ALTERNATIVE QUADCOPTER DESIGN

By

Christopher S. Moy

Final Report for ECE 445, Senior Design, Fall 2013

TA: Mustafa Mukadam

11 December 2013

Project No. 43

Abstract

The design goal of this project was to construct a quadcopter which moves by individually rotating each of the four main motors while leaving the body of the vehicle flat, as opposed to the traditional method, in which the motors are fixed and the entire flight plane rotates.

The overall goal of the project, flight, was not achieved. This paper will discuss the functional components of the project including the main control board, servos and servo control, the power management system and the feedback sensors, as well as the theory behind the yet-to-be-implemented flight control algorithm.

Contents

1. Introduction

The goal of this project was to create a quadcopter which could move in the X and Y directions while the main flight plane remained completely flat. This goal was going to be accomplished by mounting each motor on a specially made two-axis rotational mount, each with two servos, one Y-axis and one Z-axis, which would then tilt the entire motor in the direction of travel and provide both lift and forward thrust. The resulting action would operate very similarly to a front-wheel-drive car, dragging the entire vehicle in the desired direction of travel while the rear propellers provided lift and stabilization.

1.1 Project Motivation

The challenge that this project was intended to overcome was the inability of a traditional quadcopter to transport fragile cargo. When a traditional quadcopter moves, it can tilt in excess of 45 degrees. This action creates both rotational acceleration as well as a sustained state of tilt on any cargo, which is unacceptable for many objects such as glassware or a hard drive for example. By keeping a stable flight platform, any payload would remain in a stable position throughout flight. Furthermore, this design method simplifies the control theory by eliminating one of the two frames of motion which need to be accounted for in a traditional quadcopter. In traditional control design, there is a lab frame, or the X-Y-Z angles of the apparatus in relation to the ground/operator, as well as an inertial frame, which can be thought of as the various angles in relation to the apparatus itself. In this design, those frames are combined because the quadcopter can be thought of as being in a constant hover state, with an ideal tilt of zero degrees on the X-Y plane.

1.2 Functions and Features

The basic high-level requirements and features required to make this project functional are a combination of those that would exist traditionally, as well as some unique ones.

1.2.1 Functions

The functional requirements of the quadcopter included:

- Stable flight
- User control
- Efficient power management
- Safety features

1.2.2 Features

The requirements were approached via the following built in features:

- Arduino/shield flight control board
- Independent gyroscope and accelerometer
- Robust hover flight algorithm
- 4 channels radio user input
- 5V Low Dropout Out Regulator
- Manual/Automatic Kill-switch

1.3 Project Modules

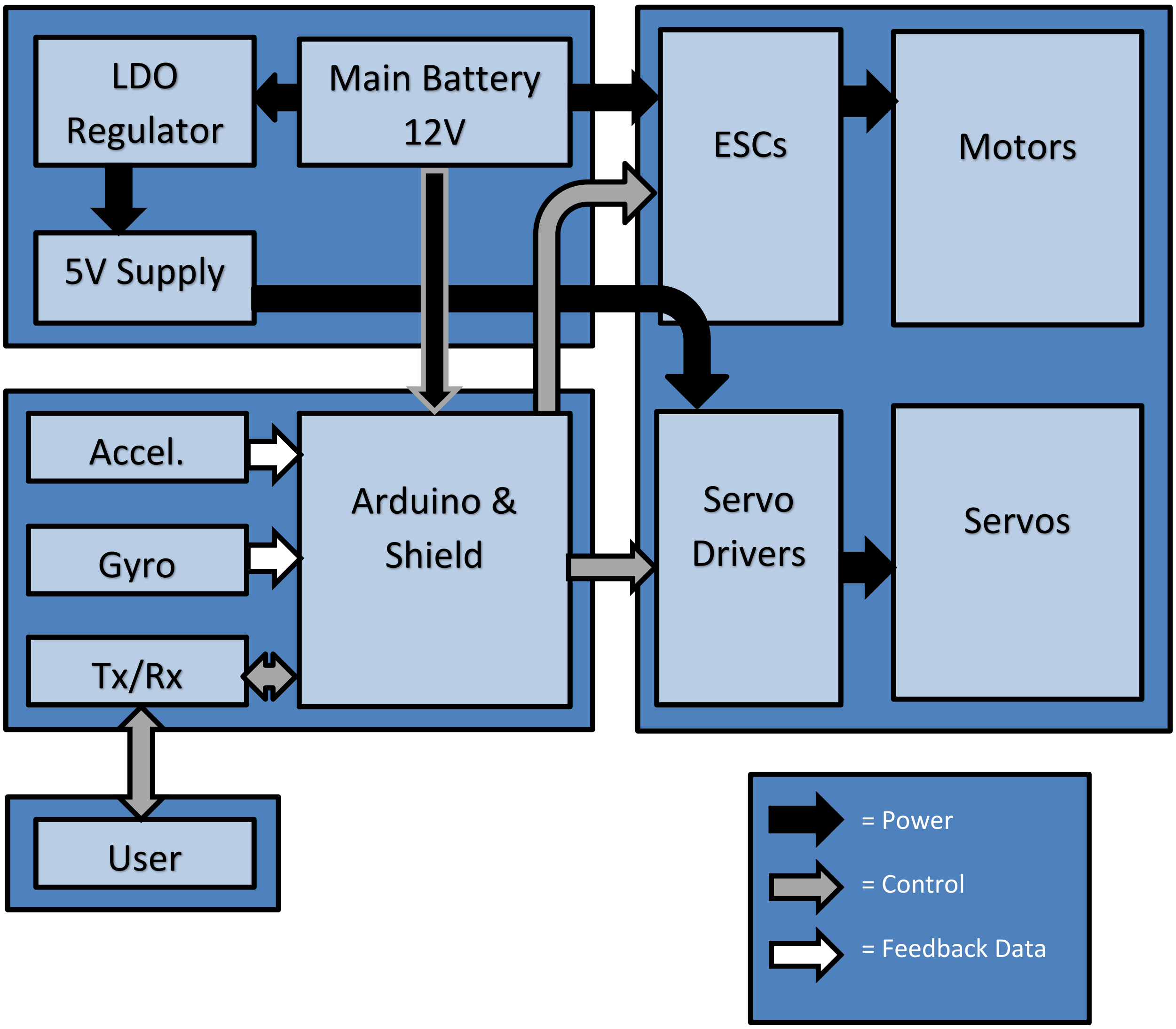

Figure 1.1: Block Diagram

1.4 Module Descriptions

1.4.1 Arduino Mega ADK/Prototype Shield

The Arduino Mega platform is the largest of all of the Arduino boards. It has 14 PWM outputs, of which 12 can be used concurrently, as well as 40 additional digital I/O pins and 16 Analog-to-Digital Converter pins, allowing for unparalleled design flexibility throughout the course of the project. Furthermore, the Mega offers a wide variety of serial communication via dedicated I^2C pins, 4 UART channels and an extra USB port, giving a limitless range of possible peripherals. The Arduino acts as a flight control board, communicating with the Feedback Sensors mounted to the shield, processing data, and then outputting stabilizing control signals to both the ESCs and Servo Control Board.

1.4.2 Accelerometer

The ADXL345 3-axis accelerometer is an IC which provides 10-degrees of precision for measuring linear acceleration along the X-Y-Z axes. It communicates with the Arduino via I^2C serial. In the implemented algorithm only the X-Y axes will be used to correct steady-state error in flight platform angle.

1.4.3 Gyroscope

The ITG-3200 3-axis gyroscope is an IC which provides 13-degrees of precision for measuring rotational acceleration along the X-Y-Z axes. It communicates with the Arduino via I^2C serial. In the implemented algorithm all three axes will be used. The X-Y axes provide basic stabilization while the Z-axis provides basic heading stability.

1.4.4 Radio Transmitter/Receiver (Tx/Rx)

The Tx/Rx Unit is a basic 2.4 GHz Band Radio Control System. The controller has six channels, although in this project only four were used. There are also 2 switches, one of which is used as a manual kill-switch for safety. The receiver outputs PWM signals of period 20ms with duty cycle between 5-10% and amplitude 5V to the Arduino.

1.4.5 Main Battery

Main Battery is a 3S1P (meaning three cells in series, none in parallel) 11.1V, 5000mAh Lithium Polymer battery pack. When fully charged each cell is at 4.2V, or 12.6V for the whole system. It is connected to the rest of the system via a simple bullet-connector harness.

1.4.6 Low-Dropout Regulator/5V Supply

The LDO 5V-regulator consists of two main components, a 5A-adjustable regulator and a 10A power PNP transistor, mounted on a monolith of heat sinks. An output voltage of 5V is set via a pair of voltage set resistors and input/output ripple is managed by 47 µF decoupling capacitors. Because the circuit is built vertically, it has a small 2 in^2 footprint.

1.4.7 Electronic Speed Controllers

The Electronic Speed Controllers (ESCs) are modules which take in a DC voltage, here about 12V from main battery, as well as a PWM signal very similar to the ones used to control servos and output and combine them to provide a three-phase power output to the motors. The ESCs are specially designed to operate a 3S1P battery and have a maximum amperage rating of 30A.

1.4.8 Brushless DC Motors

The motors are Turnigy 2217 outrunner motors. What this means is that each brushless motor has a shell which spins relatively slowly, but with a high torque coefficient, around the center axis, making these motors ideal propeller drivers.

1.4.9 Servo Driver Board

This board is a custom made board which implements the PCA9685 LED driver chip. The function of this chip is to output 16 independent PWM channels at a frequency between 40-1000Hz and a 12-bit duty cycle resolution. This means that at the chosen operating frequency of 160Hz, the PWM is accurate to 1.53 µs, making the chip a highly accurate servo driver. The chip takes a main 5V power input from the LDO to power both the servos and the chip's TTL logic, and takes an I^2C input from the Arduino to set each channel. The board also has two status LEDs, one indicating main power and the other indicating successful I^2C communication.

1.4.10 Coreless Servos

The chosen servos are coreless, 4.6 Kg-cm servos. A coreless design ensures maximum strength-to-weight ratio.

2 Design

2.1 Control Board

The Control Board acts as the brain of the quadcopter. Every element of the apparatus is connected to it in some fashion, so connectivity was a main design concern. Also considered were ease of use, I/O capability, processing speed, space efficiency, power efficiency and cost.

2.1.1 Microcontroller

The microcontroller design decision actually occurred in 2 parts. The first part, the ideal planning phase, resulted in a decision to use the newly released TI MSP430F5529 LaunchPad platform. Pertinent statistics are compared in the following table:

	MSP430F5529	Arduino Mega
Cost	$12.99	$39.97
I/O	40 (32 Digital, 8 ADC)	70 (54 Digital, 16 ADC)
PWM Channels	5	15
Serial ports	4 (1 I^2C, 1 UART, 2 SPI)	6 (1 I^2C, 4 UART, 1 SPI)
Memory	128 KB Flash, 8 KB RAM	256 KB Flash, 8 KB RAM
Logic Voltage	3.3V	5V
Clock Speed	25 kHz	16kHz
Coding Environment	TI CodeComposer	Arduino IDE
Power Consumption	3 µA (30 µA w/ LEDs)	60 mA
Footprint	33.55 cm^2	55.08 cm^2

The LaunchPad platform was initially chosen for a number of reasons. Physically, it is a significantly smaller board, requiring almost half the surface space of the Arduino Mega. Apart from size, the TI Board is 56% faster than the Arduino. Speed was truly critical in this application in particular, as the faster the control loop executed, the more stable the flight of the quadcopter. All peripherals chosen operated on I^2C, which only requires one line for any number of devices, so the additional serial ports were not necessary. Because the Servo Control Board was responsible for outputting PWM to the servos, only four onboard PWM channels were needed, which the LaunchPad could have provided. The LaunchPad also consumes an astounding 2000% less power during normal operation, a huge benefit in a power-sensitive application. Unfortunately, due to a group design issue near the end of the semester, the design had to be changed very quickly about 10 days prior to deadline, leading to a switch to Arduino for a number of reasons. First and foremost, the Arduino provides an unparalleled development environment compared to TI. While CodeComposer is a fairly easy to use piece of software after some instruction, it cannot match the sheer quantity of available libraries and sample code of that on the Arduino, making development time significantly shorter on the Arduino platform. Also, the LaunchPad must be powered by a 3.3V input. While the original design included a 3.3V regulator, the final design only included a 5V regulator and main battery. The Arduino however, has an onboard LDO capable of handling input voltages up to 20V while also providing a 3.3V regulator. This meant that the Arduino could be connected to main battery and power the 3.3V sensors via its onboard regulator, eliminating the need for an external 3.3V step-down regulator altogether. The final consideration were the readily available Arduino prototyping shields available for the Mega. Once again due to time constraints, it was not feasible to manufacture a shield for the LaunchPad, whereas one was built rather easily for use on the Mega.

2.1.2 Sensors

There were two options considered for the Gyroscope and Accelerometer, either two freestanding chips which used Analog Voltages to relay values, or a single Inertial Measurement Unit (IMU) which contained both sensors and output all data via a single I^2C interface. The choice for an IMU was made due to space constraints, complexity and accuracy of data. The IMU used required about 3 cm^2 and had 4 total pins, whereas the 2 breakouts for each sensor would have required a total of 10 cm^2 and had 3 pins each for a total of 6. While the Analog Voltage readings were enticing because of their ease in reading when compared to I^2C, the data involved is much more susceptible to noise when output as an Analog Voltage as opposed to being processed by onboard ADCs and then output serially.

2.1.3 Receiver

Originally, the receiver was going to be an XBee antenna which would transmit data back and forth via a computer base station. This design however, added an unnecessary degree of complexity to the design, as the ZigBee protocol can tend to lag, and the computer processing ability was unneeded for the basic user controls we designed. The chosen receiver was 6-channel RC receiver. It was chosen because it was inexpensive and provided a fast response to user input.

2.2 Power Supply

2.2.1 Main Battery

The Main Battery chosen was a 5000mAh, 20C, 3S1P LiPo Battery with a hard case. LiPo was chosen because it provides the highest energy density of all readily available and applicable batteries. 5000mAh was chosen because we wanted a hover time of greater than 10 minutes [1]. We calculated that hover would consume about 27A steady state current, so this would provide:

$$\frac{5\ A-hr}{1} * \frac{1}{27\ A} * \frac{60\ min}{1\ hr} = 11.1\ min \tag{2.1}$$

2.2.2 3.3V Regulator

The 3.3V regulator was an initial design concern which was abandoned when the decision to use Arduino as the Control Board was made.

2.2.3 5V Regulator

The 5V regulator was supposed to be a switch-mode regulator for the sake of high-efficiency, but due to last minute design issues was changed to an LDO for the sake of size and ease of design. Although the voltage regulator used has a variable output voltage, it is very easily set via two resistors.

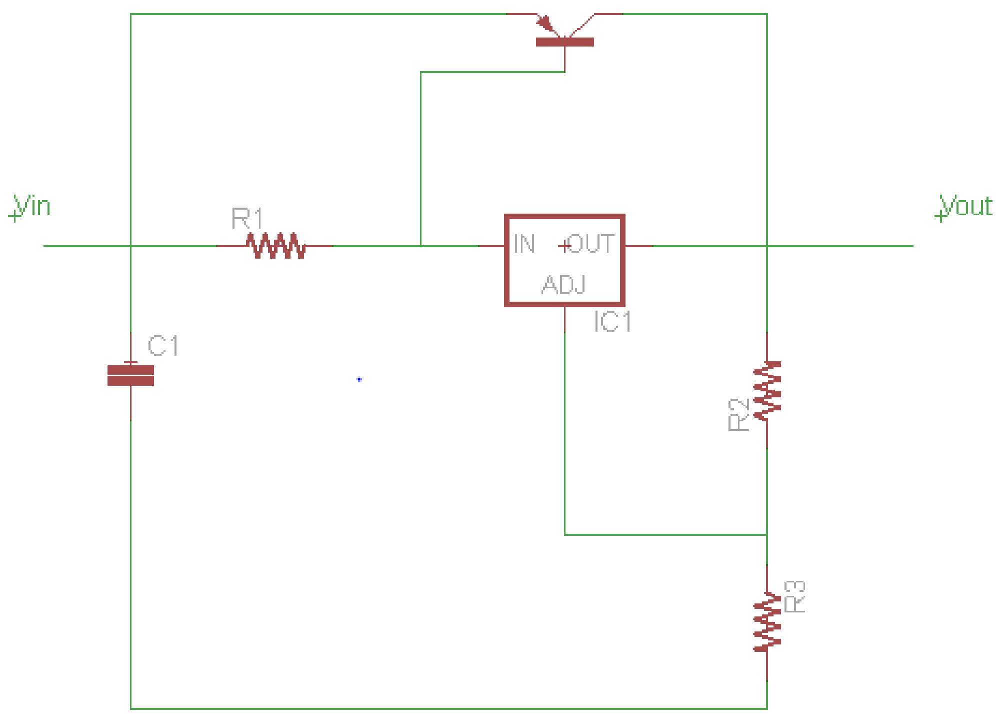

Figure 2.1: 5V Regulator Schematic

In this schematic, the selector resistors are R2 and R3, with R2 being a set 240Ω resistor, and R2 being chosen using the equation [2]

$$Vout = Vref\left(1 + \frac{R3}{R2}\right) + Iadj * R3 \tag{2.2}$$

With Vout being the desired Voltage. Vref being the voltage across R2 (1.25V with a variance of +/- .04V) and Iadj being 50 µA (with a variance of +/- 5 µA). From this R3 is found to be 713Ω, which is implemented in circuit by a 680Ω in series with a 30Ω. The resulting circuit produces an open circuit voltage of 4.99V.

2.3 Motors

2.3.1 Electronic Speed Controllers

Most ESCs behave in the exact same way, by taking in DC power and a PWM signal and outputting a three-phase power source to a three-phase brushless DC motor. The main consideration in choosing an ESC is, for this reason, current ratings. We calculated that:

Battery		
Load:	14.36	C
Voltage:	10.13	V
Rated Voltage:	11.10	V
Flight Time:	4.2	min
Mixed Flight Time:	5.3	min
Hover Flight Time:	10.4	min
Weight:	393	g
	13.9	oz

Motor @ Optimum Efficiency		
Current:	7.65	A
Voltage:	10.63	V
Revolutions*:	9590	rpm
electric Power:	81.3	W
mech. Power:	66.8	W
Efficiency:	82.1	%

Motor @ Maximum		
Current:	17.95	A
Voltage:	9.99	V
Revolutions*:	6810	rpm
electric Power:	179.3	W
mech. Power:	132.4	W
Efficiency:	73.8	%
est. Temperature:	66	°C
	151	°F

Motor @ Hover		
Current:	6.14	A
Voltage:	10.72	V
Throttle (linear):	57	%
electric Power:	65.8	W
mech. Power:	53.7	W
Efficiency:	81.5	%
est. Temperature:	36	°C
	97	°F

Total Drive		
Drive Weight:	921	g
	32.5	oz
All-up Weight:	1840	g
	64.9	oz
add. Payload:	669	g
	23.6	oz
Current @ Hover:	24.57	A
P(in) @ Hover:	272.8	W
P(out) @ Hover:	214.7	W
Efficiency @ Hover:	78.7	%
Current @ max:	71.82	A
P(in) @ max:	797.2	W
P(out) @ max:	529.7	W
Efficiency @ max:	66.4	%

Figure 2.2: Flight Power Draw Data

The maximum current needed would be 17.95A. We chose 20A, 30A burst ESCs to ensure that current draw never led to overheating.

2.3.2 Motors

The goal with the motors was to maximize power efficiency, as power efficiency is the crux of any quadrotor project. At the same time, as this quadcopter was guaranteed to weigh more than a similar fixed-rotor model, we wanted to ensure that each motor could output a maximum of at least 1kg thrust to endure that at a goal weight of less than two kilograms, the quadrotor could hover at 50% power. These motors were chosen in combination with the propeller size used, because at a 3S1P voltage level, 17.95A is the lowest amperage in comparable motors to produce the needed 1kg of thrust (1.07 kg per motor).

2.4 Servos

2.4.1 Servo Control Board

The main goal of the Servo Control Board was to eliminate the need for the Flight Control Board to output the 8 PWM signals. The servos update at a rate of 160Hz. This means that a 1-2ms pulse must be sent every 6.25ms. Because the processor cannot output multiple signals concurrently, this makes outputting the eight PWM signals, on top of the four needed for the motors, impossible. To output the servo signals alone the frequency would need to be slowed to 60 Hz. Even slower times would be necessary to include time for control and motors. Fortunately, the Servo Control Board eliminates this problem by taking over the PWM output process [3]. It takes in a simple I^2C command from the Arduino, and updates the PWM signal on a given channel. The benefit of this chip however, over all other PWM driver chips, is that is asynchronous. This means that if the Arduino does not update the PWM of a given channel, the chip will continue to output the same PWM on that channel at the set frequency. This eliminates the huge task of repeatedly updating the control registers, freeing the Arduino to execute the control loop. There are two included status LEDs. The yellow LED is connected to main 5V power, and indicates that the board is receiving 5V. The red LED, is connected to PWM channel 11. Correct functionality by the red LED indicates that the board is receiving correct commands from the Arduino and updating the registers. For simplicity of design, the chip and the servos are powered off of the same 5V line, leading to 4 total pins, 5V, GND, SCL and SDA.

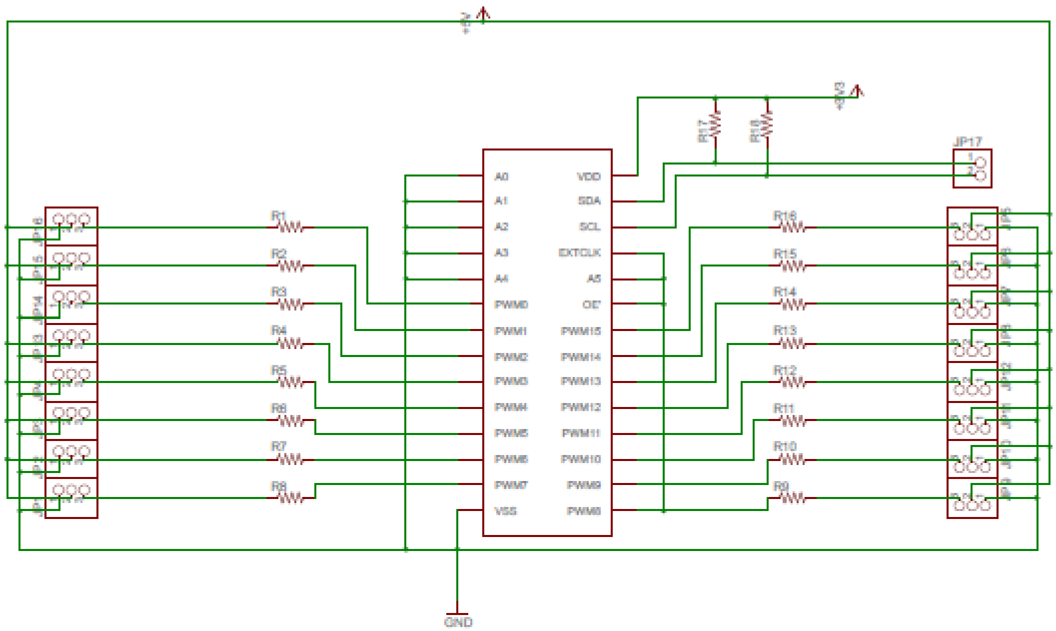

Figure 2.3: Servo Driver Schematic

2.4.2 Servos

The servos were chosen based on their strength-to-weight ratio, as well as a digital vs analog internal control. In strength, the design would output a maximum of 1 kg of force along a 6 cm boom arm at a certain angle. A 4.6 kg-cm was chosen because at 6 cm, the force output by the servo would allow for a maximum servo angle of:

$$Angle = \ sin^{-1}\left(\frac{4.6\ kg - cm}{6cm * 1.07kg}\right) = 45\ degrees \tag{2.3}$$

Which is the absolute maximum angle we would have wanted to tilt the forward rotors. A coreless servo design decreases longevity but also reduces weight. A digital model was chosen because digital servos can be updated up to 300 times per second, as opposed to analog servos which have a universal update frequency of 60 Hz.

2.5 User Control

2.5.1 Transmitter

The transmitter was originally planned to be an XBee antenna connected to a computer base station, using a video game remote as the control. This however, provided an unnecessary degree of difficulty to the design, as this method of control is very programming intensive without any appreciable results over radio control except for a more aesthetic user interface. The final decision was to use a 6-channel, 2.4GHz band radio transmitter. It is fairly easy to use and provides fast response from user to quadcopter.

3. Design Verification

A specific design goal of this project was to build the quadcopter modularly. What this means for the project as a whole is that each component could be built separately and then be brought together upon completion to create a working vehicle. This is especially important in quadcopter design because crashes are a frequent occurrence. In a system that is designed modularly, individually damaged sections can be easily replaced in the event of a malfunction, instead of having to rebuild the apparatus from scratch. Modularity also provides an avenue for project improvement during and after the conclusion of the project. Individual components can be adjusted or completely redesigned without having to deconstruct the entire project.

3.1 Control Board

3.1.1 Microcontroller

The main goal of the microcontroller was to ensure both proper peripheral communication, and a minimum of 100Hz control loop [4]. First of all, proper peripheral function was confirmed via serial output from the Arduino (on the serial monitor). Using test code, each of the sensors was read separately through the Arduino's I^2C channel and confirmed functional. Furthermore communication with the servos was confirmed both through illumination of the red servo test LED and the functionality of the servos.

3.1.2 Sensors

The main goal of the sensors was to have the ability to output absolute X-Y angle, as well as a relative Z angle, with some drift over time. This functionality was confirmed first by confirming that the raw data read from the sensors corresponded to the forces acting upon the quadcopter. Some simple movements of the quadcopter did indeed confirm that this was happening. Using the data sheets of the accelerometer chip and gyroscope chip and dividing the raw data by their respective scale factors, and then combining the data through a complementary filter [5], absolute X-Y angle, as well as Z-angle with some drift, was confirmed.

3.1.3 Receiver

The only function that had to be confirmed with the receiver unit is that it output a variable width PWM signal with pulses ranging from 1-2ms in length as the transmitter was moved from minimum to maximum. While this trend was confirmed to occur, the actual operating range was about 1.05 ms to 1.75 ms, meaning that the map function of the Arduino had to be used.

3.2 Power Supply

3.2.1 Main Battery

The battery had to be able to provide power under full thrust conditions (this means a draw of approximately 80A @ 12V), which was expected to be sufficient because as a 20C, 5mAh battery, it has a theoretical output of 100A. After firmly securing the apparatus to the test bench, the ESCs were opened to full throttle and battery function was monitored. Everything functioned smoothly over three 30 second test runs (battery recharged in between each run to ensure that cycling occurred).

3.2.2 5V Regulator

The 5V regulator was designed and constructed at a very rapid pace, which did not leave much time for testing. The verification for proper functionality was to first use a DMM to measure the open circuit voltage across the regulator. Although initially set to 5.98V, it was reduced to 4.99V, which it held steadily. After this, each servo was set to on in neutral position and connected one at a time until all eight were connected together. With all eight servos on and functional, the voltage from the regulator was 4.96V, well within the 4.8V lower limit set by the servos. One note is that the LDO tends to run hot.

Servos Attached	Vout (V)
0	4.99
1	4.99
2	4.99
3	4.98
4	4.98
5	4.97
6	4.97
7	4.97
8	4.96

Figure 3.1: Regulator Voltage Output

3.3 Motors

3.3.1 Electronic Speed Controllers

The ESCs are quite possibly the most dangerous part of the project because if they malfunction and run the motors unexpectedly, serious injury could occur. This meant that they had to be tested with open load in order to ensure that their three phase output was as expected, with zero output with zero input, input too small, or step input (pulses > 2ms). Using an oscilloscope, functionality was confirmed. The ESCs have a very close band to 1ms < pulse < 2ms.

3.3.2 Motors

Proper functionality of the ESCs was confirmed prior to any motor testing. Then, with propellers removed, the motors were run at a set input of 5% thrust (1.05ms pulses) to verify that the motors ran. After all 4 motors were verified, they were secured to the test bench with motors attached and run up to maximum thrust. The observed maximum current draw was only ~16.5A (measured by the lab bench supply), which was a full Ampere under the expected level.

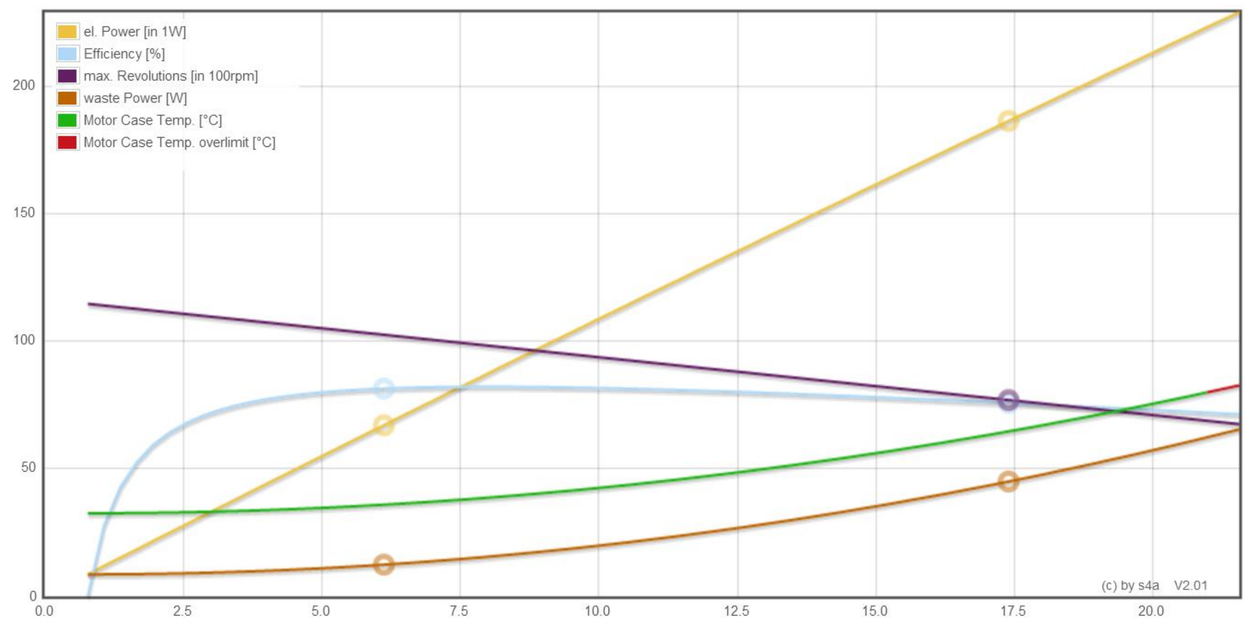

Figure 3.2: Power Consumption Graphs

3.4 Servos

3.4.1 Servo Control Board

Functionality of the Servo Control Board was key the entire project's operation because it served as the controller for the project's mode of transportation, the articulation of the servos. Once main power was connected and the yellow LED (main power indicator) was active, communication with PWM register 11 was attempted. A varied signal from 0 to 4096 (12 bit resolution) was attempted and successful. At this point a single servo was attached at a time to verify that channels 1-8 worked, and then once all eight channels plus the indicator channel were verified, all eight were connected at once and moved around using a bit of test code.

3.4.2 Servos

Testing of the servos was done via the Arduino servo library. Each servo was connected to the onboard 5V regulator on the Arduino as well as a Ground and digital I/O port, and articulated 180 degrees from side to side using test code. Functionality of all eight servos was verified prior to fabrication of the custom mounts. The actual operating range of the servos is about 170 degrees, not 180. Articulation of a full 180 degrees results in gear stripping.

3.5 User Control

3.5.1 Transmitter

The transmitter was verified during the same test as the receiver. With the receiver hooked up to power, the control stick were moved and receiver PWM output was measured to verify that corresponding lengthening and shortening occurred.

4. Costs

4.1 Parts

Part	Manufacturer	Retail Cost ($)	Actual Cost ($)
Talon V2 Frame	Turnigy	56.39	56.39
5000 mAh, 2C Battery	Zippy	31.72	31.72
Accucell-6 50W Balancer/Charger	Turnigy	24.07	24.07
30A BlueSeries ESC x 4	HobbyKing	12.05 x 4	48.20
2217 16turn 1050kv 23A Outrunner x 4	Turnigy	15.38 x 4	61.52
Slow Fly Electric Propellers 11x4.7	APC	3.98	3.98
Slow Fly Electric Propellers RH 11x4.7	APC	3.98	3.98
Power Distribution Board	HobbyKing	2.20	2.20
PCA9685PW LED Controller	NXP	1.39	1.39
HKS-9257 High Speed Servo x 8	GoTeck	9.49 x 8	75.92
9 DOF Sensor Stick	SparkFun	99.95	99.95
Arduino Mega	Arduino	39.97	39.97
Prototyping Shield	SainSmart	12.99	12.99
2.4Ghz 6Ch Tx & Rx V2 (Mode 2)	HobbyKing	25.07	25.07
Custom 2-sided PCB	Parts Shop	49.99	0.00
Custom Machining x 8 hours	Machine Shop	40.00 x 8	0.00
T0-220 Heatsinks x 2	generic	0.95 x 2	0.00
LM338K Voltage regulator	TI	0.81	0.00
TO-3 PNP 10A Transistor	On Semi.	3.21	0.00
100Ω 2-W resistor	generic	0.56	0.00
680Ω .5-W resistor	generic	0.06	0.00
30Ω .5-W resistor	generic	0.06	0.00
Total		**863.94**	**487.35**

4.2 Labor

Name	Hourly Rate ($/Hr)	Total Hours Invested	Total $
Christopher Moy	$40	180	$18,000
Total	**Ideal rate**		**Rate*2.5*hours**

4.3 Total Cost

Labor	Parts	Grand Total
$18,000	$487.35	$18,487.35

5. Conclusion

5.1 Accomplishments

High Level goals which were accomplished during the course of the semester:

- Servo Control Board with serial input and 16 channel 160 Hz output
- Successful full power up of vehicle, including control, servos and motors
- Proper articulation of servos into forwards, backwards, left, right and hover positions
- “Hops” of the quadcopter, showing that flight is possible

5.2 Uncertainties

Some of the High Level Verifications that are still uncertain:

- Although the theory behind the elimination of the inertial frame is solid, it has yet to be tested via a working control algorithm [6]
- Effects of X-Y torque on the stability of the quadcopter
- Effects of servo slippage on the apparatus during flight

5.3 Ethical considerations

The only ethics to consider here in my opinion were those of safety. Quadcopters are inherently dangerous due to rotors spinning in excess of 10,000 rpm, not only to the operator, but to anyone near any testing done. Proper judgment must always be used, when testing, along with the use of safety equipment, and if necessary warnings to the public.

5.4 Future work

In the future, a major improvement in overall alignment and calibration would be the inclusion of 5 3-axis magnetometers which allow for absolute direction sensing. One would be mounted at a perfectly square angle on the underside of each motor mount, as well as having the fifth at a corresponding square angle in the center of the flight deck. What this would do is allow for each servo to articulate so that each motor was oriented at perfect right angles to center at all times. One of the largest sources of error in this project was the fact that the position of each motor had to be hand tuned, with the 45 degree point chosen using a compass and manual adjustment. This also made each servo particularly sensitive to slippage in their mounts, as each time a servo slipped even a small amount the calibration was thrown off. Error in calibration leads to roll-pitch torque, which is the exact kind of torque that can permanently destabilize the vehicle, and was the biggest worry with this servo system to begin with.

References

1) Pca9685 16-channel, 12-bit pwm fm i2c-bus led controller. In (2010). Retrieved from http://www.nxp.com/documents/data_sheet/PCA9685.pdf

2) *Lm138/lm338 5-amp adjustable regulators*. (2013, April). Retrieved from http://www.ti.com/lit/ds/symlink/lm338.pdf

3) Pca9685 16-channel, 12-bit pwm fm i2c-bus led controller. In (2010). Retrieved from http://www.nxp.com/documents/data_sheet/PCA9685.pdf

4) Argentim, L. (2012). Pid, lqr and lqr-pid on a quadcopter platform. In Retrieved from http://fei.edu.br/~psantos/CSR/papers/PID2737725.pdf

5) Jan, P. (2013). Reading a imu without kalman: The complementary filter. Retrieved from http://www.pieter-jan.com/node/11

6) Liang, O. (2013, October 13). *Quadcopter pid explained and tuning*. Retrieved from http://blog.oscarliang.net/quadcopter-pid-explained-tuning/

Appendix A Requirement and Verification Table

Table X System Requirements and Verifications

Requirement	Verification
1. Control Board a. Minimum 100Hz programming loop b. I2C communication with serial peripherals at multiple addresses over the same line	1. Verification a. Currently limited to 100Hz using the millis(10) command b. Concurrent communication in the control loop between the servo control board (0x40) and the IMU (has 3 separate I2C addresses, one for each accelerometer, gyroscope and magnetometer) has been confirmed via serial terminal during servo operation.
2. Sensors a. Minimum 100Hz programming loop b. I2C communication with serial peripherals at multiple addresses over the same line	2. Verification a. Currently limited to 100Hz using the millis(10) command b. Concurrent communication in the control loop between the servo control board (0x40) and the IMU (has 3 separate I2C addresses, one for each accelerometer, gyroscope and magnetometer) has been confirmed via serial terminal during servo operation.
3. Radio Control a. Output a proper control PWM on 4 channels with pulses ranging from 1-2 ms.	3. Verification a. Connected the battery terminal of the receiver to 5V, and then used the remote to change PWM width. When observed on scope actual values ranged between 1.05-1.75ms. This range was corrected using the Arduino's map() function

4. Power System a. 5V regulator provides constant voltage without dropping voltage outside of a 5% band (4.8V is lowest allowable) b. Regulator does not overheat c. Main 12V battery supplies needed current to all motors and the 5V supply without failing or overheating	4. Verification a. While running all 8 servos and the control board off of the regulator, the voltage dropped to 4.95V, about a 1% band and well within the system's limits b. Questionable; the unit can run hot after extended periods due to inadequate heat sink, but has not failed c. Lacking a large enough standardized resistor to run 100A, I trusted the manufacturer specs on this one
5. Servo Control a. I2C communication with Control Board b. Confirm that the PWM output is a 1-2ms pulse with a 160Hz frequency (one pulse every 6.25s) c. Proper gear actuation (90 degrees minimum)	5. Verification a. There are 2 status LEDs on the control board. The yellow one is connected to main power and confirms that the board is "on". The red one is powered via PWM signal, and confirms that a PWM is being produced by the on-board chip. The red LED lights, confirming PWM b. Scope confirmed that the PWM's occurred at the expected values in a cascaded order c. Occurred at about 1.35ms-1.80ms