batteries hold a capacitance of 1800mAH minimum, and a 2000mAH typically. The charge time on these batteries is 2.5 hours, typically, which made it comparable to that of the NiCad charging capabilities. The battery still contains heavy metals and should be given caution if its contents spill out of the battery. Please refer to the “Battery Risks” section for further details.

Because these batteries have begun circulation in this condition, NiZn batteries have not been tested or fitted to an RC. This was due to its recommended charging capabilities. Most of these guidelines are stated below in the Battery Charging and Battery Risks sections. The recommended charge time per cell is no more than 2.5 hours (150 min). There was also an issue of the higher discharge rate. The batteries have a minimum average capacitance of 1800mAH. With a high discharge found per battery, it naturally would decay the operating time of the battery. It was recommended to use this system provided the maximum draw used battery was no more than 1A per cell. Beyond this, there was very little known about the NiZn batteries that can be confirmed at this time. Due primarily to the support given by the spec sheet, it was concluded that NiZn batteries are a good option for powering the processor, sensors, and transceiver system of the quad-copter.

Lithium Polymer (LiPo) Batteries

For the best choice in batteries, there were the LiPo batteries, the predominating rechargeable battery. LiPo batteries have a much higher voltage rating in one cell than in its predecessors, as well as slightly larger weight. However, the charge typically found in these batteries was between 3.6V to 4.7V per cell. This allowed for the construction of a powerful battery with a minimum number of cells required. This was also useful for issues concerning overall weight, which was an issue with the other considered sources.

The only concerns with the batteries themselves were the cost and the safety issues. The cost of most batteries will vary depending on the charge capacitance used. 2300mAH at 11.1V typically go for $32.99 to $38.99, while the 5500mAH at the same voltage range from $99.99 to $149.99. Careful design and timing considerations were given to these batteries to ensure these batteries do not drain too much during charging (discussed more in Battery Charging). During initial research, the two main distributers of batteries were the ThunderPower and Racer’s Edge. Though there are other distributers which at times offer better deals, these provide a more consistent supply of their batteries, and are the recommended choice for purchasing. Table 2 shows some of the recommended packs for the quad-copter which would work best. Please note, all of the following cells give a maximum continuous discharge of 30C.

Battery	Dimensions (mm^3)	Mass (g)	Price
Racer's Edge 3400mAH	47 X 138 X 23.5	N/A	$47.99
Racer's Edge 5200mAH	47 X 138 X 38.5	N/A	$149.99
ThunderPower G4 Lite RX 2250mAH	25 X 35 X 102	179	$89.99
ThunderPower G4 Lite RX 3300mAH	22 X 43 X 136	270	$109.99
ThunderPower G4 Lite RX 2250mAH	28 X 47 X 160	399	$164.99

Table 2: A list of 3-Cell 11.1V LiPo Packs (recommended)*
** Available as of 6-15-2010.*

The other issue was the safety issue in charging these batteries. Some LiPo packs come with their own charger, which was desired for the project. However, in case the pack does not come with a charger, it may be because the battery pack is a customized pack, and requires special attention. If this was not carefully monitored, these batteries can catch fire and can create severe property damage or injury. This will be discussed more in the Battery Risks Section. Because the battery packs hold a lot of power, they can be wired to the main board for powering the components. Issues arose in the power being supplied to the main board. Since there was a great deal of power being sent to the board, it's possible to melt the board and overheat the components well beyond their operating temperature tolerances. A voltage regulator with a heat sink was the best option, if testing shows it was required. It's possible this may necessitate placement of a fuse to protect the main board and its components.

2.2.2 Battery Charging

When it comes to the most common rechargeable batteries, most times all that was required was a premade charger. This was the recommended method for charging any type of AA, AAA, C, or D class rechargeable batteries. They're easy to find and typically were fairly cheap at most stores. Most battery dealers (Duracell, Energizer, and Radio Shack to name a few) sell these chargers for anywhere from $10 to $50. This was a recommended method only for batteries with alkaline battery holders. For RC battery packs, some batteries came with AC adapters for quick, easy charging. These exist for some of the smaller packs (6.0V typically, 9.6V at best). These were usually packs of "true-C" cells comprised of NiCad or NiMH. For the quad-copter, this was an insufficient voltage due to back EMF effects from the motor. LiPo battery packs and customized NiCad and NiMH packs used chargers which can control the voltage and the current being sent to the battery. Since LiPo batteries were the preferred choice of battery for this project, the battery life and safety must be considered, depending on the battery.

Most computers and cell phones which use Li-Ion batteries have a power regulation controller called a BMS (Battery Management System), which allows for voltage and current to be sent to the cells which require charging. RC batteries still require this type of charging for performance during use. Most of these batteries do not have the capabilities of supporting a controller from their respected manufacturers. The best option was to use a balancer, which does the exact same job as the controllers on a motherboard. Because these are more costly, it is recommended to use what the manufacturer recommends to all amateurs in this venture.

2.2.3 Battery Risks

Each battery was known for its faults regarding safety and usage. Without considering this, batteries will at best be financially costly, and at worst cause serious harm to the engineers, testers, and the user. This section is meant to summarize most of the concerns with handling all of the batteries listed above.

- **Overusing the batteries*.*** Regarding rechargeable batteries, once the battery is dead, it's dead. This is especially true for the LiPo and NiMH batteries. This is usually true for NiCad and NiZn batteries, with certain exception of dendrite formations. These dendrites cause a short circuit between the cathode and anode. A shot of large current could fix the problem, but this is something that is not recommended if there is more than one dendrite formation. Alkaline batteries have a tendency of leaking fluid if kept unchecked for long periods of use.
- **Overcharging batteries*.*** As stated before, more complex electronics have a means of recovering charge to a battery. If a battery becomes overcharged, it will begin to get hot. In some cases, there have been fires started by improper charging techniques. Be present when charging all batteries. Do not let them sit on a charger overnight.
- **Improper or misuse of batteries*.*** It is extremely important to double check all device connections throughout designing, prototyping, and testing. It should be noted to possibly ground oneself as an extra precaution to not incur a shock.
- **Heavy Metal Poisoning*.*** Most of these batteries are comprised of metals not naturally found in the human body with any abundance. This is especially true for Cadmium. A battery leak of a NiCad, necessitates calling 911 to inform them of possible Cadmium poisoning.

Tips for protecting the battery, the user, and the designer:

- For the designer's considerations: with every use, the battery's operating time should be recorded. The statistical analysis for handling an average time will both confirm the calculations above and allow for more accurate testing and flight time.

- Once the battery is low on energy, the user should stop operations and begin charging.
- The best choice for charging a multiple- cell battery pack is a balancer. The user should use this to allow for maximum charging of each cell. Remember: observe the battery when charging to make sure it doesn't become overcharged.
- If a battery dies in the process of testing, the user must dispose of it properly at the local waste management facility.
- If exposed to battery contents, the user must contact 911 and the lab supervisor.
- If the user start to feel dizzy, hard to breath, dry sore throat, or nauseous, the battery must be stored in a sealed, cool container, as it may be due to a leak.
- It is highly recommended for the user to double and triple check all connections throughout the project.
- The user and the designer must test for any inconsistencies in temperature, current, or voltage in the circuit.
- The user must always check the temperature of the battery. (A temperature sensor is preferred, but not required).
- The user must make sure the battery stored away in a cool and dry storage container in a dry area.
- Users must properly ground themselves when handling the battery with the copter.

2.2.4 Voltage Regulators

Distributing power to all elements of the system required the careful application of voltage and current running through each device. Maximizing the current running through each motor, which was the basis of calculating the power, left little room for high powered components elsewhere. As stated earlier, the best way to eliminate this concern was to setup a separate power source for the components. This limits the main board and all of its subcomponents of power, so using local amplifiers may not be the most advantageous decision. Even if the quad-copter uses a single 3-cell LiPo battery to power the whole system, the battery's current would be too great, and eventually lead to thermal issues in design. The best way to solve this was with a voltage regulator, in either case. A voltage regulator is a device which utilizes the ground to maintain an output voltage while maintaining the same current across itself. Voltage regulators hold two very excellent characteristics:

1. Regulators can maintain a voltage with a steady-state current.
2. Regulators can dissipate energy from the system with the use of a heat sink.

In choosing a regulator, what was considered was what kind of voltage and current each part of the main board requires. The motors were being controlled by the ESC, and require the full amount of charge needed to handle back EMF effects. This was why the only need was to consider the components of the main board for the design. Most of the parts were going to be pre-mounted sensors, so all that was required are the individual components and their power ratings. This information was available from the datasheets of each sensor, the GPS, the processors, and the components for the wireless system.

In working with a regulator, there were three different characteristics which typically control the operation of the regulator. In working with this protocol, the group established and predicted certain issues arising from the design of the main board. The three key issues were:

1. Thermal Analysis
2. Load Current
3. Maximum Voltage

This was also the chain of command for voltage regulator shutdown and protection for the regulators at National Semiconductor. Thermal values allow the determination of the necessity of a heat sink, and allow help in choosing the proper regulator(s) for the copter. Note: more than one heat sink can greatly affect the temperature of the sensors on the main board in an enclosed space.
To summarize, the regulators which would qualify for the quad-copter's guidelines are the following:

- A voltage regulator which can take in a high input voltage (10-12V max)
- Must be able to maintain current in mA range, nominally.
- Space is an issue on the main board. Using as few regulators as possible is preferred.
- Temperature must be kept to a minimum. Consideration of fans may be required if the source is too powerful.

2.2.5 Linear Regulators

Linear voltage regulators are devices which allow for an unknown input voltage to stream while maintaining a steady-state current. For this reason, this type of regulator has been called a "variable resistor". This regulator received its name by operating within the linear regions of its internal transistors and of active loads being applied. There are typically three pins which use a TO-220 packaging, usually no wider than 1.5 cm. These are easily obtainable at any convenient electronic supplier (Radio Shack and Skycraft being local retailers).

Other regulator designs allow for space constrained areas which allow for a design with more space for other sensors, if onboard sensors are recommended, known as TO-263 packaging. These are surface mounted regulators which are soldered directly to the board and can allow for any other components, be it amplifiers, buffers, inverters, sensors, or other processors, more space. This will also make design analysis easier through observing the PCB directly in terms of tracing. The one constraint is the heat sink for this device is the copper plane on the PCB. For lower power devices, especially battery powered devices, this can used to setup to handle the heating. Issues will arise from improper design in the PCB without regarding the copper plate as the heat sink. Another small packaging schema is the micro SMD packaging. This uses the same concept as the TO-263 and simplifies the structure even further by using a surface mount device with a BGA (Ball Grid Array). This has found a great deal more popularity with processor design. The fourth alternative would be to go with an LLD device, which is a solder-mounted IC device. This would allow for the same issue with room management without having to deal with any pins directly. Best of all, the devices are set for energy dissipation. There is no heat sink the device would directly need to interface, since the power needed for these devices uses a small current. Therefore, a heat sink would not be necessary at lower temperatures.

Issues arise with the latter two designs involving current limitation. Even if thermally it doesn’t exceed this limit, the device is still working with a higher power battery (11.1V with a minimum 2300mAH). Furthermore, most surface mounted technologies don’t normally regulate such high input voltages, due to their possible high currents. It was recommended to use the TO-220 or the TO-263 packaging to handle thermal issues at higher currents and voltages. Because the power of the LiPo battery was too much for most linear regulators to handle, and since current was less than 200mA for most components, a 1A regulator was all that was required for either digital or analog outputs. More than one regulator was required, and must be considered regarding the sensors, the wireless system, and the MCU.

Two types of regulation were required due to the requirements for the GPS, which will be given the maximum recommended voltage of 5V. The rest of the architecture will be in at 3.3V, typically. Later research, due to modifications in the design would allow for the MCU and the compass module to operate at these voltages. A good first choice was the LP3892 by National Semiconductor, which was rated at a maximum 1.5A load current. This is an LDO architecture which allows for a low-voltage drop of 140mV at the maximum current, 1.5A. This type of linear regulator was used for more powerful applications compared to battery-powered versions, and therefore has its own shutdown lead to connect with a microprocessor. Stand-alone versions for the processors were an excellent way to regulate the power safely to the chip, but not necessary if a dedicated voltage sensor for the LiPo batteries is already established.

Unisonic Technologies' LM7833 was also a good choice due to its simplicity. This was a standard linear regulator rated at 1A, which is still above the maximum load current. The architecture was a standard linear regulator, which stipulates a higher dropout voltage at 2.0V typically. This was a concern regarding the source, since the regulator must be between 5-7V to operate within tolerance. The benefit: there was very low noise attached to this regulator, which was typically 55µV for this particular device. This makes it an excellent device for the analog and digital components, allowing for very little EMF interference with other parts. A source of 7V or more was recommended for 5V devices.

An alternative to this was to use an adjustable regulator, such as Fairchild Semiconductor's LM317, which allows for an adjustable output voltage. Using the adjustable would simplify design and finance at once, and therefore only depend on the resistors themselves (5% resistors was fine for demonstration purposes). The current rating was the same as the LP3892, so both options are comparable. This would require more prototyping, as this type of regulator was typically used for testing purposes. This also holds for Linear Technology's LT1185, which is current rated at 3000mA, was well beyond the scope of the minimum requirements. This was an adjustable LDO regulator, with a small schematic of 3 resistors and 2 capacitors are for the design of our board in terms of space. This doesn't come in a fixed-voltage version, which is the only disadvantage to using this regulator.

Most of the linear regulators were designed for low-power applications, where current was typically in the range of less than 1A. What will determine whether or not the design was feasible to the project will be left to the power source decided upon for the board. Since all the components of the board were considered low power, the LiPo source could be overwhelming. It was highly recommended to use a separate power source for the main board to protect the board from damage created by the current supplied by the LiPo. A set of alkaline or small rechargeable batteries, like the NiMH, in a set of 4 AAs sufficed for LDO's. In regard to the standard linear regulators, alkaline or the NiZn batteries are recommended due to their higher voltages.

2.2.6 Switching Regulators

These regulators are known for operations outside of the linear regions. Due to this effect, the designs of these regulators are known for operating with a PWM, allowing it to constantly shut on/off at considerable higher speeds. This also implies a control of the power supplied to the system being controlled by a duty cycle; and not being solely governed by the direct power rating of the source. Though power was controlled by using the PWM as a switch, inductors and capacitors handle the power storage side of the equation. Many of these come in a TO-220 packaging, due to their high power handling. There were three types of switching regulators which would work excellent for our purposes. These were narrowed down based on their applications towards this project. They are the following:

- Buck Regulator – These are step-down regulators, meaning they are regulators which will cause a drop in voltage. This is the closest regulator type to that of the linear regulator.
- Boost Regulator – Regulators which increase the voltage from the source. Otherwise known as the step-up regulator
- Flyback Regulator – A more complex regulator which can allow for multiple outputs based on design. It may also be used in tandem with other linear and switching regulators.

The buck regulator is the most common of all switching regulators, in that it is a drop in voltage. By creating a drop in the voltage, low powered devices in conjunction with a higher powered device can be controlled without the worry of overloading the components of the board. National Semiconductor has a whole array of buck regulators, like the LM2575 regulator. This type of regulator uses fixed voltages at 3.3V or 5V, depending on the version ordered. This uses an internal clock for its switching frequency at 52kHz, which will need to be considered in design if power concerns arise. The architecture of the switcher application circuit was very simple and small, an excellent choice for the board. There was an adjustable version sold which was accurate to within 4% of the rated voltage, and was considered for simplicity.

Linear Technologies had some excellent counterparts, such as the LTC3830, which also uses an adjustable voltage. This was a more robust design to consider since it uses Zener Diodes and Power MOSFETs to regulate oscillation. This was a device used more for higher power systems, such as CPU devices. It also uses an SO-8 packaging schema, which was excellent for space consumption. The 3.3V fixed version of the switcher was highly stable at currents rate at ±15A, as well as stable between most temperatures from 0 – 50 C. Despite the amount of space the application circuit would occupy, this was able to yield a very strong, stable voltage source which can handle most temperatures. There was an adjustable version which was an SO-16 packaging, which allowed for a programmable switching frequency range of 100 – 500kHz. Since this would take up more space, it would be better to get the fixed voltage version, if this regulator is selected.

The boost regulator was the next most common switcher. Since these create an increase in voltage; a smaller voltage source was to be used as opposed to a larger one. This was an excellent device to use in terms of finding a low voltage device to handle more power. An issue of current arises when considering the design of this regulator. The boost switcher uses its inductor to ramp up current, thus ramping up voltage The third device was the flyback regulator, which uses the inductor to store and release voltage. This allowed for more than one output source to be used, since mutual inductance only requires coils. Furthermore, the design incorporated other types of regulators such as the ones listed above in addition to augment the flyback's ability to work as a boost regulator. This could be used if regulation was needed for a constant current. The only issue being that one more regulator was needed for the board. It was efficient in terms of design, but not space. This was an excellent alternative, but not a necessary one if the use of switchers was decided upon.

National Semiconductor's LM2585 allows for this type of regulation. This regulator had several recommended transformers, and their configurations, based on the type of application design required. This allowed for multiple outputs with greater distinction of voltages ranging as low as 5.0V to as high as 12V. Since it utilized the transformer's mutual inductance, it was possible to increase the voltage across the receiving inductor by using a higher number of coils on the inductor and putting the reference ground to the negative lead. This was another means of boosting the voltage other than the one outlined in the datasheet. The switching occurs at 100 kHz, using an internal clock to operate the switcher, which is beneficial in terms of design.

Linear Technologies' LT3573 was an excellent counterpart to the LM2585. This regulator was made to regulate more voltage than its National Semiconductor counterpart, due to its adaptive nature. This regulator uses an MSE 16-pin IC packaging to handle a programmable power switch current limit. This device was initially intended for the industrial and medical fields, dealing with applications involving higher power with sensitive requirements. This also contains a shutdown pin for the processor, which is essential for the applications stated, but not for the Quad-Copter. The maximum output current goes to 1.6A at 25C ambient temperature. Voltage slowly decays as it approaches 100C, but goes as low as 4.99V. This was the recommended regulator for testing current limitations of the microprocessor, the wireless control system, and the sensors. This as considered for more advanced versions of the quad-copter in redesign.

An issue arose concerning the magnetic field being produced by the inductor of the switching regulators. The sensitivity of the digital compass to magnetic flux was at a maximum of 5.5 gauss. Small inductors too close to the digital compass would generate inaccurate readings at lower levels, which lead to errors in interpreting flight paths and reference points, such as with GPS coordinates. Since flybacks use transformers, more magnetic flux was generated for the digital compass to overcome. Therefore, the switching regulator, though being more versatile than the linear regulator, would not be the right fit for the quad-copter.

2.2.7 Voltage Sensors

There were two critical power issues to consider, the source to the processor, and to the main battery. The processor required power to remain within its tolerated voltage range to avoid brown-out issues. The main battery needed monitoring to determine if the voltage drops too low. Since LiPo batteries were known for having sudden drops in voltage, the sensor must be capable of handling voltage drops with an adequate level of sensitivity. The best way was to use a higher power voltage detector however; the only one found was used for monitoring and charging the LiPo batteries found in hybrid cars, which is made by Linear Technologies. For it to be possible to handle this kind of power with the required sensitivity, it becomes necessary to create an impedance network to reduce the voltage down to a tolerable level, with a button battery system to act as the reference. A CR2032 button battery was considered to act as such a voltage reference for both the processor and the battery.

For the processor and the power supply, the Fairchild KA75XX family of voltage detectors was considered to detect voltages from 2.50V to 4.50V. As the application notes state, the design was meant to connect to a microprocessor in order to send the message of a voltage drop. The processor would've used the KA75290, since its minimum level of accuracy is at 2.75V, which is relatively close to the minimum value is approximately 2.7V minimum. The power supply is a different story. Since this was a higher power device, relative to most of the components on the board, the solution was to use a voltage divider circuit to drop the voltage down to a lower voltage node. To assist in dropping the current, higher resistance rated at higher power rating will help drop the voltage while maintaining a low power source. To protect the main board, it was recommended to use this as a separate module, with the possibility of using a separate microprocessor (an ATtiny8 or a PIC12 module are recommended). As stated before, a timer embedded in the main processor was recommended to determine an approximate average time of low battery life.

Analog Devices has an excellent selection of low-power comparators, but the one that fits the needs of the project best is the ADCMP341. This was a multicomparator voltage detector which can determine and anticipate the next drop in power. This will allows prediction of when the battery is near death. Like the KA75290, it was designed to handle low-power applications. The maximum supply voltage was 5.5V, just under half the voltage of the LiPo source. Like the KA75290, the best way to approach this was to create a voltage divider and use values to give a sensitive enough feedback to create its compare calculations within tolerance of the comparators, to simulate the drain of the battery.

2.2.8 Power Distribution

The motors were to be powered solely by a LiPo Battery Pack. Since most of the motors typically use over 10V minimum, it's advisable to use an 11.1V LiPo battery pack to power them. This was a more costly solution in terms of mass and cost. Racer's Edge 11.1V 5200mAH are an excellent source for all four motors. The alternative was the lesser rated 3400mAH, also from by Racer's Edge, which we can connect to one to two motors, giving each motor more power. This would also require more of the Fairchild KA75290 custom boards for each battery, which was simpler for the battery but not for the processor. For the processor and the sensors there are two voltages to be considered, 3.3V and 5V. Since these devices used very little power (less than 1A), a 4-AA source of alkaline batteries with a rated capacitance of 2500mAH was the best course of action, due to the plentiful supply and reliability of these batteries at cost. The Energizer Max E91BP were the best choice for prototyping and testing. Regarding the regulators and voltage sensor, the regulators will need to be rated for at least 1A and cannot consume a great deal of space. They also require a heat sink capability to allow for higher heat transfer from the components, which is why a TO-220 packaging is preferred. For the quad-copter, the Unisonic LM7833, and Fairchild's LM7805 were the best picks. It can handle up to 1A, which will be more than enough for a battery-powered source. The best choice for the sensor will be the ADCMP341, due to its versatility and price over the KA78XX series. The KA78XX series has a very simple design; however, the ADMP341 has a second reference voltage to help determine if the battery is near its end.

2.3 Motors

Motors were the starting point when calculating flight stability and control. The motors chosen should meet the following objectives:

- Lightweight
- High speed and torque
- Cost effective
- PWM speed controlled
- Synchronized

A motor is defined as a device which can convert electrical power into mechanical power. Unlike piezoelectric devices, which function on converting energy by vibration and friction, electric motors use induction and typically a rotor system to handle power conversion. Motors typically considered for DC applications vary in type from brushed motors, to stepper motors, to servo motors, and to brushless motors. Brushed motors are DC synchronous motors which use a commutator, a thin, small, copper sheet used to reverse current on the power supply. The brushless motor is a three-phase device which uses no commutator. The stepper and the servo motors are devices which rotate and hold a position based on the pulse width being received. In terms of the quad-copter, the stepper and servo motors operate best as a testing device for the PWM pins used on the main board; however, they have no more relevance towards this particular project.

Some common elements found in both motors to help determine the type of motor to be used for our project. This must be specified before looking for motors which are synchronized. Ideally, a synchronous motor is a motor whose electrical power matches and aligns with the mechanical power. Under these circumstances, it is necessary to get a closer look at power from a mechanical point of view. Power is defined as the energy stored or dissipated over time.

Mechanical Power equation: $P_{mech} = \frac{KE}{t}$

In the mechanical power equation, the energy is assumed to be kinetic, which yields two interpretations. One, the kinetic energy is found to be an integral of momentum. This allows for the use of density as a factor in calculating the energy being used. This can also be applied to calculating the energy required for lift, and comparing it to the energy supplied to the motor. And two, kinetic energy can be expressed as the magnitude of the torque. Energy uses the same units as torque, which can be rewritten in terms of 1 J = 1 N*m. This also means that the problem can be rewritten in terms of angular speed, as in the revised power equation below (note, the value "ω" is expressed in radians/second).

Revised Power equation: $P_{mech} = T \times \omega$

This was revised to the following equation because the data given for most motors will typically give at least one of the two values: the speed constant (K_v, in rpm/V) and the torque constant (K_t, in N*m/A). These values give a direct relation to the electrical system being implemented. More importantly, there is also a proportional relationship with the force exerted on spinning the rotor, which spins the propeller for calculating the airspeed, which calculates the momentum of the air forced down the helicopters airflow path to generate lift. Unfortunately, when looking at most of the specifications, the torque constant is not always given. Very few motors show the torque constant under minimal, typical, or maximum states. This also leaves the problem of finding the maximum torque, otherwise known as the stall torque.

As stated before, motors convert electrical power to mechanical power by means of combining induction and rotor systems. This allows for observations of the mechanical effects of the rotor with the consideration of electrical effects. Induction, like capacitance, stores energy in its windings over time and dissipates the energy back to the system. In regard to the motor design, the energy can either be transmitted via mutual inductance. This energy then creates a force which is connected to the rotor, thus turning it. The other force is the countering force back onto the source, known as the back EMF force. The power loss of the back EMF is to be considered as the one force to be overcome to allow the rotor to spin. For most motors, back EMF depends on the motor in terms of the number of poles, the windings and flux on each stator (the site of an internal inductor element of the motor), and the velocity of the rotor itself.

However, there is a way to calculate the power of the back EMF and find the torque constant, theoretically. Like torque and speed, there is a back EMF constant, K_e, which can determine the value of the back EMF and most speeds. The relationship this holds with the rotor's speed is:

Speed/Back EMF relation equation: $K_e = \frac{1}{K_V}$

Typically, this value is in mV/rpm range for most motors. Ideally, the power supplied is equivalent to the power dissipated, which would be the back EMF loss and the torque generated by the rotor. If an ideal motor is assumed, where the back EMF power is equivalent to the rotor power, the torque constant can be determined. After some manipulation of the values, the relationship between the magnitude of the back EMF constant and the torque constant are the same value in magnitude. The units for torque and current are carried over. Regarding values involving stall torque and power efficiency, these values are to be determined experimentally.

Deciding on a motor based on speed was approached by using the centrifugal force developed by the copter's propellers. Based on the information from the propeller section, an angle of attack of approximately 15 degrees was the maximum angle of attack. By vector analysis, the lift force made up a very small amount of the total centrifugal force the rotor delivers (total force*sin(15) to be exact). This can be used to calculate the total force necessary to lift the load currently attempted to use, specifically 2 kg total. To more accurately determine for each motor the total load handled, it was assumed the moment occurring in hover of each arm of the frame from the center is zero. It was also assumed there was no lateral interference, and the copter was maintaining a steady flight altitude. It was also assumed the total weight was at the frame's center point. Thus, each arm of the frame handles .5 kg of mass, and the total force per arm was 4.905 N. Thus, the total force, by calculation, was 18.9515 N. The equation for centrifugal force to be implemented is:

Centrifugal force equation: $F = 2mr\omega^2$

Where F is the total force to be generated in N, m is the mass of the propeller blade in kg, r is the radial length of a propeller blade to the center, 2 blades per propeller, and ω is the angular speed measured in rad/s. If a 10 in. diameter blade was used for the propeller with an equivalent weight of 1 gram per blade, the radial length was approximately .127m per blade, and the overall angular speed needed for lift was 273.152 rad/s, or 2608.412 rpm. Based on this initial calculation, a motor type can be determined.

2.3.1 Brushed Vs. Brushless Motors

Brushed motors were also considered commutated motors because of the copper plane which interacts with the rotor. This plane acts as a storage element to help disperse the energy inside, specifically energy derived by the current. This allowed for control of the motor by controlling the flow of power, rather than a signal. The commutator, due to the friction it causes can also create a slower-speed motor. According to the previously mentioned *Mechanical Power equation*, this would result in a higher torque, which would be perfect for lateral movement, or constant ground movement. This could also be applied to faster lateral movement in flight mode, and can allow for the craft to turn and spin at very quick speeds. This can also lead to instability in maintaining a heading in flight.

There are motors which fulfilled the quad-copter's speed requirements such as the Tamiya 53930 RC. This uses a 7.2V battery source, which was easily acquired as a NiCad or a NiMH source. The Tamiya would be an excellent financial choice for smaller, economical, versions of the quad-copter. However, the motors themselves, because they have a brush operation and can wear over time, were not recommended for flyers relying on long-term use. Brushless motors, on the other hand, use a permanent magnet system attached to the rotor. These are motors which have no commutation plane, which results in larger amounts of heat being dispersed, which occurs quickly. It also results in less friction, permitting for a longer motor life and a faster rpm. This was perfect for gaining lift force with less voltage. Since torque was very low, it will typically result in slower turns and operates optimally for gaining surveillance video and pictures. Typically, a large source of power would be desired for the motors used, ranging in voltages from 7.4 to 14.8V for most of the motors considered. The TowerPro 2410-09 Brushless DC Outrunner has a minimum voltage of 10.5V required for each motor, with a nominal current of 8.4A and Kv value of 840 rpm/V. Therefore, the brushless motor was the best option for the quad-copter.

2.3.2 Motor Control Design

When initially designing the quad-copter, a premade brushless electronic speed controller (ESC) made by the motor manufacturer was used for preliminary testing. Because no one in the group has any prior background in aerial craft design, the group decided to purchase two motors and two ESCs to determine the feasibility of the project. These two ESCs, which had been rated at 30A for testing purposes, were purchased at a more costly rate than anticipated. There was also the issue of availability and shipping with our supplier, which was addressed in California. To eliminate the majority of these issues, creating an ESC of our own is a simpler solution. By using a baseline design for motor control and the three-phase inverter commonly found in power electronics for DC motors, the design itself will be simplified to only its core components. This also allowed for us to modify the ESC with lower ratings, if the current settings of the maximum current rating of 30A are too high or too noisy.

In considering the design of the ESC, there were certain basic requirements which must be met to control the motor: the design and specifications of the motor, the input and output signals in the circuit, and the complications involved with the motor involving the power supply, the motor, and the ESC. This would allow for the group to design the inverter and the controller for optimized load voltage and load current. For the TowerPro 2410-09, the frequency which must be simulated for the motor is at 50Hz, since this is a Chinese-based motor. The PWM frequency is transmitted to the frequency of approximately 8 kHz, to strengthen the 50Hz power signal.

For the design of the inverter, the consideration of frequency analysis going to the load, the brushless motor was vital. Most motors operate at a frequency of 50Hz input at the controller, including the motor. The signal was then sent to one of six power transistor, which will act as a switch to the DC voltage, allowing the DC power supply to act as an AC source. The source of the signal is sent into the ESC from a pulse width modulator (PWM) from the main processor. To control the flow of the inverter circuit to the load, the signal is controlled using six different PWMs on a microcontroller found on the ESC. For this project, there are design considerations to be addressed: the mass of the ESC, and phase of the load or motor. The easiest one was the size, which may be slightly longer than the prepackaged ESC; however, this would've been determined in further design. The second issue was the phase of the load, which carries both resistive and inductive properties. In accordance with Lenz's Law, there was a small amount of electromotive force reversed on the motor due to induction. This lead to complications in the power supply, which further create issues regarding the accuracy of the voltage source. The solution was to create a BEMF filter, which can be done in a variety of ways. However, to avoid complexity in the ESC, an inductor and a diode was an initial start for design.

There were two ways to implement the PWM design in the system: an analog system using comparators, or a digital system using PWMs to act as switches. With the analog system, it required an analog signal which was split into three signals at 60 degree phases apart. The benefit was the design can be a simple rectifier model to be used for sending the signal to the motor, which saved circuit board real estate. The main issue which arose was the feasibility with the motor to this type of ESC. Most brushless motors require both high power and high switching speed, which was maximized at 8kHz, which was significantly higher than standard AC power requiring, 60Hz maximum (50 Hz was recommended for the design of the ESC due to the manufacturer of most motors were located in several foreign countries). The other way was digitally controlling the signal, which can be done with a microcontroller. The controller will interpret a signal and transfer the signal to the inverter as digital pulses. The benefit was that this design allows for software implementation of the analog design described above. Furthermore, the same system can be created with a maximum bandwidth well above 8kHz. The only disadvantage being the necessity of voltage regulators, which do eat some room on the board. The better of the two systems was the digital system for the sake of speed and cost. It was recommended to use a PIC18F family of processors (the PIC18F1230 was recommended), due to the processor holding 6 PWMs. If using a decoder, the Atmel ATMega8a can be used. Since it only has three PWMs, the 3-8 decoder was the recommended route for implementation.

Essentially, the cost in time was too great to be spent on ESC design. Since so much effort and financial resources will be distributed to other systems, it was concluded at this time to use the manufacturer's ESC, or its lesser priced equivalent.

2.4 Microcontrollers

In the market today there are many types of controllers varying from a total hardware solutions (FPGAs) and a completely software solutions (computers). The group choose a microcontroller because it is a limited hardware and software solution. This allows the group to have a very specialized controller, capable of satisfying the requirements.

The microcontroller chosen will be the main controller of the project. There were various manufactures of MCU currently available. The three companies that were looked at were Texas Instruments, Microchip and Atmel. The tables below examine different properties available in different chips. The requirements for the MCU are:

• Minimum of 1 16-bit timer with 4 output compare registers
• Minimum of 2 USART ports
• Minimum of 12 ADCs

2.4.1 Timers

Timers are extremely important in this project. The timers were configured to perform a PWM signal at a specific frequency. The number of output compare registers per timer was extremely important because using different timers to control the motors could result in having a different frequency. The minimum requirement for this project was one 16-bit timer with 4 output compare registers. Table 3 on the following page shows the number of timers of various MCUs.

Manufacture	**MCU #**	**Timers (16-bit)**	**Max number of compare registers per timer**
Texas Instruments	MSP430F448	2	7
Texas Instruments	MSP430F449	2	7
Microchip	PIC18F6620	3	5
Microchip	PIC18F6720	3	5
Microchip	PIC18F8620	3	5
Microchip	PIC18F8720	3	5
Atmel	ATmega64	2	6
Atmel	Atmega128	2	6
Atmel	Atmega640	4	12
Atmel	Atmega1280	4	12
Atmel	Atxmega64A1	8	4
Atmel	Atxmega128A1	8	4

Table 3: Numbers of timers for various MCUs.

2.4.2 ADC

ADCs are another part that was essential to this project. When a sensor takes a reading from the environment, the results can be displayed as an analog signal. It was the job of the ADC to take this signal and convert it into a signal that was understood by the MCU. The number of ADC was important because only one reference voltage can be used per ADC. If there was more than one ADC then there can be more than one reference voltage. The resolution of the ADC was also important, because the number of levels was an exponential value. The number of different levels follows a 2^n curve. The minimum requirement for this project was eight ADCs of at least 10-bits of resolution. Table 4 shows the number of actual ADC units are on the chip and the number of ADC pins it was multiplexed to.

Manufacture	**MCU #**	**Onboard ADC**	**Number of ADCs**
Texas Instruments	MSP430F448	1 @ 12-bits	8
Texas Instruments	MSP430F449	1 @ 12-bits	8
Microchip	PIC18F6620	1 @ 10-bits	12
Microchip	PIC18F6720	1 @ 10-bits	12
Microchip	PIC18F8620	1 @ 10-bits	16
Microchip	PIC18F8720	1 @ 10-bits	16
Atmel	ATmega64	1 @ 10-bits	8
Atmel	ATmega128	1 @ 10-bits	8
Atmel	ATmega640	1 @ 10-bits	12
Atmel	ATmega1280	1 @ 10-bits	12
Atmel	ATxmega64A1	2 @ 12-bits	16
Atmel	ATxmega128A1	2 @ 12-bits	16

Table 4: the number of actual ADC units of various MCUs.

2.4.3 Memory

Memory was another important property of a MCU. This was the area where all of the code will be stored. It needs to be sufficient to hold all of the data structures and variables that need to be declared to run all of the sensors and timers correctly. Luckily, most of the chips listed are or have drop-down versions of themselves with larger memory capacities. Table 5 shows different chips and there RAM and MEMORY capacities.

Manufacture	**MCU #**	**RAM (bytes)**	**MEMORY (Kilobytes)**
Texas Instruments	MSP430F448	2000	32
Texas Instruments	MSP430F449	2000	48
Microchip	PIC18F6620	3840	64
Microchip	PIC18F6720	3840	128
Microchip	PIC18F8620	3840	64
Microchip	PIC18F8720	3840	128
Atmel	ATmega64	4000	64
Atmel	ATmega128	4000	128
Atmel	ATmega640	8000	64
Atmel	ATmega1280	8000	128
Atmel	ATxmega64A1	4000	64
Atmel	ATxmega128A1	8000	128

Table 5: RAM and MEMORY capacities of various MCUs.

2.4.4 Communication

Another parameter that was also essential was serial communication. The GPS unit and the custom communication link both use USART, which was serial communication. Table 6 shows the different chips and the number of USARTs on the chip.

Manufacture	**MCU #**	**USART**
Texas Instruments	MSP430F449	2
Microchip	PIC18F6620	2
Microchip	PIC18F6720	2
Microchip	PIC18F8720	2
Atmel	ATmega64	2
Atmel	ATmega128	2
Atmel	ATmega640	4
Atmel	ATmega1280	4
Atmel	ATxmega64A1	8
Atmel	ATxmega128A1	8

Table 6: Number of USART ports of various MCUs

MCUs with 2 or more USART ports are listed above as multiplexing a single port would be difficult to test and could result in corrupt data.

2.4.5 Packaging

The packaging of the chip was of less importance, but needs to be looked at because of limitations of testing and mounting. Some PCB manufacturers have limitations on the type of chip that can be mounted on the type of PCB that they produce. Also testing certain types of packaging will be more difficult than others. Table 7 shows the package of various MCUs.

Manufacture	MCU #	Packaging	Number of pins
Texas Instruments	MSP430F448	LQFP	100
Texas Instruments	MSP430F449	LQFP	100
Microchip	PIC18F6620	TQFP	64
Microchip	PIC18F6720	TQFP	64
Microchip	PIC18F8620	TQFP	80
Microchip	PIC18F8720	TQFP	80
Atmel	ATmega64	TQFP, QFN	64
Atmel	ATmega128	TQFP, QFN	64
Atmel	ATmega640	TQFP, CBGA	100
Atmel	ATmega1280	TQFP, CBGA	100
Atmel	ATxmega64A1	TQFP, BGA	100
Atmel	ATxmega128A1	TQFP, BGA	100

Table 7: Package type of various MCUs.

2.4.6 Software/Programmer

Another piece of equipment that the group cannot be without was the programmer and software. The programmer will allow the group to transfer the compiled data from the software to the MCU. This was a cost that can drastically affect which chip was chosen. Table 8 and table 9 show the name and cost of the most basic programmer and the name and cost of the software. The Atmel programmer was listed as free because it was owned by a group member.

Manufacture	Programmer Name	Programmer Cost
Texas Instruments	EZ430-F2013	$20.00
Atmel	AVRISP MKII	Free
Microchip	PICkit 2	$34.99

Table 8: Most basic programming device and cost.

Manufacture	Software Name	Software Cost
Texas Instruments	Code Composer	Free
Atmel	AVR Studio	Free
Microchip		Free

Table 9: Free software from various MCU Manufacturers.

2.4.7 MCU Conclusion

With so many choices currently available the group was going with the ATxmega64A1 series. This was a relatively low cost MCU because all of the development tools are free to the group. This MCU has an abundance of ports that far exceed the minimum requirements and also has two different ADC in the chip, which will allow for two different reference voltages. This will be useful when sensors of different voltages are used. Also since there was more than one option for the memory size the correct chip will be chosen based on the size of the code. The only downside to choosing this part was that most of the features and pins on the chip would go unused.

2.5 Software

2.5.1 PWM

In order to use PWM on a MCU the group must use a timer that was built into the MCU. There are sometimes multiple timers on a MCU that can be individually modified to fit a specific purpose, in this case PWM. This will work to the group's advantage because one timer can be outputted to multiple pins on the MCU. Most timers are either 8-bit or 16-bit, with this said an 8-bit timer can count from 0-255 and a 16-bit timer can count from 0-65,535. There was also a prescaler, which was defined by a register specific to the MCU that allows the user to divide the timer by a value that was stored in the timer counter control register.

There are two ways to use timers for PWM. The first was by using an overflow timer interrupt to turn on a specific pin. When the timer reaches its maximum value (8-bit 255 or 16-bit 65,535) the user can chose to have the MCU call a specific function called the ISR and execute specific code. The other way was to use the timer in compare mode. In compare mode the user can define a specific value in the output compare register to be compared to the value currently in the timer. Depending on the MCU, there can be multiple output compare registers for an individual timer. When the value of the output compare register and the timer are equal the user can have an event occur. Using the timer counter register the user can state whether to toggle, clear or set the pin associated with the output compare register. The user can also specify to have an interrupt enabled and act inside of the ISR.

Usually, for PWM the pin associated with the output compare register need to be toggled on and off to produce a square wave. Also depending on the needs of the user, the timer control register can be set to different modes of operation. One such mode was setting it clear timer on match. This can be useful if the timer only needs to count to a specific value.

The other two modes are fast PWM and phase correct PWM. For fast PWM, it uses the timer as described above. It starts the timer at zero and sets the pin high. It counts up until it reaches the value stored in the output compare register. When that value was reached it sets the pin to low and then counts to the maximum value of the timer. When the maximum value was reached it was reset to zero and the pin was set to high again. This was repeated to produce a square wave PWM. For phase correct PWM, it uses the same timer as described above. It starts the timer at zero and sets the pin to low. It counts up until it reaches the value stored in the output compare register. When that value was reached it sets the pin to high and then continues to count to the maximum value for the timer. When the value was reached, the timer then starts counting down. Again, when the output compare register value was equal to the present value of the timer it sets the pin to low. Either one of these PWM modes could be used to produce the wave needed to drive the motors.

2.5.2 ADC Software

The purpose of an analog to digital converter is to take an analog signal such as voltage and convert it into a digital signal such as 128 or 0x80. Most MCUs have ADCs integrated directly into the silicon. When talking about ADCs they have a value called the resolution. When an ADC has a resolution of 8 bits, it means that it will convert the analog signal into a value between 0 and 255.

Another value that was specified by the user was the reference voltage. This was the maximum value that the ADC will reference from when outputting the digital representation. The number of channels the ADC has was the number of analog devices that can be sampled from. So, an 8 channel MCU can have up to 8 sensors being read (This does not take into effect that the user could use a multiplexer to get more channels out of the MCU).

Depending on the MCU, the user can select different features of the ADC. One feature of the ADC was where to source the A_{REF}. On some MCUs the A_{REF} could come from the A_{REF} pin or the A_{VCC} with a capacitor at the A_{REF} pin. The ADC prescaler was another important feature because it can be used to slow down the speed the ADC converts. There was a tradeoff though, the higher the clock the faster the ADC converts but it was less accurate. Therefore the slower the clock the slower the conversion but it was more accurate.

2.5.3 USART Software

USART is dedicated hardware for serial communication on certain MCUs. It automatically sends data across Tx and Rx lines by supplying the data to the proper registers. There are various parameters and registers that need to be set in order for proper functionality.

Some of the parameters are stored inside of the USART control and status register. There are three types of these control and status registers. This was to hold all of the parameters needed to control the operation of USART. Receive complete interrupt enable and transmit complete interrupt enable are parameters that allow the user to determine if an interrupt should be triggered if the receiving of information was complete or transmitting was complete. Also the receiver and transmitter enable are important because if it was a one-way device then there was no need to transmit on a receiver and vise-versa. The USART character size defines whether to transmit 5, 6, 7, 8 or 9-bit frames. The USART mode select parameter determines whether USART will be synchronous or asynchronous. All of these values are set in the initialization section of the code.

There was also the USART baud rate register. This determines what the transmitting and receiving speed USART communicates at. It was a 16-bit register that was governed by the following equation:

UBRR Value equation: $$UBRR = \frac{f_{OSC}}{16 \times BaudRate} - 1$$

UBRR was the USART baud rate register, f_{OSC} was the CPU frequency and baud rate was the required communication speed. With the equation the user was able to determine what value should be stored into the baud rate register. There are some misconceptions thought. The only values that can be stored into the baud rate register are integers. The problem was that if you need to place the value 7.68 into the register you would need to round up to 8. By doing this USART has a higher error percentage and the data was unreliable. The best way to optimize was to choose a baud rate that will result in a baud rate register value that was very close to an integer value.

2.6 Wireless Communications

Wireless Communications was used in this project to control and get status messages from the quad-copter. The group researched 6 standard protocols and 1 custom protocol. There are numerous solutions that will be researched in order to meet the design specifications of the desired wireless communications. Those design specifications are:

- Range: 100 m
- Data Rate: Equal Or greater than 56 Kbps
- Latency: To be 100 ms or less
- Cost: To be Less than $70.

Different wireless communication protocols are available that can meet theses specifications, but budget and time limited the choices to one system. Wireless communication protocols available to choose from include: Wi-Fi, ZigBee, Bluetooth, 6lowpan, Wave, Dash7, and a custom in-house point to point protocol. All of which were researched in the following subsections of this paper. After choosing the right protocol different chip vendors were researched to choose the part that meets the design specifications. All of these protocols have advantages and disadvantages, one might be cost, and another might be not enough time and so on.

The weight of each specification was analyzed and compared between each wireless communications protocol and between integrated circuits. The data rate is not as important as budget and range. Since the status messages and control signal that needed to be transmitted was small. On the other hand budget and range have the same importance. It would have been desirable to choose a part, Part A, with a higher range. But if this higher range cost substantially more than Part B, then Part B would have been the best choice for the quad-copter.

2.6.1 Wi-Fi

Wi-Fi is a widely used protocol for wireless communications in general. It is the best known protocol of the ones that will be researched. This section was focus on the IEEE committee standard 802.11b version of Wi-Fi. The 802.11b standard operates on the 2.4 GHz ISM (Industrial, Medical, and Scientific) band. It has a maximum data rate of 11 M b/s. The maximum range of the 802.11g standard is 38 meter indoors, and 140 meters outdoors. 802.11g can achieve latency below 100 msec. This wireless communications meets and exceeds the data rates, the latency, and the range of the design specifications. The data rate is too high for the quad-copter, as only simple commands and messages are transmitted. This wastes energy that could be used for more important parts of the quad-copter, such a motors and sensors. Since the quad-copter has no need for such a data rate, Wi-Fi was ruled out as a wireless communication solution.

Wi-Fi has many sophisticated protocols that allow it to be versatile in almost all circumstances. The versatility of Wi-Fi makes it expensive and hard to develop for. The price of one Wi-Fi transceiver is $30; this is the development IC and not a complete Wi-Fi solution. Developing a complete Wi-Fi Solution would raise the cost of 1 transceiver. The quad-copter will use two transceivers, and this puts Wi-Fi out of the range of the budget. Although the price a Wi-Fi is high and it has high data rate, a Wi-Fi solution did meet every other design requirement.

If Wi-Fi was to be chosen for the design of the quad-copter then the following IC would was lokked at. The Microchip MRF24WB0MA is an 802.11b Wi-Fi transceiver. The data rates are between 1 and 2 Mbps. The IC physical dimension is 21mm x 31mm, and it comes on a surface mount 36 pins package. The MRF24WB0MA and it can achieve a range of up to 400m. This transceiver is capable of supporting the following security protocols AES, and TKIP (WEP, WPA, WPA2 security). The support for security is an added bonus but is unnecessary for the quad-copter. This is one of the reasons why Wi-Fi is not an ideal solution for the quad-copter, any added bonus might complicate the design of the wireless communication system. The range of this transceiver is beyond good and could be a consideration that might override the pricing and complexity of implementing a Wi-Fi solution for the quad-copter. Since the quad-copter is an airborne vehicle, the farther the communication ranges with the vehicle, the better.

The design of a Wi-Fi solution using this IC can be derived from the sample circuit found on the Data Sheet. According to the sample circuit a reference design can be completed using 5 resistor and 2 capacitors. The interfacing between a microcontroller and the MRF24WB0MA is done through a JTAG port. In order to finish the Wi-Fi wireless communication system an IP software stack is needed. Microchip provides the IP stack for free and it can be implemented using Microchip's own Microcontrollers. Figure 1 shows the schematics of the sample circuit in the Data Sheet.

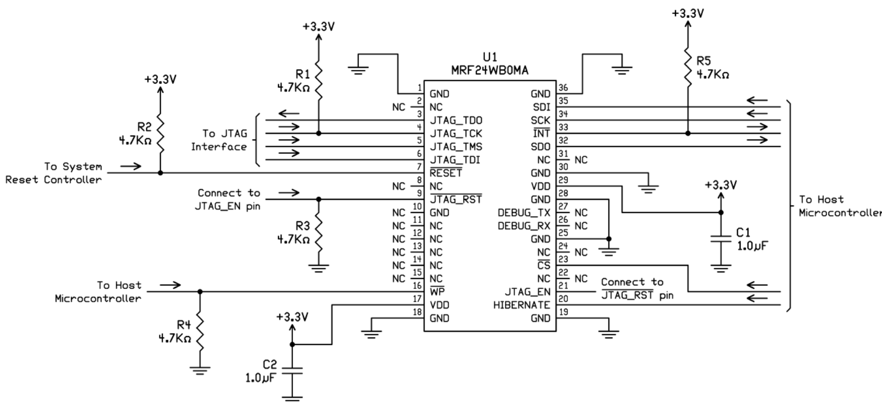

Figure 1: Reference design of the MRF24WB0MA.
Reprinted with permission from Microchip.com.

Wi-Fi, as a wireless communication for the quad-copter, is a good solution but the price puts this solution is out of the budget. The MRF24WB0MA IC can achieve a high range of 400m. The high range that this IC can achieve is a reason to consider overriding the budget. In addition to financial considerations, the extra range of the MRF24 may not be needed assuming that the quad-copter's autonomous systems become dependable.

2.6. 2 ZigBee

ZigBee is a protocol used normally on sensors networks. This protocol uses a mesh networking topology, where each device can talk to each other without a central routing device. ZigBee is based on the IEEE committee standard

802.15.4. ZigBee is considered a PWAN (Personal Wireless Area Network.) ZigBee can work in the ISM bands of 915 MHz and 2.4 GHz. The maximum data rate that ZigBee can achieve is 250 Kbits/s. ZigBee can operate at different maximum distances depending on the environment and components used. A Device that uses ZigBee can go from sleep to active in less than 16 msec. This makes ZigBee able to achieve a latency of less than 100 msec.

ZigBee meets all the design specifications of the quad-copter. In fact ZigBee is the perfect protocol for the quad-copter design. It goes beyond the needed data, and it meets the required range of 100 meters. On top of all of that ZigBee was designed to be a low power and low cost solution.

A ZigBee solution can be implemented using two options. Option one would be to buy a prepackaged system, where all that is needed is to plug and play. The second option is to buy an 802.15.4 compatible transceiver and build an in house custom ZigBee solution. Option one is the easiest but would cost more than option 2. Option two would be hard and changeling and it would be cheaper than option one. Both option were analyze and compared

They are many prepackaged ZigBee solutions to choose from. The most popular one is the XBee module. The XBee module that was discussed is the XBee 1mW Chip Antenna. This module can reach a data rate of 250 kbps. It can also achieve a range of 100 m. It also comes with 6 10 bit analog to digital converters and 8 digital IO pins. The XBee 1mW Chip Antenna costs $22.95 per module. This brings the wireless communication system to a total of $45.9 and connectors costs. At this price the XBee 1mW Chip Antenna is well below the specify budget. This makes it an ideal solution for the quad-copter. Unfortunately this is a senior design class and they wouldn't be much of a design choosing the XBee 1mW Chip Antenna as a solution. For that reason a better solution was an in house custom solution.

An in house ZigBee solution would require a transceiver compatible with the IEEE 802.15.4 standard and a ZigBee protocol software stack. They are many transceivers that meet these criteria. There are many manufacturers that make transceivers for ZigBee, such as Freescale Semiconductor, Texas Instruments, and Microchip. The ZigBee protocol stack is provided by theses manufacturers for their line of microcontrollers.

Freescale Semiconductor's ZigBee transceiver is the MC13202. Freescale provides a ZigBee protocol stack called BeeStack. The MC13202 can achieve a maximum data rate of 250 Kbps. It has a typical programmable output power of -27 dBm to +3 dBm. The output power of the XBee is 0 dBm, from this it can be inferred that the MC13202 can achieve a range of 100 m. The MC13202 comes in a QFN-32 package. Its physical dimensions are 5 by 5 mm. The sensitivity of the radio is <-92 dBm (typical).

Texas Instruments' ZigBee transceiver is the CC2520. Texas Instruments provides a ZigBee protocol stack. The CC2520 can achieve a maximum data rate of 250 Kbps. It has a typical programmable output power of up to +5 dBm. Just like the MC13202 this transceiver should be able to achieve a range of 100 m. The sensitivity that the radio can achieve is -98 dBm. The CC2520 comes in a QFN-28 package. Its dimensions are 5 by 5 mm. Microchip's ZigBee transceiver is the MRF24J40. Microchip provides a ZigBee protocol stack. The MRF24J40 can achieve a maximum data rate of 250 Kbps. It has typical output of +0 dBm. Just like the MC13202 and CC2520 this transceiver should be able to achieve a range of 100 m. The radio has a sensitivity of -95 dBm. The MRF24J40 comes in a QFN-40 package. This transceiver's physical dimensions are 6 by 6 mm.

All three of these transceivers meet the design requirement of the quad-copter. Texas Instruments' CC2520 has the best radio sensitivity, and output power. The MC13202 has the programmable output power range in case power supply becomes a concern. Both the CC2520 and MC13202 have a smaller physical size compared to the MRF24J40. All these quantities are summarized in the table 10.

Transceiver	Output Power	Data Rate	Sensitivity	Dimensions
MRF24J40	+0 dBm	250 Kbps	-95 dBm	6 by 6 mm
CC2520	Up to +5 dBm	250 Kbps	-98 dBm	5 by 5 mm
MC13202	-27 to +3 dBm	250 Kbps	<-92 dBm	5 by 5 mm

Table 10: Specifications of various transceivers.

From this table the best choice to implement a ZigBee wireless communication solution would be Texas Instruments' CC2520. The CC2520 beats the other two transceivers in terms of sensitivity and output power. These two quantities will affect the reliability and range of the wireless signal.

The last remaining factor in deciding which transceiver to use is the price. A CC2520 transceiver costs $6.08 and the development kit costs $112.75. The development kit doesn't have to be purchase since all of these transceivers are programmed through a microcontroller. A MC13202 transceiver costs $5.06 and the development kit costs $326.31. The price of a MRF24J40 transceiver is $3.88 and the development kit costs 269.99. If the cost of the development kit is ignored then the cheapest solution would be the Microchip MRF24J40. Using any of these transceivers as the basis for a ZigBee would be way below budget. In fact the price of this is below the prepackaged XBee 1mW Chip Antenna. If ZigBee is chosen as the preferred wireless communication solution then the CC2520 would be the preferred transceiver.

2.6. 3 Bluetooth

Bluetooth is a communication standard specifically designed to replace wires between devices in short range of each other. This communication standard is known for the wireless headsets used with cell phones. It replaces the need to have a wired connection between the cell and the headset. Although Bluetooth is known for short range communications, a range of 10 meters, it can achieve a range of 100 meters when using a class 1 Bluetooth radio. This specification meets the quad-copter's requirement range of 100 meters. The Bluetooth standard specifies that it can reach data rates of 1 Mbps for version 1.2, 3 Mbps for version 2.0 + EDR, and 24 Mbps for version 3.0 + HS. These data rates are beyond what is needed for the quad-copter. With these high data rates Bluetooth meets the quad-copter specification for data rate. A Bluetooth device can take up 2.56 seconds to go from standby to full connection. This is slow compared to ZigBee. This is the time that will potentially have to be waited before the remote controller and the quad-copter can communicate. This is not a big concern as long as the connection can be kept alive during flight time. Bluetooth devices can be put on HOLD and resume wireless communication almost instantly. Bluetooth have designation for a master device and a slave device; communication between the master and the slave can take a little time as 625 μs. Bluetooth meets the design specification of a latency of 100 ms or less.

Bluetooth meets the following required specifications of the quad-copter data rate, latency, and range. The last specification is the price of a Bluetooth solution for the project’s wireless communication needs. In order to determine the price 2 approaches to the solution will be looked at, just like the ZigBee solutions. The first approach is to buy a prepackage solution is the easiest approach but the most costly. The second approach is to design a Bluetooth system using a transceiver IC. This approach is harder and challenging but it also cheaper than the first approach.

The device selected for the prepackaged solution is the Roving Networks RN-41. The RN-41 is a fully qualified Bluetooth 2.1 device. It can provide data rates from 1200 bps and up to 921 kbps. This device is a class 1 Bluetooth radio, which means that it can reach a range of 100 meters. It has a transmitting output power of 12 dBm and it also has a typical receive sensitivity of -80 dBm. All of these parameters meet the quad-copter design specifications. The device cost $29.95; and the quad-copter needs two of these, one for the quad-copter itself and another for the remote controller. The total cost would be 59.90 and the costs of resistors, capacitors, inductors and wires. The RN-41 meets the budget requirements of the project. The high cost of this device comes from the added bonuses in addition to the communication system. The RN-41 comes with an mp3 codec and audio playing capability.

Roving Network’s RN-41 meets all the design specifications. If a prepackage solution were to chosen then the RN-41 is the best choice. The RN-41 also has two choices on connectors. It can be controlled through UART or USB. This flexibility will help on the decision of which microcontroller to use for the quad-copter.

A custom in house solution would require an IC that can meet the quad-copter design specifications. The manufacturers that were discussed are Panasonic and CRS. The CRS Bluetooth chip is the BlueCore4-ext, and the Panasonic is the PAN1315. BlueCore4-ext is a Bluetooth v2.0 device. It supports data speeds of 2 Mbps or 3 Mbps. The chip can transmit at an output power of +6 dBm. It comes with external 8 Mb of Flash and with 48 KB of internal RAM. The Bluetooth protocol software stack is provided by CSR. It has support for Class 2 and Class 3 Bluetooth radio. It can comes in two packages, 96-ball TFBGA with a physical size of 8 x 8 x 1.2mm and a pitch of 0.65mm, 96-ball VFBGA with a physical size of 6 x 6 x 1mm and a pitch of 0.5mm.The BlueCore4-ext chip meets all the design specifications of the quad-copter. The BlueCore4-ext comes with multiple choices for interfacing it with a microcontroller. These interfaces are UART, I^2C, and USB 1.1. This kind of flexibility is needed such in case the microcontroller chosen doesn't have enough of 1 bus.

The BluCore4-ext can be used a simple circuit that consists of capacitors, resistor, and inductors. An Antenna will have to be designed in order to meet the specification of 100m of range. CRS provides a sample circuit using a 50-ohm antenna. The price for one BlueCore4-ext is $5.91 with the resistors, capacitors and inductors it would get to around $8. A wireless communication solution using the BlueCore4-ext would cost about $16 which is below the budget. This device meets all the design specifications of the quad-copter. The reference design of the BluCore4-ext is shown in the Appendix.

Stephen R. Smith 12/12/10 9:43 PM
Comment: Appendix

The next device to be considered for the custom wireless communication solution is the Panasonic PAN1315. The PAN1315 is a Bluetooth v2.1 compatible device. It has a high sensitivity of -92 dBm and a transmission power of 10 dBm. It supports full Bluetooth data rates of up to 2187 Kbps asymmetric. This device has several interfaces through which it can be controlled. These interfaces are UART, I^2C, PCM, and SPI. With this amount of interfaces available any microcontroller can be used. Unfortunately this device only supports a Class 2 radio. This limits the range to a mere 10 m. The design specification requires a range of 100 m. Because of its range the PAN1315 can't be used in the quad-copter.

The next device that was considered for the custom wireless communication solution is the Panasonic PAN1315. The PAN1315 is a Bluetooth v2.1 compatible device. It has a high sensitivity of -92 dBm and a transmission power of 10 dBm. It supports full Bluetooth data rates of up to 2187 Kbps asymmetric. This device has several interfaces through which it can be controlled. These interfaces are UART, I^2C, PCM, and SPI. With this amount of interfaces available any microcontroller can be used. Unfortunately this device only supports a Class 2 radio. This limits the range to a mere 10 m. The design specification requires a range of 100 m. Because of its range the PAN1315 can't be used in the quad-Copter.

2.6. 4 6LoWPAN

6LoWPAN is a low power wireless personal area networks protocol. ZigBee is also a LOWPAN protocol. The difference between the 2 is that 6LoWPAN took IPv6 and made it work over the IEEE standard 802.15.4. This gives 6LoWPAN a distinct advantage of being able to communicate with almost any network that uses IPv6. But 6LoWPAN has a disadvantage is that ZigBee is mature and has a lot of protocols in its arsenal. Since 6LoWPAN and ZigBee both use IEEE 802.15.4 as a physical layer standard any transceiver IC that supports IEEE 802.15.4 can used for both.

The same research that was done with the transceivers for ZigBee applies to 6LoWPAN. This means that any transceiver that is chosen for ZigBee and meets the design specification will be chosen for 6LoWPAN. The same data rate, range, and latency that the TI CC2520 provides for ZigBee it will provide for 6LoWPAN. So 6LoWPAN meets all the design specifications of the quad-copter.

Just like ZigBee, 6LoWPAN requires a software stack to be implemented in a microcontroller. They are very few choices as to 6LoWPAN software protocol stack source code. Some manufacturers of transceivers IC, just like for ZigBee, provide the stack for free. If a suitable protocol stack is not found for the TI CC250 then another transceiver will be chosen.

2.6. 5 Z–Wave

Z-Wave is another low power, low data rate wireless network protocol. Unlike ZigBee and 6loWPAN, Z-Wave was designed with extremely low data rate in mind. The extremely low data rate makes it perfect for home automation when a light switch only needs to be told on or off. Z-Wave works on the 900 MHz ISM Band, in the US is 908.42 MHz It can achieve data rates of 9.6 Kbps and up to 40 Kbps. In the outdoor, open air conditions, Z-Wave can achieve a range of approximately of 30 m. In home automation the latency doesn't have to be low, which might mean that Z-Wave have a high latency.

Although Z-Wave can be used for turning you lights or air conditioner on and off it will not work for the quad-copter. Z-Wave doesn't meet any of the design specifications. The data rate is shy of the 56 Kbps minimum, and the range is way below the required 100 m. For this reason Z-Wave can't be chosen as the wireless communications system for the quad-copter.

2.6. 6 Dash7

Dash7 is a low power wireless communication protocol. It was designed with having a group of sensors in a mesh networking environment. This is how ZigBee and 6LoWPAN were design too. Dash7 can achieve a data rate of up to 250 Kbps. This data rate meets the required minimum of 56 Kbps. The latency of a Dash7 network is low. The most impressive stat of Dash7 is that it has a range of 2 km. This is way beyond the required range of 100 m. It can reach this kind of range because it works on the ISM frequency of 433 MHz This frequency also allows it to penetrate walls and other objects. In the case of the quad-copter, and aerial robot, the more range that can be achieved the better. Dash7 meets all the required design specifications, except price.

In order to get to a solution with Dash7, a custom solution will have to be developed. The transceiver IC that can provide a Dash7 wireless communication solution is the TI CC430F6137. The CC430F6137 is a system-on-chip solution. It comes with a MSP430 as the microcontroller and a CC1101 as the transceiver. The CC1101 transceiver works on the sub 1 GHz band of the ISM bands. The cc430f6137 can achieve a data rate of up to 250 Kbps. The transceiver has a sensitivity, at 315 MHz, of –117 dBm at 0.6 kbaud, and -111 dBm at 1.2 kbaud with a 1% packet error rate. The transmitter output power is programmable up to +12 dBm. The chip cost $13.37 and the development board costs $169.68.

A Dash7 protocol software stack is needed to complete the solution. TI doesn't provide a software stack for Dash7, but there is open source ones available. A complete circuit is provided by TI in the Data Sheet. The chip supports multiple and a different circuit is needed for the desired frequency. TI provides a reference circuit in the Data Sheet that can be used to determine the circuit needed for 433 MHz The reference circuit is shown in the appendix.

Stephen R. Smith 12/12/10 9:43 PM
Comment: appendix

Unfortunately, the TI cc430f6137 was not in stock at the moment of this writting. The part has a 15 weeks lead time which would mean the part would get here toward the end of senior design 2. For this reason TI cc430f6137 is not a choice for the wireless communication system of the quad-copter.

2.6. 7 Custom System

A custom system, for the wireless communication needs of the quad-copter, was researched. Although they are many systems on a chip and standard protocol to which to choose from, a custom can be tailored perfectly for the Quad-Copter. In the quad-copter design there are only two devices that need to communicate with each other. A complex protocol like ZigBee or 6LoWPAN is not really needed. What is needed is a simple point to point system with minimal overhead. This will free up the microcontroller to do more important tasks such the autonomous systems.

Creating a custom solution for the quad-copter wireless communication system will be challenging. In this research section different approaches were discussed in order to achieve a custom wireless communication system. This system needs to meet the design specifications of the quad-copter.

The transceivers that could potentially be used will be the same transceivers talked about in the ZigBee section. The reason for this is that a physical MAC layer is already developed for those devices. With the MAC and physical layer being taking care of, the focus was be more on the application layer and networking topology.

There a couple of networking topology that can be chosen from, such as Ring, Mesh, Star, Line, Bus, Point to Point. All of these networking topologies have disadvantage and advantages to the quad-copter For a Ring topology will go down if one device disconnect from the network. A star rely on the central hub not to fail, if it does the whole network goes down. In the Line topology, if a device goes down then the network will be separated into two parts. The Mesh topology solves the problem of the network going down by connecting a device to more than one device in the network. But this solution can become costly. Point to Point topology is link between only two devices. This is the simplest network topology. The Quad-Copter has only two devices that communicate with each other. This makes a point to point link the best solution to use as it requires less programming overhead than the rest of the networking topologies. In order to a successful wireless communication the two needs to communicate with the same language.

In order for two devices to able to communicate across any media they need a common language. This is called a protocol. For the quad-copter a simple protocol can be used. The protocol needs to be small but provide a reliable communication. All the transceivers manufacturers discussed in this paper provide a simple protocol on top of the IEEE 802.15.4 MAC and physical layer. But the protocol only works with their microcontrollers. The Quad-Copter needs to able to change microcontrollers if a company with the best transceiver doesn't have the best microcontroller for the quad-copter. For this reason a design for a custom simple protocol will be researched.

A protocol needs to provide a common language so that two devices can communicate with each other reliably. In the case of the quad-copter the devices are the quad-copter and the remote controller. The language, protocol, was researched in this paper will be as small as possible. This protocol will define how the data to be transmitted will be framed in a data package. They are an infinite ways a packet can be framed. The Wi-Fi protocol uses a long and complicated frame. Such frame is not suitable for the quad-copter. The data framing of the quad-copter wireless communication protocol will be divided in 4 parts. The first part is going to be the bit pattern that tells the device that a new packet is starting. The second part is going to be the checksum. The third part is going to be data to being transmitted. And the last part will be the bit pattern that the packet has ended. The total packet size is going to be 64 bits long. Each part of the packet is going to be 16 bit long. The reason for this is none other than the wireless communication system most likely will be using its own microcontroller, which is a 16 bit microcontroller. The quad-copter will use its own controller which will be 8 bit. Each part has to be coded in way that it won't be confused for a control signal for another part. The 16 bit microcontroller will provide a lot of room for control instructions status messages. Since the quad-copter will use an 8 bit microcontroller the lower bits of the 16 bit microcontroller will be used for controls instructions and status messages. Upper bits can use for the protocol between the two transceivers. The start and end signal of a packet will have a bit pattern that does not fall on the control instructions of the quad-copter. The bit pattern to use for the beginning and end of a packet is entirely arbitrarily. With that in mind the start of a packet will be the following bit pattern 1000000011111111. The end of a packet will be 1000000011110000. For control instructions and status messages the lower 8 bit will be used with trialing zeros, 00000000XXXXXXXX. The bit patterns 0000000000000000 and 0000000011111111 will not be used, which gives the quad-copter a total of 254 commands for control instructions and status messages. The packet might get corrupt as it travels through air, to check for this a system for error detecting will be researched.

Wireless communications is not perfect and some data will get corrupted. To combat this imperfection a few tools are available. The easiest and with less processing power needed is to retransmit the data a number of times. Eventually the correct data will get there. This type of solution is not good because the possibility of the right data never getting there. And also there is no way of checking if the right data got there. A different solution is to use what is called error correcting or error detecting code. Error correcting codes have to general sub groups the automatic repeat request, ARQ, method and forward error correction, FEC. The forward error correction method has the advantage that the data doesn't have to be retransmitted. But the processing power needed to encode and decode FEC is beyond the microcontroller used for the wireless communication system. Because of the limitation on the processing power, FEC codes will not be used. Automatic repeat request is not an error correcting code in the sense of FEC codes. Instead ARQ combines the retransmit method with a checksum or a cyclic redundancy check, CRC. The ARQ method work by checking the checksum or CRC that was sent against one that is done on the data. If the CRCs or checksums doesn't match a request to send the data again is sent. This method doesn't require as much processing power as using FEC codes, and provides a way to check if the data sent is correct.

In order to use the ARQ method a choice between CRC and checksum needs to be made. Checksum are easy to calculate but are prone to errors, because it doesn't take into account data position. If two bytes are reversed then a checksum would fail. CRCs are not prone to this kind of errors so CRC will the choice for the error detecting code for the ARQ method.

The choice for CRC algorithm is going based on whether we want to do a bitwise CRC or use a look up table. Using a bitwise CRC algorithm will minimize the memory footprint of the algorithm, but it will increase the processing time needed. In contrast using a look up table will take more memory but the processing time is will be minimize. The rest of the protocol will be how each transceiver communicates with each other and negotiate who talk and who listens. Since the ARQ method is going to be used in order to make sure the data is good, the rest of the protocol will be based the RS232 standard. Two communication commands, control instructions or status messages, have already been set. Now we need other commands that will mimic the RS232 standard. First the ARQ command will be sent on its own, and not in a packet. In fact all communications command will be sent on its own except when data is being sent. The ARQ command will be 0100000000001111. The others to be implemented are CTS, clear to send, and RTS, request to send. CTS bit pattern will be 010000001001; and RTS bit pattern will be 0100000001011.

The custom wireless communication protocol is simple, and it mimics the RS232 standard. If a device desire to talk a RTS signal is transmitted. IF a CTS is sent back whatever data is to be transmitted will be sent. If not then the device will wait for a period of time and try the RTS signal again. When the data is sent and an ARQ is sent back the same data will be retransmitted. And the process starts over again.

The custom wireless system is the best system for the quad-copter. The price of the custom wireless system is below the required $70. The data is above the required 50 Kbps. Since we are using the same transceivers as the ZigBee protocol the latency requirement of 100 ms is also met. The only this holding this system back is the time that is need to design, program, and test the system. With no experience in protocol designing or software implementation of a protocol a better solution would be to choose a simple protocol that the manufacturer of the transceiver provides.

2.6. 8 Antenna

There are two types of antenna that were researched for the use of the quad-copter. One of the two antennas is the dipole antenna; the second is a patch antenna. The patch antenna, or micro strip antenna, is a piece of flat metal put on top of a ground plane. The patch antenna is sensitive to the thickness and the dielectric constant of the PCB, for this reason the quad-copter will not use a patch antenna. The dipole antenna is an antenna made out of two quarter-wave antennas put together. Unlike the patch antenna the dipole antenna is not sensitive to the thickness or dielectric constant of the PCB.

The Quad-copter needed an antenna that can operate at the frequency of 2.4 GHz. The antenna has to have an impendence of 50 ohms, and a gain greater than 0 dBi. The antenna that was chosen was a 2.4 GHz antenna with a gain of 5 dBi and a reverse polarity SMA connector.

2.7 Sensors

2.7.1 Flight stability sensors

In the interest of flight stability, or achieving level controlled flight, a combination of sensors was required to continually monitor the roll and pitch of the quad-copter such that the microcontroller could process the data and effect real time corrections. The pitch is a measurement of the nose of the quad-copter pointing either upwards, positive pitch, or downwards, negative pitch. The roll is a measurement of the rotation around the longitudinal axis of the quad-copter with the right or starboard side down being a positive roll. The yaw is a measurement of the rotation around the vertical axis which will be discussed separately under the section "Direction/Yaw sensors" elsewhere in this paper. These parameters are illustrated below in Figure 2 for clarity.

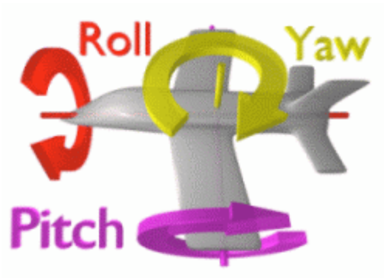

Figure 2: Visual representation of roll, pitch, and yaw.
Reprinted under Wikipedia commons license, created by ZeroOne

The system, consisting of a combination of sensors that detect pitch and roll for the purpose of maintaining level flight, needed to satisfy the following goals and objectives:

- The system should be robust enough to collect roll and pitch data of the quad-copter at a frequency that facilitates real time flight correction.
- The system selected should be advanced enough so as not to constrain data utilization methods. Meaning more rather than less capacity.
- The system should utilize a simple data transfer subsystem that demands minimal MCU processing.
- The system should preferably output an analog signal.
- The system should be compact in size.

Furthermore, the flight stability system should have met the following specifications and requirements:

- The system should relay the measurement data to the microcontroller at a minimum rate of 10 Hz.
- The system should weigh less than 20 g (total) .
- The system should cost less than $100 (total).
- The system should cost less than $100 (total).

Inertial Measurement Unit (IMU)

Research indicated that a combination of accelerometers and gyroscopes is a common approach used by hobbyists to measure and stabilize the flight of RC planes, helicopters and quad-copters. This combination of different sensors working together to establish an accurate orientation measurement relative to the ground can referred to as an inertial measurement unit (IMU). IMU's are available to hobbyists that vary in the degree of complexity according to the number of axes that are measured by various sensors. For example, a 3 degrees of freedom IMU combo board might measure 3 axes with one type of sensor whereas a 9 degrees of freedom IMU would measure each axis with three different sensor types, the third in this example perhaps being a magnetometer. Pertaining to monitoring the tilt of the quad-copter it is necessary that, at a minimum, we measure the rotation about the X and Y horizontal axes, or 2 degrees of freedom. These axes could be measured either with gyroscopes, accelerometers or a combination of both. A combination of accelerometers and gyroscopes measuring the same axis may seem redundant but, the methods of the two sensors differ and thus a more complete picture can be relayed to the microcontroller for analysis. For these reasons a 5 degrees of freedom IMU was be integrated into the design of the quad-copter. This entailed a triple axis accelerometer and a dual axis gyroscope measuring the X and Y horizontal axes. With this configuration each of the 2 critical horizontal axis will be monitored by 2 different sensor types. Sparkfun offers such an IMU as a complete unit on a breakout board along with supporting components for $74.95. While this IMU unit would meet the requirements for the quad-copter, The group has opted to purchase the components separately and integrate them into the main circuit board. This approach preferable for a few reasons: the first is cost, it is about half the price to buy separately, secondly weight, by integrating the components onto the main board the added weight of the breakout PCB is avoided, and thirdly design experience, designing a PCB layout provides the design experience that fulfills the purpose behind building the senior design project. Accordingly, the IMU components were researched separately.

Accelerometer

An accelerometer measures proper acceleration relative to free-fall along a single axis. Two and three axis accelerometers are also available that for a negligible cost, greatly expand the component's functionality. Theoretically, a two axis (X and Y axes) accelerometer would be sufficient to measure the roll and pitch of the quad-copter however, a third, Z axis, would weigh about the same and provide more data. Actually, due in part to the costs associated with mass production, a triple axis accelerometer can cost *less* than a dual axis accelerometer. Such is the case at the Sparkfun electronics website where the ADXL335 triple axis accelerometer costs $9.95 and the comparable ADXL320 dual axis accelerometer costs $13.95. The ADXL335 has a full-scale range of +/- 3 g which means that while it is quite accurate at measuring acceleration within that limited range, the quad-copter would be somewhat constrained in the types of maneuvers that it could perform. As the quad-copter is not intended for any sort of aerial acrobatics, it is concluded that a range +/- 3 g will maximize sensitivity without sacrificing any functionality. The ADXL335 has a radiometric analog output and a supply voltage range of 2.7 V to 3.3 V which corresponds to system wide power requirements. The output bandwidth is fully suitable to specifications as it can be selected within a range of .05 Hz to as high as 550 Hz or even higher for the X and Y axes. At 4 mm X 4 mm X 1.45 mm and weighing only 2 grams the ADXL335 further fulfills component objectives.

Gyroscope

At first glance it may seem that accelerometers alone might be sufficient to monitor the X and Y axes of the quad-copter however, further research indicated that accelerometers have a tendency to measure more than just gravity. Accelerometers also are influenced by vibration and centripetal acceleration. As the quad-copter was not intended to achieve fast lateral movement or banking turns, like an airplane, the effects of centripetal acceleration on the accelerometer could be minimal relative to other applications. Vibration, on the other hand, can be a serious factor affecting the accelerometers owing to the four engines rotating at differing speeds and connected by a lightweight carbon fiber frame. For these reasons a dual axis gyroscope was integrated into the quad-copter design to measure rotation about the X and Y axes. The IDG-500, integrated dual-axis Gyro manufactured by InvenSense fulfills the pertinent objectives in this area. The IDG-500 is the same chip that is used in the pre constructed 5 degrees of freedom IMU offered by Sparkfun so, it has an established history of compatibility. A closer look at the IDG-500 reveals that for $24.95 it is a particularly sophisticated angular rate sensor with integrated electronic systems that reduce vibration rejection and largely eliminate the need for external components. The IDG-500 requires a steady supply voltage between 2.7V and 3.3V which is acceptable while low compared to other quad-copter sensors. The device is not radiometric meaning that the gyro outputs are independent of supply voltage. Because of this, it should not be a problem to supply this device at 3 V, to avoid the maximum range, in conjunction with another analog devise supplied by a different voltage such as with the accelerometer being supplied at 3.3 V. The component also has two outputs for both the X and Y axes representing two different ranges and corresponding sensitivities: one range scale that measures +/- 500 degrees/ s, the other range scale of +/- 110 degrees/ s is for slower angular acceleration such as would be more appropriate for the quad-copter. The IDG-500 is pictured below in figure 3 in order to convey a sense of its size and the degree of precision required by the SMT process to correctly mount this chip. Also, a simple schematic is included that illustrates its functionality and polarity of rotation.

Figure 3: Integrated Dual-Axis Gyro (IDG-500) from InvenSense. Reprinted with permission from Sparkfun and InvenSense

Magnetometer

An alternative approach to gyroscopes for orientation sensing would be to use a three axis magnetometer in conjunction with accelerometers. Magnetometers detect and measure the surrounding magnetic field to produce a heading vector along one or more axes but, are subject to interference from local fields. Magnetometers offer an advantage over gyroscopes in that their heading is updated continually and so don't suffer from gyroscopic drift, an accumulation of error over time. The strength of a system which incorporates magnetometers is in the accuracy of the computed heading but, not the accuracy of the roll detection at all. A magnetometer with accelerometer tilt compensation seems more suitable for a high speed fixed wing project than for the quad-copter design. As the quad-copter will be slow moving it was thought that there would be ample opportunity for heading corrections however, the group learned that a higher heading update frequency would have better stabilized the yaw of the quad-copter. Tilt detection and correction is more in accordance with the goals and objectives of the flight stability system then the detection of heading.

Infrared horizon sensing

Another option for level sensing was to use infrared horizon sensing technology. This approach to level sensing is based on the difference in temperature at the horizon where the sky is measurably cooler than the ground, and this difference can be detected using thermopiles. Figure 4 below more clearly illustrates the horizon sensing system and shows the $99 AttoPilot XYZ Horizon Sensor as available from DIY Drones.

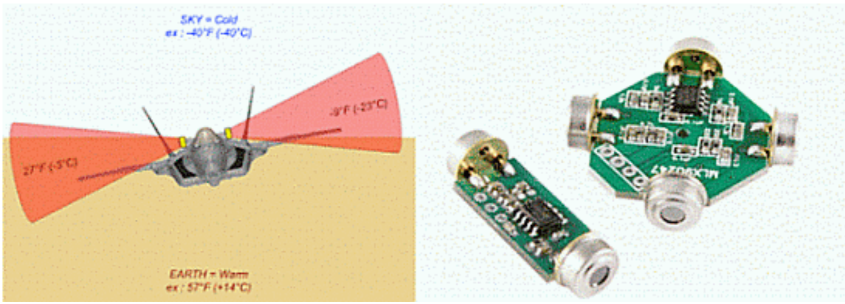

Figure 4: schematic and AttoPilot XYZ Horizon Sensor.
Reprinted from Paparazzi.enac.fr/ under the GNU free documentation license.

The problem with IR horizon sensing in relation to the quad-copter is that in addition to the price, the system is significantly less effective at lower altitudes, and that effectiveness would have been further compromised by nearby buildings. Additionally, IR horizon sensing would have been useless indoors. For these reasons, it was concluded that IR horizon sensing was not the correct sensor system for the quad-copter.

2.7.2 Distance and ground sensors

The quad-copter needed sensors in order to maintain a constant hovering altitude and to avoid collisions with either the ground or with other obstacles laterally. The downward oriented ground sensor should be capable of providing reliable and accurate measurements to the ground for low altitudes, while one or more laterally oriented distance sensors was needed for the purpose of object detection and avoidance. The lateral sensors were intended to cover an area forward of the quad copter to prevent collisions. A wider arc of detection with multiple sensors for the purpose of facilitating more advanced autonomous protocols was also considered. The sensors used for both ground and object detection may be of the same type or a combination of different types. The distance sensors chosen should have accomplished the following goals and objectives:

- The sensors should perform reliably regarding the detection of a wide range of surfaces, densities, colors, and temperatures.
- The sensors should perform reliably under a wide range of conditions including humidity, temperature, and noise levels.
- The sensor should preferably output an analog signal for ease of use.

Furthermore, the sensors should have conformed to the following specifications and requirements:

- The sensors should detect the distance to the ground within a range of 1 to 15 feet (.25 to 4.5 meters) .
- The sensor(s) should detect object distance forward of the quad-copter, on the same plane, within a minimum arc of 30 degrees, and within a range of 1 to ten feet (.25 to 3 meters).
- Each sensor should weigh less than 5 grams.
- Each sensor should cost less than $50 .
- The sensors should offer a resolution accuracy of 6 in. or better.

Ultrasonic proximity sensors

Ultrasonic sensors work by emitting a sound burst in the ultrasonic range, above 20 kilohertz, and timing the return time of the sound wave. Ultrasonic sensors have some key advantages over other types of proximity sensors in regard to the quad-copter's requirements and thus, were the only type of distance sensor used on the quad-copter. However, ultrasonic sensors are not perfect, for instance there are variations in how different surface reflect sound waves depending upon their composition and/or their angle relative to the sensor, thus limiting the accuracy of ultrasonic sensors. Another weakness of ultrasonic sensors is noise or ghost echoes from ricocheting sound waves. Noise problems can quickly multiply when using multiple sensors unless counter measures are taken. The low end line of MaxBotix ultrasonic sensors, the LV-MaxSonar-EZ (EZ0 – EZ4), met the project's general requirements for distance sensors. The different models are essentially the same but, differ slightly in the width of their detection areas, with the EZ4 having a narrower beam than the EZ0. The EZ1 provides a good balance of beam width and detection area, and costs a little less than the others at $25.95 rather than $27.95, and was originally considered as a candidate for the lateral proximity sensors of the quad-copter prior to certain design reconsiderations. These reconsiderations being in essence that a single forward oriented narrow beam could suffice as a proximity sensor and a narrower beam would be subject to less interference. The MaxBotix EZ sensor is pictured below in figure 5 along with a graphic of the target detection angle of the EZ1. A similar detection angle graphic for the EZ2 is not available but can be extrapolated somewhat from the EZ1 graphic. The graphic gives an approximation of detection angle as a function of the type of object being detected. For example, a wall is detectable at a greater angle from center than a narrow rod. This angle of detection will be a factor in how the lateral sensor array is integrated into the design of the quad-copter.

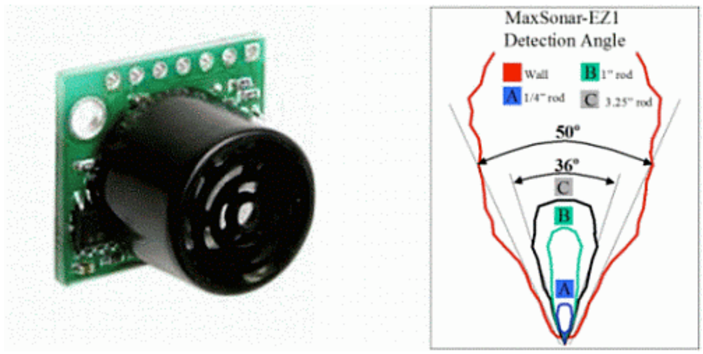

Figure 5: The LV-MaxSonar EZ and detection angle graph. Reprinted with permission from MaxBotix.

Either the EZ2, EZ3, or EZ4 with progressively narrower detection angles would provide a good fit for both ground detection and as a proximity sensor as a narrower beam translates into less noise, less ghost echoes, and thus a steadier signal. According to the MaxBotix website the EZ2 "offers a good compromise between small object detection and narrow beam width" ("Selection Guide" MaxBotix). The MaxSonar EZ sensors utilize a frequency of 42kHz and have an effective range from 6 to 254 inches (6.45 meters), which is well suited to the specifications pertaining to distance sensors. The sensors operate between 2.5 and 5.5 volts at a ballpark 2 mA and offer an output choice of analog, serial, and PWM signals, analog being preferred. The MaxSonar EZ sensors have a satisfactory resolution of 1 inch increments, which should have been more than adequate for their purpose.

Infrared distance sensors

Infrared proximity sensors work by emitting a beam of infrared light and then evaluating the returning light. Methods of evaluating the returning signals include triangulation, which considers the return angle, and modulation, which focuses on a specific modulation of the signal. Infrared range finding sensors such as Sharp's GP2Y0A02YK0F were the first type of proximity or distance sensor considered by the group. Infrared sensors are generally less expensive than ultrasonic yet, have a shorter range, for instance, the long range Sharp sensor only has an effective range of 15 to 150 cm. The main drawback, however, to using IR sensors for the quad-copter is that IR sensors are not effective outdoors due to the ambient infrared light. The modulation technique of sending and receiving infrared signals offers the best possibility for outdoor use but, even so, IR proximity and distance detection also varies according to the light reflecting capabilities of the surfaces involved. For instance materials of different colors and/or textures could significantly affect measurements. In sum, even with a sophisticated, modulated Infrared proximity sensor, if used outdoors, it would provide the quad-copter with an insufficient detection range and an unsatisfactory wide variability range due to surface characteristics.

Laser range finders

Laser range finders work by principles similar to ultrasonic sensors but use a laser pulse instead of a sound wave. Laser range finders permit a range and accuracy well beyond ultrasonic and infrared sensors but, are significantly more expensive. Laser range finders have many military applications such as target acquisition and speed detection. According to Wikipedia, handheld laser range finders operate at ranges of, "2 km up to 25 k" ("Laser…" Wikipedia). So, while laser range finding offers the best performance of proximity sensing technologies, it is largely beyond the means of many hobbyists and students due to cost considerations, at least for the time being. Hizook, the robotics news portal, reports that sub $100 laser range finders may soon become available for hobbyists due to manufacturing advances made by Neato Robotics in developing their automated vacuum robot called the Revo LDS.

2.7.3 Altitude measurement

Ideally, the quad-copter would have had a sensor to monitor higher altitudes, beyond the range of the ground distance sensor. The purpose of this component would have been be to ensure that the quad-copter stays under control of the user or under the control of autonomous systems as the case may be. In the case of the autonomous flight mode, distance sensors would have overridden altitude sensors in the case of a discrepancy. As the altitude sensor would have been be relied upon primarily when the quad-copter was beyond proximity detection range, the sensor could therefore be chosen without a great deal of emphasis on accuracy. An alternate design approach, which was ultimately used, was to to forgo a high altitude sensor altogether by programming protocols whereby the quad-copter would gently descend when beyond the range of the proximity sensors until an ultrasonic ground reading could be confirmed. However, weighing the low cost of an additional sensor versus the cost of a lost quad-copter, this alternate design strategy was not initially considered. The medium to high altitude sensor would have conformed to the following specifications and requirements:

- The sensor should be capable of measuring altitude within a range of 15 to 200 feet.
- The sensor should be precise to within 10 feet
- The sensor should be lightweight, under 5 grams.
- The sensor should not cost more than $50.

Barometric Altimeter

Barometric altimeters measure altitude by measuring air pressure, lower air pressure indicating a higher altitude. Altitude can be represented by the equation $z = cT \log(P/Po)$ where c is a constant, T is the absolute temperature, P is the pressure at altitude z, and P_o is the pressure at sea level ("Altimeter" Wikipedia). As this sensor will function to a degree as a backup or secondary system without strict accuracy requirement, the viability of the cheapest sensors available was considered. Sparkfun electronics offers Freescale Semiconductor's MPL115A1 barometric pressure sensor for $11.95. The sensor outputs a digital SPI signal measuring pressures from 115 kPa to 50 kPa, from below sea level to more than 15,000 feet with an accuracy of 1kPa, well within the relevant range and accuracy requirements. The only drawback to using this chip would have been that the SPI serial interface was unfamiliar to the group's programmers and would have required an investment in research that would only have applied to this one specific component of the quad-copter. Figure 6 below, shows the size and shape of the sensor which at 5 mm X 3 mm X 1.2 mm offers a compact solution that does not require additional supporting components.

Figure 6: digital output barometric altimeter and pin configuration. Reprinted with permission from Sparkfun.com.

Although Freescale Semiconductor's MPL115A1 fulfills the requirements pertaining to an altitude sensor for the quad-copter, a more cost effective solution has subsequently been found that additionally, utilizes the I2C serial interface rather than SPI. This other module, Hoperf electronics' HDPM01, is a combination of a barometric pressure sensor and a digital compass onto a single breakout board. The combination of both sensors onto a single unit provides a solution that both streamlines design and, at $19.90, is less than half the cost of purchasing the pressure sensor and digital compass separately. Furthermore, the pressure sensor on the HDPM01 has pressure accuracy to within 3hPa and so is slightly more sensitive than the aforementioned Freescale MPL115A1. Hoperf electronics' HDPM01 is pictured and discussed in greater detail under the heading of "compass module" elsewhere in the research section of this paper.

GPS Altitude Measurement

One possible approach to obtaining medium to high altitude measurements would have been to simply utilize the existing system within the GPS module that will be integrated into the design of the quad-copter. This would have been advantageous in terms of it being a cost free and weight free solution however, research indicated that, perhaps counter intuitively, the error of the vertical GPS measurement does not scale proportionately with the error of the horizontal components. In other words, there is too much error associated with the GPS vertical component to make it a reliable measurement for the quad-copter. The cause of this error is a combination of priorities and geometry. One expert states, "Generally, Altitude error is specified to be 1.5 x Horizontal error specification" (GPS Altitude, Mehaffey). However, according to several other sources the error can be much worse. The user forums at DIYdrones contained the comment by Jani Hirvinen, "Another important issue is that you should never trust on GPS altitudes due it might suddenly jump over 100 meters in matter of minutes. Yes some areas you can get really nice accurate altitude but then again some areas not. So, if you rely on GPS altitude, be careful" (GPS Altitude Error, Hirvinen). So, in consideration of the accumulated warnings of GPS altitude it was concluded that a low cost, lightweight barometric altimeter is a small price to pay to avoid the pitfalls of GPS altitude error.

2.7.4 Location/Navigation sensors

To implement the autonomous capabilities of the quad-copter it will require a system whereby its location can be accurately determined in relation to a map or to a starting point. There are basically two strategies for instilling an awareness of locality into a mobile robot: by using either a dead reckoning system, or a system with external references. A dead reckoning system ascertains its current location, or state, by computations based on a past state, the intervening speed, the intervening direction, or other such relevant parameters. An example of a practical dead reckoning system would be an automobile entering a parking garage that loses its GPS signal. Through computations based on the car's velocity, and manipulations of the steering wheel, an estimation of the car's position could be maintained. When the car eventually exits the parking garage, and the GPS signal is received, then the car's position would be updated by an external reference. The primary weakness associated with dead reckoning is that incremental errors accumulate both over time and relative to the amount of calculations thus, reducing positional accuracy. For these reasons, dead reckoning is generally more effective when combined with external references, as in the case of the accelerometers and gyroscopes in the quad-copter's flight stability system. The complexities and the degree of error that would accumulate in implementing a dead reckoning system for the quad-copter are formidable enough so as to reject the strategy in favor of an external reference system. A point, however, in favor of the dead reckoning system is that the limited flight time of the quad-copter would minimize the impact of error or drift. Nonetheless, having decided on an external reference system, the logical implementation involved the use of a Global Navigation Satellite System (GNSS), the United States based Global Positioning System (GPS) being the only available option of such a system. In the interest of completeness, or even curiosity, Long Range Navigation (LORAN), a terrestrial radio navigation system, was briefly examined but, this outdated system began being phased out as of February 2010. Presuming the use of GPS technology, the location/navigation system for the quad-copter should have achieved the following goals and objectives:

- The system should establish an external reference to position (latitude and longitude) and time.
- The system must have a serial output compatible with the MCU, UART preferred.
- The system should be compact and self-contained requiring minimal external support.
- Hot, warm and cold start times are not a priority, meaning that preflight setup time is not a factor, and that the time to re-lock on satellites is not prioritized.

Additionally, the location/navigation system selected should have met the following specifications and requirements:

- The system should be accurate to within 3 meters (latitude and longitude).
- The update rate should be at least 1Hz.
- The system should weigh less than 30g including antenna.
- The system should cost less than $100.

GPS module

Various GPS modules differ in their characteristics and capabilities such that it is left to determine priorities. Within the price range of under $100 the horizontal plane positional accuracy of GPS modules are clustered at roughly 3m. So, while accuracy is an important characteristic for the quad-copter, other characteristics must be evaluated to more finely tune the selection process. As the goals and objectives of the quad-copter do not include that it should move fast, this will have a bearing on the update rate at which the GPS module should operate. The slowest update rate generally available, 1Hz or once per second, should be more than enough to achieve the project's navigational requirements. This minimal update rate requirement was considered advantageous both in terms of cost and a slight increase in positional accuracy. Power requirements did not significantly affect selection criteria as almost all available modules fell within voltage and amperage ranges that could be accommodated for in the quad-copter design. Also, energy saving features such as the capability to automatically shut down were not options considered worth paying extra for. Weight of the module remained a significant consideration as is the case with just about any component of the quad-copter. Size of the module is not particularly critical however; a more compact devise was preferable. The questions of size and weight are related to the question of whether the GPS module has an antenna integrated into the receiver or a separate antenna. A particularly economical GPS receiver board was found on EBay that, although a hefty 17g, cost only $9.95 and met most project requirements. Although this module was almost bought, it was ultimately discarded on the grounds that it lacked an internal antenna. To buy an antenna separately would probably have still left an economical advantage over other choices however, the weight of the external active antenna plus the receiver board was approaching an undesirable 30 g plus. A further consideration in buying the antenna separately is that unless it is matched very precisely with the receiver, considerable signal degradation could result. Figure 7 below shows the Ublox RCB-LJ receiver board plus the antenna connection in the lower left hand corner that was almost bought for the quad-copter.

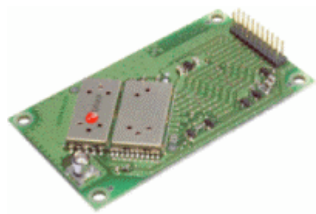

Figure 7: Ublox RCB-LJ receiver board; 71mm X 41mm.
Reprinted from the Ublox RCB-LJ data sheet under non-commercial use.

Not wanting to get bogged down by the antenna issue related to the Ublox board additional online research for an affordable GPS option was undertaken. A particularly good deal on a module offered by the DIY drones online store was eventually found. The $29.99 MediaTek 3329 is a POT (Patch On Top) module meaning that it has an antenna integrated into the ceramic case of the receiver. Comparable embedded antenna modules at Sparkfun and elsewhere were generally on offer in the $50 to $100 price range so; the MediaTek module represented a significant savings. Closer inspection of the module reveals that it had more advantages beyond cost. For example, the module only weighed 6 g which was a significant improvement over the Ublox solution that weighed 17 g plus the antenna weight. Also in the favor of the MediaTek module was that it is more recent technology than the Ublox board. This is evidenced by a greater sensitivity than the Ublox, measured at -165 dBmW versus -149 dBmW. Also, the MediaTek module offered more channel capacity at 66 versus Ublox's 16. An excess of channel capacity however, does not necessarily improve performance and according to one expert, “To some degree it is a marketing gimmick" ("Do Chipset…" GPSreview.net). The MediaTek module has a variable update rate with a maximum rate of 10Hz. The default rate of 1Hz was ideal for the quad-copter yet; DIY drones offered the firmware and support to implement adjustments if necessary. The MediaTek module required between 3.2V and 5V and a maximum of 50 mV. The module provides both a USB and UART interface, either of which was compatible with the project design and familiar to the group's programmers. Figure 8 below shows the MediaTek 3329 GPS receiver and accompanying pin configuration in order to emphasize both its compact size and minimal external support requirements. Note, the module is 16mm X 16mm X 6mm and it is a metric ruler shown.

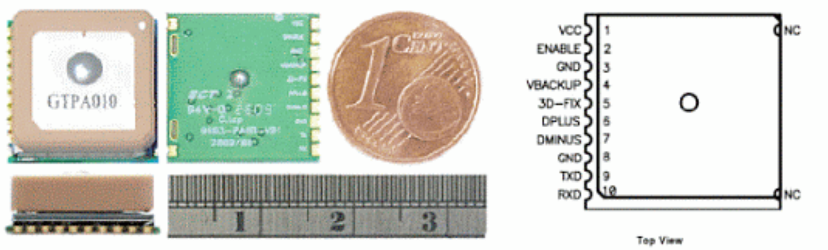

Figure 8: MediaTek 3329 GPS module with pin layout.
Reprinted with permission from DIYdrones.

2.7.5 Direction/Yaw sensor

The quad-copter required a sensor to measure its yaw, or direction. Either implementing a dead reckoning or an external reference strategy could have accomplished this. The dead reckoning strategy would have required a gyroscope that measured rotation around the vertical axis. In theory, the direction and distance to a location could have been plotted with reference to the starting state of the quad-copter. This approach however, was evaluated and dismissed in the research section of this paper pertaining to location and navigation sensors. The external reference strategy would have entailed the use of a magnetometer or electronic compass. Yet another approach would have been to utilize both methods simultaneously, thereby gaining the strengths of both systems. The gyroscope does not suffer from magnetic interference and could be used to stabilize the desired direction as determined from the externally referenced compass. While the combined approach has definite strengths its main drawbacks are price and simplicity of design. Having assumed that the digital compass module would yield reliable data and that the quad-copter would not suffer unduly from short-term directional instability, then the digital compass alone was considered sufficient as a directional sensor. An element considered in this matter was that the speed of the quad-copter was slow enough as to allow for directional error and compensation over time. An additional consideration in this matter was that a GPS targeted position may represent a disk as much as 6 m across. For these reasons, a degree of compass error was thought to be acceptable. In hindsight, a faster update rate proved to be desirable to steady the heading of the quad-copter. Goals and objectives of the externally referenced directional senor system are listed below:

- The module should establish an external reference to direction for the purpose of turning the quad-copter or plotting a course.
- The module should be reliable and not suffer from excessive magnetic interference.
- The module should be a separate subsystem so that it can be optimally placed away from interfering fields and metals.

Additionally, the system should have fell within the parameters of the following specifications and requirements:

- The module should achieve a minimal averaged accuracy of within 3 degrees.
- The module should weigh under 10g.
- The module should cost less than $40.

Compass module

A digital compass is a type of 2 axis magnetometer that determines direction by measuring the intensity the earth's magnetic field relative to its corresponding axes. The magnetometers actually measure the local magnetic field and so are subject to interference from local electrical fields and metals in the vicinity of the device. It therefore becomes desirable to optimize the placement of the digital compass on the quad-copter. The optimization of placement furthest from sources of interference necessitates finding a stand-alone device with its own dedicated PCB rather than integrate the device directly with the main circuit board. The Honeywell HMC6352 digital compass solution with breakout board, available from Sparkfun Electronics for $34.95, met the requirements for a directional sensor. The HMC6352 is advertised by Sparkfun as able to provide a "simple, clean, degree resolution compass heading" ("Compass" Sparkfun). The module is compatible with the power requirements of the quad-copter having a supply range of 2.7 V to 5.2 V at approximately 1 mV. Also, the module is equipped with an I2C serial interface which is compatible with the MCU. The module provides directional accuracy of .5 degrees, well within requirements. The module can be configured to output an update rate of between 1Hz and 20Hz, where, like the GPS update rate, It was thought mistakenly that the slow speed of the quad-copter would necessitate only a low update rate such as 1Hz. At 3cm X 3cm the breakout board is compact enough so as to facilitate optimal placement of the module on the quad-copter. Figure 9 below shows the size of the module, the pin layout of the breakout board and the associated schematic diagram.

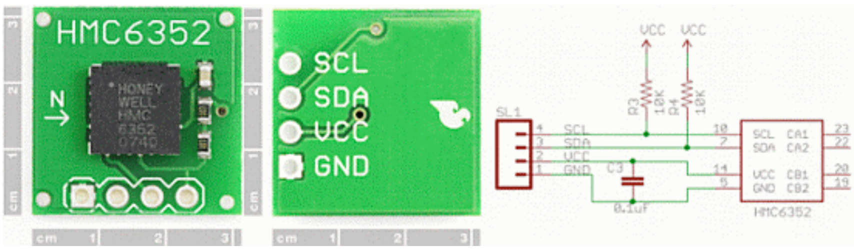

Figure 9: Honeywell HMC6352 digital compass, pin layout, and schematic Reprinted with permission from Sparkfun.com.

Significantly more expensive digital compasses could be found that integrated accelerometers into their design in order to compensate for tilting of the compass away from the horizontal plane. As those tilt sensing modules start at over $100, they were not been considered for use with the quad-copter. That the HMC6352 doesn't accurately measure direction when significantly tilted from the horizontal, was regulated to a design issue where this limitation must be compensated for. Cost was a significant factor in the component selection for the project and subsequent research into available compass modules uncovered a stronger potential candidate for a digital compass solution. The HDPM01 compass module, manufactured by the Hoperf electronic company, is offered by Futurlec electronics for only $19.90. This compass module is a little less sensitive than Honeywell's HMC6352 with a compass accuracy of 2 degrees rather than .5 degrees however, the HDPM01 is a combination module that incorporates a pressure sensor into the unit. The pressure sensor on the other hand, with an accuracy of +/- 3hPa, is slightly more sensitive than Freescale's MPL115A1 with +/- 1kPa accuracy. As the 2 degrees of accuracy of the compass module still lies within the requirements for the quad-copter, the HDPM01 offers comparable functionality to the other two units for less than half the cost. The HDPM01 operates with a supply voltage in the range 2.4V to 5.25V and with an amperage of .4mA or less which is compatible with project design. Furthermore, the module has an I2C serial interface which was more preferable to the group's programmers than the SPI interface of the Freescale MPL115A1. However, the I2C interface of this module may not be as user friendly as some other options. For example, the Hoperf pressure sensor requires the microcontroller to perform calculations on pressure and temperature values in order to get an altitude reading and, to complicate matters further, the accompanying data sheets seem poorly written and vague. Ultimately, the group choose a less cost efficient, but better documented component – the Honeywell HMC6352. The combination module is shown below in figure 10 along with a diagram of its pin configuration in order to illustrate the independent nature of the component.

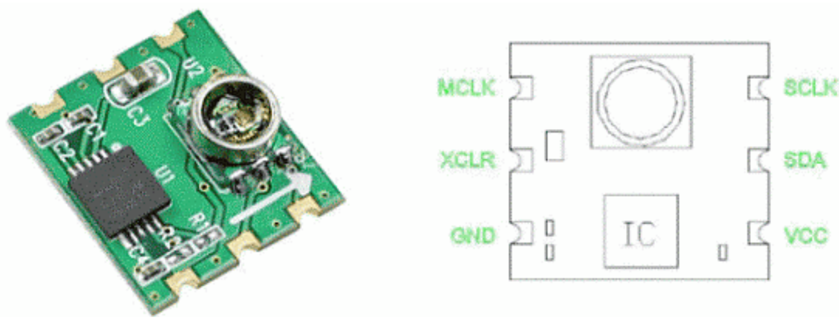

Figure 10: Hoperf electronics' HDPM01 pressure sensor and digital compass
Reprinted with permission from Futurlec.com.

The HDPM01 is similar to the Honeywell module in that it is operationally limited to the horizontal plane. Significant tilting of the quad-copter could interfere with the digital compass thus necessitating compensation in the design process. Such compensation could simply be that if it can't fly level, then the compass doesn't really matter.

2.8 Video System

The Video system of the quad-copter is going to be used to get a bird's eye view from the quad-copter the video system is one of the least critical subsystems of the quad-copter. The reason for this is that the video system will and should not affect the flying capabilities, and navigation capabilities of the quad-copter. It will not be used for obstacle avoidance or piloting. The only reason for this system is to have the ability to see from the quad-copter. The video system needs to meet the following specifications of the quad-copter.

- Weight no more than 0.5 Kg
- Have a range of 100 meters
- Consume less the 500 mA

In order to meet the design specifications different design methods will be researched. One of these methods is to get an image sensor and build everything around it. This method will be the most complicated challenging and time consuming. It has the potential of being the least expensive. Another method is to get a camcorder or webcam and just get the video from it and build the rest of the system around getting that data to a display wirelessly. This method is less challenging then the first method, but still quite complicated. If we modified this method to instead of sending the signal wirelessly the signal will be record it to onboard memory. This lessens the complexity of sending a video signal wirelessly which requires a lot of bandwidth. The last method is to simply buy a prepackage system that comes with a video camera, a video transmitter, and a video receiver. This is the less challenging, complicated but it might be the most expensive.

The first method will need an image sensor, and a microprocessor, or DSP chip, or video processor. It will also need a transceiver that has the bandwidth and data rate to send video wirelessly. If bandwidth is a problem the processing device needs to have enough power to format the video, compress it, and send it to the transceiver. The image senor needs to have a minimum resolution of 640 x 480. The following image sensors from Apatina meet this design specification as shown in table 11 on the next page.

Image Sensor	Resolution	Image Senor Size	Frame Rate
MT9V024IA7XTC	WVGA	1/3 inch	60 fps
MT9V024IA7XTR	WVGA	1/3 inch	60 fps
MT9V032C12STC	WVGA	1/3 inch	60 fps
MT9V032C12STM	WVGA	1/3 inch	60 fps
MT9V032D00STC	WVGA	1/3 inch	60 fps
MT9V032D00STM	WVGA	1/3 inch	60 fps
MT9V034C12STC	WVGA	1/3 inch	60 fps
MT9V034C12STM	WVGA	1/3 inch	60 fps

Table 11: Image sensors from Kodak that have a minimum of 30 fps

This method unfortunately will consume too time than is available to research, design, test and implement this solution. Especially on how to take the raw video data compress it to a format and send it over the air to a receiver. So instead the second method will be looked at.

The second method involves using a store bought camcorder or webcam. It will still require a processing unit capable of converting the raw video data into a format and send it wirelessly to a display unit. Finding a camcorder or webcam that meets the design won't a problem. The problem will be finding one that has information on what video format it outputs. Once that is known then the processing unit can take the video, convert it if it has to, and send it to the transceiver. Although this method is much simpler than the first method, once again the time needed for this is not available. Instead the last method will be approach for the video system.

The last method will involve using a prepackage system and simply mounting it to the frame of the quad-copter. This has the possibility of costing the most but is simpler one. Unfortunately the time needed for the other method is simply not available. For this method two wireless video kits will be looked. The first one is the Raidentech 2.4 GHz mini wireless camera, 24ghzmiwicoc. The second one is going to be Lightinthebox wireless color camera kit with audio, SFA-010256. The table 12 shows the price, range and weight of both systems.

Kit	Range	Weight	Price
24ghzmiwicoc	150 m	9 g	39.99
SFA-010256	100 m	1.3 kg	64.99

Table 12: Specifications of the two wireless video kits.

With these specs the clear choice was the 24ghzmiwicoc. This wireless kit will need it owns power supply apart from the quad-copter power supply. The transmitter and camera have the dimensions of 15mm x 22mm x 32mm. This will be small enough to mount on the quad-copter.

3 Design

3.1 Linear Control System

The linear control system is arguably the most important part of the quad-copter. It is responsible for assuring that the Quad-copter fly stable and straight. If the linear control system is not a good one then the Quad-Copter will not have a successful flight. For this reason the linear control system will take priority in research, design, prototyping and testing above the other subsystems. They are many ways to reach a solution for a liner control system. The perfect system for the quad will be one that is fast, accurate and can be implemented on an 8-bit microcontroller.

There are different types of linear control systems that can be used as the basis for the quad-copter. The simplest one is to take a desired value and apply that value to the process controlling a device. This is called and open loop control system. This type of linear control system works great in an ideal environment where there is no outside force affecting the system. The quad-copter will have 4 motors all running at different speeds. In addition to that outside force such as wind will be affecting the balance of the quad-copter. For this reason a simple control system like this one will not be effective for the quad-copter. A better solution is needed.

A different approach is needed in order to take into account the different variables that will affect the quad-copter. The solution is to use a feedback control system. In a feedback control system the difference between the actual output and a desired output is fed back into the system. Using this information the system can make adjustments to the input of the system to get closer to the desired output. An example of this is shown in figure 11.

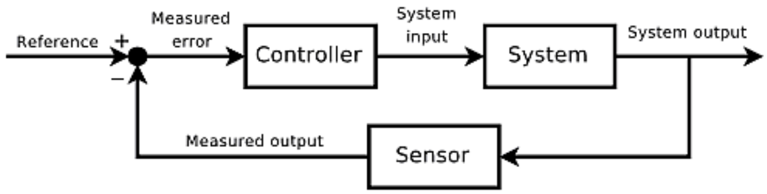

Figure 11: General block diagram of a feedback control System.
Reprinted from Wikipedia.org

In the picture above it shows that it has a reference signal, which is the desired output, and this signal is being added with an error signal. This error signal is then used as the input for a controller. This controller uses the information to make the adjustments to the input of the system. This kind of system is well suited for the quad-copter, and will the design choice for the linear control system. The critical part of the linear control system will be the controller.

In a closed loop system 4 four types of controllers are available. The controllers are the proportional controller, or P controller, the proportional-integral controller, or PI, the Proportional-Derivative, or PD, and the Proportional-Derivative-Integral controller, or PID. One of these controllers will be the basis for the linear control system's controller.
The Integral part of a Proportional-Integral controller takes a sum of all errors and uses that as the input to the system. Using only the integral term the output will behave like a straight line until it reaches the desired output. According to the paper "AVR221: Discrete PID Controller," "the summing of the error will continue until the system process value equals the desired value, and this results in no stationary error when the reference is stable." Figure 12 shows how the integral part of a PI helps in achieving the desired output. The Proportional-Integral controller is more suitable for quad-copter but it can create a little overshoot.

The derivative of a Proportional-Derivative controller adds more to the error signal which compensate for the rate of change of the error. In ""AVR221: Discrete PID Controller" the author notes that when using the derivative term "a rapid change in the error will give an addition to the system control input. This improves the response to a sudden change in the system state or reference value." The results of using a PD controller instead of a P controller is shown in figure 13. The Proportional eliminates the overshoot of a P controller but it doesn't eliminate the stationary error.

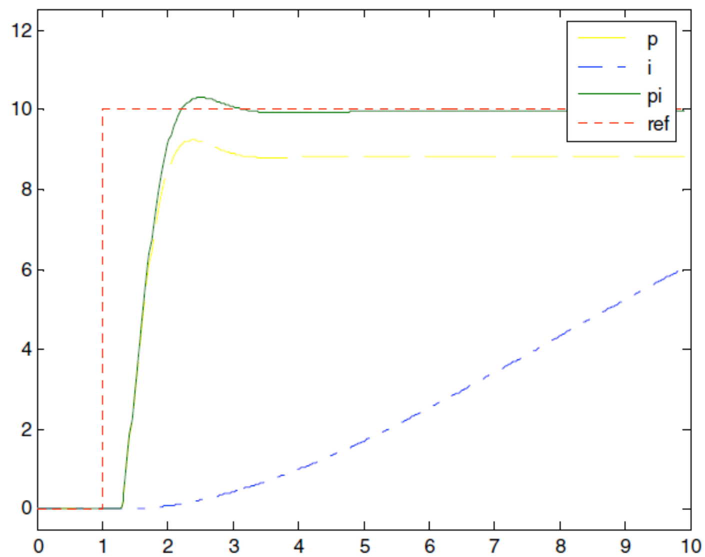

Figure 12: Comparison between a P, I, and PI controller.
Reprinted with permission from atmel.com.

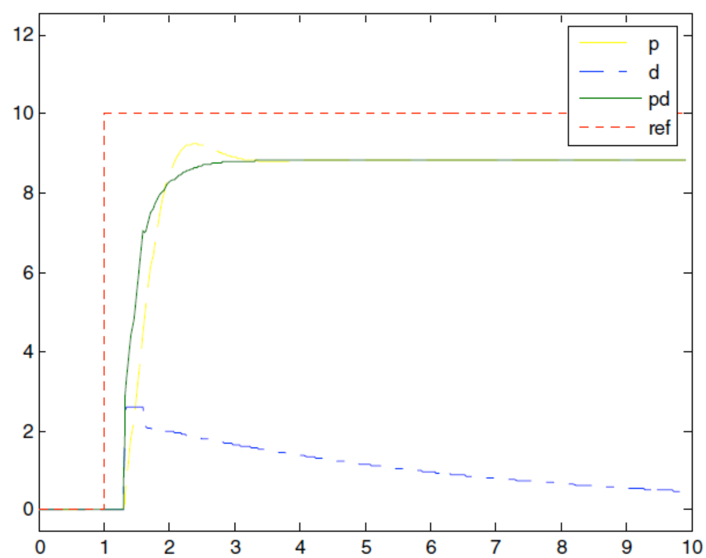

Figure 13: Comparison between a P, D, and PD controller.
Reprinted with permission from atmel.com.

The final controller is the Proportional-Integral-Derivative controller. In this controller the effects of each controller is added in order to obtain is fast and stable response. The stationary error of the P controller is offset by the I term and the overshoot of the P controller is offset by the D term and makes the response faster. For the quad-copter the PID controller will be of choice. Figure 14 shows the difference between a P controller, a PI controller and a PID controller.

On the quad-copter, an IMU system comprised of a 3 axis accelerometer and a 2 axis gyroscope will be the sensors used in order to provide the linear control system with an output signal. The IMU system will be tested in order to get the reference signal that will tell the linear control system that the quad-copter is steady and stable. The accelerometer will be used for pitch and roll while the gyroscope will be used as a heading indicator or yaw sensor.

The accelerometer and gyroscope have errors in their reading that needs to be addressed before using them as sensors for the linear control system. A gyroscope has a drift that makes the gyroscope accumulate error over time. This makes the gyroscope great at sensing for a short period of time; but poor at sensing for a long period of time. While the accelerometer is great at sensing for a long period of time it is susceptible to noise. In order to get a good signal out these sensors their output had to be combined somehow. The way that they are combined is through the use of an averaging system. This system is called the Kalman filter. Using a Kalman filter the output of the IMU system can then be used in PID controller.

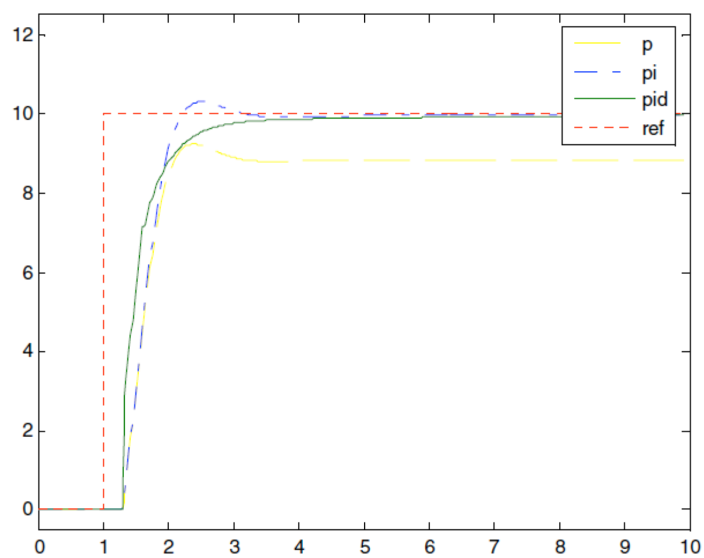

Figure 14: Comparison between a P, PI, and PID controller.
Reprinted with permission from atmel.com

The PID controller will be used to correct deviation from the desired output. The two variables that will be controlled with the linear control system will be the roll, pitch and yaw of the quad-copter. The current roll, pitch and yaw will be sensed by the quad-copter's accelerometer and gyroscope. This signal will pass through a Kalman filter, which will be implemented on an 8-bit microcontroller. This signal will then pass through the PID controller, also implemented on an 8-bit microcontroller. The PID will then calculate the amount of thrust needed to get the system to the desired output. There will be three PID controllers one for the pitch, one for the roll, and one for the yaw. The microcontroller will cycle through each PID at a set interval until the desired output is reached.

The ideal parallel form of the PID controller equation will be the basis for the linear control system. This equation is not suitable for implementation on an 8-bit microcontroller because is a continuous equation. The equation needs to be transformed into a discrete equation. The ideal parallel form equation is shown below:

Ideal Parallel equation: $$MV(t) = K_p e(t) + K_i \int_0^t e(\tau)dt + K_d \frac{d}{dt} e(t)$$

In order to make the PID controller equation a discrete the integral term and the derivative term has to be in discrete form. For the derivative term a backward finite difference form will be used. For the integral term a summation of all previous error will be used. The error signal will be a discrete function where each error will be sampled and stored in an array. The derivative term will calculate the difference between two consecutive error samples and divide it by the sampling period, while the integral term will do a summation of the array and multiply it by the sampling frequency. The discrete equation will look like this:

Discrete equation: $$x(n) = K_p e(n) + K_i \sum_{K=0}^{n} e(k) + K_d [e(n) - e(n-1)]$$

K_i term: $$K_i = \frac{K_p T}{T_i}$$

K_d term: $$k_d = \frac{K_p T_d}{T}$$

Where, T will be the sampling period and T_i the integral time constant.

The 8-bit microcontroller will limit how well the PID controller can be implemented in software. The 8-bit microcontroller has a low resolution that has to be taken in to account in calculating the PID controller term. Lastly the 8-bit microcontroller doesn't have floating point capability. Which means the PID controller equation will have to be implemented with integer math.

Getting the PID controller to work correctly will take time and effort, because it will be a trial and error process. The derivation of the PID controller equation is easy, what makes getting a PID controller to work correctly hard is getting the right gain for each term. The process used to get the right gains is called loop tuning. They are many methods that are used to get the correct gains of a PID controller. The most accurate will be using a mathematical representation of the system. Unfortunately this is not always available as is in the case of the quad-copter. The next method is to use a software package to help in tuning the PID loop. This software package will need to take into account the analysis and simulation of the sensors and motors. But at this time this also not available to use. The last method is to find the gain trough brute force, or manual tuning. Although this method will work eventually, is not suitable as the quad-copter project doesn't have the time for it. There is a method that uses some sort of manual tuning in conjunction with simple mathematical formulation and it proven to work. This method for loop tuning is called the Ziegler–Nichols method.

The Ziegler–Nichols method starts out by setting the Ki and Kd to zero. Then Kp is raised until the output oscillates, the gain that this happens at is called the ultimate gain, Ku. The period of the oscillation is called Pu. After that simple equations are used in order to find Ti, Td, Ki, and Kd. Table 13 shows the calculations of the gains.

Control Type	K_p	K_i	K_d
P	$0.50K_u$	-	-
PI	$0.45K_u$	$\frac{1.2K_p}{P_u}$	-
PID	$0.60K_u$	$\frac{2K_p}{P_u}$	$\frac{K_p P_u}{8}$

Table 13: Ziegler–Nichols method calculations

Stephen R. Smith 9/26/10 5:17 AM
Comment: check fig #

The next step is to design the software for the PID controller. The PID controller will have three methods. One will be to initialize and setup the PID controller, the other one will reset the PID controller, and the last one will be the PID controller. The first method will be called initPID. It will have as augments the desired output, the measured output, Kp, Ki, Kd, and a structure that will hold the current status of the pid controller. The initPID will be called as follows: initPID(desired output, measured output, status structure). The status structure will contain the values of the last measured error, Kp, Ki, Kd, the summation of the errors, the maximum error allowed, and the maximum summation error. The structure will look like the following:

```
Struct PID_Status {
        last_measured_error;
        summation_error;
        Kp_Gain;
        Ki_Gain;
        Kd_Gain;
        max_error;
        max_summation error; }
```

The initPID method will initialize the last_measured_error, and summation_error to zero. It will also set the PID gains to the ones decided by the user. Based on the gains initPID will calculate the max_error and max_summation error.

The second method will be called resetPID. It will responsible for setting the parameters of the PID controller back to zero. The resetPID method will use the same status structure as the initPID function and set all values in the structure to zero.

The third method will be the one that implements the PID controller. The method will be called PID_Controller. The PID_Controller method will have as arguments the desired output, the current measured output, and the status structure PID_Status. This function will do the integer math needed to calculate it the appropriate output. The function will take the difference of the desired output and measured output to calculate the error. This error will then be multiply by the Kp_Gain in order to get the proportional term. The error will then be added with all the previous errors and multiply by the Ki_Gain to get the integral term. Lastly the error will then be subtracted from the previous error and multiplied by the Kd_Gain in order to get the derivative term. Once all the PID terms are calculated they will be added together and used as the output. This function will use the variables in table 14, in order to calculate the appropriate output.

Variable Name	Function of the variable
Pterm	This will hold the Proportional term.
Dterm	This will hold the Derivative term.
Iterm	This will hold the Integral term.
error	This will be the calculated error.
temp	This will hold values to check for overflow.
desired_value	This will be the value desired value.
measured_value	This will be the measured value.
return	This will be the output variable.

Table 14: Variables used by the PID controller function.

The last part needed is the gain the PID terms. In order to calculate the gains the Ziegler–Nichols method will be used. The quad-copter will be held a platform that will enable testing of either pitch or roll one at a time. From there the Ziegler–Nichols method will be followed as previously described. At this time is uncertain whether or not once set of PID gains will be used for both pitch and roll.

3.2 Wireless Communication System

The wireless communication system will be crucial in controlling and getting status messages from the quad-copter. It will be crucial a subsystem of the quad-copter behind the autonomous system. The design of the wireless communication system will be largely software based. The hardware will be basically connecting a MCU to a transceiver and designing a balun for a 2.4 GHz antenna with an impedance of 50 ohms.

The wireless communication system will be based on a proprietary protocol created by the manufacture of the transceiver that will be chosen. The quad-copter needs a simple communication protocol, and because a custom protocol was the best idea. Unfortunately the time required to design and implement a wireless communication protocol is too much. This lead to the conclusion that whatever transceiver is chosen the manufacturer's simple communication protocol will be used.

The chosen transceiver is Texas Instruments' CC2520. This transceiver was design with low power wireless communication in mind. It was chosen for the quad-copter because it has the best overall performance in terms of radio sensitivity, output power, price and physical size. The CC2520's features are summarized below:

- DSSS transceiver
- 250kbps data rate, 2 Mbps chip rate
- O-QPSK with half sine pulse shaping modulation
- Very low current consumption
    - RX (receiving frame, -50 dBm): 18.5 mA
    - RX (waiting for frame): 22.3 mA
    - TX (+5 dBm output power): 33.6 mA
    - TX (0 dBm output power): 25.8 mA
- Three flexible power modes for reduced power consumption
- Low power fully static CMOS design
- Very good sensitivity (-98dBm)
- High adjacent channel rejection (49 dB)
- High alternate channel rejection (54 dB)
- On chip VCO, LNA, PA and filters.
- Low supply voltage (1.8 - 3.8 V)
- Programmable output power up to +5 dBm
- I/Q direct conversion transceiver

The CC2520 was also chosen because very few external components are needed for a design solution. It also comes equipped with a SPI connection and 6 GPIO, general purpose input output, pins. This makes the CC2520 a flexible transceiver to control. It also comes with RAM, which should be enough space for buffering and data storage. Another thing that makes the CC2520 an attractive transceiver is the fact that it has hardware support for the IEEE 802.15.4 MAC/Physical layer standard. This will eliminate that need to design and program a physical layer for the communication protocol.

The CC2520 transceiver works in conjunction of a microcontroller. Texas Instruments makes their wireless communication protocols and software examples only available to its own microcontrollers. For this reason the Texas Instruments' MSP430F2616 microcontroller was chosen. The MSP430F2616 has the following features:

- Low Supply Voltage Range: 1.8 V to 3.6 V
- Ultralow Power Consumption
- 16-Bit RISC Architecture
- Up to Four Universal Serial Communication Interfaces
- 12-Bit Analog-to-Digital (A/D) Converter
- Hardware Multiplier Supporting 32-Bit Operations
- 128 KB of Flash and 16 KB of SRAM

The MSP430F2616 will be handling the protocol software stack as well as any control instructions and status messages defined for the quad-copter. The MSP430F5436A and the CC2520 will be interfaced together through an SPI connection and a couple of GPIO.

The protocol that will be used for the wireless communication is called SimpliciTI. This protocol stack was developed for simple sensors networks or device to device communications. The following are the features for SimpliciTI:

- Low power - A TI proprietary low-power network protocol
- Flexible:
  - Direct device-to-device communication
  - Simple star with access point for store and forward to end device
  - Range extenders to increase range to 4 hops
- Simple - Utilizes a 5 command API
- Low data rate and low duty cycle
- Ease-of-use

The great thing about this protocol is that is implements a communication protocol using a 5 command API. This greatly simplifies the programming needed to get two devices communicating. The bad thing is that this protocol is proprietary but Texas Instrument is not expected to hand solutions for free with no strings attached.

The CC2520 transceiver comes with in a QFN 28 (RHD) package. In the package the transmitting and receiving pins are RF_N (pin 19) and RF_P (pin 17). These are the pins where the antenna will be connected. The impedance of the antenna and the CC2520 needs to be matched before an antenna can be connected. The circuit the does the matching of impedances is called a balun. A search through the TI's website comes up with a reference design in the CC2520 datasheet. The figure 15 shows that reference design.

In the reference design a balun is shown consisting of transmission lines and capacitors. The balun circuit looks too complicated so instead a new solution is needed. The circuit is complicated because the transmissions lines consists of PBC traces that are X width and length, not only that but the traces are not straight lines. This complicates the PCB design which is not wanted. Also the performance of the transceiver can be affected by the thickness of the PCB. Further search through TI's website, in the forums, a solution is found. The transmission lines can be replaced with discrete inductors or better yet the whole balun circuit can be moved into an integrated circuit that can be bought. The integrated circuit that comes in a balun circuit inside is the Murata LDB182G4510C-110.

In forums member, and TI employee, Richard Wallace detailed a way to use the Murata balun IC instead of using the reference design. He wrote that "you can use a standard Murata balun (LDB182G4510C-110) with a shunt inductor of 3.9nH and a LC filter at pin 1 of the balun component. The L should be 1.5nH and the C 2.2pF. Use a bias cap of 10nF connected to pin 3 of the balun." The drawing of how it should look like which is shown in figure 16.

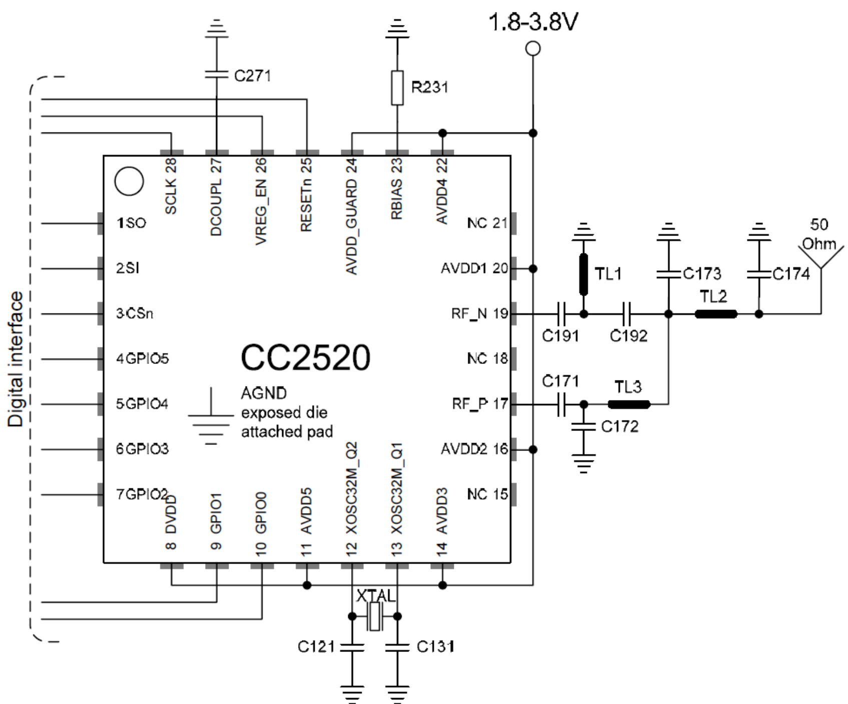

Figure 15: Reference design for the CC2520
Reprinted with permission from Ti.com

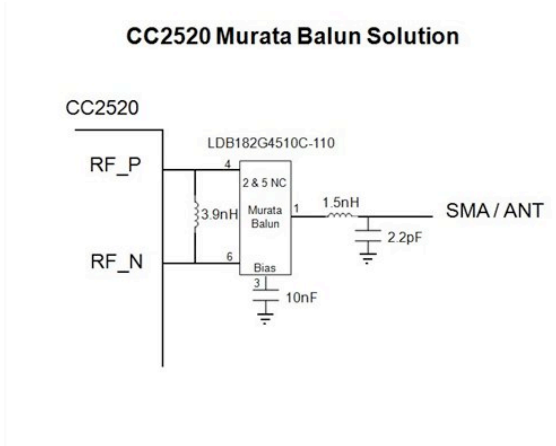

Figure 16: Balun circuit for the CC2520
Reprinted with permission from TI.com

This is the design that will be used in order to connect the 2.4 GHz dipole antenna to the CC2520. The antenna will use an RP-SMA connector.

The next step is to design the crystal circuit. According to the datasheet the crystal needs to oscillate at 32 MHz The crystal has two loading capacitors and it connected to XOSC32M_Q2 (pin 12) and XOSC32M_Q1 (pin 13). It order to keep the crystal from deviating too much from 32 MHz, a low ppm range is need. The crystal that will be used is the ECS-320-8-36CKM, made by ESC Inc. It can oscillate at 32 MHz and it has a 10 ppm range. The two load capacitor will be around 15 pf. The reason for this is so the crystal has enough capacitance to start. So In the reference design C121 and C131 will be 15 pf each.

Two more pins need to be taken care before the interface between the CC2520 and the MSP430F2616 can be designed. The pins are RBIAS (pin 23) and DECOUPL (pin 27). The RBIAS pin needs to be a resistor with a resistance of 56k ohms and a low tolerance of 1%. This is what the data sheet recommends to use. The DECOUPL pins will be a capacitor with a capacitance of 1 uf. This value was researched through the TI's forums. Member H Stewart points out that when he was using a 100 pf capacitor it would makes the internal 1.8 V regulator oscillate. He fixed the problem by using a 1 uf capacitor.

The CC2520 uses an SPI interface to connect to a microcontroller, for programming. This interface will act as a SPI slave device. The SPI pins on the CC2520 are the following pins SCLK, SO, SI, and CSn; which are pins 28, 1, 2, and 3 respectably. On the MSP430F2616 the port UCB0 is going to be used as the SPI Master interface. The pins for the UCB0 port are UCB0CLK, UCB0SOMI, UCB0SIMO, and UCB0STE; which are pins 36, 35, 34, and 33 respectably. The clock generator for the SPI interface is going to the UCB0CLK on the

MSP430F2616. This pin will generate a clock signal for SCLK pin on the CC2520. The MSP430F2616 will send data to the CC2520 through pin UCB0SIMO into the CC2520 SI pin. In order to receive data the MSP430F2616 will use pin UCB0SOMI, which will be connected to the CC2520 SO pin. This is shown in Figure 17.

The next step, in the design, is to connect the VREG-EN and RESET pin on the CC2520 to the microcontroller. The VREG-EN is going to be used in order to enable the internal registers of the CC2520. The RESET pin is as its name implies to reset the CC2520. The VREG-EN, pin 25 on the CC2520, will be connected to pin P1_0/TACLK/CAOUT, pin 12 on the MSP430F2616.

The CC2520 also has 6 GPIO pins that can be fully programmed for anything. In order to have the flexibility of using the 7 GPIOS, these pins will be connected to MSP430F2616. GPIO0 will be connected to pin P01.3/TA2, pin 15 of the MSP430F2616. GPIO1 will be connected to pin P01.5/TA0, pin 17 of the MSP430F2616. GPIO2 will be connected to pin P01.6/TA1, pin 18 of the MSP430F2616. GPIO3 will be connected to pin P01.1/TA0/BSLTX, pin 13 of the MSP430F2616. GPIO4 will be connected to pin P01.2/TA1, pin 13 of the MSP430F2616. GPIO5 will be connected to pin P01.7/TA2, pin 13 of the MSP430F2616. Figure 17 shows the connection of each pin.

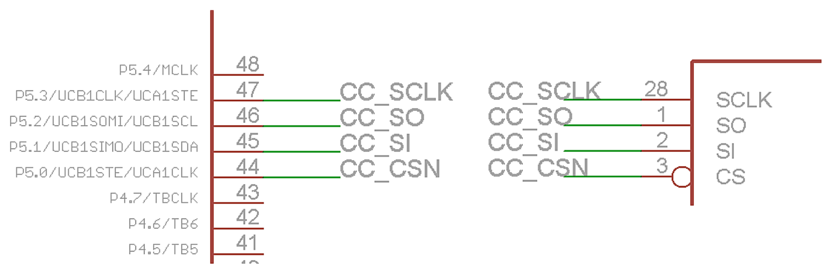

Figure 17: Schematic of interfacing the MSP430F2616 and the CC2520

Stephen R. Smith 9/26/10 5:24 AM
Comment: do something

The SimpliciTI protocol will be used for the wireless communication protocol. It will be used to create a peer to peer network. Where the quad-copter will send status messages to a remote controller and receive commands from the remote controller. The SimpliciTI API is a simple 6 function API. This will help greatly in reducing the time dedicated to getting the SimpliciTI setup and running.

The data structures that will be used for the wireless communication protocol will be the following. The first data will be a structure with two char array that will hold the GPS coordinates received from the GPS receiver. The structure will be called gps_coordinates. The first chart array will be the latitude portion of the coordinate, and it will be called gps_lat. This will be 10 characters long. The second array will be the longitude portion of the coordinate, and it will be called gps_long. This will be 11 characters long.

CC2520	MSP430F2616
VREG-EN	P1_0/TACLK/CAOUT
RESETn	P5.7/TBOUTH/SVSOUT
GPIO0	P01.3/TA2
GPIO1	P01.5/TA0
GPIO2	P01.6/TA1
GPIO3	P01.1/TA0/BSLTX
GPIO4	P01.2/TA1
GPIO5	P01.7/TA2

Figure 18: Pins connections between CC2520 and MSP430F2616
Reprinted with permission from TI.com.

Stephen R. Smith 9/26/10 5:26 AM
Comment: something

The second data structure that is going to be the variable used to send payload data through the wireless communication system. This variable is going to be an array of bytes, and it will be called payload_data. The number of bytes is going to be 12. This variable will be formatted to able to send different types of command or data. The first byte sent will define the type of command or data sent.

The first type will be a command type, and it will have the value of 10000011. The rest of the bytes will be dedicated to the commands. To tell the quad-copter to throttle up the command will the ASCII value of "up" with trailing zeros. The rest of the command will follow this format and are summarized in table 15.

Command	Value	Description
Up	ASCII value "up"	Throttle up
Down	ASCII value "down"	Throttle down
Left	ASCII value "left"	Move left
Right	ASCII value "right"	Move right
Forward	ASCII value "forward"	Move forward
Backward	ASCII value "back"	Move back
Get GPS position	ASCII value "getgps"	Get GPS from copter
Send GPS position	ASCII value "sendgps"	Send GPS to copter

Table 15: Commands for the quad-copter and opt-code value

The second type will be a data type. The first data type will be the latitude position and it will have the value of 11110001. The second data type will be the longitude position and it will have the value of 11110011. The rest of the payload_data bytes will be filled with the values of the latitude or longitude. Now we need a way to send this information through the wireless communication system.

The setup for the SimpliciTI will be as follow. First the SimpliciTI protocol will be initialized on both devices. Then a link between the quad-copter and remote controller will be attempted, once a link is successful full data transmission and begin. The first function to be call will be SMPL_Init ( ). This function will initialize the SimpliciTI protocol. The function will be called as follow smplStatus_t SMPL__Init(uint8_t (*callback)(linkID_t)). In the call linkID_t is a variable of type char that will act as the address of the device,or link ID, being initialize. smplStatus_t is an enumeration that tells the status of the initialize call.

The next step is to call the functions that will enable the two devices to link. Those functions are SMPL_Link() and SMPL_LinkListen(). One device will call the SMPL_LinkListen() function first and then the other device will call the SMPL_Link() function. The SMPL_LinkListen() is called using the following format smplStatus_t SMPL_LinkListen(linkID_T *lid). In this call *lid is a pointer to a link ID that will be the address of the second device. SMPL_Link() follows the same format and return the same smplStatus_t variable. The different values for the variable smplStatus_t are shown on figure 19.

Name	Description
SMPL_SUCCESS	Operation successful.
SMPL_TIMEOUT	A synchronous invocation timed out.
SMPL_BAD_PARAM	Bab parameter value in call.
SMPL_NOMEM	No memory available. Object depend on API
SMPL_NO_FRAME	No frame available in input frame queue.
SMPL_NO_LINK	No reply received for Link frame sent.
SMPL_NO_JOIN	No reply received for Join frame sent
SMPL_NO_CHANNEL	Channel scan did not result in response on at least 1 channel.
SMPL_TX_CCA_FAIL	Frames transmit failed because of CCA failure.
SMPL_NO_PAYLOAD	Frame received but with no application payload.
SMPL_NO_AP_ADDRESS	Should have previously gleaned an Access Point address but we none.

Figure 19: Variables within the smplStatus_t structure
Reprinted with permission from TI.com

Now that the two devices are linked together, full wireless communication can begin. To send a message the function SMPL_Send() will be used. The format to call this function is smplStatus_t SMPL_Send(linkID_t lid, uint8_t *msg, uint8_t len. In this call *msg will a pointer to the message that will be sent, in this it will be a pointer to the payload_data variable. Also the variable len will the length of the message that will be sent, in this case this will always be 12 bytes. To receive a message the function SMPL_Receive ( ) will used. The format to call this function is smplStatus_t SMPL_Receive(linkID_t lid, uint8_t *msg, uint8_t *len). Everything is the same as the send function except that *len is a pointer to a variable that will hold the length of the message received.

The last function that can be used will a function used to gain more control of the transceiver. This function is called SMPL_Ioctl(). To call this function the format is smplStatus_t SMPL_Ioctl(ioctlObject_t obj, ioctlAction_t act, void *val). The argument ioctlObject_t obj is a parameter that tells the function which object will be modified. The argument ioctlAction_t act is the action that will be performed on the object. The last argument is *val which is a pointer to a parameter information. The following code shows the objects that can be modified and what can be modified on the object.

```
enum ioctlObject{
IOCTL_OBJ_FREQ,
IOCTL_OBJ_CRYPTKEY,
IOCTL_OBJ_RAW_IO,
IOCTL_OBJ_RADIO,
IOCTL_OBJ_AP_JOIN,
IOCTL_OBJ_ADDR,
IOCTL_OBJ_CONNOBJ,
IOCTL_OBJ_FWVER,
IOCTL_OBJ_PROTOVER,
IOCTL_OBJ_NVOBJ,
IOCTL_OBJ_TOKEN
};

enum ioctlAction{
IOCTL_ACT_SET,
IOCTL_ACT_GET,
IOCTL_ACT_READ,
IOCTL_ACT_WRITE,
IOCTL_ACT_RADIO_SLEEP,
IOCTL_ACT_RADIO_AWAKE,
IOCTL_ACT_RADIO_SIGINFO,
IOCTL_ACT_RADIO_RSSI,
IOCTL_ACT_RADIO_RXON,
IOCTL_ACT_RADIO_RXIDLE,
IOCTL_ACT_RADIO_SETPWR,
IOCTL_ACT_ON,
IOCTL_ACT_OFF,
IOCTL_ACT_SCAN,
IOCTL_ACT_DELETE
};
```

3.3 Flight Stability System

The flight stability system consisting of a triple axis accelerometer and a dual axis gyroscope was an essential component of the quad-copter that needed to be successfully implemented in order for the project to be a success. The accelerometer and gyroscope can be considered collectively as an IMU (Inertial

Measurement Unit). Hierarchically, the IMU data could be structured with the X and Y axes being primarily measured by the accelerometer while being supported by averaging in gyroscope data. The degree to which the gyroscope data will be necessary can be estimated during the planning stage but, determined more accurately during the testing and prototyping phases. Similarly, data from the Z axis of the accelerometer was planned to be utilized only so far as it provided useful data for the purpose of stabilizing the quad-copter. Both accelerometer and gyro data were sent as analog signals to ADC pins on the MCU. The accelerometer was supplied by 3.3 V and provided a corresponding radiometric signal to the MCU. The gyro however, has an operating voltage range of 2.7 V to 3.3 V so, had its voltage supply regulated down to 3.0 V to eliminate the risk of bad data or worse.

The gyro outputs are described by the datasheet as, "independent of supply voltage (i.e. . . . not radiometric)" so, therefore the discrepancy in supply voltage should not incur scaling errors when integrating accelerometer and gyro data. The IDG-500 gyro provides two output ranges with different corresponding sensitivity as illustrated in figure 20 below.

Axis	Gyro Output	Sensitivity (mV/°/s)	Full-Scale Range (±°/s)
X	X-OUT	2	500
	X4.5OUT	9.1	110
Y	Y-OUT	2	500
	Y4.5OUT	9.1	100

Figure 20: Gyro output ranges
Reprinted with permission form InvenSense

The full scale range of +/- 110 deg/s is more applicable to the quad-copter and so the X4.5OUT and Y4.5OUT pins will be wired to the MCU instead of the X-OUT and Y-OUT . The Gyro is equipped with an internal low-pass filter for noise reduction and the data sheet raised the option of an adding an additional external filter but, this was not used for the quad-copter. Figure 21 below shows a wiring schematic of for the IDG-500 gyro that was implemented on the main PCB of the quad-copter.

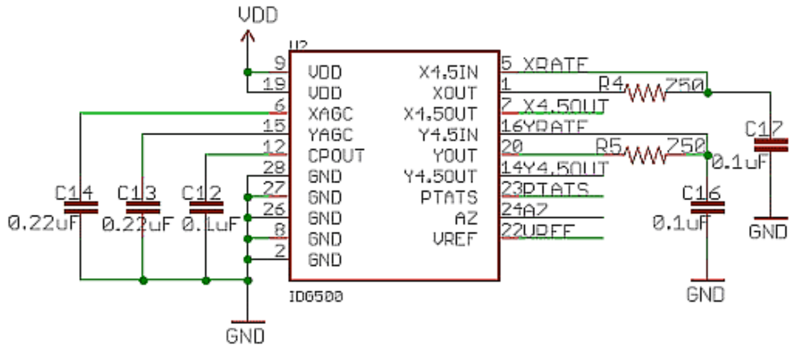

Figure 21: IDG-500 wiring schematic
Reprinted with permission from Sparkfun

Contrary to this schematic, the X-OUT and Y-OUT pins were not utilized for the quad-copter in order to simplify design where possible. Also, pin 23, an analog temperature output was not wired. Pins 6 and 15 are compensation capacitors used internally by the chip to maintain constant scaling over the temperature range. Pin 24 is the auto zero function which is used in conjunction with the X4.5 and Y4.5 outputs. Pin 22, is used as a reference for the zero function. Pins 6 and 15 are high impedance nodes that should not be coated. It is recommended that the voltage supply at pins 9 and 19 (VDD) should be isolated from system power by an RC filter and an LDO (Low Drop Out) power supply regulator. Such a schematic is provided in figure 22 below.

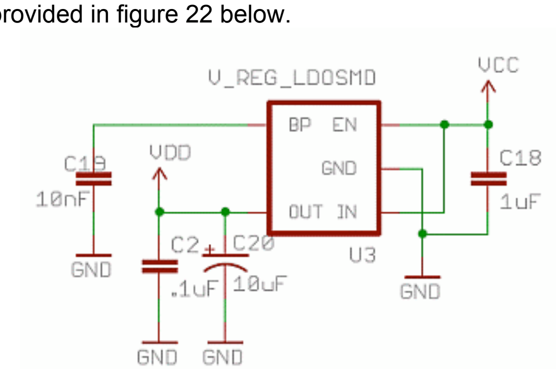

Figure 22: Power regulation for the IDG-500 gyro
Reprinted with permission from Sparkfun.

The ADXL-335 triple axis accelerometer is capable of measuring acceleration within a range of +/- 3 g. The bandwidth of the output can be selected according to the application within a range of .5 Hz to 1600 Hz for the Xout and Yout pins or between .5 Hz and 550 Hz on the Zout pins. It was set at 50 Hz. The application bandwidth is selected by varying capacitance at the X, Y, and Z outputs.

According to the data sheet, a .1 µf capacitor will provide an output bandwidth of 50 Hz, which was used for the quad-copter. The bandwidth limiting capacitors at Xout, Yout, and Zout implement low pass filtering and noise reduction. A schematic diagram detailing the implementation of the ADXL-335 on the main PCB for the quad-copter is provided in figure 23 below. The figure does not include an additional .1 µf capacitor at the VCC supply pin that was used to further reduce noise on the power supply.

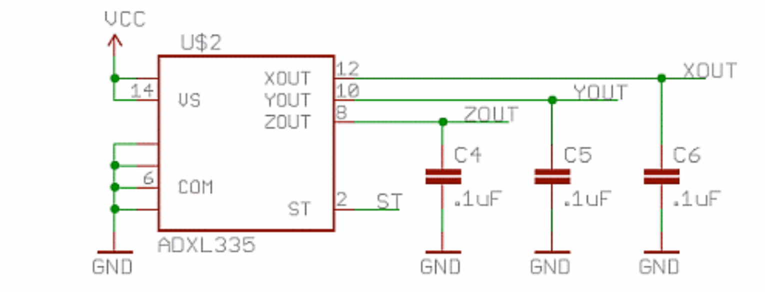

Figure 23: ADXL-335 wiring schematic
Reprinted with permission from Sparkfun.

Regarding the final layout of the accelerometer and gyroscope, they were mounted orthogonal to one another with the X and Y axes synchronized to achieve maximum programming efficiency.

The IMU could perhaps be considered as greater than the sum of its parts as the it incorporated a degree of sensor fusion between the accelerometer and the gyroscope. Sensor fusion algorithms have the objective of finding a weighted average based on multiple sensor's measurements that has a higher degree of statistical accuracy than the measurement any single sensor. The Kalman filter is perhaps the quintessential sensor fusion algorithm. The Kalman filter is a complex mathematical solution involving predicted states, estimated uncertainty of the predictions, and a computed weighted average of the predicted value versus the measured value ("Kalman Filter" Wikipedia). While the Kalman filter may offer the highest degree of accuracy among sensor fusion algorithms, the mathematical complexities involved with implementing a Kalman filter may outweigh its gains in accuracy for the quad-copter over a more primitive algorithm. The IMU for the quad-copter was be based on an on-line tutorial for a 5 degrees of freedom IMU posted by Starlino that falls into that category of "more primitive" algorithms. Starlino's algorithm differs fundamentally from a Kalman filter in that "weight is relatively fixed, whereas in Kalman filter the weights are periodically updated based on the measured noise of the accelerometer readings" ("A Guide…", Starlino). This type of approach is applicable to the quad-copter design because by changing only one coefficient, wGyro, in the code the weight of the gyro can be incrementally increased to find an optimal balance in the IMU. (Figure 24 below shows a representation of the accelerometer's axes, with R being the resultant force vector with component angles Axr, Ayr, and Azr.

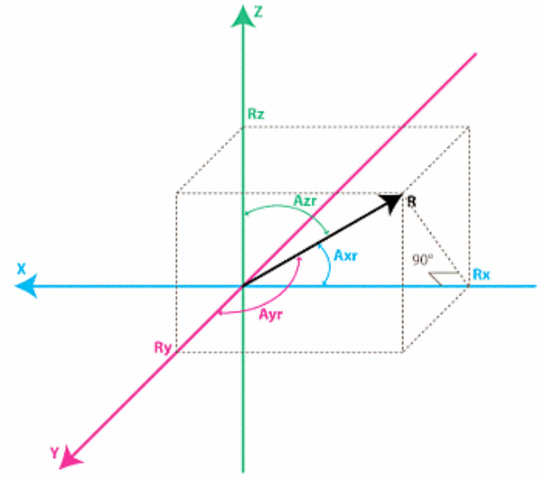

Figure 24: Representation of accelerometer axes and resultant vector. Reprinted under the condition of Starlino attribution

In order to get a grasp on the requirements for the IMU software, the group created a simulation in C that manually accepts digital accelerometer and gyroscope values. The simulation converts the raw data into angles and furthermore, provides an improved estimate of the angular state. The simulation is limited in its functionality but serves its purpose as a building block of the code to be implemented on the MCU. Component angles Axr, Ayr, and Azr from the above picture correspond to the digital accelerometer value outputs of the ADC as shown below in a fragment of code from the simulation:

```
float AdcRx     // 0 – 4095 (assuming a 12 bit ADC register)
float AdcRy ;
float AdcRz ;
printf("Enter the Accelorometer values: AdcRx,AdcRy,AdcRz \n");
scanf("%f %f %f", &AdcRx, &AdcRy, &AdcRz);
```

Similarly, the digital gyroscope values are declared:

```
printf("Enter the Gyro values: RateAxz,RateAyz \n");
scanf("%f %f", &AdcGyroXZ, &AdcGyroYZ);
x_delta = (AdcRx - 2048 );       // 1 g = 410
y_delta = (AdcRy - 2048);
z_delta = (AdcRz - 2048);
```

A subsequent step converts the digital values to g-force and determines the magnitude of the vector:

```
Gx = x_delta / 410;  //
Gy = y_delta / 410;
Gz = z_delta / 410;   // g's
Rmag = sqrt((Gx * Gx) + (Gy * Gy) + (Gz * Gz));
printf("Rmag: %f \n",Rmag);
```

The following fragment shows how the angles Axr, Ayr, and Azr are determined. This step is useful for understanding the process but might be skipped in condensed MCU code that might work entirely with digital values:

```
Axr = (180/PI) * acos(Gx / Rmag);
Ayr = (180/PI) * acos(Gy / Rmag);
Azr = (180/PI) * acos(Gz / Rmag);
Xerr = Axr - 90;
Yerr = Ayr - 90;
```

The following fragment involves the fusion of the sensors that uses and stores data pertaining to present values, and 2 past values. This code requires a good deal of trigonometry and uses floating values, both of which will have to be streamlined for use in the MCU:

```
 for (i=0;i<3;i++)
    out_past[i] = Acc_past[i];
 Acc_now[0] = (AdcRx - 2048) / 410;
 Acc_now[1] = (AdcRy - 2048) / 410;
 Acc_now[2] = (AdcRz - 2048) / 410;
 RateAxz = (AdcGyroXZ - 1843) / 2.73;    // deg/s
 RateAyz = (AdcGyroYZ - 1843) / 2.73;
 Axz_past = atan2(out_past[0], out_past[2]);
 Ayz_past = atan2(out_past[1], out_past[2]);
 Axz_now = Axz_past + RateAxz * T;
 Ayz_now = Ayz_past + RateAyz * T;
 RxGyro = (sin(Axz_now))  /  (sqrt( 1 + cos(Axz_now) * cos(Axz_now) *
                tan(Ayz_now) *  tan(Ayz_now)));
 RyGyro = (sin(Ayz_now)) / (sqrt( 1 + cos(Ayz_now) *cos(Ayz_now) *tan(Axz_now)
                * tan(Axz_now)));
if (out_past[2] < 0)
   sign_RzGyro = -1;
else
     sign_RzGyro = 1;
 RzGyro = sign_RzGyro * sqrt( 1 - RxGyro*RxGyro - RyGyro*RyGyro);
 Rgyro[0] = RxGyro;
 Rgyro[1] = RyGyro;
 Rgyro[2] = RzGyro;
 out_now[0] = (Acc_now[0] + Rgyro[0] * wGyro) / (1 + wGyro);
 out_now[1] = (Acc_now[1] + Rgyro[1] * wGyro) / (1 + wGyro);
 out_now[2] = (Acc_now[2] + Rgyro[2] * wGyro) / (1 + wGyro);
 for (i=0;i<3;i++)
    out_past[i] = out_now[i];
 for (i=0;i<3;i++)
```

Acc_past[i] = Acc_now[i];

Lastly, figure 25 below shows a screenshot of the simulation tool with manually input accelerometer and gyroscope values that correspond to a level angular state

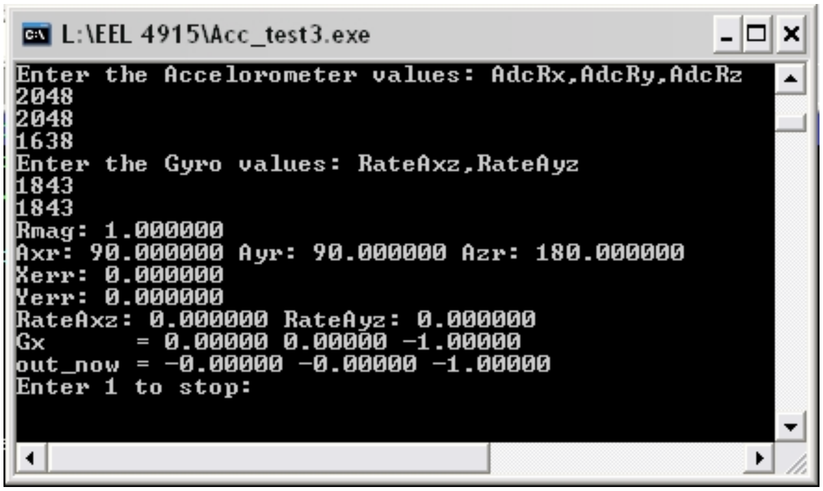

Figure 25: screenshot of IMU simulation

The Starlino sensor fusion algorithm was used as planned however; the level of stability achieved was insufficient. As a result the Kalman filter approach was tried but also with limited success. Ultimately, the best results were achieved by combining elements of both approaches in conjunction with a “homemade” algorithm that made use of the gyroscope to dampen the oscillations incurred by a P-loop algorithm.

3.4 Distance and ground sensors

While there was straightforward approach regarding how to integrate the ground sensor into the design of the quad-copter, the design of the lateral sensor system was slightly more open ended. Due primarily to concerns over cost and signal interference, and to a lesser degree weight considerations it was considered preferable to implement an effective lateral sensor system based on the least possible number of ultrasonic sensors. The group’s concept of the lateral sensor suite changed from a three-sensor system in June to a two sensor overlapping system in mid-July and finally to a single forward oriented sensor. A single forward oriented sensor was considered sufficient to alert the quad-copter to obstacles in its path when flying in an autonomous mode. The sensor for this purpose was the LV-MaxSonar- EZ2 which was used for the ground sensor as well. It was expected that optimization of the proximity sensor system would evolve during the testing and prototyping phases but will remain within certain parameters. However, the ultrasonic sensor system was only implemented in a rudimentary fashion without autonomous protocols. The ground sensor was envisioned as being mounted on the on the bottom of the quad-copter pointing directly down. There are two mounting holes on the base of the EZ sensor that could be used to screw the sensor directly to the frame. It was considered to insert a thin layer of foam or rubber between the sensor and the frame in order to minimize vibration. Mechanical dimensions of the EZ sensor are shown in figure 26 including the .124" diameter mounting holes [G].

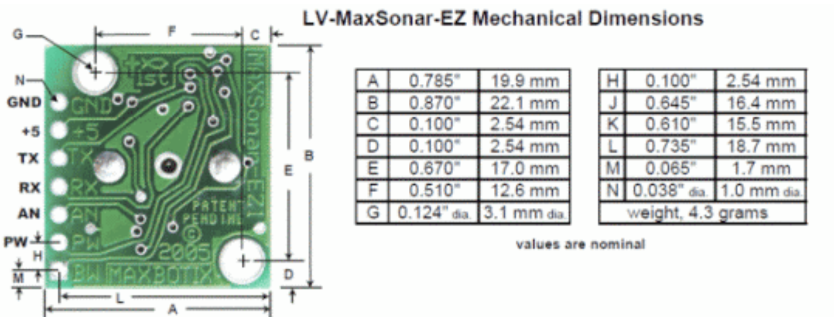

Figure 26: LV-MaxSonar-EZ Mechanical dimensions
Reprinted with permission from MaxBotix Inc.

Mounting the forward oriented distance sensor was not considered as straightforward as mounting the ground sensor. Placement and orientation of the lateral sensor needed to be optimized such that the legs of the quad-copter were not within its detection angle, or cone. A concern was that interference from the front motor and propeller could degrade performance. In relation to these constraints, two approaches developed of either mounting the sensor sufficiently below the motor or mounting it forward of the foremost motor. The final placement was planned to be determined when the quad-copter was at a more advanced stage of development during the prototyping and testing phases. Ultimately it was decided to mount the sensor on the landing gear at the “front” of the quad-copter.

According to the MaxBotix website, these sensors have a tendency to output unstable readings when electrical noise is present on the power line. Their recommendation was that, quoting from their website, "By placing a resistor in series with the V+, along with a 100uF capacitor. . .to ground, you create an effective filter (i.e. almost a placebo battery) for the sensor" (FAQ MaxBotix). This low pass filter that they recommend is shown in figure 27.

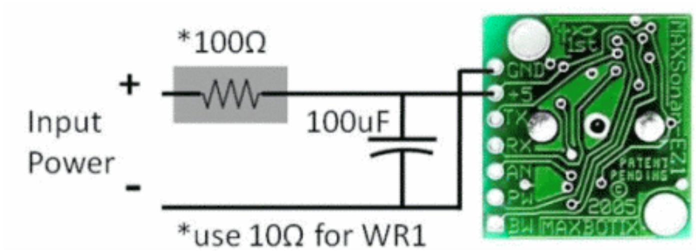

Figure 27: MaxBotix recommended low-pass filter.
Reprinted with permission from MaxBotix

Another design consideration regarding the EZ senor is that of the voltage supply. The EZ can operate within a voltage range of 2.5V to 5.5V. A higher supply voltage yields a slight increase in the range of the sensor. Figure 28 shows how the beam patterns differ between a 3.3V and 5V supply and also shows how the size of the detection field depends upon the size of the object being detected.

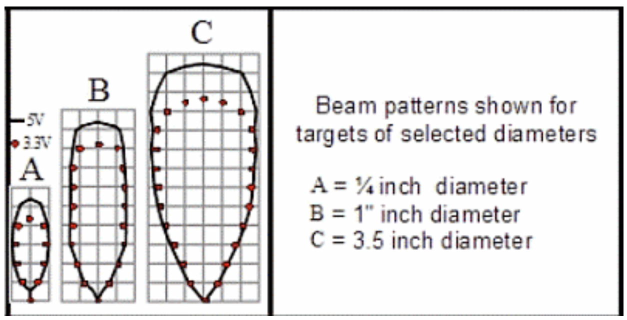

Figure 28: LV-MaxSonar-ez1 beam pattern diagram.
Reprinted with permission from MaxBotix.

In the interest of optimizing the range and effectiveness of the ultrasonic sensors, it was considered that they both be supplied at 5V and the AN analog output pin connected to a 5V ADC on the MCU. However, it was subsequently decided that supplying the sonar at 3.3 V, like the other sensors, would be easier to implement on the MCU. According to MaxBotix, the analog voltage is scaled at (Vcc/ 512) / inch which, in the case of a 5V supply yields ~9.8mV/ inch scaling.

3.5 Altitude Measurement and directional sensing

These two seemingly disparate design considerations are addressed under the same heading because a combination digital compass and pressure sensor was found online, Hoperf Electronics' HDPM01. The module requires no external supporting components and can be mounted on the quad-copter in such a manner as to minimize interference acting upon the digital compass. The module measures 17.7 mm X 16.3 mm X 4 mm as shown in figure 29.

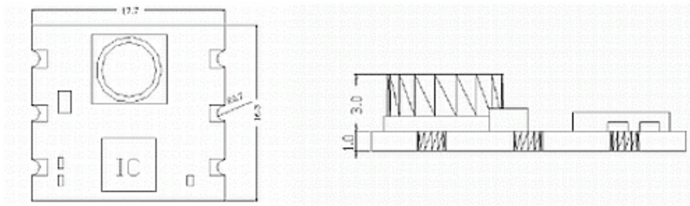

Figure 29: D103 HDPM01 Pressure/Compass combination module.
Reprinted with permission from Futurlec.

The compass could have been centrally located in order to maximize the distance to the electromagnetic fields of the four motors. Additionally, the module could have been placed so as to avoid the magnetic interference from metals and lesser electrical fields. For these reasons it was decided that the module would be mounted on the lower level of the quad-copter in proximity to the downward oriented ultrasonic sensor. This module does not have any mounting holes and so would have required a supporting bracket, clamp, or tie down of preferably non-metal composition. This sensor was not used in the final design due to difficulties with the I2C configuration and a poorly written datasheet.

The module would have required a supply voltage in the range of 2.2V to 3.6V which would have been met with a 3.3V supply due to system wide compatibility. At 3V the magnetic sensor draws approximately .4mA but, this is dependent on the frequency of requested measurements which in the case of the quad-copter would have been less. The module has integrated ADC convertors and performs some on chip signal processing with the output being relayed over the I2C serial bus. More precisely, the MCU must request data from the pressure or the compass sensor by an address over the I2C bus, parse the required data, and finally perform some calculations to refine the data. Fortunately, the quad-copter's MCU is powerful enough so as not to be overburdened by the extra calculations involved. The combination compass/altimeter module did not simplify design to the extent previously envisioned. For example, the module stores 10 sensor specific coefficients on EEPROM: C1-C7, A, B, C, and D. Some of these coefficients must be used to calculate a final pressure value along with the pressure measure and temperature measure values retrieved from within a timing sequence. An example of this process, copied from the HDPM01 datasheet, is shown below in figure 30.

Step 1: (get temperature value)

D2>=C5	dUT= D2-C5 - ((D2-C5)/2^7) * ((D2-C5)/2^7) * A / 2^C
D2 < C5	dUT= D2-C5 - ((D2-C5)/2^7) * ((D2-C5)/2^7) * B / 2^C

Step 2: (calculate offset, sensitivity and final pressure value)

OFF=(C2+(C4-1024)*dUT/2^14)*4
SENS = C1+ C3*dUT/2^10
X= SENS * (D1-7168)/2^14 - OFF
P=X*10/2^5+C7

- For altitude measurement system, recommend to use P=X*100/2^5+C7*10
- So that better altitude resolution can be achieved

Figure 30: Pressure and temperature calculation
Reproduced with permission from Futurlec.

Retrieving directional information from the compass module does not entail calculations as with the pressure module. However, the process does require accurately addressed, detailed commands from the MCU to "wake up" the sensor and retrieve the MSB and LSB for both the X and Y axes of the dual axis magnetic sensor. Note, the total resolution is 12 bits and not 16 bits.

Another design consideration regarding the HDPM01 pressure/compass combination module is the hardware implementation of the I2C bus. Figure 31 shows how both the SDA (Serial Data Line) and the SCL (Serial Clock Line) are connected both to devices and to the +VDD line across Rp pull-up resistors.

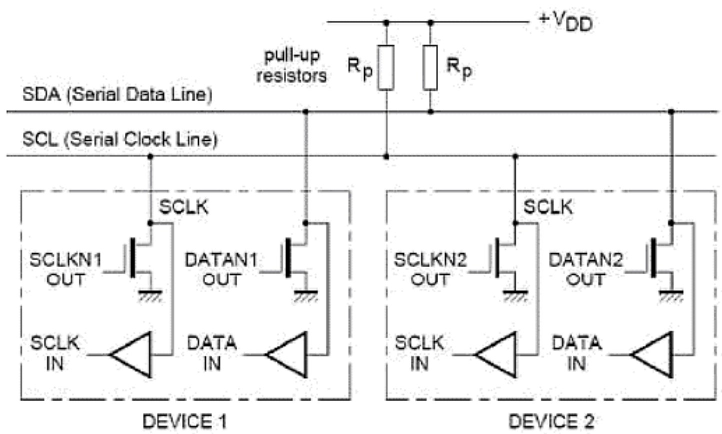

Figure 31: I2C bus, SDA and SCL lines.
Reprinted with permission from Futurlec.

The underlying concept behind this arrangement is that both the SDA and SDL must be in a high state in order for a data transfer to *start.* In order to correctly support this data transfer system the Rp resistors must be selected according to the bus length. The data sheet recommends 4.7kΩ for a bus length of between 2 and 4 inches which should be suitable for the quad-copter. It was later decided to use a general-purpose resistance of 1.8 kΩ.

3.6 Location/Navigation System - GPS Module

The Location/Navigation system of the quad-copter was determined to be the MediaTek-3329 GPS module. The module is quite compact at 16 mm X 16 mm X 6 mm, which while beneficial in terms of fitting onto the main board, could have complicated soldering and testing procedures. For this reason, the module was bought already soldered to an adapter. The module is equipped with both a USB and a UART serial interface, the UART being the preferred interface for the quad-copter. The original quad-copter design called for the module to be surface mount soldered directly onto the main PCB, however; the adapter allowed for the unit to be portable from the prototype board to final board. The module has a patch antenna incorporated into its ceramic casing, which must be mounted level, with the antenna pointing upward, for optimal reception. A protecting carbon fiber or plastic dome over the quad-copter was considered for protection that would not adversely affect reception of the GPS module. Figure 32 below shows the pin definition table for the MediaTek-3329 GPS module.

Pin	Name	I/O	Description
1	VCC	PI	Main DC power input
2	ENABLE	I	High active, or keep floating for normal working
3	GND	P	Ground
4	VBACKUP	PI	Backup power input
5	3D-FIX	O	3D-fix indicator
6	DPLUS	I/O	USB port D+
7	DMINUS	I/O	USB port D-
8	GND	P	Ground
9	TXD	O	Serial data output of NMEA
10	RXD	I	Serial data input for firmware update

Figure 32: MediaTek-3329 pin definitions
Reprinted with permission from DIY drones

In the design of soldering the component directly to the PCB, the VCC pin would have been supplied at 3.3 V. The VBACKUP would have been supplied with between 2 V and 3.3 V to facilitate normal operation, 3.3 V can be used here to simplify power distribution. A battery could be used for the backup but this would not have added any valuable redundancy to the design. The 3D-fix could be run to an LED, as with the optional breakout board, to indicate when communication has been established with a satellite. Final decision on the design was delayed until during testing and prototyping. The DPUS and DMINUS USB pins were not planned to be used except possibly during testing and prototyping. TXD was used to output the UART data. RXD, the UART receiver, was used to accept commands. The signal output was preconfigured to output a simplified binary string which, was considered applicable to the quad-copter application so as to preclude the use of the NMEA protocols. Ultimately, the NMEA protocols were found to be easier to use. The MediaTek datasheet lists the default baud rate at 9600 bps however; user comments on the DIY drones forums indicated that this might be in error. On June 24, 2010 Lew Payne writes, "Note that the new MediaTek has custom and exclusive 'DIYdrones' firmware that allows the unit to output an efficient and very compressed binary protocol. You can still change between NMEA and Binary protocol with standard MTK messages, and switch the refresh rate between 1hz to 10hz, or set any standard serial baud rate (by default is set to 38400 bps and custom binary protocol)"(Payne, "MediaTek…"). More on the GPS UART interface is addressed in the MCU design section under the code subsection.

3.7 MCU Pins

The MCU that we have chosen for this project is the ATxmega64A1 series in a 100-lead TQFP package. This package was chosen for its simplicity of use and the limited available packages for this chip. This is a very powerful chip that has many features that satisfy our project specifications and allow for design modifications if needed.

The ATxmega64A1 series feature ports A through R. Each port has multiple functions that can set depending on the needs of the user. Since this design uses multiple systems each port will be configured to do one subsystem. Port A and B will specifically be used for the ADCs. Port C will be used for programming. Port D will be used to produce the PWM signals required to control the ESCs. Port E and F will be used for USART communication with the GPS and communication module. The reason for separating the subsystems into different ports is for simplicity.

The MCU will be programmed using the AVRISP MKII In-System Programmer. This programmer uses the serial peripheral interface to download code into the MCU. The IDE for programming the MCU will be the AVR Studio 4.16. This is a free IDE from Atmel that allows for debugging and programming of Atmel chips. AVR Studio will also compile C-code into machine code or assembly code. The pins that will be dedicated to programming the MCU are 23, 24, 90, and 30-32. All of these pins are located on Port C of the MCU. Pin 23 and 24 are ground and the positive voltage respectably. The positive voltage must be between 1.6V – 3.6V. Pin 90 is the RESET pin on the MCU. The other pins needed for programming the MCU are the SCK pin (pin 32), the MISO pin (pin 31), and the MOSI pin (pin 30). These six pins are combined into a single 2 by 3 pin header that allows for quick in-system programming. If needed there is another standard for in-system programming that feature 10 pins instead of 6. It is arranged into a 2 by 5 header. This is shown in figure 33.

Stephen R. Smith 9/26/10 5:43 AM
Comment: fig #s

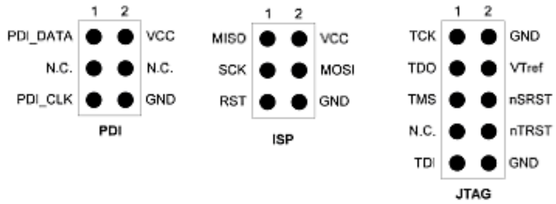

Figure 33: PDI, ISP and JTAG connectors
Image used with permission from Atmel

The ATxmega64A1 series features two on chip ADCs, eight 16-bit timers, and up to eight USARTs that are on eight different ports. Table 16 shows the layout of the chip with the pins and their functions labeled.

Pin #	Description	Pin #	Description
1	PA6, 5V ADC	34	VCC
2	PA7, 5V ADC	41	PE6, RXD1
3	GND	42	PE7, TXD1
4	AVCC	43	GND
5	PB0, 3V ADC	44	VCC
6	PB1, 3V ADC	51	PF6, RXD1
7	PB2, 3V ADC	52	PF7, TXD1
8	PB3, 3V ADC	53	GND
9	PB4, 3V ADC	54	VCC
10	PB5, 3V ADC	63	GND
11	PB6, 3V ADC	64	VCC
12	PB7, 3V ADC	73	GND
13	GND	74	VCC
14	VCC	83	VCC
20	PC5, MOSI	84	GND
21	PC6, MISO	90	Reset (Active Low)
22	PC7, SCK	93	GND
23	GND	94	AVCC
24	VCC	95	PA0, 5V ADC
25	PD0, OC0A	96	PA1, 5V ADC
26	PD1, OC0B	97	PA2, 5V ADC
27	PD2, OC0C	98	PA3, 5V ADC
28	PD3, OC0D	99	PA4, 5V ADC
33	GND	100	PA5, 5V ADC

Table 16: Pin Overview

The two ADCs are multiplexed into 8 pins each for a total of 16 ADCs. Ports A and B are dedicated to the function of the 16 ADCs. Port A consists of pins 93-100 and pins 1-2. Pin 93 is the ground and pin 94 is A_{VCC}. The voltage on pin 94 in is extremely important because it is the reference voltage for all of the ADCs on port A. Port B will be 3.3V for all sensors that require a voltage of 3.3V. Port B consists of pins 3-12. Pin 3 is the ground and pin 4 is A_{VCC}. The voltage on pin 4 is extremely important because it is the reference voltage for all of the ADCs on port B. Port B will be 3.3V for all sensors that require a voltage of 3.3V. Pins 5-12 are the eight ADCs that are associated with port B.

The eight 16-bit timers are separated across four ports. The design requires one timer with four output compare registers. Port D meets the requirement by having a 16-bit timer with four output compare registers. The output compare registers are on pins 25-28.

The USARTs that will be used for this design are ports E and F. On port E the transmitting line is pin 42 and the receiving line is pin 41. On port F the transmitting line is pin 52 and the receiving line is port 51. Table 16 shows the layout of the chip with the pins and their functions labeled.

3.7.1 MCU Code

The MCU is going to be coded the programming language C. With the help of AVR Studio the code will be converted into assembly language and then into machine code. Depending on the individual, MCU code can be organized like C code. The global variables first followed by declarations and function prototypes. Next is the main function followed by the functions. When developing code for the MCU, specific variables must be used, so the compiler can decoded it into assembly/machine code. Also when setting flags either binary, represented as 0b, or hexadecimal, represented as 0x, must be used.

3.7.2 PWM

In order for PWM to work a 16-bit timer can be used to produce the correct signal. A 16-bit timer has the ability to count from 0 to 65,535. Using an equation provided by Atmel, the top value and prescaler settings can be determined for setting the timer/counter control registers. The top value is the value that the timer will count too and either reset to 0 or start counting down to 0. The equation is as follows and it can be rearranged into a function of TOP:

Stephen R. Smith 9/26/10 5:49 AM
Comment: pov

$$F_{OCnxPFCPWM} = \frac{f_{clk\_I/O}}{2 \times N \times TOP} \qquad \textit{Therefore:} \qquad TOP = \frac{f_{clk\_I/O}}{F_{OCnxPFCPWM} \times 2 \times N}$$

Stephen R. Smith 9/26/10 5:48 AM
Comment: something

Where $f_{OCnxPFCPWM}$ is the frequency of the desired PWM, $f_{clk\_I/O}$ is the system clock, N is the prescaler value (N can only equal 1, 8, 64, 256, 1024) and TOP =

top described above. The frequency of the desired PWM is 50 Hz. Table 17 below shows the values of this function.

$f_{OCnxPFCPWM}$	$f_{clk\_I/O}$	N	TOP
50 Hz	32 MHz	1	320000
50 Hz	32 MHz	8	40000
50 Hz	32 MHz	64	5000
50 Hz	32 MHz	256	1250
50 Hz	32 MHz	1024	312.5

Table 17: Values of the PWM function

Since the value in the TOP must fit into a 16-bit register N = 1 and N = 1024 are not valid because 320000 is too big for a 16-bit register and 312.5 is not an integer value. So the value of N = 8 is sufficient and it will fit into a 16-bit register. So the value that will need to be stored into top will be 0x9C40. This can be done with the following line of code:

```
IRC3 = 0x9C40;          // sets the top value to 40000
```

The number 3 in the variable name is the timer number. The next values that will need to be determined is the range that the output compare register will need to be set in order to produce a signal of 1 ms to 2 ms for the PWM. Since 50 Hz is equal to 20 ms we can calculate that 5% of 20ms is 1 ms and that 10% of 20 ms is 2 ms. For the group to determine the range for the output control register the percentages can be multiplied to the top value. Therefore 5% of 40000 is 2000 and 10% of 40000 is 4000. For a PWM signal to produce in the proper range the output control register needs to be set to or in-between 2000 and 4000. Producing the correct signal can be done by setting the desired output control registers as follows:

```
ORC3A =  0x7D0;         // sets the output control register to 2000
ORC3B =  0xFA0;         // sets the output control register to 4000
ORC3C =  0xDAC;         // sets the output control register to 3500
```

These values are global variables and can be modified at any time. The timers are a hardware solution, so heavy computations by the CPU will not slow down the timers.

For the copter to move or stabilize the speed of the motors will need to be changed. The simplest way to do this would be to have four functions to modify each output compare registers. Each function will have a return type of void and an unsigned 16-bit integer as the input. The prototypes for the functions are below:

```
void update_PWM_ORC3A(uint16_t ORC3A_change);
void update_PWM_ORC3B(uint16_t ORC3B_change);
void update_PWM_ORC3C(uint16_t ORC3C_change);
void update_PWM_ORC3D(uint16_t ORC3D_change);
```

Inside the function, there will be an error check to make sure that the MCU is only producing a value from 2000 to 4000. If the value is more than 4000 then the value will be truncated to 4000 and if it is under 2000 then the value will be truncated to 2000. There will be certain situations that will send off an error code to the user, like if the linear control system returns a value less than 0 or more than 6000. An error code such as beeps or tones can be used to signal that there is a problem.

3.7.3 ADC

Programming for the ADC is a little confusing because there are two ADCs that are both multiplexed (8 to 1 multiplexer). There will need to be a function in main that polls all of the data and stores it in memory. This will be stored into a variable called *uint12_t* ADC_Value[16]. By using an array we can easily use a for loop to cycle though all of the values that need to be polled. We also need another variable named *uint1_t* data_Done. These variables will signal that all of the values from the sensors are ready for the linear control system or computing basic movement. There are some other variables what need to be set for proper function of the ADCs. The ADMUX (ADC Multiplexer Selection Register) will need to be set a different value every time a particular sensor is polled. This variable sets the multiplexer and sets the reference voltage. The multiplexer will be set to zero if the sensor attached to the zero pin needs to be polled. This can be accomplished by setting the variable as follows:

```
ADMUX |= 0x00;          // sets the multiplexer to zero
ADMUX |= 0x40;          // sets the reference voltage to AVCC
```

The top declaration is done every time the function polling the sensors is called. The bottom declaration is a global variable declaration and will be put in the beginning of the code. Another variable that needs to be set is ADCSRA (ADC control and status register A). This variable sets the prescaler for the ADCs, sets the ADC enable bit, sets the ADSC bit, which signals that an ADC conversion needs to be done and checks the ADIF bit, which signals when the conversion is complete. The prescaler needs to be set to its maximum because the slower the ADC the more accurate it is. The following function is called at the beginning of the code to initialize the DCSRA register.

```
void initADC()
{
        DCSRA |= 0xC7;          //sets the ADC enable and prescaler to 128
}
```

This is the primary function that will be used to convert analog signals to a digital number. The input to the function is the current ADC pin that needs to be converted. The code will sit in this function until the conversion is complete.

```
unint16_t readADC(uint8_t adc_channel)
{
         adc_channel &= 0x0F;          //Use only lower four bits
```

```
        ADMUX |= adc_channel;              //Sets the mux to correct value
        ADCSRA |= (1<<ADSC);               //Signals for ADC to start
        while(!(ADCSRA & (1<<ADIF)));      //wait for conversion to complete
        ADCSRA |= (1<<ADSC);               //Unsets the start ADC signal
        return ADC;                        //returns the converted value
}
```

This block of code will be in the main loop of code. It will poll all of the sensors from the readADC() function and place it into the correct position of the array. When this block of code has completely finished polling all of the sensors, it will set the LSB of the data_Done variable.

```
if(data_Done == 0x00){
        for(uint8_t i = 0; i < 16; i ++){
                ADC_value[i] = readADC(i);
                if (i == 15){
                        data_Done = 0x01;       }}}
```

This function passes all of the data from the readADC() into the proper position in the array ADC_value[]. It also sets the value data done if it has just finished the last sensor. The for loop can only start if the data_Done is not set, this is done because data shouldn't be overwritten if it has not been read first.

Stephen R. Smith 9/26/10 5:56 AM
Comment: josh

3.7.4 USART

The group will be using USART to communicate with the GPS and the custom communication link. There are several variables the need to be declared before code can be described. The first one is the UCSRA (USART Control and Status Register A). This register has control bits for many functions including the RX completed and TX completed bits. When the data has been sent or received the these bits will be set. The next register is the UCSRB (USART Control and Status Register B). This register contains the Rx complete interrupt enable bit, the Tx complete interrupt bit, the Rx enable bit and the Tx enable bit. The next register is the UCSRC (USART Control and Status Register C). This register contains the URSEL (USART register Select) bit, the UMSEL (USART Mode Select) bit, the USBS (USART Stop Bit Select) bit and the two LSB UCSZ bits. All of these bits need to be enabled. This can be done with the following three lines of code:

```
UCSRA |= 0xC0;        // Enables RXC, TXC
UCSRB |= 0xD8;        // Enables RXCIE, TXCIE, RXEN, TXEN,
UCSRC |= 0xCE;        // Enables URSEL, UMSEL, USBS, UCSZ1, UCSZ0
```

There is also the UBRR (USART Baud Rate Register) which is determined using the equation described in the MCU Code Research section (Page XX). This value will be set to 51 for simplicity and for a lower error percentage. This can be done with the following line of code:

```
UBRR = 0x33      //Sets the baud rate register to 51
```

To better organize the code all of these declarations can be stored in a single void function and just be called to initialize all of them at once. Interrupts are going to be used to save the CPU from having to wait for data. This will be done by using an ISR with the correct interrupt vector in the field. Once the data is done transmitting or done receiving the ISR will trigger and place the data into some field where the CPU can access it whenever it need to. The following code fragment shows the data fields and the two ISR functions:

```
char received_char;
char transmitted_char;

ISR(USART_RXC_vect){
        char received_char;             //declaration of a user variable
        received_char = UDR;    //places the char in the data
)                                        //register into a user variable
transmitted_char = ‘q’;         //char to be transmitted
ISR(USART_TXC_vect){
        UDR = transmitted_char;         //places the char into the data // register
}
```

The transmitted_char variable needs to be updated every time so that a valid bit can be transferred. If not then the value that is in the variable will be transmitted repeatedly. Also the ISR will work only if the global interrupt enable flag has been set. This can be set by inserting the following line of code:

```
sei();          //sets the global interrupt enable flag
```

3.8 Video System

The design of the video system will be mainly concern with the mounting on the quad-copter, and power requirements. The video system only weights 9 g which won’t put hardly any strain on the maximum weight of the quad-copter. The Video system will be placed right under the quad-copter. A lot of components are already going on the quad-copter and the video system should be mounted where it won’t affect the balance of the quad-copter. The power requirement of the quad-copter will be satisfied with own power supply. This power supply will be a 9 V battery. The receiver of the video system will have its power requirement fulfill by a wall outlet and an AC to DC adapter that comes with the video system.

The receiver will also be plug into a TV or any display that can take video composite cables. In the case of the quad-copter this might be a laptop with a video capture card in it.

3.9 Aeronautics

The motor to be used is the TowerPro 2410-09Y, based upon its low speed constant, lower current requirements, and its mass of approximately 55g. To help moderate control to the motor and simplify design, this will be purchased with the companies recommended ESC, due to time and cost. For the maximum lift and thrust, the 10in.X 4.5in. propeller blades will be used, as recommended by TowerPro and its sales distributers. The motors will receive their power from the ThunderPower 11.1V 30C 5500mAH, based on the availability of the batteries selected.

The best way to keep the copter from spinning is to have the propellers move the propellers in opposition to each other. The top and bottom propellers will be used to spin counter-clockwise, while left and right propellers are used to spin clockwise. The camera will be facing between top and right arms, thus designating the front. The object-detecting ultrasonic sensor will be pointing towards this front end, while the ground detector will be facing below it. This is shown in figure 34 on the next page

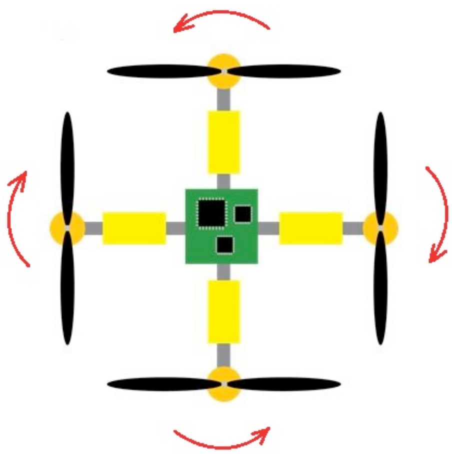

Figure 34: Basic Quad-copter Design

4 Design Summary (revised)

The final quad-copter differed in some respects to what was put forth in the design section. Those changes will be addressed here in the design summary.
The quad-copter consists of several subsystems some of which are more interwoven into the design, such as the MCU, and some that are more isolated, for example the video system. In fact, the video system was not even implemented into the final product. The block diagram in figure 35 below provides

an overview of the quad-copter's subsystems. In general, design decisions were based on such factors as, the minimization of cost, time, and weight, however; the decision to design a custom wireless system did not follow this trend, as it was made for the purpose of instilling greater design complexity into the project. The custom wireless system however, had to be abandoned and replaced with an Xbee module.

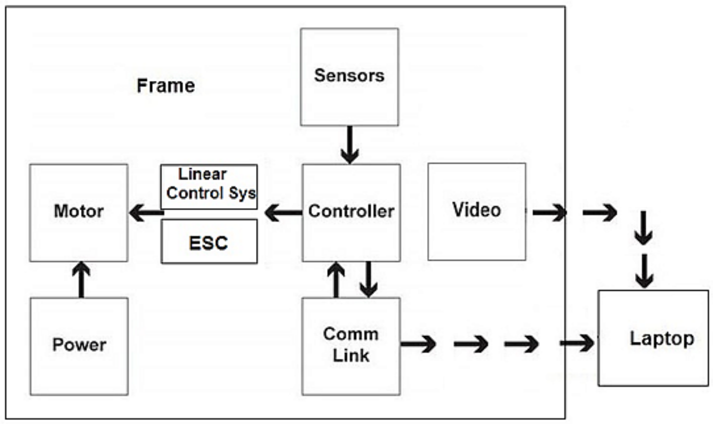

Figure 35: block diagram of quad-copter subsystems

4.1 Power

Power had been divided into two separate parts: the motors and the main components for operation and control. This was found as the best solution to minimize noise and to protect the main board from unforeseen problems based on inexperience with PCB design. Since the majority of the required power needed to be drawn was consumed by the motors, the best solution was to directly connect the lithium polymer (LiPo) batteries to the motors. Since the biggest concerns regarding the LiPo batteries were mass and cost, the best route to minimize both of these issues was to select either one very large battery or two smaller ones. This design will implement the latter. The Esprit EM-35 3-cell 35C 2250mAH were an excellent source in terms of mass, balance, and charge capacitance. The batteries were to be directly connected to the ESC, which were rated at 30A per ESC. From preliminary testing, a 5 minute window at the maximum setting was allowed, which reinforces the minimum flight time. The main power source for the main board and the remaining peripheral components were powered using alkaline batteries; a 9V battery for the video system, and a 9V battery for the main board and the remaining external components. The lowest power to be regulated on the board was implemented at 2.98V by an LM317-ADJ. This was the main power to the gyroscope, since the component's maximum voltage tolerance is 3.3V. Using this level of voltage will safely guarantee the use of component. Figure 2 depicts the schematic implemented.

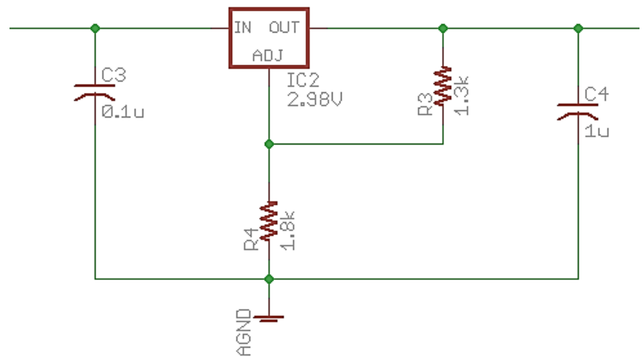

Fig. 2. LM317-ADJ setup for 2.98V

For the majority of the on-board components, there were two sources used. The first was the LD1117V33, an LDO regulator fixed at 3.3V. This regulator was essential for the operations of the on-board level logic converters, enabling the GPS unit, and powering the Zigbee module for the digital component; and it was powering the accelerometer and ultrasonic range finders to optimize analog signals sent to the ADC ports. Two separate versions of this were used: one analog and one digital, to isolate noise. The second regulator was a dedicated regulator to the GPS, the digital compass, and the MCU, which was required to operate at 5V for optimized processing speed for the MCU. The LM7805 was the recommended choice due to its availability and its reliability. The GPS module has an on-board regulator at 3.3V LDO, and requires no further modifications in terms of power, but must be considered in signal analysis.

The MCU would always emit a voltage of 5V as the voltage high when transmitting data to all I2C and UART devices. For the digital compass, this was not an issue, since it to will transmit and receive data to the MCU using 5V as a logical high. The GPS and the Zigbee module are powered at 3.3V, and would transmit and receive at a maximum logical high of 3.6V, each. To protect and preserve these devices, as well as meet the minimum specifications of 0.7*VCC for the MCU, level logic converters were placed on board to three UART ports. The third UART port acted as an auxiliary port to the MCU, and allowed for an alternative device to be implemented at a later point. The level logic converter is composed of two 10 kOhm pull-up resistors, one for 3.3V and one for 5V, as well as an Infineon BSS138 used to act as a switch between the RX line of the MCU to the TX line of the connected component. The MCU TX line is connected to a voltage divider which outputs to the RX line of the component. Figure XX demonstrates the design to be implemented

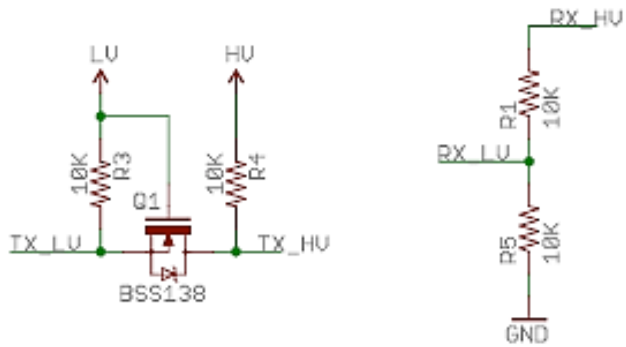

Figure XX. Schematic for the level logic converter
Provided with permission from Sparkfun.com

4.2 PCB Design Layout

The board was decided to be two-layered, due to its simplicity and cost. This allowed all group members to be able to easily trace any one component to another. This also allowed for an easier comprehension of connections from any line on the board to all surface mounted parts, which are the IDG500, ADXL335, ATMega2560, BSS138 MOSFETS, and the ferrite beads. All other components were set as through hole for the sake of quickly setting up and testing a board with minimal risk to the other components. This had the consequence of occupying a lot of PCB real estate, and so a board size of 4” X 4” is used.

During initial design, a constant was to keep power and signal lines on the analog and digital design were to remain separate. This allowed for the board to be split into an upper hemisphere for analog components; and a lower hemisphere to occupy all digital components. Power from the 9V battery was immediately split to four separate sources, due to dropout effects in the voltage regulators. All on-board regulators used a TO-220 packaging due to the simplicity of the parts, as well as the minor difficulty to acquire said parts. An LED was directly attached to the batteries lines to determine if power was running through the on-board switch and to the main board.

Power was distributed based on the needs of the components. Since no ground planes are being used, the power lines for all digital regions were reinforced for a larger width of 16 mil width. This allowed for power and ground lines which must travel a larger distance to be able to handle larger loads, as well as small AC loads, which is the case for the PWM lines. The MCU’s main power was being distributed by a power distribution method known as the star method, where the voltage and the ground were radiated outward from a central point and evenly distributed to the MCU components. This allowed for the safest means of distributing power in the case of a short or burnout of the chip, with minimal risk to all the on-board components. This method was also used for distributing a ground to the PWM ports and the XBee module. Power for the analog hemisphere was kept to the minimum specification of 6 mil width, due to its low power consumption.

Due to the small region the ADXL335 and the IDG500 occupy, the analog side was minimized in its occupied space, which allowed for more real estate needed by the digital side. This created an effect of the IMU components being off center by as much as an inch from the center of the plate. This effect were minimal in terms of gyroscopic measurements, as well as feedback to be given from accelerometer. There were also 5 ADC auxiliary ports which have been added to add other analog components for maintenance with breakout boards, and for future components to be placed.

Data lines going into the MCU from the analog sources were kept separate from the digital sources. The ADC pins all faced the data lines from the analog components for simplicity. Due to the architecture of the chip, all of the ATMege2560's digital lines were placed away from the analog lines, which simplified separating power and signal lines around the ADC. Power and ground lines were strategically placed to ensure no issues in crosstalk between the analog and digital lines, as well as minimized and separate noise between the analog and digital hemispheres.

To interface and program the MCU, a 2X3 port was established near the chip as a means of communicating and reprogramming the on-board ATMega2560. This was essential for prototyping and testing the PID, modifying functions involving the range finders and the GPS, and for presetting values into the MCU for demonstration and production purposes.

All PWM ports were located as close to the MCU as possible, allowing for the best quality signal to be sent to the ESCs. Each one had been labeled in accordance to compass directions: NORTH, SOUTH, EAST, and WEST. This acted only as a reference point to each PWM, since during the design stage, no dedicated front was decided. The only on-board UART device was the 1mW XBee module. The design of the module used smaller through-hole pin outs, at 2mm distance from each neighboring pin. The level logic converter was placed close to the UART lines as possible, and the level logic converter, like the other level logic converters for all other components. Currently, the MCU was using an internal crystal to handle all clocks and timers. However, a space for a 2-pad crystal was available for using an external oscillator, with pin outs for two ceramic capacitors.

Components which were to be mounted to plate near the board were the digital compass and the GPS. Orientation and placement were the key to operating these devices. The compass had a magnetic limit of 20 gauss, and was affected by the motors, since brushless motors are a form of permanent magnet motor. Keeping this relatively close to the board yielded the best results for controlling the yaw and heading of the Quad-Copter. The GPS antenna was placed pointing toward the sky for any data to be retrieved. Optimization of this data allowed for the highest accuracy for location.

4.3 Microcontroller

The group decided to choose a MCU for simplicity of use and due to its built in peripherals like timers and ADC converters. The Atmega2560 is the MCU that the group will use. It has four 16-bit timers, sixteen 10-bit ADC, four UART and I2C compatibility. The MCU will interface with quad-copter subsystems using PWM, ADCs, USART, and I2C which are shown below in figure 39.

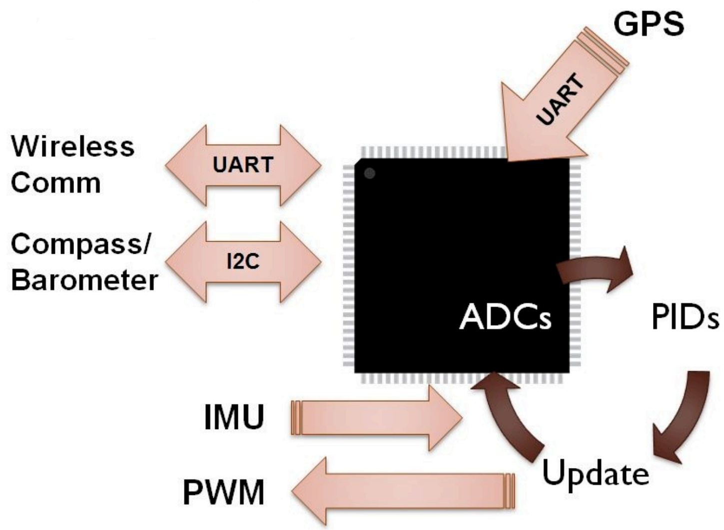

Figure 39: Overview of MCU interfaces

The MCU will use PWM to control the motors. The PWM signal will be generated by using the MCU's 16-bit timers and the output compare registers. The PWM initialization function will set the timers maximum to a value to properly have the signal modulate at 50 Hz. By varying the value of the output compare register, the speed of the motors can be increased or decreased. The MCU will further interface with quad-copter systems using its ADC converters, USART, and through I2C.

The MCU will use the ADCs to retrieve the raw analog data from the various sensors. The ADCs will be set up so that a conversion will be made every 250 KHz. The ADCs have a resolution of 10-bits, which allows the sensors to be represented by a value between 0 and 1023.

The MCU will use UART to communicate with the wireless communication unit and the GPS unit. The UART protocol was chosen for its ease of use and ease of setting up. The first UART will be connected to the wireless communication unit and the data will travel both ways. This is so control can be sent to the quad-copter and the quad-copter can send debugging information back to the user. The second UART will be connected to the GPS unit and the data will only travel from the GPS unit to the MCU.

The I2C bus will be used to communicate with the digital compass and the pressure sensor. The compass will be responsible for sending accurate heading data to the MCU so that yaw adjustments can be made.

4.4 Sensors

The quad-copter requires several sensors to perform tasks that range from critical, such as flight stability, to optional, such as the high altitude sensor. Additionally, sensors are an important part of the quad-copter's autonomous functions such as Altitude maintenance, path finding, and object avoidance. The different sensor systems can be organized as follows:

- Flight Stability Sensors
- Proximity Sensors
- High Altitude Sensor
- Direction/Yaw Sensor
- Navigation/Location Sensor

4.4.1 Flight Stability Sensors

The flight stability sensors are a critical system for the quad-copter to remain in flight. The system consists of a triple axis accelerometer and a dual axis gyroscope combined into a 5 DoF IMU. The accelerometer is the ADXL335 from Analog Devises, and the gyroscope is the IDG500 from InvenSense. The outputs from the sensors will be combined using a sensor fusion algorithm which outputs an improved estimate of the angular state. The output of the senor fusion algorithm is the input to the linear control system which adjusts the speed of each motor to maintain a level hover.

The ADXL335 is an analog sensor that can perform within a range of +/- 3 g, and is configured for a bandwidth of 50 Hz. It was supplied at 3.3 V. The accelerometer is a surface mount component which was mounted on the main PCB. Figure 40 on the following page shows the layout for the ADXL335.

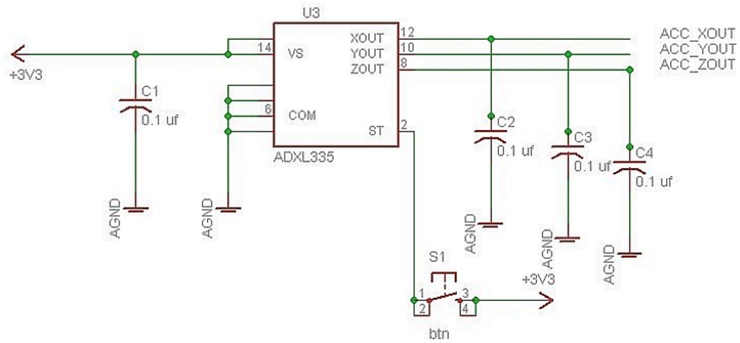

Figure 40: ADXL335 PCB layout

The 0.1 µf cap at C1 filters the supply line, and the 0.1 µf caps at C2, C3, and C4 filter the bandwidth to 50 Hz. The X, Y, and Z outputs connect to the A/D converters on the MCU.

The IDG500 dual axis gyroscope is also an analog sensor. It can effectively measure +/- 500/deg/s of angular rotation with a sensitivity of 2 mV/deg/s. Like the accelerometer, it is a surface mount component that was mounted on the main PCB. The layout of the IDG500 is shown below in figure 41.

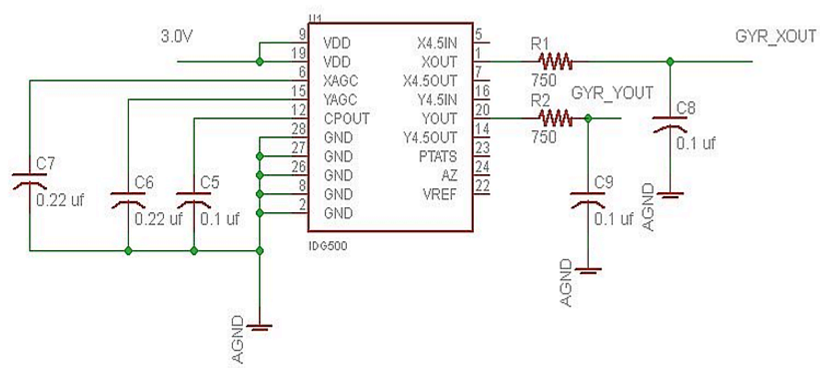

Figure 41: IDG500 PCB layout

The IDG500 was supplied at 3.0 V. The X and Y outputs are filtered and then connect to the A/D converters on the MCU. Caps C5-C7 are for internal regulation.

The sensor fusion algorithm is a simplified version of a Kalman filter. It is simplified because the relative weights of the accelerometer and gyroscope will be constant rather than fluctuating. However, the constant can be easily adjusted to optimize performance of the quad-copter. The algorithm works by using the accelerometer readings to get the present state of the roll and pitch, in contrast; the gyroscope readings are used to account for the dynamics of the angular velocity. The two readings are averaged to form an improved estimate. Optionally, the improved estimate can be weighed against past estimates and further improved. The final design of the IMU algorithm was dependent on testing to determine how sophisticated it has to be. After prototyping several algorithms, including a Kalman filter, a degree of stabilization was achieved that allowed the quad-copter to hover in a controlled manner. The goal for the quad-copter was, after all, steady flight and not aerobatics.

4.4.2 Proximity Sensors

Proximity sensors were to be used for two distinct purposes on the quad-copter: a downward oriented sensor to detect the distance to the ground, and a forward oriented sensor to detect obstacles such as trees and walls. Both sensors were ultrasonic range finders, specifically, the MaxSonar LV-EZ2s from MaxBotix. Both sensors would have been necessary for any sort of autonomous flight protocols such as object avoidance or automatic altitude control. The LV-EZ2s have a maximum range of about 20 ft. for a large object such as a wall, however; this range diminishes significantly when detecting smaller targets. Both of the LV-EZ2 ultrasonic sensors will be remote from the main PCB and connected with 6-12 in. of wire to header pins. The PCB layout of the 3 header pins for the ultrasonic sensors is shown below in figure 42.

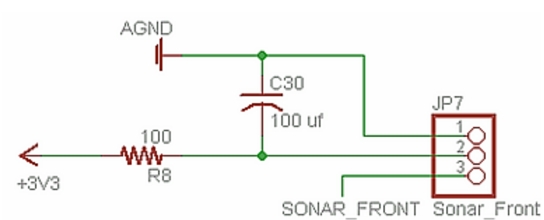

Figure 42: PCB pinouts for the LV-EZ2 ultrasonic sensors

The two ultrasonic sensors were bought, tested, and integrated into the code of the microcontroller, however; a late problem arose in programming the secondary bank of A/D converters in the MCU. This problem was partially solved by using an auxiliary microcontroller that communicated A/D data to the primary MCU via UART. This solution however was limited in its effectiveness and resulted in being able to use either the ultrasonic sensors *or* the compass module at a time.

4.4.3 High Altitude Sensor and Digital Compass

The high altitude sensor system would be used to determine altitude when the quad-copter is beyond the range of the proximity sensor. This sensor system is of secondary importance as the quad-copter could simply be programmed with an automatic decent protocol when it goes too high. The original design made use of a dual function component that combines a pressure sensor, for altitude measurement, with a digital compass. This component was the HDPMO1 from Hoperf Electronic. The HDPM01 was a cost efficient solution that combined the two sensors onto a small breakout board. Drawbacks of the HDPM01 were that the I2C interface seems particularly difficult to decode for a first time user, and this was compounded with poorly written documentation. The HDPM01 would have been remote from the main PCB and connected with wires to 6 header pins, the original layout is shown below in figure 43.

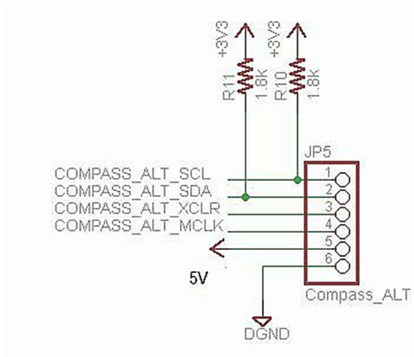

Figure 43: HDPM01, combination pressure sensor and digital compass

Due to the reasons mentioned above, the HDPM01 was not used in the final design. Instead, the barometric sensor was abandoned as unnecessary, and the HMC6352 two-axis magnetometer from Honeywell was used instead to measure yaw. Yaw is the movement about the vertical axis of the quad-copter. Yaw must be stabilized as a requirement for attaining a stable hover. Yaw can be manipulated by increasing the speed of two motors along a single axis while simultaneously decreasing the speed of the motors on the other axis. This will rotate the quad-copter in place while maintaining a net equilibrium on the vertical axis. This change to the yaw can be initiated either directly by the user giving a wireless command or autonomously by the microcontroller using sensor data from a digital compass. The digital compass used for this purpose is the HMC6352 which communicates with the microcontroller via a two-wire serial interface. The compass heading can be used as a component input of the stabilization loop to maintain a stable heading. Furthermore, the compass could have been used in conjunction with the GPS module to autonomously plot movement to a GPS coordinate in theory but not in implementation.

4.4.4 Navigation/Location Sensor (GPS)

A GPS module was integrated into the design of the quad-copter which would have been a central component of the autonomous mode of operation, had the quad-copter achieved enough flight stability. The GPS system would have allowed the quad-copter to hover in place by repeatedly returning to a point of origin, or to move towards a given coordinate. The MediaTek MT3329 GPS module was used. The MT3329 has an antenna integrated into the casing of the chip which is an optimal design for the quad-copter. It has a positional accuracy of within 3 meters and a sensitivity of up to -165 dBm. The MT3329 also has coding and firmware support available from the DIYdrones website. Originally, the plan was to mount the chip directly onto the main PCB, subsequently; the design changed to the GPS module being mounted on a breakout board and wired to the main PCB with header pins. The main reason for this design adjustment is that with a breakout board, the module can be easily tested or attached and detached from the prototype board without necessitating buying a second module.

The MT3329 interfaced with the MCU using a USART connection. Coding protocols for the USART interface is covered in the MCU design section. The GPS adapter module was connected to the main PCB with an EM406 6 pin connector. The layout for the EM406 is shown below in figure 44.

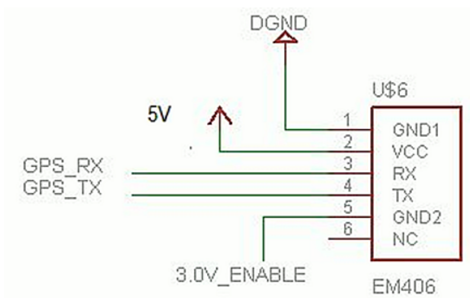

Figure 44: EM406 connector for the MT3329 GPS module

In the diagram, the Rx and Tx lines connect to the MCU. The supply is 5 V, which the adapter regulates down to 3.3 V. the enable pin is set to 3.0 V to engage the system.

4.5 Linear Control System

The linear control system was responsible for stabilizing the quad-copter. This was perhaps the most crucial system of quad-copter as it enabled the quad-copter to fly stable. The linear control system consists of a discrete PID controller designed in software with integer math. The PID receives an output from the sensors to determine if the quad-copter needs to be stabilized or not. The design of the system is as follows.

In order to make the PID controller equation a discrete the integral term and the derivative term has to be in discrete form. For the derivative term a backward finite difference form was used. For the integral term a summation of all previous error was used. The error signal was a discrete function where each error is sampled and stored in an array. The derivative term calculates the difference between two consecutive error samples and divides it by the sampling period, while the integral term will do a summation of the array and multiply it by the sampling frequency. The discrete equation looks like this:

Discrete equation: $x(n) = K_p e(n) + K_i \sum_{K=0}^{n} e(k) + K_d [e(n) - e(n-1)]$

K_i term: $K_i = \frac{K_p T}{T_i}$ *K_d term:* $k_d = \frac{K_p T_d}{T}$

Where, T is the sampling period and T_i the integral time constant. The PID controller uses 3 functions in order to initialize, calculate the PID controller, and reset the PID controller.

The first method is called initPID. It has as augments the desired output, the measured output, Kp, Ki, Kd, and a structure that holds the current status of the PID controller. The initPID is called as follows, initPID(desired output, measured output, status structure). The status structure contains the values of the last measured error, Kp, Ki, Kd, the summation of the errors, the maximum error allowed, and the maximum summation error.

The initPID method initializes the last_measured_error, and summation_error to zero. It also sets the PID gains to the ones decided by the user. Based on the gains initPID calculates the max_error and max_summation error.

The second method is called resetPID. It is responsible for setting the parameters of the PID controller back to zero. The resetPID method uses the same status structure as the initPID function and sets all values in the structure to zero.

The third method is the one that implements the PID controller. The method is called PID_Controller. The PID_Controller method has as arguments the desired output, the current measured output, and the status structure PID_Status. This function does the integer math needed to calculate the appropriate output. The function takes the difference of the desired output and measured output to calculate the error. This error then is multiplied by the Kp_Gain in order to get the proportional term. The error will is added with all the previous errors and multiplied by the Ki_Gain to get the integral term. Lastly the error is then subtracted from the previous error and multiplied by the Kd_Gain in order to get the derivative term. Once all the PID terms are calculated, they are added together and used as the output.

4.6 Wireless Communication

4.6.1 Custom Wireless System - Xbee

The wireless communication system was going to be designed with a custom solution. The solution was to use a transceiver and a microcontroller to make a custom board and develop a protocol for the device. Unfortunately due to some difficulties in getting basics commutation between the transceiver, CC2520, and the MSP430 microcontroller the custom solution was scraped.

The difficulty came about not being able to communicate to CC2520 with the MSP430. The MSP430 and CC250 were placed on separate break out boards in order to test basics functionality. This was also done to verify that the circuit for the PCB was right. After much testing and rewiring it was determined that either the soldering was bad on the QFN part, CC2520, or the CC2520 was bad it's self. We little time to figure out the problem a decision was made to scrap the custom solution and use an Xbee instead.

The Xbee module is a ZigBee compatible device, which was researched and considered as the solution for the wireless communication system. From the research it was shown that ZigBee meets all the requirements of the wireless communication system. Xbee will be used to control and get status messages from the Quad-Copter. Using Xbee simplifies the design of the system substantially. The Xbee module will be connected to the main microcontroller, the ATmega2560, through UART ports. Four pins is the minimum needed in order to communicate and send wireless data with Xbee. Those four pins are VCC, GND, DIN and DOUT. VCC is a voltage supply at 3.3 volts. GND is the ground of the main PCB board. DIN and DOUT are the RX and TX of the UART connection respectively.

Xbee modules can be setup as end device or coordinator. In order to setup an Xbee module a minimum of four registers needs to be set in the Xbee memory. The first one is called the PANID. This registers sets the ID of the network that the module is going to join. In order to change the module's own address the MY register is used. If data is to be sent to another module the address of that module is needed in the memory. The register that sets that value is the DL and DH. The DL register is the lower 16 bits address destination address, while DH is the upper 16 bits. Since only two Xbee modules are going to be used the DH is going to be set to zeros and the DL will be the actual address. The last two registers needed are the A1 and A2 registers, which sets the way an end device joins a network and how a coordinator manages the network respectively.

The A1 register has 3 bits that sets how the end device joins a network. Bit0 decides whether the end device can joins a specified network or any network that the coordinator tell it to, if bit0 is set it will allow the end device to join any network that the coordinator decides is best. This bit is not set so that the end device is only on the network of the coordinator and not any coordinator. Bit1 decides if the end device will allow the coordinator to channel that the end device is talking on, if set it the end device will allow the channel to be change by a coordinator. This bit is set so that if a channel is too noisy the coordinator can talk on a new channel and tell the end device talk on that channel. Bit2 decides whether the end device's association with the coordinator is automatic or manual, if set the end device will try to auto associate with the coordinator. This last bit is set so that the association of the end device and coordinator is automatic and requires no programming. The end device was setup this way in order to have communication as automatic as possible, be able to avoid noisy channels, and minimize network confusion.

4.6.2 Video System

The video system was a non-essential to the basic function of the quad-copter, which is to fly. Due to the complications from the wireless commutation system, PCB board delays and errors, the video system took a back seat to getting the quad-copter to fly. In the end, the video system was scrapped and all the focus shifted to getting the quad-copter in the air and stable.

4.7 Frame

A lightweight frame was designed to support all the quad-copter subsystems. The frame was entirely aluminum for securing the. Carbon fiber tubing was not chosen due to it's fragile nature in testing, as well as its cost. A prototype frame was designed with a 6 in. X 6 in. square aluminum central plate with four radiating 20 in. aluminum struts which is shown in figure 45.

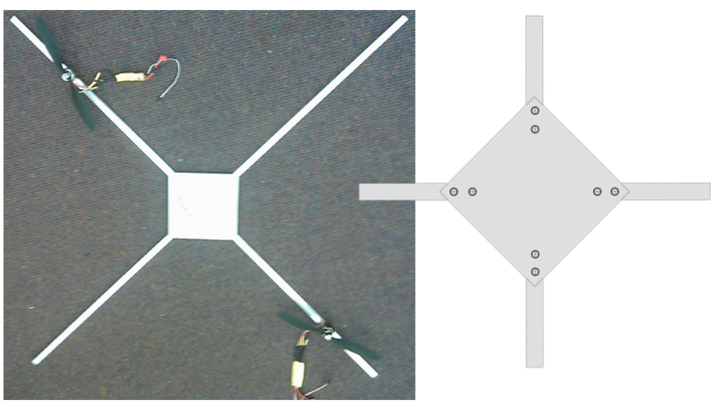

Figure 45: Aluminum prototype frame with close-up of central plate

The prototype frame was sufficient for testing the quad-copter's systems as they were developed. Modifications were made when motor testing on a single access began to twist the arms of the frame. The arms were replaced with cylindrical-shaped beams to prevent any further twists due to the motor's torque. The center plate was then perforate by a manual drill to help minimize weight, air friction, as well as to create holes for mounting the PCB. Four more rods, approximately 10 inches in length, were also attached as a means to hold the LiPo and 9V batteries, as well as the means to offset the center of mass. This offset acted as a means of simplifying testing for finding optimal PID values for the controller. The modification effected the decision of using helicopter like landing gear to taller legs attached to each end of the Quad-Copter's arms. Coat hangers were used as the landing gear, due to its dimension and its low mass. This addition created extra noise to the system, which effected all measurements from the accelerometer, the basis for the controller. Software modifications were made to account for this type of error, as will be discussed in the testing section. Extra damping material was used to help reduce overall noise, as well as kite string was attached to the four rods beneath the center plate. This allowed for the batteries and the rods to remain secure in one position, so as to keep them from shifting a great deal in testing a single axis and in flight. The final frame design is shown below in figure 46.

Figure 46: Aluminum prototype frame with close-up of central plate

5 Prototyping

5.1 Wireless Communication

Phase 1 of prototyping the wireless communication system was done using a custom board. The custom board will be the prototype for the final build. For prototyping, the custom board a PCB will be used. The reason for a using a PCB instead of a breadboard is that if the custom works this will be the board that will be used for the remote controller. On the down side is the custom board fails it will accumulate more cost and development time for a new custom board. The custom board will be using the design described in the design section for the wireless communication.

The custom board will be a small PCB board that will house all the materials that are shown in the bill of materials (section 7.2). A design consideration for the custom board will be to have all the wireless circuitry on one side and all the digital circuitry opposite of that. This will ensure that the digital circuitry will not interfere with the wireless circuitry. An SMA RP connector will be used in order to install a 50 ohm dipole antenna. The circuit is shown below in figure 47 shows how the MSP430F2616 and the CC2520 will be connected.

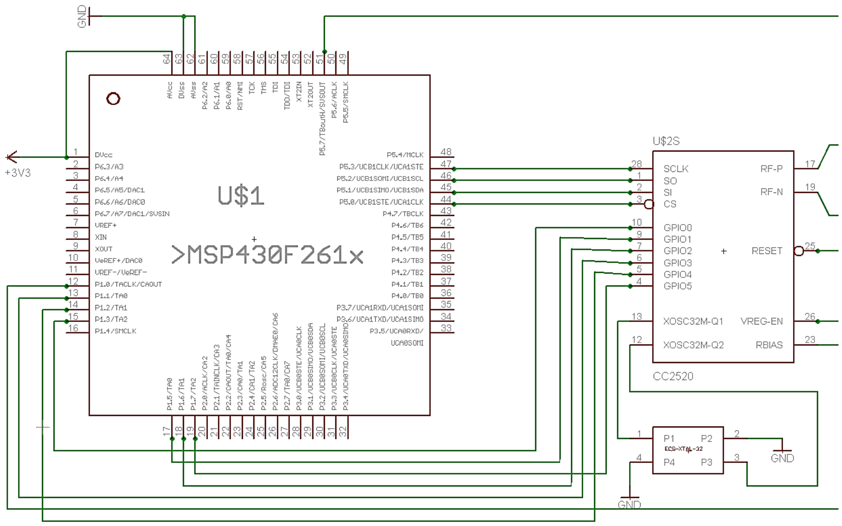

Figure 47: Interfacing the MSP430F2616 and the CC2520, the 32 MHz crystal to the CC2520.

Stephen R. Smith 12/13/10 9:13 AM
Comment: see eng

The voltage supply for the CC2520 will be connected as shown in figure 50. Also, the decoupling capacitor is shown in figure 48. The pins shown in figure 51 are part of the transceiver CC2520.

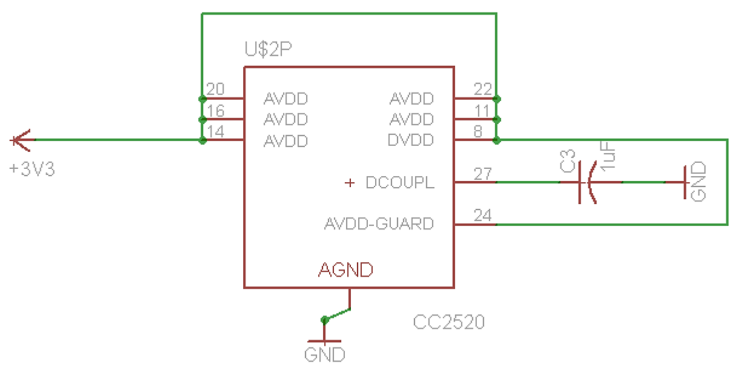

Figure 48: Power supply of the transceiver CC2520.

The final part of the wireless communication needed is the radio frequency circuitry. This circuit has the impedance matching circuit that matches the output impedance of the CC2520 and the 50 ohms dipole antenna, and the SMA RP connector for the dipole antenna. The circuit is shown in figure 49.

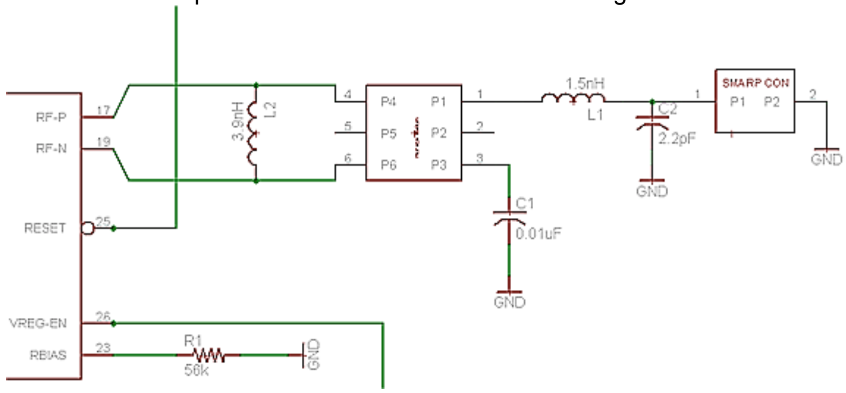

Figure 49: Radio Frequency Circuitry.

This is the prototyping circuit that will be used in order to test the wireless communication system. In order to a way to tell if the circuit is working, a couple of light emitting diodes will be connected to the MSP430F2616 general input output pins.

During the actual prototype the wireless communication ran into difficulty. The first difficulty was getting the MSP430 and CC2520 communicating with each other. The MSP430 and the CC2520 were placed on their breakout boards. The jumper wires were used to connect them with the SPI interface. Once this was done the SimpliciTI firmware was used to configure and test the CC2520. During testing it was found that the CC2520 was returning the wrong CHIP ID. Also the CC2520 would hang at random places of the code. After further testing it was concluded that the CC2520 was damaged or soldered wrong.

This difficulty brought the decision to use the Xbee instead of the custom wireless communication. The prototype for the Xbee was simple. Connect them to a computer configure them and test them. Once we had good communication one of the Xbee was placed on the Quad-Copter. Once we this was successful the Xbee was used to control the Quad-Copter.

5.2 Sensors

The sensors are one of the easiest components to prototype when it comes to this project. Most of the components come prepackaged with all the necessary resistors and capacitors already surface mounted to the board. There are other sensors that are available like this but because the chips are available by themselves, the group will design and prototype with them without the prepackaged units. There are usually to phases to prototyping, the first stage where most components are put onto a breadboard and then tested. The next stage is actually putting them on the main project board and testing it.

5.2.1 Sonar Rangefinders

These sensors are prepackaged and ready to go. The only work that the group will need to do to prototype these sensors in either stage will be to properly connect them and make sure that the proper voltage is applied.

5.2.2 Accelerometer

This is a sensor that can be bought in a package with a gyroscope, but the group will prototype it without the prepackaged unit for the second phase. For the first phase the group will use the prepackaged unit to design and test around. It will be connected to a breadboard and then connected to the MCU. All of the needed capacitors and resistors are already mounted to the breakout board. Figure 50 below is an image of the individual sensor that shows the size and pins of the breakout board. The chip will be mounted directly onto the main board at the prototyping stage. Surface mount resistors and capacitors will accompany this chip.

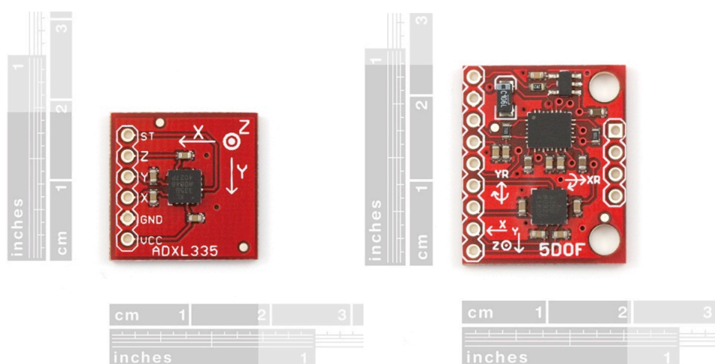

Figure 50: ADXL335 breakout board and IMU breakout board
Both figures used with permission from SparkFun.com

5.2.3 Gyroscope

As stated above, this sensor can be bought in a prepackaged unit. The group will buy the prepackaged module for testing and design purposes. All of the needed capacitors and resistors are already mounted to the breakout board. Figure 53above is an image of the prepackaged unit that shows the size and pins of the breakout board. The chip will be mounted directly onto the main board at the

Stephen R. Smith 12/13/10 9:13 AM
Comment:

prototyping stage. Surface mount resistors and capacitors will accompany this chip.

5.2.4 GPS

The GPS module is a prepackaged unit. It comes with pin outs already on the board. The unit we are using will be used for both types of prototyping. Since the module only comes one way and that is with a breakout board attached then it will be mounted to a breadboard for the first stage and then mounted to the main board via header pins for the second stage.

5.2.5 Compass/Altimeter

The compass/altimeter sensor also comes in a prepackaged unit. It comes with pinouts already on the board. Due to the delicate nature of the digital compass, the group will be using this package instead of mounting the chip themselves. So, this particular unit will be used for both types of prototyping. It will be mounted to a breadboard for the first stage and then mounted to the main board via header pins for the second stage.

6 Testing

The testing of Quad-Copter is broken in to subsections. This was done so that the testing of different subsections can be done in parallel. The final test will be the flight with all the subsections working.

During first not tethered flight test the Quad-Copter speed was increased too quickly and made the Quad-Copter unstable. This resulted in the Qua-Copter crashing and breaking one of the legs, bending all the motors, and breaking all propellers. The group had to act quickly to get the Quad-Copter back in one piece before presentation. It was concluded from that the Quad-Copter needs to be tethered down for future testing.

6.1 Linear Control System

The testing of the linear control system will be crucial for the stability of the quad-copter. This system will be the brains of the quad-copter. It will essentially read the inputs from the gyroscopes and the accelerometers and decide the needed speed of the motor in order to get a stable flight. In order to test the linear control system the quad-copter needs to be completely built with all the subsystem on it. If all the subsystems are not on the quad-copter when the testing begins, then the distribution of the quad-copter needs to be accommodated for. For example if the video subsystem is not available then a device that weights the same needs to be use in place of the video system.

The testing will be broken up in three parts. The first two parts of the test will have the quad-copter locked down to a platform. The last part of the test will be conducted while the quad-copter is in flight. The first part will be testing if the linear control system can stabilize either the pitch or roll of the quad-copter one at a time. This will show that linear control system is capable of stabilizing and doesn't need to be redesigned or tweaked. Also the first test will be used to test obtain the gains of the PID terms. Once the linear control system is able to stabilize the pitch and the roll individually, then it will be tested to see if it can stabilize the quad-copter.

The second part of the test will be to test if the linear control system can stabilize both pitch and roll at the same time. This test will use the gains of the PID terms that were found on part 1 of the linear control test. This test will be the most critical as will determine if linear control system can stabilize the quad-copter. If this test fails the control will have to be modified in order to accommodate the stabilization of both the pitch and roll. Once the second part of the test is a final test will be done to see if the results of the second test can be duplicated while the quad-copter is in mid air

6.1.1 Test I:

The platform that will be used for the first test will be made with PVC pipes. A long PVC pipe will act as a vertical stand. This vertical stand will have a base that will weight enough to withstand the weight of the quad-copter. The base will also have to withstand the forces created by the motors of the quad-copter. On top of the vertical stand will be a device that will allow the quad-copter to rotate on a plane. For example to test the pitch the device will allow the quad-copter to rotate on the YZ plane. To test the roll the device will allow the quad-copter to rotate on the XZ plane.

The first part of test will be conducted along the guide lines of the methods used for tuning a PID controller. The method that will be used for tuning the PID controller will be the Ziegler–Nichols method. The quad-copter will be put on the PVC platform to test one rotation at a time. Then the motors will be turned on and set to a predefined speed. The proportional gain of the PID controller will be set to 1 and the integral and derivative gain will be set to 0. Then the PID gain will be raised until the output of the PID controller starts to have stable oscillations. The gain at which this happens will be recorded it. Also the period of the oscillations will be recorded it. Using these values the gains of the PID controller will be calculated. The new PID controller gains will set and the system tested for stability.

The first rotation that will be tested will be pitch. To test for stability the axis that the quad-copter is rotating around, will be pushed down and up. The linear control system should be able to get the quad-copter back to a stable position. If not then the PID gains will have to be calculated again and first part of the test will start over again. If the linear control system is able to get the quad-copter to a stable position again then the next rotation will be tested, the roll rotation. The roll rotation will be tested with the gains of the PID controller that were calculated for pitch rotation. The same results that were observed during the testing of the roll should be observed with the pitch rotation. If everything is the same for roll and the pitch the second part of the test linear control test will be conducted. If not then a set of gains will be calculated for the roll rotation.

6.1.2 Test II:

The second part of the test will be conducted on the same PVC platform as the first part. The only difference is that the device that restricts the movement to rotation on one plane will be replaced with one that allows both pitch and roll rotation at the same time. This device should not allow the angle of any rotation to exceed 90 degrees. In fact the device should only allow rotation close to 80 degrees. The reason for this is to protect the quad-copter from getting damaged.

The second test will be conducted almost like the first test. The quad-copter will be pushed and pull from different angles and position with different magnitudes. The linear control system should be able to stabilize the pitch and the roll rotation. Also this should be done in within a relatively small time. If the linear control system is able to achieve this, then an in-flight test will be conducted. If not, then a lot more testing and tweaking needs to be done. Since is the most crucial test, this test will consume most of the testing time. What is expected is that the linear control system can cycle through the two rotations and fix them one at a time. If for some reason this is not optimal a linear control system that fixes both rotations needs to be developed. Once this part of test is successful a final test will be done with the quad-copter flying.

6.1.3 Test III:

The final test will be conducted with the quad-copter being in mid-flight. This test should reproduce the results obtained in second test. The quad-copter will be tethered down with 4 strings. This will be done in order to keep the quad-copter safe and people around the quad-copter safe also. The strings will be also used to pull on the quad-copter to see the linear control system can stabilize the quad-copter. If it can then the linear control system can be used on the final build the quad-copter. If it can't then the whole linear control system will have to be redesigned and tested. Another test that will have to be tested is if the linear control system can stabilize quad-copter independent of what speed the motors are at. If all these tests are accomplished and are successful the linear control system will adopted for the quad-copter.

6.2 Wireless Communication

The wireless communication system needs to be tested thoroughly. If not tested thoroughly the control of the quad-copter could be loss during flight time. This is not a desirable situation as the quad-copter could crash or be out of reach. For this reason the wireless communication will be tested for, error rate, range, data rate, and output power.

6.2.1 Test I: Error Rate

Testing the error rate of the wireless communication system will help insure that the quad-copter will not miss a command or data request. In order to test the error a group of 4 to 5 messages will be transmitted from the quad-copter 100 times each. The remote controller will then count the number of error and calculate the error. If the error rate is higher than 2 to 5 percent the wireless communication system to be adjusted or specifications lowered.

The test that was done was using an Xbee connected to a computer and one connected to the Quad-Copter. The computer is going be used to send the increase motor command, ‘m’, to the Quad-Copter. The results were that the command did not reach the Quad-Copter with a reliability of 100%. It seems that the reliability was more between 80% and 90%. This was not a problem with increasing the motors, but it became a problem when the turn off all motors command was issued.

6.2.2 Test II: Range Testing

The range of the quad-copter will be tested in an open field, and in a semi closed area like the front of the Harris Engineering Center building. For range testing the remote controller will be sending a command to the quad-copter. This needs to be special program apart from the final that will be used on the final build. This programs needs to power motor that correspond to the command that will be sent, and keep all others motors and sensors off. For example the remote controller can send the command move left. The quad-copter will only turn on the motor corresponding to the motor used to move to the left. The command will sent every minute as the range between the quad-copter and the remote controller will be increased in 5 meters intervals. Once the maximum range is found then an autonomous routine can be implemented that keeps the quad-copter within range of the remote controller.

The plan to test the range was scraped, because the group was focus on getting the Quad-Copter in the air. For all the testing that was done, range was never a problem.

6.2.3 Test III: Data Rate

The testing of the data rate will insure that the quad-copter meets the required data rate. To test the data rate the SimpliciTI protocol will be configure to send different amount of payload data in a frame. Starting with the lowest 10 bytes and ending on the highest 50 bytes. The data rate will be tested in increments of 5 bytes of payload data. Once a maximum data rate is established the wireless communication system can be configured below this data rate to insure a reliable wireless connection.

The SimpliciTI protocol was not used because the group decided to use an Xbee instead. Once again this test was also scrapped due to the focus of the project shifted to primarily getting the Quad-Copter to fly.

6.2.4 Test IV: Output Power

The testing of the output will be done in order to find the best output power for the desired error rate, range, and data rate. The test will conduct by configuring the SimpliciTI protocol of different output power and testing the error rate, range, and data rate. Once an optimal output power that gives the best results in order of priority range, error rate, data rate.

On the Xbee there is no control of the amount of power if outputs. This is set by the manufacturer and cannot be changed. Because of this test was scrapped.

6.3 Sensors

When testing the sensors, there are two parts to testing them. The first part is testing them without the MCU. The next part is testing them with the MCU. This will allow the group to make sure that if there is a problem, it can be found quickly and fixed.

6.3.1 Sonar Rangefinders

The sonar rangefinders are an analog device that will send a voltage to the MCU via an ADC pin. To test these parts, the group will need to make sure that the control cases work. The control cases are the extreme distance cases. These must be valid before making sure that the range is correct. To test the least extreme the group will place a piece of tape on or cardboard in front of the sensor. The analog signal will then be measured to make sure that the voltage is 0V or very close to 0V. The next extreme case to be measure is the maximum distance. To measure this the sensor will be placed outside with nothing in the sensors FOV cone for approximately 30 ft. The analog signal will then be measured to make sure that the voltage is 5V or extremely close to 5V. The next test is to make sure that the sensor can detect something in its view and return back to the maximum if nothing is in the sensors FOV cone. To do this the sensor will be placed in an area where nothing is in its FOV. The analog voltage should then read 5V. If it does then something will be placed in front of its FOV and then the voltage measured. If the voltage decreases, move the object from the FOV and make sure that the voltage increases back to 5V. The last test for the sonar rangefinders is to make sure that the voltage is half when the distance is half of the maximum. To do this an object will be placed at exactly 127in (10ft 7in) from the sensor. If the voltage is 2.5V then the sensor should work exactly as specified.

To test the sonar with the MCU, instead of measuring the voltage, the digital signal from the ADC pin will be converted into a PWM and sent over another pin, which can be measured by an oscilloscope. If the duty cycle of the signal is 0% then it represents 0V, if the duty cycle is 50% this represents 2.5V and if 100% then it represents 5V.

The sonar range finders were tested without the MCU, but were tested using a volt meter. The test with the PWM and Oscilloscope was not performed. The volt meter produced enough information to verify that the sonar range finder work as expected. The MCU test was not done due to time constraints. In the final build the sonar range finder were not used.

6.3.2 GPS

The GPS unit will be tested using the built-in USB ports and the build-in USART ports. To test the unit by itself, the group will need to apply power to the GPS and wait for the fix LED to turn on. This LED displays whether or not the GPS has an established link with the satellites. Next the group will attach the unit via USB and check the coordinates with a computer. Using Google Maps the group can check to see if the x and y coordinates are correct. This will be done at several locations to check for reliability and accuracy.

To test the GPS with the MCU, the group will have to make sure that the communication link works. If the link works and passes its tests then the group can test the GPS. The group will need to turn the power on and wait for the fix LED to turn on. To test the GPS the readings will need be taken from the computer GUI and verified from Google Maps. This will be done at several locations to check for reliability and accuracy. If the locations match up with the coordinates then the GPS is working correctly.

The GPS was tested and it was able to lock on the satellite and give a position. But the position was not verified with Google Maps. Unfortunately the GPS was not part of the final build.

6.3.3 Accelerometer

The accelerometer will be tested similar to the sonar rangefinder. The sensor will be connected to a breadboard and then attached to power. The group will use a voltage meter to check the voltage of the respective axis. There are six tests that need to be tested. The first test is to check the negative Z. To do this the group will place the chip flat with the bottom down. The next test is to check the positive Z-axis. The chip will need to be placed flat on its top and then the voltages measured. The next test is to check the negative X-axis. By following the pictures on the chip, it should be placed opposite of the figure with the X position and then the voltages measured. Next the chip should be rotated 180 degrees to measure the positive X-axis. The next test is to check the negative Y-axis. By following the pictures on the chip, it should be placed opposite of the figure with the Y position and then the voltages measured. Last the chip should be rotated 180 degrees to measure the positive Y-axis. Table 18 is a chart to test each axis and what the voltage meter should display on each pin.

Measured Axis	**X Pin**	**Y Pin**	**Z Pin**
Negative Z axis	1.65V	1.65V	0V
Positive Z axis	1.65V	1.65V	3.3V
Negative X axis	0V	1.65V	1.65V
Positive X axis	3.3V	1.65V	1.65V
Negative Y axis	1.65V	0V	1.65V
Positive Y axis	1.65V	3.3V	1.65V

Table 18: Chart to test axis on accelerometer

To test the data from with MCU it will be tested just like with the sonar sensors. It will be measured using a PWM signal from one of the pins on the MCU.

The accelerometer was tested and verified that the calculated values to within 3%. Just like the sonar range finder the volt meter provided enough information to verify the numbers. Once the breakout board of the accelerometer was tested, the main PCB was also tested. PCB test was done using the wireless communication system, and it verified that the accelerometer and ADCs were working on the PCB.

6.3.4 Gyroscope

The gyroscope is a hard sensor to test due to its function. It measures rotation about a specific axis. The only way to test this is when it is mounted on the quad copter and to check the data as it comes from the gyroscope. First the communication link needs to be tested because this test requires that the communication link properly works. Then a temporary value could be added to the computers GUI to check the values of the gyroscope and determine if the values are accurate or not. This would require little modification to the MCU's code to add the temporary values. Another really simple test is to check if the quad copter is stable. If it is then the group knows that the stabilization algorithms are working with the gyroscope and accelerometer properly.

The Gyroscope was tested the same way that the accelerometer was tested. The only value that the group was certain of was the zero acceleration value, which happens to be 1.35 volts.

6.3.5 Digital Compass/Altimeter

The digital compass/altimeter unit is another difficult unit to test because it communicates via I2C. The only way to test this is to connect it to the MCU. To test the function of the digital compass, the group needs to make sure the communication link works. If it does then the group can add a temporary code to the MUC to relay the value to the computer. The group will also need to add a value to the computers GUI to check the data. The test is to check how many degrees the digital compass of off. If the compass if facing North, then the offset should be equivalent to 0 degrees. If it is at East then the offset should be equivalent to 90 degrees. If it is facing South then the offset should be equivalent to 180 degrees. Last if it is facing West then the offset should be equivalent to 270 degrees. The only way to test the altimeter is to test it while on the quad copter. The easiest way would be to move the quad copter to a specific height that is known and verify it with the altimeter.

The Compass unit was tested using an external MCU. The reason for this is that the I2C bus on the main MCU did not work. The same code that works on the external MCU should work for the main MCU but it didn’t work. After much testing it was decided that the update rate of the compass was to slow for the Quad-Copter to react and compensate. For this reason the Compass not part of the stabilizing algorithm instead it was use mainly to know the heading.

The altimeter was not used because the accuracy of it was not good enough. In other words the Quad-Copter needed to be really high in the air for the altimeter to be any useful.

6.4 MCU

The MCU is the central part of the copter. If it doesn’t work properly then the entire copter will not work. With this said, testing of the MCU is crucial for the success of the project. There are multiple parts that need to be tested on the copter. The main parts are the ports the sensors communicate through.

To test the ADC on the MCU a voltage signal of 5V will be placed on every ADC on port A. A PWM signal will be produced on another pin depending on the voltage applied on the ADC pin (the ADC pin and the output pin will be set in the code). The PWM should have a duty cycle of 100%. The 5V will then be reduced to 2.5V. The PWM should have a duty cycle of 50%. Finally the 2.5V will be reduced to 0V. The PWM should have a duty cycle of 0%. This will require a large amount of reprogramming for every four pins. Next the 3.3V ADC on port B must be tested. A PWM signal will be produced on another pin depending on the voltage applied on the respective ADC pin (the ADC pin and the output pin will be set in the code). The PWM should have a duty cycle of 100%. The 3.3V will then be reduced to 1.65V. The PWM should have a duty cycle of 50%. Finally the 1.65V will be reduced to 0V. The PWM should have a duty cycle of 0%. These tests should fully test the ADC on the MCU. An addition test will be to adjust the voltage on the ADC and see it the PWM's duty cycle increases or decreases depending on the applied voltage.

To test the two USART ports on the MCU, a large part of the test will require slight modification of the code. First the group will connect the corresponding Tx and Rx pins together. This will allow the MCU to echo the character is send back to itself. In the code the character 'a' will be sent indefinite through the transmit ISR. Inside the receive code, a conditional statement will be added to test if the value 'a' was transmitted. If it is then turn an output pin high. This will allow the group to test the USART. Also the 'a' character can be changed to make sure that all the bits are being transmit and received properly.

To test the PWM function of the MCU, the group can simply attach an oscilloscope to the output pins and confirm that the frequency is 50Hz. Then the length of the PWM's duty cycle needs to be confirmed. The values at first will be hard coded into the output compare register to test their function. The value range will need to be between 2000 and 4000 inclusive. The values what will be tested are 2000, 3000, and 4000. If 2000 is currently in the output compare register then the duty cycle should be 5%. If 3000 is currently in the output compare register then the duty cycle should be 7.5% and if the value is 4000 then the duty cycle should be 10%. If the duty cycles are correct then the PWMs are functioning correctly.

To test the I2C the MCU will use the same test the digital compass/altimeter used. If that test passed then it will be assumed that the I2C bus is functioning correctly.

To test the programming of the MCU, the programmer will either be attached to the breakout board or the final main board. If AVR Studio says that the programming of the MCU was successful then the group will assume that it was properly programmed. This test will also allow the group to make sure that the target voltage is correct. Inside of the programming panel inside of AVR Studio, the software will tell the user the current level of the target voltage, which is the MCU.

Most of the peripherals of the MCU were and tested. The ADCs worked and expected. The UART also worked as expected except for that it might miss 1 or 2 characters once in a while. The PWM worked well and were able to control the motors with enough accuracy. The I2C bus on the main MCU did not work at all. It was tested several time with an oscilloscope but it showed noise, instead of a clock signal. The group decided to use an external MCU for the I2C bus and send results through the UART port.

7 Project Operation (Owner's Manual)

This is the Project Operation section for the Quad-Copter, which will overview software installation, standard operating procedure, and troubleshooting. The standard operating procedure will detail software setup, hardware setup, initial test flight requirements, and flight procedure and recommendations.

7.1 Getting Started – Software Installation

The software to be used is the Simple Terminal, provided by _. This will allow for control of the Quad-Copter via XBee module through a computer interface. If the user already has another terminal interface, such as the Quad-Copter GUI, or any other terminal-based XBee interface, this will suffice. Here are the instructrions for handling downloading and installing the Simple Terminal Interface.

Setting up the software interface:

1. Go to the webpage for Smiley's Micros which contains the required files: http://www.smileymicros.com/index.php?module=pagemaster&PAGE_user_op=view_page&PAGE_id=50.
2. Scroll down towards the bottom and find the Simple Terminal download.
3. Download the file to the computer or laptop (recommended to use a laptop, since the Quad-Copter will be used primarily outdoors).
4. Once downloaded, go to the zip file, and unzip to the Desktop, or to another folder in a convenient location.
5. Go to the setup.exe file and double click to start Simple Terminal setup.
6. Follow the prompts and use recommended settings
7. When it prompts "Finish", click finish.
8. Go to the Start Menu and click on "Simple Terminal" (this will be very easy to find on a Windows Vista/7 computer).

If software setup is successful, continue onto the Hardware Setup section. If any problems occur, go to the troubleshooting section.

7.3 Hardware Setup

This section is design to help install batteries and external components to the Quad-Copter safely. Most of the components come prepackaged and already mounted to the Quad-Copter. However, if the user decides to upgrade or modify the parts, this is a tutorial for mounting new or replacement hardware.

7.3.1 Battery Safety

The safety considerations are based on both the user's safety and the Quad-Copter. The batteries to be used are the EM-35 made by Esprit. At the time this manual is being written, these batteries have gone obsolete, and the owner is recommended to use the EM-35 Pro, which holds the same dimension, with 400mAH of charge. Differences in charge capacitance will be discussed in "Increasing Flighttime".

The batteries to be used for the motors are to be given care and consideration, since the batteries used are typical for advanced flyers. For the beginner, using a LiPo battery source for the first time comes with a higher degree of risk. To minimize risk to the beginner, these are a few tips to keep in mind before connecting the main LiPo battery to the motors:

- Make sure the batteries are sealed
- Feel the battery for bubbling or squishy areas
- Double-check the charge levels with a battery checker.
- Always operate the Quad-Copter with the battery tester installed into the discharge port.
- Do not connect the LiPo main power to the main board. This will fry the on-board circuitry. Use the 9V alkaline or it's rechargeable counterpart.
- If any of the contents of the battery are exposed, off immediately.
- If these contents are somehow ingested, or create some other form of reaction, such as nausea or vomiting, contact 911.

To install the batteries, follow the steps below:

1. Have the power switch on the main board placed in the off position and the 9V battery is detatched from the system.
2. Make sure all power connections to the ESC and the motors are securely plugged in and properly covered. If these connections are exposed, use electrical tape to cover the connections.
3. Connect the branching wires to the ESCs power port when they're unconnected to the battery.

4. Place the EM-35 battery towards the bottom of the rods beneath the main plate. Secure the battery to the rods (electrical tape is enough for supporting limited flight time.
5. Connect the EM-35 battery to the branched plugs, black lead to black lead first.
6. Connect the battery checker with to the discharge port of the battery.

7.3.2 Installing external components

The program currently installed to the main board is setup for the inertial momentum unit (IMU) sensors, the rangefinder sensors, a GPS unit, and a digital compass unit. The IMU is already on-board, and is used for flight stabilization, while the rangefinder sensors, which are used for object detection and ground sensing, are external devices. These devices use an analog signal, meaning they give values in terms of a direct voltage. The GPS and the digital compass modules use a digital signal, which is similar to a computer, transmitting 1's and 0's to the main board for operation. These require a digital signal for interpretation for the higher accuracy these devices are used to determine position and heading. All sensors, both analog and digital, require no extra programming for operation, as the main board is preinstalled with the necessary files and programs needed to operate.

The easiest of these devices to install is the GPS unit, which uses an extendable attachment to allow for the GPS to be placed with a greater degree of freedom. Here are the steps for setting up the GPS to the Quad-Copter.
To install the GPS unit:

1. Plug one end of the cord to the GPS, and plug the other into the EM406 socket (the white plug-in jack on the corner of the board).
2. Place the GPS on the adhesive surface located near the plug, with the flat side of the component facing down.

The GPS unit is ready for taking coordinates. Please note: the GPS unit has an built-in button battery which is used to help draw extra power for retrieving coordinates and sending them to the main board. It may take a few minutes after powering the Quad-Copter on to be able to get any viable data.

The digital compass being used can be powered by the main board, and is used to help stabilize the Quad-Copter. This device also is used for giving a heading based on the magnetic north pole. Please note: this north is not directly north, and it will conflict with your interpretation of north. Please note: this north is not directly north, and it will conflict with a map's concept of north. This will not interfere with the GPS coordinate systems in the main board. The compass does give orientation data back to the main board, which helps the Quad-Copter from spinning.

Installing the digital compass:

1. Get 4 female-female headed wire and connect to the following pins:
    a. SCL
    b. SDA
    c. VCC
    d. GND
2. Connect the other end of the above wires to the same pins on the digital compass
3. Orient “North” on the compass to north.
4. Place the module on the adhesive. Note: The orientation of how the compass module on the adhesive will determine north from the main board’s perspective. It is to be oriented as the user sees fit.

The next components for installation are the rangefinder sensors, which are two ultrasonic sensors. These are used as the object detection system of the Quad-Copter, which are used for basic object avoidance and ground detection. North is assumed to be the designated front of the Quad-Copter.

Installing the rangefinder sensors:

1. Connect female-female header pins to the “Sonar_F” pins. Please note the order of the pins from left to right (Use the battery connectors placed to the right of the board as the reference):
    a. GND
    b. VCC
    c. AN
2. Place the pins in the same order on the rangefinder sensors.
3. Mount the rangefinder 8 inches below the motors on the landing gear facing outward. Use adhesive and electric tape on the back end to be able to mount the device.
4. Repeat steps 1-3 for the “Sonar_B”, except secure this rangefinder below the LiPo battery using electric tape only.

7.4 Initial Test Flight Requirements

The size of the Quad-Copter is roughly 18 inches in radius, and 15 inches in height. It is highly recommended any test flights are done in a very open area, such as a large atrium, gymnasium, or an open field. This craft does move very quickly once off the ground and can easily hurt someone. If there is no intention of an indoor test flight, skip the next section.

However, if an indoor test is to be done, it is recommended to all users to use a stopper and a low friction string, like fishing wire or kite string. To use the tethered system, follow these guidelines:

- Use a garage or a pipe for mounting a ceiling.
- Place a hook over the desired landing zone of the Quad-Copter. Note: for a pipe, a hook is not necessary, but securing the string around it tightly is required.
- Directly below the ceiling hook, place a secondary hook and fasten to a plank or board of wood. Make sure the wood
- Secure the ends of the wood with small, heavy materials or objects, such as bricks or cinderblocks.
- Measure the height from the ceiling to the floor, and cut about 2/3 the height of string.
- Mount this string to the floor hook and run the other end through one of the exposed holes on the center plate.
- Cut about half the length of the string and secure to the top. Use more string if the ceiling is a pipe.
- Place a stopper, such as a metallic belt latch, on then suspended end.
- Tie the string attached to the floor to the latch. Make sure the latch is taut when securing the string. Cut away the excess string.
- Ready for the Test Flight!

7.5 Test Flight mechanics

As it was stated earlier, the software to be used for all test flights will be the Simple Terminal. All of the commands to be implemented will be used through this terminal. The following commands will be executable for test flight. See Table 19:

Command	Description
m	Increase motor speed
n	Decrease motor speed
p	Increase the gain value of the controller by .01 (preset)
o	Decrease the gain value of the controller
i	Increase the error compensation in the controller by .01 (preset)
u	Decrease the error compensation in the controller by .01
d	Increase the delay compensation in the controller (preset)
s	Decrease the delay compensation in the controller by .01
t	Takeoff – automatically increases motor speed to hover
x	Software Kill Switch

Table 19: Flight commands and descriptions.

The commands are case sensitive, and must be executed as lower-case letters. The upper case equivalents will not be registered. Therefore, make sure the caps lock is turned off before starting flight. Before turning on the main board, be sure to do a last minute flight check:

- Make sure all connections to the main are plugged in properly and securely.
- Double check the wires to the ESCs, and make sure the “NORTH” ESC is the same as the north reference on the compass.
- Make sure the 9V and the LiPo batteries are properly connected to the system.
- Make sure there is plenty of room for movement. This will not only protect anyone nearby, but other property and the Quad-Copter from getting damaged.
- Double check all devices to make sure they’re securely placed.
- Make sure the video system is tested and operational.
- Ready for take-off!

To begin flight of the Quad-Copter:

1. Turn on the main board.
2. The motors should beep. Move back immediately from the Quad-Copter.
3. Go to the Simple Terminal on the laptop.
4. On the settings menu, use channel 38400, and click OK.
5. The port should read, “Close Port”. Left click on the mouse pad or an external mouse until data is read. If data doesn’t pop up after 2 minutes, turn off the main board and repeat steps 1 and 2.
6. Once a string of data appears on the terminal, the computer is now receiving data, which can be used for tweaking the Quad-Copter’s performance. It also means the Quad-Copter is ready to receive commands. There are two options:
    a. Press “t” and see the Quad-Copter take-off.
    b. Press “m” slowly until the Quad-Copter lifts off the ground.
7. Optional – If new equipment has been added to the Quad-Copter, and the balancing seems a little off, modify the controller values slightly to compensate.

The Quad-Copter should now be off the ground and flight has begun. If for whatever reason the flight of the Quad-Copter becomes erratic, press and hold “x” to slow the motors down to initial settings.

7.6 Increasing Flight time

The flight time of the Quad-Copter is based almost solely on the charge found in the LiPo batteries. The ones currently being used are the EM-35 by Esprit. These batteries hold 2250mAH of charge. The best way to determine what would be the longest charge per single motor is given below. Table 20 gives the typical values for LiPo batteries and gives an approximate time for operation in minutes.

Battery Capacticty	Time (one motor)	Time (two motors)	Time (four motors)
1300mAH	5.74	2.87	1.43
2250mAH	9.93	4.96	2.48
2600mAH	11.47	5.74	2.87
3300mAH	14.56	7.28	3.64
3750mAH	16.54	8.27	4.14
4400mAH	19.41	9.71	4.85
5000mAH	22.06	11.03	5.51

Table 20: Flight time varying with the number of attached motors to a single battery

Remember: These are typically heavier batteries, and adding and removing the current batteries will alter the mass of the Quad-Copter. The controller values may need to be altered if this results in unstable flight. Using the tethering technique above will help, however, if it is required, test on a single axis by removing the landing gear to opposite sides and strapping the non-landing gear legs to two free standing structures approximately 3 feet apart. Then test for oscillations with slow motor speeds at first, and change gain, error compensation, and delay values as needed. Use the landing gear on the free legs to help create disturbances to test these values further.

7.7 Understanding the Terminal

The data being displayed on the terminal is displayed as the following:

- First four columns are the motor speeds in the following order: NORTH, SOUTH, EAST, and WEST.
- The next 2 columns are the values of the gyroscope and the accelerometer as it passes through a software filter. This helps negate the effect of noise generated from the frame.
- The next three values are the PID values, otherwise known as the gain, the error compensation, and the delay values. Notice how the delay value is negative. This value works best if it is at or below zero. Be very careful in modifying this value to high.
- The last set of values should be the actual values of the accelerometer and the gyroscope. Notice how these values bounce around a great deal. This is because the IMU components have micro-sized structure, which

produce these values due to vibrations in the Quad-Copter. This makes it very difficult to keep track of the actual position.

8 Project Management

8.1 Group Organization

The organization of the group was such that each person was responsible for themselves as well as the other group members. If someone in the group did not fulfill their part, others in the group would either help or encourage that person to reach their goal. This informal approach was often facilitated by 4-way conference calls on Skype. This worked fairly well throughout the lifecycle of the project.

The quad-copter project was a project that consisted of integrating various subsystems together. Since these subsystems were complex on their own they were divided up between four people. Each subsystem needed to be researched, designed, prototyped, tested and incorporated into the final build of the quad-copter. The responsibility of the subsystem was divided based on what each person is good at or interested in learning about. The subsystems and corresponding group members are shown in table 21 below.

Subsystem	Group Member
Main Software	Josh
Linear Control System	Engers
Frame	All
Motors	David
Power Supply	David
Microcontroller	Josh
Sensors	Steve
Wireless Communication	Engers
Video System	Engers, Steve
PBC Board	Dave
Autonomous Algorithms	Josh, Steve
Documentation formatting	Steve

Table 21: Delegated responsibility for each subsystem

8.2 Estimated Bill of Materials

The bill of materials, shown below in table 22, contains all the parts needed to build the full quad-copter. This list represents all of the parts that have been acquired. Shipping is an estimation, and is expected to be higher than listed.

Name	**Vendor**	**Cost**
ATMega2560	Atmel.com (Mouser.com)	Sampled ($17.53)
Various Capacitors	Various	$10.00
Various Resistors	Various	$20.00
Various Inductor	Various	$2.00
32 MHz Crystal	Mouser.com	$0.32
SMA Connector	Mouser.com	$4.04
Balun Filter	Mouser.com	$0.32
MSP430F2616	TI.com (Mouser.com)	Sampled ($9.76)
CC2520	TI.com (Mouser.com)	Sampled ($6.08)
XBee Modules(x2)	Sparkfun.com	$45.90
XBee USB Interface	Sparkfun.com	$24.95
2.4 GHz Antenna	Mouser.com	$14.64
EPP1045 propellers(x12)	Quadroufo.com	$72.00
BM2410-09 Motor(x8)	Quadraufo.com	$76.00
30A ESC(x6)	Bob's Hanger	$130.99
GPS100	Quadroufo.com	$38.50
Honeywell HMC6352	Sparkfun.com	$34.95
Xmega100 Breakout	Sparkfun.com	$24.95
FTDI USB/UART Interface	Sparkfun.com	$14.95
IMU 5DOF	Sparkfun.com	$74.95
Dual Axis Gyro – IDG500(x3)	Sparkfun.com	$24.95
Tripe Axis Accelerometer	Analog D./(Sparkfun.com)	Sampled ($9.95)
Maxbotic LV-EZ2(x2)	Sparkfun.com	$55.90
Compass Module – HDPM01	FutureLec.com	$19.90
Wires	Various	$20.00
2250mAh LiPo battery(x2)	Bob's Hanger	$80.92
LiPo charger	Supplied	$0
AA Batteries	Radio Shack	$10.00
AA Battery Case	Radioshack	$3.21
Switch	Radio Shack	$0.99
Header Pins	SkyCraft	$5.00
PCB	Advanced Circuits	$136.00
Mounting Hardware	Lowes	$10.00
Shipping and Handling	UPS/FedEx	$400.00*
Total		1411.18

Table 22: Bill of Materials

. (*) estimates and not exact values on their price.

8.3 Milestone Chart

The milestone chart in table 23 below shows how the group planned to allot time for each phase of quad-copter development. The general plan was to develop systems that have dependencies, such as sensors, earlier than stand-alone systems, such as the video system. In retrospect, production was significantly bottlenecked in developing the flight stability system.

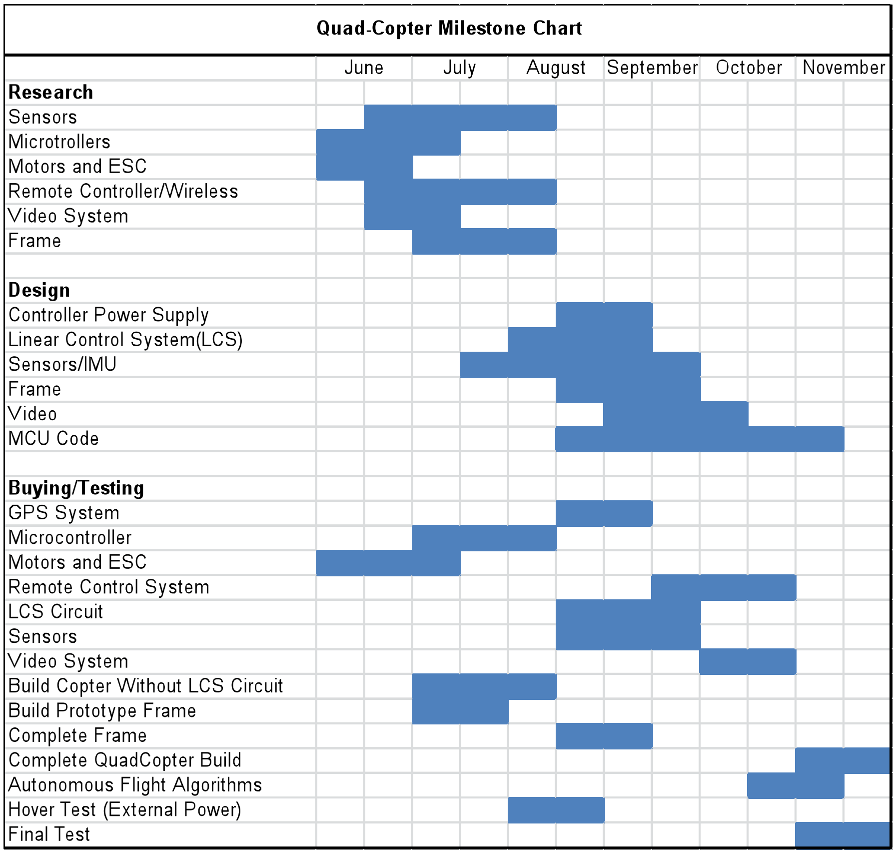

Table 23: Milestone Chart

8.4 Project Summary and Conclusions

All of the hardware to be design for the main board worked with great success. At one point, there was a variation in the IMU setup for the ADXL335 to test for increased time-response. The experiment determined the frequency response of 50Hz was the wiser consideration for filtering purposes. Problems occurred with the final board regarding the main board ground and the ESC ground. This was due to an overloading of voltage found in plugging the ESC modules power first. Simple patchwork with 24 gauge wire solved this problem to resupply and patch then board. Variations in working with the prototype board to the final board lead to many inconsistencies in the software operation. The main problems being only half of the ADC ports were operational for both boards, and the two wire bus, or I2C bus, was not operational on the second board. The use of an alternative MCU to handle these processes was for operating with the digital compass module and the rangefinder sensors.

Possible reasons for these problems were the minimal testing given to the ADC ports. This was due to assuming one set of data was operation, therefore the other port was also configured for accepting data. Once this was corrected in the code, the chip was not modifying itself for the new code. The I2C bus wasn't operational since the SCL line was emitting a steady-state response. It is possible the chip didn't configure this line properly, since the ATMega328 was able to respond to the same code with greater success. These issues require more investigation to further understand the reasoning why these modules weren't operational. Sensor data acquired from the GPS unit is acquired quickly for outdoor use. The built-in battery needs to be charged up to the point of operation before any readings are delivered to the current coordinates. The digital compass module operates with success at a very slow rate. The update rate is too slow to be implemented as a yaw control. Given more time, this sensor could be used to help with stabilization. The ultrasonic sensors used for this project are viable short range detectors. Operational problems occurred due to the required operation of the RX pin to turn on the device. These will be design modifications made for future use.

All initial test flights were held in the Senior Design lab once the preliminary frame was constructed. The testing of the single axis gave initial insight for balancing the Quad-Copter. Because of the orientation of the batteries below the main plate, the values of each PID is slightly different for each axis. The yaw control is handled as a P controller, since feedback data is sent at a very slow rate. The tethered system developed for this project allowed for safely testing for yaw control, however as the project progressed, it became apparent the string was also complicating the outcomes. This is due to the nature of a helicopter in takeoff. This was confirmed in the initial test flight, where drift was the reason for most of the unbalance in the Quad-Copter's stability. The flight also determined the lack of power being sent to all the motors. It is undetermined if this was a flaw in the code or in the amount of power in each battery. What is determined is the battery which was used for this initial test was the sudden drain in the battery which occurred at takeoff. It is recommended to use separate batteries, one for each engine, to optimize takeoff and flight of the Quad-Copter.

9 Appendices

A. References

Aeronautics

Dole, Charles E. Lewis, James E. “Flight Theory and Aerodynamics. A Practical Guide For Operational Safety, Second Edition.” John Wiley and Sons, Inc. 2000.

Bramwell, A.R.S. “Helicopter Dynamics”. John Wiley and Sons, Inc.1976. Watkinson, John. “The Art of the Helicopter”. Elsevier Butterworth-Heinmann. 2004.

Power Supply

“Energizer EN91Product Datasheet”. Energizer Corporation. < http://data. energizer. com/PDFs/EN91.pdf >.

“Radio Shack Engineering Datasheet – NiMH AA.” Radio Shack Corporation. 2004. < http://support.radioshack.com/support_ tutorials/batteries/ bt-nicd-hicap-AA.htm. >

“Powergenix NiZn AA Datasheet.” Powergenix Corporation. September 2009. < http://www.powergenix.com/docs/powergenix-specs-aa.pdf>

“ThunderPower Lipo Series Specifications Table”. RCHobbies.com. 2009 < http://www.rchobbies.org/thunder-power-specs.htm>“

Racer’s Edge LiPo Battery Packs”. Racers-edge.com. 2010.
< http://www.racers-edge.com/Welcome_to_Racers_Edge_com_s/77.htm>
Simpson, Chester. “Linear and Switching Voltage Regulator Fundamentals”. National Semiconductor. <http://www.national.com/appinfo/power/files/f4. pdf>

Simpson, Chester. “Switching Regulators”. National Semiconductor. <http://www.national.com/appinfo/power/files/f5.pdf>

“LM78XX 3-Terminal 1A Positive Voltage Regulator” Fairchild Semiconductor. 2010. http://www.fairchildsemi.com/ds/LM% 2FLM7805. pdf

"LM317 3-Terminal Positive Adjustable Regulator". Fairchild Semiconductor. 2009. <http://www.fairchildsemi.com/ds/LM%2FLM317.pdf>.

"KA75XXX Voltage Detector". Fairchild Semiconductor. 2009. <http://www. fairchildsemi.com/ds/KA/KA75330Z.pdf>.

"LP3892 1.5A Fast-Response Ultra Low Dropout Linear Regulators" National Semiconductor. June, 2006 <http://www.national.com/ds/LP/LP3892.pdf>.

"LM2585 SIMPLE SWITCHER 3A Flyback Regulator" National Semiconductor. February, 2010. <http://www.national.com/ds/LM/LM2585.pdf>.

"LM1575/2575/2575HV SIMPLE SWITCHER 1A Step-Down Voltage Regulator" National Semiconductor. April, 2007 http://www.national.com/ds/LM/LM 1575.pdf.

"LT1185 Low-Dropout Regulator. Linear Technologies. 1994. <http://cds.linear. com/docs/Datasheet/1185ff.pdf>.

"LT3830/3830-1 High Power Synchronous DC/DC Controller for Low Voltage Operation". Linear Technologies. 2001. http://cds.linear.com/docs/ Datasheet/3830fa.pdf.

"LT3573 Isolated Flyback Regulator without an Opto-Coupler." Linear Technologies. 2008. http://cds.linear.com/docs/Datasheet/3573fc.pdf

"LM7833-TF3-I-T - 3-TERMINAL 1A POSITIVE VOLTAGE REGULATOR - Unisonic Technologies". Unisonic Technologies. 2005. <http://pdf1. alldatasheet.com/datasheet-pdf/view/173515/UTC/LM7833-TF3-I-T.html>

"ADCMP341/ADCMP343 - Dual 0.275% Comparators and Reference with Programmable Hysteresis". Analog Devices.2007. <http://www.analog .com/static/imported-files/data_sheets/ADCMP341_343.pdf. Motors>

Chapman, Stephen. "Electric Machinery Fundamentals, Second Edition". McGraw-Hill, Inc. 1991, 1985.

Patrick, Dale R. and Fardo, Stephen W. "Electronic Motor Control Systems. Electronic and Digital Control Fundamentals and Applications". The Goodheart-Willcox Company, Inc. 2000.

"An Introduction to Sensorless Brushless DC Motor Drive Applications with the ST72141". STMicroelectronics, Inc. 2000. <http://www.st.com/stonline/ books/pdf/docs/7209.pdf.

“Development of the Electromotive Force”.. MicroMo, Inc.2007. <http://www. micromo.com/n390489/n.html>

“Brushless DCLDC Motor Fundamentals”. <http://ww1.microchip.com/downloads /en/AppNotes/00885a.pdf. Microchip Technology, Inc. 2003>

“Microchip PIC18F1230 Datasheet”. Microchip Technology, Inc. 2009. <http: //ww1.microchip.com/downloads/en/DeviceDoc/39758D.pdf>

“Atmel ATmega8a Datasheet”. Atmel Corporation. June 2009 <http://www. atmel. com/dyn/products/datasheets.asp?family_id=607#760.>

“2008 TowerPro Motor Specifications”. TowerPro. 2008. <http://www.towerpro. com.tw/driver/drivers/2008%20TowerPro%20Motor%20Spec.pdf>

Microcontroller

“Using the USART of AVR Microcontrollers.” Extreme Electronics. Web. 1 Aug. 2010. < http://extremeelectronics.co.in/avr-tutorials/using-the-usart-of-avr-microcontrollers/>

“Using the USART of AVR Microcontrollers: Reading and Writing Data.” Extreme Electronics. Web. 1 Aug. 2010. http://extremeelectronics.co.in/avr-tutorials/using-the-usart-of-avr-microcontrollers-reading-and-writing-data/

“Introduction to PWM – Pulse Width Modulation.” Extreme Electronics. Web. 1 Aug. 2010. http://extremeelectronics.co.in/avr-tutorials/introduction-to-pwm-pulse-width-modulation/

“PWM Signal Generation by Using AVR Timers.” Extreme Electronics. Web. 1 Aug. 2010. http://extremeelectronics.co.in/avr-tutorials/pwm-signal-generation-by-using-avr-timers/

“PWM Signal Generation by Using AVR Timers. Part II.” Extreme Electronics. Web. 1 Aug. 2010. < http://extremeelectronics.co.in/avr-tutorials/pwm-signal-generation-by-using-avr-timers-part-ii/>

“AVR Timers – An Introduction.” Extreme Electronics. Web. 1 Aug. 2010. <http://extremeelectronics.co.in/avr-tutorials/avr-timers-an-introduction/>

“Timers in Compare Mode – Part I.” Extreme Electronics. Web. 1 Aug. 2010.<http ://extremeelectronics.co.in/avr-tutorials/timers-in-compare-mode-part-i/>

“Timers in Compare Mode – Part II.” Extreme Electronics. Web. 1 Aug.2010.<http ://extremeelectronics.co.in/avr-tutorials/timers-in-compare-mode-part-ii/>

“Using the Analog To Digital Converter.” Extreme Electronics. Web. 1 Aug. 2010. <http://extremeelectronics.co.in/avr-tutorials/using-the-analog-to-digital-converter/>

“Interrupt driven USARTs.” AVR Freaks. Web. 1 Aug. 2010. <http://www.avrfreaks.net/index.php?name=PNphpBB2&file=viewtopic&t=48188>

“ATxmega64A1.” AVR Solutions. Web. 1 Aug. 2010. <http://www.atmel.com:80/dyn/products/product_card.asp?part_id=4297>

Sensors

"Ultrasonic Range Finder - Maxbotix LV-EZ1" Sparkfun.com. Web 30 July 2010 < http://www.sparkfun.com/commerce/product_info.php?products_id=639>

"Do Chipset Channels Matter?" GPSreview.net. Web 28 July 2010 <http://www.gpsreview.net/gps-chipset-channels/>

Hirvinen, Jani "GPS Altitude Error" (28 June 2010) DIYdrones/forum. Web 25 July 2010 < http://diydrones.com/forum/topics/gps-altitude-error>

Mehaffey, Joe "GPS Altitude Readout > How Accurate?" gpsinformation.net (rev 2/10/01). Web 26 July 2010 < http://gpsinformation.net/main/altitude.htm>

"Laser Range Finder/ Applications/Military". Wikipedia (rev. 2/7/2010). Web 20 July 2010 <http://en.wikipedia.org/wiki/Laser_rangefinder>

"Altimeter/ Pressure Altimeter". Wikipedia (rev. 6/2010). Web 10 July 2010 <http://en.wikipedia.org/wiki/Altimeter>

" Compass Module - HMC6352". Sparkfun. Web 12 July 2010 <http://www .sparkfun.com/commerce/product_info.php?products_id=7915>

Payne, Lew. "MediaTek MT3329 GPS 10Hz" (24 June 2010) DIYdrones/forum. Web 28 July 2010 <http://diydrones.com/forum/topics/mediatek-mt3329-gps-10hz?id=705844%3ATopic%3A163713&page=3#comments>

Starlino, "A Guide To using IMU (Accelerometer and Gyroscope Devices) in Embedded Applications" (rev. 12/29/2009). Web. 1 August 2010 <http://www.starlino.com/imu_guide.html>

Wireless Communications

“6LoWPAN.” 6LoWPAN - Wikipedia, the free encyclopedia. Web. 12 Jul. 2010. <http://en.wikipedia.org/wiki/6LoWPAN>

“ANTENNAS FOR LOW POWER APPLICATIONS.” RFM.com. Web. 21 Jul. 2010. <http://www.rfm.com/corp/appdata/antenna.pdf>

“Bluetooth.” Bluetooth - Wikipedia, the free encyclopedia. Web. 7 Jul. 2010. <http://en.wikipedia.org/wiki/Bluetooth>

“Bluetooth - An Overview.” Bluetooth; How networks are formed and controlled. Web. 7 Jul. 2010. <http://www.swedetrack.com/images/bluet10.htm>

“CC2520 Data Sheet.” TI.com. Web. 5 Jul. 2010. <http://focus.ti.com/lit/ds/symlink/cc2520.pdf>

“CC430f6137 Data Sheet.” TI.com. Web. 7 Jul. 2010. <http://focus.ti.com/lit/ds/symlink/cc430f6137.pdf>

“CSR BlueCore4-ext Data Sheet.”

“Error detection and correction.” Error detection and correction - Wikipedia, the free encyclopedia. Web. 19 Jul. 2010. <http://en.wikipedia.org/wiki/Error_correcting_code>

“IEEE 802.11.” IEEE 802.11 – Wikipedia, the free encyclopedia. Web. 5 Jul. 2010. <http://en.wikipedia.org/wiki/IEEE_802.11>

“MC13202 Data Sheet.” Freescale.com. Web. 5 Jul. 2010. <http://cache.freescale.com/files/rf_if/doc/data_sheet/MC13202.pdf>

“MRF24J40 Data Sheet.” Microchip.com Web. 5 Jul. 2010. <http://ww1.microchip.com/downloads/en/DeviceDoc/DS-39776b.pdf>

“MRF24WB0MA/MRF24WB0MB Data Sheet.” Microchip.com. Web. 5 Jul. 2010. <http://ww1.microchip.com/downloads/en/DeviceDoc/70632A.pdf>

“PAN1315 Data Sheet.” Panasonic.com. Web. 12 Jul. 2010.<http://www.panasonic.com/industrial/includes/pdf/PAN1315_Full- Spec_2010.pdf>

“RN-41 Data Sheet.” Rovingnetworks.com. Web. 12 Jul. 2010. <http://www.rovingnetworks.com/documents/RN-41.pdf>

“XBee 1mW Chip Antenna.” Sparkfun Electronics - XBee 1mW Chip Antenna. Web. 5 Jul. 2010. <http://www.sparkfun.com/commerce/product_info.php?products_id=8664?>

“ZigBee.” ZigBee - Wikipedia, the free encyclopedia. Web. 5 Jul. 2010. <http://en.wikipedia.org/wiki/ZigBee>

“Z-Wave.” Z-Wave - Wikipedia, the free encyclopedia. Web. 12 Jul. 2010. <http://en.wikipedia.org/wiki/Z-Wave>

General

Beer, Ferdinand P., Johnston Jr., E. Russell, Clausen, William E. “Vector Mechanics for Engineers – Statics, Eighth Edition”. McGraw-Hill. 2007.

Beer, Ferdinand P., Johnston Jr., E. Russell, Clausen, William E. “Vector Mechanics for Engineers – Dynamics, Eighth Edition”. McGraw-Hill. 2007.

Halliday, David, Resnick, Robert, Walker, Jearl. “Fundamentals of Physics Extended – Fifth Edition”. John Wiley and Sons, Inc. 1997.

Da Vance, Engers. Smith, Stephen. “YouTube – UCF QuadCopter G3 2 Motors Test”. http://www.youtube.com/watch?v=tpUYEwEg12w. June, 2010. <http://en.wikipedia.org/wiki/Quadrotor . “Quadrotor – Wikipedia”. Wikipedia.com. 2010. >

B. Permissions

Invensense

Engers F Davance to **stephensmith19.**

- Hide quoted text -

---------- Forwarded message ----------
From: **Beeta Ouraie** <bouraie@invensense.com>
Date: Mon, Aug 2, 2010 at 7:23 PM
Subject: RE: Permision To Reprint
To: Engers F Davance <edavance3@gmail.com>

Hi Engers,

Yes – you have permission to use InvenSense released imagery, photos and diagrams, but please use this in small print the bottom of the image:

Copyright © InvenSense, Inc. All rights reserved.

Please let me know if you have any questions, thank you.

Best,
Beeta Ouraie

InvenSense, Inc.
Marketing Communications

408.988.7993 x273

From: Engers F Davance [mailto edavance3@gmail.com]
Sent: Monday, August 02, 2010 4:07 PM
To: pr@invensense.com
Subject: Permision To Reprint

To whom it may concern,

Hi, my name is Engers F Davance. I attend the University of Central Florida and I'm doing a senior design project on a Quad-Copter autonomous vehicle. I would like to have permission to reprint data and diagrams that I found on your site. This will be used for my documentation paper.

Thank You
Engers F Davance

Gmail by Google

Joshua West <joshuagwest@gmail.com>

FW: Permission to use figures, tables, and information

Marciales, Carlos <Carlos.Marciales@atmel.com> **Tue, Aug 3, 2010 at 9:33 AM**
To: joshuagwest@gmail.com

Joshua,

Please see the comments below. Permission granted . . .

Regards,
Carlos

-----Original Message-----
From: Gillie, Miles
Sent: Tuesday, August 03, 2010 8:52 AM
To: Marciales, Carlos
Subject: RE: Permission to use figures, tables, and information

Hi Carlos

I checked with legal and it sounds fine. This is probably fair use since it is for educational, not commercial purposes. If you would, please ask them to:

1. Follow our trademark usage guidelines (as available at Atmel.com) when using our trademarks (Atmel(r), AVR(r), etc.), and

2. To attribute the material to us, with a notice at the end of his paper, along the lines of:

"Technical information from Atmel Corporation datasheets and manuals is subject to Atmel copyrights, and is reprinted with permission."

Regards,

Miles

****
Miles Gillie
Regional Sales Manager
Atmel Corporation
Raleigh, NC
Office : 919-846-3382
Mobile: 919-523-3320
****

Sparkfun

Joshua West <joshuagwest@gmail.com>

Documentation permissions

Lara Boudreaux <lara@sparkfun.com> **Tue, Aug 3, 2010 at 1:09 PM**
To: Joshua West <joshuagwest@gmail.com>

Joshua,

You may use our documentation and such as long as you credit SparkFun somewhere in your paper.

Thanx for asking Joshua. And good luck with the project!

Lara

On 8/3/10 11:07 AM, "Joshua West" <joshuagwest@gmail.com> wrote:

> Dear Sparkfun,
>
> I am a student at the University of Central Florida and my group is
> currently working on our Senior Design Project. For this project
> we must write up a document with our research and designs. We would
> like to ask for your permission to use your tables, figures, and
> information that is present inside of your data sheets and manuals in
> our final paper.
>
> Thank you,
> Joshua West
> Group 3

Jeannie Goerlitz to me

You may reprint the data on our open website, as long as you give credit to csr.com.

Regards,
Jeannie

-----Original Message-----
From: Engers F Davance [mailto:edavance3@gmail.com]
Sent: Sunday, August 01, 2010 5:24 AM
To: DL-GBL-CommercialSalesInquiry
Subject: CSR Web Site Enquiry

An enquiry has been sent to you from the CSR website:

Date: 01/08/2010

Title: Mr

First name: Engers F

Last name: Davance

Email address: edavance3@gmail.com

Company: University of Central Florida

Address: 5864 Diego ST APT B
Orlando FL, 32807

Country: United States

Telephone: 4072791830

Fax:

Enquiry: To whom it may concern,

Hi, my name is Engers F Davance. I attend the University of Central Florida and I\'m doing a senior design project on a Quad-Copter autonomous vehicle. I would like to have permission to reprint data and diagrams that I found on your site. This will be used for my documentation paper.

Thank You
Engers F Davance

MaxBotix.com

Stephen Smith to me

show details 10:09 I

Forwarded conversation
Subject: **Copyright permissions**

From: **Stephen Smith** <ssmith@knights.ucf.edu>
Date: Sat, Jul 31, 2010 at 11:41 PM
To: info@maxbotix.com

To whom it may concern,

I am an engineering student at the University of Central Florida and
am currently working on a senior design group project. We, group # 3,
would like written permission to reproduce various photographs,
and figures from the MaxBotix website and integrate them
into our project documentation. We are specifically interested in the
MaxSonar-EZ ultrasonic proximity sensors for use in our quad-copter.

We will also credit each specific instance properly.

Thank you,
Stephen Smith
ssmith@knights.ucf.com

From: **Scott Wielenberg** <scott@maxbotix.com>
Date: Mon, Aug 2, 2010 at 1:00 PM

To: Stephen Smith <ssmith@knights.ucf.edu>
Cc: Bob Gross <bob@maxbotix.com>, sales@maxbotix.com

Thanks for the email. Yes, you may use the information provided from our website provide proper credit is given like you have already stated. Please let me know if you have any questions.

Best regards,

Scott Wielenberg
Technical Support & Sales
of MaxBotix Inc.
Phone: (218) 454-0766 Ext. 2
Fax: (218) 454-0768
Email: scott@maxbotix.com
Web: www.maxbotix.com

Storediydrones.com:

On Wed, Jul 28, 2010 at 7:33 AM, Sales DIYdrones <sales@storediydrones.com> wrote:

> Hello Stephen,
>
> I am not sure of the duration for the offer but I am finding out and will let you know.
>
> Last time I checked, we were open source but I am double checking just to be sure. You can go ahead and use items from the website (and give credit for them) as needed for your project.
>
> Best regards,
> Edna
>
> On Tue, Jul 27, 2010 at 4:14 PM, Stephen Smith <ssmith@knights.ucf.edu> wrote:
>
>> To whom it may concern,
>>
>> I am an engineering student at the University of Central Florida and
>> am currently working on a senior design group project. We, group #3,
>> would like written permission to reproduce various photographs,
>> datasheets, and figures from the DIYDrones website and integrate them
>> into our project documentation. We will also credit each specific
>> instance properly. Secondly, we have an interest in purchasing the
>> MediaTek MT3329 GPS 10Hz GPS receiver module which is being offered
>> for a limited time at $29.99. It would be appreciated if you could
>> give us an estimate, if possible, as to when the special offer will
>> end.
>>
>> Thank you,
>> Stephen Smith
>> ssmith@knights.ucf.com

Microchip.com:

Use of Microchip Copyrighted Material

Microchip documentation (including but not limited to data sheets, manuals, etc.), images, website, and other original creations are valuable assets protected by copyright law. Microchip aims to protect these assets while encouraging the broad dissemination of product literature and related information in order to reach the widest market for Microchip products. To that end, we often consider requests by customers, distributors, and other parties to reproduce, translate, and/or reprint our copyrighted material in a book, CD, magazine, or other reference material to be sold or distributed on the open market. When we approve such requests, Microchip still owns all rights to the copyrighted material including any translations of such material.

If you would like to reproduce, translate, and/or reprint Microchip copyrighted material for commercial purposes you must request Microchip's written permission by following the 3-step process described below. Microchip's written permission is not required for personal use or educational (non-profit) use of copyrighted material.

Personal Use of Copyrighted Material: If you use Microchip copyrighted material solely for your personal use you do not need Microchip's written permission to use such material. However, distribution or reproduction of such materials and images to others (including posting on a website) does require Microchip's written permission.

Educational and Non-Profit Use of Copyrighted Material: If you use Microchip copyrighted material solely for educational (non-profit) purposes falling under the "fair use" exception of the U.S. Copyright Act of 1976 then you do not need Microchip's written permission. For example, Microchip's permission is not required when using copyrighted material in: (1) an academic report, thesis, or dissertation; (2) classroom handouts or textbook; or (3) a presentation or article that is solely educational in nature (e.g., technical article published in a magazine). Please note that offering Microchip copyrighted material at a trade show or industry conference for the purpose of promoting product sales does require Microchip's permission.

Permission to Use Copyrighted Material: If you intend to use Microchip copyrighted material for commercial purposes or you are not sure whether a particular use falls under the "fair use" exception described above, then you should seek permission from Microchip by following the 3-step process below.

All parties using Microchip copyrighted material (whether for commercial or educational purposes) must properly acknowledge Microchip's ownership of its copyrights and properly use Microchip's trademarks. If you would like to receive Microchip's trademark usage guidelines, please send your request to the Legal Department by sending an e-mail to legal.department@microchip.com or by faxing your request to (480) 792-4112.

To request permission to use Microchip copyrighted material, please follow these 3 steps:

1. Complete the Request to Use Microchip Technology Inc. Copyrighted Material. This form is posted below.

2. After completion, please return the Request to Use form to the Legal Department by e-mailing it to legal.department@microchip.com or faxing it to (480) 792-4112. In most cases, Microchip will review and respond to your request within 2 weeks.

3. If the request is approved, Legal will prepare a letter granting you permission to use Microchip copyrighted material subject to certain conditions. For example:
• the contents of the material may not be modified,
• the most current version of the copyrighted material must be used,
• Microchip copyright notices must be posted in the reprinted material or publication,
• Microchip's trademarks must be properly used, and
• the recipient of the material must comply with applicable export regulations.

Please note that most copyrighted materials are available on Microchip's website. For any copyrighted material that is not available on Microchip's website, you may contact the Technical Publications Manager of the Marketing Communications Department at legal.department@microchip.com.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		Applications	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DLP® Products	www.dlp.com	Communications and Telecom	www.ti.com/communications
DSP	dsp.ti.com	Computers and Peripherals	www.ti.com/computers
Clocks and Timers	www.ti.com/clocks	Consumer Electronics	www.ti.com/consumer-apps
Interface	interface.ti.com	Energy	www.ti.com/energy
Logic	logic.ti.com	Industrial	www.ti.com/industrial
Power Mgmt	power.ti.com	Medical	www.ti.com/medical
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Space, Avionics & Defense	www.ti.com/space-avionics-defense
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Video and Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless-apps

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Ublox.com:

Terms & Conditions

Print E-mail pag

Note: the following only applies to content provided on this website. For general terms and conditions for doing business with u-blox, click here.

Your access to and use of the website www.u-blox.com and documents on this website is subject to the following terms and conditions. By accessing and browsing this site, you agree to accept without limitation or qualification the following terms and conditions

Use of information
This website and documents on this website (hereinafter this website) are put at your disposition for information purposes only and information contained therein is not binding. A download, copy or print of this website is only permitted for personal non-commercial use. You may not distribute, modify, transmit, repost or use this website for commercial purposes or establish links to it without written permission of u-blox.
Disclaimer

THIS WEBSITE MAY CONTAIN CERTAIN FORWARD-LOOKING STATEMENTS, WHICH CAN CONTAIN STATEMENTS REGARDING POTENTIAL NEW PROJECTS OR POTENTIAL FUTURE REVENUES OR STRATEGY, PLANS OR INTENTIONS. ALL FORWARD-LOOKING STATEMENTS ARE BASED ONLY ON DATA, APPRAISALS AND ASSUMPTIONS MADE AND AVAILABLE TO U-BLOX AT THE TIME OF THE PREPARATION OF SUCH INFORMATION. THE ACTUAL SITUATION AND RESULTS CAN BE MATERIALLY DIFFERENT FROM FORWARD LOOKING STATEMENTS. U-BLOX DOES NOT UNDERTAKE TO UPDATE ANY FORWARD-LOOKING STATEMENTS CONTAINED ON THIS WEBSITE.

This website contains no invitation, offer or recommendation to invest or deal in securities.
u-blox makes no warranties or representations as to the correctness, completeness, accuracy or quality of this website. u-blox reserve the right to make changes, deletions, corrections andor improvements to this website and to products described on this website, at an time without any notice. u-blox shall not be liable for any direct, incidental, consequential, indirect or punitive damages arising out of access to, use of or inability to use this website or any errors or omissions in the content of this website. u-blox assumes no responsibility and shall not be liable for any damages to your computer or viruses that may infect your computer equipment or other property resulting from your access to this website.

No statement on this site is to be interpreted as a promotion or advertisement for a product or for the use of a product that is not authorized by the laws and regulations in your country of residence.

u-blox does not review any third-party websites that contain links to its own website and assumes no responsibility for the content of any such off-site pages or any other sites linked to its website.