Can'tSat: Pre-Launch Report

Ryan Xu, Ayman Gani, Brooke Mahajan, Matthew Liu, Afnan Rajab

April 2025

Prepared for the Can'tSat Pre-Launch Submission

Contents

1 Introduction

1.1 Team organization and roles

The Can'tSat team is made up of 5 team members and 1 mentor, coming together through a shared passion and interest in engineering.

- **Ryan Xu (16) - Team leader**: Ryan, having a great interest in engineering since a young age, gathered the team together as a way to further explore his passion in the subject. In the project, he is responsible for delegating roles, 3D modelling and printing the CanSat, and building parts of the CanSat itself with the help of Ishan.

- **Ayman Gani (16) - Chief Software Engineer**: Ayman was drawn to this challenge from his interest in software, hardware, and technology. He loves working on similar types of projects in his free time. He worked on a variety of the parts of our CanSat, primarily including the software and programming involved in our microcontroller and our ground station.

- **Afnan Rajab (16) - Sponsorship Head**: Afnan Rajab has always been quite curious about both aviation and engineering, even from a young age. This has cultivated into a great passion for technology. Whether he's exploring theoretical concepts or working on hands-on projects, Afnan thrives on solving complex problems and pushing the boundaries of innovation. In his free time, he enjoys playing lots of music, volunteering in various music related activities and participating in his school band.

- **Ishan Mahajan (16) - Hardware Technician**: Ishan has a deep curiosity for physics and technology and has experience in miniature electronics. He was responsible for the planning and construction of the CanSat and the ground receiver along with helping research what parts should be integrated into the project.

- **Matthew Liu (16) - Outreach Manager**: Matthew is a talented individual who thrives on curiosity and intellectual engagement, leading him to participate in this challenge. His ability to problem solve and critically think has proven very useful and often crucial in the team. He likes to draw.

- **Wensheng Xu (55) - Team Advisor**: Wensheng, or Jack, was born and raised in Northwestern China. He earned his degree in engineering and worked as a maintenance engineer for 12 years before emigrating China and moving to Canada, where he now works as an exemplary electronic technician.

Although a large portion of our group attends the same school, we have opted not to work on this project at school due to project safety and time constraints. Instead, we meet up in person to make progress on our CanSat, working on our individual tasks separately.

Although a large portion of our group attends the same school, we have opted not to work on this project at school due to project safety and time constraints. Instead, we meet up in person to make progress on our CanSat, working on our individual tasks separately.

1.2 Mission objectives

We primarily aim to benefit pilots and aviation by providing them with useful information about the conditions they are flying in. Takeoff and landing are some of the most dangerous periods of flight, so collecting relevant atmospheric information at low elevations is critical to minimizing errors and injury.

Our goal is to provide a simple, open-source, exceptionally cheap, upgradable, and repeatable method to collect data at low elevations, so that both takeoff and landing can be performed with greater security. Our CanSat measures air temperature, pressure, and latitude and longitude via sensors and relay this information to our ground station. Using these values, we will derive wind speed and direction, wind shear, elevation, and turbulence. We will plot the flight path and corresponding data on a simulation of the surrounding area on our ground station, making it easier to visualize areas of high wind shear or other forms of variance. This data will then be transformed into a standardized AMDAR signal that can be easily sent to and understood by pilots.

The CanSat is to be no higher than 115 millimetres and its diameter no larger than 66 millimetres. The weight should be between 300 grams and 350 grams. The CanSat will be launched from a rocket and deployed at approximately 1,000 metres. The CanSat's recovery system includes a parachute attached to the actual CanSat by various paracords. Optimally, the CanSat is to descend at a rate between 5 meters per second and 8 meters per second and during its descent, it should be taking measurements constantly and simultaneously transmitting it to our ground station via radio.

2 CanSat Description

2.1 Primary Mission

The primary mission of our CanSat is to collect temperature and pressure data while our CanSat is in the air after deployment, and transmit that data to the ground station at 1 Hz.

2.2 Secondary Mission

The secondary mission of our CanSat is to measure AMDAR information at various altitudes. This data will be greatly beneficial for pilots as they have more accurate readings on the conditions near takeoff and landing. Since the general audience for this data are pilots and ground stations, we will transform the data into various palatable forms such as AMDAR strings, AMDAR binary, and a 3D map visualization.

AMDAR is an acronym for Aircraft Meteorological Data Relay, and is a system created by the World Meteorological Organization. It is used to collect data such as air temperature, pressure, wind speed and direction, turbulence, and wind shear. However, most AMDAR data collections happen up in the air in commercial aircrafts, and then sent all the way down to ground stations. Since we will not be going up as high, our data will be more applicable and more accurate to conditions near the ground for takeoff and landing purposes.

METAR is a format for reporting weather conditions, and is the most common format in the world for transmitting these weather observations. METAR is used mainly by pilots and meteorologists, but requires the same data as AMDAR- the only difference is how the information is compiled and presented. For that, we decided to create programs that can do the organization and translation of the data for us.

The raw data we plan on measuring depends on sensors as most other measurement instruments are either too big for our CanSat project or simply cannot take accurate readings as we are falling at a rate of approximately 8m per second.

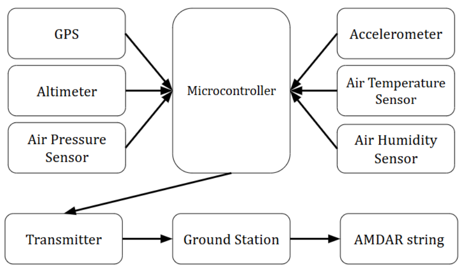

Figure 1: Outlines Data Flow

3 Structural Design

3.1 Initial Design

The creation of our initial design of our CanSat was a lengthy process. Our CanSat itself does not have many components, but due to the small stature of the Sat, structural constraints, as well as various discoveries made along the way, there were many iterations of our design.

Our CanSat design was to be split into two main segments, an 'Outside', consisting only of an outer shell, and an inner shell, containing all of the important structural components of our CanSat. Each of the components would be 3D printed, with various different types of infill to ensure the most structural integrity without exceeding the weight limit.

3.1.1 Outside

The Outside 'Skin' of our CanSat was designed in a way to keep the electrical components of our Sat from flying freely. We decided to make the 'Skin' of our Sat threaded, intertwining with our Inside in order to make a solid structure. The reasoning behind using thread is because not only is it incredibly strong against vertical/downwards force, the inside could be easily separated, revealing the components and wiring in case of any changes that needed to be made.

Creating and finding the specific dimensions of what needed to be 3D printed was a large amount of trial and error. The skin's overall diameter is 65 mm. While the size requirement of our Sat is 66 mm in diameter, we decided to shrink it by 1 mm just in case the 3D printer led to expansion of the filament. However, we did take the design all the way up to the 115 mm height limit.

We had originally created the 3D model of the Skin to have a 5 mm thickness, but decided to shorten it down to 3mm due to weight and space requirements. Furthermore, we also decided to add 6mm of extra offset at the bottom of the design to allow the nuts and bolts of our inner design to screw in properly.

The threading type that we used for both the outside skin and the inside caps is an ISO Metric Profile thread with a size of 62 mm, designation M64x4, and a class of 6g. The direction of the threads was right hand oriented.

Figure 2: 3D Representation of Outside

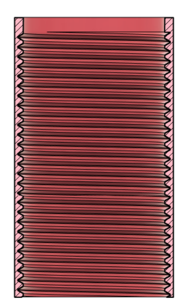

Figure 3: 3D Representation of Outside (Cross Section)

3.1.2 Inner Design

The inner design of our CanSat is divided into two parts using three 'Caps'. These caps are used to split between the actual electrical components and the parachute attachment software. Each cap that is used has a diameter of 59.5 mm. This smaller size was to account for 3D printer tolerances. While in theory the threads would be perfectly fine screwing in if they were 62 mm in diameter, we discovered in our testing that the caps printed at that size would be too large to screw into the holes. As such, we opted for significantly smaller models which screw in smoothly.

Each of the three caps that we have consist of four, 6mm in diameter holes. These holes are to allow our structural supports, four metal rods, to fit through. We decided on four metal rods due to the large amount of structural stress that CanSats have often faced during launch.

Implementing the metal rods and using them as structural supports also allowed us to stack and model electrical components around the structural points. Similar to studs in a house, the metal rods held up our Sat, with the metal's tensile strength being the main contributor of our structural stability. Despite the varying design of all of our caps, the four holes are the only components of each that are perfectly aligned and consistent.

The Bottom cap of our design was designed with 7 holes engraved at the bottom. The first four holes are for the metal rods, but we also have three holes engraved in an arc between two of the holes designated for the metal rods. The holes are also 6mm in diameter and are designed for LED lights. The LED's will shine on and off depending on which components are on at what times. The thickness of the bottom cap is 4.5 mm.

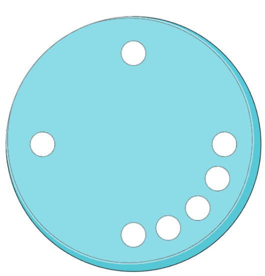

Figure 4: 3D Representation of Bottom Cap

The middle cap is designed as a separation between the 'electrical' component of the innards, and the parachute attachment system. It, as well as the four holes for the metal rods, has a large rectangular hole in the center. The hole is 50 mm in length and 20 mm in width. The hole is to allow the battery that we had planned to use through, as the entirety of the battery pack was around 100mm tall, and needed to poke through the top of the electrical section of the inside. The thickness of the middle cap is 5 mm. The Middle Cap has the highest thickness of all of the caps because we wanted at one of the caps to have a large amount of structural integrity.

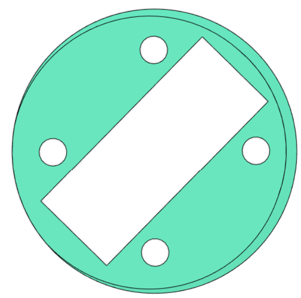

Figure 5: 3D Representation of Middle Cap

The top cap of our design was created in order to not only attach the parachute to the top of the Sat, but to also allow the main power button through as well. The button has, on the inner arc of the main holes, 4 additional spaces, each 5 mm in diameter, to allow the threads for the parachute through and into the main inner design. Along with that, there is one final, 15 mm large circular hole off the side of the cap. This is designated for the power button. The thickness of the top cap is the smallest of the three due to it not holding many of the components in place. If the top cap were to break, it would cause the least amount of problems, so we reduced its thickness to reduce excess weight.

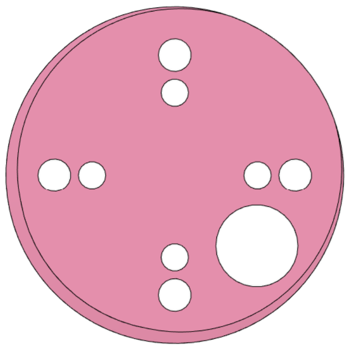

Figure 6: 3D Representation of Top Cap

3.1.3 Full Design

As mentioned earlier, we had planned for 2 full sections of the inside of our CanSat. These parts would separate the electrical components and the components of our parachute design. For the top half of the Sat, the Parachute cords are connected through each designated hole then tied to the metal rod using a clove hitch knot. The knot used is not only incredibly strong, but also works rather well with the threaded design of our metal rods. The top half is 10 mm in length, as clove hitches do not take up much space.

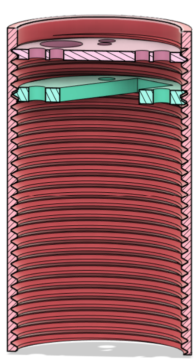

Figure 7: 3D Representation of Parachute Components, Cross-Section

The bottom half of the Sat is allocated entirely for electrical components. The battery is placed in the center, with the other various parts being placed around the battery and the metal rod components. This also makes the CanSat bottom-heavy, preventing unwanted spinning. The bottom of the metal rods are sealed with nuts beneath the bottom cap in order to increase the strength of the Sat as well.

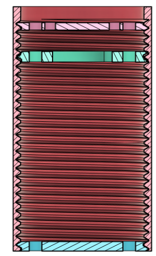

Figure 8: Cross-Section of Full Design

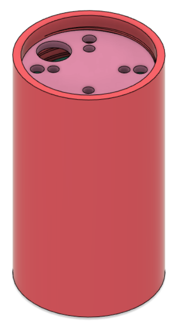

Figure 9: Aerial View of Final Design

3.2 Finalized Design Changes

While we had an original plan, many things changed drastically whilst launch day approached. Not only did components malfunction, many last minute changes were required to be made in order to fit weight requirements and space requirements as well. In order to fix these issues, many drastic changes were made to our initial design:

- **Our Middle and Top Caps were removed:** In order to maximize space we had to remove the top and middle caps. As those were not particularly necessary for our structure, their loss, while missed, was not entirely detrimental to our end design.
- **Two of our metal rods were removed:** The metal rods increased our

overall weight a lot. As such we decided that it would be best to remove it. Two of them were still kept in order to ensure max structural stability.

4 Electrical Structure

Component	Function
BME280	Pressure, temperature, and humidity sensor.
NEO-6M	GPS module and altimeter
LoRa E32-900T30D	Data transmitter
Raspberry Pi Pico	Main microcontroller
Power Button	Turns CanSat On/Off
Radio Antenna	Extends range of radio module

Table 1: Components Used

4.1 Electrical Diagrams

- *Note Diagrams are of the chip , and not of any surrounding PCBs (with the exception of the Raspberry Pi Pico and E32-900T30D), underneath diagrams are the connection points that our team used which are more representative of the PCB the chip is on

We have opted to use an BME680 that we ordered externally for the measuring of air pressure, temperature and humidity, as well as for altitude measurements. This choice was based mostly on the fact that this single part can measure many of the parameters that we need in order to produce meaningful data for our AMDAR reports. Additionally, the leap in accuracy compared to the BME280 was another factor that led to the change in parts. Due to the fact that this part was not included in the CanSat kit, the price did factor into our decision and at only $11, the BME680 surpassed our expectations in terms of accuracy and reliability.

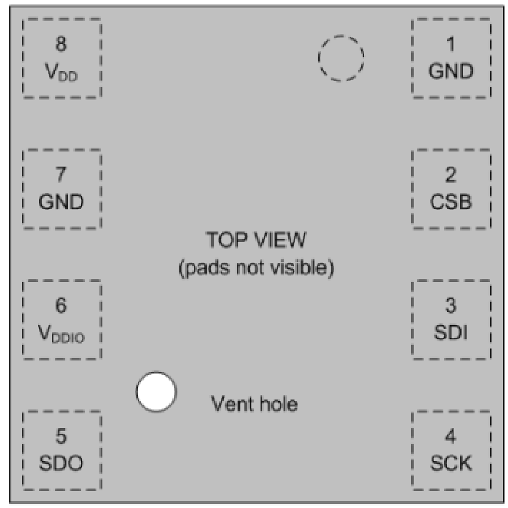

Figure 10: Top View of BMP680

- **VCC:** Power input for the component
- **SDA:** Transfers data between BME680 and microcontroller
- **SCK:** Provides timing signal for data transfer:
- **GND:** Provides safe path for electricity to leave

To pinpoint our CanSat's position as it is falling, we decided to use the NEO-M8N GPS module. As this part was not part of the provided CanSat kit, this part had to be bought online. Due to that, we considered the price to be one of the more important aspects of our decision making. At only $12 this part exceeded our expectations for what we could do with as minimal a cost as possible. The GPS is being used in order to calculate wind speed through the horizontal movement of our CanSat. This wind speed will then be used to calculate wind shear, which can prove to be a very important piece of information for pilots, especially during takeoff and landing.

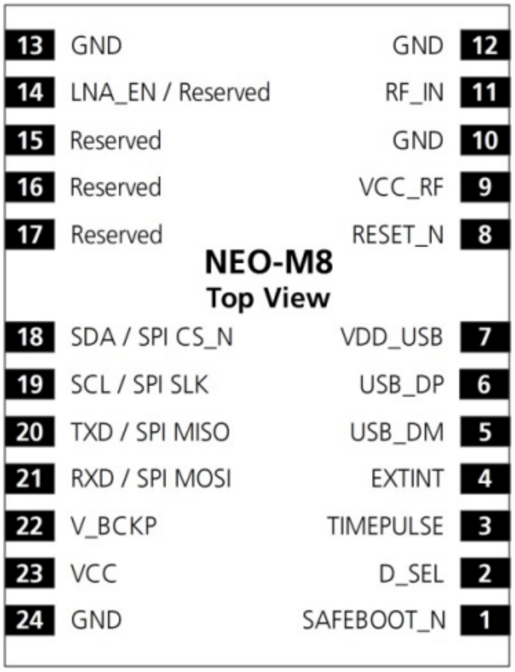

Figure 11: Top View of NEO-M8N

- **VCC:** Provides power to the component
- **RXD:** The UART input of the module, allows module to receive data from the microcontroller
- **TXD:** The UART output of the module, allows module to transmit data to the microcontroller
- **GND:** Provides safe path for electricity to leave

To transmit our data to the ground station, we will be using the LoRa E32-900T30D module. This module is very cheap and due to the fact that it is a LoRa module, it is more than capable of the transmission power needed for the distance our CanSat will be in the air. Additionally, it will be outfitted with a small antenna which will boost transmission range. This transceiver will communicate with a similar module on a ground station, attached to another Raspberry Pi Pico. This ground station is fitted with a significantly stronger antenna than the CanSat itself, which will help the CanSat communicate more reliably with the ground station.

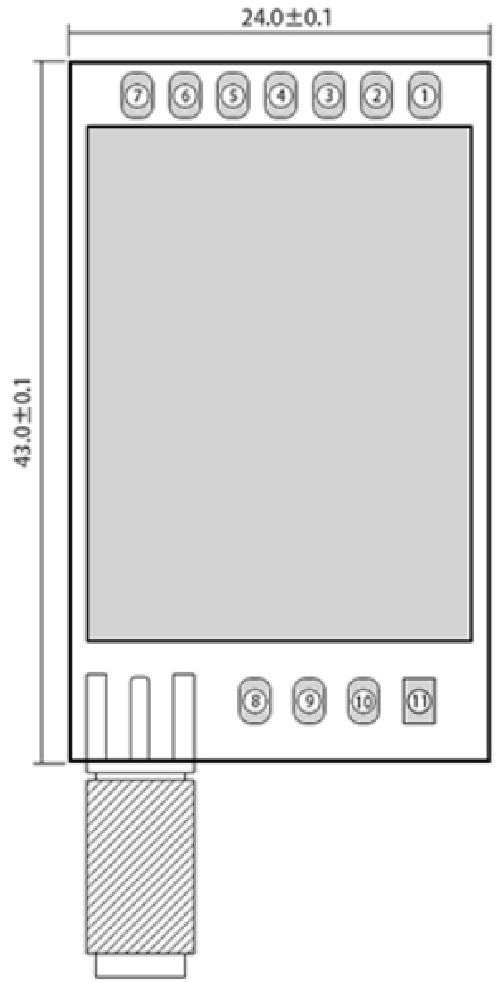

Figure 12: Top View of LORA E32

- **1: M0** Used to determine the mode of the module along with M1
- **2: M1** Used to determine the mode of the module along with M0, both M1 and M0 are required to set the frequency
- **3: RXD** The UART input of the module, allows module to receive data from the microcontroller
- **4: TXD** The UART output of the module, allows module to transmit data to the microcontroller
- **5: AUX** Indicates the working status of the module
- **6: VCC** Power input for the module, voltage ranges between 3.3V 5.5V

- **7: GND** Provides safe path for electricity to leave

To act as a microcontroller that will send all the data from the sensors to the transmitter, we have chosen to use the Raspberry Pi Pico. This part was chosen as it was readily available on hand as well as being relatively inexpensive should we need multiple, which did end up being the case for the ground station. This microcontroller is very capable of much more than what is needed for this project in terms of available pins as well as processing power. The support of different communication protocols allows us to be flexible with our part selection as we are not limited to only I^2C or only SPI protocols.

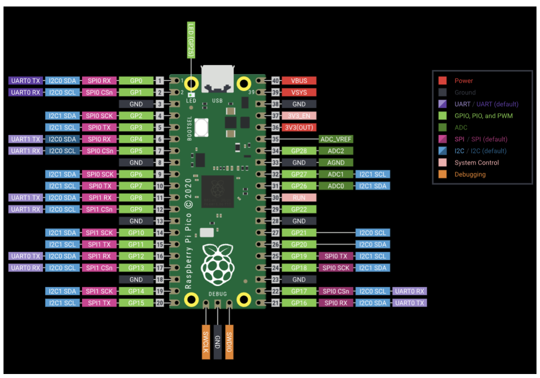

Figure 13: Documentation of PICO W

Device	Volta (V)	Current (mA)	Power (mW)
Radio Transmitter (E32-900T30D)	5.0	650mA TRC/25mA TReC*	Calc.[1] 139.6mW
GPS (NEO-6M)	3.3	12mA**	3.3 × 12 = 39.6 mW
Barometer and Air Monitor (BME280)	3.3	3.7 µA**	3.3 × 3.7 = 12.21 µW
Total	**5.0**	**675.0037**	**179.21221 mW**

Table 2: Power Consumption for Payload Components

- *TRC = Typical Response Current. TReC = Typical Recieving Component
- **Transmitting at 1 Hz

We will be using the IMREN 18650 batteries for our Cansat. Each cell has a nominal rating of 3500mAh and 3.7V. Compared to our previous design, we have ditched the battery housing in favour of a single battery. Due to this decision, our parts will have a less stable voltage resulting in inconsistencies in performance, however, we will have more room for the components. This Calculating battery life:

$$BatteryLife = \frac{BatteryCapacity}{CurrentDraw} = \frac{3500mAh}{675.0037mA} \approx 5.2h$$

5 RF Design

We will be using the EBYTE E32-900T30D at 911 MHz for our radio link. As referenced for the power consumption calculations, our downlink data rate will be at 9.6k bits per second (bps). For our radio module, we will be using the LoRaWAN protocol for its wide support and low power consumption.

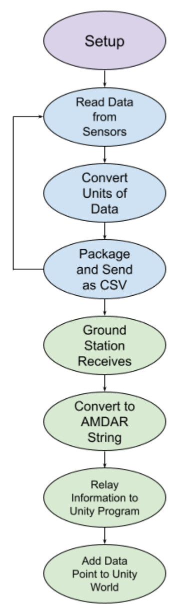

Figure 14: Design Flow Diagram

[1] Assuming 9.6 Bytes/Second transmit rate, and packet size of 45 bytes:

$$5V\left(\left(\frac{45\,\text{bytes}\times 1\,\text{Hz}}{9600\,\text{bytes/sec}}\right)\times 650\,\text{mA} + \left(\frac{9600-(45\,\text{bytes}\times 1\,\text{Hz})}{9100\,\text{bytes/sec}}\times 25\,\text{mA}\right)\right)$$

6 Software Design

The CanSat will send relatively unprocessed information to the ground station to minimize its chance of error. It uses a Raspberry Pi Pico, and is programmed using MicroPython on the Thonny IDE. Our ground microcontroller, another Pico programmed with MicroPython on Thonny, will solely receive data from the radio. We will process this data using a Python script, deriving all required components to create a valid AMDAR message, and to package the data for use for the C# Unity simulation software. This software will display the CanSat's flight path, and be easily readable for pilots containing key data, including wind shear.

Data conversion and communication with sensors is largely handled through libraries. Primarily, we used an EByte LORA E32 Micropython library, a Micropython BME680 driver, and a Micropython GPS library. These libraries handle parsing data and communication protocols in themselves, and the CanSat's software polls these libraries as is required.

The data received by the CanSat will be converted to a CSV string of the following format: "<Time>, <Temperature>, <Humidity>, <Pressure>, <Latitude>, <Longitude>, <Speed >, <Speed Direction>, <Newline >". This will be packaged into a data packet, sent to our radio module, and relayed to our ground station. The ground station will then unpack this data and begin calculations.

Our Python script processes all of the received data into two text files. Firstly, we create an AMDAR file matching the specifications that were previously mentioned. This can be processed into existing models in this format. Next, we create a simplified version of the raw CanSat data for use in the Unity program.

We are calculating the ground speed of our CanSat using our GPS. It sends the magnitude of speed and the track angle, which we send to the ground station for decomposition into component speeds via trigonometry. We assume that the horizontal speed of the CanSat will closely match the horizontal wind speed at a given elevation. Our parachute will maximize time spent in the air to ensure the CanSat's speed will reach equilibrium with that of the surrounding air.

We can also determine pressure altitude via the barometric formula:

$$P = P_0\left(1 - \frac{Lh}{T}\right)^{\frac{gM}{RL}}, \quad \text{or} \quad h = \frac{T_0}{L}\left[1 - \left(\frac{P_0}{P}\right)^{\frac{RL}{gM}}\right].$$

where g is gravitational acceleration (9.81 m/s^2), M is the molar mass of air (0.0289644 kg/mol), R is the universal gas constant (8.314 J/molK), L is the temperature lapse rate (roughly 0.0065 K/m), and P_0 is the sea level pressure (101325 Pa). Reducing the constants brings us to an equation for height as a function of pressure:

$$h = 4430 \times \left(1 - \left(\frac{P}{101325}\right)^{0.1903}\right)$$

After forming the data, we will prepare it into a string compliant to AMDAR specifications. The following tables list the specifications for AMDAR strings according to the AMDAR Onboard Software Functional Requirements Specification. Below are tables of specifications:

Table 19: AOSFRS Basic Observation Sequence format

Character number	# of char	Content	Format	Notes	Example
1	1	Observation Type Indicator	N	1	0
2-8	7	Latitude in seconds	SNNNNNN	south is negative	-2976
9-15	7	Longitude in seconds	SNNNNNN	west is negative	6081
16-22	7	Day/Time	NNNNNNN	2	879661
23-26	4	Altitude in tens of feet	NNNN		3899
27-30	4	Static Air Temperature tenths of C	SNNN		-525
31-33	3	Wind direction	NNN		160
34-36	3	Wind Speed	NNN		025
37	1	roll angle flag	N	see paragraph 3.2.4.4	U

Figure 15: Observation Sequence Format

Table 20: AOSFRS Optional Parameters Format

Header Indicator	# of char	Content	Format	Notes	Example
N/A	1 *or* 9	routine EDR	C *or* CNNNNNNNN	1	E12345678
A	3	DEVG	NNN		7
B	3	true airspeed	NNN		854
C	4	true heading in tenths of degrees	NNNN		145
D		GNSS altitude	NNNN		3750
E	1	anti-ice	N	2	1
F	2	A/C configuration indicator	NN	see paragraph 3.2.4.5	01
G	6	water vapor	NNNNNQ	see appendix C.2	123450
H	6	relative humidity	NNNNNQ	see appendix C.2	05000U
I	1	icing	N	3	1
J	4	roll angle	±NNN	Aircraft roll angle reported in whole degrees	-005
K	3	pitch angle	±NN	Aircraft pitch angle reported in whole degrees	+05
L	5	vertical speed	±NNNN	Aircraft vertical speed reported in ft/min	-100

Figure 16: Optional Parameters Format

Beyond AMDAR, we will also collect other data for use in our Unity simulation. By comparing component wind speeds, we can determine wind shear between any two measurements. This can be determined via simple geometry:

$$Magnitude = \frac{\sqrt{\Delta U^2 + \Delta V^2}}{\Delta h}$$

$$Direction = \arctan\left(\frac{\Delta V}{\Delta U}\right)$$

where U and V are component ground speeds. This data is plotted in 3D, converting latitude, longitude, and height to Unity worldspace points. We are using Mercator projection to map geodetic coordinates to Unity's 3D worldspace, according to the below formulas:

$$\Delta lat = (lat_2 - lat_1) \times (\frac{\pi}{180} \times R)$$

$$\Delta lon = (lon_2 - lon_1) \times (\frac{\pi}{180}) \times R \times cos(lat_1)$$

The simulation uses a 3D topographical map of the launch site collected from OpenTopography. A roughly 20 km by 20 km area was selected, with a maximum elevation of 982 m. The raw form of the collected heightmap is displayed on the right.

After this, we collected imagery from the Sentinel 2 satellite in the same location, and cropped it to match the area we previously selected. We combined the photography featuring spectrums of visible light to create a photorealistic simulation of the launch area terrain. This data was collected from the EU Copernicus Data Space Ecosystem Browser.

GPS points received from the CanSat will be converted into worldspace points on the Unity simulation, and when interacted with will display information regarding the CanSat at that position, primarily wind shear.

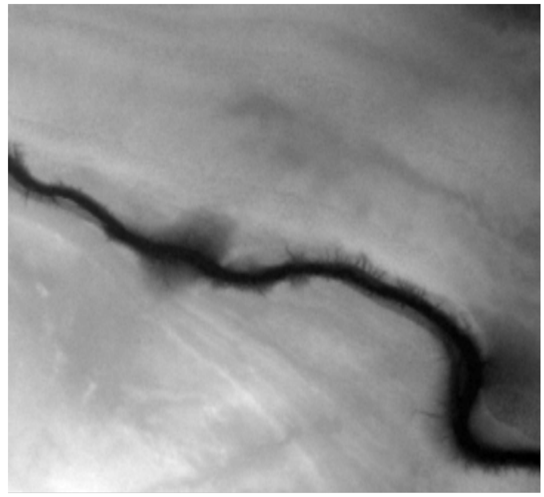

Figure 17: Launch Area Terrain #1

Below are top-down and tilted perspectives of the 3d rendered launch site terrain in the simulation:

Figure 18: Launch Area Terrain #2

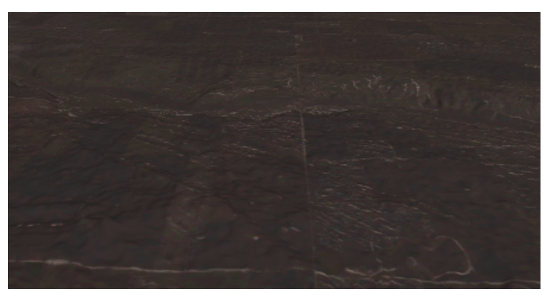

Figure 19: Launch Area Terrain #3

We expect to send data at 1 Hz, and at a descent rate of roughly 5 m/s, we estimate around 200 points of data to be sent to the ground station. Not all of these points are displayed on the Unity simulation: we will only display about 20 that can be interacted with, but the other points will help us create AMDAR data and complete the primary mission.

Storing all of these data points will not be burdensome on our ground station. Considering the size of the CSV strings, all of the data should be able to be contained in less than 75 kb.

7 Recovery System

Once the CanSat is ejected, a parachute opens up and controls our descent. The parachute is rectangular with a hole to control spin and is attached to the CanSat via a paracord. The parachute itself is made of ripstop nylon due to its great durability and resistance to wear and tear from the wind. We attached the paracord to the parachute by making 4 holes at each corner of the parachute, tying the paracord into a reef knot, and then tying it to a swivel hook for extra resistance to spinning. To mount the parachute to our CanSat, we ended up using clove hitches to tie our swivel hook to the metal rods within our CanSat. No matter how tight our knots are, there is a chance that they may untie themselves while falling, so we burned the paracord to weld the knots together. Additionally, we made the parachute bright pink to increase the visibility of our Sat.

The shape of our parachute was initially circular, but we quickly found out that a circular shape only hinders us as it is incredibly difficult for us to cut it out properly and it requires more connection points, increasing chances of tangling on descent. Thus, we decided on a rectangular parachute. The rectangular shape allows for a stable and slow descent, and the hole in the middle would allow for further increased stability and reduced oscillations, leading to the safest, steadiest possible trip down.

To figure out the optimal dimensions of the parachute we used this formula to find terminal velocity:

$$Vt = \sqrt{\frac{2mg}{pACd}}$$

or, rearranging for the area of the parachute,

$$A = \frac{2mg}{pv^2CD}$$

where p is the air density, measured in the mass per cubic meter of air (1.225kg/m^3), A is the area of the parachute in square centimeters when laid flat, Cd is the drag coefficient, a value that describes and quantifies the resistance of an object as it blows in the wind, (roughly 0.75), m is the mass of the CanSat (roughly 0.35kg), and g is the acceleration due to gravity (9.81m/s^2).

Assuming terminal velocity will be 5m/s, our parachute should have an area of approximately 0.30m^2. Using geometry, $A = lw$ should be our formula, but since our parachute side lengths are close to equal, we assumed $l = w$. Therefore the side length of our parachute should be around 54.5cm. Our spill hole is still circular, using a radius of 5cm as we found that during testing, it allowed us to reach a terminal velocity within the 5-11m/s range.

The expected descent rate would be 5m/s, and the flight time would be approximately 216 seconds.

As for locating the CanSat, our bright pink parachute will increase its visibility in the air and the GPS will help us locate it once it falls to the ground.

8 Project Planning

8.1 Time Schedule of the CanSat Presentation

Date	Milestones
March 1	Final prep meeting
March 9	Preliminary Design Report due
March 15	Finalize design
March 23	Finalize models for prototypes, begin creating them
April 11	First Presentation at McKernan Junior High
April 13	Finalize prototypes, begin final construction
April 14	Second Presentation at Old Scona Academic (Online)
April 16	Third Presentation to Roberta MacAdams Elementary
April 24	Head to Lethbridge

Table 3: Timeline of Events

8.2 Budget Sheet

Component	Cost
BME680	$14.45
NEO-M8N	$11.04
LoRa E32	$14.99
DIYmalls 915MHz LoRa Antenna	$6.00
9KM DWLIFE	$16.78
Power Button	$0.10
Raspberry Pi Pico	$4.54
TECEUM Paracord	$10.99
Total	$78.89

Table 4: Costs of Components Used

9 Testing

A variety of tests were performed to ensure the functionality of our CanSat.

Firstly, we tested the power supply of our CanSat. We charged it fully and then left it running for over four hours. Afterward, we measured the remaining battery capacity to ensure the battery life surpassed the required lower bound.

Next, we tested the sensors of the CanSat to ensure they were correctly calibrated. We brought the CanSat to areas with known temperatures, pressures, and GPS positions. By comparing our measured values to the known values, we ensured that all sensors accurately measured their environments.

Afterwards, we tested the radio communications. This involved turning on the CanSat and enabling the receiver on our ground station. We examined the data that the ground station received and how it processed it, including from a distance. This confirmed that the communication systems worked properly and remained accurate, even in conditions similar to the actual launch.

Then, we tested the software. Using a realistically generated set of data that might have been seen on launch day, we checked whether the Unity visualization correctly displayed all of it with reasonable performance. We also ensured that wind shear calculations and other derivations were accurate with the generated data.

Finally, we conducted environmental and flight tests for the CanSat itself. We dropped the CanSat or a test model with similar dimensions and weight from a moderately high height, modifying the code to release the parachute at a low vertical velocity. This allowed us to test the efficacy of the release mechanism and the parachute in a low-stakes situation.

10 Outreach Programme

The team of Can'tSat firmly believes that outreach and communication are integral to the success and impact of our project. CanSat, as a program, is inherently designed to inspire and cultivate the next generation of engineers, scientists, and innovators. To achieve this goal, Can'tSat is meticulously documenting every critical aspect of our project - from the initial design phase and preparation to the execution and final results. By sharing our journey, including the challenges we face and the milestones we achieved, we aim to not only educate others about aerospace technology but also ignite a passion for STEM in many others, similarly to how CanSat has already done for us.

Our team leveraged multiple platforms and strategies. Through physical presentations, we shared our findings, experiments, and the overall canSat experience to a wide range of audiences. We tailored our presentation to engage not only younger students- who are the future of STEM- but also peers, educators, professionals, and community members of all ages. Our elementary presentation was done at Roberta MacAdams Elementary, presenting to a group of grade 6 students, as well as many administrators that were in the room. Our junior high presentation was conducted at McKernan Junior High, detailing our project to a group of combined grade 9 and grade 6 classes. Our third and final presentation was completed online to our peers in Old Scona Academic High School, asking fellow grade 11 students with backgrounds in computer engineering and presentation to grade and give feedback to our group.

In addition to our in-person presentations, we utilized digital platforms to extend our reach even further. We contacted businesses to seek potential sponsorships and shared our team through documentations on our YouTube and Instagram platforms.

We at Can'tSat combined physical presentations with a robust digital presence to create a multifaceted outreach strategy that educates, inspires, and connects with diverse audiences, showcasing the value of our project, emphasizing the importance of STEM education and innovation , and hopefully leaving a lasting impact by encouraging others to pursue their passions in science and engineering.

Please support our Platforms!:

- https://www.youtube.com/@Cantsatteam
- https://www.instagram.com/_can.tsat66303/?hl=en