UK CanSat Competition

Team Re-LAACS

(Remote Low-altitude Atmospheric Composition Sensing)

Tonbridge School

Critical Design Review

Date:

31/01/2025

Contents

1 INTRODUCTION

Team Organisation and Roles

Project Manager - **Ming**

Overall responsibility for keeping project on track

Co-ordinating activities of the team

Creating project progress/gant chart

Updating chart regularly with progress of each strand of activity

Flagging deadlines

Ensuring Budget on track (£400)

Physical Construction Lead – **Lev**

Design of can internal and external structure

Selection and manufacturing of components for can structure

Ensuring robustness and fit of all components

Ensuring can complies with competition guidelines (size, weight etc)

Contributing to CDR report

Landing Systems Lead- **Joe**

Design of can landing system (parachute and attachments)

Selection of materials

Construction and testing

Compliance with competition guidelines (10m/sec descent speed)

Contributing to CDR report

Data Communications Lead- **Leo**

Design and construction of appropriate RF data links

Ensuring data transmitted and received securely

Ensuring RF comms complies with competition guidelines

Contributing to CDR report

Software Lead- **Ethan**

Choice of Operating System

Design, creation and testing of all systems software

Ensuring all critical software securely backed up

Contributing to CDR report

Systems Integration Lead **Samuel**

Choice of system computing components

Choice of required sensors/actuators

Ensuring compatibility with systems software

Ensure robustness and testing

Contributing to CDR report

Electrical Lead- **Mikel**

Design and construction of electrical systems

Ensuring robustness and reliability

Ensuring compliance with competition guidelines (external switch and 3 hrs duration)

Contributing to CDR report

Outreach Lead- **Ming**

Responsible for social media profile

Raising profile of the project

Liaising with external bodies/Stem ambassadors

Creating podcast

Contributing to CDR report

1.2 Mission Overview

1.2.1 Mission Objectives

The objectives are:

1. Recording atmospheric air pressure and temperature during a descent from high altitude and: -
2. Investigate the extent of human activity on the air quality of remote areas

Objective 1 is a science experiment that will provide data that can be used to test assumptions about changes in atmospheric pressure and temperature in relation to altitude.

Objective 2 is a science experiment that will use a range of sensors to carry out an atmospheric composition analysis at different altitudes using a GPS sensor to provide location details. The CanSat is to collect relevant samples, analyse them on board the CanSat then transmit the results to a base station.

Primary Mission:

For the duration of the CanSat launch and descent, the CanSat will automatically measure and record atmospheric pressure and temperature. It will be taking measurements at the rate of one per second.

These data readings will be saved to a file on the CanSat as well as transmitted to a ground station using an RF link.

The data will be saved to a file for later analysis.

Secondary Mission:

For our secondary mission, we decided on utilizing the CanSat to monitor air quality in various remote locations such as the middle of the ocean or in areas where there is a very low density of air quality monitoring stations such as Malaysia[1]. The objective is to determine the effect of human action on the air quality of even these less-covered areas. The low production cost of our CanSat would mean that you could deploy one in a remote location periodically and utilize the data it collects to supplement the global database of air quality levels rather than sending a team of people out there with instruments. We have also designed our own high-gain radio antenna to facilitate signal transmission from very long distances. We are also designing a system to determine the position and travelling direction of the CanSat via monitoring the delay between the requesting and receiving of a byte of data to the CanSat and back and dividing against

1 Narashid, Rohayu Haron, and Wan Mohd Naim Wan Mohd. "Air Quality Monitoring Using Remote Sensing and GIS Technologies.".

the known speed of the radio waves. we receive from the CanSat to ensure that we can always confirm the CanSat's geographical location precisely, even if the on-board GPS is unable to function.

1.2.2 What will you measure, why and how? (Contribution from MC)

For the primary mission, we're measuring the atmospheric pressure and temperature using a BME680 sensor.

For the primary mission, we're measuring the atmospheric pressure and temperature using a BME680 sensor.

To succeed in our Secondary mission, we will have to successfully measure the concentration of substances such as Ozone, particulate matter concentration, Carbon dioxide, and VOC concentration at regular intervals during the descent of the satellite.

These criteria are the primary metrics for the monitoring of air quality in most ground based air quality monitoring stations[1].

We have also decided that these gases are also the most relevant in terms of how humans can affect the climate and the health of other humans.

Ozone is not only a greenhouse gas, worsening climate change, but also worsens crop yields by necrotizing the leaves[2].

Particulates, other than being strongly correlated with respiratory disease in humans, can also adversely affect ecosystems through the deposition of particulates and its subsequent uptake into plants or water. Metal and organic compound particulates have the greatest potential to alter plant growth and yield[3], and these particles generally exist in the 2.5-10 μm range, precisely what our chosen sensor detects.

VOCs are major contributing factors to the formation of ozone and fine particulates in the air[4] .

carbon dioxide, while not being a pollutant, is a greenhouse gas that is produced in almost every aspect of human life, from breathing to driving, hence we wanted to also track its concentration to compare against population density to measure the extent to which its concentration is correlated with human presence.

Since the concentrations of these variables we are measuring can also be impacted by many non-human events such as weather, far-away transport of volcanic smoke or Sahara dust, we could launch the satellite in two locations, one in the populated area we wish to measure and the other in a location with very little human activity, say the middle of the pacific, and quantify the magnitude of the effect of human actions on air quality by measuring the differences between the data collected by the two satellites.

Find or derive a formula to convert the collected measurements to AQI (Air Quality Index) measurements in real time.

Accurately track the geographical location of the CanSat while it is in the air to ensure we can properly map the data it collects to its geographical location

[1] "环境空气质量标准" [Ambient Air Quality Standards] (PDF). *Ministry of Environmental Protection of the People's Republic of China*

[2] “Low-Altitude Ozone | Airparif.” Airparif.fr, AIRPARIF, 2019, www.airparif.fr/en/lozone-de-basse-altitude. Accessed 8 Jan. 2025.

[3] California Air Resources Board. “Inhalable Particulate Matter and Health (PM2.5 and PM10).” Ca.gov, 2015, ww2.arb.ca.gov/resources/inhalable-particulate-matter-and-health.

[4] Environmental Protection Department. “Volatile Organic Compounds and Smog | Environmental Protection Department.” Epd.gov.hk, 2009, www.epd.gov.hk/epd/english/environmentinhk/air/prob_solutions/vocs_smog.html.

2 PROJECT PLANNING

The students are timetabled 2 hours per week in term time. They also have access to the project materials and the DT labs in the school during their free time, which they will increasingly have to make usage of as deadlines approach.

Broadly speaking, we have split the project progress into half terms as follows:

- Interdependencies have been minimized by running all development tasks in parallel up until the final assembly and testing stage in early February
- In the first half term (Sept and Oct 2020) project definition, planning and initial prototyping.
- In the second half term up to Xmas, proof of concept and further prototyping.
- Immediately after Xmas, the software and hardware build phases commence
- Towards the start of February, build completion, debugging and testing must have started

2.1 Project Plan schedule (Contribution from MC)

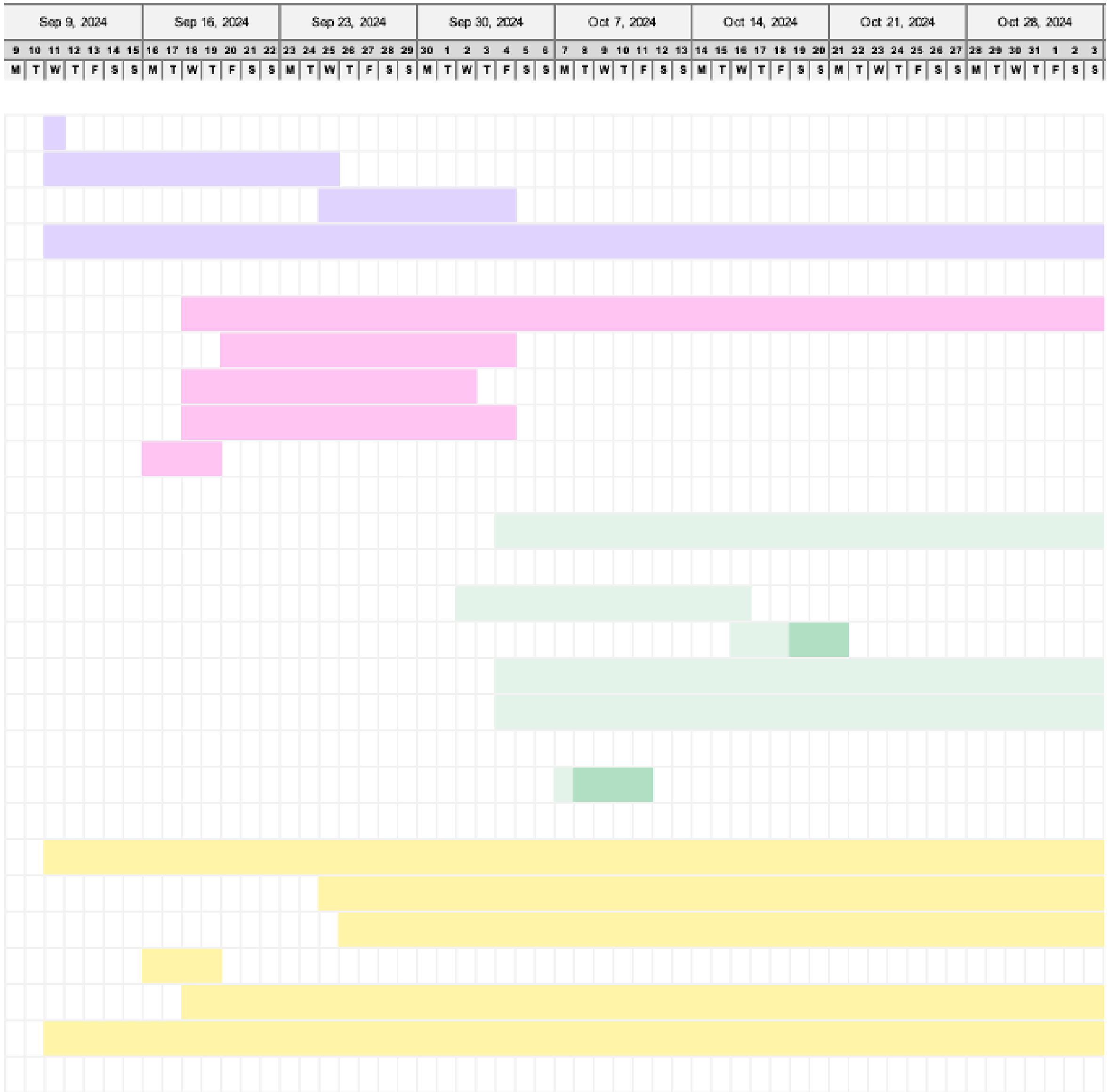
Team ReLAACS

Ming Y. Chan   Project lead

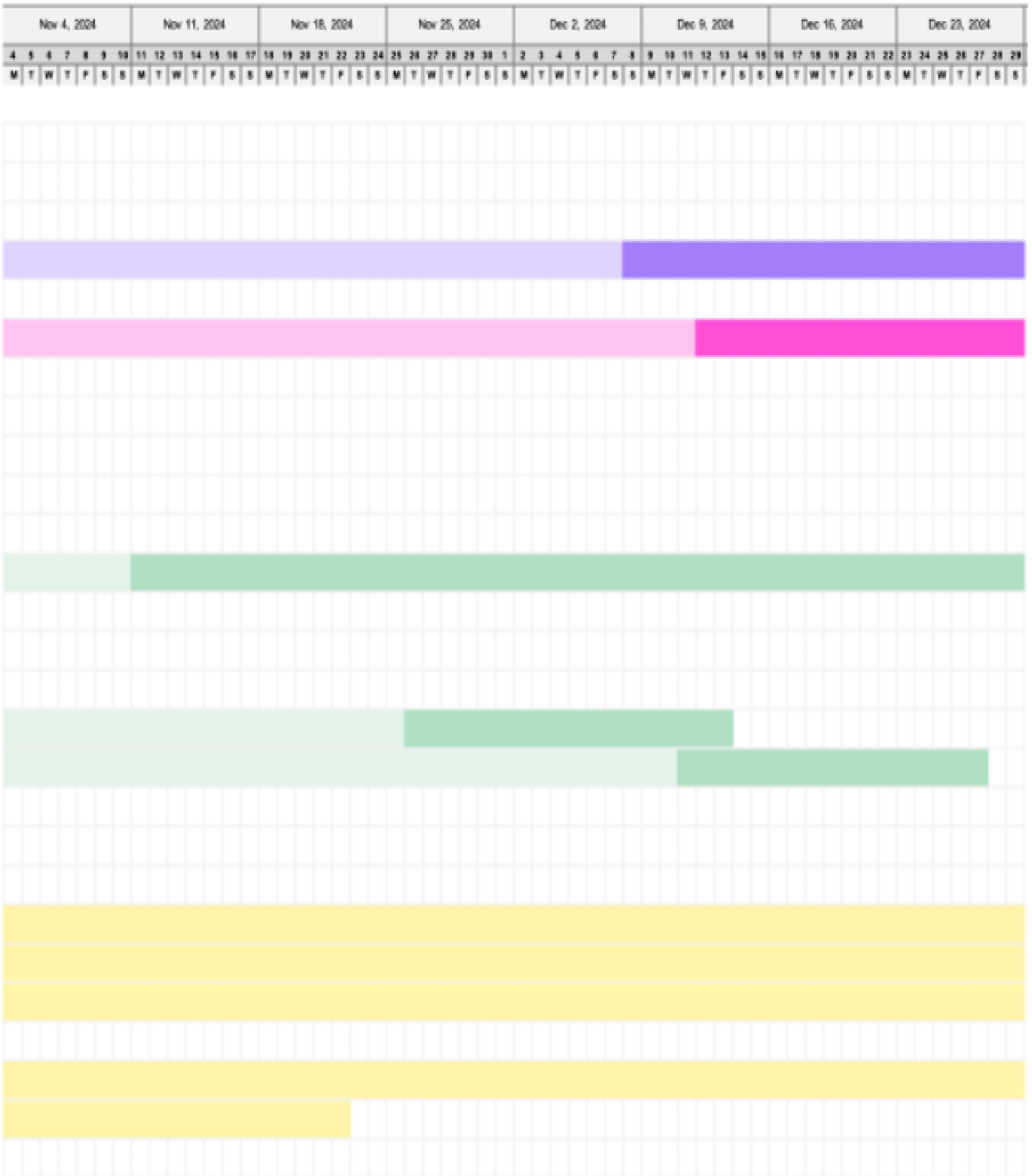
Project start: Wed, 9/11/2024

Display week: 1

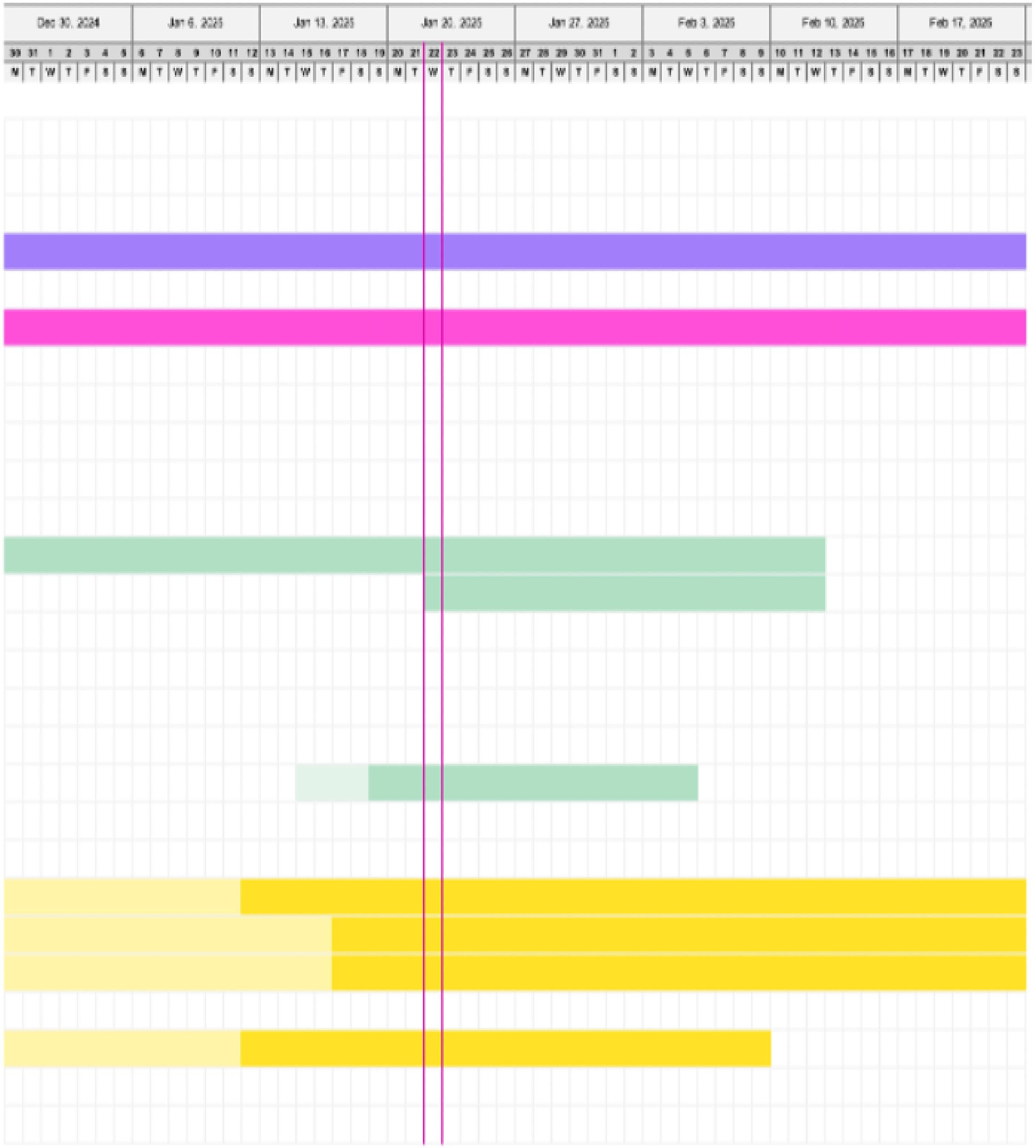
TASK	ASSIGNED TO	PROGRESS	START	END
Initiation				
Role assignment	Everyone	100%	9/11/24	9/11/24
Secondary mission planning	Everyone	100%	9/11/24	9/25/24
Sensor search	Lev Balabin, Samuel Ni, Eli	100%	9/25/24	10/4/24
Outreach	Ming Y.Chan	50%	9/11/24	3/6/25
Planning and design				
Gantt Chart production	Ming Y.Chan, Zijiao Zhou	50%	9/18/24	3/6/25
Mechanical design	Lev Balabin	100%	9/20/24	10/4/24
Parachute design	Zijiao Zhou	100%	9/18/24	10/2/24
Comms planning	Leonid Elkin	100%	9/18/24	10/4/24
Identify initial risks	Ming Y. Chan	100%	9/16/24	9/19/24
Execution				
Mechanical assembly	Lev Balabin, Samuel Ni, Mikel Eluwah	29%	10/4/24	2/12/25
Wiring	Lev Balabin, Samuel Ni, Mikel Eluwah	0%	1/22/25	2/12/25
Parachute assembly	Zijiao Zhou	100%	10/2/24	10/16/24
Parachute testing	Zijiao Zhou	50%	10/16/24	10/21/24
Comms assembly	Leonid Elkin	75%	10/4/24	12/13/24
Software development	Ethan Wang, Samuel Ni	80%	10/4/24	12/27/24
Visual interface development	Ethan Wang	20%	1/15/25	2/5/25
Doppler Effect testing	Leonid Elkin	25%	10/7/24	10/11/24
Evaluation				
Monitor progress	Ming Y.Chan	70%	9/11/24	3/6/25
Track expenses	Ming Y.Chan	70%	9/25/24	3/6/25
Evaluate progress	Ming Y.Chan	70%	9/26/24	3/6/25
Address risks	Ming Y.Chan, Zijiao Zhou	100%	9/16/24	9/19/24
Gather feedback	Ming Y.Chan	80%	9/18/24	2/9/25
Completion of PDR	All	100%	9/11/24	11/22/24
Completion of CDR	All	60%	11/22/24	1/31/24



TASK	ASSIGNED TO	PROGRESS	START	END
Initiation				
Role assignment	Everyone	100%	9/11/24	9/11/24
Secondary mission planning	Everyone	100%	9/11/24	9/25/24
Sensor search	Lev Balabin, Samuel Ni, Ethe	100%	9/25/24	10/4/24
Outreach	Ming.Y.Chan	50%	9/11/24	3/6/25
Planning and design				
Gantt Chart production	Ming.Y.Chan, Zijiao Zhou	50%	9/18/24	3/6/25
Mechanical design	Lev Balabin	100%	9/20/24	10/4/24
Parachute design	Zijiao Zhou	100%	9/18/24	10/2/24
Comms planning	Leonid Elkin	100%	9/18/24	10/4/24
Identify initial risks	Ming Y. Chan	100%	9/16/24	9/19/24
Execution				
Mechanical assembly	Lev Balabin, Samuel Ni, Mikel Eluwah	29%	10/4/24	2/12/25
Wiring	Lev Balabin, Samuel Ni, Mikel Eluwah	0%	1/22/25	2/12/25
Parachute assembly	Zijiao Zhou	100%	10/2/24	10/16/24
Parachute testing	Zijiao Zhou	50%	10/16/24	10/21/24
Comms assembly	Leonid Elkin	75%	10/4/24	12/13/24
Software development	Ethan Wang, Samuel Ni	80%	10/4/24	12/27/24
Visual interface development	Ethan Wang	20%	1/15/25	2/5/25
Doppler Effect testing	Leonid Elkin	25%	10/7/24	10/11/24
Evaluation				
Monitor progress	Ming.Y.Chan	70%	9/11/24	3/6/25
Track expenses	Ming.Y.Chan	70%	9/25/24	3/6/25
Evaluate progress	Ming.Y.Chan	70%	9/26/24	3/6/25
Address risks	Ming.Y.Chan, Zijiao Zhou	100%	9/16/24	9/19/24
Gather feedback	Ming.Y.Chan	80%	9/18/24	2/9/25
Completion of PDR	All	100%	9/11/24	11/22/24
Completion of CDR	All	60%	11/22/24	1/31/24



TASK	ASSIGNED TO	PROGRESS	START	END
Initiation				
Role assignment	Everyone	100%	9/11/24	9/11/24
Secondary mission planning	Everyone	100%	9/11/24	9/25/24
Sensor search	Lev Balabin, Samuel Ni, Ethe	100%	9/25/24	10/4/24
Outreach	Ming.Y.Chan	50%	9/11/24	3/6/25
Planning and design				
Gantt Chart production	Ming.Y.Chan, Zijiao Zhou	50%	9/18/24	3/6/25
Mechanical design	Lev Balabin	100%	9/20/24	10/4/24
Parachute design	Zijiao Zhou	100%	9/18/24	10/2/24
Comms planning	Leonid Elkin	100%	9/18/24	10/4/24
Identify initial risks	Ming Y. Chan	100%	9/16/24	9/19/24
Execution				
Mechanical assembly	Lev Balabin, Samuel Ni, Mikel Eluwah	29%	10/4/24	2/12/25
Wiring	Lev Balabin, Samuel Ni, Mikel Eluwah	0%	1/22/25	2/12/25
Parachute assembly	Zijiao Zhou	100%	10/2/24	10/16/24
Parachute testing	Zijiao Zhou	50%	10/16/24	10/21/24
Comms assembly	Leonid Elkin	75%	10/4/24	12/13/24
Software development	Ethan Wang, Samuel Ni	80%	10/4/24	12/27/24
Visual interface development	Ethan Wang	20%	1/15/25	2/5/25
Doppler Effect testing	Leonid Elkin	25%	10/7/24	10/11/24
Evaluation				
Monitor progress	Ming.Y.Chan	70%	9/11/24	3/6/25
Track expenses	Ming.Y.Chan	70%	9/25/24	3/6/25
Evaluate progress	Ming.Y.Chan	70%	9/26/24	3/6/25
Address risks	Ming.Y.Chan, Zijiao Zhou	100%	9/16/24	9/19/24
Gather feedback	Ming.Y.Chan	80%	9/18/24	2/5/25
Completion of PDR	All	100%	9/11/24	11/22/24
Completion of CDR	All	60%	11/22/24	1/31/24



2.2 Team and External Support

Competencies within the team:

Students are in their first year of studying A levels in:

Computer Science	(LE, EW)
Physics	(JZ, LE, ME, LB, SN, EW)
Chemistry	(JZ, LB, ME, MC)
Biology	(MC)
Mathematics	(LE, JZ, ME, EW, SN, MC, LB)

Resources available:

Academic Science Staff

Design Technology Technicians

3D printers

Engineering workshop

Electrical workstations

Laser cutter

Sewing machine

Resources required:

We are looking for external support for confirmation of chosen approach and the sensors required to achieve secondary mission and the atmospheric analysis for primary mission (atmospheric temperature and pressure relationship)
We have reached out to a Dr. Karn Vohra PhD at the Birmingham university school of geography earth and environmental sciences for further support and to ensure the proper application of the scientific process is present in our methodology and have adjusted our methodology according to the feedback he has given us.
Especially useful have been articles from the Ministry of Environmental Protection of the People's Republic of China and the California Air Resources Board for information on Air quality monitoring standards and atmospheric components.
We are reasonably self-reliant for designing and prototyping. We have a wide range of skills within the team. We have support from the school's Design Technology department for the mechanical construction. We will draw on support from the school's media department for use of a drone for test drops.

2.3 Risk Analysis

Running out of time

This is a complex mission and there is a risk that we will run out of time. To mitigate against this, we have produced a Gantt chart that specifies the responsibilities of each individual and the deadline for their objectives.

Risk – Team member dropping out/falling ill

Our school offers a certain degree of flexibility in extracurricular pursuits, so it may be possible for one of our members to drop out from the competition at any time, to counteract this we have made sure that all work possible is done in a communal document and tasks that are not conducive of this have their documentation backed up on a communal OneNote page.

Rapid unscheduled disassembly of CanSat in mid-air

We do not have the facilities to fully reproduce the forces exerted on the can by a rocket launch. We have therefore tried to make the cansat as robust as possible and will carry out test drops from a drone.

Risk – Insufficient testing leading to the CanSat not performing as expected

We do not have the facilities to fully reproduce the forces exerted on the can by a rocket launch. We have therefore tried to make the cansat as robust as possible and will carry out test drops from a drone. Objective 2 is particularly challenging, and it is anticipated will require several tests to develop a robust working cansat as well as challenges integrating the large number of sensors and transmitting the data over an RF link over long distances.

CanSat not surviving landing process

There is a serious risk of severe mechanical damage occurring during test drops and rebuilding required. To prevent this we have built the CanSat as robust as possible, and have also created back-ups of most parts we are using to ensure that any damage is quickly repairable

Weather related hazards

The wiring in the CanSat may be damaged by rain or the CanSat may be blown into a dangerous area (a highway or a river), damaging it. Good weather will be required for the test drops.

Risk – Certain tasks being delegated insufficient time

Available time has been focused on the mechanical/electronic software components. There is a risk we will run out of time for building and testing a landing system. To mitigate this the deadlines in the project plan are plotted as late as possible, with those responsible for the tasks being strongly advised by the project manager to complete them as soon as physically possible, usually just under a week in advance of when the project plan places the deadline, allowing some extra breathing room in case of a last-minute emergency.

3 CANSAT DESIGN

3.1 Mechanical design (Contribution from LB)

Design of can in CAD software:

I designed the can in Onshape. I did this to ensure everything would fit and I did this so that I could then use the 3d files to 3d print the can. I chose Onshape because I was already familiar with it, and it is easy to run so I could work on the can without problems. To work out the layout of the can I first either found or created 3d models of all the components, then arranged them in an assembly. I then added holes to the component board so that I could attach all the components to the component board using M2.5 bolts.

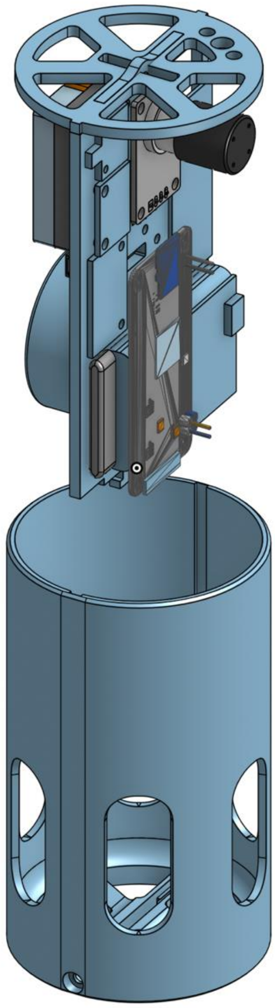


Fig 1. CAD Model

Design Decisions

I decided to make the can out of two parts, an outside shell for protection and a component board and top part that everything would be attached to. I did this because it would allow me to easily replace the outside shell without having to remove any components or connections if the shell breaks for example.

I decided to attach all the components to a middle removable component board that slides into a groove in the outside shell. This ensures that all the components are easily accessible and always in a fixed position relative to each other.

I added holes to the bottom and to the sides to allow for a lot of airflow. I decided we need a lot of airflow due to the large number of sensors (5) that need air. I decided to have a lot of holes on the bottom as the bottom doesn't need to be very strong and the bottom has the most pressure and so will result in the most airflow.

Materials and Processes

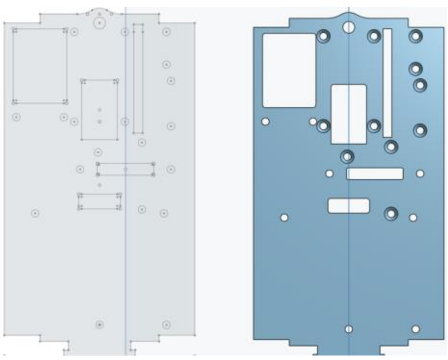


Fig 2. 2D design acrylic backplane

I 3d printed the outside shell out of PETG. I 3D printed the shell because 3D printing is very accurate and can quickly make complex designs, allowing me to implement a system to screw the shell to the component board, and allowing me to quickly iterate on the design (there have been 3 iterations for the shell). Furthermore, I already had a 3d model of the shell from 3D modelling the can.

I used PETG because it is less prone to shattering, as one of the previous iterations, which was made from PLA did, yet easier to print with than materials such as ABS.

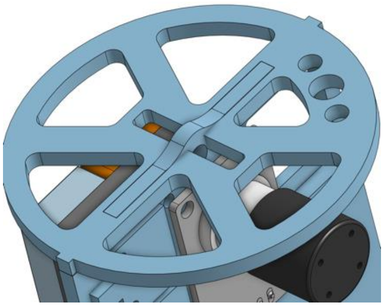
I laser cut the component board and the top out of acrylic because laser cutting is very precise, meaning I was sure all the components would fit. I used acrylic because it is very durable, meaning it won't degrade over multiple launches and it is hard, so it will hold its shape and not deform.

Iterations

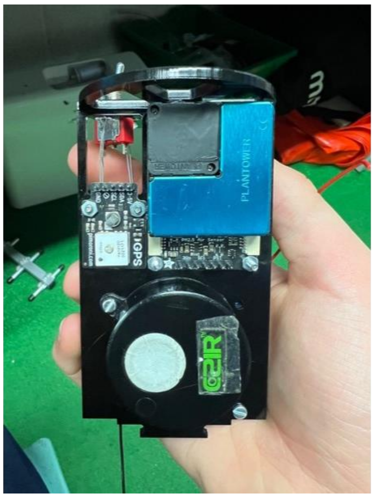
I have made 3 iterations to the design of the can.

V1 à V2

This iteration introduced airflow holes to the sides as I was unsure if enough airflow was going to reach the sensors because of the large CO_2 sensor. The component board was also altered to fit some more sensors.



Construction



Firstly, holes were chamfered with a chamfering tool and the top was cemented to the component board using tensol-12. Then components were attached using mostly M2.5 bolts and nuts. The switch and LEDs on the top were pressure fitted, which was possible because of the accuracy of the laser cutting.

Fig 3. Initial test component assembly

3.2 Electrical design (Contribution by LB/ME)

	Component Chosen	Reasons	Price £
Processor	Pi Zero	Students are familiar with pi architecture and programming in python.	9.60
Temperature Sensor	BME 680	Availability of python scripts, i2c compatibility with Pi zero. Meets requirements for 4 of the mission objectives.	18.90
Pressure Sensor	BME 680	It also measures humidity to an accuracy of ± 3% Relative humidity	
Ozone Sensor	SRAQ-G015-A	Cost and size. Requires I2C adaptor	45.54
Particulate Sensor	PMSA003I	Cost and size, very high sensitivity compared to other options. I2C compatability	42.60
Carbon dioxide Sensor	CozIR A CO2 sensor	Size and fast response speed. Requires I2C adaptor	116.50
GPS	PA1010D GPS Breakout	Cost, size and i2c integration	29.70
RF Tranceiver	Gamma 62M	Cost, availability, range, size and ease of use. Serial connectivity	15.93
Power	LP103454-Standard Type LiPo Battery	Ease of charging (USB) and power capacity for mission duration Availability, size and cost	7.19
VOC sensor	CCS811 Carbon Monoxide CO VOCs	Size, cost and I2C compatability	22.04
Total Cost			308.00

The I2C communication bus was used to receive data, as it can handle a near unlimited number of inputs connected in parallel, which supports our host of sensors required to measure air quality. The CO2 and O3 sensors did not offer I2C ports, so instead their analogue outputs were connected to an ADC which had them. ADCs were chosen over UART converters as they are relatively simple compared to the many communication protocol setups required for UART serial conversion.

More information on sensors:

The BME 680 sensor is a very versatile sensor, which can measure, Temperature, humidity, pressure and gas resistance. It has a low power consumption and is an integrated 3 in 1 sensor, which saves space and reduces the number of components we need for our mission. It is very compatible with Raspberry pi zero through the I2C communication protocol. There are also some limitations, like lack of accuracy of the pressure sensor at very high altitudes, as well as some temperature drift over time.

The SRAQ-G015-A ozone sensor can measure ozone to a broad range of concentrations. From low parts per billion to parts per million. This allows it to be used for different environments. It is compatible with Raspberry pi zero, but it will require an I2C adapter to convert analogue signal to digital signal. This sensor can be affected by temperature and humidity levels, which can affect the sensor's performance and accuracy.

The PMSA003I particulate sensor is a high accuracy sensor which measures PM2.5 and PM10 concentrations in the air. It is compatible with raspberry pi zero using either I2C connections or UART serial communication. This sensor is also vulnerable to extreme weather conditions affecting its accuracy. Calibration and good weather are essential for optimal performance.

The CozIR A CO2 sensor is also compatible with Raspberry pi zero and can use the UART serial communication. Once the GPIO pins are connected and the serial communication is enabled, you can use python to read the data. It also requires an I2C adapter. The sensor is susceptible to the calibration drift and has limited measurement range.

The CCS811 Carbon Monoxide is a gas sensor which is used to detect a wide range of volatile organic compounds in parts per billion as well as CO_2 in parts per million. It is also very compatible with raspberry pi zero as it uses i2c communication protocols. It also operates at 3.3V, which raspberry pi provides. However, this sensor has limited gas detection range and eCO2 levels are estimated between 400ppm – 8200 ppm.

Power Wiring

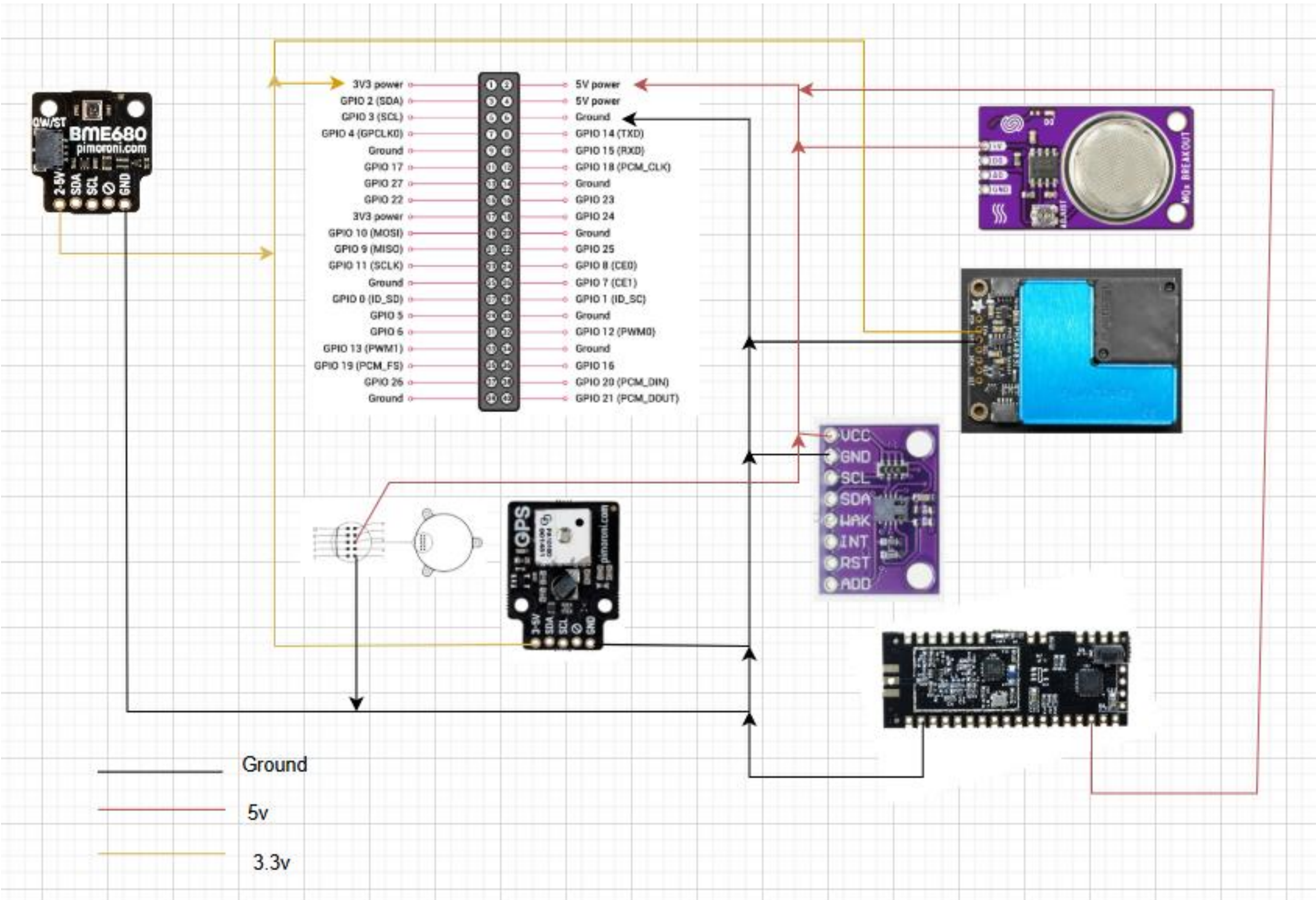


Fig 4 Cansat internal power wiring

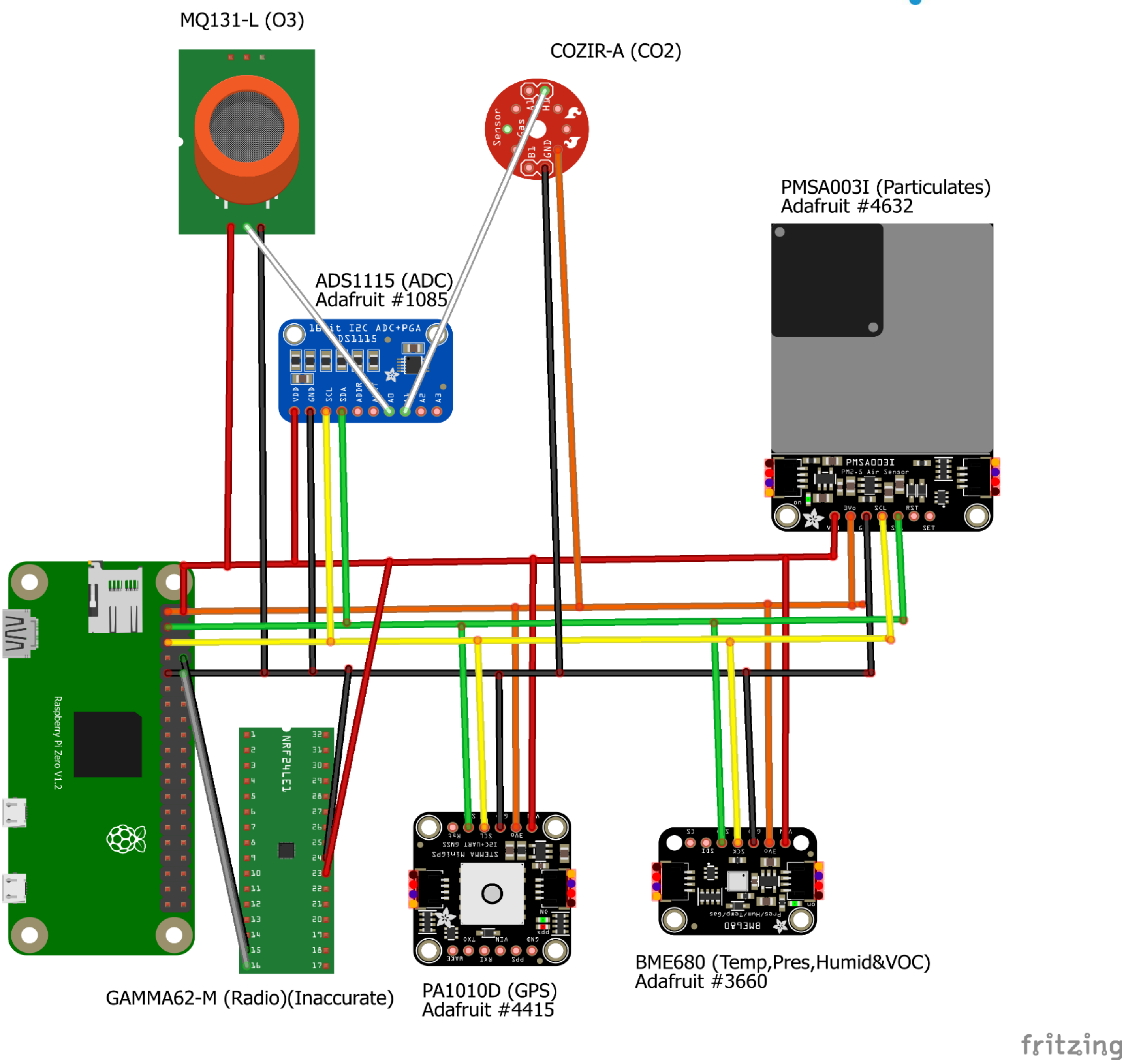


Fig 5 Cansat data wiring

3.3 Software design (Contribution from SN/EW)

Python was chosen as the programming language because it is a requirement for Computer Science A-level students, so it is readable by any members of the team who wish to contribute to the software, with the radio communication being one such example. Python also offers a great number of libraries for sensors and components, which were fully used when writing the code.

Following this, the Raspberry Pi Zero was selected as the central controller. This is because its main programming language is python. It also has the ability to handle multiple sensors through multitasking as a microcontroller. It is cheap, meaning it is easily replaceable as a component, which is critical in harsh environments such as the interior of a Cansat.

To support the many sensors required for comprehensively measuring air quality, the code for each sensor was written in separate files so that they could be run simultaneously using shell scripting. This can maximise the number of readings and increase the quantity of data. This was chosen over multithreading within a python program as it is more robust and not as needlessly complex. This also allows for each program to be fairly easy to read and debug.

The code itself was written in such a way so that it would be quickly readable and have a consistent format across files.

When interpreting the analogue signals transmitted by the two non-I2C devices, unexpected problems arose. The voltage output of the CO2 sensor rose linearly with the detected levels, but the only relation between the O3 reading and analogue output was that percentage resistance change rose logarithmically along with higher O3 levels. To model this, the change in voltage was measured as Power remained constant, so $V^2 \propto R$. Online data plotted on log axis were referenced, and using data points and A-level knowledge I modelled a function that accurately converted percentage change in voltage to O3 levels.

During the development of the transmission system, I encountered an issue where frequent OS errors occurred when sensors attempted to access the same text file simultaneously. The text file acted as a packet that was sent periodically by a separate python file to counteract a known issue of too many python processes accessing the radio transmitter causing it to halt. This was very common due to many sensors being opened at the same time and reading data at the same interval.

To resolve the issue and minimise risk, I used [FileLock(file)] to introduce a queue to opening files and replaced all instances of [packet = open(file)] with [with open(file) as packet] to remove the risk of the file being opened indefinitely and potentially wiping all data.

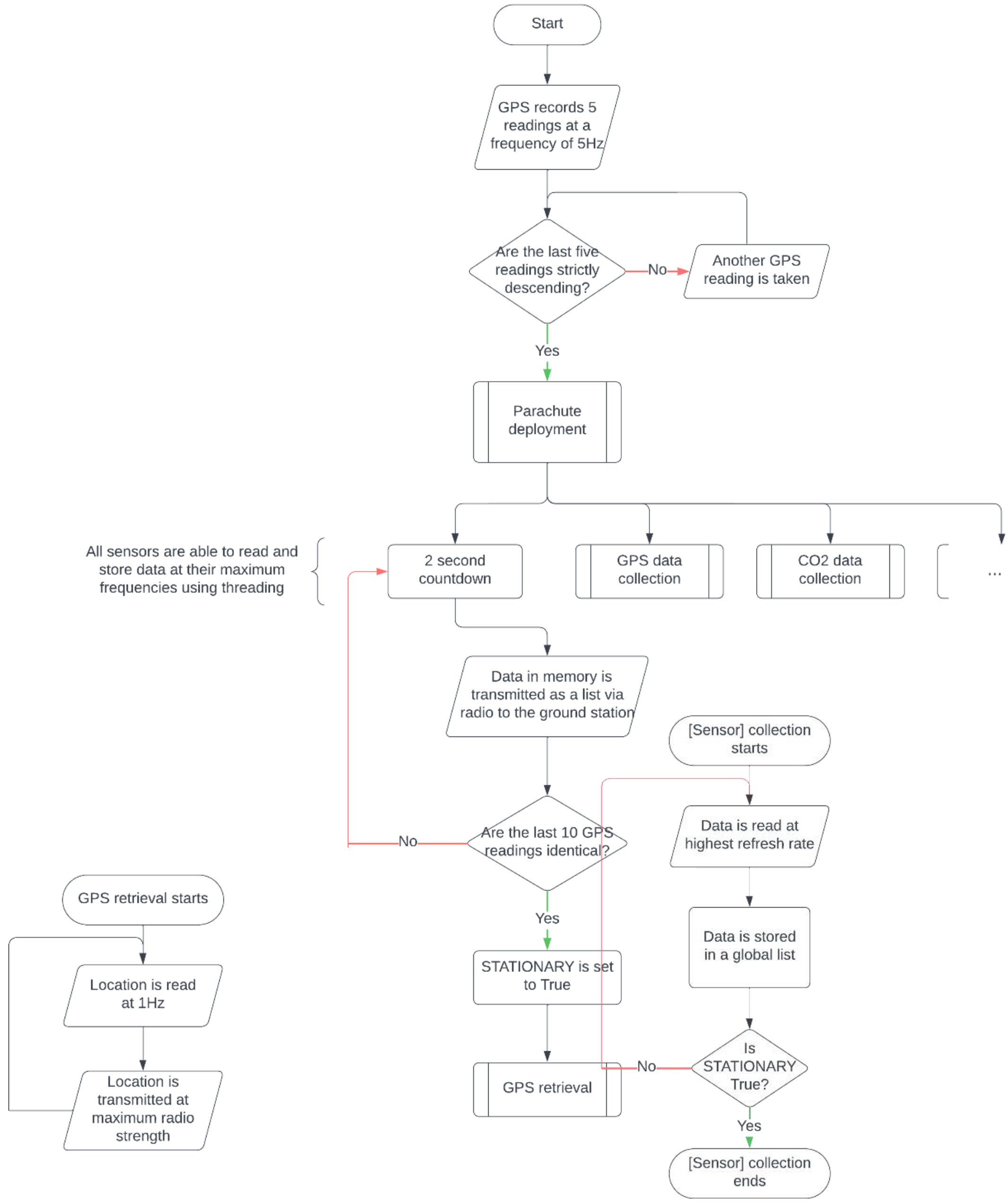


Fig 6 software design flowchart

Data presentation

Our first ideas about presenting the information is via a webpage, since it is easier to create a GUI on a webpage that allows user interaction during data presentation. The steps required are as follows:

Once the sensor readings have been transmitted back to a laptop, the code runs on the laptop processes the data and calculates the AQI. It then uploads the information onto the website. The webpage will consist of a map of the UK coloured according to the population density. We will then use a drone to fly our can around various local locations to take sample measurements, and use the information collected from the GPS to add datapoints onto the map of air quality samples. Each of the datapoints on the map should be buttons that could be clicked to display a Cartesian plane of the altitude mapped against the air quality index measured in each position. In addition, the specific measurements for the various gasses in each test will be displayed on the bottom of the page via tables and bar charts, where the measured data will be compared with the average data for particles concentration across the whole of UK. By choosing to take samples in various areas with diverging population density, I expect to find a correlation between the mean AQI in the area and the population density, thus measuring the effect of human activity on the environment.

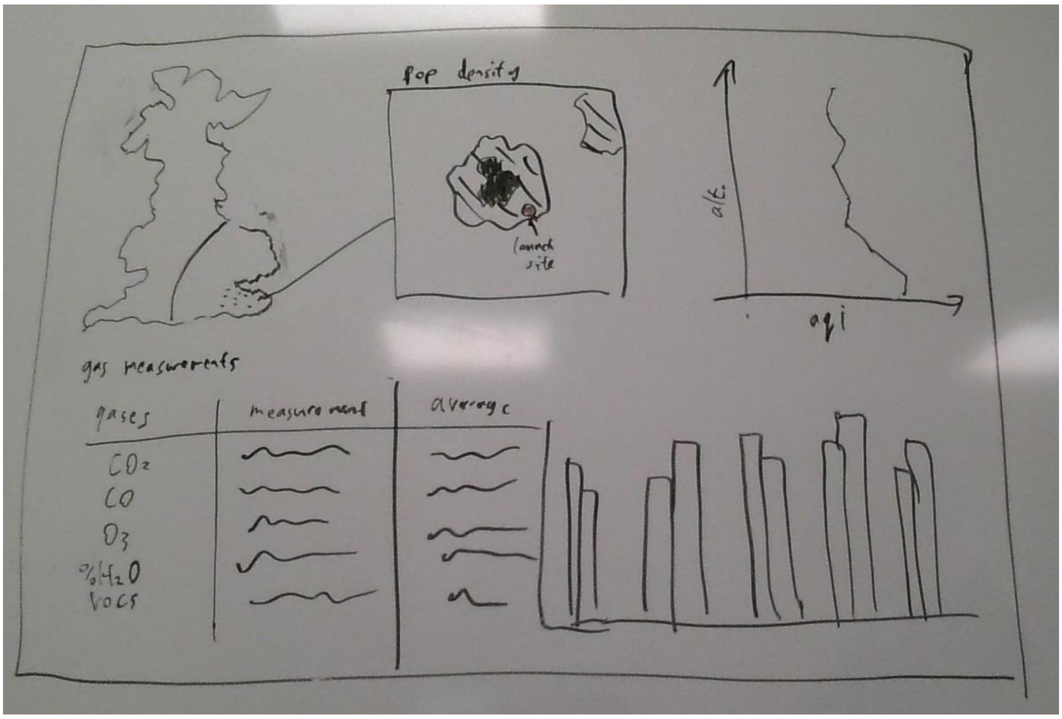


Fig 7. A sketch of the webpage we expect to produce

I choose to use a webpage for data presentation because of many reasons. Firstly, I find it easier to display maps and graphs when using a website. Secondly, the website can include buttons that allows for user interaction with the information, as well as making our data look more interesting to catch the attention of the users. Finally, this webpage should be easily accessed, making practical data on the effects of human activity on the environment available to everyone. However, using a webpage can also present certain problems. For example, one of the potential issues I acknowledged is the possible difficulty in real time automatic transmission of the data received from the device to the webpage.

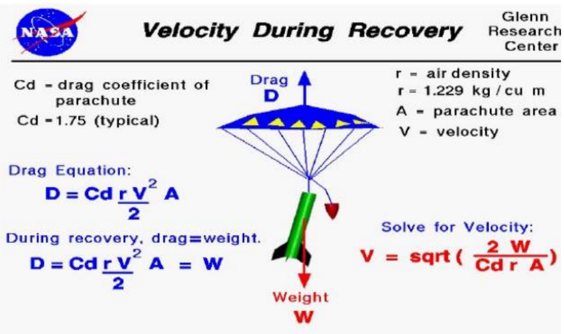
3.4 Landing and recovery system (Contribution by JZ/ME)

Requirements:

- The can is dropped from around 300 to 400 metres height.
- The decent time of the Cansat when falling from 300 metres is limited to 30 seconds.
- The descent rate must be at least 10 m/s.
- The parachute connection must be able to withstand up to 500N of force.
- The strength of the parachute must be tested, to give confidence that the system will operate nominally.

Calculation and research:

To make a parachute that allows the can to reach the ground at a speed of approximately 10m/s from a height of 300 meters, I first searched up equations relating to aerodynamics parachutes and did some rough calculations. I also found some websites that can output the ideal parachute size when the weight of the can and the time of descent are inputted.



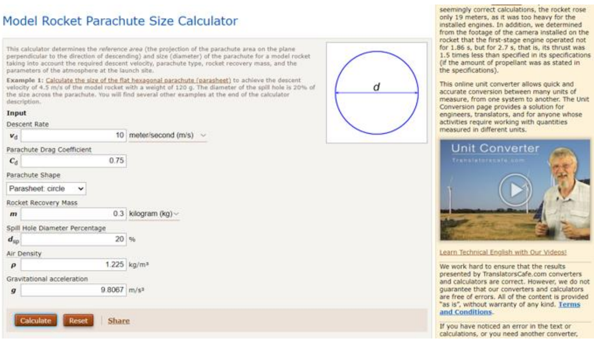


Fig 8 and 9 parachute design

I decided against a rectangular canopy because it generates less drag and may be less stable than the circular canopy, which is quite crucial to our project because a stable descent is optimal for data collection. One advantage of a rectangular canopy is it provides manoeuvrability. However, this is not necessary for our can.

The Ram-air Inflatable Parachute also provides excellent stability and drag but is too difficult to make because it involves adding layers inside the parachute to trap air chambers which requires a lot of sewing. Since our can would be descending at 10 meters per second, such large drag would not be needed.

After testing with a parachute which has only four strings attached to the canopy, I decided to increase the number of strings to improve stability. Considering how easily the strings get tangled together when there is only four of them, it may be optimal to use six strings instead of eight, as it is possibly the best solution to the trade-off between complexity and stability.

In the end, I decided to settle for a circular chute with a diameter of 64 centimetres and an apex vent in the centre with diameter of around 14 centimetres. Although multiple calculations that I have performed (as well as parachute size calculators online) have suggested that a diameter of around 30 cm for facilitate a 10m/s descent speed, I made my parachute to be larger than necessary. This means I can test the parachute and trim the edges to reduce its diameter if needed. This is helpful because it means we do not have to risk making a parachute that is too small and find ourselves needing to make another one when the launch date is close.

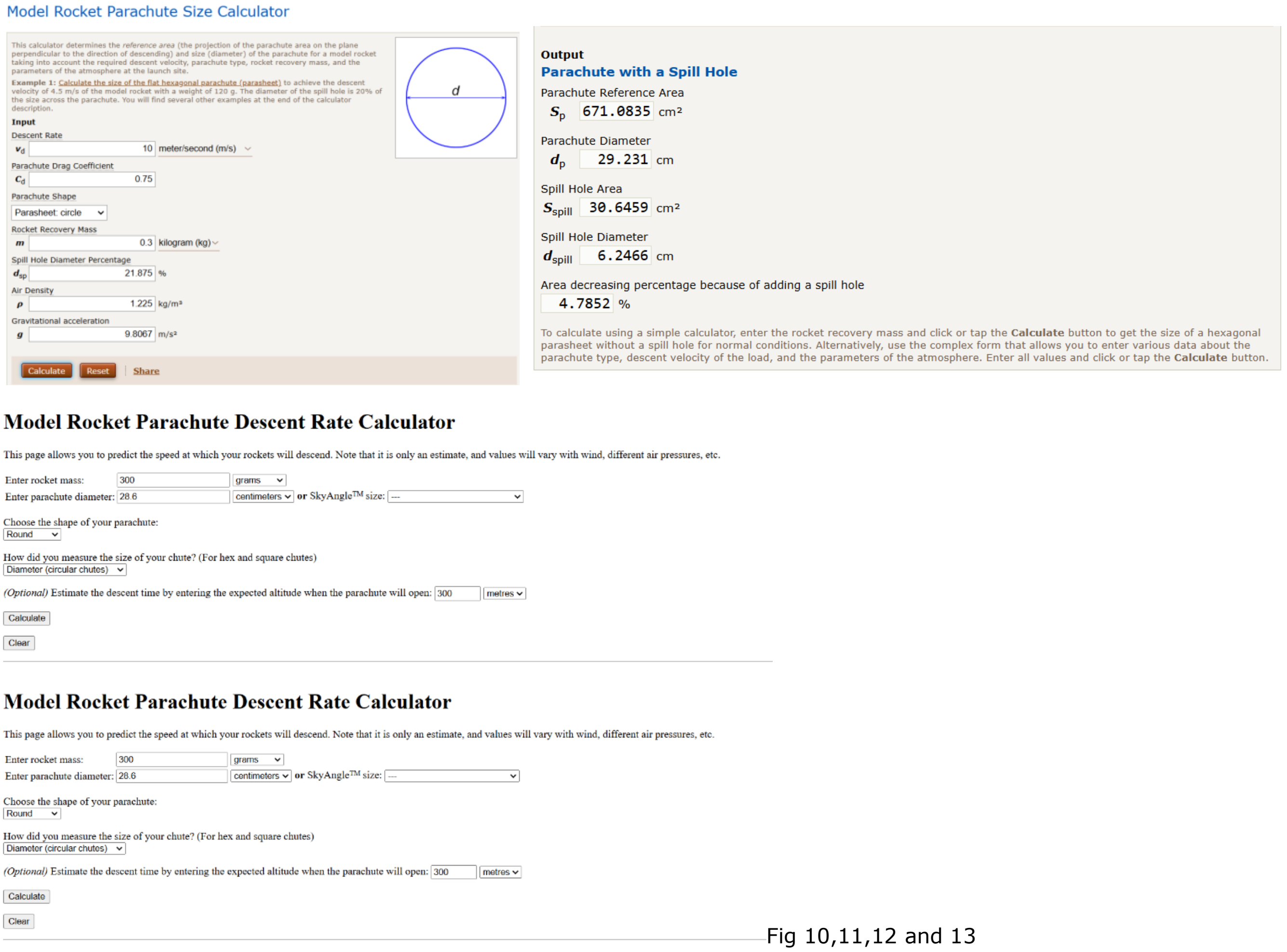


Fig 10,11,12 and 13

The apex vent is added to allow air captured by the parachute to escape instead of staying inside the parachute, eventually leading to high pressure which could cause instability. For optimisation of the functionality of the sensor modules, stability needs to be as high as possible because we are trying to detect samples such as particulates in the air, which requires stable air flow. This is supported by multiple online rocketry forums (1), YouTube videos (2), and even a report by the Illinois Junior Academy of Science (3).

Production of parachute:

Fig 14 parachute sewing

Initially, I wanted to cut out trapeziums and sew them together to form a parachute, but then I realised that stitches in the middle of the parachute may severely lower the pressure that the fabric could withstand. Therefore, I chose instead to cut out a circular piece of nylon fabric and then cut out the apex vent. To strengthen the fringe where I will attach the strings, I folded over 2 centimetres of fabric and stitched it three times. I then hole-punch the fringe of the fabric and attach small rings on the holes to help improve the

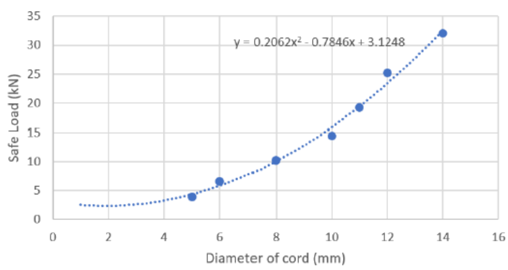
After putting together the parachute I have designed, I made sure to test the strength and functionality of the parachute.

Firstly, to ensure the connection between the parachute and the can itself is sturdy enough to withstand an acceleration of 10-20G (4), I attached the parachute to the can using a small metal ring. This means that if the cords and the points of connection are strong enough, the parachute would be able to withstand the force. To assess this, I found two tables comparing the diameters of nylon strings and its maximum breaking force and extrapolated them (after taking the mean for the values) on excel. The results are shown below. By using x=60, we can predict the safe load to be 1074 newtons, which is well over the 196 newtons that the cord would experience during launch.

Diameter	Diameter	Tensile Strength	Working Load
inches	mm	pounds	pounds
1/4"	6	1,700	340
3/8"	10	4,100	820
1/2"	13	7,300	1,460
5/8"	16	14,600	2,920
3/4"	19	19,000	3,800
7/8"	22	28,100	5,620
1-0"	25	33,200	6,640
1-1/4"	32	52,000	10,400
1-1/2"	38	74,000	14,800
1-5/8"	41	92,100	18,420
1-3/4"	45	109,500	21,900
2-0"	51	129,700	25,940
2-1/4"	57	179,400	35,880
2-1/2"	64	196,000	39,200

Rope Diameter		Minimum Breaking Strength		Safe Load (Safety Factor 12)	
(in)	(mm)	(lb_f)	(kN)	(lb_f)	(kN)
3/16	5	880	3.91	73.3	0.326
1/4	6	1486	6.61	124	0.551
5/16	8	2295	10.2	191	0.851
3/8	10	3240	14.4	270	1.20
7/16	11	4320	19.2	360	1.60
1/2	12	5670	25.2	473	2.10
9/16	14	7200	32.0	600	2.67
5/8	16	8910	39.6	743	3.30
3/4	18	12780	56.8	1070	4.76
7/8	22	17280	76.9	1440	6.41
1	24	22230	98.9	1850	8.23
1 1/16	26	25200	112	2100	9.34
1 1/8	28	28260	126	2360	10.5
1 1/4	30	34830	155	2900	12.9
1 3/8	32	38250	170	3190	14.2
1 1/2	36	48600	216	4050	18.0
1 5/8	40	57375	255	4780	21.3
1 3/4	44	66150	294	5510	24.5
2	48	84600	376	7050	31.4

Figs 15, 16 and 17



The mechanics lead, Lev, and I then pulled the parachute and can in opposite directions with considerable force, resulting in the point of connection on the can to fracture. This helped mitigate risks of mechanical damage to the can in future tests because the structure has been strengthened.

Fig (14) Electron microscopic image of structure of nylon ropes.

Fig (15) The colour of parachute is chosen to be orange to allow for maximum contrast with the sky.

Secondly, I climbed to a height of ten meters and dropped the parachute with a 300g wooden block attached to simulate the weight of the can. We measured a descent time of 2.98 seconds, allowing us to estimate a total descent time (over 300 meters) to be roughly 30.6 seconds, which is close to the requirements of the competition. However, I am aware that this test drop is done inside a building, whilst on the date of launch, various environmental factors could disrupt the descent. Therefore, I plan to test it again when my team travel to more remote areas to use a drone to test the can.

References:

(1): https://www.rocketryforum.com/threads/parachute-spill-holes.20043/

(2): https://www.youtube.com/watch?v=xdEmV6w_xpc - “Why This Tiny Hole Makes Your Rocket Parachute 10x Better”

(3): https://nar.org/member-only-reports/rd/NARAM-61/N61-Hole-y-Chutes-Zurek.pdf

(4):CansatGuidelines:https://assets.ctfassets.net/pc40tpn1u6ef/6Ztxr599ixdHx7lZYpSv7u/d0439e8b0d3f9baa15f8a0540e6e7718/UK_CanSat_Competition_guidelines_2024-25_Final_v1.0.pdf

Fig (5): https://www.translatorscafe.com/unit-converter/ja/calculator/parachute-size/

Fig (6): https://descentratecalculator.onlinetesting.net/

Fig (7,8): https://www.translatorscafe.com/unit-converter/en-US/calculator/parachute-size/

Fig (9,10): https://descentratecalculator.onlinetesting.net/

Fig (11): https://www.engineeringtoolbox.com/nylon-rope-strength-d_1513.html

Fig (12): https://denverrope.com/rope-strength-guide/

Fig(14):Uconn health, central electron microscopy facility: https://health.uconn.edu/central-electron-microscope-facility/

3.5 Ground support equipment (Contribution from LE)

For the ground equipment I will need to design a radio receiver that will be able to pick up sensor information from the CANSAT probe, the maximum distance of which to me will be around a kilometre and a half. To be safe my target is to design one that can detect from 2km. The receiver itself if very weak at actually receiving signal, hence I will need to build an antenna for it. My best option was the yagi uda antenna. It is simple to build and offers good gain compared to its counterparts. Furthermore, the fact that it is directional helps avoid interference therefore improving data transmission consistency. I chose the frequency at which we will operate on to be 868Mhz as you don't require a licence to use it and to avoid interference from the more commonly used 433Mhz.

Antenna design:

To begin I needed a piece of software that could help me model it. I chose CST studio as it is free and features elements to help me optimize my design.

Next, I needed to plan certain aspects of my antenna, for instance boom length and number of elements. I opted for 8 directors as it vastly improves gain, and 5 angled reflectors, which also improve gain, front to back ratio and consequently decrease interference.

Fig 18 – table displaying every variable I tweaked to optimise design

Parameter List

Name	Expression	Value	Description
A	= 46.98	46.98	
D	= 17.1	17.1	
F	= 1.3	1.3	
H	= 200	200	
I	= 0	0	
J	= 5	5	
L	= 40	40	
LG	= 120	120	
O	= 20	20	
P	= 0	0	
Q	= 0.11	0.11	
R	= 5	5	
Rr	= 0.11	0.11	
S	= 0.34	0.34	
T	= -5	-5	
W	= 345.38301	345.38301	
X	= 127	127	
<new parameter>			

Here is the meaning of each variable:

- A – Vertical distance between midline and secondary reflectors
- D – Horizontal distance between Primary reflector and secondary reflector
- F – Temporary variable for boom width (used to simulate how boom will affect gain)

- H – Temporary variable for boom length
- I – Experimental variable that affects horizontal distance between primary and secondary directors (I tried to make curved reflector array instead of straight)
- J – vertical offset of entire boom
- L – length of screws
- LG – I tried to make radius of reflectors different from directors
- O - Variable to try and experiment to see if the gain will be improved by adding an exponential increase to the length of directors
- P – Variable to try and experiment to see if the gain will be improved by adding an exponential increase to the distances between directors
- Q – The ratio of first director spacing from the dipole compared to the wavelength of the operating frequency
- R – radius of elements
- Rr – ratio of reflector to dipole length compared to the wavelength
- S – ratio of distances between directors compared to the wavelength
- T – offset variable for something
- W – the wavelength of 868MHz or λ
- X – director length

1. I modelled the design into CST using approximate ratios I found online. For example, the dipole needing to be 0.5 λ to 0.55 λ.

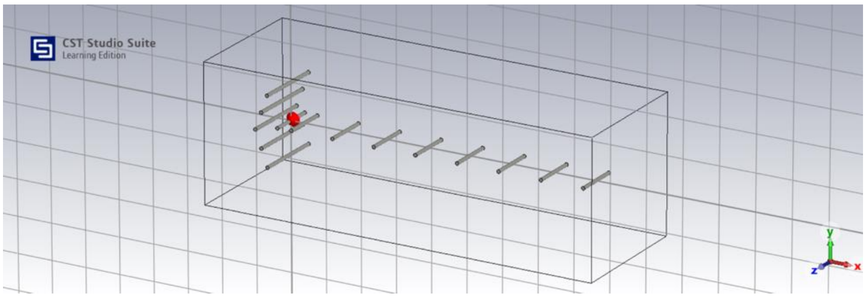


Fig 19 – antenna design modelled with only antenna elements

2. To optimise the design, I had to find the best lengths and distances between my components. To optimise the reflectors, I experimented with the values and observed what effect they had on the radiation pattern. A large lobe facing the opposite way from where the antenna is pointing indicates a bad design. Optimising the lengths of the directors and distances between almost the same, the difference being the focus was on getting a larger frontal lobe. The first director also needed to be a different distance apart than the rest, which was approximately 0.2 λ rather than the 0.45 λ Antenna elements modelled in CST

3. Furthermore, I modelled all conductive items like screws that could cause interference in my antenna to make sure that the simulation was as accurate as possible.

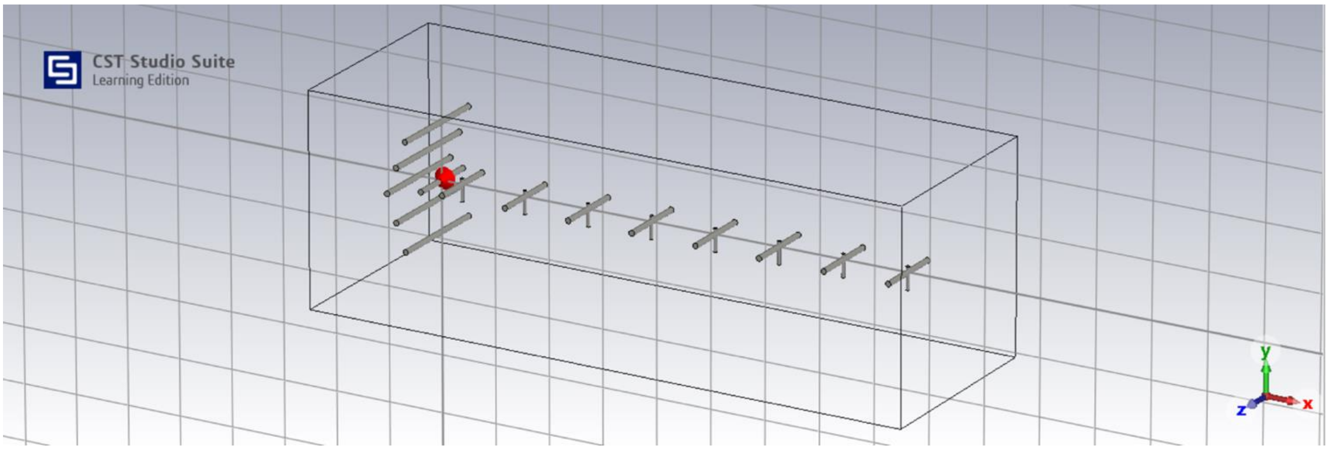
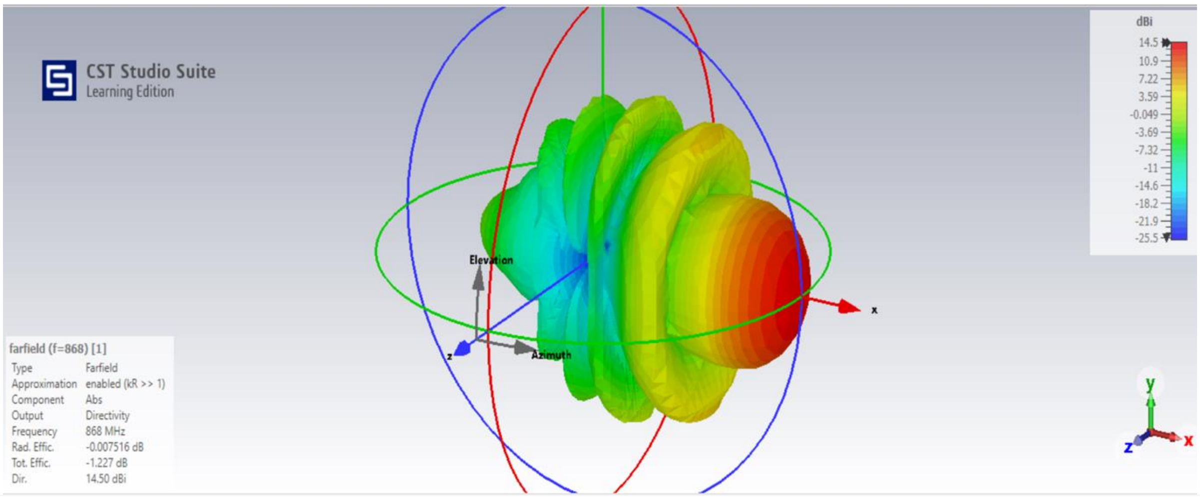


Fig 20 – antenna modelled with screws

4. To simulate, I set up the solver which CST includes and got the radiation pattern, S and A parameters, and the frequency vs gain graph.

Fig 21 – the radiation pattern of the antenna



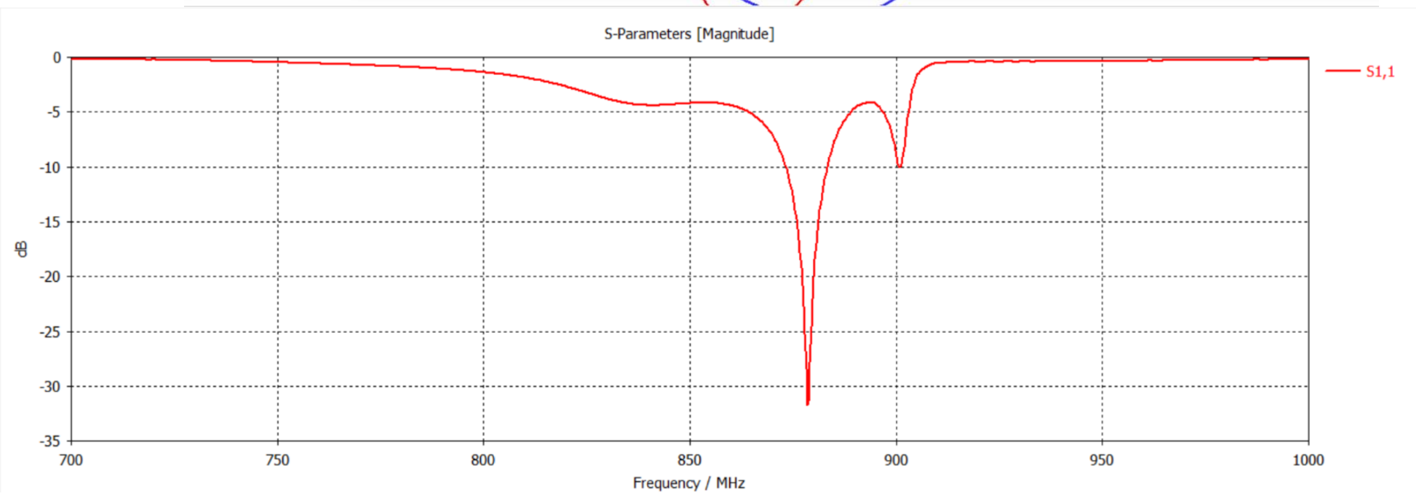


Fig 22 - S-parameters (sudden dip at 868Mhz indicates quite optimal design)

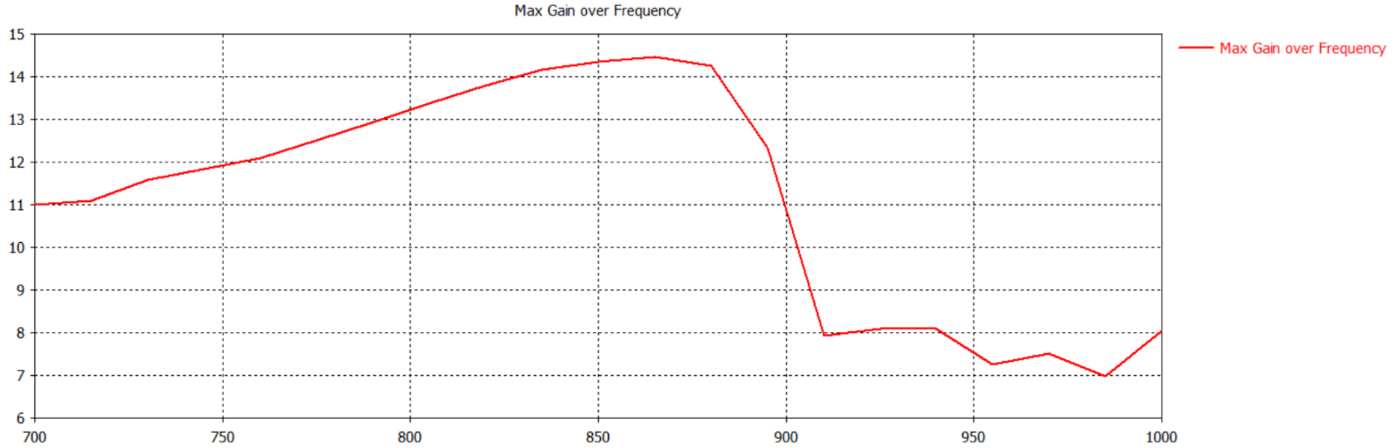


Fig 23 - Gain over frequency graph (21 results) shows optimal frequency is around 868

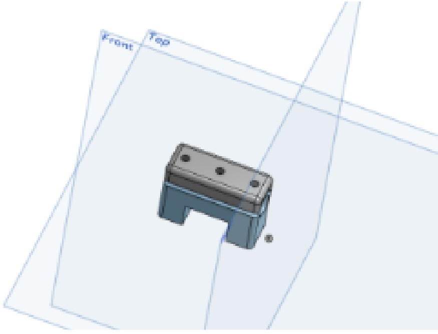
Building – I build the antenna in the school's DT workshop. I made 3d printed brackets to hold the directors so that they can be screwed on. This allows any component to be replaced easily.

Testing – I ran 3 tests to see if the antenna works. I used last year's CANSAT to transmit temperature and humidity for me to pick up. I will be using the PUTTY software to interpret the data and save it as a CSV file so that I could instantly open it in excel.

1. Test 1 – the first test was 5m away from the CANSAT in an enclosed area. This was to see if the wiring was functional
2. Test 2 – I had a volunteer walk with the CANSAT from school to the nearby car park (300m) and see if the signal fades. It didn't.
3. Test 3 – This was the 2km test. I went to a railway bridge on the other side of town and asked a friend to turn on the CANSAT by phone, getting a consistent signal. I also tested how much I could angle the antenna and still get a signal and got a very comfortable range.

Fig 24. To make it easier to hold, I also added a grip and a gun like stock.

Fig 25. Brackets for mounting elements



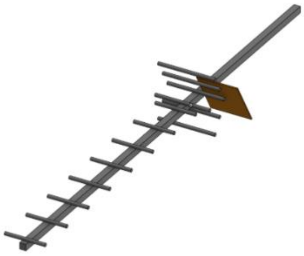


Fig 26 CAD drawing of Yagi excluding brackets

To receive the antenna signal and transmit it to the computer I need to translate it to serial data.

My goal for the comms system is for it to be handheld while still allowing a full range of motion. To achieve this, I planned to replace the PUTTY software on my laptop with a program running on a raspberry pi that would be attached to the antenna. This necessitates internal antenna power and a control panel to turn it on/off. Furthermore, having my own code running on a raspberry pi3 would allow me to customise it freely and tailor it to achieve my goals of measuring distance between the CANSAT and my antenna using signal delay.

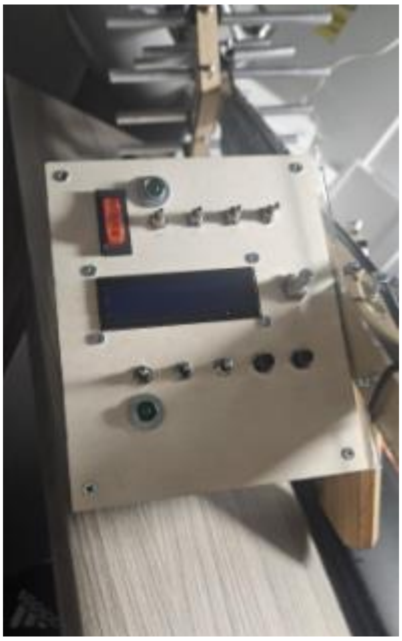
Initial electronics design:

- A box containing the raspberry pi, gyroscope, battery and receiver.
- An interface box that has switches to help me operate the antenna.
- I sketched a rough design of the electronics using 2D design. Despite my eccentric choice of software, the final product greatly helped me organize the wiring

A quirk of the design is the use of two chips to charge and discharge the battery. This is because the first chip supports charging via USB-C which is very convenient however does not feature output contacts. The second chip has both input and output contacts, hence why it is used. I’ve also encountered severe heating issues with the second chip when it is used to directly charge the battery.

As a proof of concept, I made temporary plywood boxes. The face of the control panel was designed to have 5 switches (main power, pi3 power, gyro power, lcd power and transmitter power) which would later allow me to individually test components. One LED shows main power status and the other mirrors the LED on the transmitter which allows me to monitor if data is being sent/received. 3 switches and two buttons are used for GPIO for the program. Finally, an LCD screen and a potentiometer display information while transmitting. I also included a 3-amp fuse to protect the circuits from a potential short.

Figs 27,28 and 29. Finished control panel and battery box



3.6 Testing

Date	Type of test	System being tested	Status

Week 4	Bench test of sensors	Test primary mission sensors	Tested working OK
Week 5	Component fit	Test all components for fitting into the can	Final assembly built and components trial fitted OK
Week 6	Bench test of RF coms	RF comms	Tested working OK
Week4	Sensors	UV	Tested working OK
		Ultrasonic	Tested working OK
		BME680	Tested working OK
		Geiger	Tested working OK
Week 8	Software	Test integrated systems functionality	Still outstanding
Week 8	Landing systems	Parachute and attachments	Still outstanding
	Range check RF Power capacity	RF comms Battery	Still outstanding Still outstanding

3.7 Overall testing for launch

1	Physical structure	Check for damage/loose components
2	Power	All systems show activity (led status)
3	Transmission	Data received at base station
4	Parachute	all attachments secure and strings not tangled

4 Outreach programme (Contribution from MC)

Fig 30. Cansat presentation to year 9 physics class

For our outreach program we have conducted presentations to classes of younger students within our school in order to both spread awareness about the issues we are seeking to tackle and to introduce them to the Cansat competition format in case they may take an interest themselves. We have had the delight to hear from many of those we presented to that they themselves had a passion for all things engineering related, and plan to enter the competition themselves when they come of a suitable age.

We have also arranged a field day where we visit a local primary school and, after summarizing the concept of Cansat and what we will be doing for our own entry, we will construct some paper rockets with the students and launch them with compressed air along with some similar activities to get them thinking about engineering challenges and demystify the field early in life, hopefully inspiring some future aeronautical engineers.

We will also make a promotional video of the competition at the regional launch at Westcott. This will be shared to the school via the school's internal website and also externally on the school's website pages.

5 LAUNCH DAY PREPARATION*

5.1 Launch checklist/countdown

5.2 Post mission checklist

		Dur'n	Action
1	Check for physical damage	5 mins	
2	Check still transmitting data	5 mins	
3	Check data received and saved on ground station laptop	5 mins	
4	Turn off power on cansat	1 min	

5.3 Launch Day risk log

	Risk	Mitigating measures	Check completed
1	Landing system fails to deploy correctly	Check for damaged parachute cording, tangles in the parachute cording, tears on the canopy	
2	CanSat undergoes rapid unscheduled disassembly	Check CanSat for obvious structural faults. Carry spare parts at all times	
3	Radio systems fail to transmit live information during descent	Check the antenna is properly receiving data from CanSat before launch	
4	Weather blows the CanSat into a danger area	Check that the breeze has died down at T-Liftoff	

5.4 Results analysis procedure

For the primary mission the atmospheric pressure and temperature will be saved on the ground station laptop as a csv file and used to plot a graph in Excel.

For the secondary mission we will transmit the ??????, also for later analysis on the ground station laptop.

6 LESSONS LEARNED*

It

*Only required for final report