P82: Stratospheric Exploration Vehicle

Patrick Mojsak

EE 4349: Senior Design

Dr. Wetz

03/27/2012

Table of Contents

1. Abstract………………………………………………………………………………………..1
2. Background……………………………………………………………………………………1
3. Introduction……………………………………………………………………………………1
4. Technical Work………………………………………………………………………………..2
   1. Stratospheric Exploration Vehicle Overview…………………………………………..2
   2. Hardware………………………………………………………………………………..5
      1. Microcontroller…………………………………………………………………5
      2. Sensors………………………………………………………………………….8
      3. Miscellaneous Hardware……………………………………………………...10
      4. Hardware Performance………………………………………………………..11
   3. Software……………………………………………………………………………….13
      1. Language……………………………………………………………………...13
      2. Files Utilized………………………………………………………………….13
   4. Results………………………………………………………………………………....17
      1. Balloon Launch…………………………………………………………….....17
      2. Data – General………………………………………………………………...17
      3. Data – First Segment………………………………………………………….17
      4. Data – Second Segment……………………………………………………….19
      5. Data – Third Segment…………………………………………………………21
      6. Data – Fourth Segment………………………………………………………..23
      7. Data – Temperature Measurements…………………………………………...24
5. Conclusion…………………………………………………………………………………...25
6. References……………………………………………………………………………………26
7. Appendices…………………………………………………………………………………...27
   1. Appendix 1: main.spin File Code……………………………………………………….27
   2. Appendix 2: temp\_sensor\_main.spin File Code………………………………………...28
   3. Appendix 3: Third Segment Excel Data Results………………………………………..29

**1. Abstract**

This paper is the final report for the Stratospheric Exploration Vehicle (SEV) P82 in EE 4349 Senior Design (SD) at the University of Texas at Arlington (UTA). Most amateur weather balloon designs are prone to many problems. The SEV is a vehicle designed to acquire data of the Earth’s atmosphere as well as to base improvements on in future amateur weather balloon designs.

**2. Background**

The study of the Earth’s atmosphere is critical for many reasons including weather forecasting, climatology, and atmospheric chemistry. This is done by various aerial vehicles, the most basic being that of a weather balloon. Weather balloons used by the National Weather Service (NWS), and deployed on a regular basis, are also called radiosondes, due to the fact that they transmit radio frequency (RF) signals of data that they acquire back to ground stations. These are typically only intended for one-time use since they are difficult to recover.

The launching of weather balloons has also become a popular hobby, especially among amateur HAM radio users. Reasons for launching balloons could range from a fun scavenger hunt to an attempt to break a world record. The current world record for the highest altitude reached by a weather balloon is approximately 55 km held by the Japan Aerospace Exploration Agency (JAXA). A typical amateur weather balloon will carry an instrument package containing a GPS receiver as well as a transmitter to relay its position so that the user can locate the balloon once it has landed. Other miscellaneous items, such as eggs, may also be sent up in the balloon to observe the effects of reduced air temperature and pressure in a near-space environment for things such as science projects.

Due to the nature of amateur weather ballooning, there are no set standards for construction, and balloons are prone to many problems. Parachutes not activating, position transmissions ceasing, and lost packages are not uncommon. In most cases the cause for such malfunctions cannot be determined. Although the SEV is essentially a reinvention of the wheel, it does have a unique purpose in that it collects data to help analyze and determine such problems.

**3. Introduction**

The goal of this senior design project is to design and construct an instrument package for an amateur weather balloon that will both acquire the necessary data on which to base improvements on in future amateur weather balloon designs, as well as other miscellaneous atmospheric data. A weather balloon’s flight can be divided into five phases: launch, ascent, burst, descent, and landing. The launch, ascent, descent, and landing phases are self-explanatory; however, the burst phase may not seem so obvious. As a balloon ascends to a higher altitude, the ambient air pressure drops, however, the gas pressure within the balloon remains constant. As a result, the balloon expands to a larger volume than at its launch site, eventually to a point where the material cannot stretch any further and bursts. Once this occurs, the balloon begins its descent.

Each phase of the balloon’s flight poses its own unique challenges. Of these, the launch phase contains the least amount of problems and unknowns due to the fact that the balloon is under direct observation. Problems that may occur include separation of the balloon from the package, especially in the case where multiple balloons are used. During the ascent phase, rotation of the balloon’s instrument package is not uncommon, even at relatively high rates. As the balloon ascends to higher altitudes in the stratosphere, low temperatures begin to cause a big problem for any electrical components. During the burst phase, tumbling of the package in a low-gravity environment can be unpredictable and problematic especially for proper parachute deployment. During the descent phase, position transmission is critical for locating the package, and may have ceased due to the low temperatures encountered at high altitudes. Lastly, an improper parachute can be problematic during the landing phase due to the possibility of a hard landing and resulting damage. Activation of certain devices during the landing phase are also critical to assist in locating the package. These can include aural (a sonic annunciator) or visual (a strobe) devices.

For this project, several pieces of data were acquired to observe these common problems. This includes measurement of the rotational velocity of the package (via a MEMS gyroscope), measurement of the package’s attitude (bank and roll angle) and G-force along all three axes (both via an accelerometer), and temperature measurement at various locations within the package. Other data acquired includes ambient pressure, barometric altitude, local terrestrial magnetic field strength, and battery pack voltage. All of this data was logged onto a micro SD card to be analyzed upon retrieving the vehicle.

**4. Technical Work**

**4.1 Stratospheric Exploration Vehicle Overview**

The following are the components that make up the Stratospheric Exploration Vehicle:

**Balloon**: A latex, hydrogen-filled balloon was used to ascend the package to its maximum altitude. The balloon was approximately 6 feet in diameter at the surface. However, at lower pressures at higher altitudes, it is expected to expand to approximately 20 feet in diameter prior to burst. Attachment and filling was done at the launch site with the assistance of our sponsor. Hydrogen was selected due to its lower cost as well as the fact that it is even lighter than helium.

**Parachute**: The parachute was used to allow the instrument package to safely descend at a reasonable rate. The release mechanism was relatively simple and automatic by simply attaching the parachute directly to the balloon’s support string. Upon balloon burst, the parachute takes over as the instrument package begins to fall.

**Support string & VHF J-pole antenna:** The support string attaches the balloon, parachute, and instrument package together. It is approximately 15-20 ft. in length. Coincident to the support string is VHF J-pole antenna. The antenna was used to transmit the vehicle’s GPS position which was acquired via circuitry within the instrument package. The antenna transmits on HAM frequency of 144.39 MHz omni-directionally to various repeater stations. This data was then encoded to the internet in a way that one could view the vehicle’s position graphically on a website.

**Lego astronaut observation platform:** A plank of plywood, approximately 2.5 ft. in length, was used to station a Lego astronaut with the U.S. and Texas flags in front of one of the cameras. The Texas flag was accidentally glued upside down in the rigging process. In January of 2012, two Canadian teens launched their own weather balloon with a Lego man holding the Canadian flag positioned in front of a camera. This balloon reached an altitude of 80,000 ft. The team felt that it was their duty as Americans to not let the Canadians go unchecked. The figure below shows both the U.S. Lego astronaut as well as the Canadian Lego man.

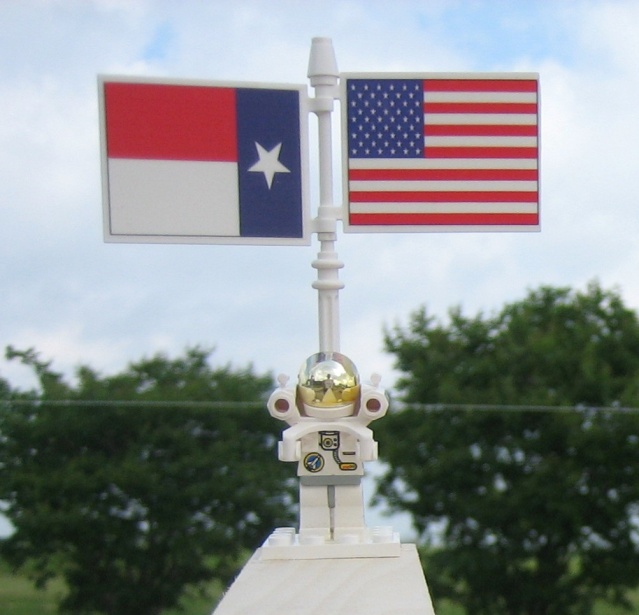


Figure 1: U.S. Lego Astronaut & Canadian Lego Man Respectively

**Instrument package housing**: The instrument package housing consists of a Styrofoam bucket to contain all hardware, sensors, cameras, etc. pertinent to the instrument package. The figure below outlines the entire vehicle as a whole, as well as the layout of the instrument package.

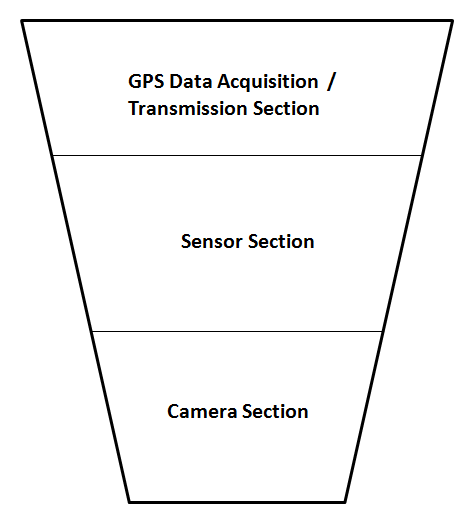


Figure 2: Overall Stratospheric Exploration Vehicle Layout & Instrument Package Housing Layout Respectively

As detailed in the figure on the previous page, the instrument package consists of three sections: the GPS data acquisition/transmission section, the sensor section, and the camera section. The GPS data acquisition/transmission section was constructed by our sponsor, and consisted of the necessary hardware to acquire and re-transmit the vehicle’s GPS position. Acquisition was done via a GPS module with the antenna mounted near the top of the instrument package. Position data was then re-transmitted via the VHF J-pole antenna as described previously.

The sensor section was constructed by the senior design team, as this was the task for the project. The sensor section consisted of two printed circuit boards (PCBs), each with their own respective circuits, as well as battery pack. The sensor section is outlined in greater detail below.

* **Primary sensor platform circuit**: This circuit, self-contained on its own PCB, contains all the necessary circuitry to run and acquire data from the following sensors:
  + **Pressure sensor**
  + **Accelerometer**
  + **MEMS gyroscope**
  + **Magnetometer**
  + **Battery voltage measurement**
* **Temperature sensor platform circuit**: This circuit, once again on its own PCB, contains all the necessary circuitry to run and acquire data from six temperature sensors. These sensors were placed in different locations in the instrument package as follows:
  + **GPS data transmission/acquisition section**
  + **Sensor section** (two were used)
  + **Outside air temperature** (two were used)
  + **Camera section**

The two PCBs were then attached back to back to save on space, with a piece of cardboard placed in between. External wires then connected the two PCBs to the battery pack.

Lastly, the camera section contained two cameras, with a piece of Styrofoam placed on top to provide separation between the camera and sensor sections. One camera was a still frame camera that took photos in 30 second intervals. The Lego astronaut observation platform was placed in front of this camera. This was done since the still frame camera contained enough memory so as to take pictures for the duration of the entire two-hour flight. This was critical since photographic evidence of the U.S. Lego astronaut enduring all phases of the flight was needed. The second camera recorded video in high definition (HD), but only had enough memory to record approximately one hour of video. Both cameras were angled just slightly below the local horizontal so as to view the Earth’s terrain as the vehicle ascended. On the following page, photographs of the different sections of the instrument package are shown, as well as the overall instrument package itself.

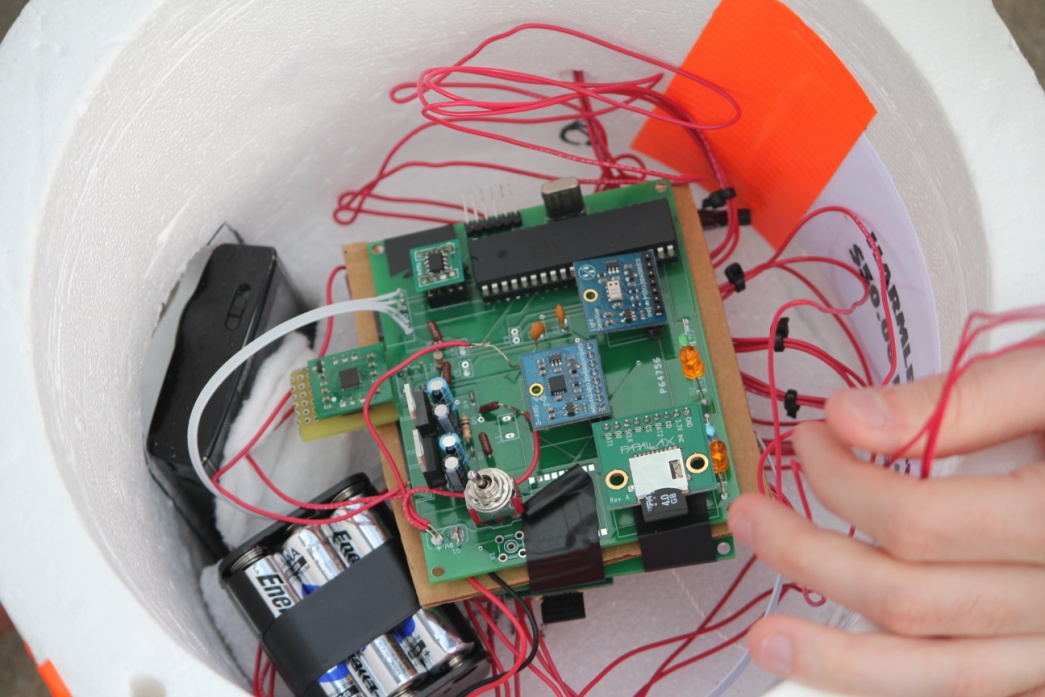
 

Figure 3: Instrument Package Overview: Top left – Camera Section, Top Right – Sensor Section, Bottom Left – GPS Data A/T Section, Bottom Right – Overall Instrument Package

**4.2 Hardware**

**4.2.1 Microcontroller**

The microcontroller of choice for this project was the Parallax P8X32A Propeller Microcontroller. This was selected due to its multi-core processing capability (8 processors, or “cogs” to choose from), the high-level language in which it utilizes, as well as its compatibility with a wide range of sensors that cater to this microcontroller. The Propeller microcontroller is also relatively large when compared to other microcontrollers, and has 40 pins, 32 of which are designated for general I/O.

Two separate microcontrollers were used in this project for the sensor section, one on each PCB. This was done since temperature data took far too long to log (approximately five seconds for each reading, resulting in thirty seconds to run through all six temperature sensors) and would throw off time-sensitive data such as accelerometer or gyroscope data, which needed at least two to three samples per second to provide accurate data.

The table below outlines the pin allocation for the microcontroller utilized on the primary sensor platform circuit.

|  |  |  |
| --- | --- | --- |
| Pin Number | Pin Designation | Usage |
| 1 | **P0** | **SCL** connection for **magnetometer**. |
| 2 | **P1** | **SDA** connection for **magnetometer**. |
| 3 | **P2** | **Battery voltage** Sigma-Delta **input** connection. |
| 4 | **P3** | **Battery voltage** Sigma-Delta **feedback** connection. |
| 5 | **P4** | **SDA** connection for **altimeter**. |
| 6 | **P5** | **SCL** connection for **altimeter**. |
| 7 | **P6** | NC |
| 8 | **P7** | NC |
| 9 | **VSS** | **Ground**. |
| 10 | **BOEn** | **Ground**. |
| 11 | **RESn** | **Prop Plug RST.** |
| 12 | **VDD** | Connected to source voltage of **3.3V**. |
| 13 | **P8** | NC |
| 14 | **P9** | NC |
| 15 | **P10** | **DO** connection for micro **SD card adapter**. |
| 16 | **P11** | **CLK** connection for micro **SD card adapter**. |
| 17 | **P12** | **DI** connection for micro **SD card adapter**. |
| 18 | **P13** | **CS** connection for micro **SD card adapter**. |
| 19 | **P14** | **Green LED** connection. |
| 20 | **P15** | **Red LED** connection. |
| 21 | **P16** | **CS** connection for **accelerometer**. |
| 22 | **P17** | **DIO** connection for **accelerometer**. |
| 23 | **P18** | **CLK** connection for **accelerometer**. |
| 24 | **P19** | **Yellow LED** connection. |
| 25 | **P20** | **Blue LED** connection. |
| 26 | **P21** | NC |
| 27 | **P22** | NC |
| 28 | **P23** | NC |
| 29 | **VSS** | **Ground**. |
| 30 | **XI** | Input for **crystal oscillator**. |
| 31 | **XO** | Output for **crystal oscillator**. |
| 32 | **VDD** | Connected to source voltage of **3.3V**. |
| 33 | **P24** | NC |
| 34 | **P25** | NC |
| 35 | **P26** | NC |
| 36 | **P27** | NC |
| 37 | **P28** | **SCL** connection for **EEPROM**. |
| 38 | **P29** | **SDA** connection for **EEPROM**. |
| 39 | **P30** | **Transmit** data via **Prop Plug** to programming PC. |
| 40 | **P31** | **Receive** data via **Prop Plug** from programming PC. |

Table 1: Microcontroller Pin Allocation for the Primary Sensor Platform Circuit

The table below outlines the pin allocation for the microcontroller utilized in the temperature sensor platform circuit.

|  |  |  |
| --- | --- | --- |
| Pin Number | Pin Designation | Usage |
| 1 | **P0** | **Blue LED** connection. |
| 2 | **P1** | NC |
| 3 | **P2** | NC |
| 4 | **P3** | **Temperature sensor #5.** |
| 5 | **P4** | NC |
| 6 | **P5** | NC |
| 7 | **P6** | NC |
| 8 | **P7** | **PCB temperature sensor.** |
| 9 | **VSS** | **Ground**. |
| 10 | **BOEn** | **Ground**. |
| 11 | **RESn** | **Prop Plug RST.** |
| 12 | **VDD** | Connected to source voltage of **3.3V**. |
| 13 | **P8** | NC |
| 14 | **P9** | **Temperature sensor #1.** |
| 15 | **P10** | **Temperature sensor #4.** |
| 16 | **P11** | NC |
| 17 | **P12** | NC |
| 18 | **P13** | **Red LED** connection. |
| 19 | **P14** | **Green LED** connection. |
| 20 | **P15** | **Temperature sensor #3.** |
| 21 | **P16** | **DO** connection for **SD card adapter.** |
| 22 | **P17** | **SCLK** connection for SD card adapter. |
| 23 | **P18** | **DI** connection for SD card adapter. |
| 24 | **P19** | **CS** connection for SD card adapter. |
| 25 | **P20** | **Temperature sensor #2.** |
| 26 | **P21** | **Yellow LED** connection. |
| 27 | **P22** | NC |
| 28 | **P23** | NC |
| 29 | **VSS** | **Ground**. |
| 30 | **XI** | Input for **crystal oscillator**. |
| 31 | **XO** | Output for **crystal oscillator**. |
| 32 | **VDD** | Connected to source voltage of **3.3V**. |
| 33 | **P24** | NC |
| 34 | **P25** | NC |
| 35 | **P26** | NC |
| 36 | **P27** | NC |
| 37 | **P28** | **SCL** connection for **EEPROM**. |
| 38 | **P29** | **SDA** connection for **EEPROM**. |
| 39 | **P30** | **Transmit** data via **Prop Plug** to programming PC. |
| 40 | **P31** | **Receive** data via **Prop Plug** from programming PC. |

Table 2: Microcontroller Pin Allocation for Temperature Sensor Platform Circuit

**4.2.2 Sensors**

*Pressure Sensor*

The Parallax Altimeter Module MS5607 (#29124) was selected as the pressure sensor. This was due to its wide range of measurement (10 to 1200 milibars) as well as its compatibility with the P8X32A microcontroller.

*Magnetometer*

The magnetometer of choice is the Honeywell 3-Axis Compass Module HMC5883L (#29133). Although it is manufactured by Honeywell, it is highly compatible with the Propeller microcontroller and there is an abundance of Spin code available for this device. It is highly sensitive, and capable of displaying local magnetic field strength and direction about all three axes (X, Y, and Z).

*Temperature Sensors*

The Dallas Semiconductor Temperature Sensor DS18S20 was selected as the temperature sensor of choice due to its wide range of measurement (-55° C to +125° C), simplicity, and abundance of example spin code.

*Accelerometer*

The Hitachi H48C 3-Axis Accelerometer was the accelerometer of choice due to its relatively low cost, versatility, and wide operational range. Although it was difficult to find spin code for this device, it proved to be worth the time. Pitch angle, roll angle, as well as the G-force about all 3 axes is provided by this sensor.

*MEMS Gyroscope*

The Parallax 3-Axis Gyroscopic Module L3G4200D was selected due to the abundance of example spin code by the manufacturer and its compatibility with the Propeller microcontroller. It will also provide very useful information in terms of the spinning (rotational velocity) of the instrument package as it ascends. The figure below illustrates the inertial orientation of the instrument package necessary for the accelerometer and gyroscope.

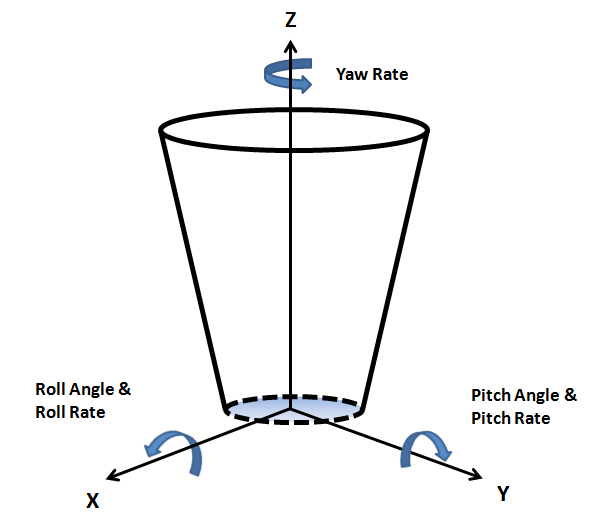


Figure 4: Instrument Package Inertial Orientation

*Voltage Measurement*

Although voltage measurement is not necessarily a sensor, it is discussed in this section since it is one of the data that the sensor section acquires. Voltage measurement was accomplished via a sigma-delta circuit as shown below.

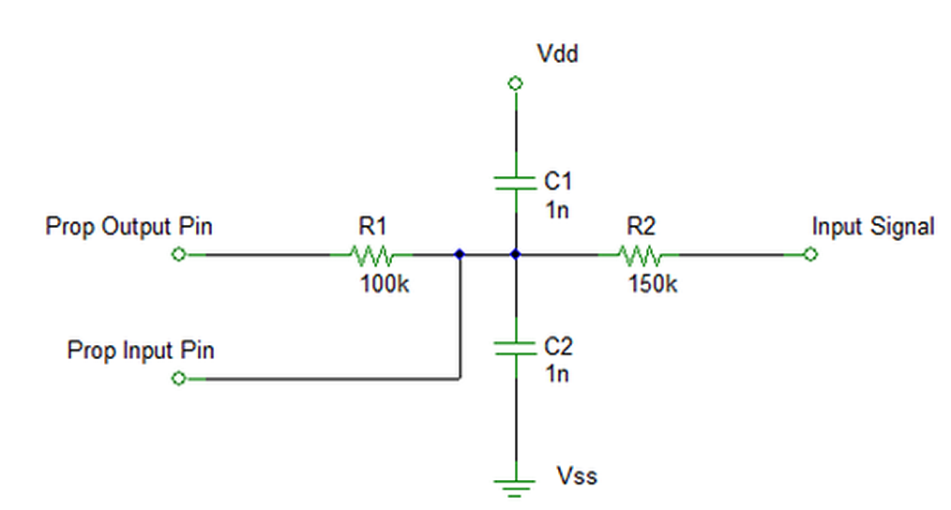


Figure 5: Sigma-Delta Circuit for Voltage Measurement

Below are the equations utilized to compute the resistor values R1 and R2.

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

Where Vmax is the maximum readable voltage, Vmin is the minimum readable voltage, and Vdd is the supply voltage. In theory, using the equations above, R1 should be 29k Ohms to provide the capability of a maximum of 10 Volts to be measured. However, trial and error showed that R1 must be between 6k and 7k Ohms to provide a maximum readable voltage of 10 Volts.

The “Prop output pin” and “Prop input pin” were connected to the primary sensor platform microcontroller via pins 3 and 4 respectively. A file called “ADC.spin” was then used to analyze the voltage off the “Prop input pin.” The resulting value computed by the microcontroller is a 5-digit number that correlates to a certain voltage. By using a voltage source and viewing the serial terminal output from the microcontroller, it was found that a linear relationship existed between the ADC values and the actual voltage. The term “ADC” is derived from the name of the file which is a sampling program utilizing an internal analog to digital converter (ADC) to sample and measure voltage. The figure on the following page displays the relationship between the microcontroller values and the actual measured voltage.

Figure 6: ADC Value and Actual Voltage Correlation

**4.2.3 Miscellaneous Hardware**

*EEPROM*

The 24LC256 EEPROM was selected as a result of recommendation on part of the datasheet for the Propeller microcontroller. It contains 32kb of memory to save code on. It has been found however, that this will not be large enough for our project, and a larger EEPROM will have to be utilized.

*Micro SD Card Adapter*

The Parallax Micro SD Card Adapter was selected due to its compatibility with the Propeller microcontroller. Compatible code was also easy to find. The micro SD card of choice was 2 GB, which is more than enough for the purpose of the project.

*Light Emitting Diodes (LEDs)*

Colored LEDs were used for startup initialization indications as well as debugging and error indications. Four colors were used on each PCB: blue, yellow, green, and red. In general, blue, yellow, and green are used for startup indications, and red if there is an error requiring a reset.

*Voltage Regulators*

Two voltage regulators were used to step down the 9V source (battery pack) down to 3.3 Volts and 5 Volts to certain busses for component use. The voltage regulators used were the ST LF33 and LF50. These were selected due to low dropout, low current draw, and wide operating temperature range.

*Switches*

Two single pole double throw switches were used, one on each PCB, as on/off switches. Turning the switch for either PCB on feeds power to all components, causing the microcontroller to run through its boot-up sequence off the EEPROM.

**4.2.4 Hardware Performance**

Certain aspects of hardware performance needed to be analyzed for the purposes of this project. One aspect is that of power consumption. Below is a table of the voltage limitations on each sensor.

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | **Minimum voltage (V)** | **Maximum voltage (V)** | **Optimal supply voltage (V)** |
| Microcontroller | 2.7 | 3.6 | 3.3 |
| EEPROM | 2.5 | 5.5 | 3.3 |
| Pressure sensor | 3.3 | 6.5 | 3.3 |
| Accelerometer | 4.5 | 5.5 | 5.0 |
| Gyroscope | 2.7 | 6.5 | 3.3 |
| Temperature sensor | 3.0 | 5.5 | 3.3 |
| Magnetometer | 2.7 | 6.5 | 3.3 |
| SD card adapter | 3.3 | 3.3 | 3.3 |
| Voltage regulator (LF33) | 3.3 | 16.0 | NA |
| Voltage Regulator (LF50) | 5.0 | 16.0 | NA |

Table 3: Component Voltage Limitations

On the following page is a table depicting the actual voltage supplied to each component within its respective PCB, the actual current draw of that component, as well as any pertinent operational notes. It should be noted that current draws with an asterisk (\*), such as those for the LEDs, are only negligible since they are inactive for the vast majority of the flight. However, if the LEDs are left on, such as during an error in initialization, current draw will be disproportionately high. This is not considered an issue however, since if any errors occur, these will be viewed by the user and the circuit appropriately reset.

It can also be seen from Table 4 that the total sustained current draw is 140 mA. This total current draw was used in calculating the amount of time available on the battery used. A single 9V, 720 mA hr Energizer lithium-ion battery was utilized. Dividing 720 by 140 gave a time of just over 5 hours battery life, more than adequate for the mission.

|  |  |  |  |
| --- | --- | --- | --- |
| **Primary Sensor Platform Circuit** | | | |
| **Component** | **Actual Supply Voltage (V)** | **Current Draw (mA)** | **Operational Notes** |
| Microcontroller | 3.3 | 29 | Continuous use |
| EEPROM | 3.3 | Negligible | Brief use, only upon initialization |
| Voltage regulator (LF33) | 0-10 | 0.50 | Continuous use, supply voltage varies with battery voltage |
| Voltage regulator (LF50) | 0-10 | 0.50 | Continuous use, supply voltage varies with battery voltage |
| Pressure sensor | 3.3 | 10 | Continuous use |
| Accelerometer | 5.0 | 10 | Continuous use |
| Gyroscope | 3.3 | 10 | Continuous use |
| Magnetometer | 3.3 | 10 | Continuous use |
| SD card adapter | 3.3 | 10 | Continuous use |
| LEDs (4) | NA | Negligible\* | Brief use, only upon initialization |
| **Primary Sensor Platform Circuit Total** | | 80 mA |  |
| **Temperature Sensor Platform Circuit** | | | |
| Microcontroller | 3.3 | 10 | Continuous use |
| EEPROM | 3.3 | Negligible | Brief use, only upon initialization |
| Temperature sensors (6) | 3.3 each | 40 | Continuous use, intermittent |
| SD card adapter | 3.3 | 10 | Continuous use |
| LEDs (4) | NA | Negligible\* | Brief use, only upon initialization |
| **Temperature Sensor Platform Circuit Total** | | 60 mA |  |
| **Cumulative Total** | | 140 mA |  |

Table 4: Overall Sensor Platform Current Consumption

Another aspect of performance that must be analyzed is that of temperature limitations. A table of temperature limitations for each component is shown on the following page.

|  |  |  |
| --- | --- | --- |
| **Component** | **Minimum operating temperature (°C)** | **Maximum operating temperature (°C)** |
| Microcontroller | **-55** | 125 |
| EEPROM | **-40** | 85 |
| Pressure sensor | **-40** | 85 |
| Accelerometer | **-25** | 75 |
| Gyroscope | **-40** | 85 |
| Temperature sensor | **-55** | 125 |
| Magnetometer | **-30** | 85 |
| SD card adapter | **0** | 70 |
| Voltage regulators | **-40** | 125 |

Table 3: Component Operating Temperature Ranges

Due to the nature of the mission for this vehicle and the environment it was exposed to, the lower limit of temperature limitations for each component is deemed most important. The above table was a basis for the degree of insulation and heating that was provided in the instrument package. Heating was done primarily by hand warmers, and there is a slight chance that the upper limit of the temperature limitations for some components may be reached, although this is not likely.

**4.3 Software**

**4.3.1 Language**

The language of choice for the Propeller microcontroller is Spin. This is due to its compatibility with the Propeller. It is also a high-level, object-oriented language that is relatively easy to use even for novice programmers. Interfacing to the microcontroller with a PC, communicating between multiple files, and overall coding is made easier with this language.

**4.3.2 Files Utilized**

**Primary Sensor Platform Code**

**Main**: The main will function as the forefront performing directives by calling all necessary files and commands.

* **Main.spin**: This file calls all data acquisition files at certain intervals. The data acquisition files are listed below.

**Data acquisition files**: These files perform the function of acquiring data from the various sensors used in the instrument package.

* **Accel\_sensor.spin**: This file acquires data from the accelerometer and writes it to the SD card.
* **Altimeter\_sensor.spin**: This file acquires data from the pressure sensor and writes it to the SD card.
* **Gyro\_sensor.spin**: This file acquires data from the MEMS gyroscope and writes it to the SD card.
* **Compass\_sensor.spin**: This file acquires data from the magnetometer and writes it to the SD card.
* **Battery\_voltage**: This file acquires battery voltage from the sigma-delta circuit and writes it to the SD card.

**Engines**: These files are the largest and least modified. Most of these are user-made, while some are from the manufacturer. They provide the necessary information to drive several pertinent programs (such as Parallax Serial Terminal) and sensors.

* **Adc.spin**: Provides necessary data for voltage measurement.
* **Fsrw.spin**: This code provides the necessary support for logging data onto an SD card. FSRW stands for File System Read/Write. It gets called by any files needing support for writing onto the SD card.
* **ASCIIO\_STREngine\_1.spin**: A large engine used to perform many different types of conversions. Although this is a broad, versatile engine, its only purpose for this project is to convert decimal values into character strings to allow logging onto the micro SD card.
* **Safe\_spi.spin**: Provides necessary data to run fsrw.spin.
* **29124\_altimeter.spin**: Provides necessary data to run the altimeter.
* **H48C Tri Axis Accelerometer.spin**: This engine provides the necessary code to run the accelerometer.

On the following page is the software flowchart for the primary sensor platform. It should be noted that three files, GPS\_float\_lite.spin, GPS\_sensor.spin, and Float\_string.spin were not used in the final project. Initially, it was planned to have a GPS module integrated with the sensor platform to provide GPS time and position. However, while integrating all components together at the last minute, it was found that the GPS module was not working. As a result, since it wasn’t required, it was removed from the sensor platform for the time being.

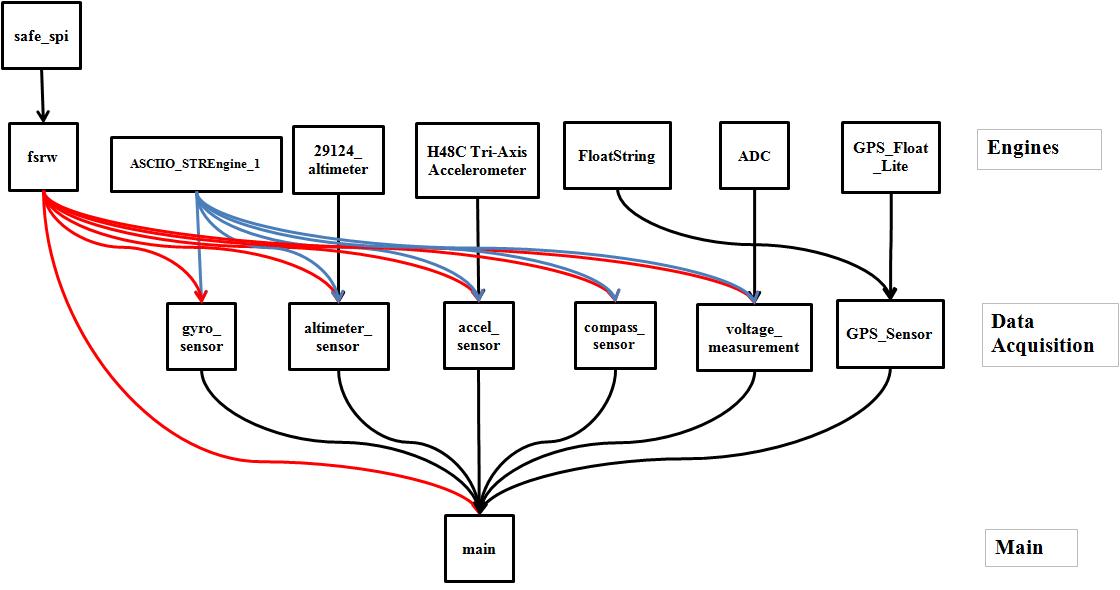


Figure 7: Primary Sensor Platform Software Flowchart

**Temperature Sensor Platform Code**

**Main & Data Acquisition**

* **Temp\_sensor\_main.spin**: Functions to acquire data from all temperature sensors and write to the SD card.

**Primary Engines**

* **Floatmath.spin**: Performs pertinent float math operations required for temperature computations/conversions.
* **Floatstring.spin**: Performs pertinent float string operations for formatting temperature outputs along with units to be written to the SD card.
* **Onewire.spin**: Primary engine for driving temperature data acquisition and running the temperature sensors.
* **Fsrw.spin**: Provides the necessary support for logging data onto an SD card. FSRW stands for File System Read/Write. It gets called by any files needing support for writing onto the SD card.

**Secondary Engines**

* **Safe\_spi.spin**: Provides necessary data to run fsrw.spin.

Below is the software flowchart for the temperature sensor platform.

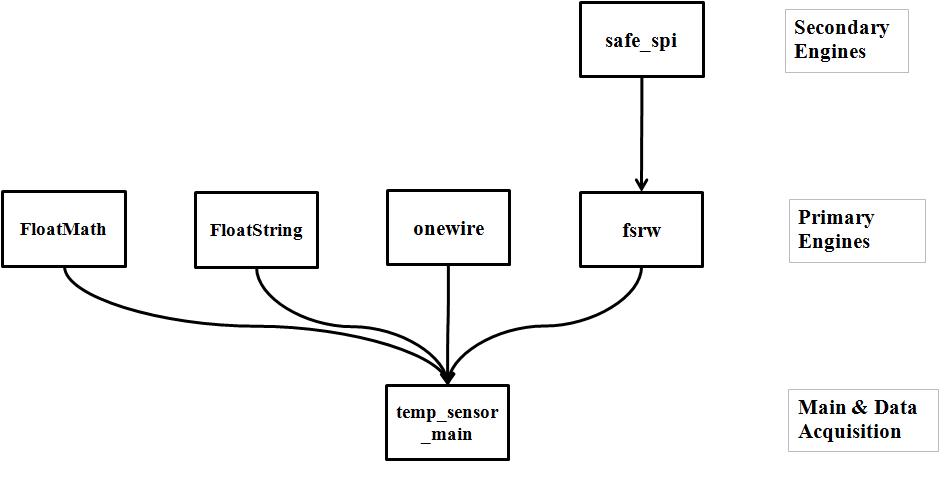


Figure 8: Temperature Sensor Platform Software Flowchart

**4.4 Results**

**4.1.1 Balloon Launch**

The balloon launch took place on the morning of April 28, 2012 at 9:00am local time. The launch site was a field behind Hillsboro high school. Hillsboro was chosen as the launch site due to the projected flight path of the vehicle. It was desired to have the vehicle land in a grass field away from an individual’s property ideally. Since this could not be accurately predicted, it was simply desired to have the vehicle *not* land on water (i.e. a lake), a road, or any heavily populated metropolitan area. As a result, since upper level winds were forecasted to send the vehicle directly east, Hillsboro was deemed an excellent location since the expected landing site was likely to be near Gun Barrel City.

Launch conditions were not favorable due to high winds (approximately 20-30 mph) and low overcast (approximately 800-900 ft). In hindsight, it was probably best to wait for another day with more suitable conditions. However, since it was the end of the semester, the team was left with no choice but to launch that day. The outcome of the launch can be viewed on the documentary short, “Stratocasters – The Movie.” In summary, the launch was a failure. The balloon was not filled with enough hydrogen to adequately lift it in the given conditions. In fact, there may have been hardly enough hydrogen to lift the balloon even in calm conditions. This was due to error on part of the launch crew. Upon release, the instrument package impacted the ground several times before becoming airborne for a brief period of time. The vehicle then flew at a low altitude of about 30-50 ft. before impacting a tree, separating the balloon from the instrument package. The package then tumbled to the ground near an auto shop, where it was recovered by the team.

**4.1.2 Data – General**

Amidst the failed launch, data was still collected, which helps to determine more accurately what happened to the package during its short flight. It was noted that there were three power resets of the sensor platforms, followed by a complete shutdown. The cause of these power resets have not yet been determined. The power resets resulted in four segments of data. This is due to the fact that if the circuits are turned off then turned back on, the microcontrollers will re-initialize using the data stored on the EEPROMs, and will continue to log data. A reset can be identified in the Excel file by re-written headers. It must also be noted that the time reference for all data is not in units of time, but samples of data. For the primary sensor platform, samples were taken 4-5 times per second. For the temperature sensor platform, one sample was taken approximately every 30 seconds.

**4.1.3 Data – First Segment**

From the data collected, and based on knowledge from the launch, the first segment begins at the moment the sensor platforms were turned on, and ends at some point before the release. The duration of the first segment is approximately 2 minutes and 40 seconds. Based on the pressure sensor’s data, it can be seen in figure 9 on the following page that the altitude and pressure were relatively constant. Slight disturbances can be seen in altitude, which range from 5-10 ft. deviations. This likely occurred when the instrument package was picked up and carried from its startup point directly behind the high school to a more suitable field behind the high school. It can also be noted that the pressure for that day was around 989 milibars.

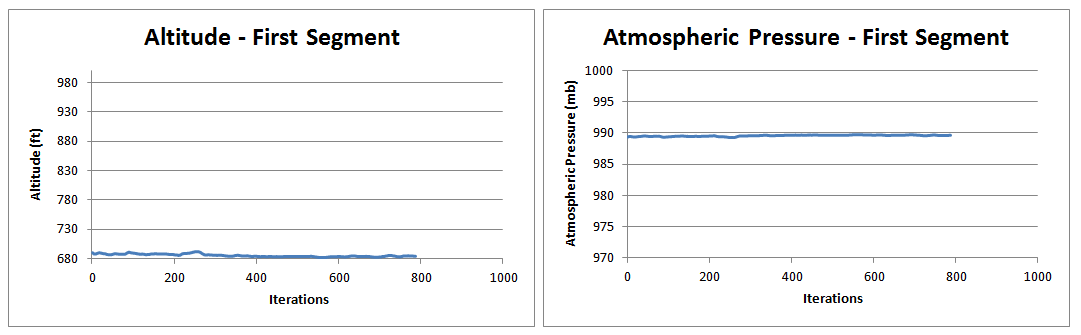


Figure 9: First Segment Barometric Altitude & Atmospheric Pressure Respectively

Below are four graphs depicting the instrument package’s attitude during the first segment. The four graphs include roll angle, pitch angle, roll rate, and pitch rate. It can be noted that there is a relatively large angle change during the first 150 iterations of the roll angle graph. There is also a slight change in pitch angle during this time. After this, the angle remains fairly constant, although some noise can be observed, which is simply due to the sensitivity of the accelerometer and gyro. This angle change can be explained by the manner in which the sensor platforms were turned on. Since the switches were mounted directly onto the PCBs, the PCBs had to be picked up and held at an angle to turn the switches on. Afterwards, the PCBs were set inside the instrument package.

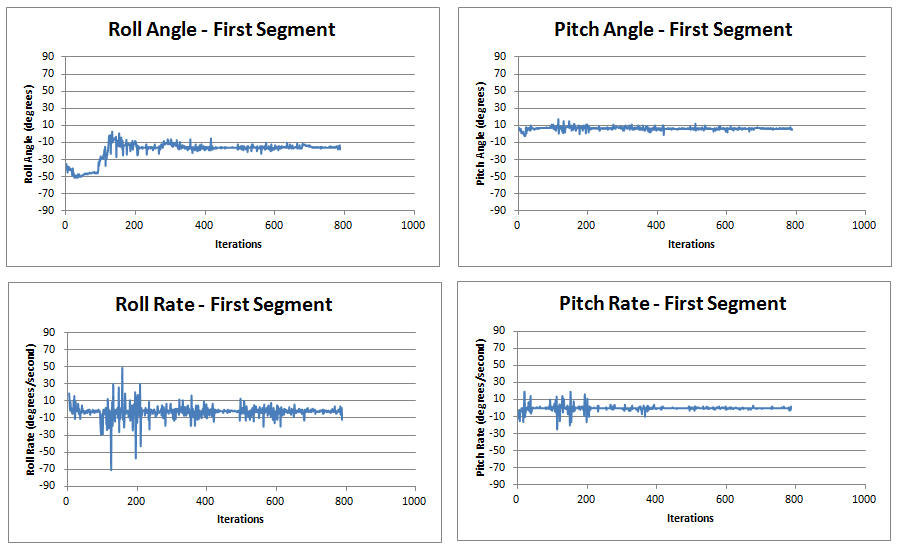


Figure 10: First Segment Vehicle Attitude

The figure below displays what can be considered miscellaneous data, which includes battery voltage and local magnetic field strength. The battery voltage graph displays a constant value of approximately 43000 for the duration of the first segment. Referring to figure 6, a value of 43000 correlates to approximately 10V. This makes sense since a fully charged 9V battery contains approximately 10V. The magnetic field strength graph shows a mean of 580 on the Z axis, 250 on the Y axis, and 90 on the X axis. Dividing each value by 1090 gives a magnetic field strength of 0.08 Gauss on the X axis, 0.23 Gauss on the Y axis, and 0.53 Gauss on the Z axis. Averaging each of these values gives a mean magnetic field strength of 0.28 Gauss. This value makes sense since the Earth’s magnetic field strength varies from approximately 0.25-0.65 Gauss. Since the Earth’s magnetic field is strongest at the poles, and weakest at the equator, one can assume that at the launch site’s location (approximately 32°N latitude) the Earth’s magnetic field strength would be relatively weaker, and as a result on the lower end of the field strength range. It can also be noted that the electronics had minimal effect on the magnetometer where it was positioned.

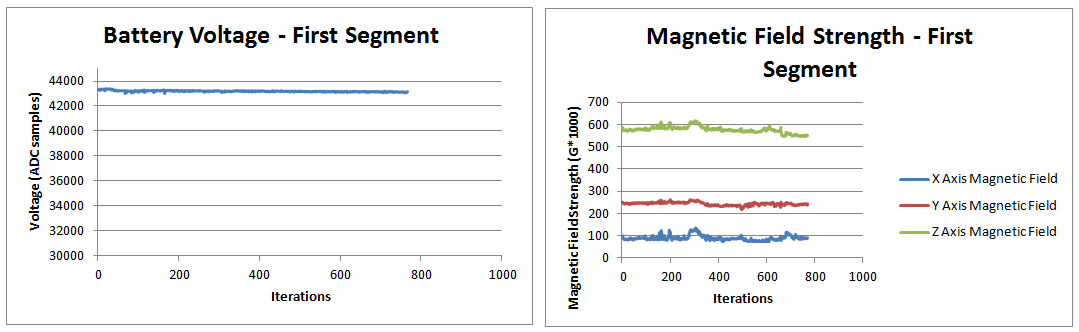


Figure 11: First Segment Battery Voltage & Local Magnetic Field Strength Respectively

**4.1.4 Data – Second Segment**

Like the first segment’s data, second segment data was also fairly uneventful. It is assumed that the second segment took place after the first segment ended, but before release. Figure 12 below shows altitude and atmospheric for the second segment, which closely resembles that of the first segment.

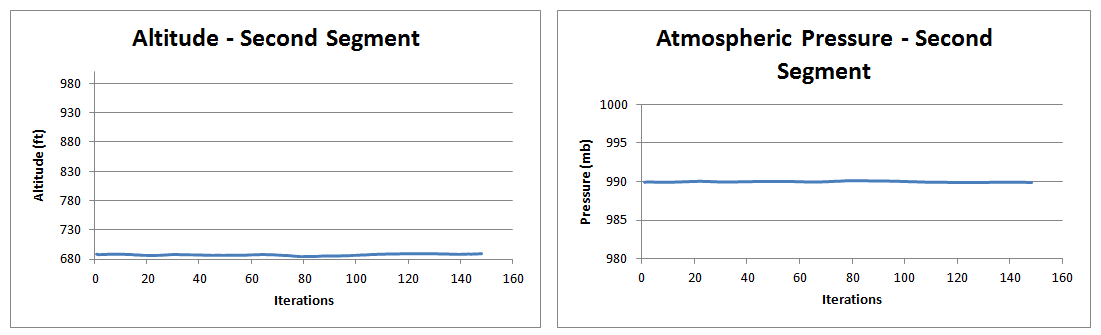


Figure 12: Second Segment Altitude & Atmospheric Pressure Respectively

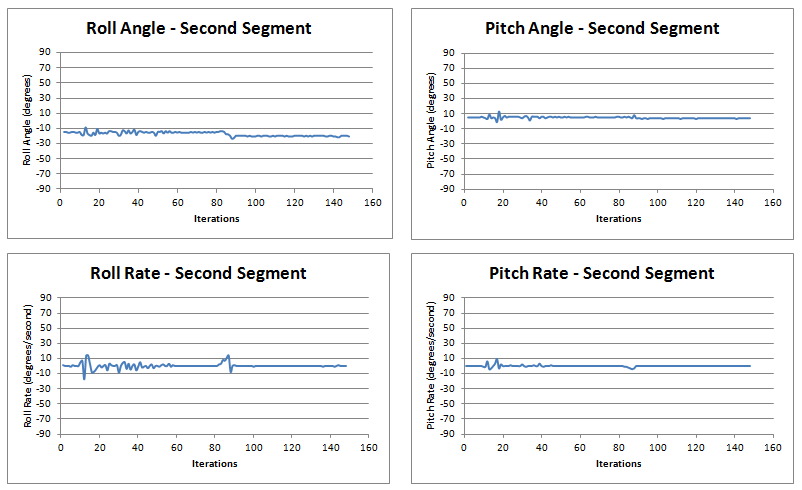


Figure 13: Second Segment Vehicle Attitude

Figure 13 above illustrates the second segment attitude. It can be seen for the most part that the vehicle was stationary for the second segment, with the exception of a slight disturbance around the 80th iteration. This was probably a result of bumping the instrument package while sealing the instrument package housing just before the flight.

Figure 14 below shows battery voltage and magnetic field strength for the second segment. It can be seen to largely resemble that of the first segment, with the exception of the disturbance mentioned above being shown on the magnetic field strength graph. The offset of the magnetic field strength after the disturbance can be explained by a shift in the sensor. As a result of the shift, the flux lines strike the sensor at different angles, resulting in the same overall average field strength, but different values for each axis being offset by the same amount.

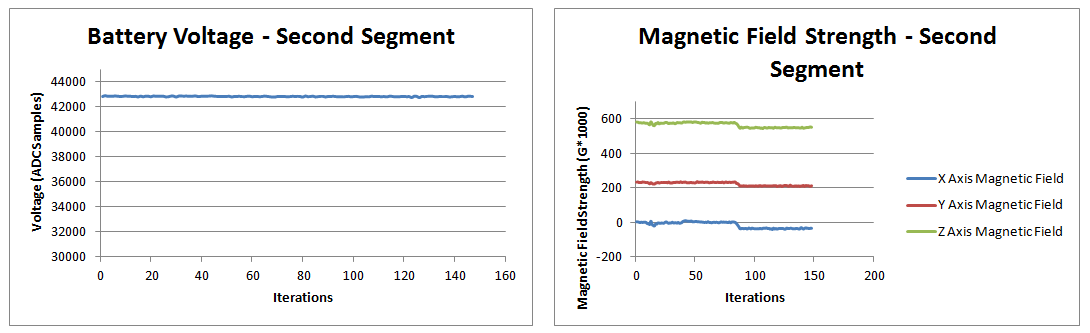


Figure 14: Second Segment Battery Voltage & Local Magnetic Field Strength Respectively

**4.1.5 Data – Third Segment**

The third segment is assumed to begin at some point shortly after release until final impact. The third segment data lasts approximately 30 seconds.

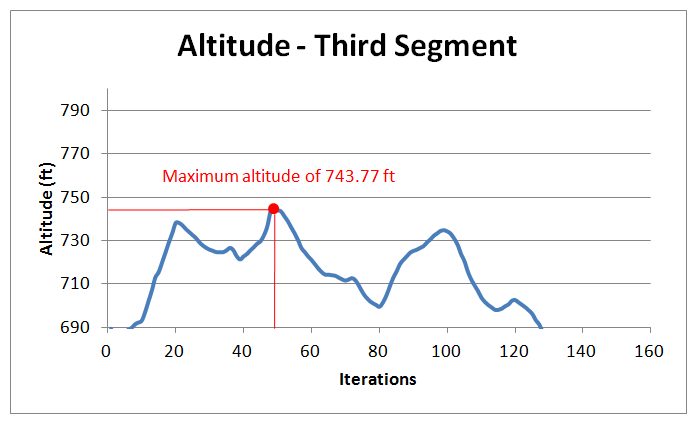


Figure 15: Third Segment Altitude

Figure 15 above displays the altitude during the third segment. The maximum altitude of the package, 743.77 ft. MSL (53.77 ft. AGL) can be seen around iteration 50 (10 seconds into the third segment). Atmospheric pressure is simply the inverse of the altitude according to a function based on the standard atmosphere.

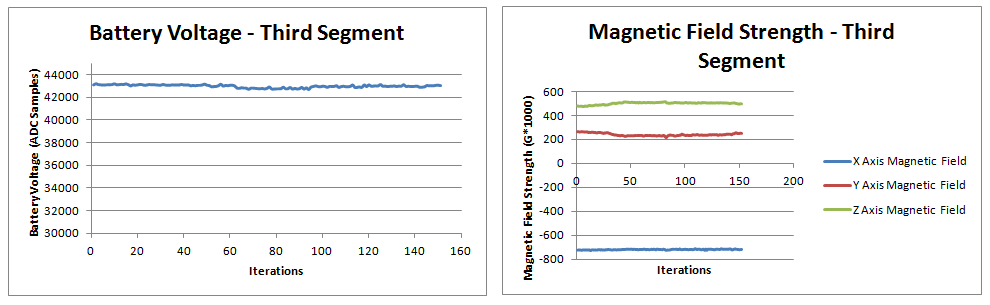


Figure 16: Third Segment Battery Voltage & Magnetic Field Strength Respectively

Figure 16 above once again displays battery voltage and magnetic field strength for the third segment. Battery voltage continues to remain unchanged due to the short amount of time that has passed. The magnetic field strength graph, however displays altered values. This is likely to be a result of the tumbling of the package during this phase, resulting in shifting of the magnetometer’s orientation.

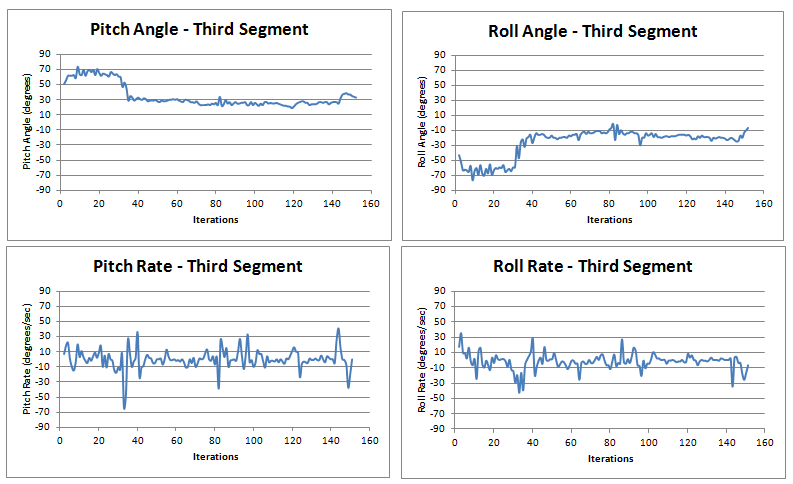


Figure 17: Third Segment Vehicle Attitude

Figure 17 above outlines the vehicle’s attitude during the third phase. It can be noted that a large change in attitude takes place in the first 5 seconds of this segment. It can be seen that the pitch angle transitions from 70° to 20-30°, and the roll angle transitions from -60° to -20°. This can be assumed to be the time at which the instrument package erects as it becomes airborne. Attitude then continues to remain relatively constant for the rest of this segment.

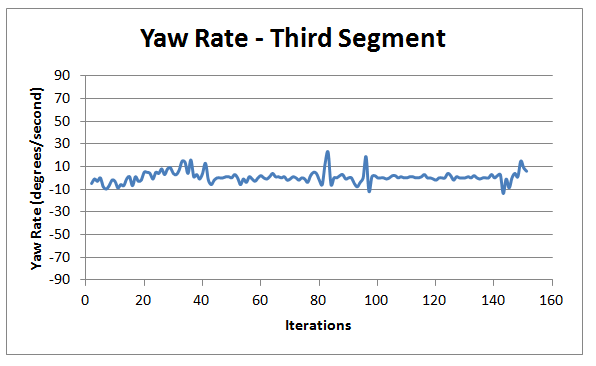


Figure 18: Third Segment Yaw Rate

Figure 18 above displays the yaw rate of the instrument package during the third segment. It can be seen that it remains fairly constant near zero for the majority of the time. Exceptions include the transition at the beginning as outlined above, and a few disturbances. This is likely due to the occasional steady rotation and reversal thereof observed in the on-board camera of the package.

**4.1.6 Data – Fourth Segment**

The fourth segment is assumed to take place after the instrument package’s separation from the balloon and final impact with the ground and ends a short period of time after it was picked up by an individual that found it.

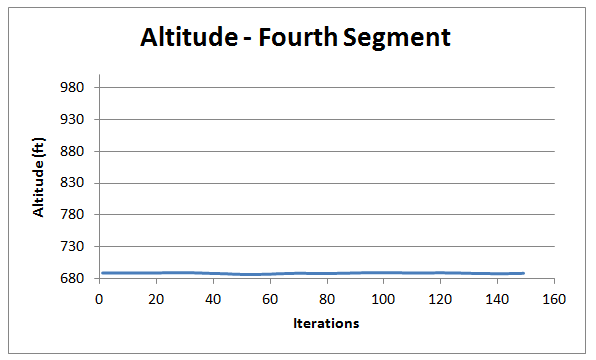


Figure 19: Fourth Segment Altitude

As can be seen from figure 19 above, altitude remains fairly constant for the duration of the fourth segment. This is largely why it is believed that the fourth segment took place after the balloon’s final impact with the ground.

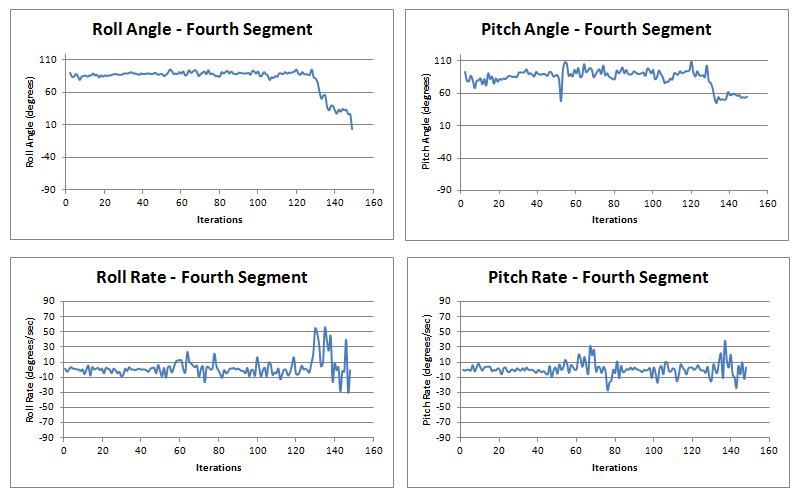


Figure 20: Fourth Segment Vehicle Attitude

Figure 20 on the previous page shows the vehicle’s attitude during the fourth segment. For the majority of the time, the roll and pitch angles remain fairly constant around 90°. It is assumed that this is a result of the package lying on its side. The package erects however to almost 0° roll angle and 60° pitch angle from the 130th to the 150th iteration. It is assumed at this point that the individual that found the instrument package picked it up. The final shutdown of the system occurred after this time.

**4.1.7 Data – Temperature Measurements**

Figure 21 below displays the temperature data across all segments.

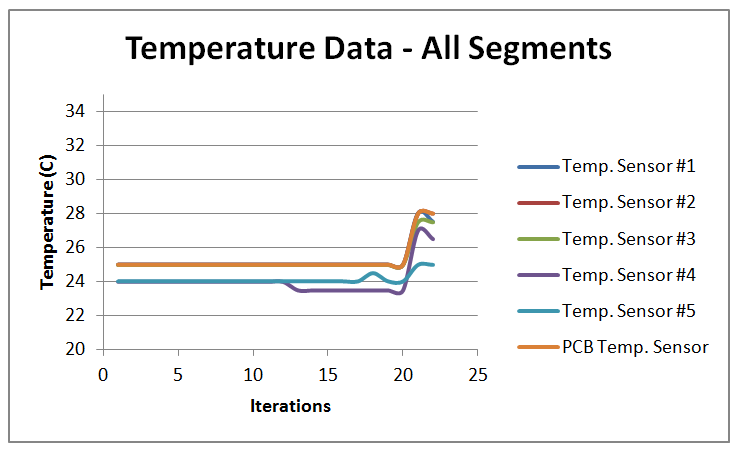


Figure 21: Temperature Data Across All Segments

Due to numerous power resets, the short duration of most segment intervals, as well as the fact that a single temperature measurement cycle took approximately 30 seconds, temperature data ended up being skewed and inaccurate. As a result, only a general trend can be determined. As can be seen from the graph, temperature remains constant for most of the time, but eventually increases. The value staying constant initially is self-explanatory, however, the eventual increased can be explained by the fact that hand-warmers were added at the last minute before the launch to help keep the electronics warm during the vehicle’s ascent.

**5. Conclusion**

In short, although the launch was a failure, the hardware collected data as intended, and helped to paint a more accurate picture of what occurred during the flight, which was deemed to be a success. A second launch is planned for late May, before which several improvements will be made to the hardware. The first and foremost will be that of GPS integration. The GPS module will be re-integrated with the sensor platforms and made working. This will allow the data to have a time reference as well as position. Improved rigging of the sensor section will also be done. During this flight, the two PCBs were basically “slapped in there” and as a result the inertial orientation was not accurate. They were also not secured, which may be a result of the power resets (from being tossed around). This will also be improved prior to the second launch. Lastly, better conditions will also be ensured during the next launch. Calmer winds and clear skies will be sought out. In conclusion, although the launch did not go as planned, the outcome and data recovered was considered a success on part of the senior design team.

**6. References**

1. Parallax, "Altimeter Module MS5607," MS5607 Data Sheet, Nov. 2011
2. Parallax, "micro-SD Card Adapter," 32312 Data Sheet, Apr. 2011
3. Parallax, Propeller P8X32A Data Sheet, Nov. 2007
4. Parallax, "Compass Module 3-Axis HMC5883L," 29133 Data Sheet, Nov. 2011
5. Hitachi, "H48C 3-Axis Accelerometer Module," 28026 Data Sheet, Jul. 2007
6. Maxim, "High-Precision 1-Wire Digital Thermometer," DS18S20 Data Sheet, Aug. 2010

**7. Appendices**

**7.1 Appendix 1: main.spin File Code**

**7.2 Appendix 2: temp\_sensor\_main.spin File Code**

**7.3 Appendix 3: Third Segment Excel Data Results**