Colorado Space Grant Consortium

SENSOR PACK & CAMERA SYSTEM DEMOSAT REPORT

EOS PUMAS

Written by:
Colin Borror, Chris Hargett, Daniel Kroneberger, Charles Larsen,
& Marco Villasuso

December 15th, 2023
Arapahoe Community College

Table of Contents

1.0 Mission Overview

This semester (Fall 2023) Daniel, Charlie, Colin, Marco, and Chris came together with the intention of designing an experimental payload that would allow us to develop the technical and collaborative skills necessary to carry out more ambitious edge of space science projects for both future Space Grant missions as well as those we aspire to take part in throughout our careers. With that in mind, this semester we elected to design a basic sensor array package capable of accurately measuring the changes in temperature (both internal and external), pressure, telemetry, and humidity of a non-propulsion/non motorized system approaching the edge of space conditions of our upper stratosphere. This data would also serve as an indicator of the extent to which the intended function of the different design elements of our experiment were successful. Alongside the main objective of our payload, we opted to take on the daunting, but potentially incredibly rewarding endeavor of integrating imaging data into our mission. This would not only capture the memory-worthy images of our payload flight to the edge of space but could provide valuable insight as to what potential events, if any, might occur that could be capable of severely impacting the integrity of collected data. In addition, having a visual aspect also allows for the visual representation of the effects of high altitude conditions on the atmosphere and matter including experimental equipment. In order to achieve this, the team met at least once weekly to discuss plans, delegate tasks, and review modules together. Throughout the process of developing and designing our first Demosat project, we were able to successfully acquire a multitude of skills including and relating to, but not limited to: Programing, computer science, electrical wiring and installation, data analysis, structural design, along with the types of parameters for adequate flight testing for both structural and electrical components during launch and recovery, and as well as time management and communication. The culmination of our efforts and these newly developed skills resulted in what we agree to be the successful payload experiment carried out upon launch of a high altitude weather balloon Saturday, Nov 11, 2023.

2.0 Design

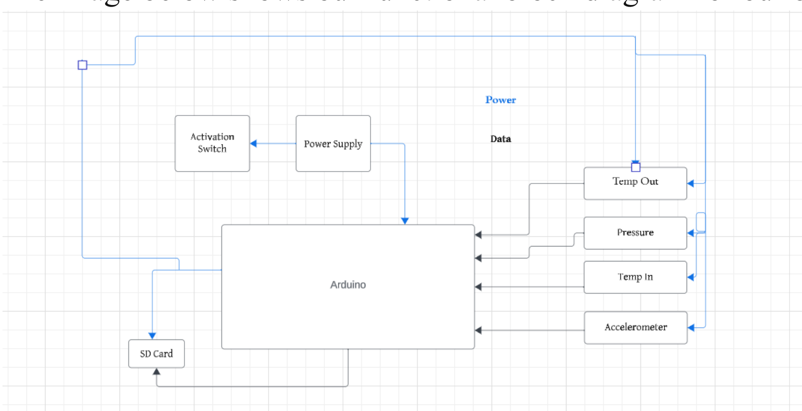
Following sections are our basic sensor pack setup, camera system design and our payload design.

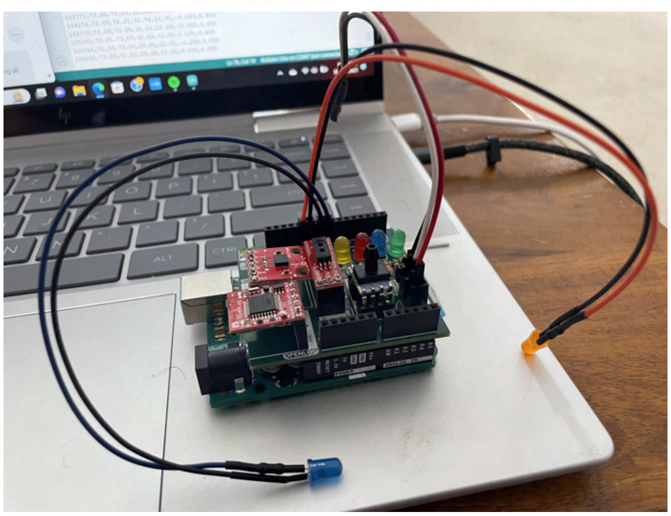
2.1 Sensor Pack

For our payload, we used the DemoSAT basic sensor kit which contained:
1 SEN-09569 SparkFun Electronics Humidity Sensor
1 TMP36 SparkFun Electronics Temperature Sensor (Internal Temperature)
1 SEN-11050 SparkFun Electronics Temperature Sensor (External Temperature Sensor)
1 SSCDANN150PGAA5 TruStability® Board Mount Pressure Sensor
1 ADXL335 SparkFun Electronics Accelerometer

By including each of these sensors in our payload, we hoped to get general measurements of atmospheric conditions that we can later take into account designing future payloads. These sensors all communicated data readings to the openlog, which were stored on our SD card to be viewed later. All sensors were connected via a shield board which was connected to our main arduino.

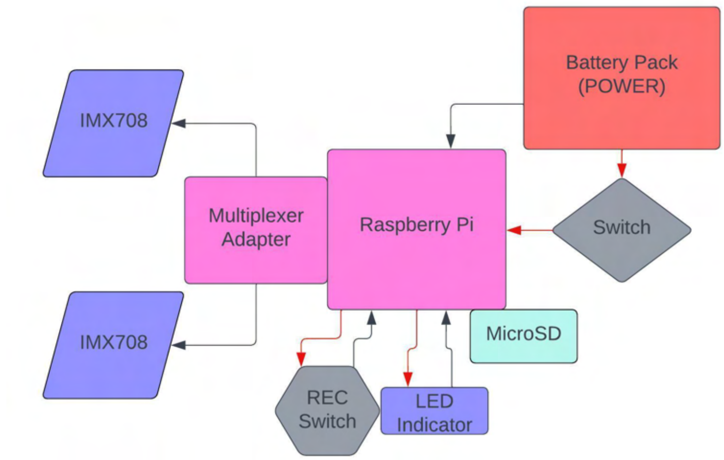
The image below shows our functional block diagram for our board:



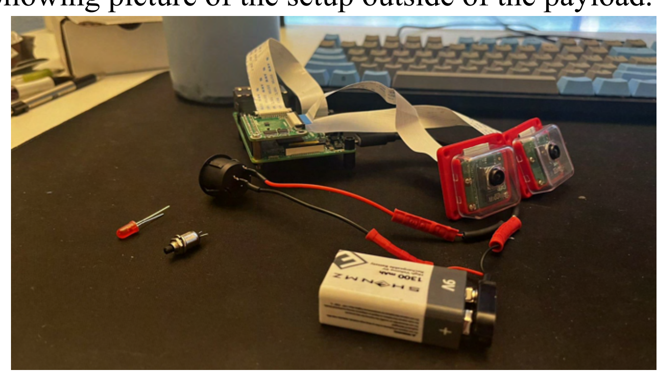


This image shows our completed arduino uno kit with all sensors attached and operational

2.2 Camera System Design



Shown in the diagram above is the original wiring diagram for the RaspberryPi and Camera System. It showcases the RaspberryPi receiving switch-controlled power from a battery pack. Of course including a MicroSD card with the OS and other needed files, and as far as onboard connections go… Lets start with the ArduCam MultiCamera Plexer Adapter. This adapter connects to the 24 pin camera cable on the RaspberryPi and allows for two cameras to be connected instead of only one. Into this adapter we had purchased two IMX708 wide angle cameras. The other onboard connections included a momentary switch (used to start/stop recording). As well as an LED indicator light for use in headless mode to understand all operations including code. With the following picture of the setup outside of the payload.



With the code consisting of power on function, Run **OFFICIAL.py** via **rc.local**, then Start Input loop via GPIO Pins (Momentary Switch), INPUT RECEIVE: Flash LED via GPIO Pins (short duration) & Begin Time Lapse Recording (und), Save on GPIO Input, Auto exit set for 3 hours. The following below is a picture of a portion of the code.

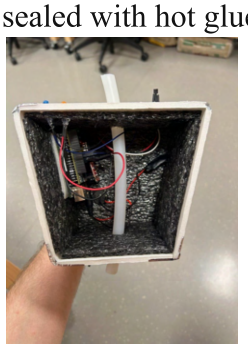
```
#!/usr/bin/python3
import time

import RPi.GPIO as GPIO

from picamera2 import Picamera2
from picamera2.encoders import H264Encoder

button_pin = #FILL THIS
led_pin = #FILL THIS

#initialize GPIO code
GPIO.setmode(GPIO.BCM)
GPIO.setup(button_pin, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.setup(led_pin, GPIO.OUT)

#initiliazes PiCamera2 code
picam2 = Picamera2()
    # video_config = picam2.create_video_configuration()
    # picam2.configure(video_config)
picam2.resolution = (3840, 2160)
picam2.framerate = 4
encoder = H264Encoder(bitrate=20000000)
    # encoder = H264Encoder(qp=300)

#File settings
timestamp = int(time.time())
output = f"/home/demopi/Desktop/TIMELAPSE/test{timestamp}.h264" #set output

#define LED flash to x amount of times
def LEDFLASH(x):
    for _ in range(x):
        GPIO.output(led_pin, GPIO.HIGH)
        time.sleep(0.3)
        GPIO.output(led_pin, GPIO.LOW)
        time.sleep(0.3)

#OFFICAL RECORD LOOP
recording = False
try:
    while True:
        button_state = GPIO.input(button_pin)
        if button_state == GPIO.LOW:
            if not recording:

                recording = True

                picam2.start_recording(encoder, output)

                LEDFLASH(5)
        elif: button_state == GPIO.LOW and recording == True:
            picam2.stop_recording()
        else:
            time.sleep(3000000)
            picam2.stop_recording()
```

2.3 Structural Design

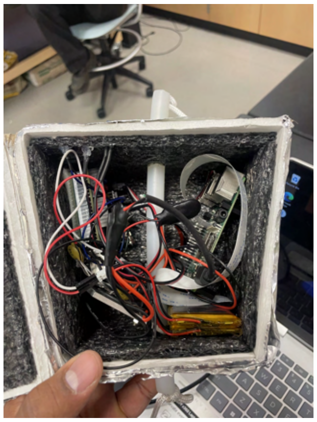
For our payload design, we chose to use foam core board as our structural base with black foam insulation lining the interior of the payload. This material would not only insulate the payload from atmospheric temperatures, but would help the payload maintain its shape and withstand any impacts. Additionally, we designed the exterior of the payload as a trapezoid with the two slanted walls at a 10 degree angle. This would allow the payload to have additional room to house the arduino boards while allowing our cameras to get the best possible view of earth given their 120 degree field of view.

An image of our payload housing without any interior pieces (It is held upside down in this photo).

The tube that the attachment rope would be fed through was pushed through the middle of the payload. and the arduino kit was placed in the top corner of the payload, with the raspberry pi and camera in the opposite corner to balance the weight. Because the smaller side of the trapezoid was meant to be at the bottom of the payload, the batteries were housed there to allow the payload to be mounted properly during flight. LEDs, external temperature sensors and cameras needed to be on the exterior of the payload, so holes were cut in the exterior in order to allow the systems to pass through. The holes were sealed with hot glue in order to keep the payload insulated.



An image of the arduino being housed in the payload



Final interior image of payload after recovery. Battery packs for the arduino and raspberry pi can be seen at the bottom, while the raspberry pi can be seen added at the top right of the payload.

3.0 Management

At the start there were four of us, Colin, Chris, Daniel, and Charlie, then toward the end of the project we added Marco who was a tremendous help to the team. Most of us dabbled in a little bit of everything in the project, with learning and developing skills the main goal. Colin and Charlie constructed the payload, Colin, Chris, and Charlie tested the sensors, Daniel was the lead on the camera portion, and Chris the lead on the coding portion of the Arduino sensor pack. Marco helped with pre launch and launch as well with this report. Our schedule ended up being: the Preliminary Design Review on September 18th, the Critical Design Review on October 9th, Flight Readiness review on November 6th, with the launch occurring on November 11th.

4.0 Budget

Team EOS Pumas

Raspberry Pi Idea

ITEM	Link	Part #'s	VOLTS	AMPS	WATTS	Weight	UNIT PRICE	#	TOTAL
Raspberry Pi Zero W	ef=sr_1_5?crid=194HNOM8XCSY4&keywords=Raspberry+pi&qid=169649693		5	0.5	2.5	9.07	$ 26.95	1	$ 26.95
Arducam MultiPlexer Adapter	https://www.amazon.com/gp/product/B07VD9XLYH/ref=ewc_pr_img_6?smid=A2IAB2RW3LLT8D&psc=1		5	2	10	18	$ 29.99	1	$ 29.99
Micro SD Card - 32GB	https://www.amazon.com/gp/product/B0C1Y87VT3/ref=ewc_pr_img_5?smid=A23RK926ENQ7Q6&th=1		0	0	0	<5	$ 6.99	1	$ 6.99
Camera 1 - IMX519 (80°) W/ Case	Y2KTAF4IC&keywords=arducam+autofocus+raspberry+pi+camera&qid=16964	IMX519	5			35.72	$ 29.99	2	$ 59.98
Camera 2 - IMX708 75°(D) Autofocus W/ Case	KTAF4IC&keywords=arducam+autofocus+raspberry+pi+camera&qid=1696494	IMX708	5			35.72	$ 45.99	2	$ 91.98
Camera 3 - IMX708 120°(D) Autofocus V3 W/ case	KTAF4IC&keywords=arducam+autofocus+raspberry+pi+camera&qid=1696494424&sprefix=arducam+autofocus+rasph	IMX708	5	0.5	2.5	35.72	$ 59.99	2	$ 119.98
~~Camera 4 - (Lens)~~	~~camera&pd_rd_i=B0867C9S2P&pd_rd_r=6c23c721-84cd-4e5d-ac65-7b6ae6975d62&pd_rd_w=ImGob&pd_rd_wg=4Oc~~	~~OV5647~~	~~5~~			44	~~$ 18.99~~	~~2~~	~~$ 37.98~~
REC Button			3.3	20 mA	0.06	<5	$ -	1	$ -
LED Indicator			3.3	~10 mA	0.03	<5	$ -	1	$ -
Rocker Switch			0	0	0	<5	$ -	1	$ -

OPTION	***TOTAL***	
Camera Option 1	**$123.91**	~75 g
Camera Option 2	**$155.91**	~75 g
Camera Option 3	**$183.91**	~75 g

Attached above is an extensive graph overviewing all required aspects of the RaspberryPi/Camera System. This graph aims to include Volts(v), Amps, Watts, Weight(g) and Price($) of all necessary components. Included were 4 different types of cameras as this graph was created and used before we had decided to move forward with the IMX708 Ultrawide(120) cameras. All other camera options are shown in a darker gray to make a distinction that they were not used in the final creation of our payload.

Then along with the cost of the demosat kit at $170.88.

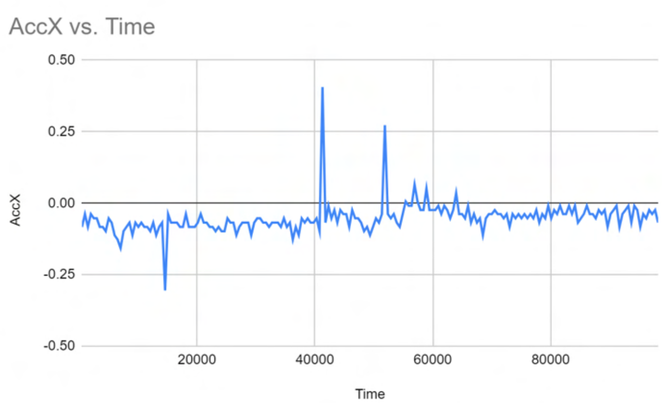
The final weight at launch was 460 grams, and was above our original weight of 351 grams due to complications with the camera system.

5.0 Test Plan and Results

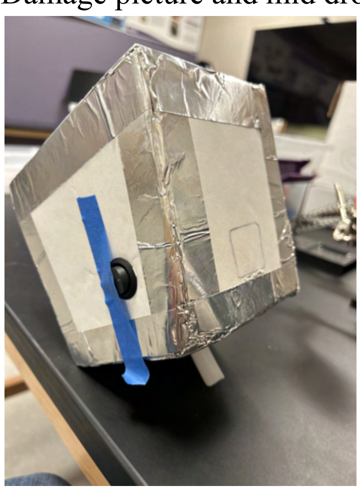
We planned to do six tests, the mandated drop test, cold test, stair test, vacuum test, whip test, and then our own stress test to test the longevity of the sensor and camera systems, unfortunately, we ran out of time to complete the longevity stress test and prioritized the mandated tests for launch.

5.1 The Drop Test

The first test we performed was the drop test, beyond a little bit of small damage (reference pictures below), we received good data (graph below) with the accelerometer and the other sensors.

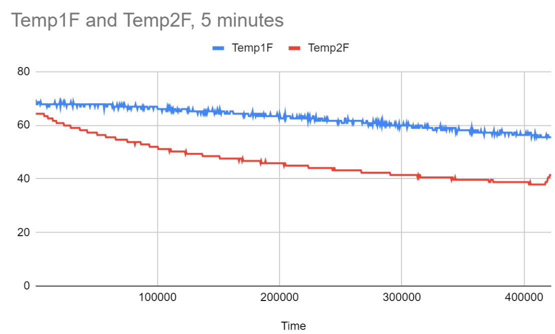


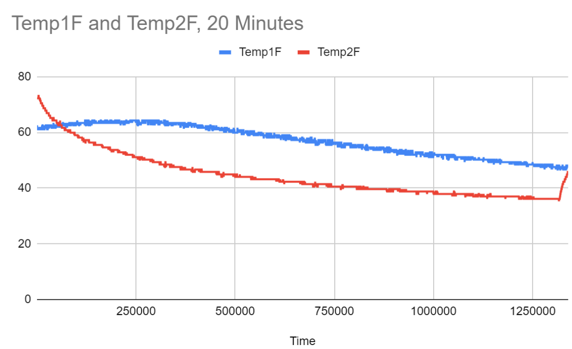
Damage picture and mid drop picture:




5.2 The Cold Test

The second test was the cold test, which we ran into a major setback. The demosat appeared to be operating with the light on designating power and the leds flashing to denote the recording data. After waiting 4 hours we checked the data, there was no data recorded. After that we realized a major flaw, which was also evident in previous demosat launches: alkaline batteries can fail at sending the correct voltage needed to run the arduino, lithium batteries are the way to go. After doing the test two more times for a 5 minute interval and 20 minute interval we received the following data.

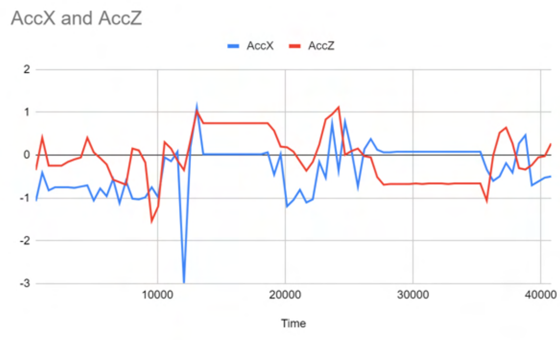




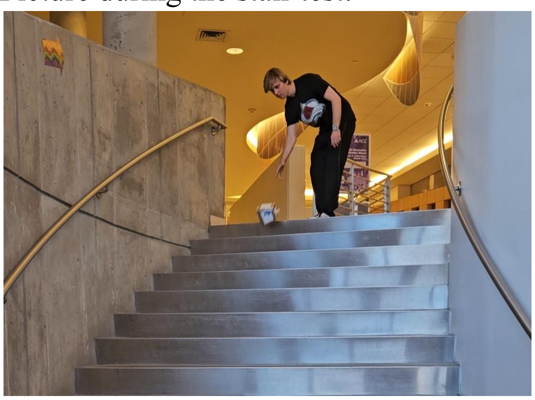
The Demosat during the cold test:

5.3 The Stair Test

The third test we performed was the stair test, which the payload performed nominally receiving the following data.

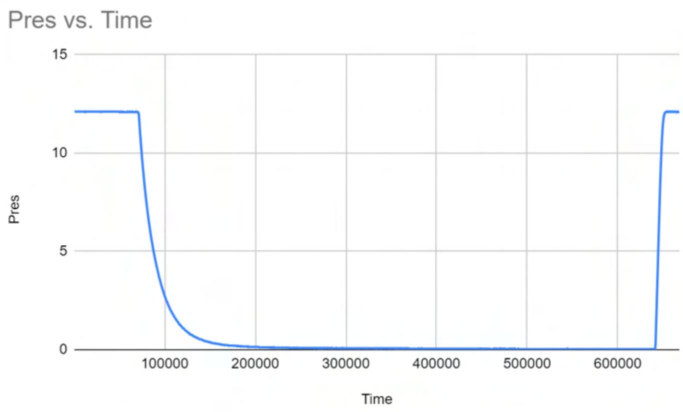


Picture during the stair test:

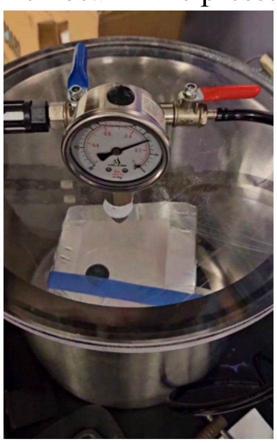


5.4 The Pressure Test

The fourth test was the pressure test in which we had little concern that the payload would not pass and we received the following data.

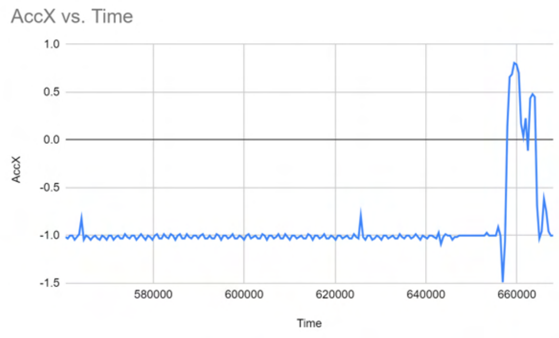


Demosat in the pressure container:

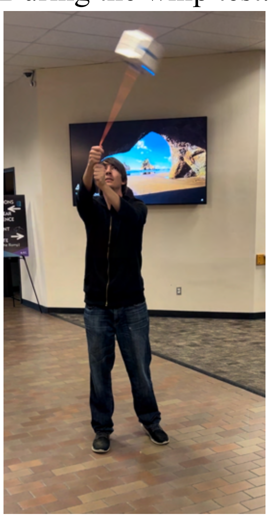


5.5 The Whip Test

The fifth and last test we performed was the whip test, where we had one malfunction, how we tied the test cord to the payload causing the flight tube to tear through the demosat, after retying and resecuring it with tape, the following data was recorded in which the constant circle of the craft is visible with the sharp acceleration back with the whip.



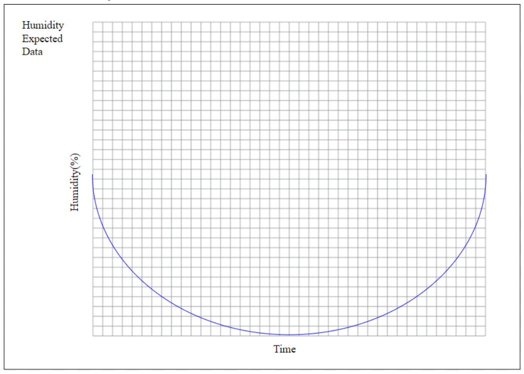
During the whip test:



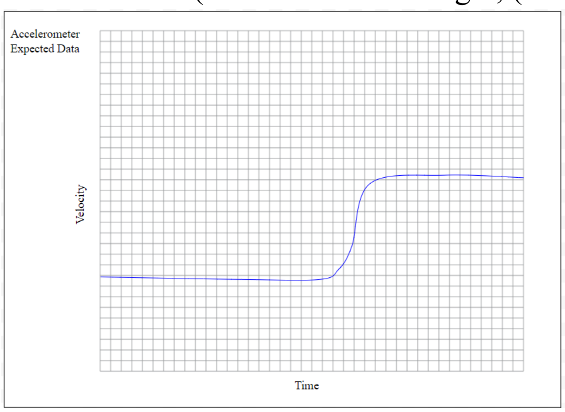
6.0 Expected Results

The expected results for this launch are rather straightforward. Before the camera system broke we expected to see pictures and then with the kit sensor pack we expected to see something similar to the graphs below.

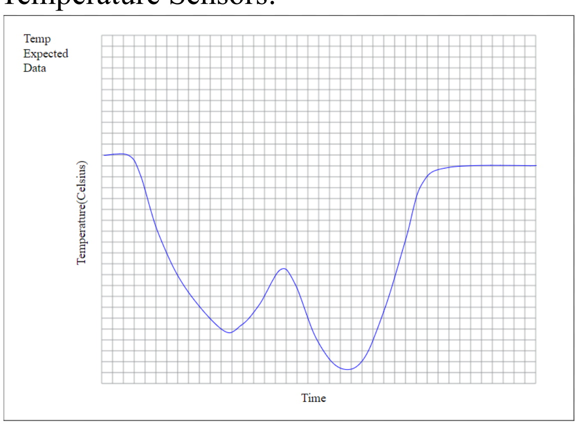
Humidity Sensor:



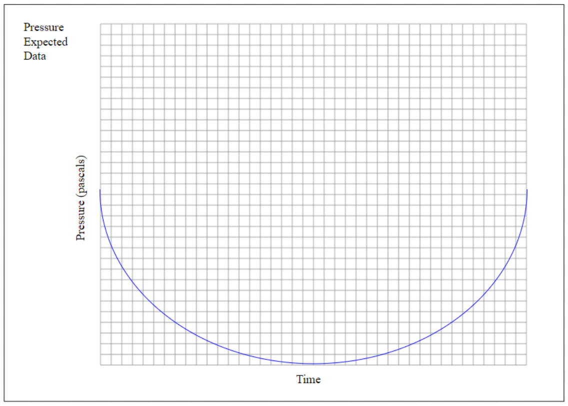
Accelerometer (at the start of the flight, (note: the line would be more jagged)):



Temperature Sensors:



Pressure Sensor:

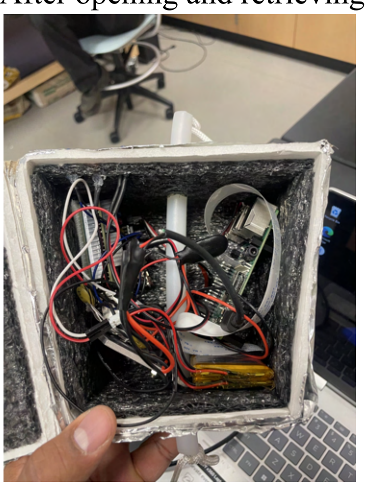


7.0 Launch and Recovery

With unfortunately only three of us (Colin, Daniel, & Marco) being able to attend to the launch we started off with a disadvantage, as we also left a little late and we were having malfunctions with the camera system and it was a mad rush in the vehicle on our way there to reseal the demosat and to update the camera system, unfortunately we ran out of time and the camera, the raspberry pi, and the batteries launched inoperational, with the system acting as a pseudo heater, with the demosat being overweight, but luckily another team was underweight. Colin was the payload handler and Daniel handled the tracking system for the balloon. After launch we drove to Anton, Colorado, and later to a cornfield with a half mile walk to recover the payload.

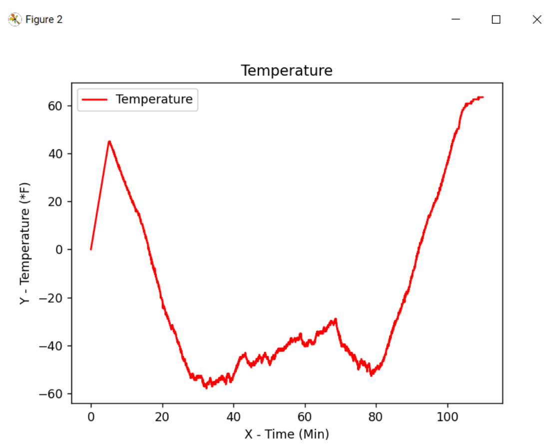
At the recovery site:

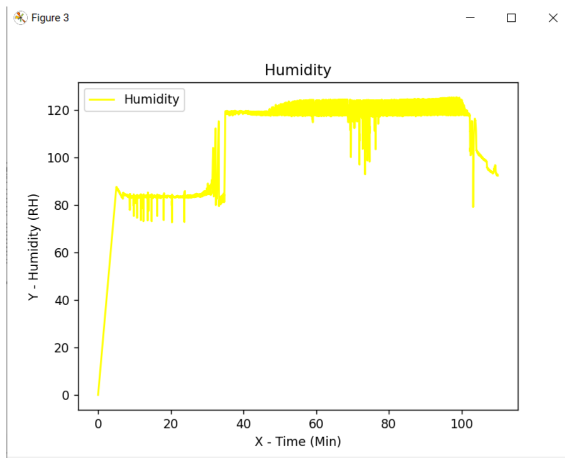
After opening and retrieving the data:

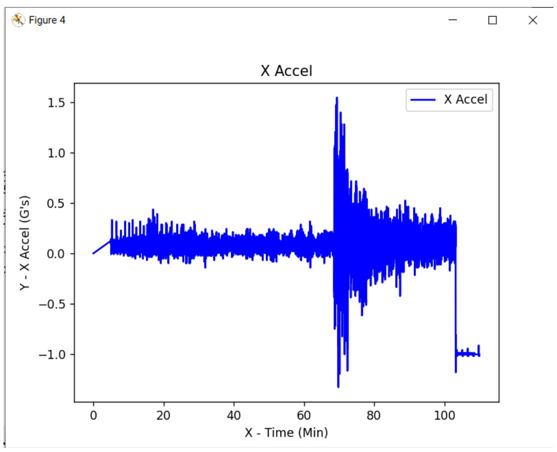


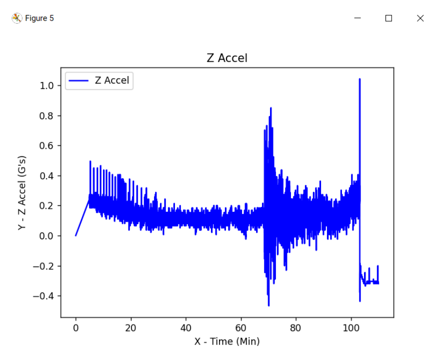
8.0 Results, Analysis, and Conclusions

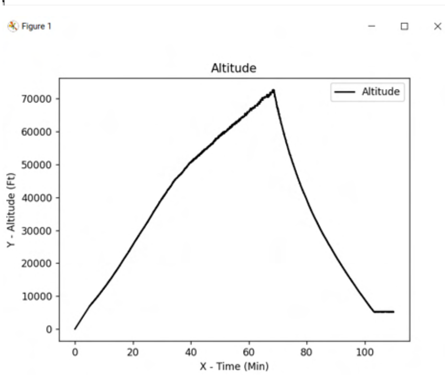
Working only with the on-board sensors that were sent in the package, we were able to collect a vast amount of data and use it to make predictions and changes for our next flight. Two of the most important sensors that were used were the pressure sensor and the accelerometers. Using the pressure sensor and by rearranging the barometric pressure formula to solve for altitude, you can get a rough estimate of how far up you are. The accelerometer also proved very useful for determining when the balloon popped, so we can time our experiments right and safeguard the more fragile pieces from the G forces experienced when the balloon bursts. The results that we got from the data were expected and matched up very well with what was described in some of the briefings. The balloon burst at about 79,000 feet and our data matches that very closely, giving a rough estimate of around 70-75,000 feet when the balloon burst. It is a little off because the data file only reads to 2 decimal places, and the barometric equation only assumes that the atmosphere is ideal, which in reality it is not. Our team wrote a simple program in python to analyze the 833KB of data we gathered during the flight, and using that we created graphs to track how the flight went. Below are some of the graphs that came directly from the data.











Based on the graphs, around the 70 minute mark, it is observed that there is a large spike in the accelerometers (blue graphs), and the altitude reaches a peak of around 75,000 feet. This indicates that the balloon burst, given by the drastic drop in altitude that follows the 70 minute mark. The temperature graph is as expected, dropping to almost -60 degrees fahrenheit at the peak altitude, and then slowly coming back up right before touchdown at the 100 minute mark. The why and what of our mission was answered, giving us a place to put our foot in the door in DemoSat. The data we got from this mission is very valuable for our next mission, where we can estimate the altitude and have a very good estimate of when the balloon pops based on acceleration and pressure data. We hope to use this for an experiment next semester in which we may have to open an airtight container in space, so it is critical to know the altitude and when the balloon pops. Due to some last minute changes, we were not able to get the cameras on our Raspberry Pi to work, and so we didn't get any of our own pictures from space. Next flight, we hope to remedy this and get 4k pictures if we can allot the time.

9.0 Lessons Learned

Power Source Challenges and Voltage Requirements:

Recognize the critical importance of power source compatibility and voltage requirements, especially when integrating devices like the Raspberry Pi into the payload. In future projects, thoroughly assess the power needs of each component, ensuring that power sources are compatible and can provide stable voltage levels throughout the mission. Consider the use of voltage regulators or dedicated power management systems to address specific voltage requirements and prevent potential issues arising from power fluctuations.

Lesson Learned in the Integration of Raspberry Pi in Headless Mode:

To say there were MANY lessons learned in using a RaspberryPi for the first time is a complete understatement. Taking on this Linux based system running as its own computer instead of its Arduino/microcontroller counterparts had an extreme and well versed learning curve. First, was understanding different forms of Raspbian OS and their dedicated use and means of integration. This second, came with understanding how to connect to a board like the RaspberryPi Zero W without a dedicated keyboard, mouse, or monitor. This meant that we had to connect and run the board solely through a host wifi server and using SSH via Windows Powershell (Terminal). Once connected via SSH, it would be important to then connect via VNC server so we had graphical remote desktop access. All of these lessons alone presented their individual struggles and challenges that made the entirety of this camera system extremely difficult. This later came in combination with the difficulties of the Picamera2 library. The Picamera2 library is a Python-based library that is required to be used to run any Arducam accessories with Python. It was a recently developed library with little documentation and little online resources for help that made coding this project in python especially difficult. Then came the challenge of making all of this happen and run the required python script on startup of the RaspberryPi. After getting this issue figured out, later troubleshooting revealed our power source was an issue. This led to a fun 6 1/2 hour lab session of learning all that there is to learn about circuitry. Using newly learned voltage calculations and other things, we used a voltage regulator, heat sink, and different forms of batteries to form what looked like a homemade bomb. But, at the end of the day we also made a working power source.

Every single one of these tasks required a great amount of learning. It was not always a fun time, but this amount of dense new material left exponential room to grow and learn which should not be taken for granted.

Though the camera system decided to produce black screens 4 hours before launch, creating a large issue with a small amount of time to solve. I would not denote any part of this learning experience.

Post-Mission Data Analysis Lessons Learned:

Plan for comprehensive post-mission data analysis. Develop protocols for handling large volumes of data collected during the mission and establish a systematic approach to interpret the results. This will help in extracting valuable insights, identifying areas of improvement, and informing future design iterations.

Testing Protocols Lesson Learned:

In future projects, allocate sufficient time for thorough testing as soon as possible, including simulations of worst-case scenarios, with the payload mission ready as of the Flight Readiness Review. This will give you sufficient time to document and analyze the results meticulously, use the insights gained to make informed design adjustments and improvements for enhanced system reliability and performance.

10.0 Conclusion and Message to Next Year

In conclusion, the team's collaborative efforts resulted in the successful construction, testing, launching, and retrieval of an experimental payload, as well as a positive learning experience for all members. This semester we were presented with many challenges over the course of this project including time management, planning, social dynamics and the balancing of academic coursework with club activities. However, the team's solid communication and mutual respect for one another and the project at hand allowed us to mitigate many of the barriers in the way of our project's success. Future club members should be aware that the DEMOSAT program is a challenging endeavor, but exceptionally worthwhile opportunity. Being able to cooperate as a team in a group setting, working on a significant project with peers, and learning technical skills are invaluable experiences that will continue to pay off for many years to come. And now, having developed a plethora of skills and a better understanding of our individual strengths, we are planning for an even more epic next semester that we believe will make an impact on the Colorado Space Grant program for years to come.

Remember, often, it is not the complexity of the obstacles our world presents that limit the heights humankind may reach, but the limits of will it allows most to be determined by its wants and not its needs.

11.0 Acknowledgements

We would like to thank Jacob Tipsword for mentoring us through this Demosat project. To also thank Jennifer Jones and Jacob Wikowsky for additional questions and support. As well as Bernadette Garcia Galvez and all of Colorado Space Grant Consortium, and Edge of Space Sciences.