SEARCUE

School of Engineering Science
Simon Fraser University
8888 University Drive
Burnaby, BC V5A 1S6

November 6, 2014

Dr. Andrew Rawicz
School Of Engineering Science
Simon Fraser University
Burnaby, British Columbia
V5A 1S6

Re: ENSC 440 Design Specifications for Search and Rescue Quadcopter

Dear Dr. Rawicz,

Enclosed in this document is our design specification for the Searcue system. The system uses an Unmanned Aerial Vehicle (UAV) with a camera mounted on it and is capable of both manual and autonomous flight. This system can be used in many applications, videography, geological surveys and security.

We aim to assist with search and rescue operations particularly paying attention to geographical areas that are not safe which ensures safety, saves time and money.The system provides birds eye view and operators are able to navigate a particular area sufficiently and communicate with the search and rescue teams.

This design specification document describes the functionalities of the proof-of-concept model. All hardware and software components of our product are examined to meet the standard requirements including functional requirements, safety considerations and test plan. We will refer to this document throughout the development phases of our product.

Searcue comprises of five well-varied and dedicated engineering students: Lekabari Nghana, Hesam Fatahi, Avi Gill, Gurjeet Matharu and Mehrdad Ahmari. Should you have any questions or comments on this proposal, please feel free to contact Gurjeet Matharu at 778-828-3254 or by email at gsm9@sfu.ca.

Sincerely,
Gurjeet Matharu
CEO
Searcue

Enclosed: Design Specification for Search and Rescue Quadcopter

SEARCUE

Design Specification for Search and Rescue System

Project Team: Lekabari Nghana
Hesam Fatahi
Avi Gill
Gurjeet Matharu
Mehrdad Ahmari

Contact Person: Gurjeet Matharu
gsm9@sfu.ca

Submitted To: Dr. Andrew Rawicz – ENSC 440W
Dr. Steve Whitmore – ENSC 305W
School of Engineering Science
Simon Fraser University

Issue Date: November 6th, 2014

Revision: 1.0

Abstract

British Columbia, in all of its beauty and terrains, can pose dangerous when it comes to rescue operations. The lives of individuals and the safety of the rescue team are very important. The time spent trying to figure out a situation before acting can be long and this is mainly due to environmental factors.

Searcue is a system designed to help improve search and rescue efforts. Following the functional specification document, we further designed the system to fit the functionality. Searcue enables both autonomous and manual flight using an UAV. The main components of the UAV system include an onboard microcontroller with sensors that help control flight and a GPS that enables sending information of a particular location to a server. With a camera mounted on the UAV, we are able to collect videos in a particular search area and also process them.

This document describes the design processes for the proof-of-concept model including all hardware and software designs and the communication amongst the parts.

Table of Contents

List of Figures

Glossary

UAV- Unmanned Aerial Vehicle, aircraft without human pilot

FPS – Frames Per Second

GPS - Global Positioning System

ESC - Electronic Speed Controllers

DOF - Degree of freedom

IMU - Inertial measuring unit

CSA - Canadian Standards Association

PID – Proportional Integral Derivative

BCSARA – British Columbia Research and Rescue Association

CW – Clockwise

CCW – Counter Clockwise

LiPo – Lithium Polymer

NiCad - Nickel-Cadmium or

Li-Ion - Lithium-Ion

NiMH - Nickel-Metal Hybrid

1. Introduction

The main purpose of the project is to assist in search and rescue operations. In BC where there are lots of mountainous areas with different terrains, rescue operations can be delayed. [1] There are areas that pose dangerous for humans and may lead to ineffective searches. In order to reduce human limitations, we came up with the idea of a search and rescue system that uses a UAV with a camera mounted on it for both manual and autonomous flight. The camera is used to capture images using digital image processing. The operator can specify a particular area to search in autonomous flight mode and can also switch to manual mode, where the UAV can be controlled to fly down for a better view of the situation. Using this information the search and rescue team can respond to different terrains with the necessary equipment needed for the rescue. As a proof-of-concept model, we would design the system to find an object in a field using digital image processing and send the Global Positioning System (GPS) location to the operator.

1.1 Scope

This document includes information regarding the overall design, as well as detailing the different components of the project. The system will be able to operate like a standard quadcopter, capable of autonomous and manual flight. The context will include the hardware, software and the interactions between the different modules. This document will serve as a guide throughout the development phases.

1.2 Intended Audience

The intended audience for this design document includes all members of Searcue and any other faculty or students who wish to use it as an educational resource. It is strongly recommended that anyone using the system go through full training in its use by a member of our team familiar with the system.

2. System Requirements

2.1 System Overview

The Searcue system must be able to efficiently assist the search and rescue team(s) that it has been deployed with. This requires the Searcue system, consisting of an UAV, manual controller, image acquisition device, and image-processing server. These integrated parts will perform as autonomously as possible, as demonstrated in figure 1.

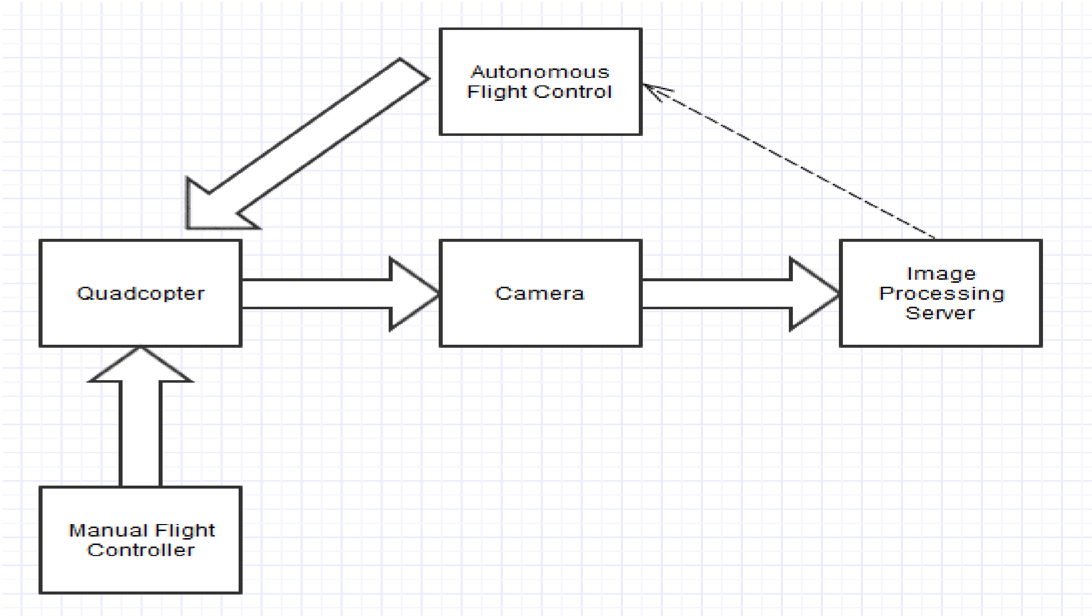

Figure 1: The Searcue system consisting of its fundamental components and their relations.

This requires the system to take into consideration various conditions, when selecting each of the components that make up the UAV. Due to the fact that Searcue will be deployed in various terrains, the system will need to communicate with each of the parts effectively over an adequate range. Furthermore, the system's main purpose is locating lost objects or person(s), which requires the use of global positioning system (GPS). Alongside the use of GPS, the system must be able to autonomously survey an area. These are the main design specifications driving the selection of the various components within the Searcue system.

3. General Specifications

3.1 Camera

The camera needs to be light in weight and able to record video in high frames per second (FPS) due to the continuous movement of the UAV. To meet these requirements we researched many different cameras, which ultimately led to deciding on the GoPro Hero3+ Camera. This camera is very compact in size allowing us to mount it on the UAV with ease. The GoPro weighs only 73 grams not impacting the weight of UAV by a large factor. [2] The camera can record at up to 1080p HD at 60 fps meaning it can record even while in movement without too much video degradation.

3.2 Quadcopter Principle

Lots of research went into deciding what parts to use and integrating the UAV system. We chose to go with a quadcopter rather than other multi-rotors, because it is a mechanically simple system which means lower cost, better robustness and it is electronically stable. Quadcopters are also able stay in the air longer if there's enough battery power. Weight is also an important factor and it depends on the size of the UAV. For long flight times a light platform is the best solution. The UAV is based on two main components the microcontroller and structure seen in figure 2.

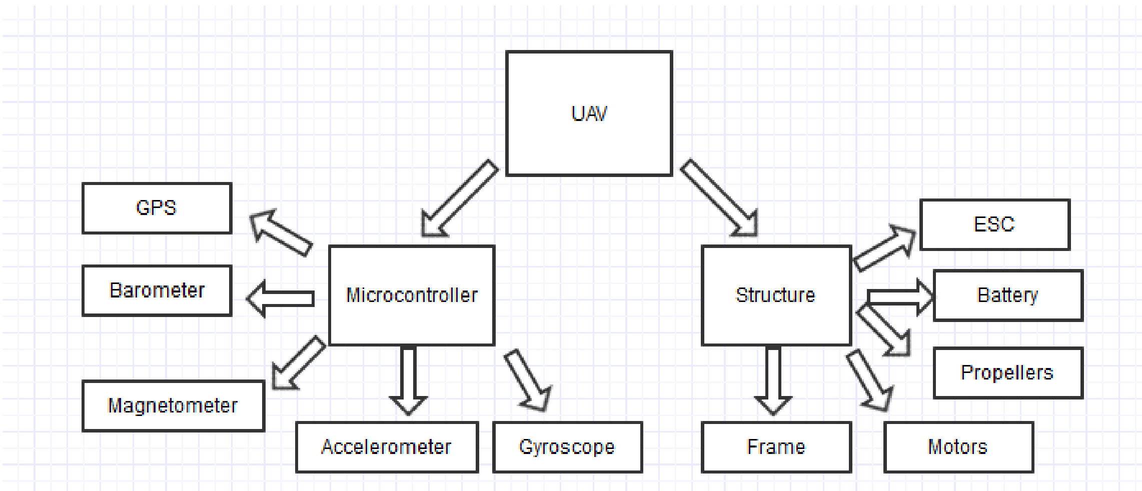

Figure 2: The UAV with its microcontroller and structural components.

4. Structural Specifications

4.1 Frame

The frame is very important because it holds the entire system and has to withstand falls and system vibrations. We chose the X-shaped frames instead of plus or H frames to allow enough distance between the propellers and for better stability during flight. Furthermore, the arms of the X-shaped are visually minimized within the camera footage. Overall the X-shape frame allows for better image capturing and stability. We chose to use the F330 Glass Fiber quadcopter frame as it is an X-shaped frame, light in weight, sturdy and cost effective.

4.2 Motors

In order to choose a motor, we took into account the weight of the entire system and ensured that the selected motors provide enough thrust for our system. [3] We also considered efficiency, the higher the efficiency, the better the motors. Less efficient motors waste power and result in lower thrust. For this project, we decided on 3-phase 2212 MultiStar motors with 920 RPM/V. The motors are able to with stand a maximum current of 20A and to draw 130W. We chose these motors because they were a good compromise between efficiency, power and cost.

The way the quadcopter maneuvers is through the yaw, roll, throttle, and pitch as shown in figure 3.

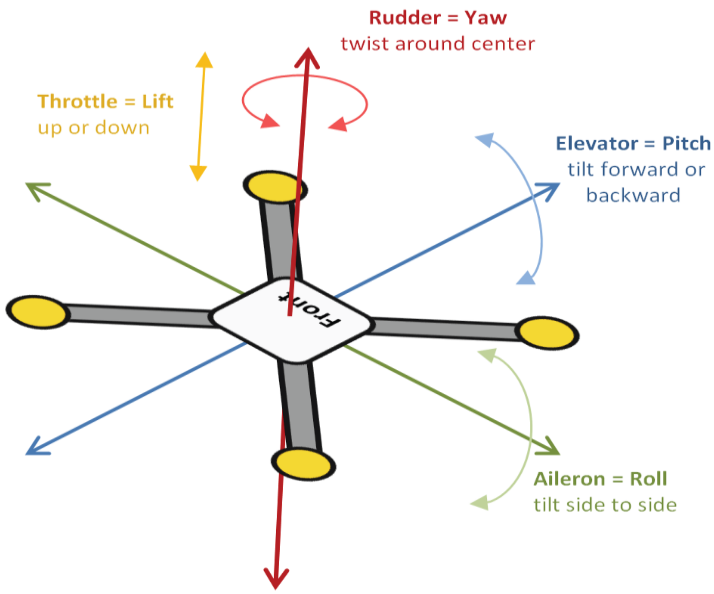

Figure 3: Angles of rotation of quadcopter. [4]

The manual transmitter controls these aspects of the quadcopter. Furthermore, the system needs to be stable while it moves through the air, this is accomplished by utilizing motors that spin in both counterclockwise and clockwise directions, minimizing momentum allowing for the throttle to increase (by the increase of motors). This principal is demonstrated in figure 4.

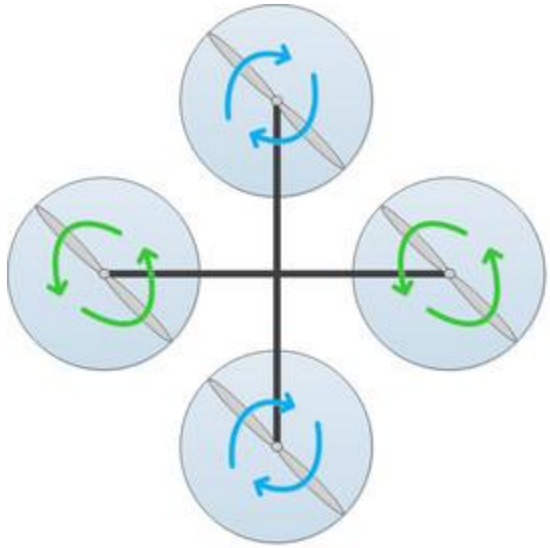

Figure 4: The rotation of the corresponding opposite motors. [5]

4.3 Electronic Speed Controllers (ESCs)

Normally you would need an ESC for each individual motor but this isn't the case using the QBrain ESC which is 4 ESCs into one all-in-one unit. It is able to operate on 2~4S li-poly or 5~12 nickel cells make this a very versatile ESC [6]. It also reduces Electromotive force (EMF) and is more reliable than using individual ESC's [7]. The ESC is a pulse-width modulation controller (PWM), applying full voltage to the motor, turning it on and off rapidly. By varying the ratio of on-time to off-time, the speed control varies the average voltage that the motor sees. The QBrain effectively has 4 ESCs in one unit keeping wires tidy and eliminating the need for a power distribution plate.

4.4 Propellers

Furthermore, we chose a 5mm domed hex nut that supplies 2 clockwise and 2 counter clockwise rotations. The propeller is lower pitch and smaller sized propellers reduce wobbling when hovering thus improving stability. With well-balanced motors and propellers, we are able to improve battery lifetime.

4.5 Lithium Polymer (LiPo) Battery

A battery with a high energy storage/weight ratio is required and LiPo battery fulfills this requirement. LiPo batteries have the ability of being charged many times without losing capacity (generally up to 400 times), making them a better option over conventional rechargeable batteries such as Nickel-Cadmium (NiCad), Lithium-Ion (Li-Ion) or Nickel-Metal Hybrid

(NiMH) batteries [8]. We chose to use the Zippy 4000mAh as it is a LiPo battery producing 4000mAh, which is sufficient for our quadcopter. [9]

5. Microcontroller Specifications

5.1 Arduino Mega 2560

To get the quadcopter to fly we used an Arduino Mega 2560 as the microcontroller. A microcontroller is needed since we need to transmit controls from a transmitter to the quadcopter, as well as gathering data from the sensors. This microcontroller was used as opposed to the Arduino Uno or others due to the number of inputs and outputs we needed. Additionally, the pre assembled flight controller currently available on the market does not have a magnetometer, which is important for our application. The magnetometer is important to orient the quadcopter and sweep/search areas. Furthermore, the chosen sensors for our flight controller are very robust and effective. The technical specifications of the Arduino Mega 2560 are given in table 1 and the Arduino Mega 2560 is shown in figure 5.

Microcontroller	**ATmega2560**
Operating Voltage	5V
Input Voltage(recommended)	7-12V
Input Voltage(limits)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O pin	40mA
DC Current for 3.3V pin	50mA
Flash Memory	256 KB of which 8KB used by bootloader
SRAM	8KB
EEPROM	4KB
Clock Speed	16MHz

Table 1: Technical specifications of the Arduino Mega 2560. [10]

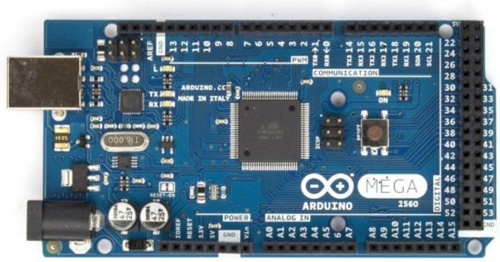

Figure 5: The Arduino Mega 2560. [10]

We then have an AeroQuad shield to mount onto the Arduino Mega to simplify the connections. On the AeroQuad shield we have a sensor stick, a level converter, a barometer, as well as inputs for both the motors and receivers. Figure 6 shows a schematic for the AeroQuad shield.

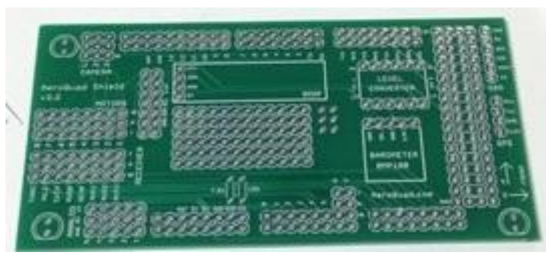

Figure 6: The AeroQuad Shield mounted on the Arduino Mega. [11]

5.2 Accelerometer, Magnetometer and Gyroscope

The sensor stick chosen consists of an ADXL345 accelerometer, a HMC5883L magnetometer, and the ITG-3200 gyroscope. [12] The 3 axis accelerometer is needed to obtain the orientation of the quadcopter (pitch and roll) and the accelerometer does this by giving the value of the gravitational field, which always points towards the earth. A magnetometer is used because it gives direction readings in the 3 axes with reference to the current orientation, essentially behaving as a compass. Finally, a gyroscope is used since it gives the angular rate around the 3 axes in degrees per second. Using all three of these sensors the complete orientation and position of the quadcopter can be determined. The level converter gives the capability to use these sensors as it has a 3.3V or 5V option.

5.3 Barometer

The barometer we used was the BMP-180 which is needed since pressure changes with altitude, thus giving us the ability to use this as an altimeter. [13] Using this we are able to hold the quadcopter altitude during flight giving us more flight control. Furthermore, when the Searcue system recognizes a positive match for the object or person it is searching for, it must hover in place. This is accomplished by a balance of forces, generated by the motors and gravity, as shown in figure 7.

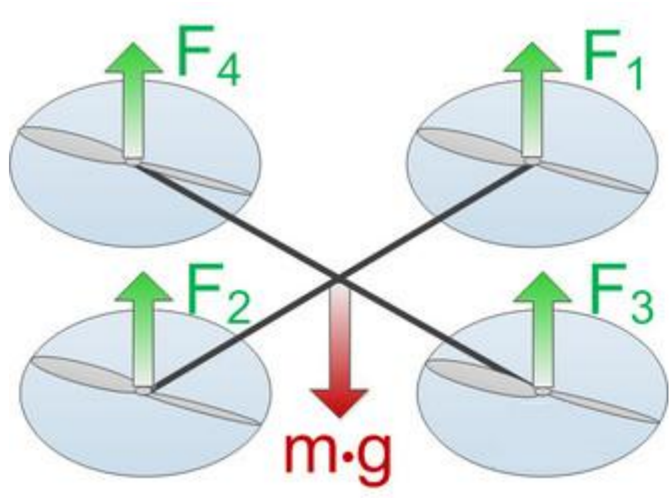

Figure 7: Balance of forces due to the motors and gravity. [5]

The sum of the forces F1 through F2 must be equal to the gravitational force (mass multiplied by gravity). This balance of forces is initiated at any desired barometer value.

5.4 GPS

The GPS unit is required to navigate and locate the individual or objects of interest. The GPS shield and GPS unit allows for a simple connection to the arduino mega. This is a key component of the Searcue system and is important due to the fact that it is easily integrated into the quadcopter.

5.5 Flight Controller

To decide the flight controller, we considered only open source flight controllers. This limited our choices to ArduPilot, OpenPilot and AeroQuad. Table 2 determines the various characteristics of these flight controllers.

	ArduPilot	OpenPilot	AeroQuad
Cost(in Canadian Dollars)	$250	$200	$170
Sensor Capabilities	9 DOF	9 DOF	9 DOF
Control Configuration	PI + P	PI + PI	PID

Table 2: Comparison of open source flight controllers. [14][15]

From the table it is quite evident that the AeroQuad flight controller is the most cost efficient. The AeroQuad also has the best control ability using a PID controller giving the UAV the most stability.

6. Image Processing Server

Utilizing an open source transmitter module and a HawkEye transmitter and receiver pair, we are able to send a live video feed to our image processing server. This server will take care of all the image processing algorithms and will be hosted on a laptop. Once the server recognizes a positive image match, it will notify the user and send an interrupt to the quadcopter halting its current search. The HawkEye transmitter was chosen due to its open source software support, namely openLRSng and its ability to readily integrate with our microcontroller.

7. Communication Requirements

We are using Turnigy 9X- 9 channel transmitter with Module and an 8 channel receiver. The receiver uses Pulse Position Modulation (PPM) encoder and the transmitter uses Pulse Code Modulation (PCM). The PCM codes the FM signal digitally and sends it to the receiver. The receiver then decodes the signal to utilize it. Since noise is not a recognized code the receiver ignores it and by doing so reduces noise and improves performance. In order to operate the quadcopter we needed a minimum of six channels, three for translational movements namely heaving, swaying, and surging, and another three for rotational movement such as pitching, yawing, and rolling.

During the design process we decided to purchase a 9 channel transmitter, because the price difference was not significant and the extra channels give us the freedom for expansion in future projects. The transmitter is a Mode II transmitter, which means it uses the right hand stick to control rolling and pitching and the left hand stick to control yawing and throttling. Another advantage of the chosen transmitter is that it is programmable, which make it possible for us to modify the way it reads our command and deliver it to the receiver. Another decisive factor in choosing Turnigy over other products such as Spektrum DX61 was the price. We decided to have a receiver from the same brand to eliminate any compatibility issues.

8. Test Plan

The test plan is in two phases, a test plan covering the UAV and the other for the image processing server.

8.1 UAV Test Plan

8.1.1 Microcontroller Testing:

- Supply a stable voltage to the microcontroller via USB and check the arduino using the blink tutorial[16]
- Confirm the required voltages are at the right pins (0V, 3.3V and 5V) using a digital multi meter thus checking the voltage level converter
- Confirm the connection between the Arduino Mega and the shield is working correctly and data is being received from the sensors
- Check that the sensor stick consisting of the accelerometer, magnetometer and gyroscope is measuring values precisely
- Check that the barometric pressure sensor is accurately measuring altitude
- Verify that the GPS is obtaining proper co-ordinates

8.1.2 Structural Testing:

- Test the ESC and motors via the following procedure:

    1. Connect ESC to motors via a throttle hub (an adapter that came with the ESC used for the initialization of the motors)
    2. Connect the throttle hub to channel one of the receiver
    3. Pair the transmitter and receiver
    4. Attach the LiPo battery to the ESC
    5. Test the maximum throttle of the motors from the transmitter controller
    6. Calibrate motors and ESC

- Mount the ESC, LiPo battery and motors onto the frame and check for vibrations and the sturdiness of the frame

8.1.3 UAV Integration

- Mount the assembled microcontroller (arduino, shield, GPS and sensors) onto the semi completed UAV frame (with motor, battery, propellers and ESC) giving us the completed UAV frame
- Turn the quadcopter on and test for stable flight

8.2 Image Processing Server Test Plan

- Verify and debug the code and algorithms for the image processing server
- Connect a camera to and allow for a live feed via a wired connection and ensure the image processing server is recognizing any objects or persons adequately
- Change the camera connection to a wireless medium, which will be a Hawkeye transmitter/receiver pair.

8.3 Searcue System Integration Test Plan

- Connect the camera to the completed UAV and test video feed during flight
- Final test will be to locate a colored object such as a ball within an unobstructed field autonomously and to obtain GPS coordinates of the ball
- Be able to switch from autonomous to manual mode after object of interest is found

9. References

[1] BCSARA (2014). Rescue Techniques. Retrieved from http://www.bcsara.com/sar-groups/rescue-techniques/

[2] CNET (2014). GoPro Hero3+ (Black Edition) Specifications. Retrieved from http://www.cnet.com/products/gopro-hero3-plus-black-edition/specs/

[3] Oscar. (2014, October 3). How to choose Motor and Propeller for Quadcopter and Multicopter. Retrieved from http://blog.oscarliang.net/how-to-choose-motor-and-propeller-for-quadcopter/

[4] ELEV-8 Quadcopter Kit (#80000) Information and Assembly Guide. Vol 1.1, pp.16 of 26. Retrieved from http://www.parallax.com/sites/default/files/downloads/80000-ELEV-8-Quadcopter-Info-Assembly-Guide-v1.1_0.pdf

[5] Qdrone A Zinq-Based Quadcopter (2012, January 22) Flight Phsyics. Retrieved from http://qdrone.jimdo.com/project/flight-physics/

[6] Hobby King (2010). Q Brain 4 x 20A Brushless Quadcopter ESC 2-4S 3A SBEC. Retrieved from http://www.hobbyking.com/hobbyking/store/__36674__q_brain_4_x_20a_brushless_quadcopter_esc_2_4s_3a_sbec.html

[7] 3D Robotics. (2014). Advanced Multicopter Design. Retrieved from http://copter.ardupilot.com/wiki/advanced-multicopter-design/

[8] Moyano Cano, Javier (2012, August 29). Quadrotor UAV for wind profile characterization. Retrieved from http://e-archivo.uc3m.es/bitstream/handle/10016/18105/PFC_Javier_Moyano_Cano.pdf?sequence=1

[9] Admin (2013, June 24). How to Safely Handle LiPo Batteries. Retrieved from http://www.flyingcameras.ca/blog/how-to-safely-handle-lipo-batteries/

[10] Arduino (2014). Arduino Mega 2560. Retrieved from http://arduino.cc/en/Main/arduinoBoardMega2560

[11] AeroQuad (2014). AeroQuad Shield v2.2. Retrieved from http://www.aeroquadstore.com/AeroQuad_Shield_v2_2_p/aq2-002.htm

[12] AeroQuad (2014). 9 Degrees of Freedom – Sensor Stick. Retrieved from http://www.aeroquadstore.com/9_Degrees_of_Freedom_Sensor_Stick_p/sen-10724.htm

[13] RP Electronics (2014). Barometer/Thermometer/Altimeter – BMP180. Retrieved from http://www.rpelectronics.com/af-1603-barometer-thermometer-altimeter-bmp180.html

[14] OpenPilot (2014). Revolution Hardware Kit. Retrieved from http://store.openpilot.org/home/21-revolution-hardware-kit.html

[15] 3DR (2013). APM 2.6 Set. Retrieved from http://store.3drobotics.com/products/apm-2-6-kit-1

[16] Arduino (2014). Blink. Retrieved from http://arduino.cc/en/tutorial/blink

