Student-Built High-Altitude Balloon Payload

Russell Jeffery

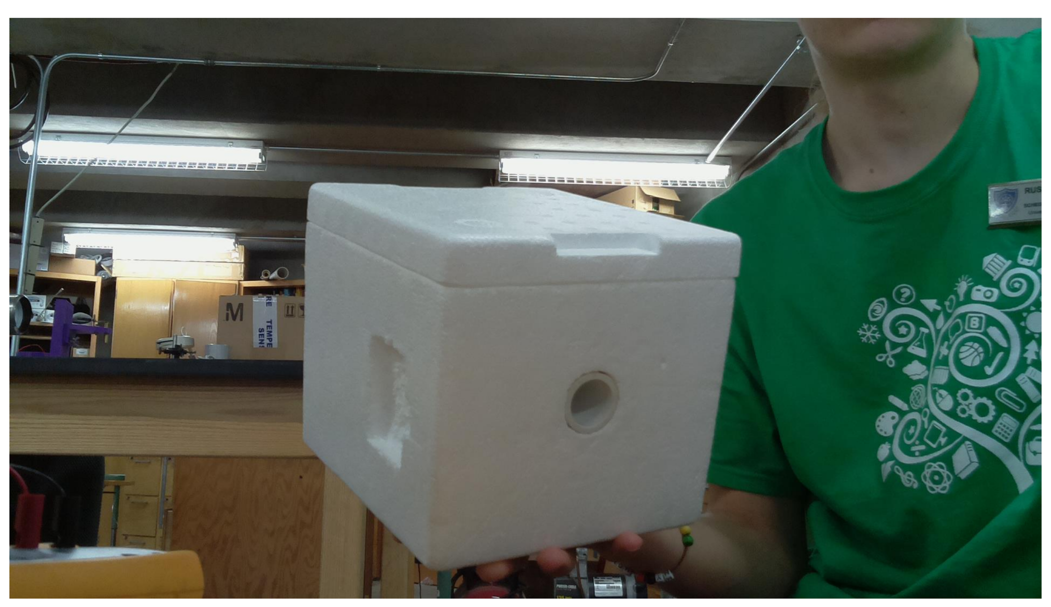
Faculty mentor: Dr. William V. Slaton

University of Central Arkansas

Department of Physics and Astronomy

ABSTRACT

A payload was designed for a high-altitude weather balloon. The flight controller consisted of a Raspberry Pi running a Python 3.4 program to collect and store data. The entire payload was designed to be versatile and easy to modify so that it could be repurposed for other projects: The code was written with the expectation that more sensors and other functionality would be added later, and a Raspberry Pi was chosen as the processor because of its versatility, its active support community, and its ability to work easily with sensors, servos, and other hardware. The program collected data from a GPS breakout board, a Raspberry Pi camera, a geiger counter, two thermocouples, and a pressure sensor. The data collected clearly shows that pressure and temperature decrease as altitude increases, while β-radiation and γ-radiation increase as altitude increases. These trends in the data follow those predicted by theoretical calculations made for comparison. This payload was developed in such a way that future students could easily alter it to include additional sensors, biological experiments, and additional error monitoring and management.



A side view of the styrofoam box that the payload would be flown in.

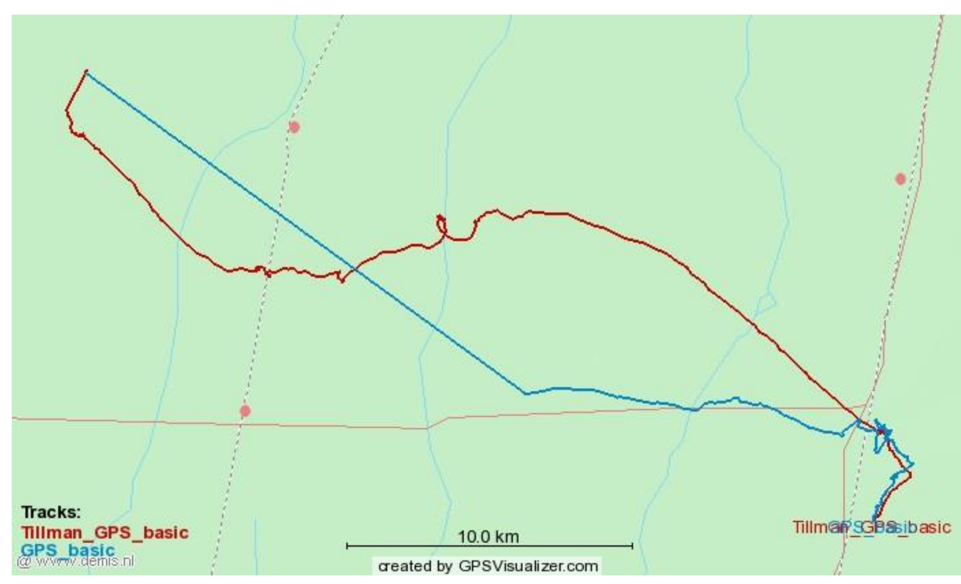
Goals

- Collect atmospheric data and compare it to models
- Design a versatile platform
    - Easy to modify
    - Able to record data from a variety of sensors
    - Able to control hardware such as lights and servos
- Get cool pictures from close to space

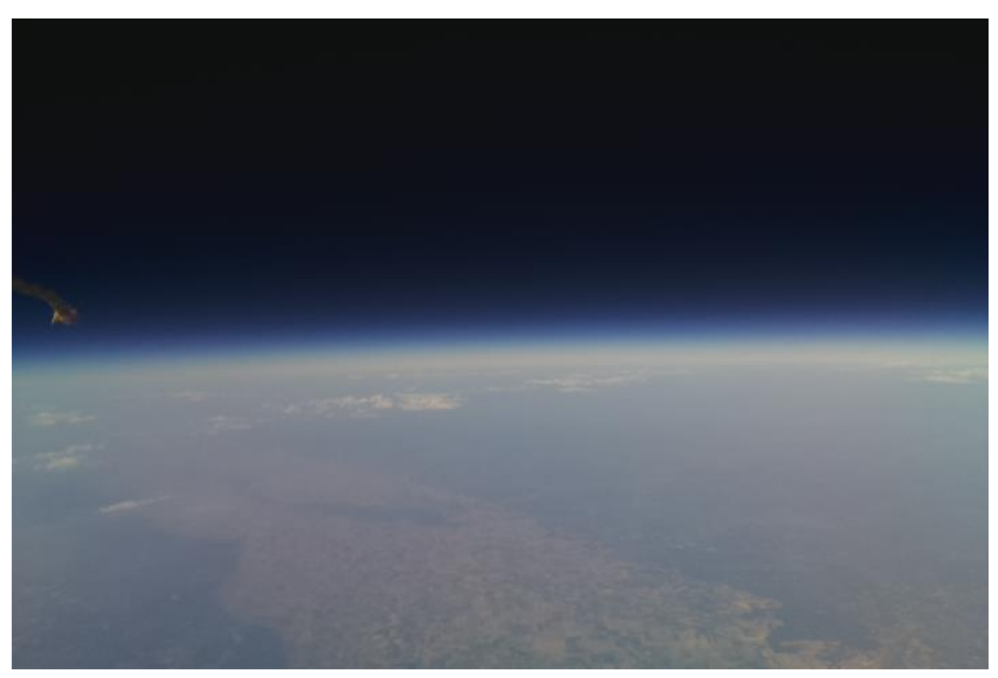
The Flight: ABS-46

The balloon was launched from an airport near Cherry Valley, AR. Between the time that it was turned on and the time that it was recovered and turned off, the flight lasted for about 130 minutes, collecting over 900 points of data and pictures.

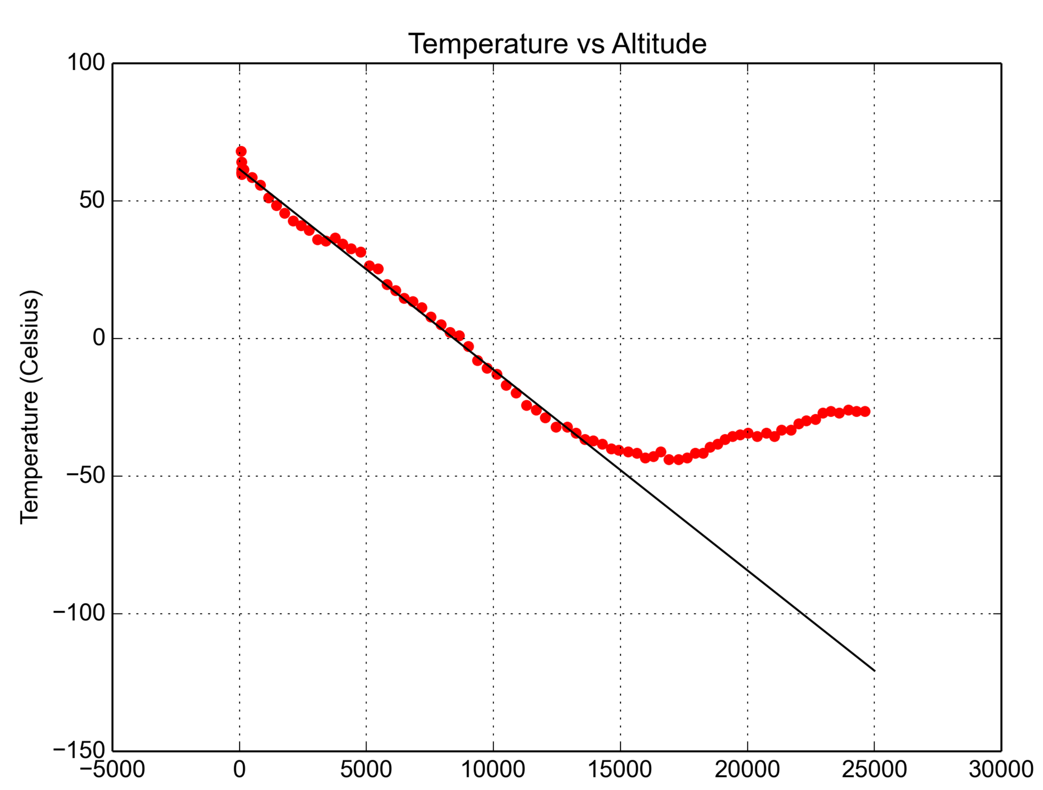
Below are the flight paths according to our GPS unit (blue) and the higher-end GPS used to track the balloon (red). As this graph makes very clear, the data diverges. Partly, this is because our GPS does not report data if it is above 60,000 feet—around 20,000 feet below our maximum altitude.



The location as recorded by two GPS units.



An image taken from the stratosphere at 23,000 meters (75000 feet)



Data Collected

The data that we collected is precisely the kind of data that NASA and NOAA collect to help them understand weather patterns, the environment of space, and our changing climate. Below are graphs of some of the data that was collected during the flight of ABS-46.

The pressure by altitude can be modeled by the barometric equation:

$$P(h) = P_0 e^{-mgh/kT}$$

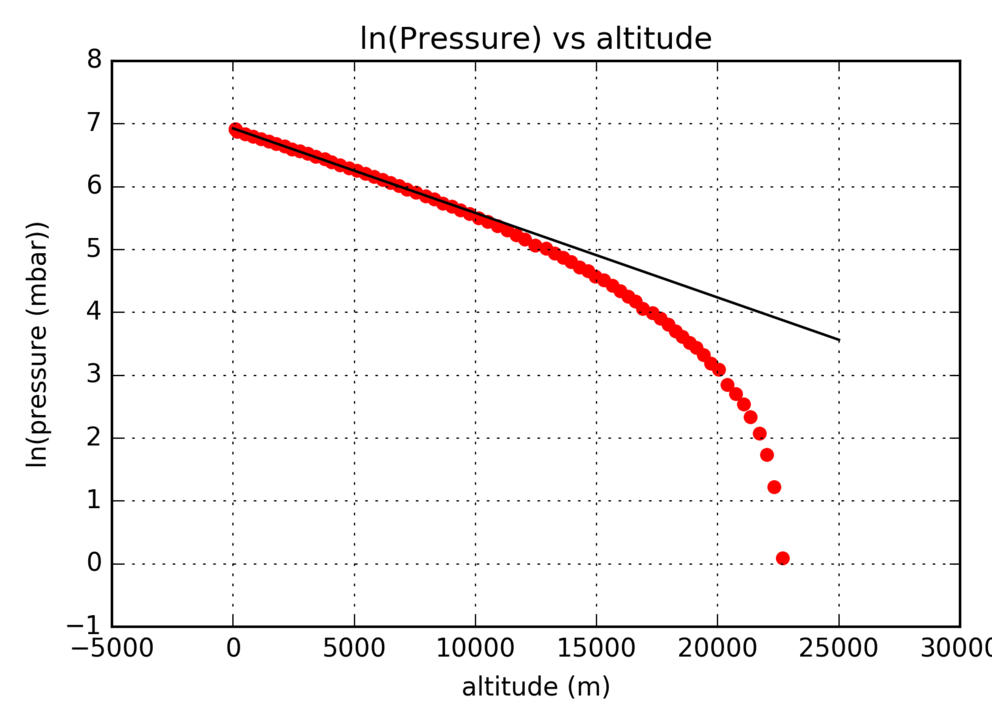
The graph below shows the natural log of both axes, so we would expect to see a linear plot. The graph clearly deviates from this, leading us to theorize that the sensor got too cold. Before that, however, the slope of the fit line is close to what we would expect. assuming the average air temperature is -4.0 degrees Celsius (the mean of our temperature readings).

Using least-squares fitting, we find that:

$$slope = (-1.345 \pm 0.008) \times 10^{-4} \text{ (1/meters)}$$

This is close to what our model predicts:

$$slope = -1.27 \times 10^{-4} \text{ (1/meters)}$$



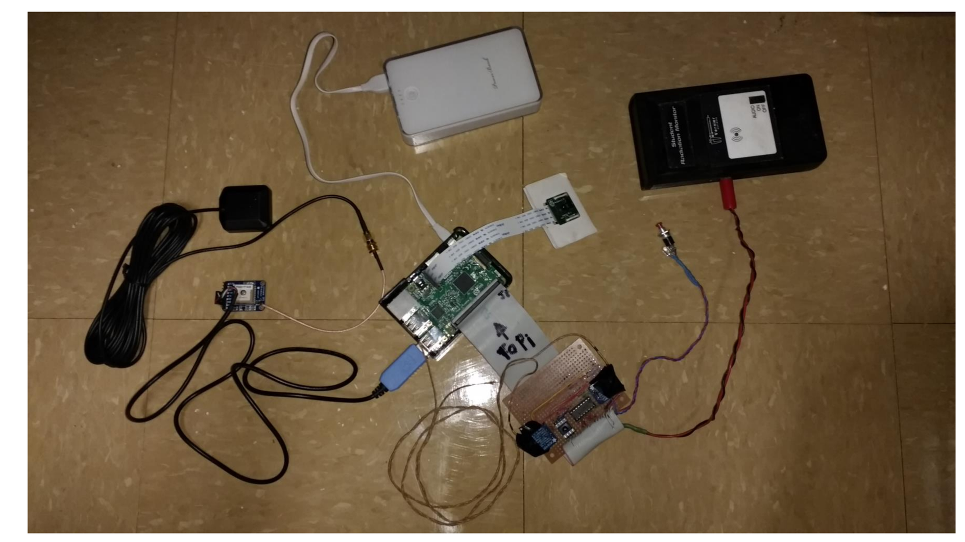
In the temperature graph to the left, the pressure falls steadily with rising altitude until around 16,000 m—the tropopause—after which it begins to rise again as the altitude increases into the stratosphere. Using the same method as above to find the slope of the first half of the data:

$$slope = -7.33 \pm 0.08 \text{ (C/km)}$$

As predicted, this measurement is close to the approximate dry adiabatic lapse rate of -10 C/km

Flight Controller

The program, written in Python 3.4, collected data from a GPS, a camera, a geiger counter, two thermometers, and a pressure sensor. While building the hardware was important, writing the code took more time and constituted a more significant achievement than any other part of the project.



The hardware flown on ABS-46

Issues and Lessons Learned

- Our GPS module stopped reporting new data at an altitude of 7600 meters (25,000 feet). This happened because the GPS used could not report any higher than that.
- At just around the highest point in the flight, the Raspberry Pi stopped collecting data. This was probably caused by the processor getting too cold, so next time, the payload will likely need a heater of some kind.

Future Applications

- This device might be re-purposed for . . .
    - Controlling a robot aircraft
    - Maintaining a certain air speed in a wind tunnel
    - Measuring the temperature in an incubation chamber and posting the information to a website
    - Demonstrating to students how different STEM fields are related
    - Teaching students how to use programming to solve real world problems

ACKNOWLEDGEMENTS

This project was made possible by an Arkansas Space Grant Consortium (ASGC) Workforce Development Grant.