# Autonomous Position and Altitude Control System for High Altitude Balloons

ECE4012 Senior Design Project

Section L03, High Ball
Project Advisor, Dr. Cohen

Manning, Mathew W, mathew@gatech.edu
Redder, Brandon A, brandonr@gatech.edu
Richardson, David K, drichardson42@gatech.edu, Team Leader
Sanchez, David, dsanchez36@gatech.edu
Scott, Kristine L, kscott43@gatech.edu
Watters, Kyle, kwatters6@gatech.edu

Submitted

2018 December 4

# Executive Summary

The Georgia Institute of Technology in collaboration with the Massachusetts Institute of Technology Lincoln Laboratory developed a low-cost control system for high altitude balloons (HABs). There has been substantial interest in the application of HABs in communication and monitoring of regions not served by traditional infrastructure. Several examples include disaster recovery scenarios, rural areas, and active conflict regions. Many HAB designs suffer from short flight durations, prohibitive cost, or lack position control. The developed system employed a low cost altitude control system to increase flight duration. The payload weighed under six pounds to minimize the regulatory burden and allow the balloon to be operated during inclement weather conditions. In order to keep the design under the weight limit and improve structural stability, a novel cutdown mechanism was developed. To maximize flight duration and regional coverage, an altitude control system was developed using mechanical methods to release helium and ballast as well as a control algorithm. The control algorithm and ballast release have been demonstrated as functioning, but the helium release mechanism will need to be modified for future work. This altitude control could be used in conjunction with wind measurements to provide a position control system in the future. The development cost for the design was $77,117.24 with the cost of hardware per payload being $1,772.94.

# Table of Contents

**Executive Summary** ii

1 **Introduction** 1

1.1 Objective 1
1.2 Motivation 2
1.3 Background 2

2 **Project Description and Goals** 3

3 **Technical Specifications & Verification** 4

4 **Design Approach and Details** 5

4.1 Design Approach 5
4.2 Codes and Standards 17
4.3 Constraints, Alternatives, and Tradeoffs 19

5 **Schedule, Tasks, and Milestones** 20

6 **Final Project Demonstration** 21

7 **Marketing and Cost Analysis** 25

7.1 Marketing Analysis 25
7.2 Cost Analysis 25

8 **Conclusion** 29

9 **Leadership Roles** 30

10 **References** 32

**Appendices** 36

# Autonomous Position and Altitude Control System for High Altitude Balloons

## 1. Introduction

The MITLL High Ball team's directive was to design, launch, and control a high altitude balloon (HAB). The control system utilized helium and ballast release mechanisms in order to adjust the balloons altitude. The High Ball team requested $1000 for the development and testing of a viable HAB control system.

### 1.1 Objective

The goal of this project was to develop an altitude and location control system for a latex HAB to allow long duration flights over a general location. The control algorithm took into account the balloons altitude and vertical velocity to determine when to release helium or ballast. The altitude change due to dropping ballast was estimated using a simulation which employed the balloon's lift force and payload weight. Weather forecasts from the National Oceanic and Atmospheric Administration provided wind speeds at distinct pressure levels allowing for prediction of horizontal movement over time [1]. This data was used to develop a model for the projected flight path under various control algorithms.

## 1.2 **Motivation**

High altitude balloons offer a low cost solution for deploying RF and sensor packages. Traditionally, the flight time of an HAB is limited by the balloon bursting. Altitude control systems can drastically improve the flight duration resulting in a larger variety of applications. The presented control system will allow for prolonged deployment over a specific region.

Project Loon, a balloon system developed by Google, uses tennis court sized balloons to provide cellular service to low population regions. These balloons are capable of multi-month flights with both altitude and location control [2]. Although this system has proven to perform well, it is prohibitively expensive for many applications. A similar position control system deployed on a latex balloon will provide a low cost alternative.

## 1.3 **Background**

As latex HABs rise, the decreasing air pressure causes the balloon to expand until the balloon bursts, ending the flight. Flight times are limited to approximately 3 hours. To overcome this hurdle, zero-pressure and super pressure balloons have been designed to increase flight durations. Both balloon types utilize polyethylene film which has a lower elasticity than latex to minimize expansion [3]. Zero-pressure balloons must drop ballast or vent helium to maintain altitude during diurnal cycles, and they can carry loads of up to 6,000 lbs for flights lasting 7-15 days on average [3]. Super pressure balloons rely on a thicker more rigid film material to limit expansion due to changing temperatures, and they can carry loads of up to 2000 lbs for flights lasting over 100 days [3]. For smaller organizations with instrument packages on the scale of 6 lbs these balloon options are unrealistic.

ValBal is a project out of Stanford that produced an altitude control system for latex balloons. The ValBal system holds the current world record for latex balloon flight duration, 121.5 hours [4]. The current design weighs approximately 2.5 kg without ballast and uses a 1500 g balloon with 9 kg of lift allowing the user to include 5 kg of ballast and instruments [4]. Each payload costs $1000 which makes it viable for a much broader range of users [4].

## 2. Project Description and Goals

The goal was to design a location control system to be used with HABs. Due to time constraints, the design was limited to controlling altitude and relying on the user to manually adjust for varying wind currents. The altitude control system consisted of pressure sensors for determining altitude as well as a mechanical release method for ballast and helium. Motion and location of the balloon was monitored through the use of GPS. An Arduino Mega 2560 was used for data processing and control of mechanical components. The system was deployed in a single payload so that an additional payload may be attached. The second payload could be used for RF or sensing applications. System features include the following:

- Physical control mechanism for helium and ballast release
- Altitude determination through pressure sensor calculations
- Position determination through GPS monitoring
- Manual control of HAB from ground station
- Autonomous HAB flight
- Recoverable payload

# 3. Technical Specifications & Verification

Table 1. Technical Specifications

| Feature | Specifications | Measured Results |
|---|---|---|
| Operating Voltages | 3.3v, 5v | 3.3V, 5V |
| Battery Capacity | 21000 mAh | At least 3.5 hours |
| Payload Weight | 2500 g | 2245.282 g |
| Ballast Weight | 220 g | 340.194 g |
| Operating Altitude | 18 km to 27.4 km | < 4.5 km |
| Ambient Operating Temperature | -60℃ to 40℃ | Functional at -14.7℃ based on altitude |
| Internal Operating Temperature | -35℃ to 40℃ | 3.5℃ to 19.5℃ |
| Operating Pressure | 1.0 to 1000.0 hPa | 589.0 hPa to 994.5 hPa |
| Maximum Recoverable Drop Velocity | 7 m/s | At least 4.27 m/s |
| Water Resistance | IPX4 | Untested |
| Flight Duration | 24 hours | 2.5 Hours |

Table 2. Communications Specifications

| Feature | Specifications | Measured Results |
|---|---|---|
| Latency | 5 seconds (for message <70 bytes) | 47.8 seconds |
| Maximum Usable Distance | Global | Confirmed at 25 miles |

# 4. Design Approach and Details

## 4.1 Design Approach

### 4.1.1 Mechanical Design

***Ballast Release***

Ballast was dispensed via a rotary mechanism from the base of the payload. The dispenser assembly consisted of 3D printed parts which operated similar to a gumball machine. A 3D rendering of the design is shown in Figure 1. A set amount of ballast first moves from a chamber in the payload to the hopper. The hopper is then rotated 90° causing the ballast to fall out of the hopper. This rotation also allows ballast to fill the next section of the hopper to be used for the next release. This mechanism prevents a direct connection from the ballast chamber to the exterior of the payload. The hopper was connected to a servo which controlled the number of rotations.

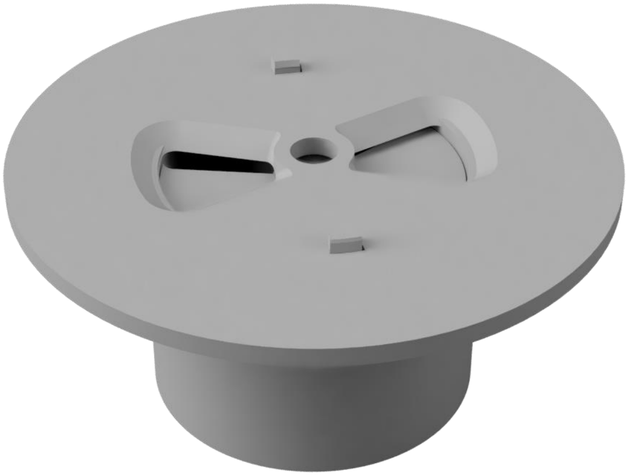


Figure 1. 3D rendering of the ballast dispenser assembly

### ***Helium Venting***

Traditional HABs are connected to their payload by a long cord, but this prevents helium from being released from the balloon. The payload was mechanically connected to the balloon via PVC pipe attached to a mounting bracket on the top of the payload. The balloon was attached to the PVC pipe using zip ties. The PVC is attached to the mounting bracket using an adaptor with two protruding fins that are locked into place by clamps on the mounting bracket. These clamps are held in place using nylon cord. A rubber stopper attached to a spring at the bottom of the bracket blocks the adapter vent to prevent helium from escaping during normal operation. The assembly can be seen in Figure 2. The stopper is attached to a servo located within the payload by a thin rod allowing the servo to open or close the vent. During the flight, a leak was developed. It is uncertain the exact cause, but some potential causes are as follows:

1. The mount and stopper were slightly porous to helium. Helium is a very difficult gas to trap successfully, so it is possible the 3D printed parts were allowing the helium to leak through them.
2. The stopper was not fully sealed against the opening allowing small amounts of gas to escape over the duration of the flight. Once the balloon was attached, it was impossible to check if the stopper was pressed firmly against the opening.
3. There was a non-stopper related hardware failure. This could include the connection between the balloon and the PVC allowing helium to escape, or small cracks in the PVC or mounting bracket.

Further examination and testing is required to determine the exact cause of failure. This is a key flaw in the design that needs to be addressed in future work.

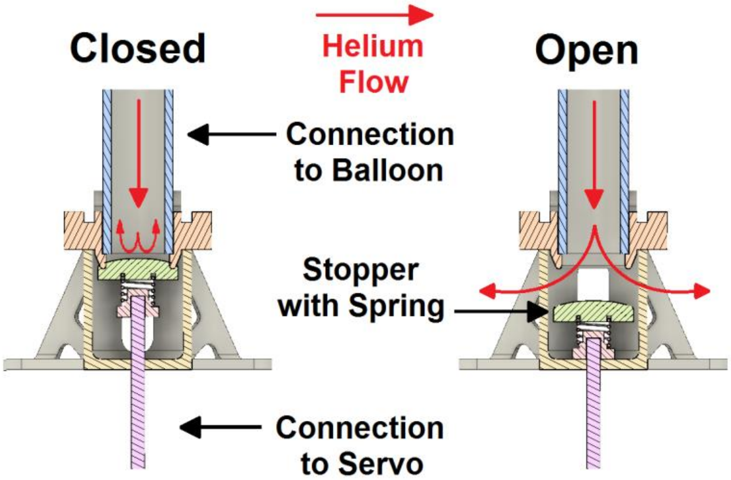


Figure 2. Helium Venting System

### *Cutdown*

The balloon was rigidly connected to the payload through the helium venting assembly described above. This connection was held in place with a small section of nylon cord as discussed above. Nichrome heater wire was wrapped around the cord similarly to a solenoid. This wrapping can be seen in Figure 3.

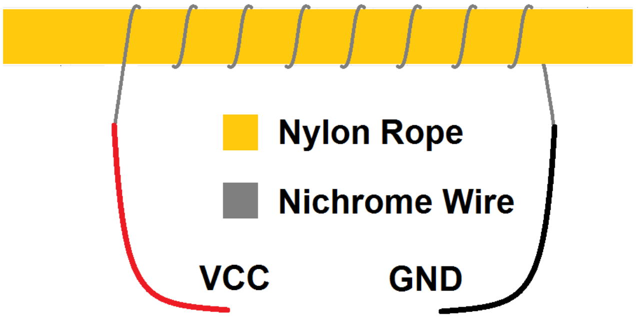


Figure 3. Nichrome wire wrapped around nylon cord

When the cutdown command is given, the controller biases an N-channel mosfet driver energizing the nichrome wire. This bypasses the voltage regulation circuitry and provides a direct connection from the battery to the nichrome. To control current, the driver is connected to a pulse width modulation pin on the controller. This mechanism, shown in Figure 4, was tested prior to launch and shown to work under the weight of the payload while empty. A video of this test can be seen in Appendix A.

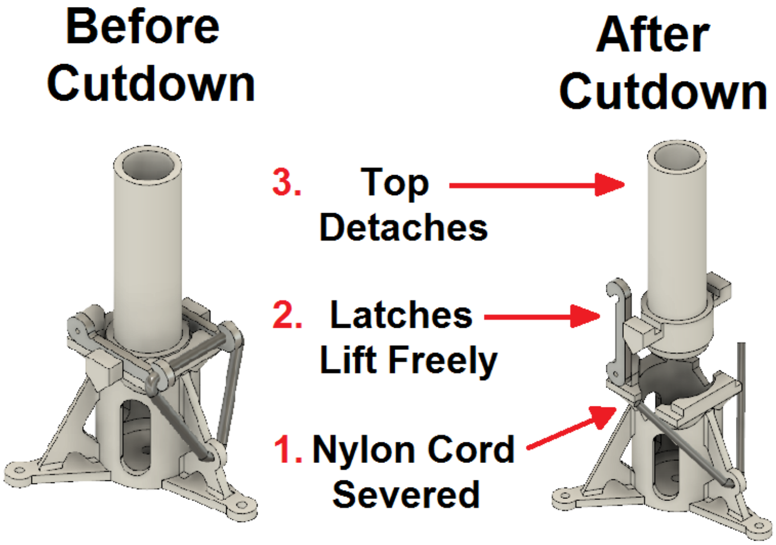


Figure 4. Cutdown Mechanism

***Parachute Deployment***

Due to the direct connect between the payload and the balloon, the traditional method of an inline parachute was impossible. The parachute was instead attached to the helium release mounting bracket. With no method of keeping the parachute above the payload, it will fall below and begin to fill with ballast as it is released. To counteract this, a carbon fiber rod was attached to the top of the payload. The parachute was then held to this rod via a plastic loop on the top of the parachute. This prevents the parachute from trailing under the payload while still allowing the parachute to deploy. A video of the drop test can be seen in Appendix A.

### 4.1.2 Electrical Design

***Communication***

Communications between the balloon and the ground station are facilitated through the Iridium satellite network by the RockBlock 9603, a breakout board utilizing the Iridium 9603 satellite transceiver. The breakout board utilizes four communications pins to interact with the microcontroller. The first two pins are RX and TX connections to allow messages to be sent to and from the device. The other 2 pins allow the RockBlock board to be placed in a low power state to preserve energy and to inform the microcontroller when a new message has arrived. The module uses an external 1621 Mhz patch antenna to communicate with satellites [5]. Outgoing messages are rerouted through nearby satellites before being sent to an Iridium ground station where the message is sent via an email attachment to the user [5]. Incoming messages are uploaded as an HTTP POST to the RockBlock website before being sent back through the same channels to the transceiver.

***Microcontroller***

The microcontroller used was the Arduino Mega 2560 development board (based on the ATmega2560). It was selected due to being available from the previous development team. It fit the communication and use requirements for managing the payload. Using a smaller board is possible if size or weight becomes a significant limiting factor, but at only 37g, the Arduino mega is not a significant source of weight [6]. The microcontroller is connected to the remainder of the electronic components via a custom PCB.

### ***Sensors***

Several sensors were used to provide data to the controller. In addition, some sensors were removed from the design due to technical constraints. The code for reading the data from the sensors is included in Appendix B (Sensors.cpp and Sensor.h). Each of the sensors are summarized below.

#### *GPS Module*

The GPS module used was the SparkFun XA1110 GPS [7]. This module was selected specifically due to its “balloon mode” which allows it to continue functioning above 65,000 feet. This was the primary method of monitoring the position of the balloon.

The GPS altitude is also sent to the ground station and displayed on the graphical user interface (GUI). This altitude was not used in the control algorithm due to the limitations of GPS altitude measurements [8]. Instead the altitude measured from the altimeters was used.

#### *Altimeters (2x)*

The altimeters used were the SparkFun Pressure Sensor Breakout - MS5803-14BA [9]. This sensor was selected due to the ease of integration, and its pressure range of 0-14 bar with a minimum resolution of 0.2 mbar. At altitudes of 70-90 thousand feet, the atmospheric pressure is 17.6-44.9 mbar [10]. Since altitude determination was the basis of the control algorithm, two altimeters were included in the design to mitigate any potential for sensor failure. Due to time constraints, no sensor failure detection was implemented in the code. Instead, the measurements from the two sensors were averaged. Future work should include more robust failure detection which can be done by comparing to the GPS altitude or an expected value.

### *Temperature Sensor - Removed*

The temperature sensor initially specified in the design was the TI LM335Z/NOPB [11]. This sensor was specified due to its range overlap (-40 to 100 °C) with the temperatures expected at altitude. Originally, the temperature sensor was to be used as a control for an internal heating element. To preserve battery and reach the desired temperature the heater was going to be turned on and off. It was ultimately removed from the design as the electronics were estimated to produce enough heat to stay within the operating temperature range. For monitoring of the payload conditions, temperature data was sent to the ground.

During testing, the temperature sensor was reading inaccurate and inconsistent data. This may have been due to a hardware fault earlier in the semester, but the cause of failure was not determined. This led to the removal of the temperature sensor from the design. In its place, the pressure sensors' built in temperature sensors were used which have an accuracy of ±0.8℃ [9]. For internal temperature monitoring, this was determined to be sufficiently accurate; however, if future work needs more accurate temperature measurements, a dedicated temperature sensor should be added back to the design.

### *Inertial Measurement Unit (IMU) - Removed*

The IMU initially specified in the design was the SparkFun 9DoF Razor IMU M0 [12]. This sensor was intended to be used to calculate acceleration both normal and tangential to Earth as well as provide a check for the altitude and position measurements. Unfortunately, the selected IMU presented significant software challenges. While an IMU provides valuable information for position monitoring, future designs should use a different IMU. The issues with the presented IMU are outlined below:

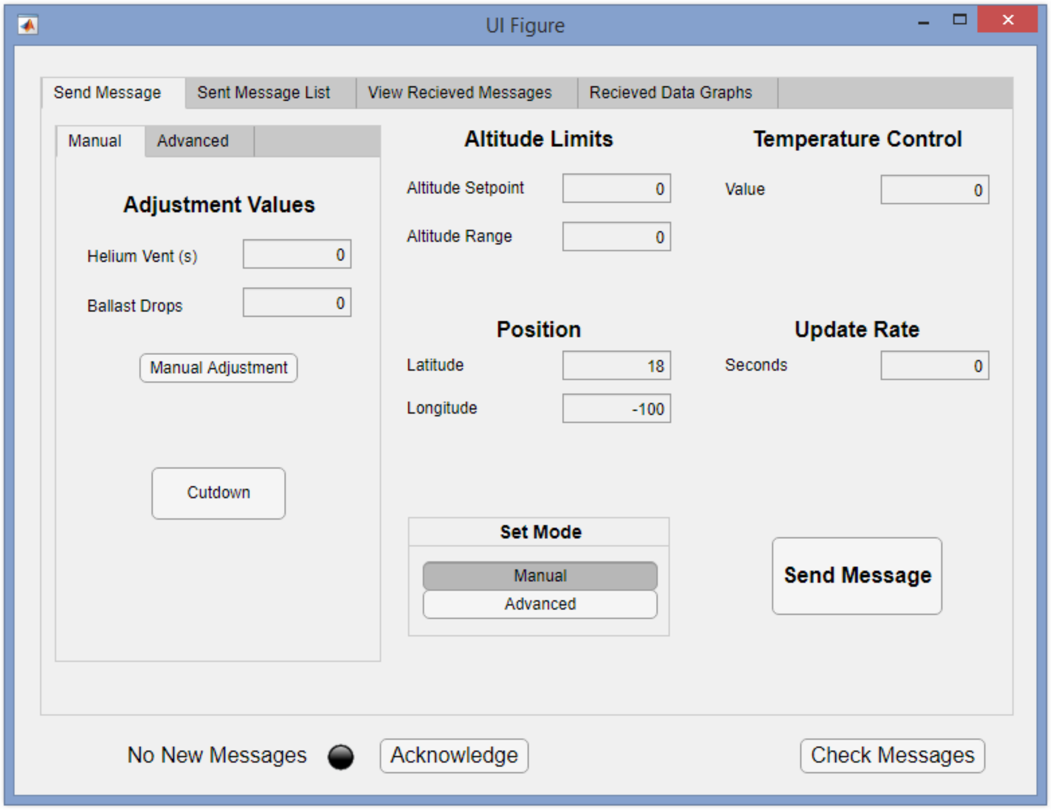
1. The x, y, and z axes for the acceleration measurements differed from what was printed on the board. Per the datasheet, the axis orientation for the compass outputs differs from that of the accelerometer and gyroscope outputs, and the printed axes correspond to the compass. To avoid confusion, the following changes were made to the acceleration outputs to match the axes printed on the board: x acceleration to the y axis, y acceleration to the x axis, and z acceleration to the -z axis.
2. The provided repository for the chip had errors calculating roll, pitch, and yaw. This required changes to the firmware on the internal microprocessor to account for these differences. The current state of the firmware is shown in Appendix C.
3. Communication between the IMU on-board Arduino and the primary Arduino MEGA was unsuccessful. This is what led to the IMU's removal from the final design.

In addition to the above mentioned issues, future work should aim to make the code robust to initial position upon powering on the IMU. The sensor outputs are relative to the initial position rather than absolute measurements. The current design requires the IMU be in a specific orientation when turning on. The IMU must be level with the x axis pointing north for determining the vertical and tangential directions respectively. Future designs should use the acceleration due to gravity to find the true vertical direction and the magnetometer to find the true tangential direction. This will provide more accurate data and prevent any errors due to misalignment when turning on the device.

### 4.1.3 Software

***Ground Station Interface***

The ground station interface consists of a GUI which allows the user to compile new messages, view previously sent messages or received messages, and view real time graphs of received data. As the payload sends messages to the ground, that information is retrieved automatically and the GUI displays an incoming message notification. Each new message contains the balloon's GPS coordinates, altitude, interior temperature, and pressure measurements along with current software settings. Outgoing messages contain specifications for the altitude control algorithm and manual commands for venting helium, dropping ballast, and cutting down the payload. The layout of the main GUI tab can be seen in Figure 5. For brevity, the layout of each page of the GUI and the code for the GUI can be found in Appendix D and E respectively.



Adjustment Values

Helium Vent (s) 0

Ballast Drops 0

Altitude Limits

Altitude Setpoint 0

Altitude Range 0

Temperature Control

Value 0

Position

Latitude 18

Longitude -100

Update Rate

Seconds 0

No New Messages ●

Figure 5. GUI Main Page

### ***Altitude Control***

The altitude control system was separated into three subsystems, the helium release, the ballast release, and the autonomous control loop. The software for the helium and ballast release can be seen in Appendix B (Actuators.cpp and Actuators.h). To control the helium and ballast release, servos were rotated a specified amount to either open/close the helium vent or release a set of ballast. These functions could be triggered either by the autonomous control or by communication from the ground station. The code for the autonomous control loop can also be seen in Appendix B (Control.cpp and Control.h). Both the altitude and vertical velocity were weighted in order to determine when to release helium or ballast. These weights were determined using a simulation. Initially, the algorithm was intended to use the vertical acceleration found by the IMU, but due to the removal of the IMU, the vertical acceleration was also removed. A simulation of the control algorithm showed that the removal of the vertical acceleration had a negligible impact on the effectiveness of the algorithm.

#### *Issues and Considerations*

In its current state, the payload must be launched in manual control mode. This is because the altitude control loop has no guard on releasing ballast if the balloon was just launched. As a proof of concept, this functionality works, but future work should aim to allow for fully autonomous flight from launch to landing. A similar guard should be placed on the helium release while the payload is falling in order to reduce battery consumption.

Future work can also be done to improve the control loop. The weights for the different parameters can be optimized based on measured helium release rates. There may also be additional parameters that play a key role in how the balloon operates such as temperature or wind conditions that are not currently accounted for. Additionally, for longer flights, it may be possible to use the natural warming and cooling of the atmosphere to prolong the flight.

### *Iridium Communication*

Messages to and from the balloon are in the form of a segmented binary vector. Each segment is a predetermined length and represents a different parameter. To limit overall data usage the minimum and maximum values of each parameter were determined along with the resolution of the sensors involved. This allows the parameter to be represented as an integer value with the following formula; $Encoded\ = Round(\frac{Value - Minimum}{Resolution})$. The encoded value will be an unsigned integer with a minimum value of 0 and a known maximum value, meaning only N bits are needed to store the value without losing precision given $Maximum\ \leq\ 2^N$. Figure 6a shows the layout of messages sent to the payload and Figure 6b shows the layout of messages received from the payload. Functions were developed for both the microcontroller and the ground station to encode and decode messages using this predetermined structure. The code for the communication can be seen in Appendix B (Communications.h and Communications.cpp).

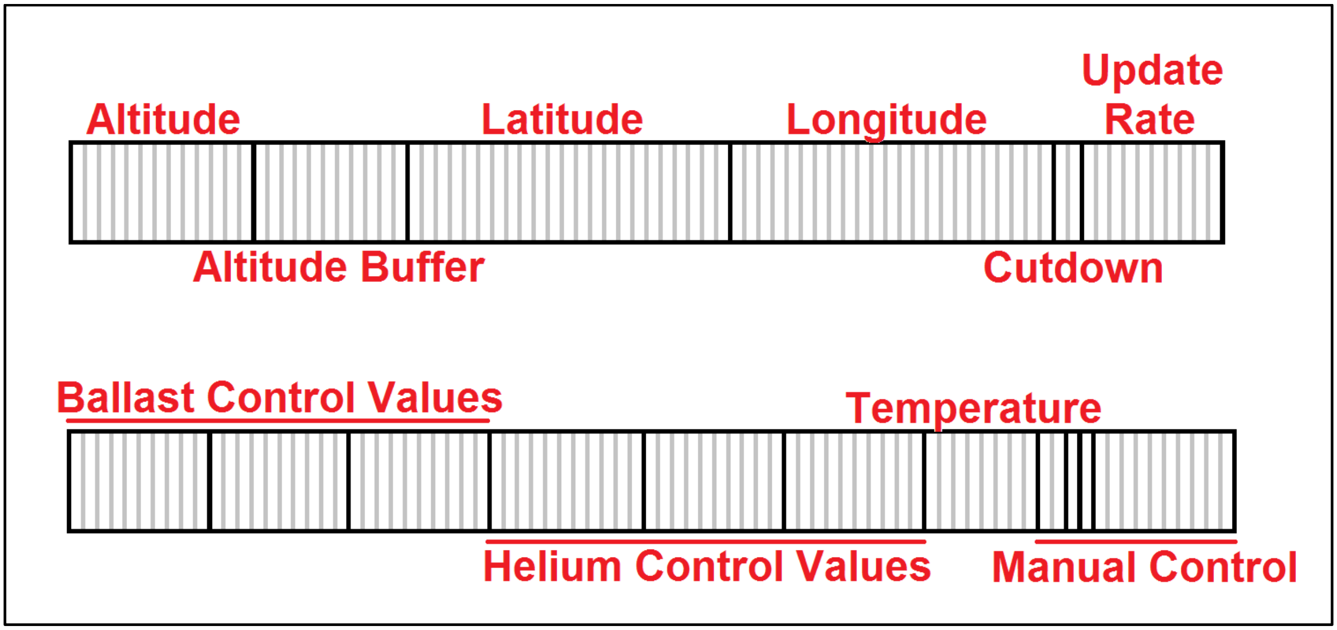


Figure 6a. Message Structure from Ground to Payload

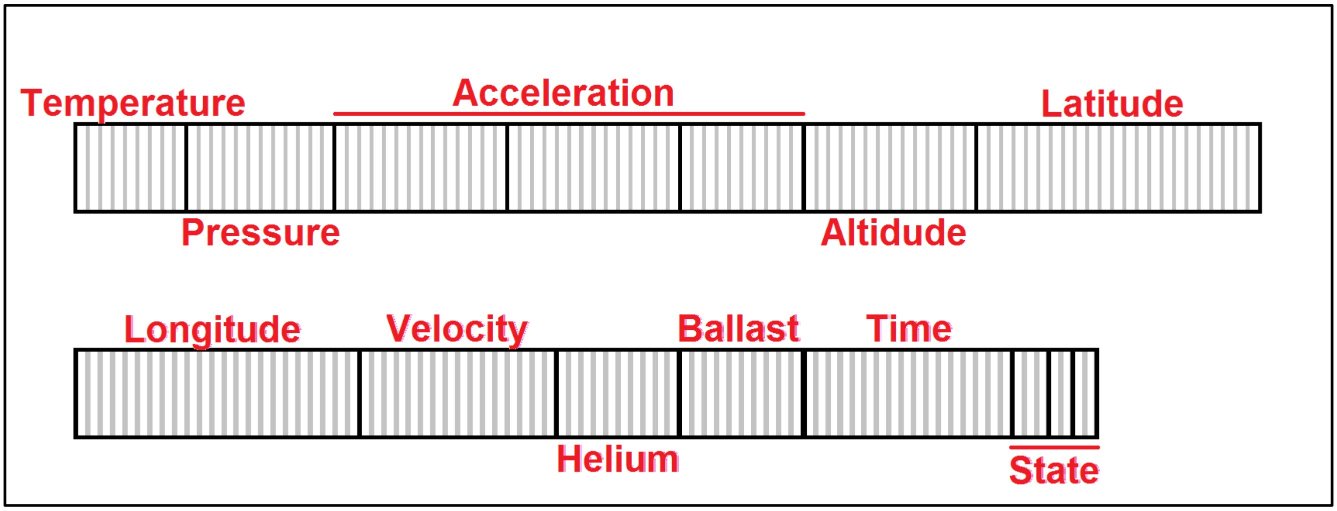


Figure 6b. Message Structure from Payload to Ground

### *Flight Simulation*

Current HAB flight simulation tools operate on the assumption that the balloon rises until it pops then begins falling. This does not provide an accurate model when the payload is actively preventing the balloon from reaching the altitude at which the balloon pops. To circumvent this, a flight simulation tool was developed. This model utilizes weather data from the Global Forecasting System [1] to determine the wind conditions, pressure, humidity, and temperature at a given time and position. The data is used to determine the buoyancy, drag, and gravitational forces acting on the balloons given the current velocity, mass of the payload, and mass of helium in the balloon. These forces allow the balloon's instantaneous acceleration to be approximated over time, even when changes to the system occur during helium venting or dropping ballast. The simulation outputs the ambient weather conditions, position, acceleration and velocity of the payload which can be viewed graphically. The simulation ends when the payload reaches the ground or when a preset time is reached. While the simulation cannot account for all variables, it allows the user to approximate how the balloon's trajectory will change in response to the altitude control system. This was utilized to determine appropriate variables for the control algorithm. The full flight simulation code can be seen in Appendix F.

### *Temperature Control*

Temperature control was omitted in the final design due to the short expected duration of the flight. For short flights, the heat generated by the electrical components provides sufficient heat to keep the interior temperature above the operating range. This may not be the case for longer duration flights. If that is the case, a heating element can be added to the design to maintain the internal temperature.

### *Wind Detection*

Due to time constraints, wind detection was not implemented in the design. This prevents the design from autonomously adjusting position, but it is still possible through manual changes from the ground station. This requires the person at the ground station to determine the wind conditions from the available data and correctly send helium or ballast release commands. Future work should aim to determine wind direction on board to allow for autonomous position control. This may require an IMU to monitor the payload’s orientation and horizontal acceleration.

## 4.2 Codes and Standards

FAA Title 14 - Part 101 - Subpart A (Appendix G) describes the conditions under which unmanned civilian vehicles may be operated within the United States. Part 101.1 specifies that any unmanned free balloon must comply with subpart D with the exception of balloons that:

- Carries a payload package that weighs more than four pounds and has a weight/size ratio of more than three ounces per square inch on any surface of the package, determined by dividing the total weight in ounces of the payload package by the area in square inches of its smallest surface
- Carries a payload package that weighs more than six pounds
- Carries a payload of two or more packages that weighs more than 12 pounds
- Uses a rope or other device for suspension of the payload that requires an impact force of more than 50 pounds to separate the suspended payload from the balloon

The regulations in Subpart D (Appendix H) directly interfere with the intended application of the system, so the payload is designed to comply with the exceptions listed in Part 101.1.

Part 101.7, as it applies to unmanned free balloons, prohibits the operation of the balloon in a manner that may be hazardous to other persons, or their property. It also prohibits dropping any objects, if such an action creates a hazard [13].

FCC Title 47 - Part 97.113 lists prohibited transmissions on amateur radio wavelengths. Relating to this project, no station shall transmit communications for hire or for material compensation, direct or indirect, paid or promised, except as otherwise provided in these rules [14]. This regulation makes amatueur radio an unsuitable means of communication for the system due to the commercial nature of the project.

FCC Title 47 - Part 22.925 prohibits the use of cellular phones while airborne, preventing the system from using a cellular phone to communicate with the payload during flight [15].

The Iridium 9603 transceiver utilized in this design has been granted authorization by the FCC as a Licensed Non-Broadcast Station Transmitter under the conditions listed in FCC Title 47 - Part 25 [16]. The grant of equipment authorization requires that the transmitter be installed to provide 20 cm of separation distance from all persons, and must not be operation in conjunction with any other transmitter within a host device [17].

## 4.3 Constraints, Alternatives, and Tradeoffs

Due to FAA regulations, the payload weight must be less than 6 lbs [13]. This restriction resulted in a tradeoff between component selection and ballast weight. Maximizing the amount of ballast is important for increasing flight duration, but longer flights require a larger power supply. The structure for the payload also contributed to the weight because it had to be sturdy enough to hold the cutoff mechanism for the balloon and provide insulation. Though greatly reduced through iterative design, the weight of this housing was a large portion of the total weight. The current design opted to increase structural rigidity instead of increasing flight duration. This decision was made to improve the chances of payload recovery, and to show a proof of concept for the altitude control mechanisms.

Due to the altitude of HAB flights, the system must be able to sustain operations in temperatures as low as -60° C. Many electronic components fail at this temperature, so special considerations were taken into account when designing the payload. The enclosure used was insulated so that less heat would be lost, and the heat generated by the electronics on the inside helped maintain internal temperature. The heater originally proposed was not used due to reasons discussed in Section 4.1.3. The test flight did not reach an altitude at which the internal temperature could be tested, so it is not currently known if a heater is necessary. The payload remained in the operating temperature during the test flight. The lowest recorded internal temperature was 3.5°C, measured while the lid was open prior to launch. During the flight, the internal temperature steadily rose due to the heat generated by the electrical components.

## 5. Schedule, Tasks, and Milestones

The GANTT and PERT chart are shown below in Appendix I and J respectively.

Brandon Redder performed the majority of the payload hardware design and assembly. He also did the PCB design and fabrication. The hardware assembly was a critical element, and the last thing completed on the critical path. Mathew Manning helped Brandon with hardware assembly and testing. Mathew's tasking played an important role in meeting deadlines and ensuring all elements of the hardware were functioning. David Richardson wrote and maintained the control algorithm and all the sensor interfaces except the IMU. He assisted with the IMU calculations, and worked on the communication between the main Arduino and the IMU. Additionally, he was involved in the testing and debugging of both hardware and software. These tasks were on the critical path and were of medium complexity. The primary source of difficulty was with the IMU communication and software debugging. Kristine Scott wrote the firmware for the IMU to calculate acceleration and orientation values. She also assisted with software, and communication debugging. Although the IMU was removed from the design, it was a difficult task that took a significant amount of time. Her assistance with debugging was also important to the launch of the payload on time. David Sanchez developed the day of launch checklist, as well as determining a location to launch from. While this tasking was fairly simple, it took time and was one of the most important tasks of the project. He also played a significant role in the cutdown code and testing as well as debugging of the software. His testing of the GPS caught a significant error in the code that would have resulted in data loss at high altitude. Kyle Watters wrote both the simulation and communication code for the project. Communication with the payload was one of the most critical features of the design, as without it there would be no way to test any of the other design elements. This in conjunction with the simulation code posed a moderately difficult task that took a considerable amount of time. The main loop was written by David R., Kristine, David S., and Kyle. This was the most critical piece of software in the design but was of relatively low difficulty.

## 6. Final Project Demonstration

The final project demonstration was done in two parts, in flight reporting and ground testing. The following specifications were demonstrated during and immediately prior to the flight:

- The hardware was operating at the specified 3.3V and 5V.
- The time from powering on to payload recovery was 3.5 hours which demonstrated a minimum estimate for battery capacity.
- The payload weight was measured to be 2245.282 g which is below the 2500 g limit. This allowed the ballast weight to exceed its specification at 340.194 g.
- The altitude reached 4.5 km during the flight which falls well below the desired range of 18 km to 27.4 km. This was due to a helium leak likely due to a hardware failure as discussed in the Helium Release subsection of §4.1.1. A plot of the altitude vs time can be seen in Figure 7.
- The minimum ambient operating temperature experienced during flight was -14.7℃. Testing at the minimum specified temperature was impossible to the previously mentioned maximum altitude falling below the expected altitude.
- The operating temperature during the flight ranged between 3.5℃ and 19.5℃ which was within the required range. The temperature was shown to increase over the duration of the flight as shown in Figure 8. This indicates the payload may be able to self regulate temperature without an internal heating element.

- The minimum pressure experienced during the flight was 589.0 hPa. The payload remained operational at this pressure, but pressures below that were unable to be tested. A graph of the pressure vs time can be seen in Figure 9.
- The drop velocity was measured to be 4.27 m/s and the payload was fully functional after recovery. This velocity was calculated from the altitude vs time graph shown above in Figure 7.
- Water resistance was untested as it was deemed unnecessary for the initial prototype
- The flight duration accomplished was 2.5 hours.
- Latency was measured to be 47.8 seconds.
- The usable distance for the payload communication was confirmed to be at least 25 miles from the ground station and was consistent when the payload was in view of satellites. After the payload left the ground, the communication was more consistent and reliable than it was on the ground. This is thought to be because there was less blocking the view of the payload.
- The ballast release mechanism was shown to work after the payload had landed, prior to recovery. It was impossible to see if ballast was released during the flight due to a helium leak, but the post landing tests showed that it was working through remote communication.
- Helium release was shown to have an impact on the buoyancy of the balloon and it was shown that the blocking method was not fully functional from the path that the flight took. The effects of this leak can be seen in Figure 7 where the altitude follows the path expected of a continuous loss in lift.

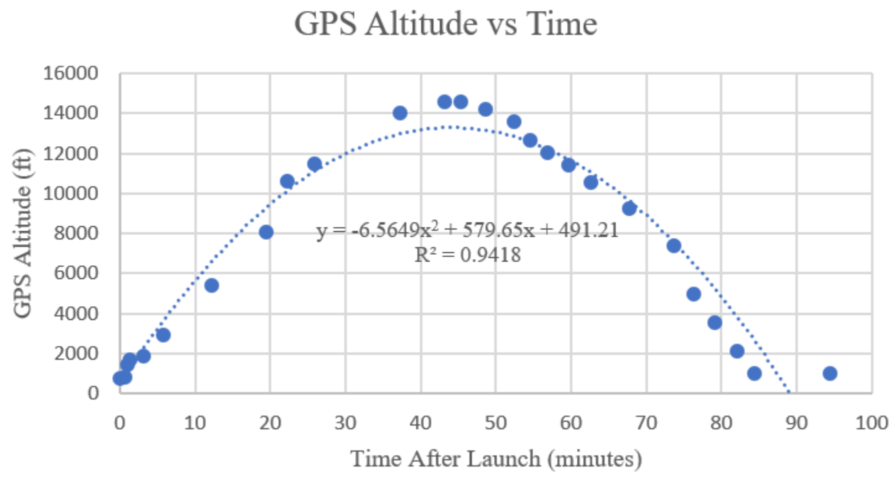


Figure 7. Plot of balloon altitude vs time from the test flight

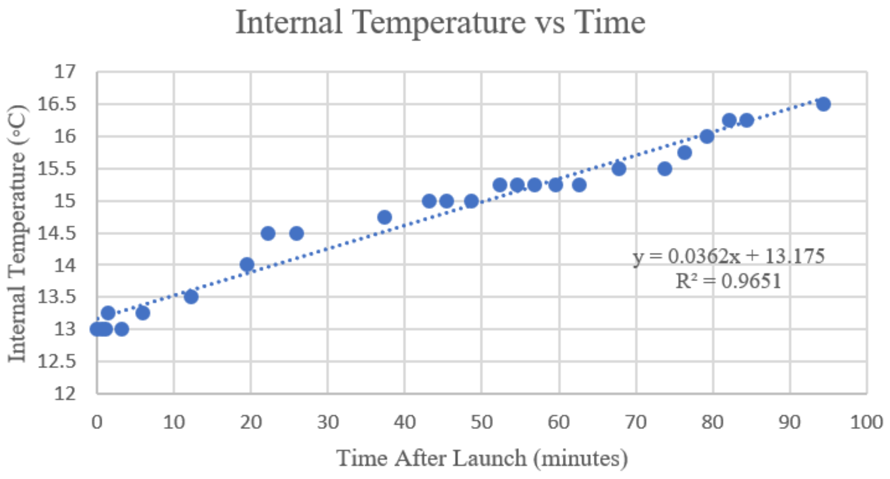


Figure 8. Plot of payload temperature vs time from the test flight

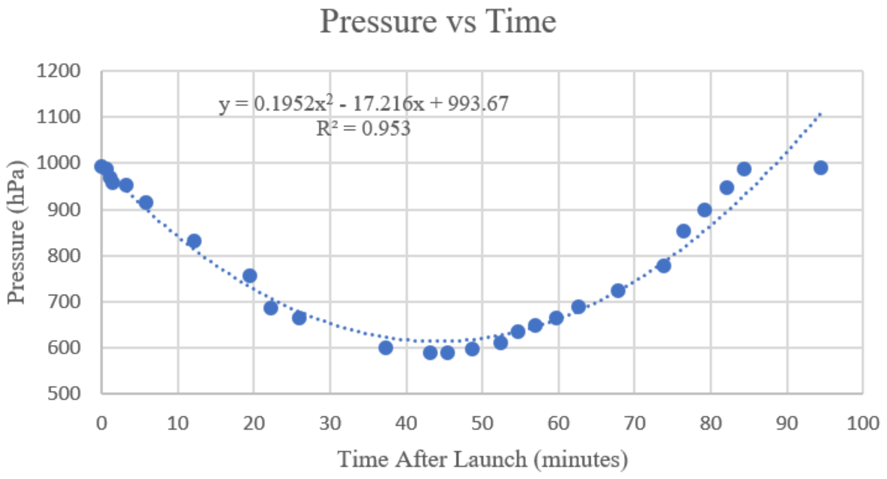


Figure 9. Plot of pressure vs time from the test flight.

Ground testing demonstrated the functionality of the payload and is as follows:

- The parachute deployment method was tested by a four story drop test. A video of the drop can be seen in Appendix A.
- The cutdown method was tested iteratively so that the best wrapping method for nichrome wire could be determined as well as the effects of cutdown on the other electronic components. The method of wrapping can be seen in the Cutdown subsection of §4.1.1 and the current draw was not enough to affect the other components.
- Ballast dispensing was demonstrated both on the ground and during the flight. The dispensing mechanism was tested on the ground and can be seen in a video in Appendix A.
- The helium venting mechanism was demonstrated to work mechanically, but from the launch, it was shown to be ineffective at blocking the release of helium.
- The physical functionality of the payload demonstrated the functionality of the software and communications. The data being taken can be seen in the logs in Appendix K.

# 7. Marketing and Cost Analysis

## 7.1 Marketing Analysis

It is not plausible to launch an HAB project at the same scale as Project Loon unless adequate funding is acquired [2]. Stanford University's ValBal team developed a low cost alternative. The ValBal project utilized less than $1,000 to achieve the longest recorded latex balloon flight. The Stanford team's aircraft used an altitude control system to extend the flight duration [18]. In addition to altitude control, the proposed design aimed to develop a position control system. Though the project did not demonstrate autonomous position control. Changing altitude with known wind directions will allow for directed changes in position. The presented work could be easily built upon to develop an autonomous position control system.

The ValBal design exceeds the weight limit specification which requires users to follow FAA regulations 101.33, 101.35, 101.37, and 101.39 [13] [18]. The proposed design weighed less than six pounds which will allowed for an additional six pound payload without needing to follow the above FAA regulations. Of particular importance, the proposed system may be operated during inclement weather conditions without prior notice to the FAA [13].

## 7.2 Cost Analysis

### 7.2.1 Hardware Costs

The HAB control system consisted of electronic, mechanical, and structural components. The previous design effort purchased some supplies that were reused shown in Table 3. These items are listed separately because they were purchased by the previous team. They were included in the final cost calculations.

Table 3. Previous Supplies

| **Item** | **Estimated Cost** |
|---|---|
| Arduino Mega 2560 | $38.50 |
| 5 ft Diameter Parachute | $50 |
| 1200 g Latex Balloon | $120 |
| RockBLOCK Iridium Module | $250 |
| Servos | $80 |

The additional parts necessary to build the design are listed in Appendix L. The listed components provide enough materials for a single launch. Additional launches will require new balloons and potentially hardware if the payload is unrecoverable.

In addition to hardware, the satellite communication in use requires a data plan with monthly costs and additional charges to purchase credits as shown in Table 4. The credits are needed to send or receive messages with each action costing 1 credit.

Table 5 shows the cost of each of the individual subsystems. An entire bill of materials is included in Appendix L.

Table 4. Communication Data Plan Charges

| | Cost per each | Quantity |
|---|---|---|
| Monthly Subscription Fee | $12.75 | 3 |
| 200 Credits | $25.50 | 1 |
| 1000 Credits<br>(Bundle Discount) | $102.02 | 1 |
| Total Cost | $165.78 | |

Table 5. Subsystem Costs

| **Project Subsystem** | **Cost** |
|---|---|
| Payload | $231.84 |
| Helium | $250 |
| Electrical System | $371.88 |
| Cutdown | $126.71 |
| Communications | $611.07 |
| Ballast | $181.75 |
| **Total** | **$1,772.94** |

### 7.2.2 Development Costs

The design team will consist of six engineers. Table 6 shows the number of work hours per engineer estimated for the duration of the project.

Table 6. Number of Work Hours per Engineer

| **Task** | **Hours** |
|---|---|
| Sensor Integration | 13 |
| Algorithm Development and Implementation | 28 |
| Research | 5 |
| Meetings | 15 |
| Parts Fabrication | 15 |
| Assembly | 20 |
| Ground Testing | 10 |
| Launches and Launch Preparation | 15 |
| Presentation | 2 |
| Reports | 12 |
| **Total** | 135 |

Each engineer is estimated to have a salary of $65,000, per year [19]. The United States Office of Personnel Management defines one work year as 2087 hours [20]. Using these estimates, the hourly rate of a single engineer is $31.15. A fringe benefit rate of 31.9% and an overhead rate of 120% are assumed [21]. The total development costs under these assumptions are shown in Table 7.

Table 7. Total Project Costs

| **Project Section** | **Cost** |
|---|---|
| Parts | $1,607.16 |
| Communication Costs | $165.78 |
| Labor | $25,231.50 |
| Fringe Benefits (31.9% of Labor) | $8,048.85 |
| **Subtotal** | **$35,053.29** |
| Overhead (120% of Subtotal) | $42,063.95 |
| **Total** | **$77,117.24** |

## 8. Conclusion

In its current state, the payload is capable of releasing ballast based on manual or automatic control schemes. The helium release mechanism allowed helium to leak over the duration of the flight. While this does demonstrate that the opening was large enough to vent sufficient gas to reduce altitude, it is a significant design flaw that needs to be addressed in future work. In order for the payload to adequately control altitude, a method for preventing the helium from venting unintentionally must be developed. Testing of the control algorithm was also limited due to time constraints and insufficient test flight data. Since the test flight only reached approximately 4.5 km, there was not enough data gathered to provide an accurate model for how to control the balloon. Future flights would provide valuable feedback on how the current control algorithm performs, as well as where potential improvements can be made.

During the development of the payload, certain errors were made that should be avoided during future work. Of particular note, was the selection of an IMU with a separate on board Arduino, and the milling of the custom PCB. Due to the on board Arduino, the team was unable to get communication between the IMU and the main microcontroller. This made determining vertical acceleration impossible which limited the data available to the control algorithm. The removal of the IMU also prevented checking the altitude and position data against the IMU's acceleration and rotational data as discussed in the initial proposal. In the future, an IMU that connects directly to the main controller would be easier to communicate with and use. When milling the PCB, multiple shorts were developed. This ultimately led to the failure of the GPS and IMU sensors. These components were replaced, but in the future, any PCBs should be examined closely to prevent this from happening. Having the PCB milled by a professional board house would likely reduce the chances of this happening again.

## 9. Leadership Roles

Brandon Redder was responsible for maintains the group's website as the webmaster. He also led hardware assembly. Mathew Manning led the recovery team after the payload landed. He also played a key role in the hardware assembly and testing. As team leader, David Richardson oversaw communication between the team and advisors. He provided direction and made final decisions regarding both hardware and software design elements. He is also the expo coordinator for the team. Kristine Scott led the integration of the different sensor elements. She was also the documentation lead, meaning she was responsible for ensuring all documentation was submitted in a timely manner. David Sanchez oversaw pre-launch preparation. This included launch site selection, pre-launch checklists, and ensuring all necessary items were brought to the launch site. He was also the primary contact for the cutdown software implementation. Kyle Waters oversaw both the simulation and satellite communications. In this role, he developed models for the balloon’s flight path, and used this to help define control parameters. He was also responsible for all communication between the payload and ground station.

# 10. References

[1] National Oceanic and Atmospheric Administration, “Global Forecast System (GFS),” July 14, 2018. [Online]. Available: https://www.ncdc.noaa.gov/data-access/model-data/model-datasets/global-forcast-system-gfs [Accessed: July 20, 2018].

[2] X Development LLC, “Technology,” *Project Loon*, 2018, [Online]. Available: https://loon.co/technology/ [Accessed: Jun. 8, 2018].

[3] B. Dunbar, H. Zell, “Types of Balloons”, *National Aeronautics and Space Administration*, Aug. 3, 2017 [Online]. Available: https://www.nasa.gov/scientific-balloons/types-of-balloons [Accessed July 19, 2018].

[4] Andrey Sushko, Aria Tedjarati, Joan Creus-Costa, Kai Marshland, Paige Brown, Davy Ragland, John Dean, Sasha Maldonado, John M. Pauly, “Advancements in low-cost, long endurance, altitude controlled latex balloons (ValBal),” In Proc. *2018 IEEE Aerospace Conference*, 2017 pp. 1-9. [Online]. Available: https://ieeexplore.ieee.org/document/8396588/ [Accessed: July 20, 2018].

[5] Rock Seven Location Technology, “RockBlock 9603,” RockBlock 9603 datasheet, [Online]. Available: http://www.rock7mobile.com/downloads/RockBLOCK-9603-Product-Information-Sheet.pdf [Accessed: November 30, 2018]

[6] Arduino, [Online]. Available: https://store.arduino.cc/usa/arduino-mega-2560-rev3 [Accessed: December 2, 2018]

[7] Sparkfun Electronics, “SparkFun GPS Breakout - XA1110 (Qwiic) Hookup Guide,” [Online]. Available: https://learn.sparkfun.com/tutorials/sparkfun-gps-breakout---xa1110-qwiic-hookup-guide?_ga=2.210384540.827711392.1543778489-836632187.1455911519 [Accessed: December 2, 2018]

[8] M. Graham, “GPS Versus Barometric Altitude: The Definitive Answer,” xcmag.com, Jul. 23, 2011. [Online]. Available: http://xcmag.com/news/gps-versus-barometric-altitude-the-definitiveanswer/. [Accessed Dec. 3, 2018].

[9] TE Connectivity, “Miniature 14 bar Module,” MS5803-14BA datasheet, Aug. 2017 [Online]. Available: https://www.te.com/commerce/DocumentDelivery/DDEController?Action=showdoc&DocId=Data+Sheet%7FMS5803-14BA%7FB3%7Fpdf%7FEnglish%7FENG_DS_MS5803-14BA_B3.pdf%7FCAT-BLPS0013 [Accessed: November 30, 2018]

[10] AVS, “Atmospheric Pressure at Different Altitudes,” avs.org. [Online]. Available: https://www.avs.org/AVS/files/c7/c7edaedb-95b2-438f-adfb-36de54f87b9e.pdf [Accessed Dec. 3, 2018].

[11] Texas Instruments, “LMx35, LMx35A Precision Temperature Sensors,” LM135, LM135A, LM235, LM235A, LM335, LM335A datasheet, May 1999 [Revised Feb. 2015]. Available: http://www.ti.com/lit/ds/symlink/lm335.pdf [Accessed: November 30, 2018]

[12] Sparkfun Electronics, “9DoF Razor IMU M0 Hookup Guide,” [Online]. Available: https://learn.sparkfun.com/tutorials/9dof-razor-imu-m0-hookup-guide?_ga=2.238172809.827711392.1543778489-836632187.1455911519 [Accessed: December 2, 2018]

[13] Code of Federal Regulations Title 14 Part 101 (14CFR101), Moored Balloons, Kites, Amateur Rockets, Unmanned Free Balloons, and Certain Model Aircraft [Online]. Available: https://ecfr.io/Title-14/pt14.2.101#sp14.2.101 [Accessed: July 19, 2018].

[14] Code of Federal Regulations Title 47 Part 97.113, Amateur Radio Service, Prohibited Transmissions [Online]. Available: https://ecfr.io/Title-47/se47.5.97_1113 [Accessed: July 19, 2018].

[15] Code of Federal Regulations Title 47 Part 22.925, Public Mobile Services, Prohibition on Airborne Operation of Cellular Telephones [Online]. Available: https://ecfr.io/Title-47/pt47.2.22#se47.2.22_1925 [Accessed: July 19, 2018].

[16] Code of Federal Regulations Title 47 Part 25, Satellite Communications [Online]. Available: https://ecfr.io/Title-47/pt47.2.25 [Accessed: July 19, 2018].

[17] Iridium Communications Inc, “Iridium 9603 Short Burst Data Modem: Declaration of Conformity,” *Iridium Short Burst Data (SBD)*, July 19, 2018. [Online]. Available: https://www.iridium.com/file/24245/ [Accessed: July 20, 2018].

[18] A. Sushko, A. Tedjarati, J. Crues-Costa, S. Maldonado, K. Marshland, and M. Pavone, “Low Cost, High Endurance, Altitude-Controlled Latex Balloon for Near-Space Research (ValBal),” In Proc. *IEEE Aerospace Conference ’03*, 2017, pp. 1-9. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7943912&isnumber=7943554 [Accessed: July 20, 2018].

[19] Georgia Tech School of Electrical and Computer Engineering, “What is ECE at Georgia Tech,” *Georgia Institute of Technology*, 2018. para. 11, Apr. 4, 2018 [Online]. Available: https://www.ece.gatech.edu/what-ece-georgia-tech [Accessed July 19, 2018].

[20] US Office of Personnel Management, “ Pay & Leave: Pay Administration,” *OPM.gov*, para. 1. [Online]. Available: https://www.opm.gov/policy-data-oversight/pay-leave/pay-administration/fact-sheets/computing-hourly-rates-of-pay-using-the-2087-hour-divisor/ [Accessed: July 19, 2018].

[21] Georgia Tech Office of Sponsored Programs, “Facilities & Administrative Rates,” *Georgia Tech Office of Sponsored Programs*, para. 3, July 6, 2018. [Online]. Available: http://www.osp.gatech.edu/rates [Accessed: July 19, 2018].

# Appendix A

This appendix contains a collection of videos showing testing procedure for the cutdown mechanism as well as the ballast and helium release system.

The videos are linked from our website located at:

www.HighBallSeniorDesign.com/project_documentation.html

They can also be directly accessed from here:

https://photos.app.goo.gl/7F5YTKmhSC1EK4um7

# Appendix B

This appendix contains the source code for the main payload Arduino Mega.

The code is located on a public GitHub linked from our site:

www.HighBallSeniorDesign.com/project_documentation.html

It can also be accessed directly here:

https://github.com/BrandonRedder/HighBall/tree/master/Software/Control%20Payload/Control%20Payload%200-1/Payload_Main

# Appendix C

This appendix contains the source code for the IMU’s Arduino SAMD.

The code is located on a public GitHub linked from our site:

www.HighBallSeniorDesign.com/project_documentation.html

It can also be accessed directly here:

https://github.com/BrandonRedder/HighBall/tree/master/Software/IMU

## Appendix D

This appendix contains screenshot of each page of the communication and control GUI.

The screenshots are stored on in a Google Photos album linked from our site:

www.HighBallSeniorDesign.com/project_documentation.html

It can also be accessed directly here:

https://photos.app.goo.gl/FRyCTfw28V91FpCCA

# Appendix E

This appendix contains source code for the GUI application.

The code is located on a public GitHub linked from our site:

www.HighBallSeniorDesign.com/project_documentation.html

It can also be accessed directly here:

https://github.com/BrandonRedder/HighBall/tree/master/Software/Ground%20Communication

# Appendix F

This appendix contains the source code for the controls simulation.

The code is located on a public GitHub linked from our site:

www.HighBallSeniorDesign.com/project_documentation.html

It can also be accessed directly here:

https://github.com/BrandonRedder/HighBall/blob/master/Software/Ground%20Communication/Flight_Simulation.m

# Appendix G

This appendix contains FAA regulations pertaining to balloon flight.

The regulations are maintained by Cornell and linked from our site:

www.HighBallSeniorDesign.com/project_documentation.html

They can also be accessed directly here:

https://www.law.cornell.edu/cfr/text/14/part-101/subpart-A

# Appendix H

This appendix contains FAA regulations pertaining to balloon flight.

The regulations are maintained by Cornell and linked from our site:

www.HighBallSeniorDesign.com/project_documentation.html

They can also be accessed directly here:

https://www.law.cornell.edu/cfr/text/14/part-101/subpart-D

# Appendix I

This appendix contains a Gantt chart showing the project timeline and major milestones.

The chart is linked from our site: www.HighBallSeniorDesign.com/project_documentation.html

They can also be accessed directly here: www.HighBallSeniorDesign.com/documentation/Gantt.pdf

# Appendix J

This appendix contains a PERT chart showing the project timeline and major milestones.

The chart is linked from our site: www.HighBallSeniorDesign.com/project_documentation.html

They can also be accessed directly here: www.HighBallSeniorDesign.com/documentation/PERT.pdf

# Appendix K

This appendix contains sensor data from our test flight.

The data is stored on a public GitHub linked from our site:

www.HighBallSeniorDesign.com/project_documentation.html

It can also be accessed directly here:

https://github.com/BrandonRedder/HighBall/blob/master/Final%20Documentation/FlightData.xlsx

# Appendix L

This appendix contains a final bill of materials for the project.

The BOM is stored on linked from our site:

www.HighBallSeniorDesign.com/project_documentation.html

It can also be accessed directly here:

https://drive.google.com/open?id=1pH3nLefY7gUAFXORD74aRVZBpK-clhbAkCagKy8D83Y