# Mobile Aerial Surveillance System (MASS)

Henry Chan, Eric Hernandez, Derrick Shrock, and Sanjay Yerra

University of Central Florida Department of Electrical Engineering and Computer Science, Orlando, FL

***Abstract*—Surveillance is the first line of defense against intruders and quick detection of unwanted activity in your vicinity can make the difference of your personal safety. The Mobile Aerial Surveillance System (MASS) was developed in order to provide the user with high-altitude surveillance of an area and provide basic detection of human objects within the rotatable camera view. MASS's structure is our own homemade blimp design which is large enough to support the microcontroller, camera, servos, and motors but also designed to provide a smooth flight for the user to view the real-time video on our C# graphical user interface (GUI) design. The blimp movement and camera system will only be controllable through our GUI design which needs a laptop or desktop to be plugged into our base stations which we refer to as the Groundstation design.**

**Index Terms - Surveillance, Blimp, Microcontroller, Programming, Wireless Communication, Computer Vision, Servomotors, Patrol Flight.**

## I. INTRODUCTION

The Mobile Aerial Surveillance System (MASS) is a large blimp which has a camera system attached to it to provide the user with a overview of their surrounding area. The potential for the system is to give the user the ability to fly overhead with the camera system and auto-detect any people in the area. If they want to keep surveillance on one person in the camera view, they will be able to start tracking that persons' movement with a click of a button on the User Interface for the blimp system on a computer. Although the blimp is a odd way of creating a Unmanned Aerial Vehicle (UAV), we chose it as a more stable way of aerial video capturing the area. More conventional ways of designing surveillance UAVs is helicopters, quad-copters, and airplanes, but most of these design fly at high-speeds or have a unsteady structure where the camera feed can become too unstable for image processors to detection objects within the frame.

This project has qualities that are not only good for interest in security, but from engineering stand point, it is also very green because this system requires no fuel or natural resources. The only resource it needs is the electricity to charge the batteries required to power the propellers and other fundamental parts it requires. From a personal standpoint, we as engineering students have been studying the fundamentals for a while now. This project will require us to not only use the skills we have been learning through our education, but will also require us to ascertain skills that we have to research and further investigate in order to complete the main task at hand and not to mention will be a great stepping stone for the key into being successful engineers.

This papers will first cover the components in our design and their general purpose in our design in Section II (System Components). The parts described are the parts we are currently testing on the blimp and are subject to change if we cannot it to perform its function. In Section III (Implementation and Design), we will cover the functionality of our blimp and the cool features we are putting into the programming of the microcontroller with relation to the system components and the Graphical User Interface (GUI) in which the user can choose options of what they see (video feed) and how the blimp is controlled (directional buttons).

## II. SYSTEM COMPONENTS

In order to communicate between all the components in the system we had to connect them in a certain way, Figure 1 below shows our connections between subsystems and their label in our project.

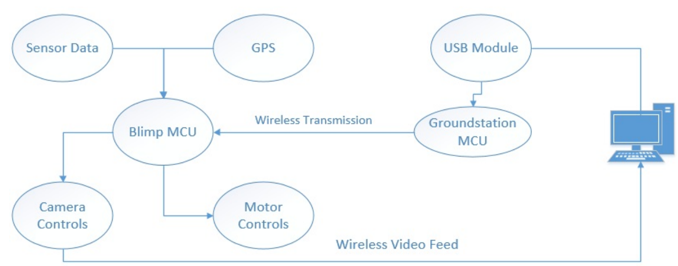



Fig. 1: Block Diagram of Components

The main camera has to be on its own transmission line for video feed. The communication between the two microcontrollers has to be wireless and have a decent range (100-400 meters) to go up in the air high enough for suballiance. The blimp microcontroller has to interface with the GPS, the sensor data, and control all the servo motors on the camera and the motors. All of these components working together give us our surveillance system and is made for us to implement the features we want on them, such as Detection, Tracking, and Auto-Routing.

### A. Wireless Communication

The most important part of the design would have to be the communication to our high altitude flying blimp, if we lose the connection to the blimp, it may float away and we would never see it again. The main functionality we wanted the components to focus on were the ease of integration into our microcontrollers, low power, and long range. We chose two different frequencies to work with, the main control link works on the 2.4Ghz band and the camera link will work on the 5.8Ghz band, this is in order for the frequencies not to interfere with each other. We researched into trying to put the video and string data into the same transmission band, but we found that the bandwidth requirement and the control protocol to distinguish to two signals weren't easily implementable and would take too long for our project time frame.

*1) Control Link:* The transmission to control driving the blimp and the camera movements is done by a 2.4Ghz RF transceiver, this module is the NRF24L01+. The transmission range of the nRF24L01+ sending out data at 256 kbps has a range of 1000 meters. These range test are open area range test, which means that there are no trees or buildings in the way. Our hopes is for it to at least get 200-300 meters from this transmitter, but if there are trees or buildings in our way the range of the transmitter decreases dramatically and we would still need to put in a failsafe in the blimp if we lose transmission. The input voltage for the transceiver is 1.9 to 3.6 volts, and we will supply a 3.3 volt power line on the microcontroller to the nRF24L01+. The current draw for the transceiver module is 11.3mA at our optimal 0dBm output power and for the receiver the maximum it will be is 13.3mA.

*2) Camera Link:* The transmitter and receiver combo for the Camera System is a 5.8 GHz system in order not to interfere with the 2.4 GHz frequency of the control transmission. The combo that was purchased was a 5.8 GHz STB Wireless Sharing Device AV Transmitter / Sender & IR Remote Extender With 200M Transmitting Distance. The 200 meter distance was originally too short for what was needed but after some experimenting 200 meters was decided that it was perfect. The modules are slightly bulky and the antennas are large and plastic but the overall weight of the transmitter is very light and will be placed inside a wooden carrier. The size was a little troubling but the weight of 72 grams with the antenna included was decided to be good enough to work with. The camera will be directly connected to the transmitter via RCA cables. The transmitter then will transmit the picture to the receiver which had to be connected to an analog to digital convertor in order to produce a picture on the laptop.

### *B. Blimp Control System*

The Blimp Control System handles all the servo controls and the input from the sensor and GPS module on the Blimp while it is in the air. This system uses the NRF24L01+ for long range data communication, for the directions in movement from the GUI and thus the user.

*1) Microcontroller:* On the Printed Circuit Board of the MASS is the Atmel Atmega328p. The Atmega328 has a RISC architecture and has numerous possibilities when it comes to controls. The microcontroller is used in the Arduino Uno an entry level development board that is used by hobbyists all over the world. The micro controller has a full 32kBytes of programmable flash memory. Although the amount of memory needed was a guesstimate in the beginning of the semester, the amount of memory in the Atmega328p was more than enough. The package that was used was the PDIP which has 28 pins. This packaging was used for the sole purpose of needing to program the microcontroller, if need be, on a breadboard. Other possible reasons were for the freedom from surface mounting and easier testing. The ATmega328p has serial communication capabilities.

The Atmel Atmega328 has ports for I2C, SPI and UART, all of which will be used in our blimp to make it run fully. The SPI port will be connected to the wireless transceiver which will send in string commands from the user which in turn tell the microcontroller to implement commands such as turning on or off servo motors. The I2C port will be used for an inertial measurement unit, or as it is called the IMU. Through the various addresses, the data from the surface mount modules will be sent to the master which is the microcontroller in this case. This data will be used for direction of the blimp (magnetometer), stability (gyroscope), and motion (accelerometer). The UART port will be used for reading in data strings from the on board GPS module. This data will be used to orient the position of the blimp anywhere in the world.

*2) Global Positioning System (GPS):* We choose the 927-A2100-B model by Maestro Wireless. The A2100-B model is the first model made by Maestro using CSR's SiFRstarIV chip on GPS modules. It is a fast, highly responsive GPS that has accuracy down to -163 dBm. The GPS module that is on the PBC of the MASS is the Time to First Fix GPS module. It has 20 possible channels meaning, 20 possible satellites to receive GPS coordinates from. This allows for the most reliable position with the least amount of noise. The module is a Surface mount device. The codes that will be read from it are the GSA and GLL codes. GSA will find active satellites and GLL is for finding the longitude and latitude of the module.

*3) Inertial Measurement Unit (IMU):* The IMU consist of a Gyroscope, Accelerometer, and the Magnetometer. The board that we are using is the Atmel AV 4018 which designed as a daughter board for the Atmel AVR Xplain line of development boards. Our plans are to incorporate this board and grab data from it on our Atmel Atmega328p Microcontroller. The IMU uses the I2C serial output for data output on all three modules, from the IMU 3000 Gyroscope we receive the degrees, from the Accelerometer we receive the g's of the x, y, and z directions, and from the Magnetometer we receive the current heading in degrees (the compass for our routing between GPS coordinates).

### *C. Camera System*

The camera system will be placed just slightly towards the front of the blimp while the motors will be in the front and the gondola will be in the middle. The camera system should have two ways of being moved. When on the autopilot feature the IMU or more specifically the gyroscope should sent data to the Atmega 328 microcontroller and the Atmega would then account for the tilting in the blimp and move the system accordingly. When on user controlled interface mode the camera system can be controlled by using the left trigger stick to pan up to 180 degrees or tilt180 degrees. The 180 degree limit worked well because too much tilting will just show a video of the blimp while too much panning will lead to see a balsa wood gondola which does the flyer no good when trying to survey the ground.

*1) Camera:* In order to put the surveillance in MASS a camera system was needed. The system included a camera and a stabilization system. The camera module that was used for the MASS was the SONY SC2000 CCD 540 TVL MINI. This camera met all the specifications that were needed and had numerous examples of how it worked online. Two other cameras were possible options before this camera was chosen. The GoPro Hero Three Black Edition and 1/3" Sony CCD 480 were other options. The 1/3" Sony CCD had a good amount of features as well and was the first camera purchased. After four weeks of fighting with the international website all hope was lost on this camera and a change to domestic cameras become a necessity. Thus this lead to choosing the SC2000. The SC2000 met the three criteria that were most important: cheap, light and fast shipping.

The camera measures at a minuscule 30mmX10mmX10mm which is approximately the size of a quarter. Weights a slight 32 grams and seeing as this is a blimp and weight is at a premium this was a major win. It would also ship from a domestic warehouse courtesy of Amazon and would be in within a few days. The camera has a 540 horizontal resolution which sounded on the lower end of clarity, however after some comparing video quality with other cameras it matched up very well. It will definitely be strong enough to identify people and items from several hundred feet up in the air. The camera came ready-made to be hooked up to a transmitter or a wall plug in. It came with a set of RCA cables that typically go straight into a TV monitor and it came with a wire connection which is to plug directly into a transmitter which was the original plan for the project. The camera uses a 12 Volt power source which is by far the largest voltage consumed on the blimp as well as a 150 mA of current. Due to these larger values the camera uses its own power source of four three volt lithium batteries in series. The camera also had a very minimal delay coming straight from the transmitter/receiver combo to the laptop. The camera will be set up on a stabilization system.

*2) Stabilization System:* The stabilization system is a very simple set up made of two open metal boxes that have grooves set up on the flaps that slide together in order to give movement. The following picture will help clarify the exact description of how the stabilization system looks. It is held in by several self-tapping screws. Figure 2 shows an example of what the stabilization system will look like.

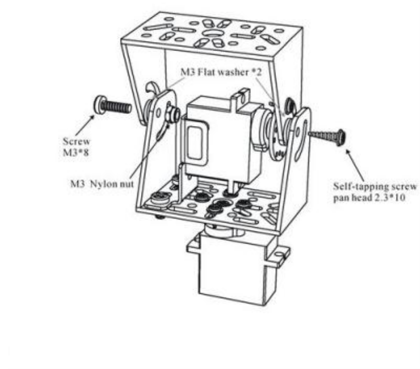



Fig. 2: Camera System Stablization System

*3) Servos:* As can be seen the camera stabilization system also has two mini servos attached to it. The two mini servos are 9g rated and run on 5.5 volts. One servo controls the rotational movement of the camera or the panning motion while the upper servo controls the verticality of tilting of the camera. The camera system was designed to be super lightweight but still dependable enough that if some rain or windy weather came through the camera would survive.

The servos will be controlled by the microcontroller by the use of pulse width modulation signals. Depending on the input from the user the servo will pan or tilt at a determined amount of degrees. For instance a push to the left will rotate the pan servo 3 degrees. The servos will be powered by a Li-Poly 11.1 Volt 2200 maH battery. The battery has two output voltages, one will go to the motors and one will go to the servos. The second output consists of a voltage, a ground and a control cord. The output voltage will be 5 volts which corresponds perfectly to the voltage needed for the servos.

*4) Video Receiver Encoder:* Finding a convertor was the most difficult part. After numerous hours searching a video capture card was the answer. Originally the Elgato was going to be used to for the conversion but it only recorded in mpeg4 format. The OpenCV program was not able to read this format so instead EZCAP was bought. EZCAP along with VLC made it possible to stream video into a program called direct show. This will all be discussed in the GUI section later. The EZCAP uses ULEAD software to show and edit video which would not work with the software that was needed. Like what was stated before VLC was used to stream into Direct Show which is compatible with the OpenCV software. The transmitter uses a 5 volt power source with 1000 mA current draw which will be powered by the battery on the blimp. The receiver will be on the ground so directly plugged into an electrical receptacle will be fine. The EZCAP will just be plugged into the laptop and no powering is necessary.

### *D. Groundstation*

The purpose of the ground station is to act as a base station for all communication to the blimp and to the controlling programs (the GUI). We want to interact with a computer program, so we added a USB connection for the computer from the Groundstation microcontroller. The ground station will also house the second transceiver, the first being on the blimp, for long range communication (up to 750 meters, but but only expect 200-300 meters at best). To control all these communication modules we will need a microcontroller to direct out all the signals to the corresponding places as well as set the transceiver to transmission (TX) or receiver (RX) mode. Next to the ground station also plugged into the computer is the Camera Video Feed's Receiver box with the encoder. This receiver is part of the ground station setup but will not be included in the component box housing the data control link.

*1) Microcontroller:* The microcontroller we chose for the ground station is from the Microchip PIC line. We used a different company of microcontrollers (from the Atmel Atmega) since we wanted experience with different types of microcontrollers and their communications with each other. We chose the PIC18F2620, we went through a different microcontroller, the PIC18F2550, but we found out that hardware UART and hardware SPI shared the same pins. So we looked into getting a new PIC getting both serial communications hardware defined on seperate pins and came up with the PIC18F2620. We run the PIC at 40Mhz using a external crystal and have the SPI pins used for the NRF24L01+ transceiver. We programmed the microcontroller in MPLAB X and use the C18 compiler which is a free microchip PIC C library compiler for the PIC18 family of microcontrollers.

*2) USB Connection:* For the USB connection we used a UART to USB module, the CP2102. This module fit our needs for connection to the computer and simply uses the UART pins of the microcontroller to interface with it. The nice thing about this module is that we do not have to use a 48Mhz clock on the microcontroller to control the speed of the USB device, instead it is all built into the CP2102 module. We only use the RX and TX with ground pins of this module since we do not want to power anything through this module and we do not need the reset pin, but it does come with 3.3V and 5V power lines for USB charging.

### *E. Blimp Structure*

For the balloon of the blimp, we decided to make our own, since a advertising blimp cost upwards of $700 for the weight lift that we needed. It turned out much cheaper to get the material and bond it together into a balloon ourselves. It might not be as aerodynamic as a bought balloon but it fits our purposes pretty well. The undercarriage and the axel mount for the motor are also part of components we needed and are custom built to house our PCB and other components.

*1) Envelope:* The envelope is the main structure of the entire assembly design. Since the envelope is the building block of the MASS it needed to be measured out perfectly. Made up of six different sections, also known as gores (refer to figure for section design), which had to be measured out perfectly, the envelope ended up coming out to eight feet in length and three feet in diameter. The design process of the envelope is displayed in Figure 3 below.

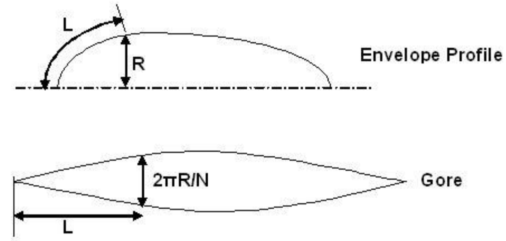



Fig. 3: Main envelope section designing process. L is the value of the arc length, R is the value of the radius, and N is the number of sections in total.

It is made out of a lightweight plastic material that was mended together with a heating process using an iron. Although it was effective, the mending process took a lot of testing and many different variants of trials, but in the end the envelope is of the non-rigid design not including any type of frame and also takes the popular elliptical shape. Some of the problems we encountered while mending the sections together were the fact that we kept getting holes from the iron heating the plastic too quickly, however when we lowered the heat, it did not want to seem to melt the plastic to a strong hold. As a solution we just added patches to where the holes had developed and for the ones we could not see we added a rubberized sealer to the seams.

*2) Stabilization Wings:* The stabilization wings are made of lightweight insulation foam. Cut out into four equally measured, semi-oval shaped wings, they will account for the MASS flying at a straight like pathway and are designed to cut down on movement from wind by making the system more aerodynamic. They are attached to the envelope in the rear by line anchors.

*3) Gondola:* The gondola serves the purpose of containing the circuitry of the system. To maintain the consistency of being lightweight we used balsa wood as the building material. This is attached to the envelope in the central bottom area for weight distribution. Since the material is thin and lightweight our GPS and transceiver for the controls and camera system will not be affected as far as any interference is concerned.

*4) Motor Mounting Bracket:* Since we want our system to be as stable as possible we wanted to mount the motors in the best area to maximize the stability. We had originally planned to put the props and motors in the gondola, however we figured since the gondola is located at the bottom of the blimp it would not be as stable. We decided to build a separate bracket using balsa wood to hold the motors and mount it toward the front.

### F. Blimp Movement (Motors)

The ability to control our blimp and having smooth pitch and yaw movement is crucial so the user does not feel they don't have good control on the blimp. The system for the blimp movement was implemented to give us flexibility in setting speeds and to reduce the components needed to make our turns.

*1) Brushless Motor:* We are going to have two brushless motors running the blimp propulsion. We decided to go with the Brushless Outrunner 2217-4 motors. The require li-po batteries and run at 80% efficiency. These motors are connected to a mounting bracket and will be located toward the front of the system. Since we will be controlling the speed of the motors to be able to turn we set them on an axel type system to control the pitch.

*2) Axel Servo:* The axel servo is simply just a servomotor controlling the pitch of the prop motors. It is a 180 degree servo the will go from the motors pointing straight down to straight up. It is the s3004 model, which is a heavier and stronger servo.

*3) Electronic Speed Controller (ESC):* The BP 30 AMP Brushless Electronic Speed Controller is what we are using to control the speed of the motors. These speed controllers, one for each motor, will help with not only the speed of the system, but the ability to turn the entire blimp as well.

## III. DESIGN AND IMPLEMENTATION

In this section, we discuss the features of the blimp using the components described in the previous section, along with the final integration of the components together. The main features we bring to the table by integrating these components together is: the auto-stablization of the camera through the sensor data, tracking and following system through the camera system and OpenCV, auto-routing a user defined patrol path using the GPS and magnetometer, and manual user control of the blimp movement through the brushless motors and the servos by commands send out from the computer side GUI.

### A. Blimp Control System

The program on the blimp controls the movement of the servos by the received commands of the transceiver unit. Table 1 below shows the the commands along with the string command used to trigger it is. The string command is always 6 characters long and the first 3 characters determine which system the servo is on, so "mot" for motor control (driving the blimp), and "cam" for camera controls (for turning the camera). The next three numbers denote the angle at which the servo should turn or increase speed to, which control line (which servo) to do it to, and the last number is which direction the servo turns in 1 for positive and 0 negative, which we denoted while testing the command and can vary between how the servo is oriented in the structural design.

| Command | CmdString |
|---|---|
| Motor Left Turn | mot311 |
| Motor Right Turn | mot321 |
| Motor Pitch Up | mot301 |
| Motor Pitch Down | mot300 |
| Camera Left Turn | cam310 |
| Camera Right Turn | cam311 |
| Camera Pitch Up | cam301 |
| Camera Pitch Down | cam300 |
| Auto Pilot On | auto11 |
| Auto Pilot Off | auto22 |

Table 1: Commands send to blimp for servo movements.

The process for speed control is currently being tested in two different ways. The first way is to set speed control levels, having 5 different pre-defined setting levels and the user can pick between the 5 for the throttle. the second way is to use a accelerate and decelerate button, this would send commands to decrease or increase the motors certain increments at each button press and would have more flexible ranges the blimp can accelerate up to.
The last two commands on the list happen when the user selects tracking and following or the patrol path on the GUI. The blimp will no longer take the manual commands for movements and instead base it off specialized commands. The commands for the tracking and following has varying degrees it needs to turn to keep the person in the center of the screen. but will almost always turn the camera servos instead of having to rely on the motor controls. For the patrol path, the blimp will wait to accept 5 or 6 GPS coordinates send out by the computer GUI. Then it will use the magnetometer on the blimp to calculate heading using the formula below.

$$\theta = \mathrm{atan2}(\ \sin(\Delta\lambda).\cos(\varphi_2),\ \cos(\varphi_1).\sin(\varphi_2) - \sin(\varphi_1).\cos(\varphi_2).\cos(\Delta\lambda)\ )$$

Formula for Degree between two GPS Coordinates

*1) Eagle PCB Design:* As every project has to have a PCB design this one was no different. This PCB design was actually very easy considering almost all of the components were going to be put on headers. The Eagle design had only a handful of components including a NRF2401L+ transmitter, a Maestro A9207H1035 GPS Module, Atmega 328 microcontroller and IMU module. Only the GPS and the microcontroller will actually be soldered onto the board while the other components will be attached to male headers. No power will run throughout the board but instead because there are only 4 devices that need power so a prototyping perf board will be connected to every device. The PCB will only be using two layers with very limited and simple connections. Figure 4 below show the final PCB layout of the blimp controller.

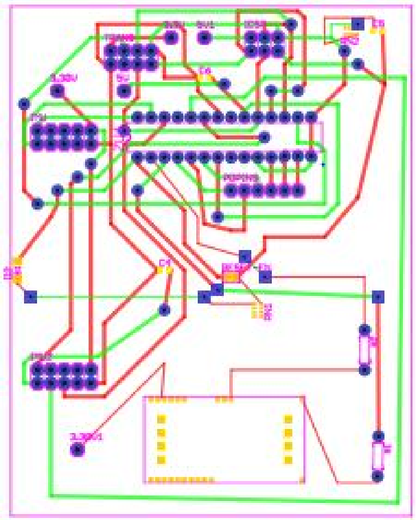


Fig. 4: Blimp Main Controller Eagle PCB Design

A reset button is also implemented as a recovery to the board. There are header pins also implemented in order to run a JTAG programmer through. The IMU was the most difficult part because it has two sets of male pins that need to be exactly 460 mil apart which meant making the board revolve around it. The microcontroller uses DIP packaging and will have 28 pins. Most of the pins will be unused but were hooked up to open headers just in case one of them is needed. Most of the unused pins are ADC pins. The actual board will be 3.26" by 3.93". Which will fit into the gondola perfectly and leave more space for other components.

### *B. Camera System*

Design for the camera system was pretty forward. The only real design is the actual stabilization system. The design was to be made simple and lightweight. Thin plated metal mixed with two mini servos distinctly gave the requirements desired for the system. It was put together by screws, self tapping screws and nuts. The system is also user friendly seeing as almost any servo size can be used in case they are broken during installation. The implementation for the camera video feed will go from the camera to the the transmitter through AV cords. Then the transmitter will send it to the receiver which will then send the signal to the EZ CAP. This video capture card turns the signal digital so it is readable on a laptop. Finally VLC makes the program runnable with OpenCV and the video feed is completely streamed. The implementation of the stabilization system involves having user controls or automated controls which run through the microprocessor to tell the servos which way to pan or tilt. The user will press a button on the joystick and the GUI will transmit the data to the transmitter and the microcontroller will then tell the servo motors to move accordingly.

### *C. Graphical User Interface (GUI)*

The program to interface with the Surveillance Blimp will be programmed in Visual Studio C# in order to create a graphical user interface for the user to easily use the blimp. The primary features of this program will include a link to the Video Feed on the Balloon and a user control system for the acceleration/deceleration/ascent/descent/turning of the blimp which will include a buttons on GUI for users to press.

On the Video Feed for the GUI we take in the data from the USB connection from the video encoder. We use an OpenCV wrapped for the C# language called Emgu CV, it has all the same functions of OpenCV written in C++ but using the C# language to call and edit it. To detect people on the video feed we use the HOGdescriptor for person/pedesterian detection. The objects detected are highlighted by a rectangle and then the center of the object is calculated using the detection rectangle as a base shape to find the center.

We put a mini map on the GUI for the user to click the points and the program to gather the longitude and latitude from the points clicked. These points are then sent to the blimp to auto path on a confirm button click. The mini map uses a library created for C# called GMaps, which pulls map data on from OpenStreetView and displays it on the screen, it also makes the map interactive through various events you can apply to it, one such event we apply is on_mouse_click. The markings are placed on the map as you click and there is a button to clear them.

The other feature integrated into the video feed is the Tracking and Following System. For this system we make use of OpenCV to detect the person in the frame. We are limiting the person you can track to one person, since we had trouble doing multiple detection, originally there was going to be an overlay on top of the camera feed but returning the images from it did not work out correctly. So we have button that one openCV detects a person, you will click on it and it will keep that person in the center of the frame by sending commands to the blimp to move the camera servos or the motor controls.

### *D. Blimp Structure*

We have estimated the blimp to be around seventy cubic feet of air. Since helium has a lifting ration of about 28.2 grams per cubic foot we have estimated to lift about four and a half pounds. This will have an affect on the aerodynamics of the structure by obtaining ‘lift,’ or in our case, weightlessness. We do not want the blimp to float, but we want it to fall at a very slow rate.

### *E. Power System - Blimp*

The whole blimp will be powered by two Li-Poly 11.1 volt 2200maH batteries and four 3 volt lithium watch batteries. The four 3 volt batteries will be placed in series to power a 12 volt camera. The 11.1 volt batteries will be responsible for powering the motors and the perf board. There are two DC brushless motors with electronic speed controllers that will be directly hooked up to the batteries. These will take up most of the current given off by the batteries and power the blimp so it can move. The other components will be powered by the second output from those same batteries.

The perf board will be responsible for powering all of the units on the blimp minus the camera and the aforementioned motors and speed controllers. It will have two batteries connected to it that will power the 3 servo motors, the transmitter and all the PCB components. The two batteries will implement two 5 volt voltage sources which will be regulated and dispersed to the correct components. There will also be a 3.3 volt regulator which will regulate voltage to all the components that use 3.3 volts instead of 5 volts. The perf board will nested alongside the PCB inside the gondola.

### *F. Power System - Groundstation*

The power system of the ground station was determined by the need of a 5V and 3.3V rail. The power consumption from the transceiver, the CP2102 (UART to USB) module, and the PIC18F2620 required these fails and the current draw from it was very low (around 200mA at most). We decided to use linear voltage regulars to and a outlet adapter which supplies the initial 9V to step down to 5V and 3.3V. The voltage regulator we use to step 9V to 5V is the LM2940C-5.0 which can handle inputs from 7V to 26V and steps it down to 5V. Then we cascade the LM2940 with a LM3940 which steps down the 5V to the 3.3V we need for the transceiver module. We put a 22uF capacitor on the input of the voltage regulars and a 46uF capacitor on the output of the voltage regulators to smooth out the power input and the LM2940 actually won't work without capacitors of specific values on it. We chose the linear voltage regulators over the switching regulators because the circuit design of the switching regulator was too complex to put on the perf board and took too much area and the current draw on the components wasn't high enough to where we would generate a lot of heat on the voltage regulators (we never max out the 1 amp limit on the voltage regulators).

## IV. TESTING

Testing for the controller occurred on the Arduino Uno development board. The Arduino Uno is powered by a USB to SPI connector and has the Atmega328P, which is in the MASS blimp. The Uno was used to write and test code that was eventually used in the final version of the blimp. Most of the functions in the blimps programming used the open source Arduino libraries. Code developed to control all of the various aspects of the blimp system. In order to get signals to the blimp, the blimps program needed a plethora of serial communication inputs. The final blimp needs to use three different methods of serial communication, I2C, SPI, and UART for input data from the various modules. The program HyperTerminal was downloaded to test reading input and output data to and from the microcontroller board.

Testing the various servo motors was done by inputting incrementing duty cycle percentages into the motors. This program design layout did eventually end up into the blimp's final programming. The final blimp will feature a remote control that will allow the user to speed up or slow down the blimp as well turn the camera or the blimp itself. To test the camera stabilization system, the Arduino board had to get serial data from HyperTerminal. The user would input a button press which correspond to an increment in the percentage of the duty cycle of the Pulse Wave that servo motors read in for controls. Through testing and various specifications from group members the code was developed to eventually completion.

The testing for the motors involved researching the electronic speed controllers and learning the quirks of the motor. There was very little information no how to control the motors without a direct link to the RC Transmitter/ controller. Testing needed to be done to figure out how a microcontroller could emulate the signal sent by those kinds of controllers. By testing various duty cycle percentages and finding out the proper arming sequence, the motors were able to be controlled very well. The testing found that the motors did not work with a duty cycle of less than 50%. To compensate for this the group tested out differing sizes of propellers as well as different motor sizes. Testing for the blimp's turning system was done by continuously changing how the duty cycles on the blimp blimps two brushless motors. Eventually the group members decided on an economical approach on how to implement turning in the blimp. The decision was to read in the command that the user wanted to put in, left or right, and then a function would be called that would implement turning the blimp. The blimp turns by simply slowing down the opposite motor. For example, to turn right, the right motor will lose up 30% of duty cycle relative to the left motor. This difference in motion will cause the blimp to turn. The reduced speed of the motor will in theory last for up to 3 seconds.

Testing the GPS module was only possible after the final PCB was in, due to that fact that the funds needed for another GPS module and the breakout board needed to test it were unavailable to the group. Testing was done by sending the serial data from a terminal to a USB to UART module. To test the IMU, I2C serial communication needed to be implemented. This required testing on the Arduino Board to fully integrate the gyroscope and accelerometer data into the blimps microcontroller. Testing the SPI was needed to receive controls from the ground station and ultimately from the user. Testing and writing the code to implement the serial communication was done on the Arduino Uno development board.

Testing the transceiver unit for the blimp and connection to the computer GUI happened separately from the blimp but ultimately it will all be tested together. the transceiver unit test was created by a mock blimp receiver unit to communicate with. The final board design is the tested perf board design and will be housed in a box. We tested the received right commands of the blimp by using a LED light display corresponding to different commands received by the transmitter. We send these command using hyperterminal with the connected UART to USB module. Then we moved onto adding the serial communication to the C# GUI and tested the commands on button pushes of the GUI.

Full deployment of the blimp in the air is being tested, but there are still some problems with the integration on the ground without the floating envelope. Once all the ground test are completed and tested to our satisfaction we will begin all the user control test in the air and test out the features we implemented with the GUI to communicate to the blimp. We will also test out what will happen on transmission loss and hopefully the balloon won't float away while we are debugging those problems.

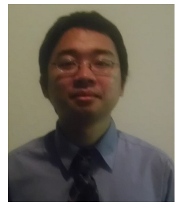
## V. CONCLUSION

Our project design was a fun project in which we got to fly a remote controlled blimp and spy on people from above. Our surveillance blimp's aim was for an alternative way of personal security while introducing cool features such as the tracking through the visions algorithms provided in OpenCV, and patrolling through GPS coordinates on the microcontroller for automatic flight using sensor data. Trying to smooth out the motor controls was a difficult process and we feel like there is much more progress to be done in the smoothing out of the controls for the users. We gain many new experiences in working as a group and researching new topics and this project is only the starting point in our future own personal hobby projects.

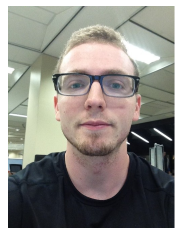

## VI. ACKNOWLEDGEMENTS

The authors wish to acknowledge the assistance and guidance of Dr. Samuel Richie.


## VII. BIOGRAPHIES



Henry Chan was born in Hollywood, Florida. He is a Computer Engineering student at the University of Central Florida with a focus on Hardware Design in FGPAs using VERILOG. Currently he is pursing job opportunities as a FPGA designer after he graduates.



Eric Hernandez was born in Chicago, Illinois. He plans to graduate from the University of Central Florida with his Electrical Engineering degree in the fall of 2013. He plans to continue his studies and earn his Master's degree in Power. He is currently employed at EXP as an electrical intern.

Derrick Shrock was born in Orlando, Florida. He is an Electrical Engineering student at the University of Central Florida and will receive his BSEE in December of 2013 with a focus on filter design and power analysis. Currently pursuing job opportunities in motor power generation and simulation.

Sanjay Yerra was born in White Plains, New York. He is an electrical engineering student with a passion in power and electrical engineering. Currently he is employed at exp as electrical engineering intern. He plans to pursue a master's degree in electrical engineering after graduation.