

Quad-Copter

David Malgoza, Engers F Davance Mercedes,
Stephen Smith, and Joshua West

School of Electrical Engineering and Computer
Science, University of Central Florida, Orlando,
Florida, 32816-2450

Abstract — This paper is a presentation of the design methodology and realization of the Quad-Copter, a model aircraft based on a four-propeller design. The Quad-Copter can be controlled by radio transmission or operate under the guidance of limited autonomous protocols. Flight stability of the Quad-Copter is achieved using a five degrees of freedom (DoF) inertial measurement unit (IMU). Sensor data is integrated and processed using a proportional-integral-derivative controller (PID controller), a feedback loop maintained by an on-board Atmel microcontroller. All subsystems of the Quad-Copter are designed to minimize weight and cost where practical.

Index Terms — Kalman filters, Microcontrollers, Oscillators, Robot programming, Sensor systems and applications

I. INTRODUCTION

The Quad-Copter is a small lightweight hover-capable vehicle that can be controlled over a custom wireless system. 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 detected and corrected. All of these 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 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.

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. The block diagram in figure 1 below provides an overview of the Quad-Copter's subsystems.

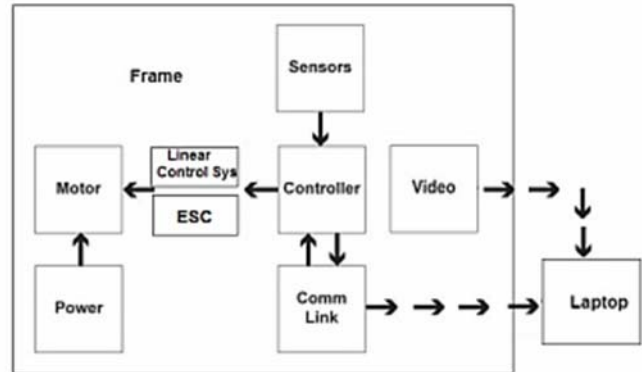


Fig. 1. Block diagram of Quad-Copter subsystems

II. SENSOR SUBSYSTEMS

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 subsystems can be organized into the following categories: flight stability sensors, proximity sensors, yaw or direction sensor, and the navigation sensor (GPS).

A. 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 are combined using a sensor fusion algorithm, which outputs an improved estimate of the angular state. The output of the sensor fusion algorithm is the input to the linear control system which adjusts the speed of each motor to maintain a level hover. The sensor fusion algorithm used is based on the Kalman filter but with an adaption that uses the gyroscope to monitor and correct for angular velocity about the Quad-Copter's axes.

B. Proximity Sensors

Proximity sensors will 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 will be ultrasonic range finders, specifically, the MaxSonar LV-EZ2s from MaxBotix. Both sensors are 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.

C. Yaw/Direction Sensor

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 two-axis magnetometer from Honeywell, 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 can be used in conjunction with the GPS module to autonomously plot movement to a GPS coordinate.

D. Navigation/Position Sensor (GPS)

A GPS module will be integrated into the design of the Quad-Copter which will be a central component of the autonomous mode of operation. The GPS system will allow 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 will be 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 has 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 will interface with the MCU using a USART connection. The GPS adapter module is connected to the main PCB with 15 cm cable with an EM406 6 pin connector.

III. POWER

Power has 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 is consumed by the motors, the best solution is to directly connect the lithium polymer (LiPo) batteries to the motors. Since the biggest concerns regarding the LiPo batteries are 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 are an excellent source in terms of mass, balance, and charge capacitance. The batteries will be directly connected to the ESC, which are rated at 30A per ESC. From preliminary testing, a 5 minute window at the maximum setting is allowed, which reinforces the minimum flight time. The main power source for the main board and the remaining peripheral components will be 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 is implemented at 2.98V by an LM317-ADJ. This will be 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.

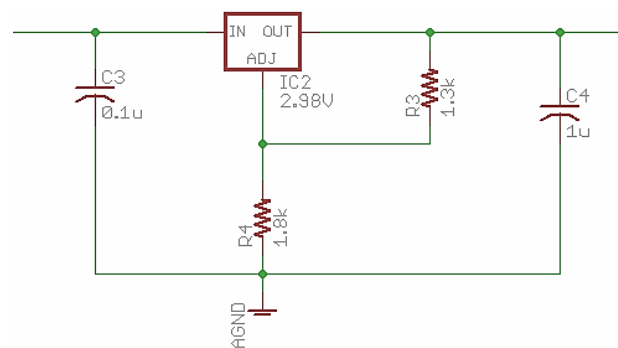


Fig. 2. LM317-ADJ setup for 2.98V

For the majority of the on-board components, there will be two sources used. The first is the LD1117V33, an LDO regulator fixed at 3.3V. This regulator is 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 will power the accelerometer and ultrasonic range finders to optimize analog signals sent to the ADC ports. Two separate versions of this will be used: one analog and one digital, to isolate noise. The second regulator is a dedicated regulator to the GPS, the digital compass, and the MCU, which is required to operate at 5V for optimized processing speed for the MCU. The LM7805 is 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 will always emit a voltage of 5V as the voltage high when transmitting data to all I2C and UART devices. For the digital compass, this is not an issue, since it 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 will 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 $0.7 \cdot VCC$ for the MCU, level logic converters were placed on board to three UART ports. The third UART port acts as an auxiliary port to the MCU, and allows 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.

IV. LINEAR CONTROL SYSTEM

The linear control system will be responsible for stabilizing the Quad-Copter. This will be the most crucial system of Quad-Copter as it will enable the Quad-Copter to fly stable. The linear control system will consist of a discrete PID controller that will be design in software with integer math. The PID will receive an output from the sensors and determine if the Quad-Copter needs to be stabilized or not. The design of the system is as follow. 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. Discrete equation (1) is shown below along with the K_i term (2) and the K_d term (3).

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

$$K_i = \frac{K_p T}{T_i} \quad (2)$$

$$k_d = \frac{K_p T_d}{T} \quad (3)$$

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

The first method will be called `initPID`. It will have as arguments the desired output, the measured output, K_p , K_i , K_d , 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, K_p , K_i , K_d , the summation of the errors, the maximum error allowed, and the maximum 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 be 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 K_d_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.

In order to tune the PID the Ziegler–Nichols method is used. The Ziegler–Nichols starts out by setting the K_i and K_d to zero. Then K_p is raised until the output oscillates, the gain that this happens at is called the ultimate gain, K_u . The period of the oscillation is called P_u . After that simple equations are used in order to find T_i , T_d , K_i , and K_d . Table 13 shows the calculations of the gains. The equations to calculate the gains are shown in table I.

TABLE I
ZIEGLER-NICHOLS METHOD EQUATIONS

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

V. MICROCONTROLLER

The group decided to choose a MCU for simplicity of use and due to it's 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 3.

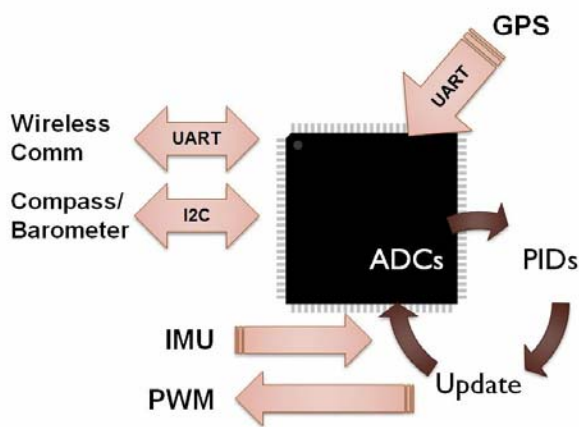


Fig. 3. Overview of MCU interfaces

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 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. A third UART will be placed as a means of observing data directly from the MCU, as well as an auxiliary port to use a UART connection.

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

The main software will be interrupt driven. The reason for this is to keep MCU running and not waiting for an event to happen. While the MCU is calculating the PID equations it can still receive data through UART and I2C.

VI. SOFTWARE

The software will be responsible for controlling all aspect of the Quad-Copter. These include the action-perception loop, polling of the sensors, and controlling movement. The code is written in the C programming language, using the AVR Studio IDE.

A. Action-Perception Loop

The action-perception loop is the main loop in the software. This loop will run continuously, acting on sensor perceived sensor data. The software needs to keep track of the time it takes to run the action-perception loop for sensor data calculations. A minimum of 50 Hz must be maintained in order to keep flight.

B. Polling of the Sensors

The sensors are polled using the internal ADC of the MCU. The ADC has a resolution of 10 bits. The sensors

are continuously polled at a rate of 250 KHz. This is done because the MCU can perform more accurate conversions at this speed.

C. Movement Control

The software for movement control consists of a sensor fusion algorithm, and a PID loop. The sensor fusion algorithm is responsible for taking the accelerometer and gyroscope sensor readings, and combining them into a better estimated angle. This algorithm takes the force projection vector of the accelerometer and relates it to the gyroscope instantaneous velocity. It does this by projecting the X or Y component of the force vector onto the Z plane. The gyroscope measures the speed of the angle between the projection and the Z axis. Through this the algorithm can produce a better estimate of the force vector, which can then be used to calculate the angle position of the Quad-Copter.

This estimated angle is then run through a PID loop. The PID will produce the adjustment of the motor needed to correct the position of the Quad-Copter.

VII. WIRELESS

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. There are many prepackaged ZigBee solutions to choose from. The most popular one is the XBee module. The XBee module that will be 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

For the purposes of this project, the minimum connections are all which are required. These pins would be the VCC, the ground, and the DIN (or the RX) pin, and the DOUT (or TX) pin. Because this will also be our means of manually controlling the quad-copter, a very fast

baud rate is required. Therefore, the baud rate chosen for the quad-copter is 38400, in accordance with the specifications set by the XBee module, as well as the frequency set by the control and status registers found in the MCU.

To have data sent and received for debugging purposes, a freeware software package is used, called Simple Terminal. Through a terminal-style interface, the XBee can self-test and receive data from another XBee module, allowing for an interface to the MCU wirelessly. Since the XBee has no means of directly connecting to a PC or laptop, a breakout USB board is used to create a connection via an FTDI USB interface chip. Output data consists of, but is not limited to: motor speeds, accelerometer and gyroscope PID values, compass values, range finder data, and GPS data. This data has been essential for testing PID values, as well as simple functions, such as the takeoff command.

VIII. FRAME

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 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 4

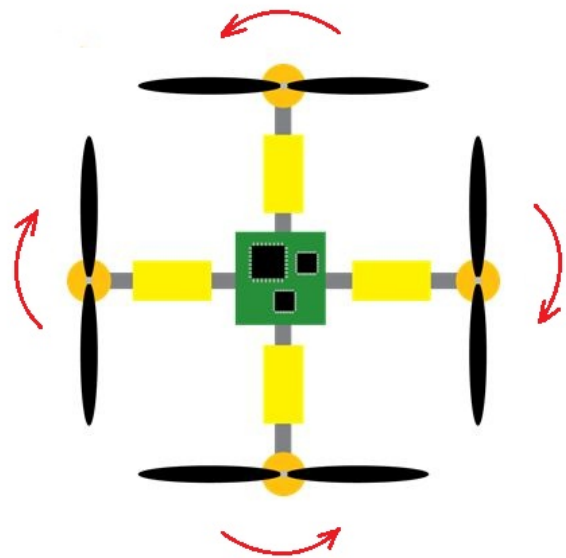


Fig. 4. Basic Quad-copter Design

A lightweight frame must be designed to support all the quad-copter subsystems. The frame will be either entirely aluminum or a combination of aluminum and carbon fiber tubing. Carbon fiber tubing has a slight strength to weight

ratio advantage over aluminum but is more expensive. 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 5.

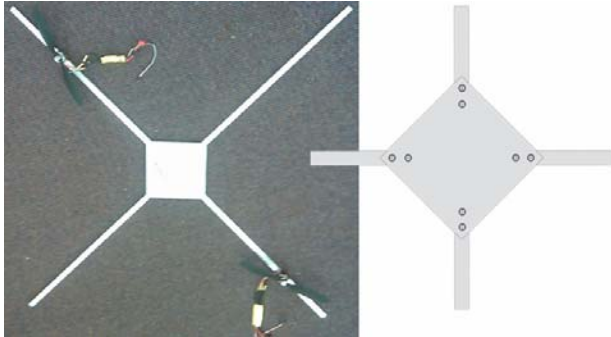


Fig. 5. Aluminum prototype frame with close-up of central plate

The prototype frame should be sufficient for testing the quad-copter's systems as they are developed. If the prototype frame works well enough, then it will simply be used as the final frame; which is in accordance with the goal of achieving functionality, rather than optimum performance. If improvements on the frame become necessary then they may include: a smaller 5 in. X 5 in. central plate to reduce weight, holes drilled in the central plate to further reduce its weight, or carbon fiber tubes instead of aluminum. Another area in which the prototype frame could be improved upon is that it may conduct excessive vibration. The aluminum struts used on the prototype frame are L beam shaped which, conducts more vibration than square shaped beams of a comparable material. Excessive vibrations can be particularly disruptive to the accelerometer and gyroscope. One method for addressing the potential vibration problem would be to use double sided vibration dampening tape to attach the main board to the frame, thus insulating the sensitive IMU components. A further step that could be taken to reduce vibration would be to use some of the vibration dampening material at the motor mount/frame junction to absorb excess vibration at its source, the motors.

The motors to be used will be the TowerPro 2410-09Y, which use an 11.1V source. These were chosen based on their mass of 55g and their speed/voltage constant of 840 rpm/V. To maximize thrust, the largest recommended propellers will be used, which is rated at 10"X4.5". The final version of the Quad-Copter is displayed below in figure 6. Notice the placement and orientation of the batteries are connected to a separate aluminum rod system, which offsets the center of mass. Statically, it's more effective for a hover state, since most lateral forces

are assumed to be zero. This design simplifies control of the roll and pitch in a hover state.



Fig. 6. Final frame

For flight, the motors will be set to 50% power, well within both calculated and observed tolerance, to insure flight. Tensor wire is used to keep the bottom structure stable to prevent a weight imbalance. Batteries will be held in place by electric tape and will be kept as close to bottom structure as possible.

IX. HARDWARE LAYOUT

When designing the PCB, there were several issues to consider in design, from part placement to power distribution to signal analysis to electromagnetic interference issues which effect internal and external communications of the main board. For these reasons, a simple design is required for tracing power and signal lines, distinguishing on-board components, and considering required solder mounted parts for optimized signals across each line.

The board was decided to be two-layered, due to its simplicity and cost. This allows all group members to be able to easily trace any one component to another. This also allows 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 are set as through hole for the sake of quickly setting up and testing a board with minimal risk to the other components. This has 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 allow 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 is immediately split to four separate sources, due to dropout effects in the voltage regulators. All on-board regulators use a TO-220 packaging due to the simplicity of the parts, as well as the minor difficulty to acquire said parts. An LED is directly attached to the batteries lines to determine if power is running through the on-board switch and to the main board.

Power is distributed based on the needs of the components. Since no ground planes are being used, the power lines for all digital regions will be re-enforced for a larger width of 16 mil width. This allows 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 is being distributed by a power distribution method known as the star method, where the voltage and the ground are radiated outward from a central point and evenly distributed to the MCU components. This allows 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 is also used for distributing a ground to the PWM ports and the XBee module. Power for the analog hemisphere is 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 is minimized in its occupied space, allowing for more real estate needed by the digital side. This creates an effect of the IMU components being off center by as much as an inch from the center of the plate. This effect will be minimal in terms of gyroscopic measurements, as well as feedback to be given from accelerometer. There are 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 are kept separate from the digital sources. The ADC pins all face the data lines from the analog components for simplicity. Due to the architecture of the chip, all of the ATMege2560's digital lines are placed away from the analog lines, which simplifies separated power and signal lines around the ADC. Power and ground lines are strategically placed to ensure no issues in crosstalk between the analog and digital lines, as well as to minimize and separate noise between the analog and digital hemispheres.

To interface and program the MCU, a 2X3 port is established near the chip as a means of communicating and reprogramming the on-board ATMege2560. This is 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 are located as close to the MCU as possible, allowing for the best quality signal to be sent to the ESCs. Each one has been labeled in accordance to compass directions: NORTH, SOUTH, EAST, and WEST. This act only as a reference point to each PWM, since during the design stage, no dedicated front was decided. The only on-board UART device is the 1mW XBee module. The design of the module uses smaller through-hole pin outs, at 2mm distance from each neighboring pin. The level logic converter is 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 is using an internal crystal to handle all clocks and timers. However, a space for a 2-pad crystal is available for using an external oscillator, with pin outs for two ceramic capacitors.

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

X. CONCLUSION

The Quad-Copter was an ambitious engineering project that challenged all members of the group in several different areas of learning. The fight stability algorithm in particular caused a development bottleneck as the group explored different options to optimize the system. A custom wireless communication system was painstakingly designed but, had to be abandoned due largely to technical difficulties in achieving a sufficiently high quality of solder work. Designing a main PCB became problematical when on occasion certain components had to be changed which necessitated modifications throughout the board. The difficulties, however, were an important aspect of the learning process.

ACKNOWLEDGEMENT

The authors wish to acknowledge the assistance and support of the UCF radio club and especially the assistance of Nathan Bodnar who assisted with solder work and PCB design. Also, Dr. Weeks who advised on the ground plane layout of the PCB, and Dr. Richie who was often available for advice and suggestions.

BIOGRAPHIES



David Malgoza will graduate from the University of Central Florida with a BSEE in December 2010. He is planning to enter the workforce upon graduation and to pursue

graduate studies in the field of power electronics. Dave's interests include classical and jazz music.



Engers F. Davance Mercedes will be graduating from the University of Central Florida with a BSEE in December 2010. He is planning to enter the workforce upon graduation and to

pursue graduate studies in the field of communication theory and radio frequency engineering. His interests include video games and cooking.



Stephen Smith will be graduating from the University of Central Florida with a BSCpE in December 2010. He is planning to enter the workforce

in the field of computer engineering upon graduation. His interests include reading and martial arts.

Joshua West will be graduating from the University of Central Florida with a BSCpE in December 2010. He is planning to enter the workforce in the field of embedded system engineer upon graduation. He currently interns at SAIC, and is hoping to have a career there.

REFERENCES

- [1] "ATMega640/1280/1281/2560/2561 Datasheet" AVR Solutions Web. 1 September 2010.
- [2] "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?](http://www.sparkfun.com/commerce/product_info.php?products_id=8664)
- [3] "Ultrasonic Range Finder - Maxbotix LV-EZ1" Sparkfun.com. Web 30 July 2010 http://www.sparkfun.com/commerce/product_info.php?products_id=639
- [4] "Compass Module - HMC6352". Sparkfun. Web 12 July 2010 http://www.sparkfun.com/commerce/product_info.php?products_id=7915
- [5] 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>
- [6] 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
- [7] Davance, 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.
- [8] "AVR221: Discrete PID Controller". AVR Solutions Web 1 May 2006 http://www.atmel.com/dyn/resources/prod_documents/doc2549.pdf

