Final Report

Team 4: IUBSAT

Table of Contents

1. Introduction

1.1. Team Introductions

Figure 1.1.1: Team members from left to right, top to bottom: Will Brenneke, Annabel Brinker, Joseph Patus, Gourav Pullela, Lucas Snyder, and Caleb Vrydaghs

Will Brenneke is the system manager. His concentration is in Computer Engineering with experience in C, Python, System Verilog, and VHDL. He has interned at NSWC Crane and currently works in the Systems Assurance and Integrity Lab at Indiana University.

Annabel Brinker is the presentation manager. Her concentration is Computer Engineering. She has expertise in C, C++, Python, Verilog, and GNURadio. She has had previous experience with an internship at NSWC Crane, Navient, and various research work.

Gourav Pullela is the prototype integrator. His concentration is Computer Engineering, and he has experience in High Performance Computing, C, C++, Python, and SOLIDWORKS. He has previously worked as an undergraduate instructor at Indiana University Bloomington.

Lucas Snyder is the project coordinator. His concentration is Bioengineering, and he has experience with High-Performance Computing, Molecular Dynamics, Python, and C. He has previously worked at NASA Goddard Space Flight Center and Purdue University’s Rosen Center for Advanced Computing.

Caleb Vrydaghs is the report manager. His concentration is in Computer Engineering with various experience in C, C++, PHP, Verilog, Python, and GNURadio. He has previously worked for Enersys through Alpha Technologies as an Embedded Firmware Developer.

1.2 Resource Management

The team will commit 30-man hours per week to working on the project. This commitment includes class time, group meetings, and individual work time. The total project budget is $200 of capstone budget from class and an undisclosed amount from our project sponsor.

The available resources that the team has on-hand include FDM, SLA, SLS Additive Manufacturing machines, Laser Cutters, Electronic Testing equipment. The team was also given items from previous balloon flights. This equipment includes a power distribution board, battery distribution packs x8, battery harness x2, 5000maH battery, 1m parachute x3, 1.5m parachute, helium hose, eagle flight computer 1x rev 1 and 1x rev 5, paracord, 1500g HAB, audio beacon, beacon, radio bug antenna, pi camera, combined sensor board, GoPro Hero 3+, Bruton camera battery, clear case, mount x2, rechargeable battery and charger, rubber bands, wires, bolts/screws/etc, electrical tape, tape measure x2, tweezers, zip ties, and more.

1.3 Weekly Meetings

For the first semester the team meets during class every Tuesday and Thursday from 3:00p to 4:15p. In addition to this meeting the team meets every Sunday from 11:00a to 2:00p. During class times time is devoted to required deliverables so that one individual team member is not solely responsible for creating and submitting these. The team also occasionally meets with our project mentor during class on Thursdays. The meeting on Sunday is devoted to project design and development as well as assigning tasks to members for the following week.

For the second semester the team meets every Saturday 2p- 5p in addition to class every Tuesday and Thursday 9:45a- 11:00a. About halfway through the semester the team determined additional meeting time was necessary. An additional meeting time was added on Friday 12p- 2p. The Friday meeting was determined on a week-to-week basis whether it was needed.

1.4 Communication

The team primarily communicates through Teams. The group also created a group chat for more immediate communication between the members. Files, resources, and deliverables are kept in the Teams channel for easy access to every member. Communication with the teams' project sponsor is done through email and his slack.

2. Motivation

2.1 Mission Statement

This project is aimed at advancing the IUBSAT suborbital payload structure, utilizing COTS electronics, and involving rigorous testing to perform an imaging mission during a total solar eclipse at high altitudes to emulate suborbital conditions.

2.2 Project Significance

We are in the path of totality of the total solar eclipse taking place on April 8, 2024. This will allow us to simulate extreme temperatures below -50 degrees Celsius when the payload is at its peak altitude during totality, which will help improve our understanding of reliable electronic systems.

3. Project Information

3.1 Key Features

The designed payload has two primary goals: Imaging and Communication. Imaging will be managed by a 4-camera array of wide angled sensors to approximate 360-degree video as closely as possible. Through this the intention is to maximize potential time for imaging of the eclipse by covering every angle. Communications are handled as a down-link from the CubeSat to the ground station to receive encoded video packets to have live video output.

3.2 Bill of Materials

Team 4: IUBSAT
Final Bill of Materials

#	Component	Nominal Value (w/ units)	Distributor	Part # (Distributor)	Part # (Manufacturer)	Manufacturer	Package	Package Type	Price Each	Qty	Price Total	Part Link	Flown on 04/08
1	Arducam 4 Camera Breakout Board	1	UCTRONICS	B0396	B0396	Arducam	N/A	N/A	$ 140.00	2	$ 280.00	https://www.uctronics.com/camera-	Yes
2	Wide Angle Lenses	1	UCTRONICS	B0180	B0180	Arducam	N/A	N/A	$ 10.00	8	$ 80.00	https://www.uctronics.com/arducar	No
3	Raspberry Pi 4B	1	Amazon	4296	Raspberry Pi 4 Model B 4GB	Raspberry Pi	N/A	N/A	$ 55.00	2	$ 110.00	https://www.adafruit.com/product/4	Yes
4	NiceRF RF4463F30	1	Tindie			NiceRF	Module	SMD	$ 8.00	5	$ 40.00	https://www.nicerf.com/fsk-front-en	No
5	STM32 F411 "Blackpill"	1	Adafruit	4877	STM32F411	STM	N/A	N/A	$ 30.00	2	$ 60.00	https://www.adafruit.com/product/4	No
6	ST-Link Programmer	1	Adafruit	2548	ST-link V2	STM	N/A	N/A	$ 13.00	2	$ 26.00		No
7	SDRplay RSP1A	1	HAM Radio Outlet	RSP1A	SDRplay RSP1A	SDRplay	N/A	N/A	$ 110.00	2	$ 220.00	https://www.sdrplay.com/rsp1a/	No
8	RTL-SDR v3	1	Amazon		RTL-SDR		N/A	N/A	$ 34.00	1	$ 34.00	https://www.amazon.com/RTL-SDR-	Yes
9	Arrow 2 (7-element Yagi)	1	Arrow Antenna	Model 440-7	Model 440-7	Arrow Antenna	N/A	N/A	$ 128.00	2	$ 256.00	https://www.arrowantennas.com/ar	Yes
10	BME680	1	Adafruit	3660	BME680	Adafruit	Carrier Bo	TH	$ 19.00	2	$ 38.00	https://www.adafruit.com/product/3	Yes
11	LSM303AGR	1	Mouser	4413	LSM303	Adafruit	Carrier Bo	TH	$ 12.50	2	$ 25.00	https://www.adafruit.com/product/4	Yes
12	SPOT Trace	1	SPOT			Saved by SPOT	N/A	N/A	Provided	1	$ -	https://www.findmespot.com/en-us	Yes
13	StratoTrack	1	High Altitude Science			High Altitude Sc	N/A	N/A	Provided	1	$ -	https://www.highaltitudescience.co	Yes
14	StratoTrack Programming Cable	1	High Altitude Science	Strato USB Cable	Strato USB cable	High Altitude Sc	N/A	N/A	$ 39.00	1	$ 39.00	https://www.highaltitudescience.co	Yes
15	Audio Beacon	1	?				N/A	N/A	Provided	1	$ -	N/A	No
16	VGE Battery Pack	1	Amazon				N/A	N/A	Provided	1	$ -	N/A	No
17	Anker PowerCore Slim 10K PD	1	Amazon			Anker	N/A	N/A	$ 20.00	1	$ 20.00	https://www.anker.com/products/a1	Yes
18	1500 g Balloon	1	High Altitude Science	Weather Balloon 1500g	Weather Balloon 1500g	High Altitude Sc	N/A	N/A	$ 159.00	2	$ 318.00	https://www.highaltitudescience.co	Yes
19	1500 g Balloon	1	Kaymont Consolidated Ir	HAB-1500		Totex	N/A	N/A	$ 115.00	1	$ 115.00	https://www.kaymont.com/product-	No
20	5ft Parachute	1	Rocket Man	5ft m RocketMan Parachute	RocketMan	Rocket Man	N/A	N/A	$ 55.00	1	$ 55.00	https://the-rocketman.com/recovery	Yes
21	Nylon Rope	1	?				N/A	N/A	Provided	1	$ -	N/A	Yes
22	Carabiners	1	?				N/A	N/A	Provided	6	$ -	N/A	No
23	3 ft SMA Jumper	1	Amazon				N/A	N/A	$ 7.29	1	$ 7.29	https://www.amazon.com/dp/B081I	Yes
24	1 ft SMA Jumper	1	Amazon				N/A	N/A	$ 5.00	1	$ 5.00	https://www.amazon.com/dp/B07P	No
25	Latex Gloves	1	Kroger				N/A	N/A	$ 8.00	1	$ 8.00	https://www.kroger.com/p/kroger-di	No
26	Waterproofing Tape	1	Menards			Duck Brand	N/A	N/A	$ 10.00	1	$ 10.00	https://www.amazon.com/Duck-Wa	Yes
27	Gas- Travel expense	1	Various				N/A	N/A	$ 65.00	1	$ 65.00	N/A	Yes
28	LoRa Expansion for Pi	1	UPUTRONICS				Carrier Bo	GPIO	$ 44.00	1	$ 44.00	https://store.uputronics.com/index.	Yes
29	BNC to SMA Cable (50 Ohm, 4ft)	1	Amazon			uxcell			$ 9.50	2	$ 19.00	https://a.co/d/22vRyPN	Yes
30	512GB MicroSD Card	1	Amazon	EVO Select Micro SD	EVO Select Micro SD	Samsung	N/A	N/A	$29.99	1	$ 29.99	https://www.amazon.com/SAMSUN	Yes
31	Nagoya UT-72G	1	Amazon	Nagoya UT-72G	Nagoya UT-72G	BTECH	N/A	N/A	$34.89	1	$ 34.89	https://www.amazon.com/Authentic	No
32	Watt Power Meter	1	Amazon	Meecher					$ 12.99	1	$ 12.99	https://www.amazon.com/Electricity	No
33	Stubby Antenna	1	Amazon	TWAYRDIO 15.4-Inch Whip Antenna, SM/	TWAYRDIO 15.4-Inch Whip Antenna, SM/	TWAYRDIO			$ 16.99	1	$ 16.99	https://www.amazon.com/Antenna-	Yes
											$ 1,969.15	Grand Total	

Figure 3.2.1 Bill of Materials

3.3 System Overview

Stephen Downward in Canada has an open-source project where he worked on a part of a CubeSat launched in September of 2023. He worked on "Real Time Video From a High Altitude

Balloon" which is in line with the team's project goals. This system is modeled on his outline to create a legal, cost-effective, and feasible design.

CubeSat: Combination of Pi4B for video processing and hardware encoding, an ArduCam CAMarray for 360-degree video, a STM32 Blackpill for video packeting to be transmitted, RF4463 transceiver to send video packets down to our ground station. System power will be handled by one primary battery pack at 5V and the Pi will be able to distribute system power off of its power rail. The CubeSat also contains a small sensor array consisting of a BME680 and LSM303AGR that we will use to sample flight and atmospheric data throughout the duration of the flight.

Ground Station: SDRplay SDR hooked into an Arrow II Yagi Antenna will be the reception hardware that will run into a laptop to receive and decode packets.

3.4 Background Information

3.4.1 NASA State-of-the-Art Small Spacecraft Technology

NASA provided a guide called *State-of-the-Art of Small Spacecraft Design* which gives an in-depth view of the factors involved in State-of-the-Art Small Spacecraft Design.

For example, in section 1.3 of the guide, NASA outlines their Technology Readiness Level (TRL), a metric which can be used to measure prototype readiness. Within the scope of our project, we would likely at most achieve a TRL-7, as we are only launching a single prototype to a test environment.

The Small Spacecraft Avionics (SSA) (8.3) section outlines a list of requirements for reliable microsystems and provides a list of verified onboard computing systems according to their TRL value with information on technical specification such as whether the device is Radiation Hardened, which can be very useful in this scenario.

The RF Communications (9.2) section provides information on Radio Frequency (RF) communications along with information on relevant frequency bands, which in our case would be VHF (Very High Frequency) and UHF (Ultra High Frequency), with very useful information such as a system architecture example and other design considerations for a communications system.

3.4.2 ESA 3D printing CubeSat bodies for cheaper and faster missions

The European Space Agency (ESA) has explored 3D printing CubeSat bodies for faster and more cost-effective missions. The cost of Additive Manufacturing using engineering grade materials such as PEEK (polyether ether ketone) that can withstand high temperatures, with the emergence of newer technologies has become more efficient than ever, allowing anyone to 3D print a CubeSat body, rapidly prototyping it, and deploying it to test. The allows for much faster prototyping and with high performance materials, allows for direct deployment, with an end-to-end part.

3.4.3 Relevant Equations

Link Margin/Budget: The Received Power at the Rx side in dBm -

$$P_{RX} = P_{TX} + G_{TX} - L_{TX} - L_{FS} - L_M + G_{RX} - L_{RX}$$

Where,

P_{RX} = Received Power (dBm)

P_{TX} = Transmitter Power Output (dBm)

G_{TX} = Transmitter Antenna Gain (dBi)

L_{TX} = Losses from Transmitter (cable, connectors etc.) (dB)

L_{FS} = Free-Space Loss (dB)

L_M = Misc. Losses (fade margin, polarization misalignment etc.) (dB)

G_{RX} = Receiver Antenna Gain (dBi)

L_{RX} = Losses from Receiver (cable, connectors etc.) (dB)

Figure 3.4.3.1 Link Budget Equation

Free Space Path Loss: It is the difference in magnitude between two isotropic radiators - d (m), f (H), and c (speed of light in m/s)

$$\begin{aligned}
\mathrm{FSPL(dB)} &= 10\log_{10}\left(\left(\frac{4\pi df}{c}\right)^2\right) \\
&= 20\log_{10}\left(\frac{4\pi df}{c}\right) \\
&= 20\log_{10}(d) + 20\log_{10}(f) + 20\log_{10}\left(\frac{4\pi}{c}\right) \\
&= 20\log_{10}(d) + 20\log_{10}(f) - 147.55,
\end{aligned}$$

Figure 3.4.3.2 Free Space Path Loss Equation

Using the free space path loss equation to determine the maximum theoretical distance.

$\mathrm{FSPL} = 20\log_{10}(d) + 20\log_{10}(f) + 20\log_{10}(\frac{4\pi}{c}) - G_{Tx} - G_{Rx}$

Solving for d (distance): $d = 10^{90+G_{Tx}+G_{Rx}-(20\log_{10}(f*10000000))-(20\log_{10}(4pi)/c)))/20)}$

D= 219.34 Km

We wrote a simple python program that would calculate distance for us when given transmission frequency and transmitting and receiving gain.

```
[11]  import math

[12]  pie = math.pi
      c = 2.99792458e8
      print(pie)
      print(c)

3.141592653589793
299792458.0

[15]  f = float(input("Enter frequency in MHz: "))
      gtx = float(input("Enter transmitter gain (db): "))
      grx = float(input("Enter receiver gain (db): "))

Enter frequency in MHz: 433
Enter transmitter gain (db): 30
Enter receiver gain (db): 12

      d = 10**((90+gtx+grx-(20*math.log10(f*1000000))-(20*math.log10((4*pie)/c)))/20)
      print(d/1000,"km")

219.34257749167995 km
```

Figure 3.4.3.3 Python Program for Transmission Distance Calculation

3.5 Power Analysis

We collected typical power consumption data for all payload components using datasheets and other resources. Below are the estimated power requirements for the payload for the duration of the flight. We found during isolated testing that our system drew between 7-8.5W and could run for 5.5 hours off the 10,000MaH battery bank we used.

Part	Quantity	Max Current (mA)	Estimated Current(mA)	Voltage (V)	Time (Hours)	Estimated Power(Wh)
Camera System						
Arducam 4 Camera Breakout Board	1	800	200	5	3	3
Flight Computer						
Rasberry Pi 4B	1	3000	800	5	3	12
TX Model						
STM32 Black Pill	1	1000	50	3.3	3	0.495
Nice RF RF4463F30	1	550	500	5	3	7.5
Sensor Array						
BME680	1	20	5	3.3	3	0.0495
LSM303AGR	1	10	2	3.3	3	0.0198
Payload Power Consumption (Wh)	23.0643					

Figure 3.5.1 Power Analysis Spreadsheet

3.6 Subframe Design

3.6.1 Subframe Design Implementation

Our subframe design was an ongoing challenge to find a design that fits the 2U aluminum chassis (Figure 3.6.1.1) made by the previous group that worked on this project. Caleb and Gourav both made contributions to this to design and fabricate an effective model. In the end Caleb due to time constraints designed and printed the teams last few subframe revisions to neatly contain all electronics into a subframe that can easily slide out of the aluminum chassis so that the payload can be easily assembled and disassembled.

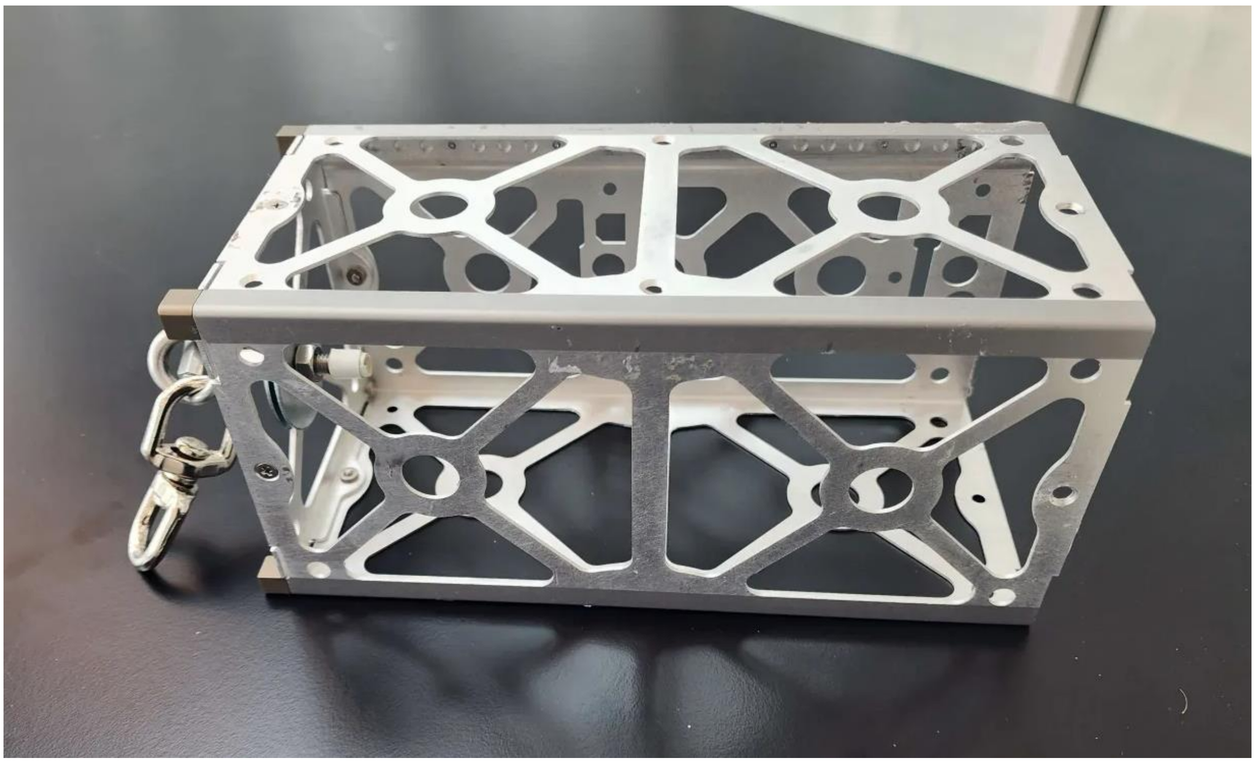


Figure 3.6.1.1 2U Aluminum CubeSat Frame

3.6.2 Subframe Design Iterations

3.6.2.1 Subframe REV-001

Taking inspiration from ESA and Princeton's CubeSat designs, our team is working on developing a CubeSat subframe that incorporates power and data buses into the chassis itself. Our goal is to accomplish this using an engineering grade thermoplastic, PC (Polycarbonate) and a highly conductive all-metal filament manufactured by Kupros Inc. called Cu-29. Polycarbonate has a high heat deflection temperature of 135°C (@ 0.45 MPa/66psi) making it suitable to work with other high temperature and energy materials. For this project, ESD-safe (Electrostatic Discharge Safe) PC will be used as a substrate with the Cu-29 filament. Cu-29 has excellent conductive properties, largely due to it being made with Tin based carrier filament that is combined with copper nanoparticles to maximize conductivity and printability. The resistivity of Cu-29 is 1.56E-07 Ωm (compared to 4.90E+14 Ωm for PEEK), which is orders of magnitude higher. This would generally come with a high cost, but Kupros Inc. was gracious enough to provide our team with several hundred grams of filament for testing in exchange for data and relevant feedback on their filament.

Below is an image of the first iteration of our subframe layer. The specific layer of the Subframe design pictured is the Flight Controller/MCU Layer of the CubeSat payload. This layer will be fabricated to test the properties of Cu-29, and its properties in conjunction with the ESD-safe PC being used as the substrate for the Subframe design.

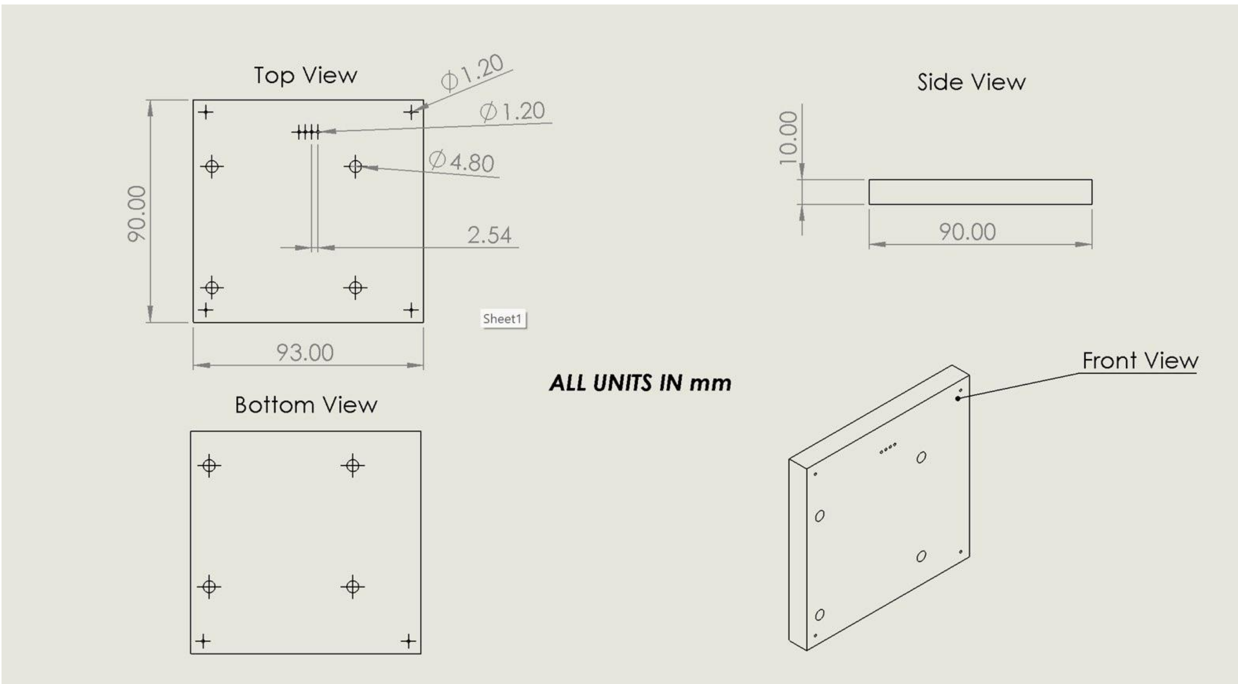


Figure 3.6.2.1.1 REV-001 Subframe Engineering Drawing

3.6.2.2 Subframe REV-002

We decided to pivot our design as we were unable to make headway with the 3D printing equipment we were using for multi-material 3D printing. The printer we were using (Creality Sermoon D3 Pro) and the printer and slicer came with some learning curves as they were relatively new and lacked documentation and technical support. We were not able to print ESD-safe PC or the Cu-29 filament, as we were unable to get both the extruders functioning properly.

We modified our design to separate our flight computer and components like our imaging system (ArduCam Camera Array) and power (battery pack) to be flexible as we chose our parts.

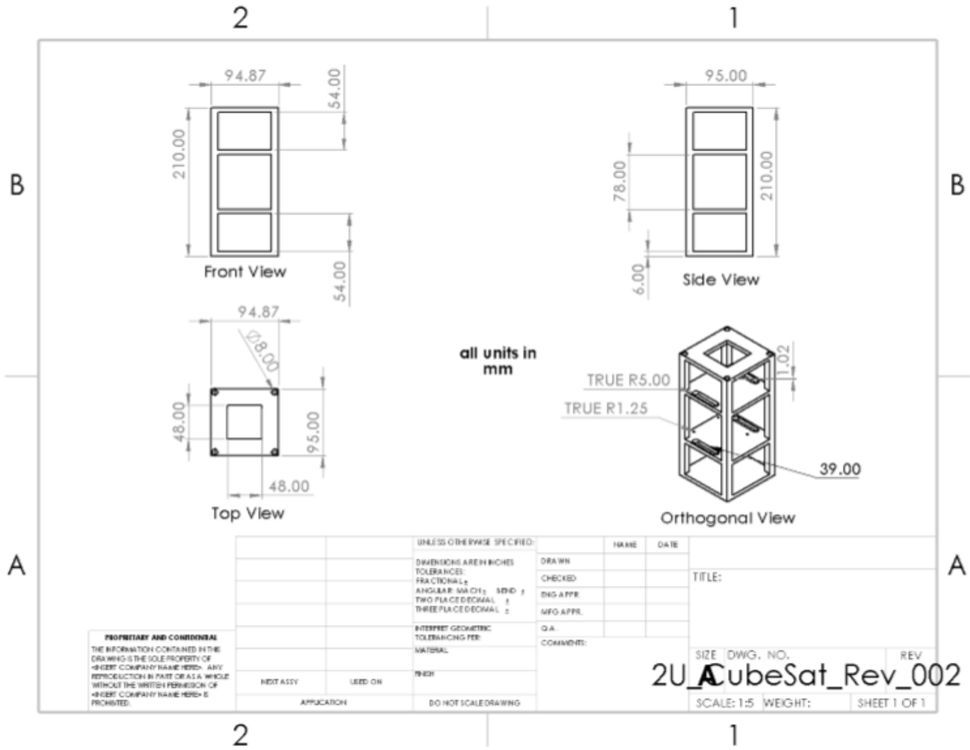


Figure 3.6.2.2.1 Subframe REV-002 Engineering Drawing

3.6.2.3 Subframe REV-003

We made some adjustments to our previous revision, moving around camera mounts so that we could get a wider range of coverage around the payload. We set it up in such a way that we could mount cameras facing four directions while also safely mounting our power and flight computer components.

With this iteration of the subframe, we had some issues with tolerances of mounting holes and overall printability of the design, as it was designed into a single piece. We were able to get around some of these issues while slicing the parts, but the software and printers were only able to do so much.

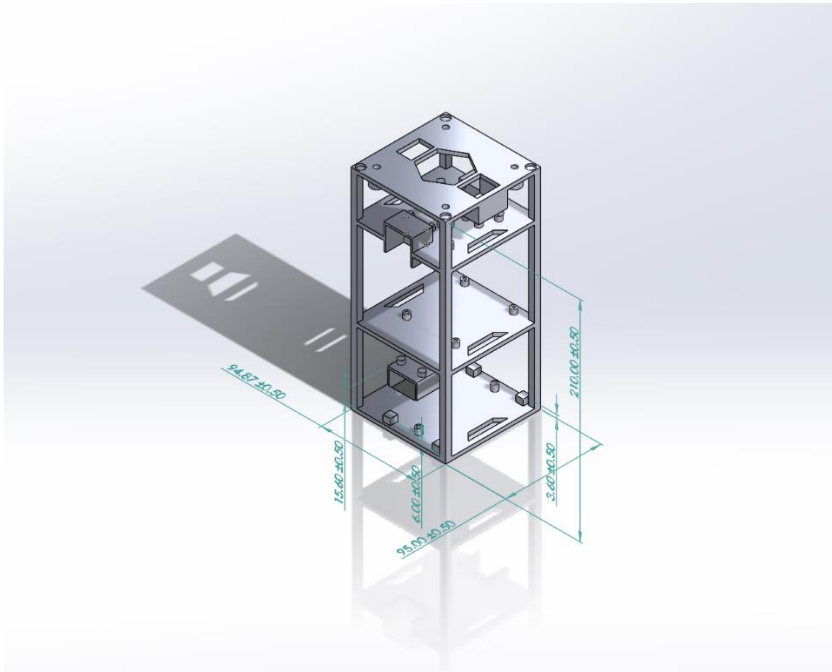
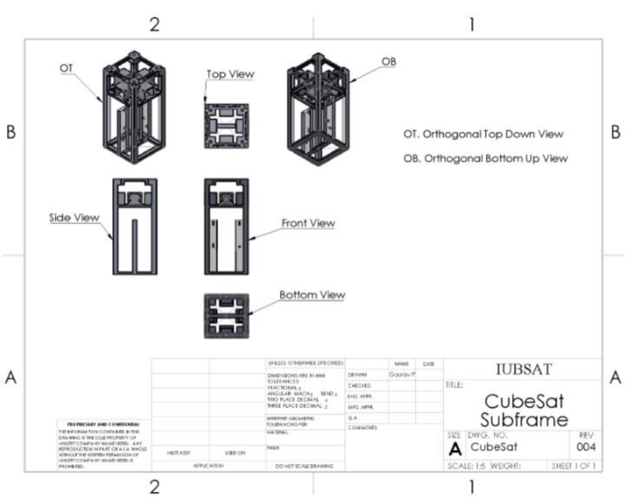


Fig 3.6.2.3.1 Subframe REV-003 Orthogonal View

3.6.2.4 Subframe REV-004

This design iteration we were able to fix the issues with screw tolerances but there were still some issues with printability – we attempted to use Polycarbonate and Nylon-12 (Carbon Fiber Reinforced), but the personal printers we were using were not in a controlled environment, leading to some print failures. We were able to finalize a design in terms of component placement and were able to place most of our parts without issue.



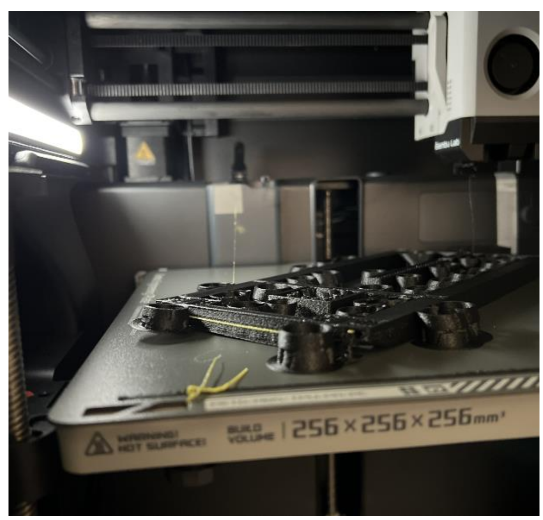


Figure 3.6.2.4.1 (*LEFT*) Subframe REV-004 Engineering Drawing (*RIGHT*) Print failure due to warping

3.6.3 Final Subframe Design

Below is a live 3D model of our subframe that readers can manipulate if this report is opened as a .docx. The actual .stl file will be included in our sensor codebase.

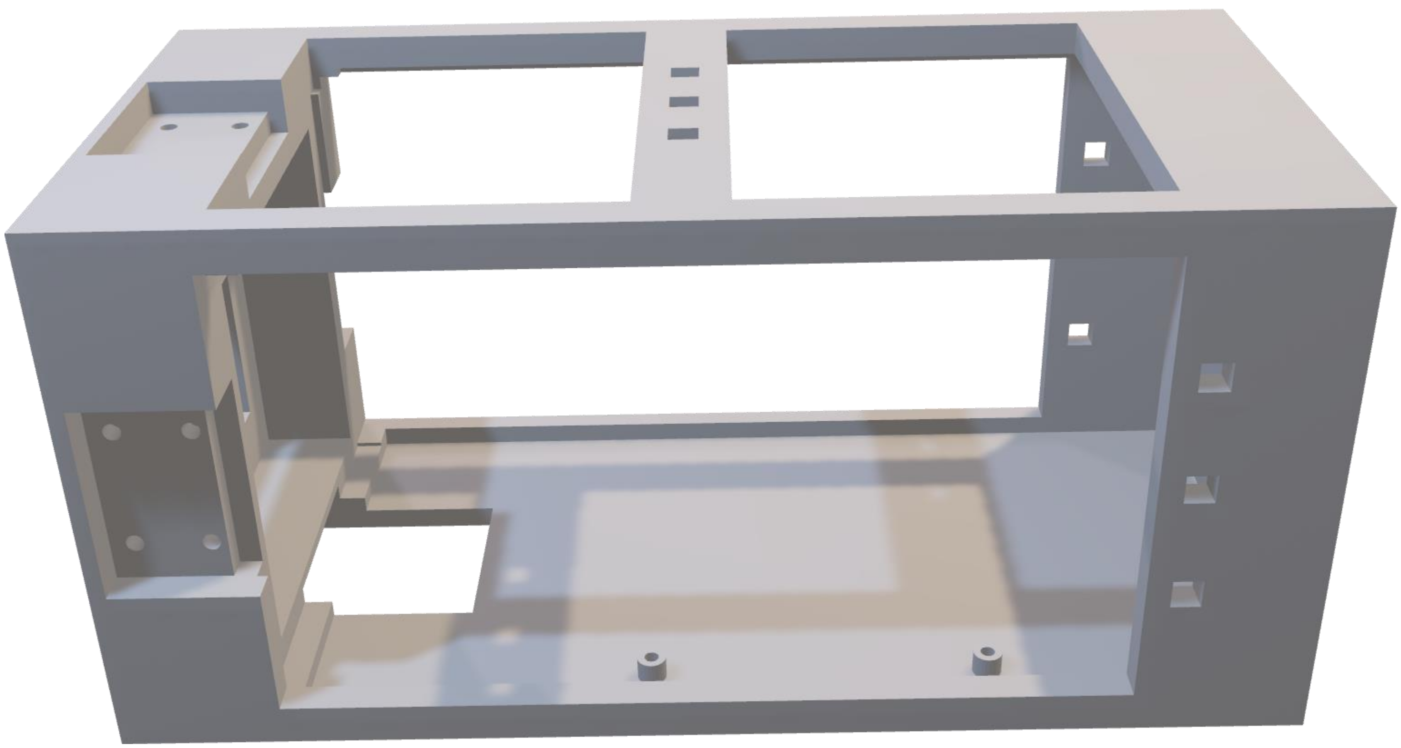


Figure 3.6.3.1 Subframe Engineering Model

4. Gantt Chart

4.1 Semester One

TASK TITLE	START DATE	DUE DATE	% OF TASK COMPLETE
Payload Planning			
Detailed Block Diagram	9/1/23	10/30/23	100%
Parts Summary	9/1/23	10/30/23	100%
Parts comparison	10/1/23	10/15/23	100%
Part Review	11/1/23	11/23/23	100%
Order Parts	11/1/23	11/30/24	100%
Hardware Testing			
Sensors (temp/ pressure)	12/1/23	2/30/24	30%
Camera/s	12/1/23	2/30/24	0%
Communication system	2/1/24	3/15/24	0%
Batteries	2/1/24	3/15/24	0%
Flight Computer	2/1/24	3/15/24	0%
Non-CubeSat Hardware Parts			
Balloon	1/1/24	1/30/24	0%
Cable	1/1/24	1/30/24	0%
Parachute	1/1/24	1/30/24	0%
Tracking	1/1/24	1/30/24	0%
Order Parts	2/1/24	2/15/24	0%
CubeSat Prototype - PDR			
Documentation	9/1/23	4/30/24	50%
CAD Design of Sub-frame	9/1/23	12/30/23	50%
Prototype of Internal Hardware	1/1/24	3/30/24	0%
Fabrication of the Subframe	1/1/24	1/30/24	0%
Assembly of Prototype	2/1/24	3/30/24	0%

Figure 4.1.1 Gantt Chart - Semester One

4.2 Semester Two

TASK TITLE	START DATE	DUE DATE	% OF TASK COMPLETE
Payload Planning			
Detailed Block Diagram	9/1/23	10/30/23	100%
Parts Summary	9/1/23	10/30/23	100%
Parts comparison	10/1/23	10/15/23	100%
Part Review	11/1/23	11/23/23	100%
Order Parts	11/1/23	11/30/24	100%
Hardware Testing			
Sensors (temp/ pressure)	12/1/23	2/30/24	30%
Camera/s	12/1/23	2/30/24	0%
Communication system	2/1/24	3/15/24	0%
Batteries	2/1/24	3/15/24	0%
Flight Computer	2/1/24	3/15/24	0%
Non-CubeSat Hardware Parts			
Balloon	1/1/24	1/30/24	0%
Cable	1/1/24	1/30/24	0%
Parachute	1/1/24	1/30/24	0%
Tracking	1/1/24	1/30/24	0%
Order Parts	2/1/24	2/15/24	0%
CubeSat Prototype - PDR			
Documentation	9/1/23	4/30/24	50%
CAD Design of Sub-frame	9/1/23	12/30/23	50%
Prototype of Internal Hardware	1/1/24	3/30/24	0%
Fabrication of the Subframe	1/1/24	1/30/24	0%
Assembly of Prototype	2/1/24	3/30/24	0%

Figure 4.2.1 Gantt Chart - Semester Two

5. Similar Projects

5.1 Princeton University TigerSats CubeSat Kit

The Princeton University TigerSats program has released a novel CubeSat design, using threaded aluminum rods to function as pairs of power and data buses respectively. Two of the threaded rods are 5V and Ground, and the other pair are SDA and SCK for I2C communication which can be utilized by mounting PCBs of different embedded assemblies within the CubeSat prototype. This reduces the overall mass of the CubeSat design and is a unique method to handle device I/O and power transfer. The CubeSat Kit is entirely built using PC104 embedded elements and has several other innovative components that allow the prototype to be more versatile.

5.2 Real Time Video from High Altitude Balloon – Stephen Downward

Stephen Downward developed a video streaming system using software-defined radio over the 70cm (430MHz) band. His system used an SDRplay RSP1A and an Arrow II Yagi antenna for receiving, and a RF4463F30 with a Raspberry Pi 4 for transmitting. With this combination, Stephen was able to encode video via H264 into packets, which could be streamed to the ground station and decoded. Video fidelity was 640x480 at 12FPS throughout the flight. His measurements of $\frac{E_b}{N_0}$ indicate that the downlink is capable of transmitting a viable feed well beyond 50km.

Stephen wrote his own codebase for encoding and decoding the video stream. His code used gstreamer as the multimedia library.

5.3 UTChattSat – John Barney, Lucas Nichols

The UTChattSat is the predecessor to the IUB-SAT and the baseline for our payload implementation. Many of the components of this design will be reused in some capacity for the IUB-SAT launch. A few of these components include the location APRS/GPS systems, the aluminum chassis, and the Raspberry Pi 4B. The physical design will closely replicate the UTChattSat due to its proven ability to maintain internal temperature. The alternatives discussed in Section 8 represented the additional communication system that we will be adding to this design to achieve live video downlink from the high-altitude balloon.

5.4 Wright State University -Real-Time Video Transmission from High Altitude Balloon

This Wright State University interdisciplinary senior design project attempted to receive live video using an off-the-shelf amateur television transmitter. The payload featured a VM-70X transmitter tuned to the 70 cm band capable of transmitting at 5 Watts. Importantly, their report notes that they were unable to receive video about one minute after launch. Due to our size, weight, and power constraints, we found that this system would not be appropriate for our design.

5.5 ESA PEEK CubeSat

The European Space Agency fabricated a Polyether Ether Ketone otherwise known as PEEK CubeSat chassis with the goal of creating a more environmentally resilient CubeSat. PEEK has a unique set of characteristics when compared to other additive manufacturing materials. Its main selling point is that PEEK has a melting temperature of 350 degrees Celsius which makes more thermally resistive than Lead. PEEK also has a very low conductance compared to other 3D printing materials; and is quite durable under tensile and compressive loads.

6. Objectives

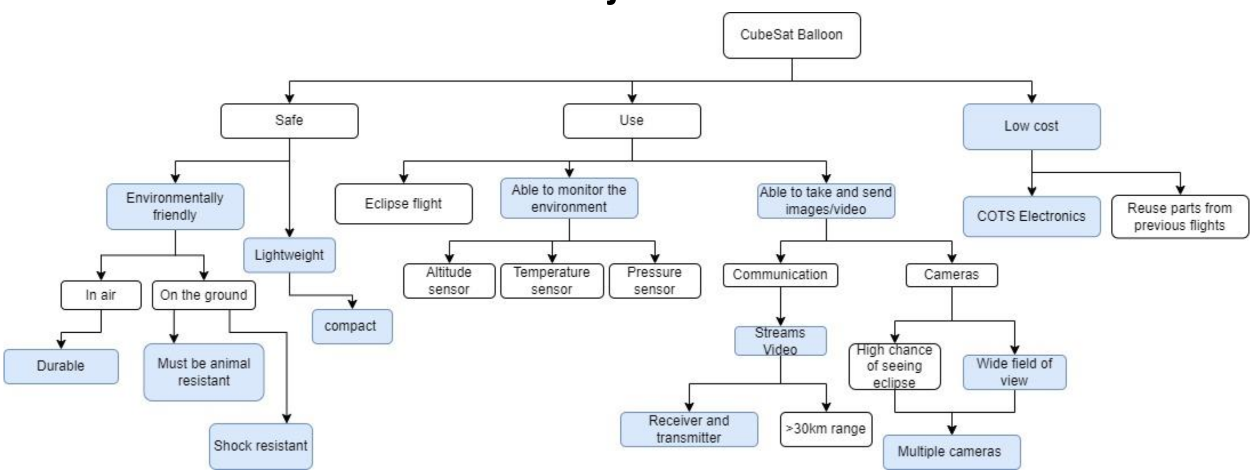


Figure 6.1 Objectives

7. Constraints

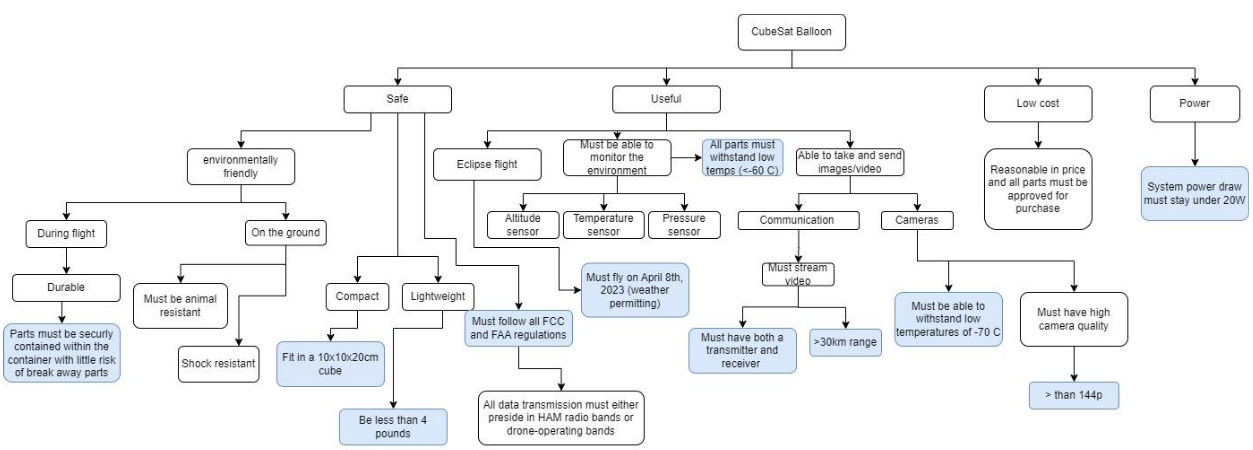


Figure 7.1 Constraints

8. Alternative Designs

8.1 Morphological Chart

Function	1	2	3	4
Imaging	ArduCAM Array	IMX519 Pi Cam	IMX219 Pi Cam	Go Pro
Video Transmission (VTX)	RF4463F30 1W Transceiver	1.3GHz RMRC FPV Transmitter	434MHz Radiometrix NTX2	RM-70X ATV Transmitter
Video Reception (VRX)	SDRplay RSP1A	RTL-SDR	1.3 GHz RMRC FPV Reciever	
Measure Temperature and Pressure	BME680	BME280	MPL3115A2	MS8607
Measure Magnetic Field and acceleration	LSM303AGR	MMC5983MA	HMC5883L	BMM150
Power	Phone Power Bank	LiPo drone cell	AA/AAA batteries	DC-DC voltage steppers on a 12V pack

Figure 8.1.1 Morphological Chart

8.2 Alternative I: Amateur TV/Slow-Scan Digital Video

The first iteration of a downlink imaging system was based on Wenet project developed by the Amateur Radio Experimenters Group (AREG) in Australia. This high-speed imagery payload was designed by Mark Jessop and used to transmit live images from a high-altitude balloon. The payload utilizes Frequency Key Shifting (FSK) modulation and LDPC forward error correction to achieve a 115 kbit/s data rate in the 70cm band. This allows high-resolution images (1920x1440) through Slow-Scan Digital Video (SSDV). Our goal was to replicate this system within our payload to allow for live image transmission during the eclipse.

Using this design would allow us to implement a low-cost and proven downlink system with a developed software repository. The project is documented extensively in a GitHub repository which makes it an ideal candidate for a downlink system. One major issue we faced with this implementation is the legal challenges surrounding this project in the United States. This project was launched in Australia which is assigned a different amateur radio region. These regions determine the laws that apply to the transmission of data in the amateur bands. For the case of this project, using a LoRa transmitter would occupy about 300KHz of bandwidth. LoRa is a technique of spreading the signal across a large part of the spectrum to achieve greater transmission range. This method also has low data rates that would not allow for live video transmission which explains the group opting for high resolution images which were received every few seconds. The code base would require extensive modifications to allow for video capabilities and we decided to continue researching alternative designs for one that better suited our mission.

8.3 Alternative II: First-Person View (FPV) Drone Technology

The enthusiast First-Person-View (FPV) drone market offers compelling video transmission hardware off-the-shelf components. These systems are designed to be relatively simple for users to operate; they are “plug-and-play." There are two primary categories of FPV equipment: analog and digital. These two types may be further subdivided into operating frequencies, like 5.8GHz, 2.4GHz, 1.3GHz, and 900MHz.

Digital systems are newer and are capable of streaming high-definition video. However, that luxury comes at the cost of high prices and limited transmission range. In addition, the newness of digital FPV also means that hardware selection is limited to a select few companies who sell proprietary systems.

Analog FPV systems are a much more mature market segment. Analog transmission has been used for over a decade and, as a result, has an extensive selection of compatible components made by dozens of manufacturers. Analog video transmission also benefits from longer maximum ranges because the video transmitters (VTxs) can tolerate higher wattages. Furthermore, the high level of supply associated with the diverse selection means analog FPV systems are much cheaper than digital. For these reasons, our team investigated how an analog FPV system might be adapted for use on a high-altitude balloon.

To create a working FPV system, there are four main components:

- Video Transmitter (VTx)
- VTx Antenna
- Video Receiver (VRx)
- VRx Antenna

Perhaps the most urgent problem in our use case is range. The VRx antenna is the component which offers the best opportunity to extend transmission range. By using an antenna with more gain, one can trade beamwidth for more range.

The balloon is expected to reach a maximum altitude of 33km. Adding lateral distance traveled, the distance from any ground station to the balloon will be far beyond the range of even the most sophisticated, high-gain FPV system. Drone hardware is simply not designed to operate at such extreme distances. At the time we researched this option, our group was willing to sacrifice transmission at the peak of the flight in return for the relative simplicity and off-the-shelf nature of the system.

Of course, the FPV setup was not exactly plug-and-play. The power requirements for different components represented a major hurdle. The VTx is not a low-power device; over the course of the flight, we calculated a (conservative) power draw of 25.2 Wh. In addition, some components required 12V input while the Raspberry Pi required 5V input.

9. Flight Analysis

9.1. Balloon Performance Calculator

The online balloon performance calculator used to determine the correct positive lift required to burst at the target altitude of 33km and to ensure the ascent time was not too long or short.

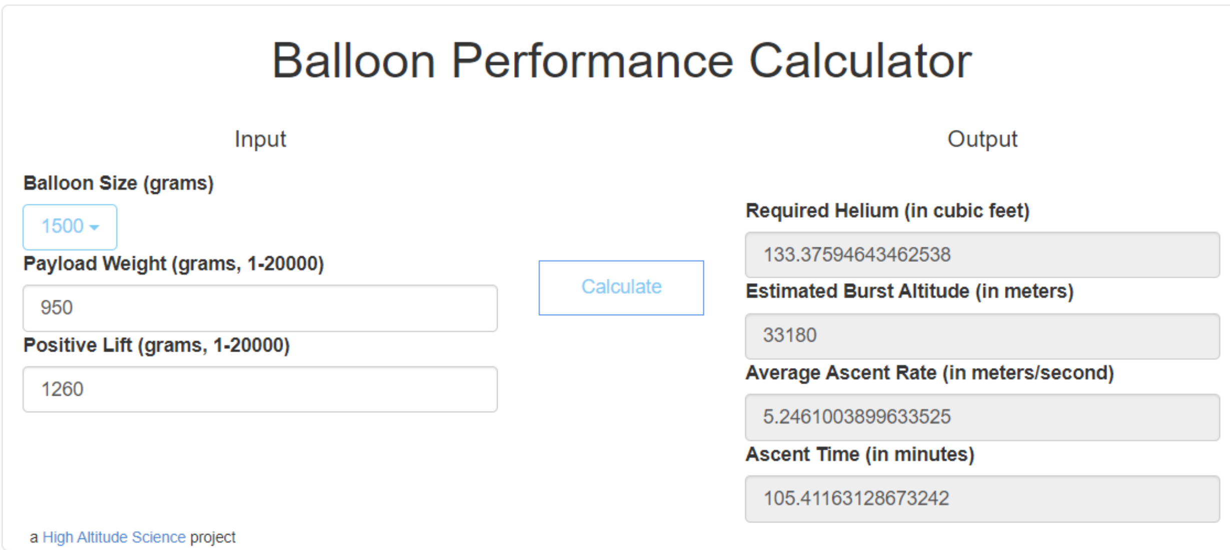


Figure 9.1.1: Balloon Performance Calculator

9.2 Parachute

When flying a high altitude balloon the goal is to have the decent be around 5 m/s any higher is a danger to person and property and any slower risks further lateral travel. The payload weight including cabling for the parachute was estimated to be around 2.2 pounds putting the payload drag calculation for landing speed with a one-meter parachute to be over 6 meters per second which is significantly faster than the team wanted. However, with a 1.5 meter parachute the landing speed was estimated as 4.7 m/s, almost too slow, meaning that the payload would require a size in between these two, however parachutes like this are not sold from reputable sources. It was decided a 1.5 meter would be used despite the slower decent speed prioritizing safter over length of travel.

9.3 Weather

9.3.1 Jet Stream patterns

Launching on the eclipse was challenging mostly due to the changing jet stream patterns during this time of year. The prior night a storm system moved through the state changing the jet stream speed up until hours before the flight. Below is the jet stream prediction from a few days before the flight and then the prediction from the day before the flight.

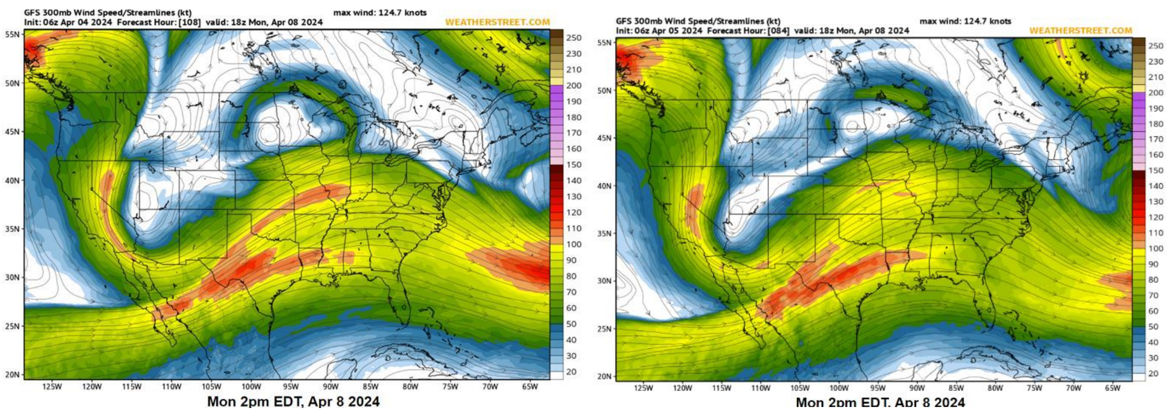


Figure 9.3.1.1 Jet Stream speed predictions

Paying attention to the jet stream was important as it is one of the main factors in the distance the balloon will travel over the ground. Calculations were run to help the team determine which launch time would allow the payload to exit the jet stream before eclipse totality. We determined this should have been approximately one hour after launch. This timing is critical, as images gathered in the jet stream are unlikely to be of any quality due to the significant wind speeds pushing the payload.

9.3.2 Cloud Coverage

As the FAA does not allow flights with more than 50% cloud coverage it was important for the team to watch the weather. On launch day, the skies were fairly clear with cloud coverage hovering at around 35%.

9.4 Flight Plans

Day of flight:

- 10:15a Arrive at the Geology building and load equipment into cars
- 10:40a Arrive at Kroger and get donuts
- 11:00a Arrive at launch location
- 11:00a-11:30a Set up tables and tent
- 11:30a-12:15p Software and transmission testing
- 12:15p-12:50p Rigging
- 12:50p-1:20p Final preparations
- 1:36p Launch
- 1:45p-3:30p watch eclipse
- 3:35p Leave Bloomington to recover payload
- 8:30p Payload recovered
- 1:30a Arrive back in Bloomington with payload

9.4.1 Launch Location

To run the ground station a few things from our launch location were necessary. The team needed to have electricity and enough room to lay out the rigging and perform all necessary tests. It also had to be an open space with a low tree line. Due to the convenience the location was chosen to be an empty lot next to the project sponsors' house. This location supplied us with ample power in addition to additional tools.

9.4.2 Flight Path

Leading up to the launch many different flight path simulations were run. On the day of the launch many calculations were run leading right up to the launch. Below is a simulated run the morning of the launch. It is preferred to have the balloon pop and land in an unpopulated and rural area. Because these requirements were met the launch continued as planned. The team attempted to calculate and launch the balloon at the correct time to be in the path of totality just as the balloon was rising out of the jet stream.

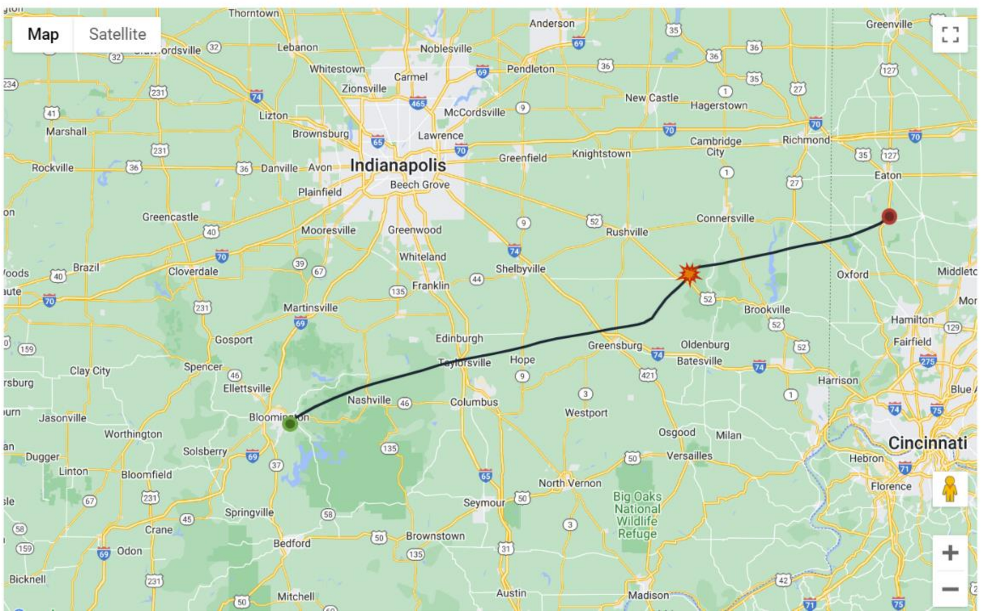


Figure 9.4.2.1: Predicted flight path

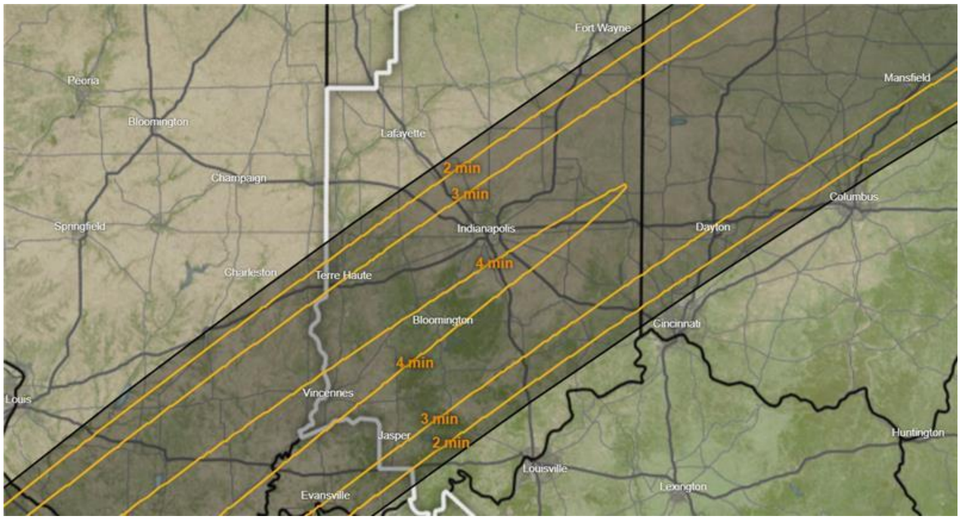


Figure 9.4.2.2: Map of eclipse totality

9.4.3. Notice to Airmen

Figure 9.4.3.1: Map of airspace along the flight path

The notice to airmen (NOTAM) was placed 24 hours before launch Sunday at 1:30p. In accordance with regulations, the radial distance and linear distance from the launch location to the nearest VOR station had to be determined. In relation to the launch location, the closest VOR station was at the Monroe County Airport (KBMG) also known as OOM. The launch time and time to 100,000 ft is required. These times must be presented to the operator in Zulu time which for most of the year +5, but during daylight savings will change. The time to 100,000 ft was calculated using a simple program combined with our average ascent time. This information can be given in a time range. For the Lauch of April 8th, the launch window given was 5:30-6:30 Zulu and the time of 100,000 feet (about 30.48 km) was between 100 and 145 minutes (about 2 and a half hours) after launch. Other required information included the height of the balloon, identifying information, and direction of travel.

9.5 Tether Test

In preparation for the flight the team performed a tether test to understand and practice rigging of the balloon and payload. The test also served as a technology test to test the electronics of the payload. The test was successful with the transmission working after a quick reboot inside Luddy. Prior to the test permission had to be granted by the IU office of Insurance, Loss Control and Claims.

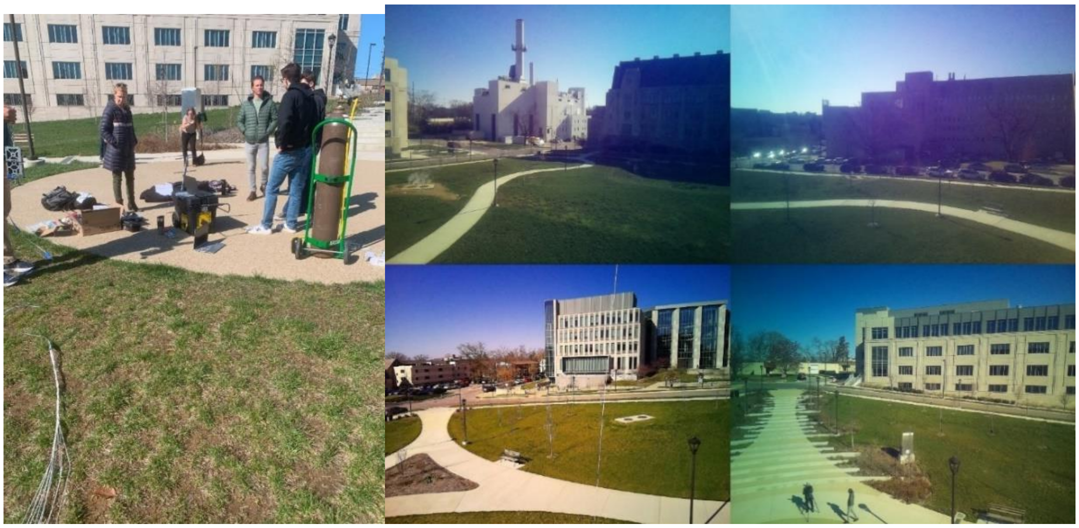


Figure 9.5.1 Tether Test Images

10. Block Diagram

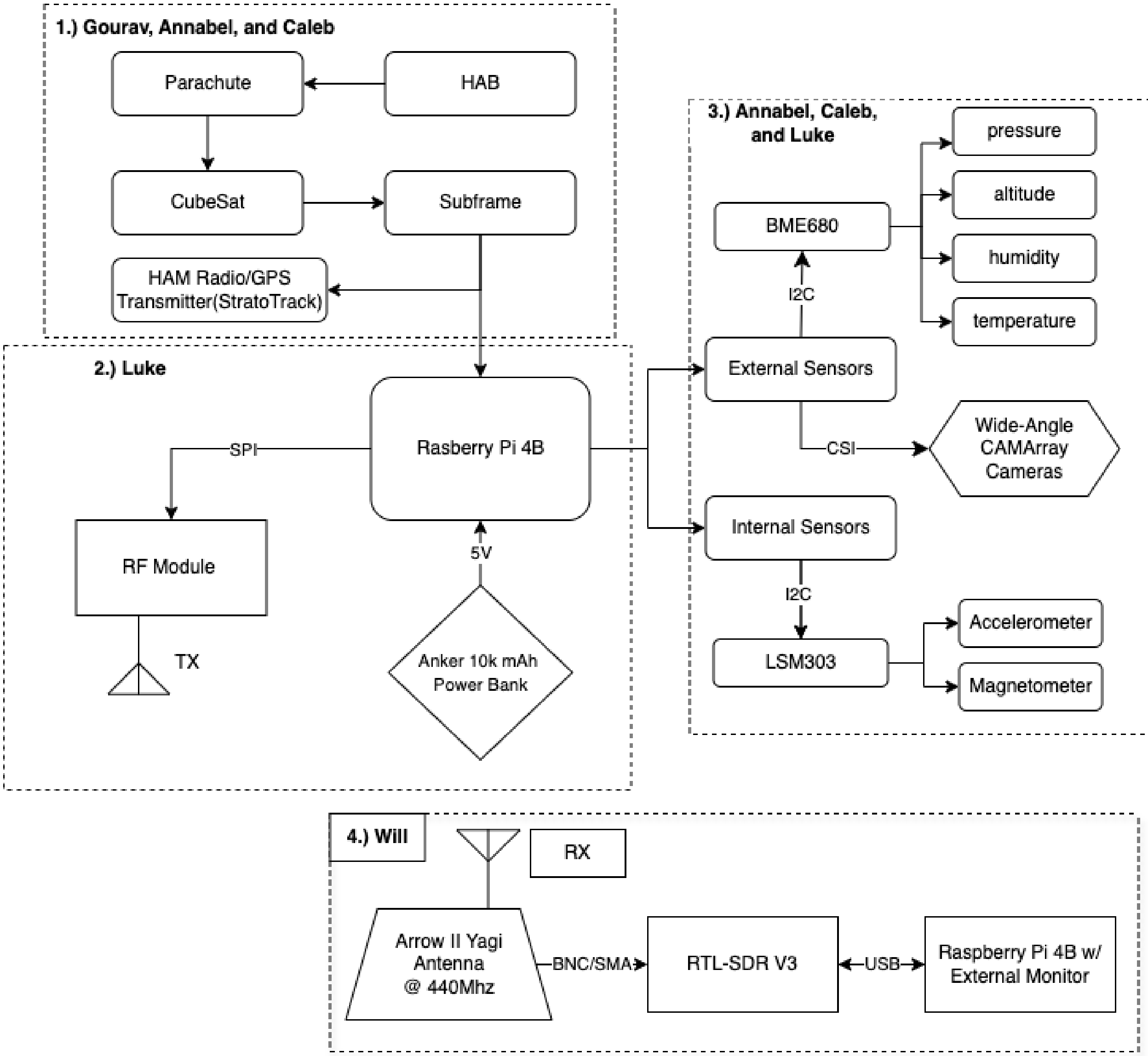


Figure 10.1 Block Diagram

11. Flight Computer

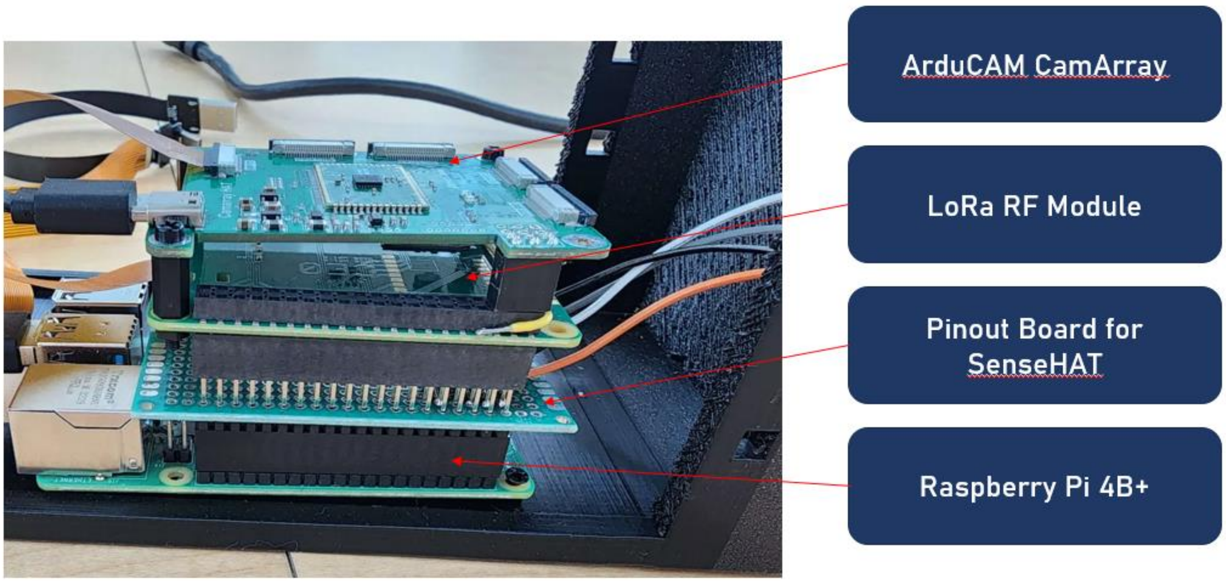


Figure 11.1 Flight Computer Stack

11.1 Sensor Block

Caleb fabricated the payload's SenseHAT to provide stable connection which allowed for proper I2C connections (data and power) from the sensors to the pi. Annabel wrote and adapted the sensor code to work within our software infrastructure to create callable functions in our transmission system. She also did research and determined how to calculate the altitude on the BME sensor.

11.1.1 Hardware

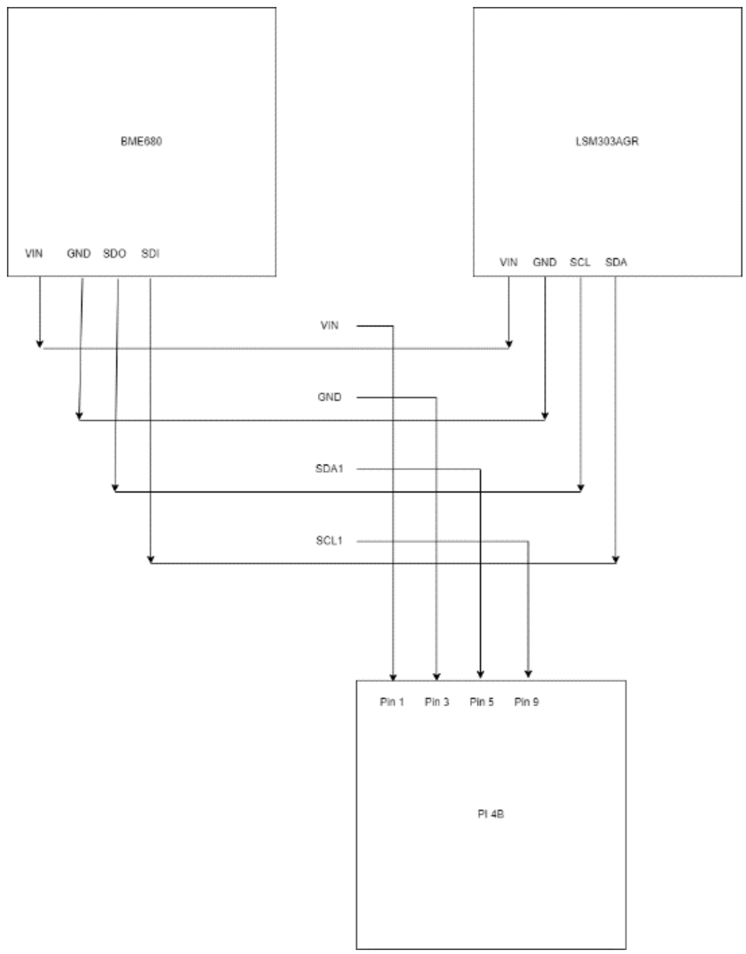


Figure 11.1.1 SenseHAT Pinout

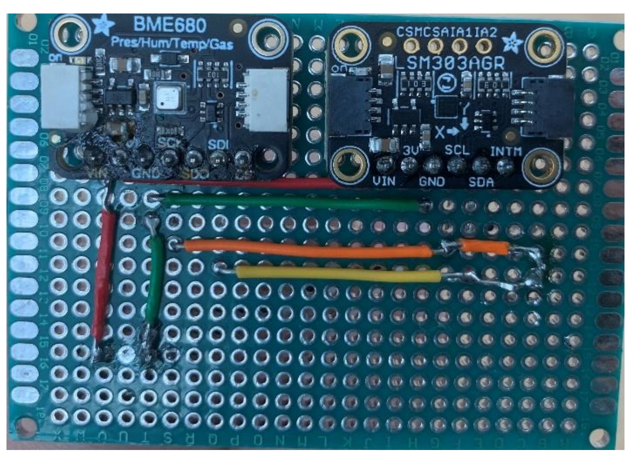


Figure 11.1.2: SenseHat Completed Design

11.1.2 Software

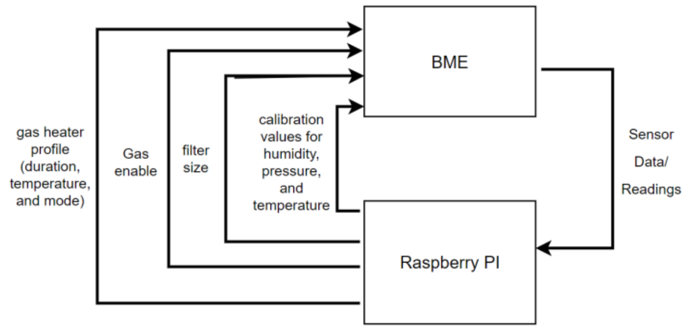


Figure 11.1.2.1: BME Sensor Software Block Diagram

The BME680 sensor tracks temperature, pressure, and humidity using a function call and calibration data. Using this data we then can do some post processing to track the altitude as well. To measure the gas resistance, used in tracking air quality, is a bit more difficult. The BME sensor is a heated type of metal oxide-based sensor. This allows for a longer sensor life and increased accuracy; however, humidity can impact results. For these reasons you must select the gas heater profile which includes the duration, temperature, and mode of the gas sensor.

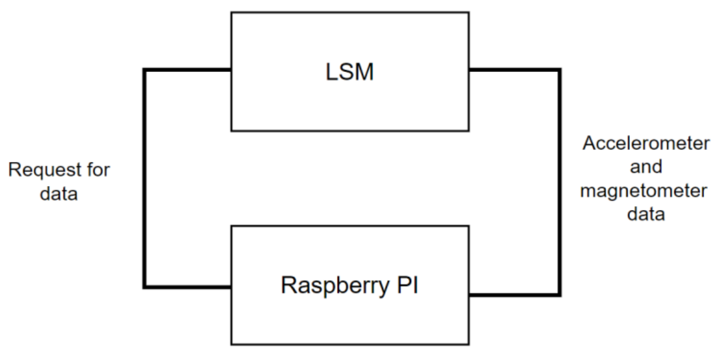


Figure 11.1.2.2: LSM Sensor Software Block Diagram

The LSM data retrieval is much simpler and is just a call to the sensor asking for information. All the data is collected and saved to a file on the pi.

11.1.3 Altitude

There is no built-in altitude calculation within either of the two sensors used. Altitude is an important data point to gather so we had to use the resources within the sensors to determine the altitude. The International Barometric Formula was used. This formula uses a pressure change of Δp = 1hPa and the fact that this then corresponds to 8.42m above sea level. This essentially uses the pressure at sea level and the current read pressure to calculate the altitude with an error rate of plus or minus 1m.

$$\text{altitude} = 44330 * \left(1 - \left(\frac{p}{p_0}\right)^{\frac{1}{5.255}}\right)$$

Figure 11.1.3.1: International Barometric Formula

The International Barometric Formula requires the user to know the pressure at sea level. Bloomington is 804ft (~245m) above sea level which requires additional calculations to determine what the pressure would be at sea level for Bloomington. A derived formula was used for the additional calculation, this is included in the figure below.

$$p_0 = \frac{p}{\left(1 - \frac{\text{altitude}}{44330}\right)^{5.255}}$$

Figure 11.1.3.2: Sea Level Calculation Formula

11.2 Cameras

The payload contains an Arducam CamArray quad-camera HAT. This HAT synchronizes four separate MIPI camera modules so they can be mapped to the single MIPI CSI-2 slot on the Raspberry Pi. Specifically, each camera uses the 8-megapixel IMX219 sensor found in the Raspberry Pi Camera Module 2. This configuration results in a maximum still-image resolution of 3280 x 2464, wherein each camera captures individually at 820 x 616. The resulting image is a composite of the captures from the four individual cameras, producing a four-panel image.

The stock lenses offer each camera a 62.2° x 48.8° field-of-view. Originally, wide-angle lenses were planned for the flight, but testing revealed significant chromatic aberration which overshadowed any benefit from the increased FOV.

In software, a device tree overlay for the IMX219 sensor (dtoverlay=imx219) must be added to the Raspberry Pi’s boot configuration.

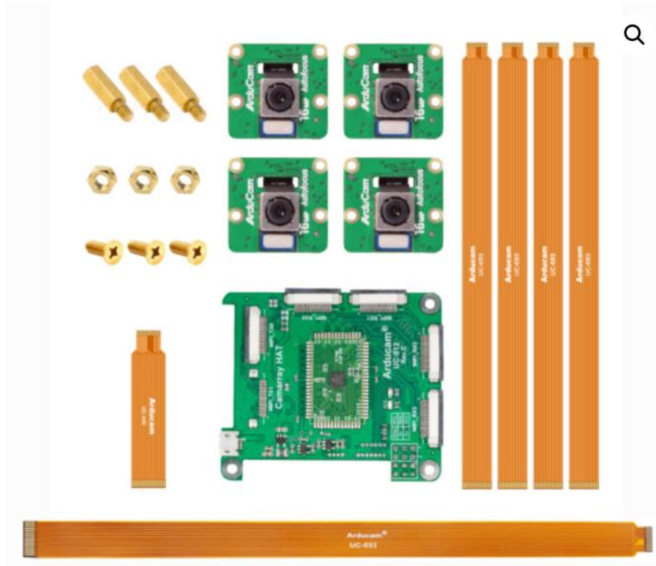


Figure 11.2.1 Arducam CamArray

11.3 Flight Code

The flight code is written in Python and leverages the multiprocessing library to utilize all four of the Raspberry Pi 4's cores. Specifically, unique processes are dedicated to:

1. Capturing photos
2. Encoding them using SSDV
3. Transmitting the encoded packets
4. Polling the sensors and writing to a csv file

Photo capture is handled by calling the command-line tool libcamera-still, which interfaces with the CamArray. The code is designed so that when first run, this process will scan the images directory for existing files and, if necessary, pick up the numbering where it previously left off. This prevents older image data from being overwritten.

The process governing SSDV waits on the capture process to append image IDs to a shared data structure (a manager list) which is then indexed to find the latest addition. This newest ID is then passed to the encode_image function which attempts to overlay telemetry and downscale the image before calling an external program ssdv for packeting. Downscaling is necessary to maintain feasible transmission speeds and not oversaturate the operating bandwidth of our radio band. Packeting involves 2FSK demodulation, followed by low-density parity-check (LDPC) correction. The callsign is embedded in the header of each packet. Encoding is synchronized with transmission using shared events so that resources are not wasted encoding images which can never be sent.

The transmission process interfaces with the RFM98W Transceiver to send the packeted data through the antenna at the requested frequency. Transmitted images are tracked in a set to avoid repeated transfers. As previously mentioned, a sync event is set when transmission completes, signaling the encoding and sensor processes to continue.

The sensor process runs either every 10 seconds, or upon completion of a transmission, whichever is sooner. Several failsafe measures had to be built into the sensor code to gracefully handle sensor errors, most notably disconnection from the flight computer. The sensor routines involved setup and writing to a CSV data file, which contained all the measurements from the BME680 and LSM303AGR. Entries are UTC timestamped.

Overall, the flight code is designed to enable quick transmission by parallelizing the tasks required to operate the cameras, radio, and sensors.

11.4 Lora Expansion Board

The Uputronics Raspberry Pi+ LoRa Expansion Board with an integrated HopeRF RFM98W Transceiver allowed us to transmit SSDV images over the 70cm band using 2FSK modulation. The PCB was modified adding a connection between the Pi's UART GPIO14 TXD (Pin 8) and the GPIO2 pin on the RFM98W. This connection is what allows us to perform 2FSK modulation using the Raspberry Pi. Sending packets to the RFM98W was handled strictly by the Wenet transmission software and due to its complexity, we heavily relied on this software to avoid delays in our project timeline.

12. Ground Station

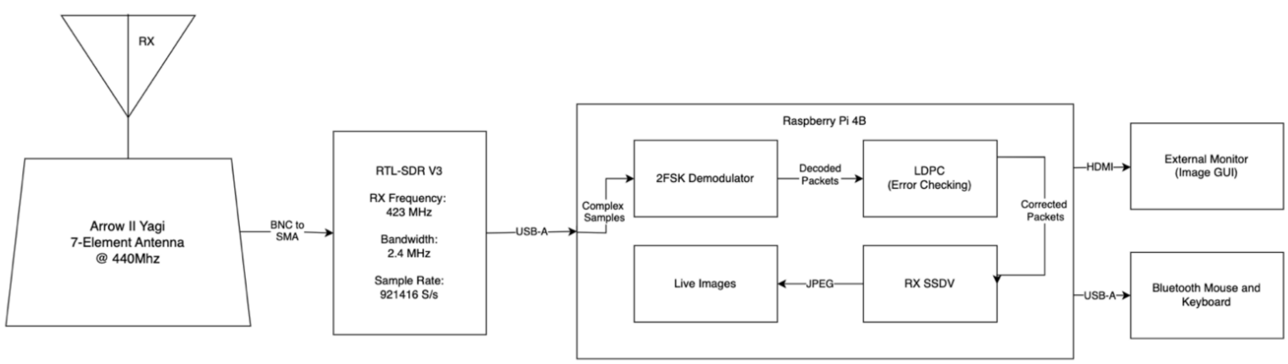


Figure 12.1: Ground Station Block Diagram

12.1 SDRplay RSP1a

The SDRplay RSP1a is a receive-only SDR with a max sample rate of 10Ms/s. This SDR offers an increased sample rate and bandwidth compared to an RTL-SDR V3 dongle and could capture the live Slow Scan Digital Video (SSDV) unlike the RTL-SDR. The RSP1a requires proprietary drivers which made it more challenging to work with and there was much less online support than expected. The real-time video system which was written by Stephen Downward is a modified version of the Wenet project which allowed for low-resolution video compared to highresolutions images in the original Wenet project. Using this modified system required additional SDR performance to achieve real-time video which is why the RSP1a was originally chosen over the RTL-SDR.

After encountering issues with the SPI communication and setting up the ground station to receive video in the modified Wenet system, we decided to abandon the real-time video to meet our flight deadline. We replaced this system with the original Wenet project which could transmit high resolution images at a much slower rate. This choice allowed us to use the much simpler RTL-SDR V3.

12.2 RTL-SDR V3

The RTL-SDR V3 is a cheap and low performance SDR which is very popular among the community due to its cost and open-source drivers. The performance of this device was deemed too low for our original plan to receive real-time video at our ground station, however, when we switched our efforts to original Wenet system, it became a much better choice. Due to its popularity, the RTL-SDR V3 has tons of online forum support which was important for debugging issues. Using this SDR made for a plug-in-play ground station once the Wenet Docker image was set up on a Linux computer.

12.3 Wenet (Software)

The Project Horus Wenet software [8] is an open-source project designed specifically to transmit/receive live high-resolution images from a high-altitude balloon. After pivoting away from the real-time video system, which is an enhancement the Wenet software, we fell back on the original Wenet software for our flight. This software was designed by Mark Jessop and Steven Honson, members of the Amateur Radio Experimenters Group (AREG) located in Australia. The receiver software allowed us to demodulate SSDV images transmitted by the Raspberry Pi Lora Expansion Board. The software is set up in a convenient docker image which makes it simple to set up and deploy without dependency issues.

12.4 Ground Computer

The Wenet ground station software was deployed on a Raspberry Pi 4B as it provided a cheap and easy-to-use Linux OS for the docker image. The GitHub repository [8] contains instructions for setting up the receiver and includes information on all the required hardware. The images are displayed via a local host web interface which will function without a network connection. Image 12.4.1 showcases the ground station computer running before our flight.

12.5 Yagi Antenna

Following recommendations by the Amateur Radio Experimenters Group (AREG) and other online forums, we chose the Arrow II Yagi 7-Element Portable Antenna centered around 434 MHz. This antenna was connected to our RTL-SDR using a BNC-to-SMA cable and mounted to a tripod. Yagi antennas are directional, so moved the antenna as needed to pointed toward our payload during ascent.

Figure 12.5.1: Ground Station Computer

13. Payload

13.1 Tracking

There were two tracking methods included in the payload including the StratoTrack APRS Transmitter and the SPOT Trace Satellite Tracker.

13.1.1 StratoTrack

The StratoTrack APRS Transmitter [11] is a GPS transmitter designed for use on High Altitude Balloons. It transmits telemetry including GPS coordinates, altitude, temperature, and barometric pressure over the 2-meter HAM radio band (144.39 MHz) for North America. APRS is the Automatic Packet Radio System that allows users to view their telemetry data via a web browser (aprs.fi). APRS works by using the digipeater network to relay your packets to the nearest IGate which then uploads the data to the browser. The StratoTrack contains an unlocked GPS module, meaning it can transmit packets even at extreme altitudes (>30 km). Considering subscription-based devices such as the SPOT Trace are not functional at and above jet stream altitude, this is the best way to track your balloon for most of the flight.

The StratoTrack is not perfect, however. It requires line-of-sight to the nearby digipeaters for the data to be received. This means that the StratoTrack only works after it has reached an altitude of about 1000ft. The APRS network received our first packet at 784ft and maintained connection until its last transmission at 1640ft during the descent. The StratoTrack allowed us to visualize the changes in our flight path which turned out to be vital, due to our balloon travelling about 100 miles further than predicted. After the StratoTrack failed in the previous project (UTChattSat), our goal was to prioritize testing the device for our launch. However, given the altitude requirement, it is challenging to test.

One crucial modification we made to the payload design was substituting the metal cable connecting the parachute and payload with a durable nylon rope instead. The StratoTrack is intended to be secured vertically on this cable and we believe that using a metal cable could have negatively affected the antennas, resulting in a degradation of the APRS signal. Considering the StratoTrack's great performance in our flight, we believe that this change did fix the issue. Per the StratoTrack user manual, the device was secured along the rope with the GPS antenna directed upward, the top APRS antenna secured to the rope, and the bottom APRS antenna free hanging.

We purchased the StratoTrack's proprietary reprogramming cable to put Caleb's new callsign on the device (KD9ZSC-11). The "-11" SSID is used to indicate which type of transition is being used (-11: balloons, aircraft, spacecraft, etc.) These SSID's make it possible for aprs.fi to include icons (e.g. a balloon) for each APRS transmitter. Figure 13.1.1 shows the aprs.fi map after using the tracking feature to isolate the KD9ZSC-11 callsign. Each dot represents a new APRS transmission being received.

Figure 13.1.1 StratoTrack APRS Transmitter

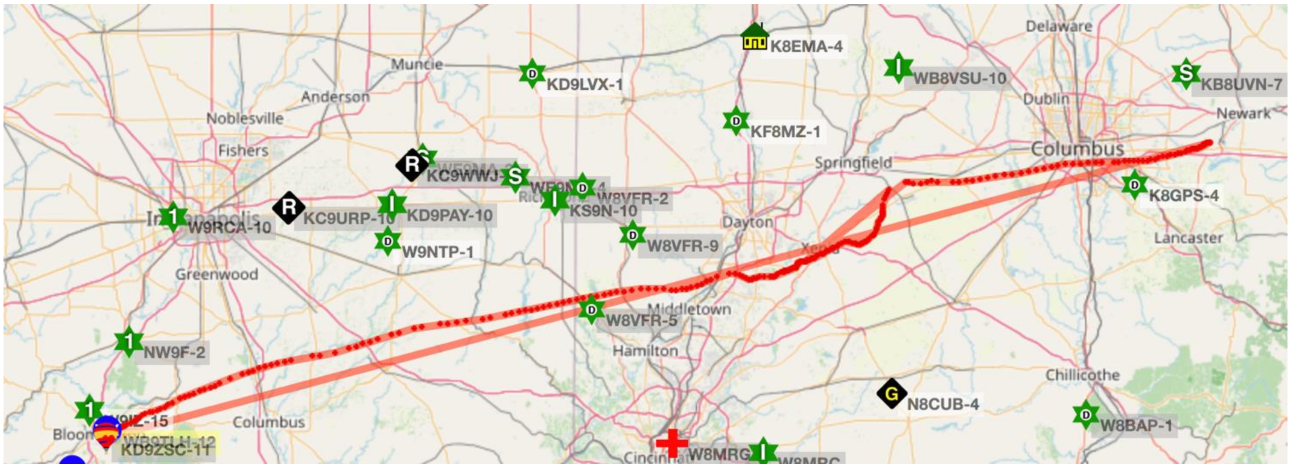


Figure 13.1.2: Payload flight path tracked on aprs.fi from the StratoTrack.

13.1.2 SPOT Trace

The SPOT Trace [12] is a subscription-based satellite tracking device used to locate the device during low altitude periods of the flight. Unlike the StratoTrack, this GPS device is used to track the payload after it has landed and APRS data is no longer available. This is the most important tracking system in terms of payload recovery as without it, finding your payload becomes significantly more difficult. Using the last APRS ping at ~1600 ft in our case, the search area for the payload can become quite large depending on the ground wind speeds. The SPOT Trace provides GPS location with about 20 ft of precision. After updating the subscription and updating the device firmware, we tested the device by giving it to a team member to walk around with on campus. The device performed as expected and successfully pinged while outside. The device is intended to update its location every 5 minutes while movement is detected. This can be problematic because the balloon can ascend to non-operational altitude within the first 5 minutes of flight. Our device transmitted at the launch site minutes before our launch, however, the device failed to accurately transmit any GPS data throughout the flight, even after the payload landed. Like the UTChattSat results, the SPOT Trace failed to provide any useful information and it is recommended to find an alternative satellite GPS module for future flights. Fortunately, our payload landed in the front yard of a remote residential town and the landowner called us and provided his address. Without this call, it is fair to assume the payload would not have been recovered. We arrived about 2 hours after the payload landed and the SPOT Trace had still not successfully connected to the satellite network. Three hours into our drive home from the landing location, the SPOT Trace regained connection and pinged near Indianapolis. Given the unpredictability and unreliability of the device, we strongly recommend replacing the device. The current device is estimated to be about 10 years old, so buying a newer model from Saved by SPOT could prove to be an effective solution.

13.1.3 Audio Beacon

The audio beacon is a 9V powered speaker which provides a loud siren to help locate the payload during recovery. This device would be useful in scenarios where the payload lands in trees or brush. The audio beacon was tested and planned to be deployed for our flight. However, given our time restrictions, adding the audio beacon to the exterior of the payload was overlooked in the final minutes before launch. The SPOT Trace and Audio Beacon are intended to be secure to the exterior of the payload which can make the assembly process of the final payload hectic without proper planning. The alternative is to sacrifice battery life by turning on the Pi earlier because it cannot be powered on after being placed in the insulation. To clarify, we were fortunate that our payload was found by the landowner and in most scenarios the audio beacon would be needed to quickly recover the payload.

13.2 Insulation

We decided to insulate our payload using standard Styrofoam to prevent cold jet stream winds from affecting our electronics. Due to a battery or software failure just 15 minutes into the flight, it is hard to quantify the benefits of the insulation. Given the temperature data of the UTChattSat flight, it is reasonable to assume that the insulation does provide some protection against the jet stream winds for electronics in the payload interior. Our payload was in the jet stream for almost double our flight prediction and the StratoTrack shows that the exterior was around -40 C for about almost a combined two hours during ascent and decent. With that, the insulation would have provided some protection given we did not encounter problems earlier in the flight, but it is hard to determine at what time our battery failed.

13.3 Rigging

The rigging for our payload consisted of the following components:

- Balloon: A 1500g Weather Balloon from High Altitude Science.
- Parachute: 5 ft RocketMan parachute with the string connected to the rope below using a locking carabiner.
- StratoTrack: APRS Transmitter connected to the 15ft rope in between the parachute and payload using zip ties. Recommended to be placed at least 12ft above the payload to prevent interference.
- Rope: 15ft rope rated at 50lb of force to connect payload and parachute.

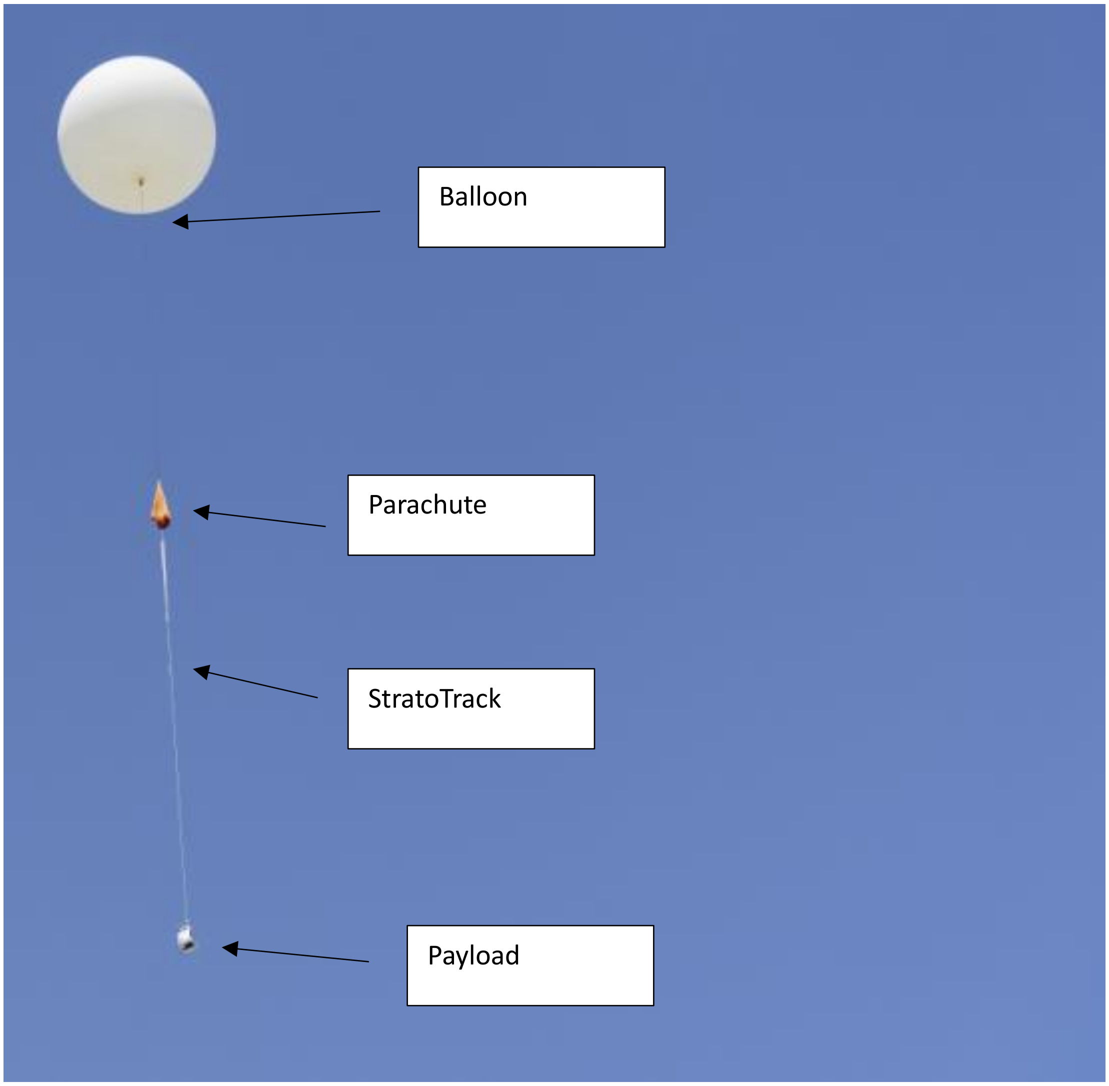


Figure 13.3.1: Rigging Layout

14. Safety Concerns

Safety Concerns are somewhat minimized in the scope of launching a high-altitude balloon. There is little risk for personal injury or catastrophic failure of any kind. The main concern is abiding by FAA regulations to not disrupt or interfere with any other happenings. These regulations will primarily include constraining our system to less than 4lbs as there is no cut-away system within the CubeSat which bypasses many FAA regulations that a larger payload would incur. The team also must alert the FAA as to when, where, and the approximate direction the CubeSat will travel on the day of the launch so that there is awareness.

15. Environmental Considerations

There are many environmental concerns when launching a high-altitude balloon. The first concern is small parts that can fall away during flight and can land on the ground to be consumed by wildlife in the area. Another concern is wildlife finding the CubeSat, balloon, and parachute after landing and being able to consume parts of the project before it can be recovered. There is also a risk that we cannot recover the project. Not recovering the project is also a major environmental concern. The project consists of electrical components, including a battery. If these were to not be recovered, they would start to corrode, especially if it were to land in a body of water. The team also must ensure that the enclosure is safe for the environment. The location of the launch is also important to note. Being launched in Southern Indiana there are many farms. The team should take caution when dealing with livestock and produce. These concerns are similar to the concerns listed above. The battery itself is an additional environmental concern. Depending on the weather on launch day, overheating is a major concern. This can cause a small fire, which depending on landing location, can cause a larger fire.

16. Ethical Considerations

It is important to follow good radio operator etiquette when using HAM radio. Although it is a subjective topic, radio etiquette ensures that the rights of other HAM users are not infringed upon. In addition, in the event that the payload lands on private property, it is important that the team respects the rights of landowners. In such a case, the team will ask for explicit permission to retrieve the payload from the private property.

17. Regulations

17.1 FCC

The FCC regulations are mainly discussed in Title 47, telecommunications. In part 97 of title 47 of the FCC regulations, amateur radio services are discussed in depth. The teams' system is running in the 70 cm band in the amateur television (ATV) frequency. The ATV frequencies in the 70 cm band include 420-432 MHz and 438 to 444 MHz. These bands are both classified as ultra-high frequency, UHF. Transmissions are also given a three-symbol label describing the emission. Our transmission is classified as an image due to its F1F coding. This means that the transmission if frequency modulated (F) is a "single channel containing quantized or digital information without the use of a modulation sub-carrier, excluding time-division multiplex" (1)[10], is television- video (F). These classification codes are important as they define what transmission you have and what band you are allowed to transmit in.

In order to transmit in the 70 cm band, one must hold an amateur license as well as a station license. This allows for the legal transmitting within 50 km of the Earth's surface aboard any vessel or craft that is documented or registered in the US. The person with these licenses must be in physical control of all transmitting devices. When the station is automatically controlled, there must be other regulations involved depending on the band and power of the transmitter.

The FCC procedures and practices are updated every October first so the team checked in often to make sure no additional regulations were adopted.

17.2 FAA

An unmanned free balloon is subject to many regulations by the FAA. However, to be an unmanned free balloon the payload must be greater than four pounds [10]. Since our design is projected to be less than four pounds, we are not subject to the FAA regulations regarding unmanned free balloons, however, most guidelines are followed to ensure maximum safety.

FAA title 14 provides weather stipulations. For flights there must be less than 50% cloud coverage and no precipitation as well as visibility of 5 miles or more. [10]

Additional guidelines include but are not limited to:

1. No use of cellular devices is permitted to track at high altitudes
2. Payloads cannot exceed a package weight/size ration of 3 oz per square inch measured on any side of the package
3. No payload can exceed 6 pounds in weight
4. No rope or cable should be used, which requires more than 50 pounds of force to separate
5. Cannot create a hazard to other people or property
6. Cannot drop objects
7. Prelaunch and during launch notices required to be submitted
8. Cannot launch from an airport without prior permission
9. Cannot fly over a town or open-air group of people for the first 1,000 feet of vertical flight
10. Balloon and payload cannot be a hazard to people if it should hit them
11. If flying before sunrise or after sunset a strobe light that can be seen 5 miles away must be attached
12. The balloon envelope is equipped with a radar reflective device that will present an echo to surface radar operating in the 200 MHz to 2700 MHz frequency range

18. Standards

Standard: 14 CFR 1.F.101.D
The FAA Unmanned Free Balloon standard states regulations regarding the use and deployment of unmanned free balloons/ weather balloons. This includes size constraints, notification requirements, and attachment methods. These standards must be followed to assure the safety of aircraft and bystanders in the area. We will discuss this standard in-depth later in the paper.

Standard: Title 47 CFR 15 Subpart B
The FCC Unintended Radiators standard has a portion under part 15.103 exempted devices, we fall under the part a exemption which states “A digital device utilized exclusively in any transportation vehicle and aircraft”. We also are also transmitting using our own personal HAM radio license within the designated 420-450MHz ranges allocated by the US for amateur radio usage.

Standard: ISO 17770:2017
The ISO Space systems- Cube satellites standard defines the subclass of satellites called a “picosatellite” as the CubeSat. By their definition, “CubeSats provide a low-cost platform for testing and space qualification of the next generation of small payloads in space”. We knew from the beginning that we would be abiding by this standard, so our entire design language has been centered around abiding by the definition of a 2U CubeSat. Including keeping our payload under 4lbs, having an integrated cutoff system due to the balloon intentionally bursting at its peak altitude, and keeping our dimensions within the 2U specifications.

Standard: IEEE 1625-2004
The Standard for Rechargeable Batteries for Portable Computing provides guidelines for the quality and reliability of rechargeable batteries for portable computing. A portable battery cell will be used in our payload to power our flight computer and the attached sensors and components. The standard includes critical testing practices which were considered before selecting a battery cell for our flight.

Standard: IEEE 211-1997
The IEEE Standard Definitions of Terms for Radio Wave Propagation contains useful terminology used in radio system and equations used to calculate wave propagation. This standard was utilized to calculate free-path loss estimates for our radio system. The standard also provides a general overview of basic radio terminology that we reviewed.

19. Intellectual Property

19.1 US Patent 1- Satellite Imaging System

US Patent US9052571B1 outlines a system for capturing wide-area aerial images. It specifies parameters relating to camera composition, frames for holding the cameras, focal length, camera angle, and FOV.

We are not infringing on intellectual property as this system uses a fundamentally different design, components, and is intended primarily for downward imaging from an airplane.

19.2 US Patent 2- CubeSat

US Patent US9248924B2 consists of specifications relating to a CubeSat system, its methods, and apparatus relating specifically to pico-class satellites.

We are not infringing on any intellectual property regarding this patent as the outlined system is specific to their outlined system configuration which we are not using or imitating in any manner.

The rest of this patent outlines system methods relating to their specific CubeSat implementation.

19.3 US Patent 3- RF CubeSat System

US Patent US11005165B2 refers to a system and method to use a transceiver antenna mounted on the outer panel of a CubeSat. This essentially outlines a method of wrapping a CubeSat in an RF antenna to provide a higher degree of uplink sensitivity.

We are not infringing upon any intellectual property since this patent outlines a mounting system for a specific RF antenna setup relating to a CubeSat. We will instead either use a trailing antenna or an internal antenna for transmission.

20. Flight Results

20.1 Data

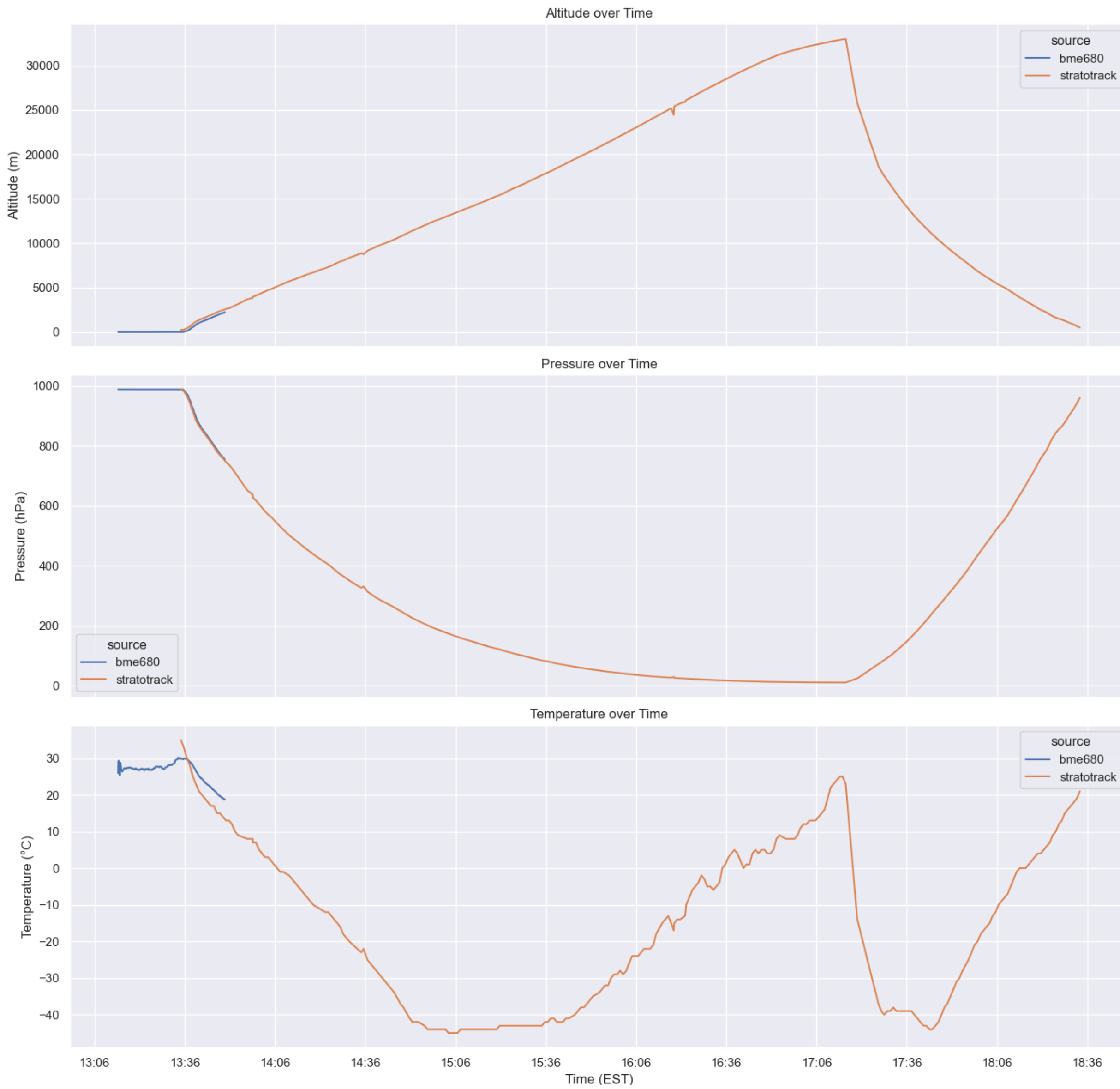


Figure 20.1.1 Flight data from the BME680 and Stratotrack sensors over time, beginning at launch

Figure 20.1.1 shows the sensor data over time. Altitude values were calculated post-launch using the pressure readings. As expected, pressure and temperature decreased as altitude increased.

The temperature data is especially helpful, as it shows when the payload flew through the jet stream. The high winds significantly dropped temperatures, down below -40°C. Additionally, because these plots share the x-axis, we can use the corresponding altitudes to find that the Jetstream is between 11-17 km.

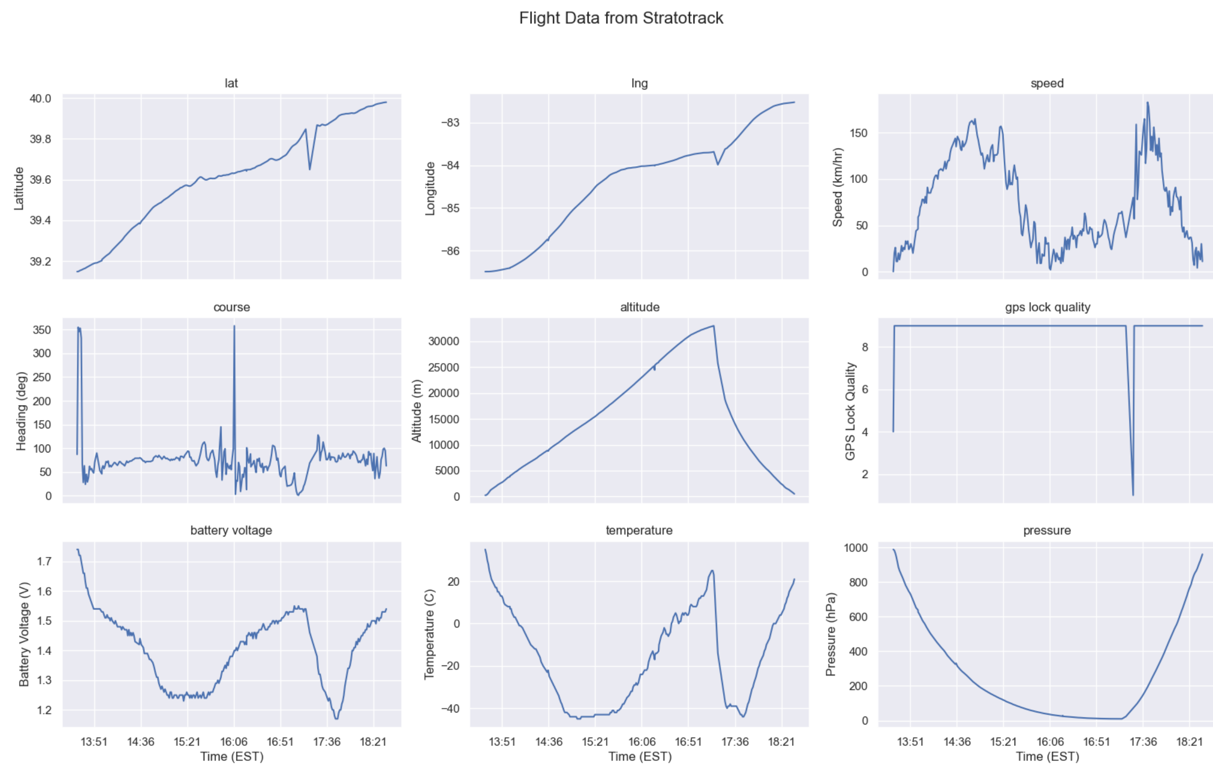


Figure 20.1.2 All StratoTrack Data

Additional data captured from the StratoTrack shows that the payload reached a maximum speed of approximately 160kph (100mph). The latitude and longitude are also becoming more positive, indicating a northeastern direction, which is correct.

Colder temperatures in the Jetstream caused the StratoTrack's battery to decrease in voltage, but then rebound when temperatures increased. We also maintained a strong GPS lock throughout the flight, except at the moment of bursting, when the lock broke momentarily. This was cited in the StratoTrack documentation and was expected by the team.

20.2 System Failures

For all the success we had throughout this project we anticipated missing a few key issues. The nature of a project of this type can be nearly unending depending on team size, project ambition, and many other key factors. This section will delineate what we considered our major failures of this project were, all of which occurred on launch day. One could attribute this to poor planning but, frankly, we considered so many other factors that we simply missed a few.

The first key failure was related to the amount we filled the balloon. Throughout the past year we consistently struggled to speculate on what our final payload weight would be, and, to some extent, we dismissed it as a problem that should have been fixable on launch day. It was our understanding that by weighing our payload while we were filling the balloon, we would be able to adjust positive lift as necessary. However, due to our desire to give our battery the best chance of lasting the full duration for the flight, we delayed full assembly until the last 45 minutes before launch. Our system required that we power on the Pi early in the assembly process due to the semi-sealed nature of the payload (i.e. it was not possible to power on the device after enclosing it). Caleb tried to mitigate this hurdle by designing the Subframe such that it could slide into the metal chassis for easy assembly. Despite this design consideration, our assembly was still completed too late and by the time we knew the system's total weight (~1120g compared to our expected 950g from previous weights) the balloon had been filled and tied off. The major miscalculation in weight came from us forgetting to weigh the insulation exactly; we had factored in an estimation but did not account for enough excess weight due to the amount of tape we used to secure the payload. Greater precision regarding system weight and a greater allotment of lift would have fixed this problem. Ultimately, though, the payload did fly--but instead of it being a 3-hour flight, it was over 5-hours. This caused the payload to fly ~250 miles which was 4.5 hours of driving both ways. Our recommendation for future groups attempting this project: Overfill your balloon.

The second key failure we speculate to be related to the battery bank we chose. Our predecessors used a VGE power bank near 5500MaH of power capacity. We chose to use an Anker smart battery of 10000MaH which in our later testing gave 5.5 hours of system runtime which is what we expected. However, we should not have used a smart battery. This specific Anker battery has several “smart” shutoff mechanisms and modes. None of which are useful in this specific application. Gourav and Caleb discussed using a Buck converter and Energizer Ultimate Lithium batteries to give ~5.2V @4A of power which would have been the better choice. However, Professor Loveless advised against using this in favor of using a fabricated power board used in the past which is frankly too large and overbuilt to be useful in a payload of this size. Which led us to using the Anker battery which was a mistake. Our recommendation for the future would be to fabricate a discrete power system like what Gourav and Caleb discussed which would be much more useful in this specific application.

The third key failure relates to the SpotTrace. Simply put, this device did not work effectively. We vehemently recommend against using the SpotTrace we used in this project. It never once succeeded in pinging its location throughout the flight and after it landed. In fact, this device only pinged in Indiana once it was turned on at the launch site and pinged again after the team had returned with the payload to Indianapolis. The Stratotrack was our saving grace in the tracking department and APRS as a whole is key to tracking devices of this nature. Our recommendation is to both reuse the Stratotrack and either purchase an upgraded unit from SpotTrace in the future or build a discrete GPS system, but we simply didn’t have the time or manpower to consider those options heavily enough.

20.3 Software Issues

Log data collected post-flight revealed significantly less information about the failure of the payload than anticipated. Kernel-level logging was not enabled, nor even considered until afterwards. The only log available for review included output from the flight script itself, which could never be informative in the case of a hardware fault such as a power failure. Indeed, the logging was written with software failures in mind to facilitate debugging, resulting in a log file which shows the script operating as expected before a sudden cutoff.

Of course, even if journalctl had been configured for persistent logging, it is unknown if any specific information about the power state would have been included. Nevertheless, we feel that any future teams should be aware of syslog and journalctl, and use them.

In addition to shortcomings in the logging, our Raspberry Pi was not configured with a watchdog process outside of the flight code scripts. Having an external watchdog would almost certainly have reduced the possibility of a software failure. Although we do not have any credible indication of the script failing, implementing a watchdog to restart the flight script would have been a useful failsafe. Of course, such a watchdog would only be useful if the Pi had power, so it is not directly related to the issue we experienced, but future teams should implement this regardless.

20.4 Reflection

Our post-flight investigation led us to find several issues that we had missed in our various system tests throughout the project. In general, if we had created a well-documented testing process for our system, we may have discovered the issues that caused the total system fault we experienced.

However, hindsight is 20/20 and, to some extent, the team expected that oversights might occur due to varying degrees of burnout or lack of contributions. Furthermore, we are confident that had we been made aware of and settled on the Wenet/SSDV approach sooner, we could have expanded the testing significantly and refined the system to make it much more robust.

Given the amount of effort we dedicated to this project, and coupled with the support of various professors, we feel quite proud of our achievements even if our flight did fail in the aforementioned ways. If we learned anything, it is that no risk--even if known--can be fully engineered away, nor can every risk be exhaustively accounted for.

21. Acknowledgements

The team would like to thank Professor Daniel Loveless for his continued sponsorship throughout the project. The team is also grateful to Professor Bryce Himebaugh for his guidance and advice on all aspects of radio communication. The team also thanks Professors Mary

Loveless and Suha Lasassmeh for their direction during the project and support with purchasing components.

22. Appendix

22.1 Link to Images

22.2 Link to Codebase

22.3 Presentation Slides

22.4 Links to Tools Used

22.4.1. APRS Tracking - Used for getting real time location on the balloon during flight.

22.4.2. Flight Path Predictor - Used for predicting the path of the balloon on a given day.

22.4.3. Balloon Performance Calculator - Used for calculating the amount of helium and upward lift.

22.4.4. Aeronautical Charts - Used for calculating the radial and linear distance for the NOTAM.

22.4.5. Sondehub Flight Predictor - Used as a secondary flight path predictor.

22.4.6. Jet Stream Forecast - Jet stream prediction tool.

22.4.7. NASA Solar Eclipse - Prediction of eclipse path of totality.

22.4.8. IU Liability Office - Used to inform the office of Liability about the tether test.

23. References

[1] “Princeton Cubesat Kit (all PC104!) | tigersats,” Princeton University, https://tigersats.princeton.edu/cubesatkit (accessed Nov. 6, 2023).

[2] “Everything RF,” Link Budget Calculator - everything RF, https://www.everythingrf.com/rf-calculators/link-budget-calculator (accessed Nov. 6, 2023).

[3] “Free-space path loss,” Wikipedia, https://en.wikipedia.org/wiki/Free-space_path_loss (accessed Nov. 6, 2023).

[4] “3D printing CubeSat bodies for cheaper, faster missions,” ESA, https://www.esa.int/Enabling_Support/Space_Engineering_Technology/3D_printing_CubeSat_bodies_for_cheaper_faster_missions (accessed Nov. 6, 2023).

[5] “2022 SOA summary - NASA,” State-of-the-Art Small Spacecraft Technology , https://www.nasa.gov/wp-content/uploads/2023/05/soa-summary-2022.pdf (accessed Nov. 7, 2023).

[6] “500 kbps from a high altitude balloon,” Stephen’s Site, https://www.scd31.com/posts/high-speed-balloon-comms (accessed Nov. 6, 2023).

[7] “Real time video from a high altitude balloon,” Stephen’s Site, https://www.scd31.com/posts/real-time-balloon-video (accessed Nov. 6, 2023).

[8] Projecthorus, “Projecthorus/wenet: Wenet (‘The swift one’) - transmit and receive code for the project Horus High-speed imagery payload.,” GitHub, https://github.com/projecthorus/wenet (accessed Nov. 6, 2023).

[9] “Indiana Repeater Council,” Indiana Repeater Council Serving Indiana Amateurs Since 1971, https://www.ircinc.org/?page_id=390 (accessed Nov. 6, 2023).

[10] “The Federal Register,” Federal Register :: Request Access, https://www.ecfr.gov/current/title-14/chapter-I/subchapter-F/part-101/subpart-D/section-101.35#p-101.35(a)(1) (accessed Nov. 6, 2023).

[11] “Stratotrack APRS transmitter,” High Altitude Science, https://www.highaltitudescience.com/products/stratotrack-aprs-transmitter (accessed Spring 2024).

[12]“SPOT Trace Satellite Tracker with GPS,” Saved by SPOT, https://www.findmespot.com/en-us/products-services/spot-trace (accessed Spring 2024).