Poppins - The Autonomous Weather Balloon

ECE 445 Design Document Fall 2020

Marc Havery (marcdh2), Arturo Cuevas (acueva8), Katie Stapleton (kes8)

Team 15

TA: Yifan Chen

**Abstraction**

Our team designed an intelligent payload system that provides self recovering capabilities for high altitude weather balloons. This proof of concept focuses specifically on guiding the parachute supported payload to a predestined GPS location. It accomplishes this by collecting real time data from a GPS sensor and digital compass, which is then interpreted by the microcontroller to provide adjustments to servo motors that pull on parachute brake strings in order to turn. After completing our design, we were not able to comprehensively test our design from a high drop, but with our simulation results and test trials we can see the fundamental implementation was a success

Table of Contents

[**1 Introduction**](#_4n0nyz15cvsk) **3**

[1.1 Objective](#_ydns1tu1jtie) 3

[1.2 Background](#_sel2zaderfe3) 3

[1.3 High-Level Requirements](#_85a7r4arhrji) 4

[**2 Design**](#_oe2vxzg4et1n) **5**

[2.1 Navigation System](#_397jqhllw4gq) 5

[2.2 Steering System](#_35xegnr7j0yn) 6

[2.3 Power System](#_vdt0gspopnyc) 7

[2.4 Software](#_zbze7aafpjb8) 7

[2.5 Schematic Block Diagram](#_jl718ftlqov) 8

[**3 Requirements & Verification**](#_m0we54t1h5uo) **10**

[3.1 Navigation System](#_5q0h1cqmw4iv) 10

[3.2 Steering System](#_s2726gg6exmv) 10

[3.3 Power System](#_jn8w4ndlsual) 11

[3.4 Software](#_2dvjnpeku9sw) 11

[3.5 Full System Test](#_1fo7tzyke2pm) 12

[**4 Cost and Schedule**](#_6igzuvg1fcg) **13**

[4.1 Cost Analysis](#_mxtupk89puer) 13

[4.2 Schedule](#_bbxknk42t3ig) 14

[**5 Conclusion**](#_6ie4nlm5d7tv) **15**

[5.1 Accomplishments](#_pnkt9hofo2nm) 15

[5.2 Uncertainties](#_evf7on3fbclf) 15

[5.3 Safety and Ethics](#_u0bpu7oevixd) 15

[5.4 Future Work](#_6a7u33jutgce) 15

[**Appendix 1: Requirement & Verification Table**](#_uvrx6e9qf3hc) **16**

[**REFERENCES**](#_53pf2vksry8e) **18**

#

# **1 Introduction**

## 1.1 Objective

High altitude weather balloons are an essential part of data collecting for many sciences, ranging from weather forecasting to pollution monitoring. Two weather balloons from about 900 sites around the world are launched every day of the year [1]. They consist of a large balloon, usually filled with either helium or hydrogen, and a payload, which often has sensors and computers to obtain data from the heavens. As the balloon and payload float up, the air pressure decreases, and the diameter of the balloon expands until it pops.

Once it has popped, the payload deploys a parachute and returns to the ground, where data is collected. Researchers can predict where the weather balloon will land using calculations, but ultimately, due to outside variables such as jetstreams, they have to rely on GPS to retrieve the payload. Unfortunately, this is an issue when balloons often land in hazardous conditions as it often results in the data collected, as well as the payload, getting destroyed or lost.

Our solution gives the balloon a way to guide itself and reposition or apply tension to its parachute’s brakes to land at a designated GPS location. Our custom payload, displayed below in Figures 1 and 2 that includes power, navigation, and steering systems, must follow FAA regulations for utilization by other weather balloonists.

## 1.2 Background

Meteorologists, researchers, and the military deploy these balloons with a diverse and expensive payload of equipment that can exceed thousands of dollars. Besides the financial investment, scientists pour countless hours of work to collect this data that can improve the fields of meteorology, photography and astronomy [2].

The landings of these small unmanned aircraft are also important as they can cause fatal damage. The high frequency of weather balloons and growing urban sprawl are increasing the chance a weather balloon could land in trees, bodies of water, residential areas, or even highways. Due to the nature of the electronic payload, there are cases of crash landings where the payload starts a high blaze [3]. In the wrong situation, these balloons could cause wildfires, destruction of property, or vehicle accidents.

The ability to determine a secure landing location for weather balloon payloads provides a potentially invaluable tool to the scientific community. With just a small amount of overhead, researchers can confidently launch their balloons and ensure it will avoid large catastrophes and loss of their valuable data.

##

*Figure 1. Physical diagram of Poppins parachute and payload systems..*

##

*Figure 2. Physical diagram of payload with outer view (left) and inside view (right).*

## 1.3 High-Level Requirements

* The payload needs to be able to interact with the parachute correctly. It must be able to steer it right or left in much the same way a skydiver would, but use motors to pull on the strings.
* Our software needs to be able to make correct, logical decisions on which way to turn based on inputs from the various sensors. It must be able to decide which action is optimal at every point, and interact with the motors to make that decision happen.

#

# **2 Design**

**

*Figure 3. Poppins High Level Block Diagram*

Figure 3 shows that at the highest level our design can be divided into a small Ram-Air parachute and a custom payload. The light weight payload accomplishes our desired parachute flight guidance by incorporating a navigation system that directs the motors of our steering system. These motors are flasen to our parachute’s brake strings in order to turn the payload back on course to the target GPS location in real time. Each of these system components will be explained further in the following sections.

## 2.1 Navigation System

The main component of this project and what allows us to compute the logic for steering and landing is the navigation system. This system takes in input from the various sensors and uses a microcontroller to decide how to power the motors to steer the parachute. This system is connected entirely to our PCB and is powered by its dedicated battery. The PCB also connects the output decision of the nav system to the motors in the steering subsystem.

2.1.1 Microcontroller

 For the brain of the navigation system, a suitable choice was an AVR microcontroller as it can operate fast enough and at a compact size. Our selection was an ATmega328P-PU which takes in as input real time data from the GPS and digital compass. The input of a target GPS coordinate, a current GPS coordinate, and a compass heading are computed every 2 seconds based on the navigation software (See section 2.6) to either turn the left or right steering motor.

 The largest design issue of our project spawned from this component, and completely changed our PCB design. We discovered through breadboard testing that the microcontroller’s communication with the motors was being constantly interrupted by the GPS device. Luckily the ATmega328 provides hardware pins that allowed us to create a PWM signal that was separate from the timer used by the GPS.

2.1.2 GPS Device

 The GPS device provides reference for the payload to know where it is located in relation to the target. Our choice was an Adafruit Ultimate D GPS Breakout, which communicated directly to the microcontroller and powered the digital compass with its noiseless 3.3 V output.

2.1.3 Digital Compass

 The navigation system gets its current heading direction from the use of a digital compass. Our SparkFun HMC6343 Breakout digital compass provides cardinal heading direction to the microcontroller and is powered entirely by the GPS device.

 The decision to rely on the GPS to power the digital compass was made in order to simplify the complexity of our PCB. The compass was the only 3.3 V mandatory device and would have required an additional voltage regulator. To increase modularity of the design, the PCB should not have this connection and should instead consist of more compatible devices for the selection power supply.

## 2.2 Steering System

 The steering system is the main mechanism for communicating the navigation system’s directional decisions to the parachute. The system consists of two digital servo motors, guided by the microcontroller, that pull on the steering strings of the ram air parachute. By having these two arms we can make the parachute turn left, turn right, and stay on target.

2.2.1 Motors

The two digital servo motors receive data signals from the microcontroller and are powered by their own power supply. Each motor is zip-tied to our payload and the motor arms are tied to each of the brake strings.

A limiting factor we encountered during the design was that motors demand a large amount of current, especially when stalling. To accompany this effect, we went through another design iteration of the PCB so that the motor’s power traces were wide enough to hold the current (See section 3.2.1) .

2.2.2 Parachute

 The ram-air styled parachute is fundamental to the design, as it is capable of being flown with the left and right brake strings. The design we chose was based on current uses of RC paramotor parachutes, as they are the appropriate size of about 2.4 meters. High altitude weather balloonists most commonly use a circular parachute because it is a simple drop and catch design, however this style fits those designs and supplies more piloting functionality.

## 2.3 Power System

The power system must be able to power the two digital servo motors as well as the whole navigation system. For any project including motors, it is best practice to dedicate a power supply to combat large fluctuations in current draw.

2.3.1 Lipo Battery

The lipo battery has the sole purpose of providing constant voltage and necessary current to the two digital motors.

2.3.2 Battery Pack

The battery pack is used to power the whole navigation system and ensure the GPS also has a constant voltage and current source.

2.3.3 Step Down DC-DC Buck Converter

 The DC-DC buck converter will be used to step down the main 7.4 V lipo to 3.3 V for the Compass and 5 V for the motors, microcontroller, LCD, and GPS.

## 2.4 Software

 The payload makes constant and real time decisions based on the information provided by the sensors. The algorithm is based on the FSM in Figure 4 and the control flow in Figure 5 below. GPS and compass initialization and communication is accomplished through Sparkfun [5] and Adafruit [6] libraries.

**

*Figure 4. Finite State Machine for project software.*

**

*Figure 5. Flow Diagram of project software.*

## 2.5 Schematic Block Diagram

The Schematic Diagrams in Figures 6 to 8 below encompasses our PCB design including components from the power, navigation, and steering systems.

*Figure 6. Overall schematic block diagram. GPS continues to the left and power system on the bottom.*

* *

*Figure 7. GPS schematic. Inputs and outputs connected to the overall schematic.*

*Figure 8. Power system schematic. Outputs connected to the overall schematic.*

**

*Figure 9. Final physical PCB layout.*

# **3 Requirements & Verification**

*For a comprehensive list of requirements, see Appendix 1 for each subcomponent.*

## 3.1 Navigation System

3.1.1 Microcontroller

 The requirements for our ATmega328P-PU microcontroller had to align with the selection of market GPS devices and digital compasses. After our purchase of both devices, we required a microcontroller with at least 12 digital pins and a clock speed of 12 Mhz for the GPS. To verify our ATmega was comatabile, we ran the sample scripts for each device on the breadboard level and affirmed our initial research that they all could communicate with the microcontroller.

3.1.2 GPS Device

 The Adafruit Ultimate D GPS Breakout provided the necessary update frequency of 1 Hz, lightweight design, and max altitude of 60,000 ft for our high altitude payload. The GPS was verified by running the Adafruit sample script and measuring the serial monitor to see an update rate of 20 Hz.

3.1.3 Digital Compass

 The digital compass was a flexible component as it only required IC2 connections and lightweight design under 1 oz. The Sparkfun HMC6343 Breakout fit these requirements and was verified with the sample Github script from Sparkfun.com.

## 3.2 Steering System

3.2.1 Motors

The major requirement of the motors was that they could supply at least 15 kg per cm of torque. This was determined by researching the standard motor strength of our variation of RC parachute. We verified that our 20 kg per cm motors sufficed by loading various weights to the arms and measuring their current draw with a multimeter.

3.2.2 Parachute

 The parachute we need for our project needed to be highly maneuverable and light weight. A small Ram-Air parachute served our requirements of being able to carry the payload weight of 10 oz and have steering strings that can be fed into our internal steering system [4]. Weights were hung from the support strings to prove it could hold the design.

A hole in our design logic lives here, as we have no quantitative data on this type of parachute at high altitude. A perpendicular experiment needs to be conducted in order to say this type of parachute has air cells large enough to hold our payload’s weight.

## 3.3 Power System

3.3.1 Lipo Battery

The lipo battery is dedicated for only powering the two digital servo motors. The main requirements are such that the battery can supply the operating voltage and max current draw of the two motors. After purchasing our motors, we required a Lipo that could supply at least 5 V and a max current draw of 2.3 Amps. Our selection of a 2S 7.4 V Lipo battery completed these requirements and was verified with a multimeter.

3.3.2 Battery Pack

A simple battery pack is needed to power the navigation system and must also fulfill the necessary operating voltage and total current draw. After our purchasing of the navigation system components, the total current draw was 150 mA and a operating voltage of 5 V. For testing purposes we utilized 4 AAA batteries to get a total of 200 mA draw and a voltage of 6 V. The pack was verified with a multimeter.

3.3.3 Step Down DC-DC Buck Converter

 We utilized two DC-DC buck converters to step down the 7.4 V lipo to 5 V for the motors and step down the 6 V AAA’s down to 5 V for the navigation system. Our converters also supported the max current of 2.5 Amps and were verified with a multimeter.

## 3.4 Software

The software needed to correctly navigate the payload to target coordinates. As seen above, we based our algorithm on the fact that having a fixed threshold for your “green zone” (the range of directions where no turns are needed) is enough to get infinitely close to a target (ignoring factors like wind), assuming turns can be completed quickly. This method also requires that the threshold is always set below 90 degrees, making the maximum “green zone” size 180 degrees.

To show that our algorithm works, we made a simulation and plotted some results. The below plots show the distance between the payload and the target location against time. The simulations had the payload at a constant velocity of 1 m/s, and placed the target at around 55 meters away from the payload’s starting point. This shows that our algorithm works fundamentally under perfect conditions.

Looking at the below graphs in Figure 10, it appears optimal to have as small a threshold as possible. This is true in the perfect simulation world, but overcorrection and wind are severe issues in real life scenarios, so we decided to go with a 45 degree threshold to be safe.





*Figure 10. Turning logic simulation results.*

## 3.5 Full System Test

For our final demonstration of the project’s functionality, we held the payload containing the batteries, PCB with sensors and buck converters, motors attached to the side of the box, and parachute brake strings tied to the end of the motor arms. As we demonstrated, data from the GPS, compass and software logic was displayed on the LCD. Once the GPS coordinates were set and the compass was calibrated through rotating the PCB around the x-axis, the PCB was functional. This was apparent because as we turned the payload to aim the front away from our previously set target, both the output logic variables signaling the motors (TL, TR) changed values and the respective motor arm pulled down on the parachute break.

For example, in the test, the compass data told us we were facing an orientation of 11 (about North) degrees, the ideal direction the software calculated was 58.15 (about Northeast) degrees, so we were 47.15 degrees to the left of our target. Our system then triggered the right motor to turn right which was shown through both the TR variable being set to one as seen on the LCD screen and the right motor arm pulling down on the right parachute brake.

# **4 Cost and Schedule**

## 4.1 Cost Analysis

4.1.1 Labor

Factors that determine the labor cost:

* Average UIUC graduate engineer hourly wage = $50/hour
* Overhead multiplier = 2.5
* Estimated hours worked to complete project = 8 hr/wk \* 6 weeks = 48 total hours
* Number of team members = 3 people

We use the following equation to calculate the labor cost for one person:

*Labor Cost = (hourly wage) x (overhead) x (total hours) x (team members)*

 *= ($50/hour) x (2.5) x (48 hours) x (3 people)*

 *= $18,000*

Therefore, our total labor cost is estimated to be $18,000.

4.1.2 Parts

|  |  |  |  |
| --- | --- | --- | --- |
| **Component**  | **Description**  | **Distributor** | **Price** |
| Parachute | 2.4 m Paramotor Sail | Hobby King | $33.67 |
| Servo Motor x2 | 20 KG Digital Servo | Amazon | $33.98 |
| Microcontroller | ATmega328P-PU | Amazon | $5.99 |
| Lipo Battery | 7.4 V Lipo Battery | Amazon | $17.99 |
| GPS Receiver | Adafruit Ultimate D GPS Breakout | ECE445 Lab | $0 |
| Electronic Compass | SparkFun HMC6343 Breakout | ECE445 Lab | $0 |
| Logic Level Shifter | SparkFun Logic Level Converter - Bi-Directional | Sparkfun | $3.00 |
| LCD Display | Standard LCD 20x4 | Digikey | $27.06 |
| Voltage Regulator x2 | 5 A DC-DC Adjustable Buck Converter 4~38 V to 1.25-36 V | Amazon | $8.99 |
| Styrofoam Box | Insulated Foam Container | Amazon | $10.00 |
| Crystal Oscillator Clock | Bojack 16 Mhz Crystal Oscillator Clock | Amazon | $9.95 |
| Ceramic Capacitors  | 2 x 22 pF 1 x 100nF  | Personal Inventory | $0.00 |
| PCB  | 100 mm x 80 mm Printed Circuit Board  | PCB way  | $23.18 |

*Table 1. Cost of project components.*

The total cost of the physical components in our project is $173.81

4.1.3 Total

Adding up the previous totals from labor and components, the entire project’s final cost is:

$18, 000 + $173.81 = **$18,173.81**

## 4.2 Schedule

|  |  |  |
| --- | --- | --- |
| **Complete By:** | **Finished Tasks:** | **New Tasks:** |
| 10/02 | Design Document | Begin purchasing parts |
| 10/05 | All parts purchased | Begin designing PCB based on parts (Katie, Arturo) |
| 10/10 |  | Begin designing accurate depiction of payload (Marc) |
| 10/15 | All parts acquired except parachute, PCB design finished (Katie, Arturo), Depiction of payload finished (Marc) | Begin unit testing on breadboard (All) |
| 10/25 | Finish unit testing payload on breadboard (All) | PCB begin soldering (Arturo) and test ATmega328 |
| 10/30 | Finish PCB soldering(Arturo) and ATmega328 testing (Marc) | Unit testing the completed circuit (All) |
| 11/02 | Unit test finished and payload assembled (All) | Begin final testing |
| 11/04 | Test complete system | Begin filming demo |
| 11/08 | Demo | Final presentation |

*Table 2. Development Schedule*

# **5 Conclusion**

## 5.1 Accomplishments

 Overall, our payload system worked well on ground level tests. It was able to correctly steer the parachute based on data from its sensors. In conjunction with our successful software simulation, it shows that with no wind interference and a large enough drop height, our payload would be able to land much closer to a target than a traditional weather balloon payload.

## 5.2 Uncertainties

 Unfortunately, we were unable to perform any drop tests, which would have been helpful in determining how much the wind and imperfect turns would have affected our results. Because we don’t perfectly know how wind will interact with the parachute, it is hard to determine if countermeasures are necessary. After performing a drop test it will be possible to fine tune some hyperparameters as well as make the decision as to whether or not different functionality is needed for the wind.

## 5.3 Safety and Ethics

 The utility of this design can only be measured if it follows all safety and ethics guidelines. When the device is fully operational it faces issues with privacy and public safety. Weather balloons are useful tactic devices for military surveillance operations because they do not show up on radar and are silent. Also, the additional capability to land a payload at a GPS location also means this device could be used as a weapon. These concerns will need to be weighed by both engineers and the FAA to ensure the safety of the community and environment are kept a priority [7].

## 5.4 Future Work

 Looking forward, we want to make improvements for high altitude and efficiency. For high altitude, we will need to use larger batteries that can sustain the navigation and steering system, as well as a heating system to make sure the batteries don’t get too cold as they get higher. For efficiency, we want to perform drop tests so we can see to what extent wind affects our algorithm, and then act accordingly. This could include adding wind speed and direction sensors, and then altering our algorithm to account for wind.

 Finally, we would like to restructure our design into some kind of lightweight addon, so that it can be easily attached to normal weather balloons to get the benefit of a more predictable landing, but without all the hassle that our group went through during this project.

# **Appendix 1: Requirement & Verification Table**

**Table A1 System Requirements and Verifications**

|  |  |  |
| --- | --- | --- |
| Requirement | Verification | Verification Status(Y/N) |
| 1. Microcontroller1. Capable of executing at least 12 Mhz single level instructions per second.
2. At least 12 IO to interact with all sensors.
 | 1. Sample program tested with a similar workload using the LCD to verify the program output.
2. Verify through the data sheet and test each pin configuration individually.
 | Y |
| 2. GPS Device1. Needs the ability to calculate coordinates with margin of error with less than 50 meters.
2. Updates location at a minimum of 1 Hz.
 | 1. Go to a known coordinate location and compare to the GPS output coordinates.
2. Read out the update speed using ATmega and LCD.
 | Y |
| 3. Digital Compass1. A digital compass with I2C connections for our microcontroller.
 | 1. Verify heading direction accuracy is within 0.5 degrees using microcontroller.
 | Y |
| 4. Motors1. Holds the rotational position against the force of the wind on the parachute with a minimum pull of 15 kg each.
2. Lighter than 8 ounces each.
 | 1. Can verify 1 by attaching a 15 kg weight to a string, and seeing if a motor can pull it.
2. Measure on a scale.
 | Y |
| 5. Parachute1. A ram-air parachute capable of holding the full weight of the payload (max 6 pounds).
2. Includes a left and right brake to toggle the parachute’s yawl.
 | 1. Hold up full payload for a time period equivalent to flight time.
2. Observing the parachute’s glide ratio stays within the golden mark of 5:1 by comparing the drop height to the distance traveled.
 | Y |
| 6. Lipo Battery1. Rechargeable battery supplying enough current for the motors with a 149 mA - 2.344 A draw.
2. Supplies 7.4 volts. Highest operating voltage is 5 volts.
3. Stores at least 1000 mAh of charge.
 | 1. Connect the battery to a sample circuit and verify using a multimeter.
2. Connect the battery to a sample circuit and verify using a multimeter.
3. Leave the battery connected to the sample circuit for 4 hours and confirm voltage stays above 7.4 volts with a voltmeter.
 | Y |
| 7. Battery Pack1. Battery pack that supplies enough current for the circuit with a total 150 mA draw.
 | 1. Connect the battery to a sample circuit and verify using a multimeter.
 | Y |
| 8. Step Down DC-DC Buck Converter1. Supplies 3.3 V or 5 V +/- 5 % from a 7.4 V DC source.
2. Draws current in a range of 0-275 mA
 | a,b. Connect the battery to a sample load circuit and verify with an oscilloscope that the voltage stays within 5% of 3.3 V or 5 V, while also drawing 275 mA. | Y |

#

# **REFERENCES**

[1] Weather.gov. n.d. Weather Balloons. [online] Available at: <https://www.weather.gov/bmx/kidscorner\_weatherballoons#:~:text=Twice%20a%20day%2C%20every%20day,the%20US%20and%20its%20territories> [Accessed 17 September 2020]

[2] Highaltitudescience.com. n.d. Intro To Weather Balloons – High Altitude Science. [online] Available at: <https://www.highaltitudescience.com/pages/intro-to-weather-balloons> [Accessed 17 September 2020].

[3] Hallmark, B., 2019. High Altitude Balloon Crashes In Upshur County. [online] https://www.kltv.com. Available at: <https://www.kltv.com/2019/01/06/high-altitude-balloon-crashes-upshur-county/> [Accessed 17 September 2020]

[4] Globalspec.com. n.d. *Parachutes Selection Guide | Engineering360*. [online] Available at: <https://www.globalspec.com/learnmore/specialized\_industrial\_products/transportation\_products/parachutes> [Accessed 18 September 2020].

[5] bboyho Sparkfun, “sparkfun/SparkFun\_HMC6343\_Arduino\_Library,” *GitHub*, 2020. [Online]. Available: https://github.com/sparkfun/SparkFun\_HMC6343\_Arduino\_Library. [Accessed: 10-Dec-2020].

[6] dherrada Adafruit, “adafruit/Adafruit\_GPS,” *GitHub*, 2020. [Online]. Available: https://github.com/adafruit/Adafruit\_GPS. [Accessed: 10-Dec-2020].

[7] Ieee.org, "IEEE IEEE Code of Ethics", 2016. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 29- Feb- 2016].

[8] Faa.maps.arcgis.com. n.d. Arcgis Web Application. [online] Available at: <https://faa.maps.arcgis.com/apps/webappviewer/index.html?id=9c2e4406710048e19806ebf6a06754ad> [Accessed 18 September 2020].