Multiple UAV Coordination

Final Project Report

By:

Ethan Hoerr, Dakota Mahan, and Alex Vallejo

BRADLEY UNIVERSITY

Department of Electrical and Computer Engineering

Advisor:

Dr. Joseph Driscoll

April 16, 2015

Abstract

This report will discuss the results of the Multiple Unmanned Aerial Vehicle (UAV) Coordination senior design project, in which multiple quadcopters will autonomously navigate a defined region, capture images of the region below, and merge the photos into a single image. The Statement of Work section will provide system objectives and metrics, as well as nonfunctional and functional requirements, an overview of the system design, and an economic analysis of the project. Next, the Design Testing and Validation section will provide details regarding the procedures used to verify the system, the results of which shall be analyzed in the Conclusions section.

Table of Contents

I. INTRODUCTION AND OVERVIEW

A. Problem Background

In present society, there has been a global fixation with drones and unmanned aerial vehicles (UAVs). The applications of these machines range from military, research, commercial, and entertainment purposes. The project described in this document seeks to use multiple UAVs to accomplish the task of generating an overhead image map of the Bradley Alumni Quad. Using autonomous control, the project ideally will require minimal user interaction.

Prior research inspired the design process of this project. In [1], an overview is given for the basis of a "swarming" or "flocking" coordination algorithm. In [2], a leader-follower configuration is utilized for controlling multiple UAVs to complete a task. In [3], the advantages of using multiple UAVs parallel the scale of the task to be completed.

From [1], key objectives for a "swarming" or "flocking" formation scheme are given. This includes a method for maintaining multiple UAVs in formation, as well as a system for group obstacle avoidance. The elegance with flocking is that high-level results can be achieved using a relatively low-level control system. Adding additional units to the flock or swarm is also relatively simple, as the group dynamics are not defined by a master controller, but rather by the sum of the actions of each individual unit.

In [2], the effectiveness of a leader-follower configuration in the use of multiple UAVs is researched. The task to be completed is the aerial visual tracking of a moving ground target. In the project described in this proposal, this leader-follower configuration is used in conjunction with the "swarming" configuration described in the preceding paragraph. This goes to show that not one coordination scheme is necessarily superior to another, but that choosing the right scheme requires a broad view of the task at hand.

[3] demonstrates the benefits of using multiple UAVs to accomplish a task. In this particular assignment, multiple fixed-wing drones navigate a region of airspace, measuring vertical wind speed of discretized regions of air space. Using the data captured during the mission, drones can reroute to take advantage of upward drafts that will prolong flight time without expending onboard power. The immediate benefit of using multiple drones for this task is that more air space can be measured simultaneously. An additional benefit from using more than one drone is that the group as a whole can benefit from one another's data, which can then increase the chances of a successful mission.

From [1]-[3], the efficiency and reliability of drone tasks can be greatly improved by introducing additional UAVs into the equation. While the topics discussed in [2]-[3] suggest different forms of added efficiency from the use of UAVs, the underlying theme is that, if implemented properly, the use of multiple autonomous UAVs will offer some advantage in accomplishing tasks.

B. Problem Statement

This project seeks to coordinate multiple unmanned quadcopters in parallel, equipped with cameras and various sensors to navigate, survey, and photograph a plot of land over many segments efficiently, in order to generate a full area map. The quadcopters must work together to map an area in the most time-efficient manner, avoiding unnecessary duplicate work. Individual images collected from the quadcopters will be combined to form a complete map. The quadcopters must also avoid collision with its surroundings.

C. Constraints of the Solution

The Multiple UAV Coordination project is limited by several constraints:

- Follower drone battery life: 7 minutes flight time
- Leader AR Drone battery life: 18 minutes flight time
- Hak909 drone payload: 251 grams
- AR Drone wireless communication cutoff distance: 60.96 meters
- $1000 Budget
- Execution of aerial photography task must be at a distance above ground level between 3 meters to avoid human height, but less than 121.92 meters (latter requirement as per FAA Advisory Circular 91-57)

II. STATEMENT OF WORK

A. Nonfunctional Requirements

There are several nonfunctional requirements the system must observe. The system is required to create an overhead image map of the Bradley Alumni Quad. The aerial images from the flight must be merged into one composite image after the flight. Finally, the UAVs should protect themselves and the environment from damage.

B. Functional Requirements

The system will adhere to a set of functional requirements as well. The UAVs must be equipped with manual override control; the UAVs must work together to accomplish the task; the system must make use of at least three UAVs; the system must avoid collision; and the system must merge all photos into an aggregate map of the region.

C. Design Overview

1) System Block Diagram

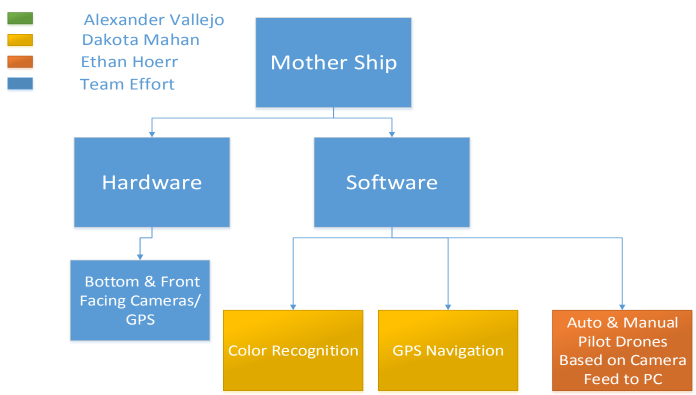

Figure 2 - AR Drone Mothership Block Diagram

Position data is gathered through the AR Drone Mothership box through the use of the on-board magnetometer and additional GPS sensor module. The accuracy is ~ 8 meters. Using the magnetometer to set a direction, we can use the GPS to follow waypoints pre-defined in the desired region. For detecting potential obstacles, by using Laplacian and bilateral filtering, we are able to detect edges in the image, while reducing unwanted edges. To detect the follower drones using the downward-facing camera, subtracting 1/3rd of the unwanted colors from the color channel we want, thresholding it, and then eroding it to reduce noise, we are able to find the center of mass of the quadcopters we desire.

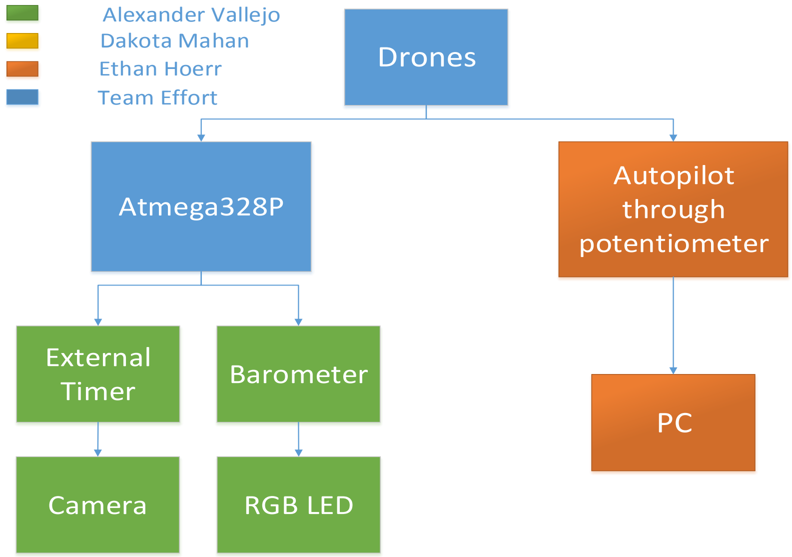

Figure 3 - Hak909 Follower Drone Block Diagram

The drone subsystem is shown in Fig. 3. An ATmega328P microcontroller-based circuit will be responsible for triggering a timed-interval camera affixed to the bottom of each Hak909 follower quadcopter UAV. The microcontroller will also gather altitude data from a pressure sensor and update 2 status LEDs based on whether the UAV is flying too high, too low, or within a desired operating range. The Hak909 follower UAVs will be controlled by a digital potentiometer circuit, which is run with an ATmega328P microcontroller. The digital potentiometers are replacing the existing handset controller joysticks for each Hak909 follower UAV. The control program receives data over a serial port to update the digital potentiometers status to reflect changes in throttle, yaw, pitch, and roll.

2) Hak909 Digital Control Subsystem

Figure 4 - Hak909 Quadcopters

Our system was designed to detect the color of each quadcopter or Unmanned Aerial Vehicle (UAV) to allow for autonomous follower drone control. Therefore, these Hak909 quadcopters were used and painted a Cyan, Magenta, and Yellow for image detection.

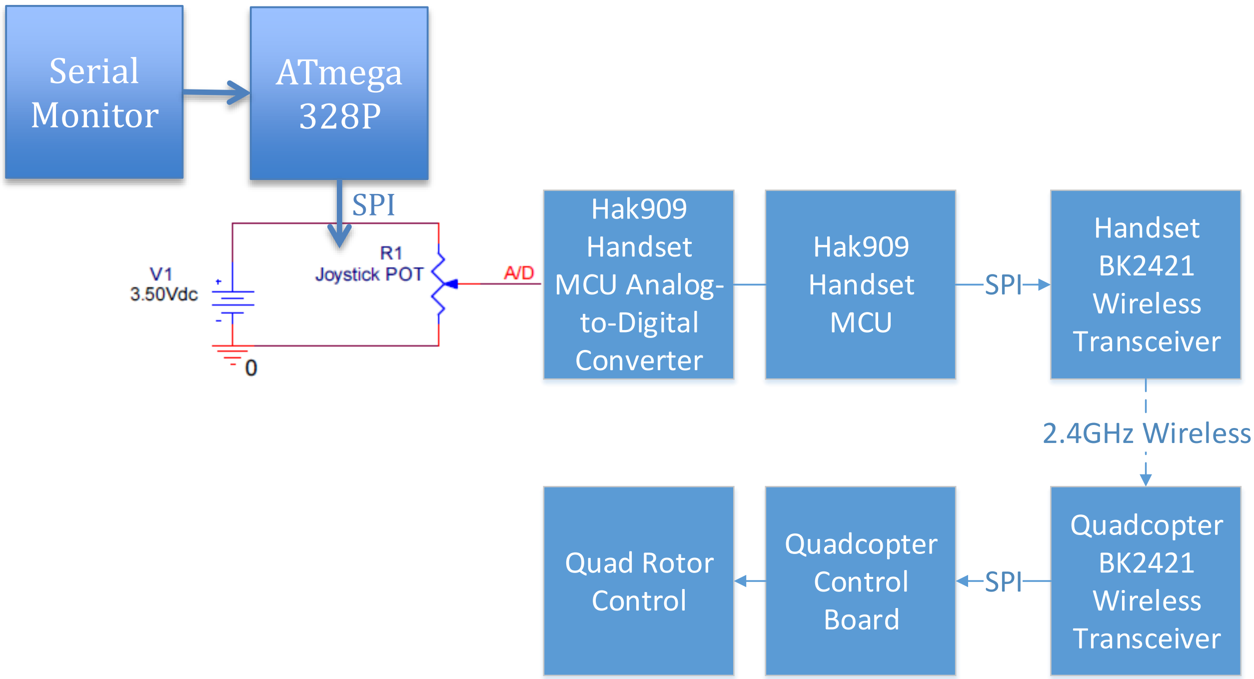

Figure 5 - Digital Control Subsystem Flowchart

Figure 5 shows the subsystem flowchart for the Hak909 digital control system. A software serial monitor scans for user-inputted control parameters formatted as four separate bytes delimited by commas, containing throttle, pitch, yaw, and roll data parameters. The ATmega328P program then extracts these control values from the string and writes the values to each of the four digital potentiometers. Each joystick is replaced by the digital potentiometer circuit, which is a Microchip MCP42010 10 $k\Omega$ dual digital potentiometer in parallel with a 10 $k\Omega$, the terminals of which were wired in place of the existing joysticks on the Hak909 quadcopter controller.

Once the four control digital potentiometers update their wiper position, the corresponding control voltage is read from the wiper to the analog-to-digital converter on the Hak909 handset microcontroller. Control data is then transmitted from the handset control board to the Hak909 quadcopter through 2.4GHz wireless transceivers. Upon receiving control signals, the Hak909 quadcopter will then update rotor speed as appropriate.

3) AR Drone Subsystem

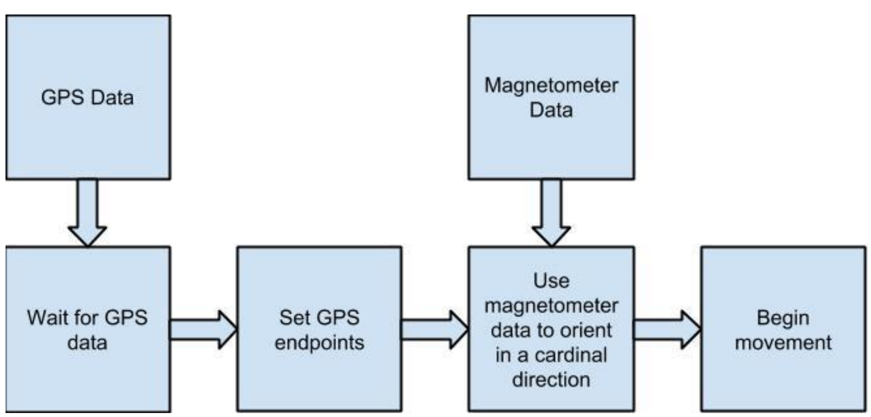

Figure 6 – GPS and Magnetometer Detection Flowchart

From Fig. 6, using the GPS data and magnetometer together allows the project to find it's heading to allow it to move towards certain GPS locations. Without the magnetometer readings, the project would not be able to head towards the GPS heading because the quadcopter does not know where north is, and would move off course attempting to fly towards the designated point. The designated point is 78 meters north and 78 meters west of the initial GPS value.

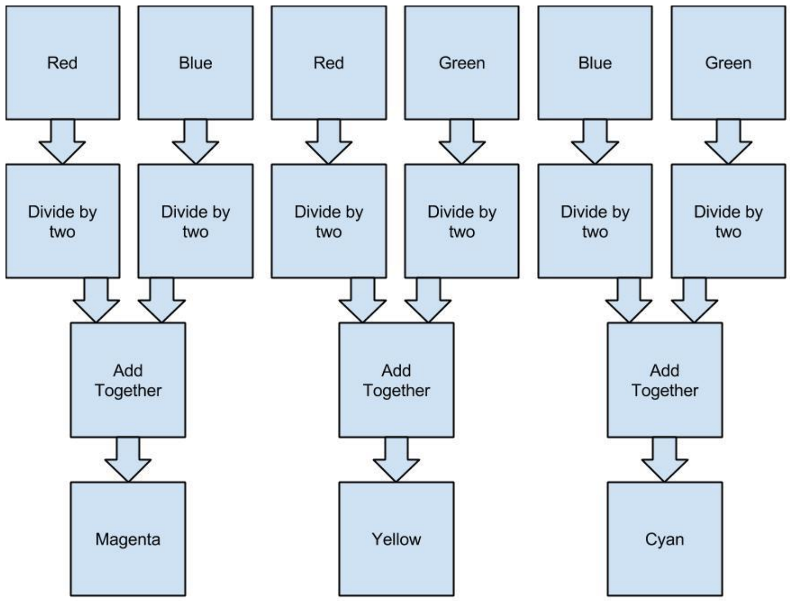

Figure 7 – Conversion to CYM

Using the above diagram, conversion to cyan, yellow, and magenta allows a range of 0-254, which is sufficient for our project.

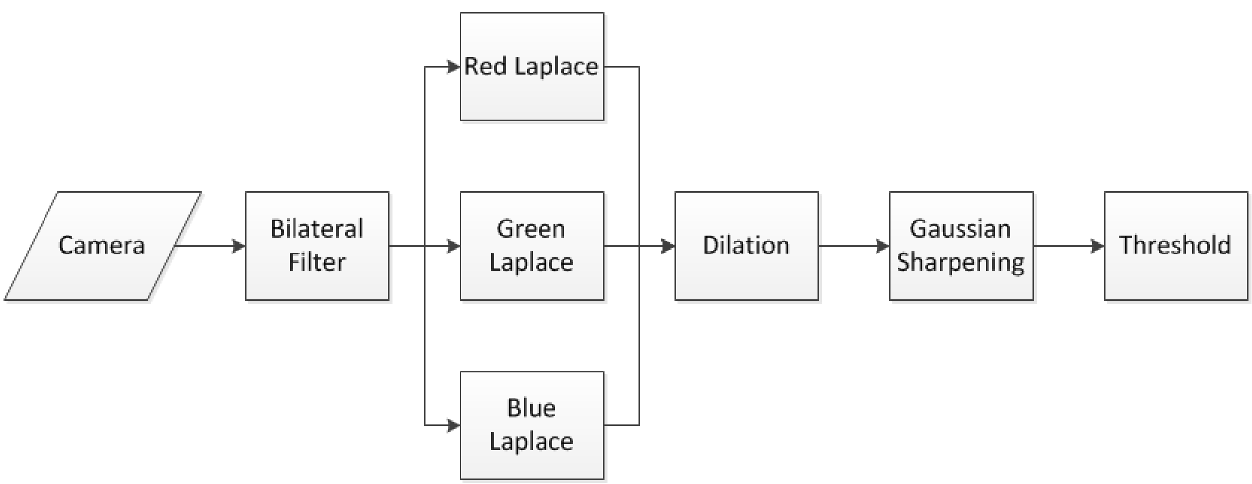

Figure 8 – Bilateral, Laplacian Filter Flowchart

The Bilateral filter and the Laplace filter in the diagram above are the most important parts of this process. The Bilateral filter removes noise and keeps edges intact, while the Laplacian filter gives the edges of the image. The dilation is used to increase the size of the edges in our image. This introduces noise, however, so we then use Gaussian sharpening to remove the noise and get a clearer view of the image. We then threshold this image to just keep the edges.

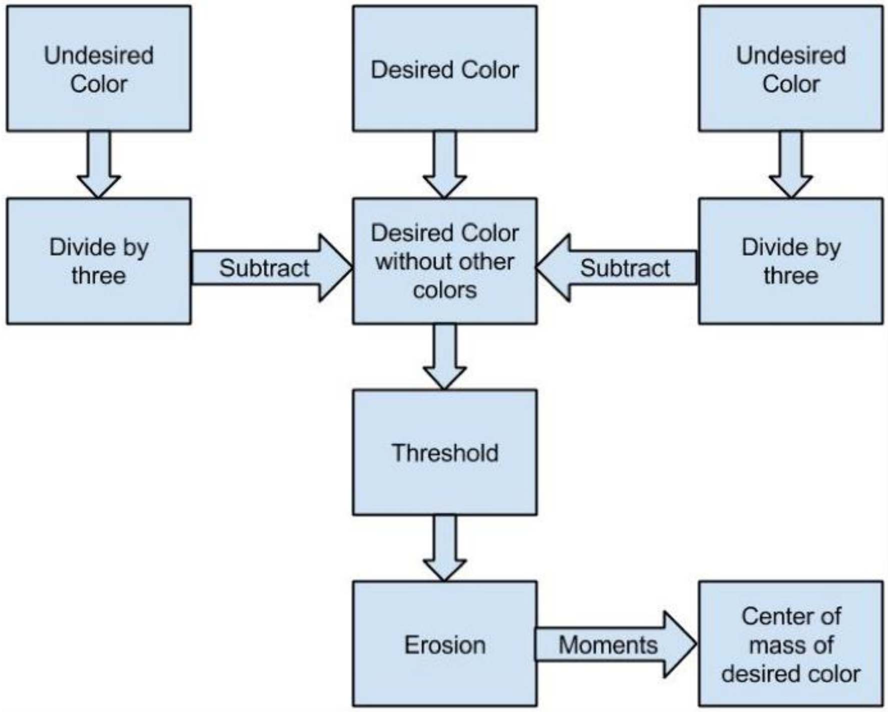

Figure 9 – Erosion Diagram

We use division by three to reduce the desired image. This is used because we want to check to see if it is a combination of color channels instead of a desired color. The threshold is used to determine if it is the color we desire, and the erosion is used to remove any noise in the resulting threshold value. Finally, we use a moment generating function to find the center of mass of the resulting image.

4) Altitude Sensing Subsystem

During the project there was different ideas that were conceived and thought of; we needed a way to be able to detect and communicate the altitude to the different subsystems. A Barometer is the initial thought that refers to being able to sense the pressure in the air, then having the ability to detect the altitude [8].

Figure 10 - Bosch BMP 180 Barometer Pressure Sensor

In order to be able to control the movements of the follower drones, the Hak 909 Quadcopters, we had to implement the directions of X,Y,Z. The operation of X and Y would be controlled using the image processing, while being able to correct for the Z direction [7]. Therefore, the Bosch BMP 180 Barometer in Fig. 10 was added for the spatial correction.

5) Autonomous Camera Subsystem

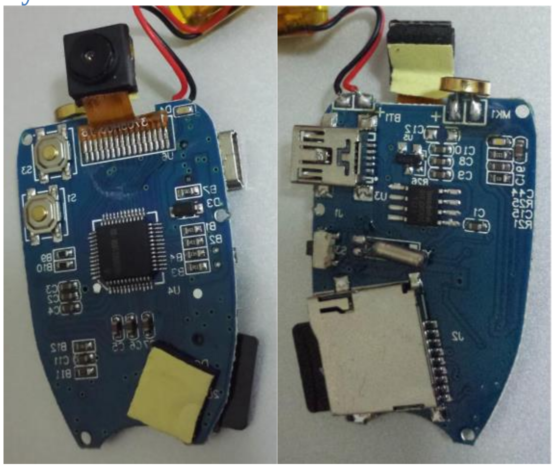

Figure 11 – 808 Pinhole Camera #25

In order to be able to parallel map the ability to take pictures autonomously which this camera if setup correctly using the button configuration in Fig. 11, we use to take pictures autonomously [9]. For future research, there is the capability to power on, video record, and power off autonomously if used properly.

D. Division of Labor

Dakota Mahan is responsible for the detection of the drones, object detection, and GPS Navigation.

Alex Vallejo is responsible for researching/Order Quadcopters (Hak909), Camera (808 Pinhole Camera), Barometer (BMP180), Microcontroller (Atmega328P); creating Circuit for 808 Camera, ATmega328P, BMP180, and Status LED; creating and implementing Perfboard of Circuit/Hardware Setup (Paint Quadcopters, Altitude Detection, Autonomous Camera); auto-batching images with timestamp, and stitching them together.

Ethan Hoerr was responsible for the development of the digital control system for the Hak909 quadcopters. Work involved in this process includes designing, testing, and verifying an appropriate digital control scheme for the Hak909 quadcopter. From a software perspective, a program was developed for the ATmega328P to read serial command data over UART from Dakota's image-based control program, and to convert this data to command bytes to be sent to the digital potentiometer circuits inside the Hak909 quadcopter controller. The hardware aspects involved reverse-engineering the Hak909 quadcopter controller; identifying critical circuit components inside the controller, such as the joystick potentiometers, microcontroller, and the wireless transceiver; completing partial disassembly of the existing control circuit; identifying appropriate replacement parts, namely, the Microchip MCP42010; and finally, integrating the hardware design into the existing control circuit.

E. Hardware

The hardware used for the digital control system includes:

- 1x ATmega328P
- 2x MCP42010 10 $k\Omega$ digital potentiometers
- 2x 1/4W 10 $k\Omega$ resistors
- Hak909 quadcopter and controller; Hookup wire; Breadboard.

The hardware used for the altitude sensing and parallel mapping is shown below:

- 3x ATmega328P
- 3x BMP 180
- 3x 16 Mhz Crystal
- 3x 18pF Capacitor
- 3x 22pF Capacitor
- 3x Perfboard
- 21x ¼ W Resistors (seen in Appendix A)
- 3x 9V Lithium Ion Re-chargeable Battery
- 3x LM 7805 5V Regulator
- 3x 10 MM Red LED
- 3x 10 MM Blue LED

The hardware that was chosen for the list above was from using previous knowledge of the microcontrollers, regulator, and LEDs. As for the different resistances were determined by the calculated current and voltage in the data sheets to power the BMP 180, MCP42010, LM7805, ATmega328P and not drawing too much current form the 9V battery.

Three perfboards were chosen because we needed to attach them to three different Hak909 Drones that will be below the AR Drone 2.0, giving them parallel mapping and altitude sensing capability.

F. Software

The software for the digital control system reads four integer data bytes from the serial port at 9600bps. The program parses the incoming bytes, each of which are separated by commas, and saves these bytes to a different variable for each control parameter: throttle, pitch, yaw, and roll. The newly-read variables are then printed to the serial monitor to verify correct data transmission. Then, an SPI command is sent to each of the digital potentiometers containing the variable for each control parameter, which updates the wiper position on the control potentiometers. As for the altitude sensing, this will take into account the initial position and then compares it to the continuous position until the microprocessor is reset. Then the sensing will be communicated to the image processing to give the correct position to the Hak 909 Drones through the digital control system.

G. Interface

In order for the overall system to function, it is imperative that each subsystem can properly interface with one another. Appendix B shows a high-level flow chart for the overall system.

In order for the Hak909 Digital Control subsystem to receive command signals, a 9600 bits per second baud rate UART connection is established with the image processing laptop. The laptop transmits control signals, which the ATmega328P captures and converts to digital potentiometer wiper positions. The digital potentiometers are then read by the microcontroller on each of the Hak909 handsets, which send data over a 2.4GHz wireless RF signal to the Hak909 quadcopters to control flight.

Used UART to transmit data to Ethan’s code Using UART, we are able to send information from the control program to the device responsible for moving the drones.

The ability using these components comes with being able to read the Barometer data using a 26ms delay in between each reading to give an error of +/- 10 inches of error for altitude readings. Along with giving a 1 second timer of sending a 150 mA pulse to the camera button which will open and close the button switch. Once this is done the barometer will send the altitude to the ATmega328P through I^2C

communication that will then compare this to the set standards of our desired altitude of being in between 15 and 25 feet. If the follower UAVs the Red 10MM LED will turn on, if the BMP 180 is in between 15 and 25 feet after take-off no LED will be on, and finally if the sensor is above 25 feet the Blue LED will turn on. After the LED is on or off the image processing subsystem will detect this and determine further calculations. The Circuit Design is shown as below in the Appendix, Figure.

H. Economic Analysis

Part	Cost	Quantity	Total Cost
Quadcopters(Hak 909)	85.00(1) then 125.00 (2)	3	$ 335.00
Microcontroller(Atmega328P)	$ 4.38	6	$ 26.28
Barometer(BMP 180)	$ 9.95	5	$ 49.75
Digital Pototentiometers	$ 2.00	6	$ 12.00
16 Mhz Crystal	$ 0.53	3	$ 1.59
22 uF Capacitor	$ 0.10	3	$ 0.30
18 uF Capacitor	$ 0.10	3	$ 0.30
RGB	$ 1.95	12	$ 23.40
10 MM	$ 1.00	10	$ 10.00
LM 7805(5 V Regulator)	$ 1.95	3	$ 5.85
Resistors	$ 0.10	50	$ 5.00
Camera(808 Pinhole Camera)	$ 9.95	3	$ 29.85
SD Card	$ 8.95	3	$ 26.85
AR Drone 2.0 Flight Recorder	$ 106.98	1	$ 106.98
9 V connectors	$ 3.75	3	$ 11.25
Final Cost			$ 644.40

Figure 12 – Project Spending

III. DESIGN TESTING AND VALIDATION

A. Digital Control Subsystem

The Hak909 digital control subsystem contains multiple components. The ATmega328P program was designed for UART read/write transmission and SPI interfacing with the Microchip MCP42010 digital potentiometers. The MCP42010 was installed in the Hak909 handset controller to provide for digital control of throttle, pitch, yaw, and roll. Finally, we needed to verify that a user-controlled killswitch could terminate flight on command.

To test the UART functionality, we used a serial monitor program with a USB serial connection to the ATmega328P. We tested sending 4 control parameters – throttle, pitch, yaw, roll – by sending them via serial and printing them back to the monitor to verify correct transmission. Figure 13 shows a screenshot of a received serial message containing the control parameters.

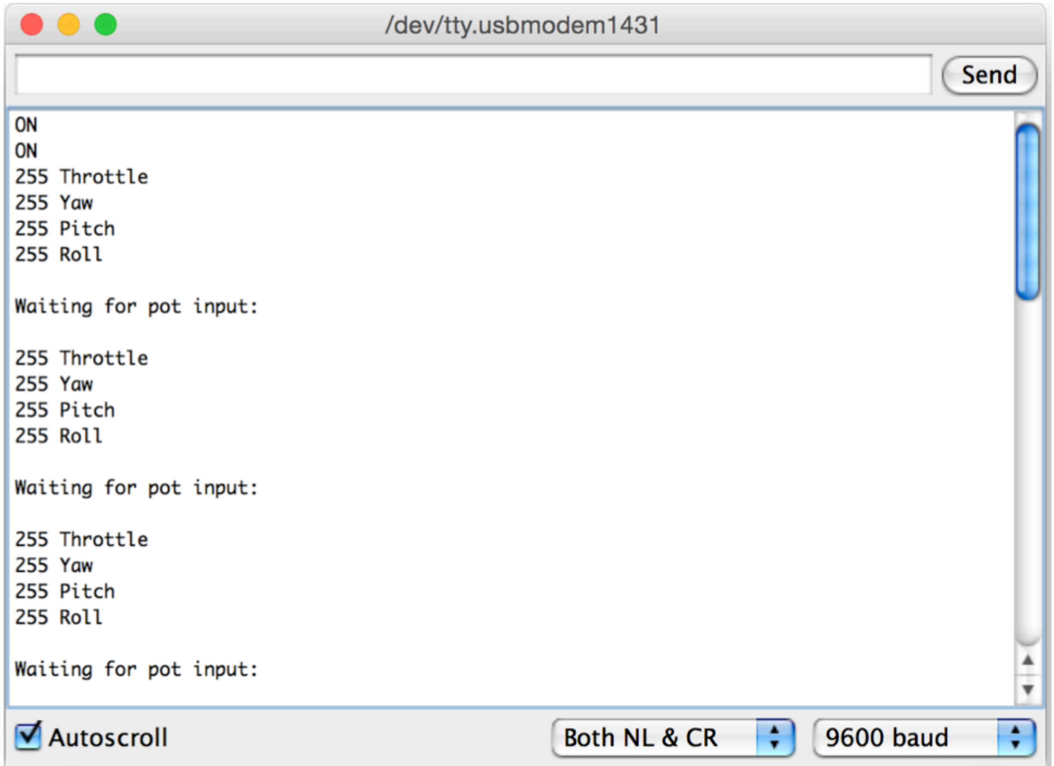

Figure 13 - Serial Monitor with Control Parameters

To test the ATmega328P interface capability with the Microchip MCP42010 digital potentiometers, a test circuit was assembled, shown in Fig. 14. Using the serial monitor, a value of 0 was sent to the digital potentiometer, which corresponds to a wiper position of close-circuit with the B terminal of the potentiometer. With an inherent wiper resistance of 75Ω, the resistance between the wiper and B terminal measured at about 90Ω. Then, I tested a max value of 255 sent to the digital potentiometer, which results in a closed circuit between the wiper and A terminal. Because the wiper is now furthest from the B terminal, the full resistance of the digital potentiometer should exist between B and the wiper terminals. We measured this resistance at about 4.9kΩ. As each MCP42010 is equipped with two digital potentiometers per chip, we tested both potentiometers on the chip using the same process.

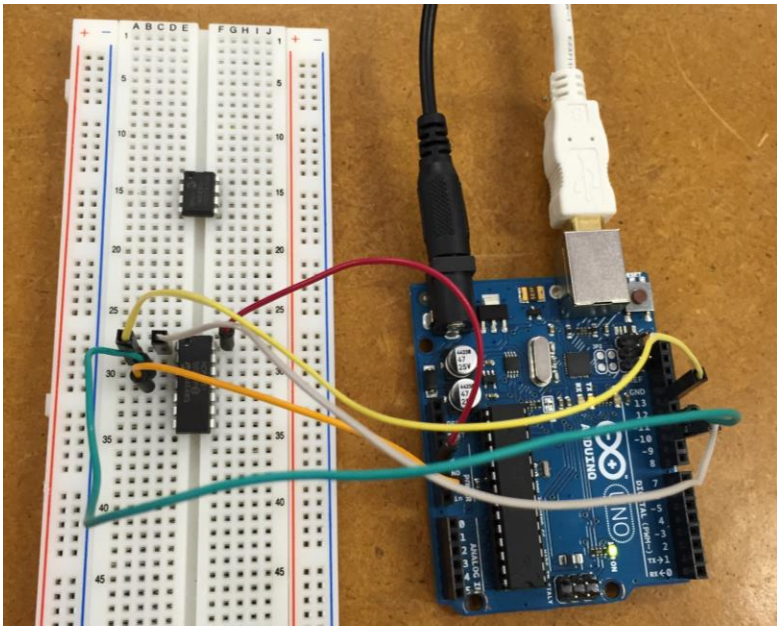

Figure 14 - Digital Potentiometer Test Circuit

Once the serial communication and digital potentiometers proved functional, testing of the hacked Hak909 controller was all that remained for the subsystem. After removing the existing controller joysticks by desoldering them, we soldered header pins into the joystick through-holes in the PCB. Using jumper wires we connected the B, wiper, and A terminals of each of the four digital potentiometers to the corresponding terminals on the Hak909 controller for the throttle, pitch, yaw, and roll joystick potentiometers. We powered on the controller after powering on the Hak909 quadcopter. The controller buzzer beeped to indicate a successful connection. Then, we slowly increased throttle until the rotors began spinning. We tested the throttle over the entire range from 0-255; there was a dead band from about 0-25 which we attributed to friction. We also tested yaw, which has a neutral (zero) value at 128. We verified the yaw commands were working by checking that only two rotors were spinning at once, with speed varying based on the magnitude of the digital potentiometer signal. We were unable to test pitch and roll due to unforeseen circumstances in which the Hak909 quadcopter controller began malfunctioning.

The user-controlled killswitch was verified rather simply. A low level throttle signal was sent to the Hak909 to maintain consistent rotor movement. Once this control signal had been established, we simply turned the power switch on the controller to the "Off" position, then back to the "On" position to reset the quadcopter. The throttle command was returned to 0 after the controller was powered up again.

B. AR Drone Subsystem

Figure 15 - Color Detection of Hak909 Quadcopters

Figure 15 shows the color detection is able to locate the quadcopters in non-ideal environments. As the quadcopters will be on the Alumni Quad which is a low traffic area, and is mostly grass, the image processing will allow us to determine the location of the drones and control them.

Figure 16 – Edge Detection

As shown in Fig. 16, the edge detection allows us to find objects between 1-10 meters. This is a worst-case scenario, however, as the majority of the alumni quad is open field, as such, the quadcopter will be able to determine objects in the alumni quad.

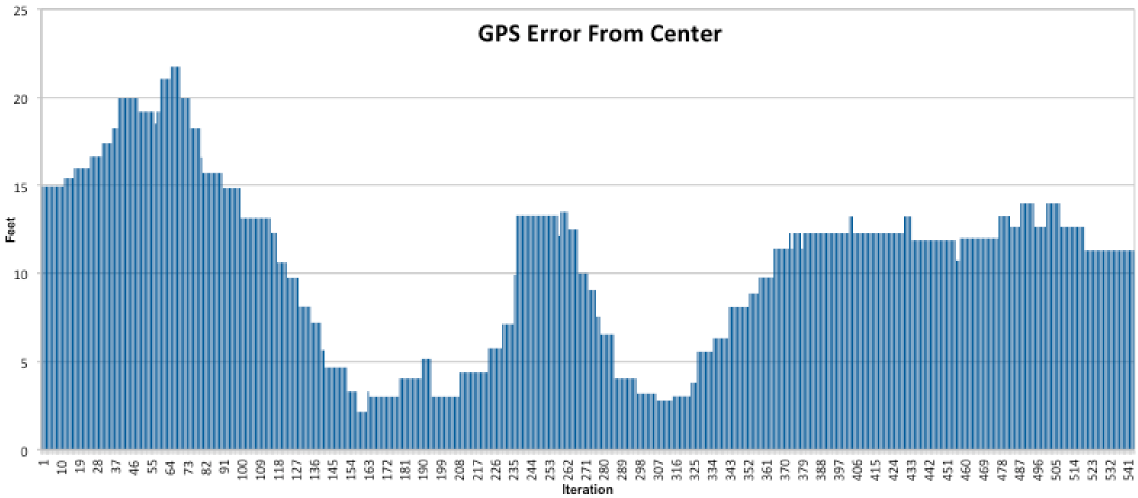

Figure 17 – GPS Error Data

Figure 17 shows the GPS data has approximately 8 meters of error at maximum. The magnetometer data is highly dependent on location, as magnetic fields in the vicinity can throw off the magnetometer. We were unable to test the magnetometer, however, allowing 5 degrees of error allows us to have a 7 meter error at maximum GPS error, which is still within the 100 meter radius that we can control the quadcopter in. Thus, we are confident in the ability of the magnetometer and GPS to navigate a region.

C. Altitude Sensing, Camera, Image Stitching Subsystems

The hardware portion of subsystem is to be attached to each Hak909 quadcopter which is design to be able to communicate the results of the to the other subsystem, then able to map out the region that is below.

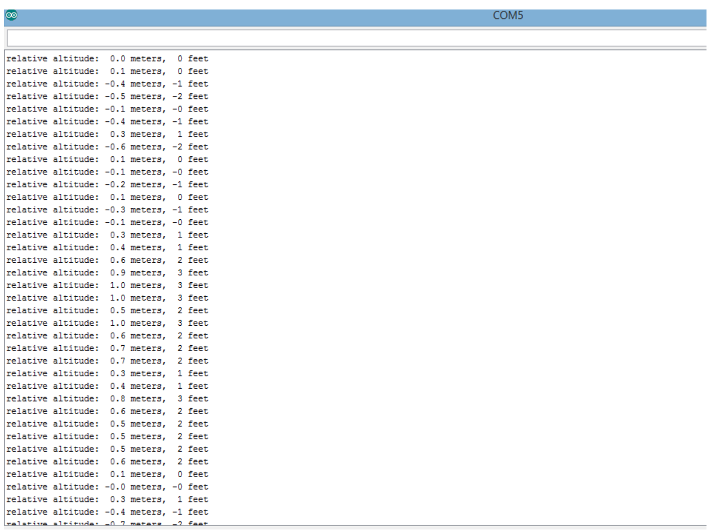

Figure 18 – Altitude Data

In order to test the capability of the Design there was a hardware segment that had to be tested before a perfboard design was implemented. Using serial communication to be able to show the position while the BMP180 was lifted into the air, and then rested back onto the table it showed that it went up to 3 feet, and then back down to 0 to -1 feet (within 10 inches). Now to be able to power the red LED there was the necessary specifications of above 20mA and 2.2V and the blue LED needs 20mA and 3.2V. This was the purpose of the ATmega328P comes from being able to send 150mA which will be divided through the 100-200 ohm resistors (for amperes).

Figure 19 – Initial position (Left) to 30 Feet (Right)

Sending the signal based upon position was shown using the elevator in Jobst Hall, to be able to go from the first floor to the fourth floor (0-30 feet) after being attached to the perfboard and quadcopters.

Figure 20 – 808 Pinhole Camera Attached to Hak 909.

Once the ability to turn on the pin depending on height, all that was needed was to implement a 1 second external timer that would be shown that the capability of sending a 150mA pulse from the ATmega328P was in practical rather than in theory. Being able to send this pulse allowed for the ability to solder to the shutter button of the camera one to the selected pin and the other to ground to create a switch on each pulse as shown in Fig. 20.

Finally when the images are taken and put into a SD card, each has a timestamp to be removed, shown in Fig. 21. Then after the timestamp was removed the missing timestamp would be auto-filled based on surrounding pixels, which was done in Photoshop. Once the timestamp is removed it will take the plethora of images and be able to stitch them together show in Fig. 21. Finally the stitching images are put into a collage to create a single flat parallel map.

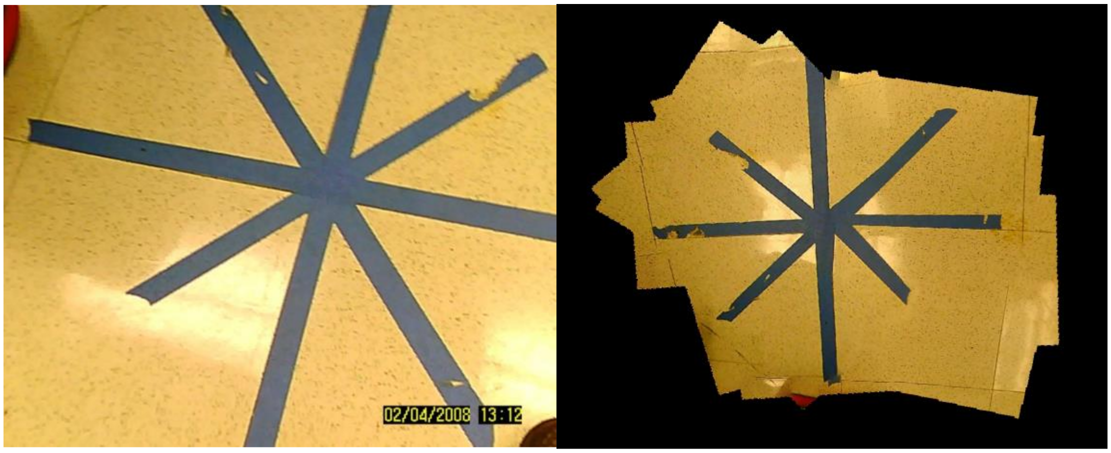

Figure 21 – Image of 808 camera (Left) and multiple images filled and merged (Right)

IV. Conclusions

We were able to complete a majority of different subsystems required for the overall goal of having multiple unmanned aerial vehicle (UAV). This being said, there is still future work that needs to occur such as the autonomous flight development through the potentiometers and communications with the image processing. If the project were to be continued, it is probable that the device would work in a unison flocking scheme. Therefore, the project was a successful endeavor, but leaves room for improvements and completions.

V. Acknowledgements

We would like to thank our senior project advisor, Dr. Joseph Driscoll for seeing us through this project, along with Dr. Jose Sanchez and Dr. Gary Dempsey for instructing us throughout the research. Thank you to the Electrical and Computer Engineering Department at Bradley University for supporting our work, and any opinions, findings, conclusions or recommendations expressed are those of the authors and do not necessarily reflect those of the Electrical and Computer Engineering Department.

VI. References

[1] W.S. Hall, S. Jian, D. Yang, H. Xi, "A multi-UAV dynamic formation control method," Chinese patent CN102591358 A, July 18, 2012.

[2] S. Quintero, G. Collins, J. Hespanha, "Flocking with Fixed-Wing UAVs for Distributed Sensing: A Stochastic Optimal Control Approach", Dept. of Electr. & Comput. Eng., Univ. of California, Santa Barbara, Santa Barbara, CA, Tech. Rep. ISSN 0743-1619, June 19, 2013.

[3] N. Depenbusch, J. Langelaan, "Coordinated Mapping and Exploration for Autonomous Soaring," Penn. State Univ., University Park, PA, Tech. Rep. AIAA 2011-1436, Mar. 31, 2011.

[4] (2014). Flock 1 [Online]. Available: https://www.planet.com/flock1/

[5] D. Marr, E. Hildreth Proc. R. Soc. Lond. B 1980 207 187-217; DOI:10.1098/rspb.1980.0020. Published 29 February 1980

[6] R. Manduchi and C. Tomasi, "Bilateral Filtering for Gray and Color Images" in IEEE International Conference on Computer Vision, Bombay, India, 1998, pp. 839-846

[7] S. D. Levy and J. Stough, AR.Drone AutoPylot [Online]. Available: http://home.wlu.edu/~levys/software/ardrone_autopylot/

[8] (2014). BMP180 Barometric Pressure Sensor Hookup. [Online]. Available: https://learn.sparkfun.com/tutorials/bmp180-barometric-pressure-sensor-hookup-

[9] (2014). Bit-Switch Micro-Sized R/C On-Off Switch. [Online]. Available: http://www.rc-cam.com/bitsw.htm

[10] (2014). #8 Camera. [Online]. Available: http://www.chucklohr.com/808/C8/index.html

[11] (2014). Controlling toy quadcopter(s) with Arduino. [Online]. Available: http://dzlsevilgeniuslair.blogspot.com/2013/11/more-toy-quadcopter-hacking.html

[12] (2014). #8 Camera. [Online]. Available: http://www.chucklohr.com/808/C3/Sven/1007Sven.html

[13] (2014). #8 Camera. [Online]. Available: http://www.chucklohr.com/808/C3/Sven/1007Sven.html

VII. Appendix A

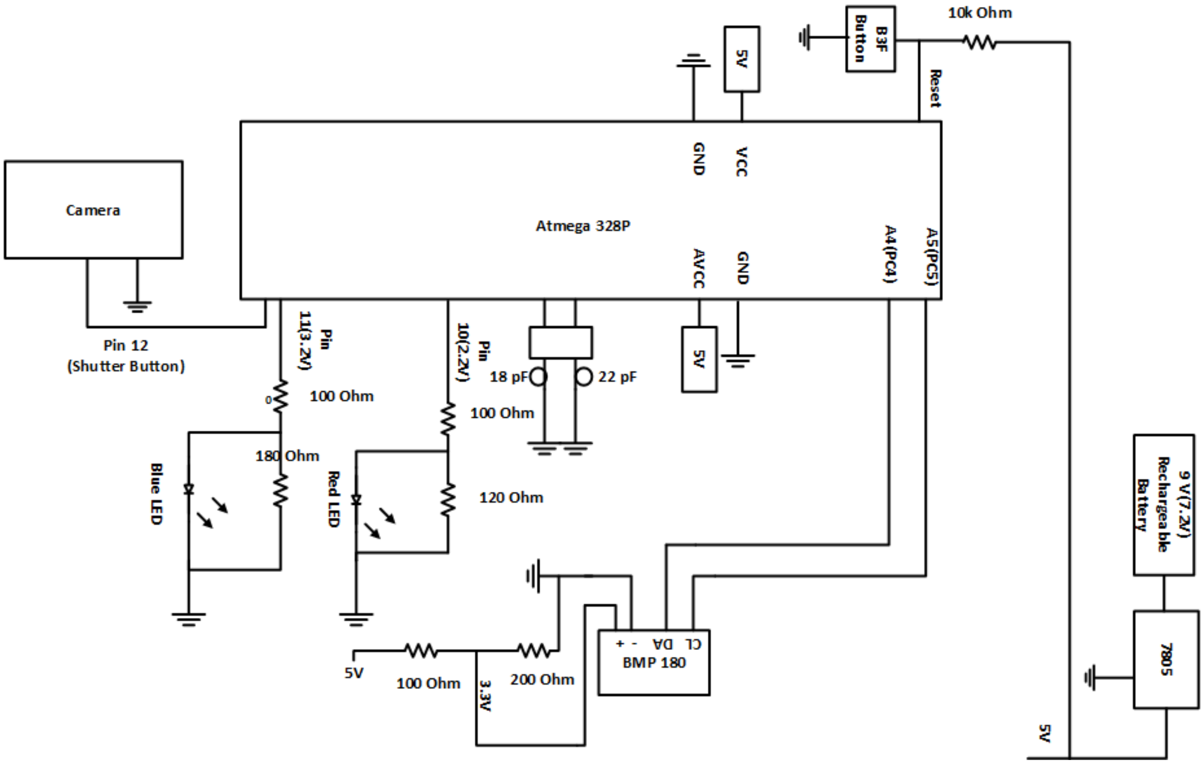

Figure 22 – Altitude Detection LED Circuit

VIII. Appendix B

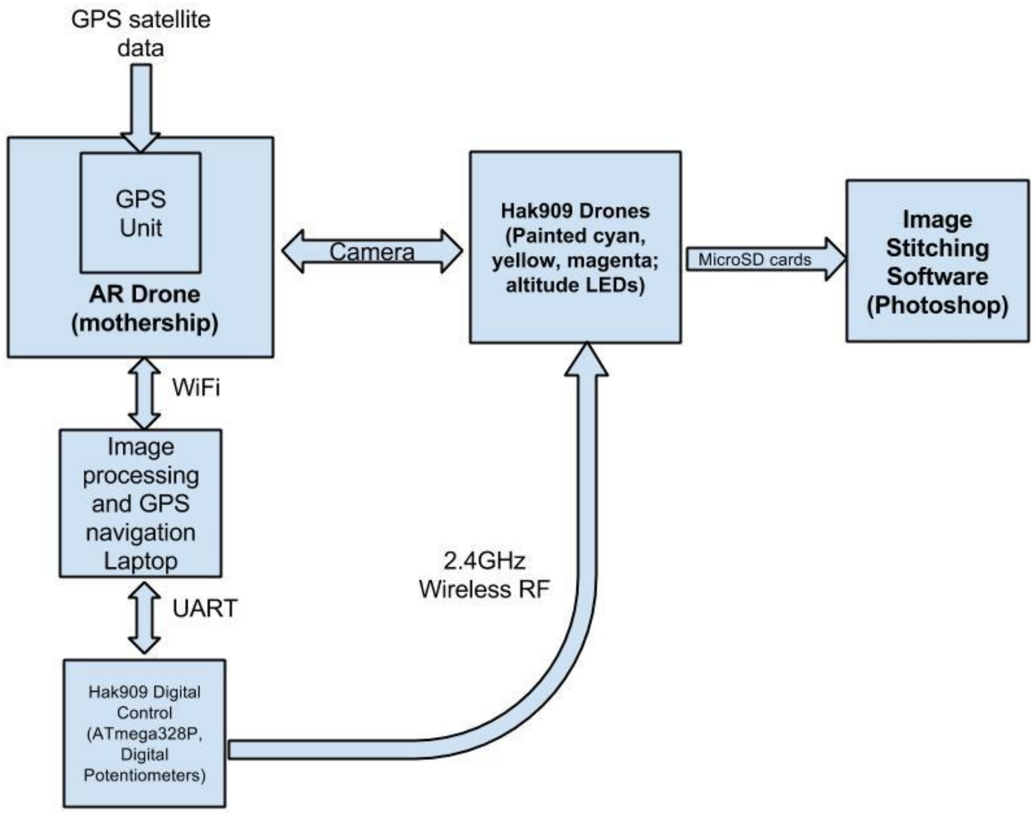

Figure 23 – High-Level System Flowchart