WIND DRIFT TEST VEHICLE

Atmospheric Data Collection Device

2022 - JASPER CUSIEL 374-147816868-93601

TABLE OF CONTENTS

ABOUT THE PROJECT

This project is about the development of a data-collecting device for Rocket Lab. Rocket Lab is a public American aerospace manufacturer and launch service provider, with a New Zealand subsidiary. The company operates lightweight Electron orbital rockets, which provide dedicated launches for small satellites.

Inside Rocket Lab

The device I built is a wind drift test vehicle for collecting accurate data on atmospheric conditions that the rocket parachute is subjected to during testing. This would allow further development and tuning of the parachute that is used for the electron booster re-entry phase of recovery.

HOW DID I COME ACROSS THE PROBLEM?

While exploring avenues for projects for this scholarship, I decided to contact Rocket Lab as I have an interest in aerospace engineering. They proposed two issues they had discussed with their engineers but hadn't had time to look into or develop a solution for.

The first of which, was a cheap and simple remote weather station that can log or transmit conditions at a test site to provide a record of the wind speed, direction, air pressure and temperature. This is something that they often miss or rely on forecast data only when performing a test. The device would also be useful in recovery operations as it provides up-to-date information on the conditions at the landing site.

The second option was to build a wind-drift test vehicle that could be dropped when performing a test of a parachute system. It would record its position during descent to provide a record of wind speed with direction at altitude during the test. It would also log and broadcast its position to allow it to be recovered and the data to be downloaded after the test.

After careful consideration of the two options, I settled on setting out to develop a solution to the wind drift vehicle option. I chose this option because I felt that it would be more complex than the remote weather station project. Thus the project would be much more challenging allowing me to explore various facets of engineering.

IMPACT OF THE PROJECT

This project would be important to Rocket Lab as it would allow them to collect data surrounding the conditions on the day of testing their various systems. By essentially building a scale model of the full size rocket stage, the collected data would be a scale representation of the real thing. Thus data can be collected with a smaller, low cost, and accurate device rather than testing with the much more expensive full sized rocket stage. This gives Rocket Lab an accurate record of the conditions on the day of testing allowing them to accurately interpret testing data i.e. whether specific environmental factors impacted the collected results.

BRIEF AND SPECIFICATIONS

My key stakeholder for this project is Matt Darley who is a recovery systems manager for Rocket Lab. However, since he is based in Auckland, most consultation would be done via email. After replying to Matt Darley indicating I was keen to pursue the wind drift test vehicle issue, he replied with a list of requirements and nice-to-haves.

Our requirements for the wind drift vehicle are as follows:

- Target descent rate 10 m/s.
  - Ideally tunable by adding or removing ballast mass – this could be as simple as a water bottle that we add or remove water from, so able to achieve 8 to 12 m/s, or 5 – 15 m/s if possible.
- Deployable from a helicopter – dropped out of a door or belly hatch.
  - Likely aircraft Bell 429, Airbus EC-130, Sikosky S-92
  - Possibly static-line deploy the parachutes to ensure it opens, but may be possible to just throw it
- Target mass <5kg, but some flexibility if tunable ballast.
  - Suggest <10 kg as limit for easy handling.
- Deployment from altitudes between 1000 ft and 10,000 ft.
- Inflation within 100-200 ft to minimise altitude lost when dropped.
- No glide, so that test vehicle only drifts with the wind.
- Record GPS altitude and lat/long time history to allow calculation of wind profile.

Nice-to-haves:

- Record pressure and temperature (possibly humidity) to allow calculation of air density.
- Transmit / telemetry of altitude & location would be useful to allow instant calculation of wind profile.
  - Enables parachute test to happen before wind drift vehicle is retrieved.
  - Also aids in finding and retrieving the vehicle.

INITIAL RESEARCH INTO EXISTING SOLUTIONS

From here I began looking into the types of solutions that already existed or could be adapted to fit Rocket Lab's requirements.

Initially, I began to look into solutions in the model rocketry space as the model rockets had many of the same characteristics as the vehicle I would be building. I found that it was very common for model rockets to include the following; a flight computer for control of sensors and parachute deployment, an altimeter for collecting altitude readings and potentially a GPS to track the model rocket's flight as well as to recover the model rocket. Many of these components were simple, cheap, off-the-shelf components so thus would be readily available. It seemed that I could use very similar or equivalent hardware for tracking the descent of my wind drift test vehicle.

Essentially the main difference between a model rocket and my device is the fact that the vehicle would be dropped at an altitude rather than being rocket-propelled to that altitude. Therefore it would seem fairly straightforward to adapt the concepts used in model rocketry to this specific application.

From here I began by breaking down the list of requirements Matt had given me into its core components. The list began as follows:

Microcontroller
- The miniature computer to handle collecting data from the various sensors while writing that data to a storage medium as well as transmitting that data to the ground station.

Communication
- The device to transmit data back and forth between the wind drift test vehicle and the ground station a device on the ground to receive data from the vehicle)

Position
- A GPS module to collect position information of the vehicle.

Pressure, Temp and Humidity
- An altimeter combined with a temperature and humidity sensor, these could be separate devices also.

Power
- A method of powering the device, most likely a battery.

Main Body
- A tube of sorts to house the data logging equipment (cardboard or fibreglass tubes are commonly used in model rocketry)

Parachute
- A device to slow the descent of the vehicle.

Electronics Compartment
- A space to house all of the sensitive electronics.

Ballast Bay
- A space to house mass such that it could be added or removed to increase or decrease the total overall mass of the vehicle.

CONTEXT CONSIDERATIONS

Since a lot of the parachute system testing by Rocket Lab is done over water, a key characteristic of my wind drift test vehicle is the fact that it will need to be 100% waterproof. This is so that the electronics do not get damaged due to water ingress as well as the fact that if water is ingressed, it may cause the test vehicle to sink. Therefore I will need to ensure that the final wind drift test vehicle is waterproof and this will need to be tested.

Another environmental factor is the fact that the ground station could potentially be located multiple kilometres from the ground station so a strong wireless connection to receive GPS coordinates is key to not losing the vehicle.

PROJECT COST

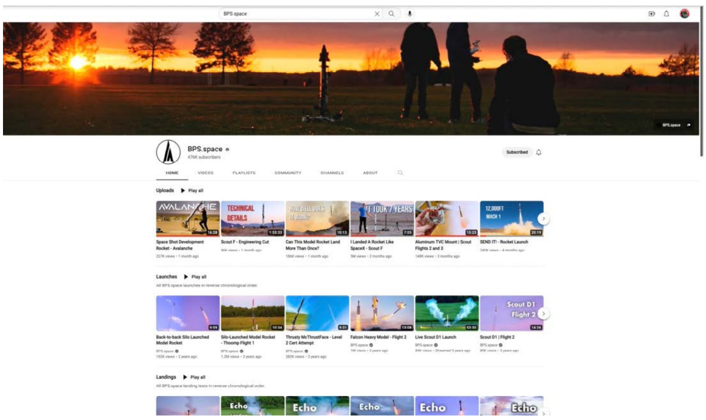
Even though the wind drift test vehicle will be low cost compared to that of the full-size test vehicles, my solution will still cost a little due to the nature of prototyping. Therefore it is key that I carefully consider the material and components I use to build the device such that they are fit for purpose in order to keep the overall cost of the project to a minimum.

USABILITY

Since the wind drift test vehicle will be dropped out of a helicopter, the device may be handled by multiple personnel. For this reason, I will need to ensure the device is small enough to be handled easily, rugged enough to withstand general handling without damage to critical parts and simple to operate to ensure data is collected and not lost. Therefore this should ensure that the data can be collected the first time without having to re-drop the vehicle, thus reducing the associated operational costs since operating a helicopter is quite expensive.

ELECTRONICS RESEARCH

The first aspect of the wind drift test vehicle I would investigate is the electronics I would use as it would make up a large component of the project.
I began by scouring model rocketry forums and watching several hour-long videos from the channel BPS Space on youtube. From there I collated a handful of parts that I thought would be suitable to use based on those used in the Youtube videos I watched.

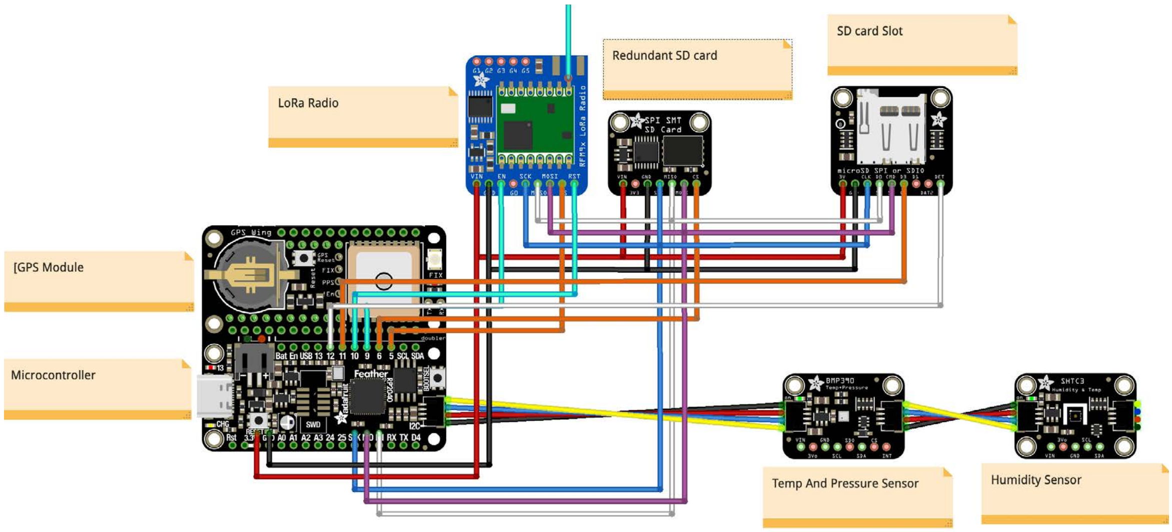


BPS Space's Youtube Channel

A side note; I have quite a bit of knowledge in this kind of maker/electronics space having many personal projects as well as being interested in making things from a very young age. This knowledge combined with knowledge from being a member of various robotics clubs as well as working for a local technology company played a large role in informing many decisions. This is due to having worked with many similar products and general knowledge simply through being interested in the maker community.

Most of the parts I settled on were from the very popular Adafruit Industries who produce a lot of DIY-based products as well being high quality for a reasonable price. I planned to use their Feather development boards due to their modularity, ease of use and simplicity - various modules can be stacked on top to quickly add or remove sensors and functions to a project.

WIRING MOCKUP

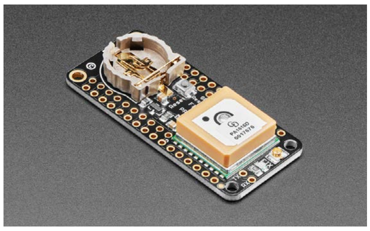


Inital wiring diagram

Above is a wiring diagram that I created in Fritzing, a wiring diagram creation tool, to help me plan how each sensor would interface with the main microcontroller board.

GPS MODULE

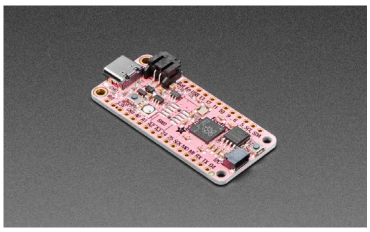
The GPS module I chose was the Adafruit Ultimate GPS feathering with the MTK3339 chip which boasted a positional accuracy down to 1.8m while consuming very little power while operating. The MTK3339 chip was the only option offered by Adafruit, however, it came in a few different board formats - a standalone breakout board, feather wing or Raspberry Pi hat. Since I wanted to stick to the feather ecosystem for simplicity's sake I chose the feather wing version of the Ultimate GPS.



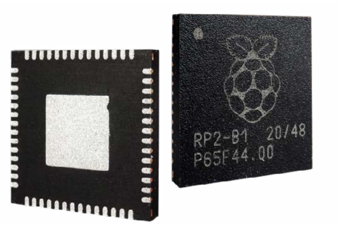
Adafruit Ultimate GPS Feather Wing

MICROPROCESSOR

This would plug into the microprocessor. For this I chose the Adafruit Feather RP2040 as the microcontroller chip on the board (the RP2040 from the Raspberry Pi Foundation) was one that I had previous experience with from a digital technology project last year. For this reason, I knew that it would be more than powerful enough to process, store and transmit the data I would be collecting.



Adafruit RP2040 Feather



Raspberry Pi Foundation's RP2040 Chip

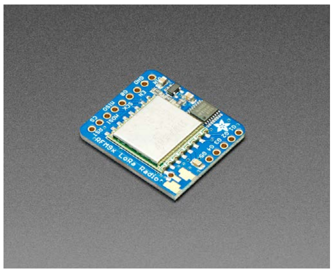
COMMUNICATION / TELEMETRY

The method I settled on to transmit the collected data to the ground station was a low-frequency radio. Many model rockets used the XBee Pro sub gigahertz radios for stable radio communication as far as 5km. However, after looking into the details of how the radio system worked, I stumbled upon the fact that it operated on the same frequency as LoRaWAN. A Bit of Googling confirmed that it was essentially a 900Mhz LoRaWAN radio.

XBee PRO Radio Module

“The LoRaWAN® specification is a Low Power, Wide Area (LPWA) networking protocol designed to wirelessly connect battery operated ‘things’ to the internet in regional, national or global networks, and targets key Internet of Things (IoT) requirements such as bi-directional communication, end-to-end security, mobility and localization services.”

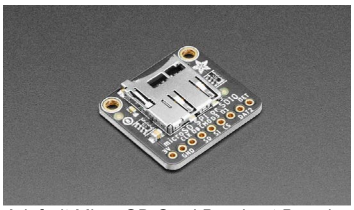
Since Adafruit sold LoRa radios that operated on 900Mhz (the Oceania frequency reserved for free public use), I could very easily add a feather-wing LoRa board into my design. This also helped to cut costs as the Adafruit LoRa board was roughly half the price of the Xbee radios. Thus saving money in certain areas meant that it could be spent on other aspects of the project.

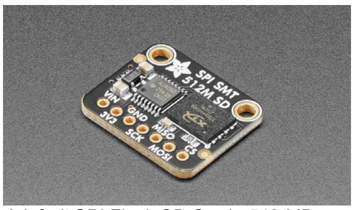

Adafruit RFM95-W LoRa Breakout Board

DATA STORAGE

I planned to use two micro SD cards onboard the test vehicle to improve the redundancy of the device. This would mean that the recorded data would be copied to both SD cards such that if one was broken, the other would hopefully be intact such that the data would not be lost.

For this reason, I decided to use a removable micro SD card and an SPI flash card. The SPI flash card is essentially the memory module (similar to that found in a micro sd card) soldered to a breakout PCB. Therefore since the memory chip is soldered to the PCB, there is no chance of the memory card being accidentally removed. Vibrations during the vehicle’s descent could easily cause the micro SD card to come loose potentially corrupting the collected data and the data would not be collected after it comes loose. Therefore a copy of the data on the solid-state flash chip ensures redundancy to prevent data loss. This redundancy aspect and ensuring all parts are fit for purpose is a key consideration in this project as it is relatively expensive to use the device due to it needing to be dropped from a helicopter as well as having high stakes to collect the required data.

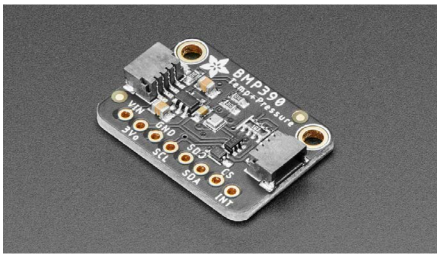

Adafruit Micro SD Card Breakout Board



Adafruit SPI Flash SD Card - 512 MB

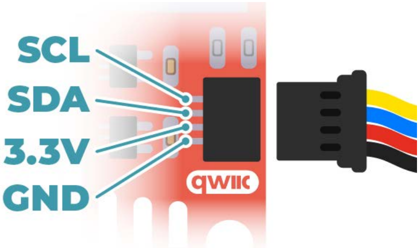
ALTIMETER

The altimeter I chose for this project is the Bosch BMP390 on the Adafruit StemmaQT breakout board. This is a precision barometric pressure and altimeter combo chip that also can read air temperature. The barometer and altimeter function of the chip is essentially measured using a precision pressure sensor.



Adafruit BMP390 Barometric Pressure and Altimeter

However, the term barometric pressure sensor relates to the sensor being fixed and measuring the subtle change in atmospheric pressure at a given place compared to the altimeter function which relates the pressure to a given altitude. I chose the BMP390 from bosch as it boasts accuracy down to ±1m. It also uses the StemmaQt / Qwiic system (a standardised four-pin JST connector to allow sensors to be quickly and easily connected to development boards) which would make assembly very straightforward.



SparkFun Qwiic Connector

Adafurit also describes it:

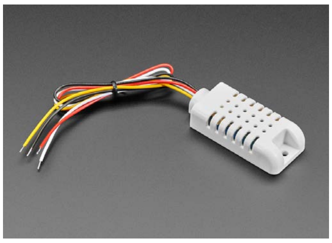
"similar to its earlier versions but even better… excellent for environmental sensing or as a precision altimeter…intend this sensor to be used for drones and quadcopters"

---

which seems to fit the bill for my application.

TEMPERATURE AND HUMIDITY SENSOR

The temp and humidity sensor was about the most difficult issue to find a solution to. The main factor that made a suitable sensor hard to find was the fact that it had to be waterproof. However, since the humidity sensor measures the amount of moisture in the air, if the sensor were to get wet at all, it would read 100% humidity. Therefore a waterproof humidity sensor is quite uncommon.

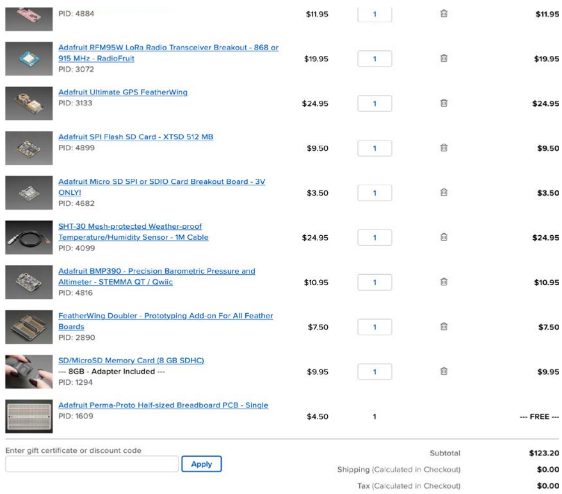

SHT-30 Non Waterproof Version

Luckily Adafruit sold the SHT30 which was waterproof and had a special sintered metal cover to prevent the sensor from getting wet but allowing air to pass through to allow accurate humidity readings while not submerged. Adafruit describes the product as being suitable for projects such as outdoor weather stations where the station could get wet. Unfortunately, since this was a bit of a niche product as their other options to keep outdoor weather station equipment dry such as a Stevenson screen (not suitable for my application as it allows water ingress if submerged), the price was around double that of the same sensor on a breakout board.


Adafruit SHT-30 Temperature/Humidity Sensor

FINALISING THE PART LIST

With this list of components, I went onto Adafruit's website to buy parts as they don't have a significant markup like some education stores here in NZ that sell the same parts. However, while pricing up the components I chose, the total cost for just the electronics would be roughly $215 just for vehicle electronics. I felt this would be far too expensive as there were still many more parts to buy.



PID: 4884	$11.95	1	$11.95
Adafruit RFM95W LoRa Radio Transceiver Breakout - 868 or 915 MHz - RadioFruit PID: 3072	$19.95	1	$19.95
Adafruit Ultimate GPS FeatherWing PID: 3133	$24.95	1	$24.95
Adafruit SPI Flash SD Card - XTSD 512 MB PID: 4899	$9.50	1	$9.50
Adafruit Micro SD SPI or SDIO Card Breakout Board - 3V ONLY! PID: 4682	$3.50	1	$3.50
SHT-30 Mesh-protected Weather-proof Temperature/Humidity Sensor - 1M Cable PID: 4099	$24.95	1	$24.95
Adafruit BMP390 - Precision Barometric Pressure and Altimeter - STEMMA QT / Qwiic PID: 4816	$10.95	1	$10.95
FeatherWing Doubler - Prototyping Add-on For All Feather Boards PID: 2890	$7.50	1	$7.50
SD/MicroSD Memory Card (8 GB SDHC) --- 8GB - Adapter Included --- PID: 1294	$9.95	1	$9.95
Adafruit Perma-Proto Half-sized Breadboard PCB - Single PID: 1609	$4.50	1	--- FREE ---

Enter gift certificate or discount code
Apply

Subtotal	$123.20
Shipping (Calculated in Checkout)	$0.00
Tax (Calculated in Checkout)	$0.00

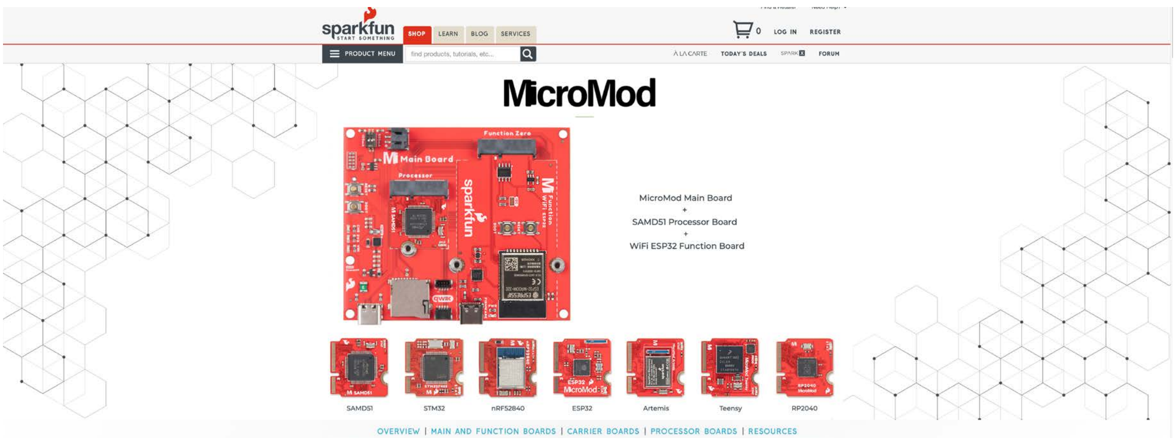
Adafruit Shopping Cart

PIVIOTING TO AN ALTERNATIVE

This prompted me to start to look into other equivalent parts, maybe looking outside of Adafruit's feather ecosystem and into others or even into single-component breakout boards that I could combine.

While researching I came across Sparkfun's micro mod system - a solderless, modular interface ecosystem that uses the M.2 standard to mix and match your choice of a processor with specific Function Boards or stand-alone Carrier Boards. Essentially this system allowed you to choose the main microcontroller, and add it to a breakout board that would interface with other modular sensor boards allowing you to prototype without soldering.

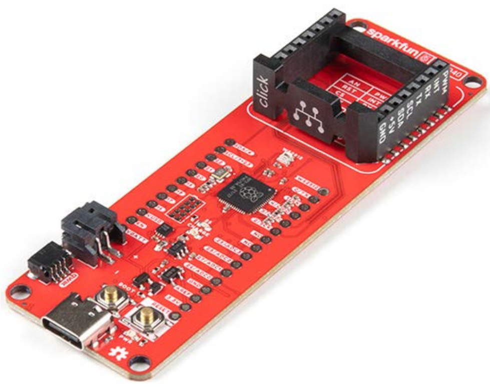
While this was looking promising and the individual boards were cheap, after replicating the Adafruit-based option with equivalent parts the total cost would come out to be very similar to the Adafruit-based system. So the process began again and I went researching to find a cheaper equivalent.



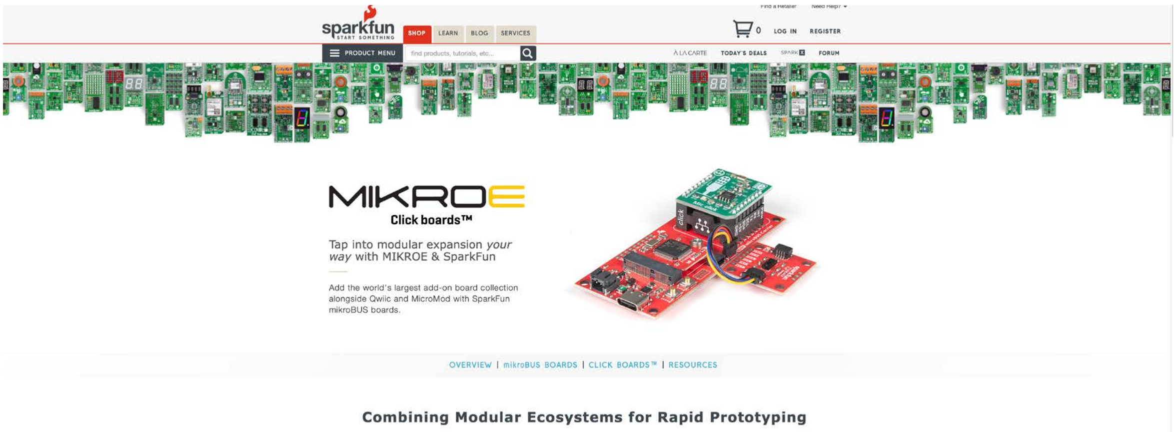
Sparkfun's About MicroMod Webpage

MIKROE CLICK BOARDS

That's when I found Sparkfun had another alternative - the Mikroe click boards. This was an RP2040-based board that has the Qwiic connector allowing me to connect the temp and humidity sensor and altimeter. It also has the Mikroe bus socket which allows you to simply connect any of their over 1200 click boards via a standardised socket. Since Mikroe Electronica, who produces the click boards, makes so many boards, the cost of the boards is quite cheap. Therefore combining the Sparkfun Mikroebus board and one of their GPS boards, the cost of the electronics came down to around $200.



SparkFun RP2040 MikroBUS Board



SparkFun's About MIKROE Click Boards Webpage

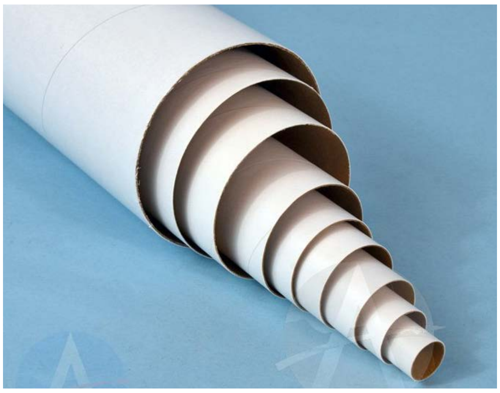
The GPS click board I chose was the GNSS 7 click board based around the Ublox Neo M9N (the successor to a long history of high-performance GPS modules from the reputable Ublox company), the board boasted metre level accuracy and excellent RF mitigation. The RF interference (Radio Frequency) mitigation was quite an important feature as the GPS module would be close to the LoRa Radio so interference from the radio would not be desirable as it could alter the reported GPS location.

Mikroe Electronicka's GNSS 7 Click Board

A wee side note is that the GPS module uses the GNSS system (Global Navigation Satellite System), an international multi-constellation of satellites including the GPS system whereas GPS specifically refers to the North American Global Positioning System. For the sake of simplicity and more common use of the term GPS, I will use the GPS term to refer to the GNSS 7 click board even though this is not technically correct.

INITIAL IDEAS FOR THE BODY OF THE DEVICE

Having settled on a more cost-effective electronics solution, I moved over to brainstorming ideas for the enclosure that would house the electronics while meeting Rocket Labs' requirements of being able to add/remove mass from the device. The vision that had been forming in my head while researching model rocketry was to simply put the electronics into a long tube, much like the main body tube of a model rocket. However, this tube would be of a much greater diameter thus allowing it to be shorter making it easier to handle and manoeuvre from the side of a helicopter.



Cardboard Body Tubes From Apogee Rockets

Normally a model rocket body tube would be made of a thick cardboard or fibreglass tube. Since cardboard disintegrates in water and fibreglass tubes are extremely difficult to work with without the correct tools, these materials were not suitable. Having built a few drones in the past, I considered the use of a large-diameter carbon fibre tube. It was both strong, and light, I had the tools to work with the material and it was available from places overseas such as Aliexpress for cheap. The downside was that shipping could be up to a month but since it was not a critical part to get started building the vehicle, it should theoretically show up in time for me to put all of the parts together into the tube.

Large Diameter Carbon Tube

However, after consulting my Rocket Lab, Matt advised me that carbon fibre was not RF transparent meaning that radio waves were strongly attenuated i.e. wouldn't easily pass through the carbon tube. "*A word of caution about carbon fibre – it is a great structural material, but is also opaque to RF, so any antennas will need to be mounted externally to ensure a signal. Plastic or GRP might be a simpler option*."

Large Diameter PVC Pressure Pipe

This would mean any antennas would need to be mounted externally. It also meant that the antenna would have to be in the line of sight to the ground station which would work but would mean if the tube was oriented such that it blocked the line of sight, the connection to the ground station would be lost. This was not ideal so Matt proposed the use of PVC pipe or GRP (Glass reinforced pipe) as an alternative as both materials were RF transparent. I decided to go with PVC pipe as it was easier to find at the local plumbing shop than GRP which seemed to be more of a specialty product.

CHARGING THE DEVICE

Another factor to consider was how I would be able to charge the onboard battery as well as a method to get the recorded data from the internal micro SD card. Since the micro SD card will be buried deep inside the electronics stack, it would require disassembly to access the micro SD card which would be very inconvenient.

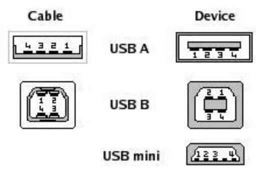
I had worked with XLR connectors at work so was familiar with how to build the cabling and receptacle socket. The XLR connector is commonly used in audio equipment and thus is readily available. A quick search around suggested that there were waterproof versions of the jack available which would tick the waterproof aspect of the project.

Neutrik XLR Plug

Neutrik XLR Jack

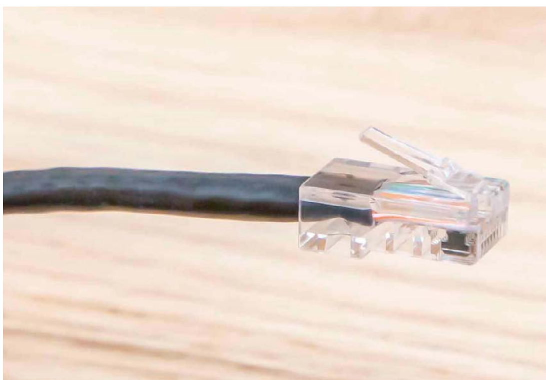
XLR Plug Sealing Cover

This seemed like a feasible option however these XLR connectors only came with up to four pins, making them suitable to carry up to a USB 2.0 signal (+5v, Ground, Data TX and Data RX). However since the microcontroller and GPS board both have a USB connection, two XLR connectors would be required. This was the main reason I switched to another solution. The XLR jacks would occupy roughly half of the total space leaving very little space for the other various sensors and antennas.



Pin	Signal	Color	Description
1	VCC	■ (red)	+5V
2	D-	□ (white)	Data -
3	D+	■ (green)	Data +
4	GND	■ (black)	Ground

USB 2.0 Pinout and Wire Colour Codes



RJ45 Connector - Commonly Used For Ethernet

Another type of cable I had made extensively at work was the ethernet cable or more specifically the RJ45 connector. These were very common and could be found at the local hardware store. The RJ45 connector is also used in a lot of cinema and broadcast solutions so ruggedised waterproof jacks aren't hard to come by. While researching, I found several other more proprietary options which used specialised connectors. I shied away from this idea as a lot of them were cheaper than the RJ45 connector but if I needed to make more cables than I had connectors, I would need to re-order from online stores which would significantly delay my project.

The model I settled on was the Seal Jack RJ45 Coupler from Belfuse as it met the waterproofness requirements, had 8 pins to facilitate the two USB connections and could easily be accessed via a toolless bayonet lid. Steward connectors' description of its application also seemed to describe a use case very similar to mine.

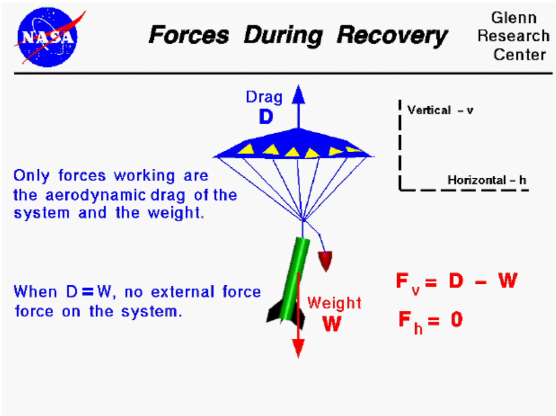
"Reliability and functionality are a necessity when the need for Ethernet begins to move outside the typical office environment and under the Internet of Things umbrella. Moisture and other environmental factors limit the capabilities of electronics in these environments thus requiring a sealed yet reliable cable assembly and jack combination that protects against such factors."

Belfuse Seal Jack CAT 5E RJ45 Coupler

PARACHUTE

The wind-drift test vehicle would need to target a decent rate of somewhere between 5ms to 10ms. This means that an appropriate-sized parachute would be required to slow the descent of the vehicle.

The parachute essentially opposes the force of gravity acting on the mass of the vehicle, thus if the forces are balanced, i.e. act in equal and opposite directions, the vertical velocity of the vehicle will be constant. Due to the size and mass of the vehicle, research on model rocket forums suggested that a model rocket parachute would be in the neighbourhood of the correct size for my application. Many large-scale model rockets seemed to be around 3 to 5kg. Since model rockets should return to the ground intact, the appropriately sized parachute gives them a decent rate of approximately 5 - 10ms.



Parachute Force Diagram

Simple Nylon Parachute

Apolo Style Model Rocket Parachute

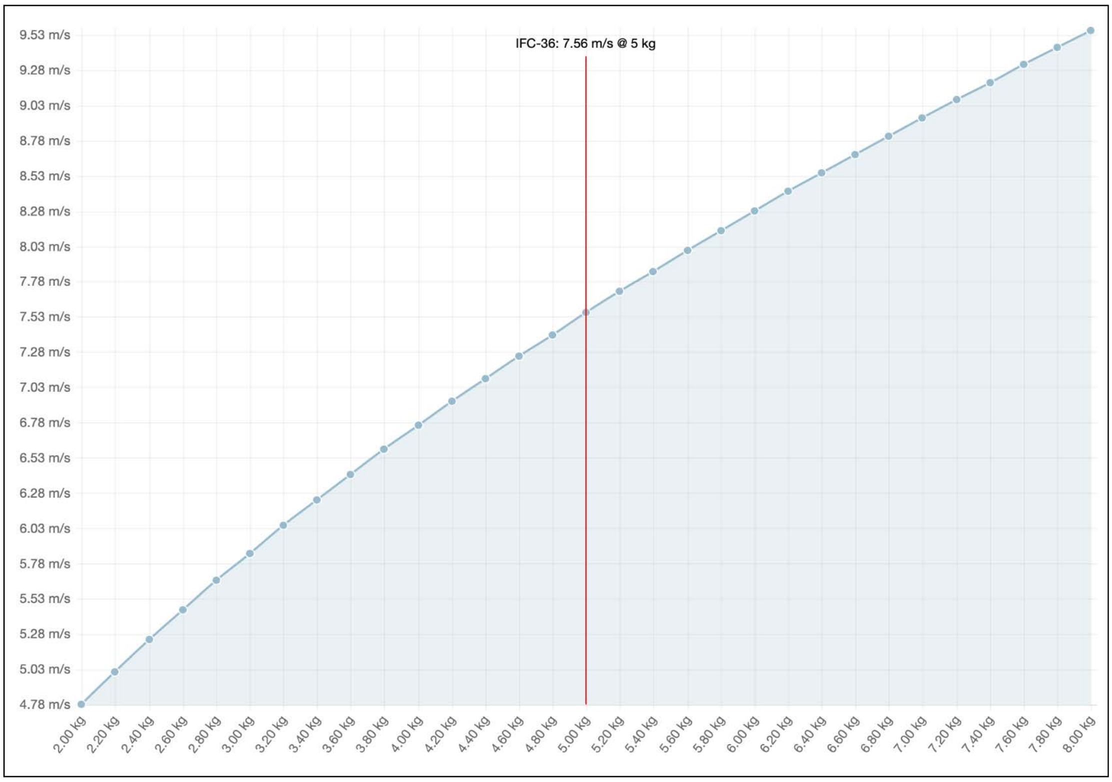
Eliptical Parachute

FRUITY CHUTES

While searching these model rocketry forums, a parachute manufacturer that kept being mentioned was Fruity Chutes. Looking on their website I came across a tool on their website called the parachute descent rate calculator. This tool would allow you to select two parachutes that they sell, set the rocket target mass and calculate the theoretical descent rate of the rocket.

This was especially helpful as it also helped me to determine the mass range of my vehicle that would produce the vertical descent speed in the 5 - 15 ms range.

Descent Rate vs Weight

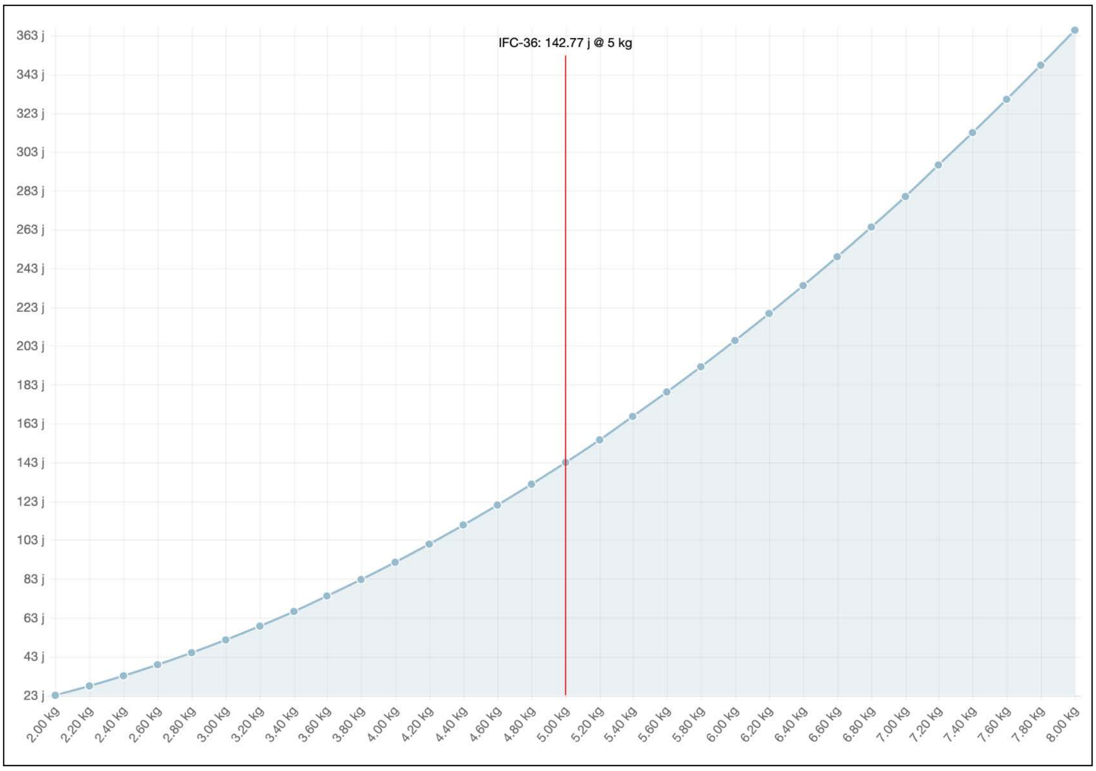


Descent Rate vs Weight for a 36 inch Parachute With a CD of 2.2

This meant that my vehicle would need to be in a mass range of such that the target speed set by my stakeholder could be achieved. This meant that I would need a parachute of approximately 36 inches in diameter with a CD of 1.5 - 2.2. The term “CD” refers to the coefficient of drag of the parachute, essentially the higher the value, the more drag the parachute produces and thus a slower resulting vertical descent rate.

This means you can either use a smaller parachute with a higher CD or a larger parachute with a lower CD to achieve the same vertical descent rate. From research, the main reason for choosing the smaller parachute with a high CD is for weight and space-saving on smaller model rockets without compromising on the descent rate. For my application, the aforementioned factors are not a concern as this is more important where total mass can affect things such as maximum rocket-powered altitude.

Impact Energy Joules vs Weight



Impact Energy vs Weight for a 36 inch Parachute With a CD of 2.2

Handily, the tool also provided the expected impact energy of the mass hitting the ground. This would be useful for making sure the bottom of the vehicle is strong enough to withstand impact.

While these parachutes were reasonably priced at around $200 for a good quality parachute, once shipping from the United States was included, the total cost skyrocketed to around $500 shipped. At this point, I was considering seeing if the local upholstery shop would be able to stitch me a parachute. However, this was quite a critical system whereby if it were to fail, I could lose the whole vehicle.

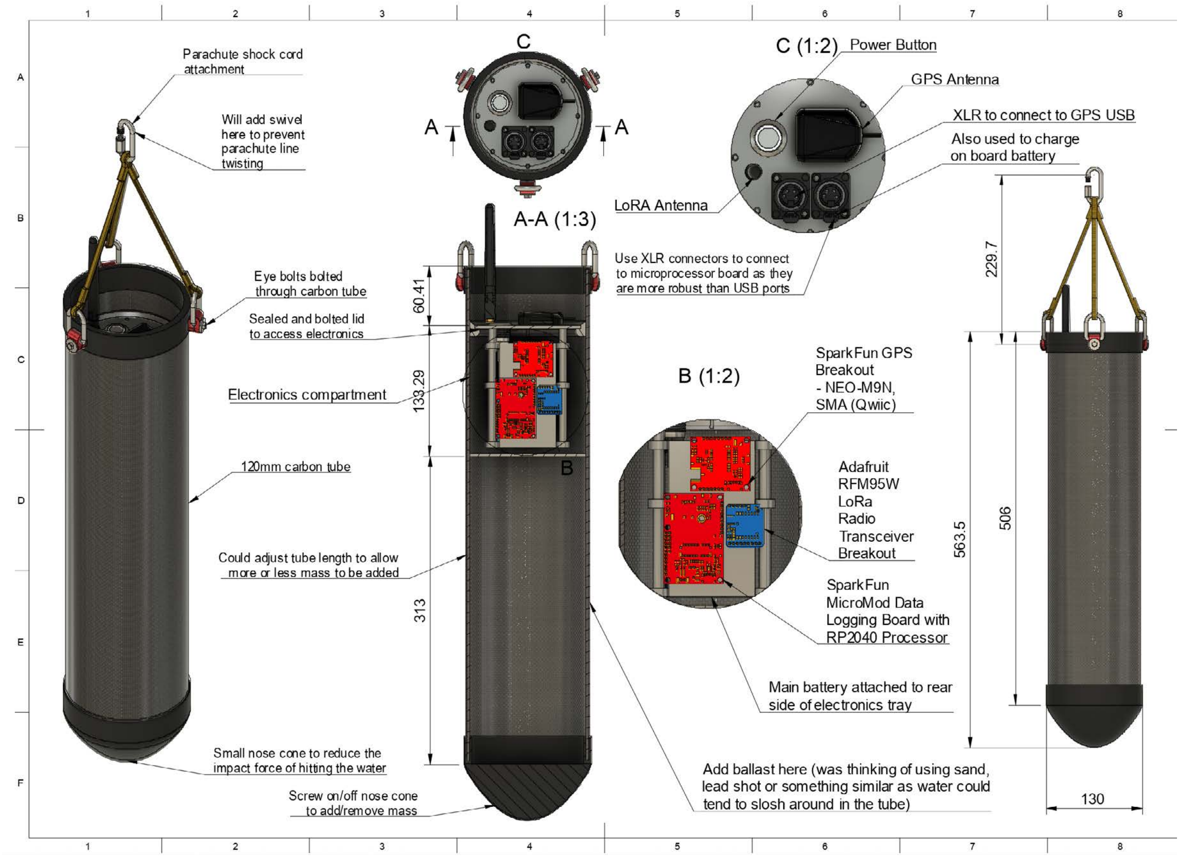
I ended up contacting Rocket Lab about this issue and they mentioned they actually might have some spare parachutes laying around. Since I am no expert in model rocketry or parachute selection, I decided to let Rocket Lab pick something suitable as they are experts in this field.

CAD MOCKUP

Having seemingly sorted a solution to most of the glaring issues I could think of without physically building a prototype.

From here I moved over to Fusion 360, a very popular, industry-grade computer-aided design software I had self-taught myself. Below is the quick mockup of the design I had in mind.

Since I would be sending this back to my stakeholder I made sure to label all of the parts and the intentions I had.



First Mockup of Design in CAD

A side note; the change to the single RJ45 connector came after I had made the drawing. I was keen to just get a rough mockup back to my stakeholder ASAP.

FINAL OUTLINE OF COMPONENTS

With this email, I wrote out a description of each component and my intentions for how it would all go together:

“Attached below is a quick drawing I made outlining some of the key attributes of the concept.

Microprocessor:

- I am planning to use the SparkFun MicroMod Data Logging Carrier Board with the RP2040 processor since it allows the gps module and any other required sensors to be quickly attached via the Qwiic connector. It also has a sd card slot to record log files and a connector to attach a battery.

Communication:

- I am planning to use the Adafruit RFM95W LoRa Radio for telemetry as LoRA operates on the public 915MHz frequency. The modules are advertised to do a few miles line of sight with very basic wire antennas and suggest in their product videos that with proper antennas they could do much further.
- This would connect to a base station which would in turn be connected to a laptop to visualise and calculate wind profiles. At this stage I think I would start with building the vehicle first then look into setting up the telemetry system a little later on.

GPS Module

- I am planning to use the SparkFun GPS Breakout - NEO-M9N, SMA (Qwiic) as it provided horizontal accuracy down to ~1.5m, supported all of the GPS constellations and an update rate of up to 25Hz. The module also has “Excellent RF interference mitigation” which would be good since the LoRA radio will be near the GPS module.
- The vertical accuracy is not listed in the data sheet but a bit of googling suggests it is usually about 3 times greater than the horizontal accuracy so might supplement the altitude readings with a barometric pressure sensor for a bit more accuracy.

Pressure, Temp and Humidity:

- I was planning on using the bosch BMP390 barometric pressure sensor with a vent tube with a few bends and fine mesh on the inlet to mitigate water ingress as there are few waterproof pressure sensors that I could find.
- I was looking at the SHT30 Mesh protected Temp and Humidity sensor from Adafruit which would work and could be mounted on the lid as well. This seems to be about the only option for weather proof humidity sensors as weatherproof temp sensors are fairly common.

Power:

- I was not too sure on the exact battery capacity to use as the GPS module consumes 36mA while tracking, the LoRa module consumes ~120mA at its highest power setting while transmitting and the other sensors would consume very little power. Thus if I budgeted for 250mAh a 2500mAh battery should last ~10 hours. I presume this should be suitable but could definitely use a higher capacity battery if required.

- I was planning to use a waterproof XLR connector to connect to the main microcontroller as well as the GPS module to upload code/charge the onboard battery as they are a bit more robust than usb c cables. Main Body:
- I was thinking of using a carbon tube here but could definitely use a PVC tube for prototypes to keep costs down. I was thinking of 3D printing (as I have one at home) the inner frame pieces, lids and compartments out of some strong filament.
- I presume the bottom would need some sort of nose cone to reduce the impact force of the vehicle hitting the water. However I would assume it would not want to penetrate too deep into the water so the nose cone in the drawing is mainly just for illustration purposes.

Parachute:

- I had a look around for model rocket parachutes and found a few suitable ones, however they were very expensive when factoring the cost of shipping from the US. Dad suggested that you guys might have some old drogue shoots that may be about the correct size? Otherwise if I could find a pattern, the local upholstery shop might be able to stitch me something. The parachute would be attached via three eye bolts through the carbon tube so that all force is through the tube and not the top lid so that it isn't ripped off.

Electronics Lid:

- The waterproof LoRa and GPS antennas would be attached to the lid of the electronics bay. The lid would be sunk into the tube such that none of the antennas can get caught and pulled off by the parachute lines. I will probably use a smaller LoRa antenna than the one in the picture but will have to shop around a bit for a suitable one. Could also mount the SPOT trace here or inside if it doesn't need an unobstructed view of the sky.
- A push button power switch is depicted in the drawing, however maybe some sort of switch that could be 'locked' in the on position with a screw would be better so electronics are harder to be accidentally powered off. I thought about using a key switch but was thinking that the key could be easily lost.

Ballast Bay:

- Mass would be added/removed via a screw on the nose cone to make it quick and easy to adjust. This would probably have a small captive screw that could be done up to prevent the nose cone from backing off.
- I was thinking of using sand or potentially lead shot as the ballast material as water may tend to slosh around. This ballast could be in small bags or containers such that mass can be added in increments. The length of the tube could be adjusted to provide a wider range of ballast mass if needed. The diameter of the tube could also increase to provide more volume for ballast.

That seems to be about all of the small details I think of at this stage without prototyping anything. Let me know if this concept would tick all of the requirements for the wind drift vehicle and if there are any glaring issues."

FEEDBACK

From this I received the following feedback from Rocket Lab:

"This looks great! You've clearly put a lot of thought and effort into this – very impressive. You seem to have ticked all the boxes for our requirements.

Per your questions, we are happy to help with parachutes – we have a number of small model rocket chutes, or could order something suitable. We've used RocketMan and FruityChutes before.

A word of caution about carbon fibre – it is a great structural material, but is also opaque to RF, so any antennas will need to be mounted externally to ensure a signal. Plastic or GRP might be a simpler option.

A sealed box to house the electronics might make ensuring water tightness easier.

In terms of sourcing parts, the best option would be if you can provide a shopping list – URL, part name and quantity and we will order. It gets tricky trying to reimburse costs, so Rocket Lab sourcing and supplying parts would be the preferred option and keep our accounts team happy. We can also claim the GST and often have accounts with common suppliers which may provide a discount.

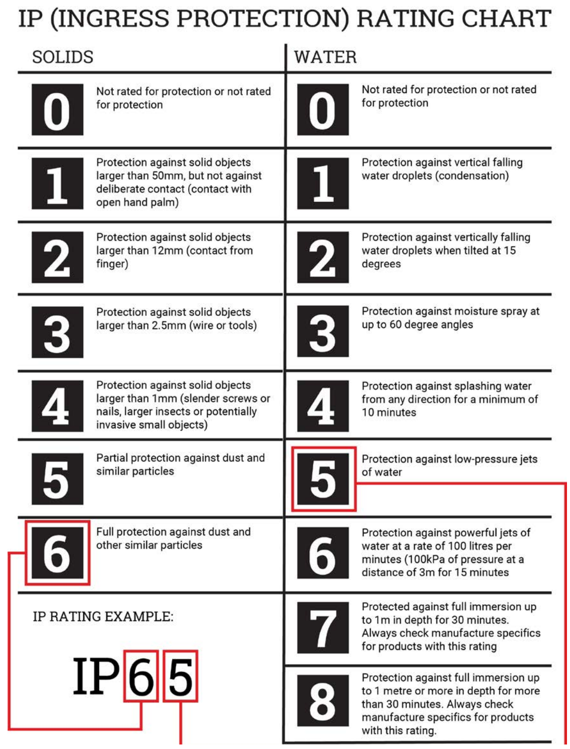
Many thanks and look forward to hearing from you."

ELECTRONICS ENCLOSURE

From this feedback, I changed to the aforementioned PVC pipe for the main body tube and began looking into a suitable waterproof enclosure for my electronics as per Rocket Labs' suggestion.

I had previously seen many different types of waterproof enclosures used in DIY projects that I think would apply to my project. These usually consist of a bottom enclosure with a lid that has an O-ring seal around its perimeter.

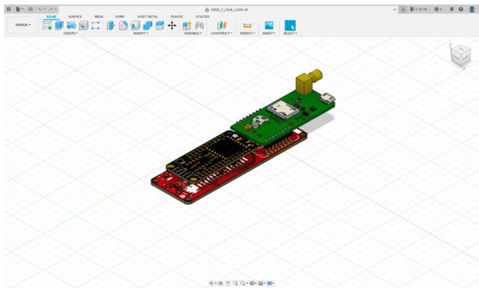
These enclosures come in a variety of IP (Ingress Protection) ratings which indicates the degree of protection the enclosure provides. The first digit indicates the intrusion protection level provided by the enclosure. The following digit indicates the moisture protection level. For my application, an IP67-rated enclosure would be more than sufficient as it provides total dust protection including other particulates and protection against full immersion for up to 30 minutes at a depth between 0.15 - 1m.



Ingress Protection Rating Chart

Due to the relatively high descent rate, the vehicle will be dropped over water as an impact above 5ms could result in damage to the vehicle. I would assume the IP67 rating would be sufficient as it would be unlikely it would take more than an hour to recover the vehicle and this enclosure would be inside the body tube so it acts more as a backup water ingress protection method.

Luckily all of the electronics I planned on using had CAD models supplied by the manufacturer so I was able to assemble these parts in Fusion 360 so I could verify the size requirements of the enclosure I would need. With these dimensions, I searched Digikey and Mouser and came across a suitably sized enclosure that would be slightly bigger than needed to ensure my parts would fit.



All of the Imported PCB Models Assembled

The material the enclosure was made of was also another consideration. Fully metal enclosures would shield the electronics from RF (radio frequency) interference but made customising the enclosure much more difficult. Although since my GPS module already had RF interference attenuation, the RF shielding of the enclosure would not be necessary. Therefore I decided to go with the ABS plastic enclosure with the clear lid as it would be easy to customise with the tools I had.

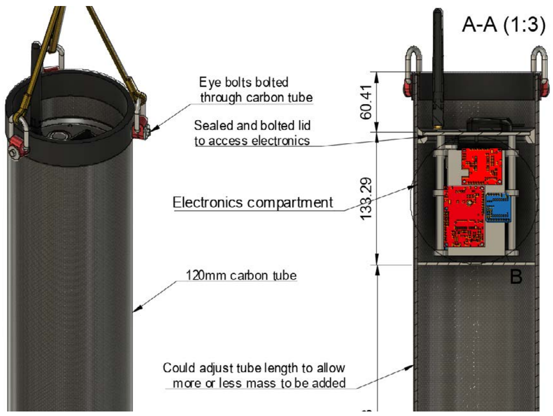
Metal IP67 Enclosure

ABS Plastic IP67 Enclosure

It also meant that I would be able to see the various indicator LEDs on the electronics such as the power led, battery charging status led and GPS PPS (Pulse Per Second) to verify various things are powered and working while the enclosure is sealed.

LORA ANTENNA

Ideally, all of the sensors, antennas, etc would be protected by the body tube extending past the top lid (see diagram below) such that it would not be possible for the parachute attachment straps to catch on anything. This would mean that the shortest possible LoRa antenna would be required so as to not have to sink the top lid deep into the tube. Thus keeping the total length of the vehicle to a minimum ensures it's easy to handle.



Electronic Mounted Slightly Below Top of Tube

I researched to find the shortest antenna possible. The most common style is the half-wave 900 Mhz LoRa antenna. However at around 20 cm in length, it would be too long, so an alternative would be needed.

This is when I found out about quarter-wave antennas which are half the length of the half-wave antenna but require the use of a ground plane below the antenna. Essentially the ground plane acts to reflect the radio waves back towards the antenna and thus increase signal strength. This means the antenna can be half the size but combined with the ground plane, perform the same as the larger quarter wave antenna.

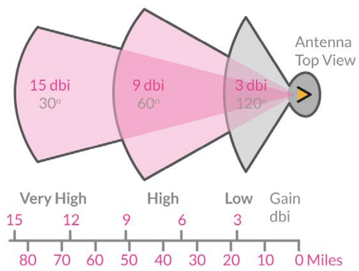
1/4 Wave Ducky LoRa Antenna

For the antenna styles, I decided to choose the “ducky” antenna due to its omnidirectional radio wave radiation pattern. Other people on model rocket forums indicated having good success with the use of a “yagi” antenna.

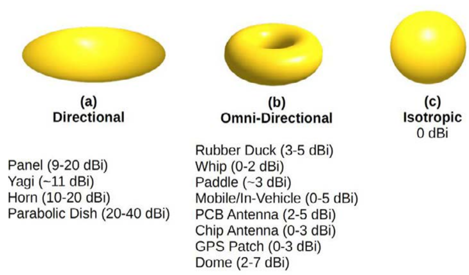
These antennas boast a highly directional radio wave radiation pattern but have the downside of needing to have a line of sight to the receiving/transmitting antenna. This wouldn’t be suitable for either the transmitter or receiver in my application as it is likely the vehicle would be so far away that it would be impossible to see it to accurately point the antenna at the vehicle even though it would technically be a “line of sight”. Also, it would be extremely difficult to have the antenna on the vehicle always point towards the ground station.

Yagi Antenna

For these antennas, a good measurement of their capabilities is antenna gain, in dbi, which essentially measures an antenna’s ability to radiate more or less in any direction compared to a theoretical antenna. From this a low gain antenna of close to 2dbi would be most suitable as I don’t need to transmit over extremely long distances but need the most spherical radiation pattern to ensure no matter the vehicle’s orientation, a signal is broadcast and can be received. From here I looked for antennas that were also listed as being suitable for outdoor applications or were IP rated since the antenna would be externally mounted and thus exposed to water, dust and debris. Searching Digikey and Mouser, I settled on the form RFsolutions due to it being intended for outdoor use and was the cheapest option at roughly $16.



Antenna Gain Diagram



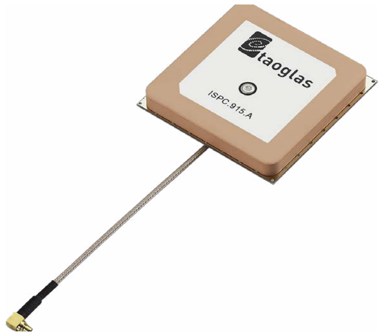
Antenna Types and Radiation Pattern

GPS ANTENNA

With the LoRa antenna sorted, I began researching antennas for the GPS board I intended to use as the board didn't contain an onboard antenna. From the information I could find online, almost every GPS antenna consisted of the same flat ceramic patch style antenna.

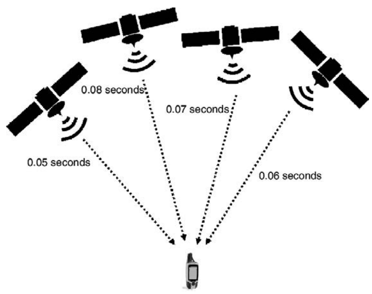
Helical Antenna

This was aside from the helical antenna alternative but due to being extremely expensive, upwards of $350, it was definitely out of the scope of this project. The ceramic patch antenna was much more affordable at roughly $20 to $30 so was the go-to option for this level of project. The main difference that determined antenna performance was the size of the ceramic patch. Essentially the larger the antenna, the better it was at receiving signals from satellites.



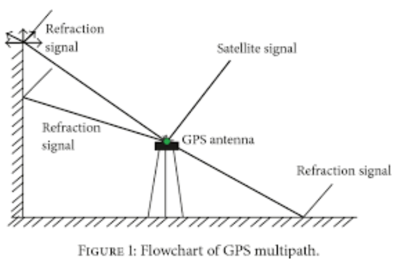
Bare Ceramic Patch Antenna (25x25mm)

The most common size is a 25mm square patch as it strikes a balance between performance and cost to manufacture so hence is the most popular option. For this reason, this size would be the one that I ended up choosing. Since there were so many manufacturers of essentially the same product, I sorted the products by price and set a minimum IP67 water resistance rating. From there, it was essentially down to choosing which one looked the best quality for the price. I ended up choosing one from the same manufacturer of the GPS module that I was using.



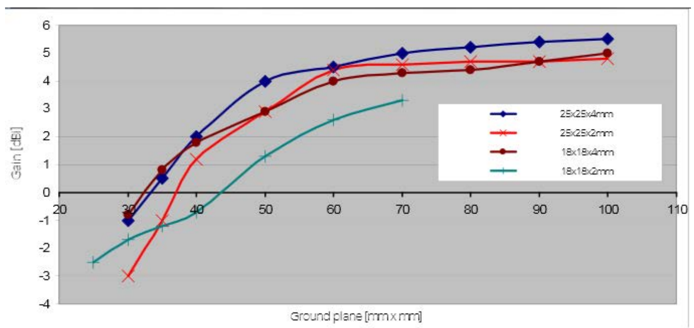
GPS Time Siganl From Satellite

While looking through all of this, I came across several Youtube videos and forum posts that suggested the use of a ground plane for GPS antennas would significantly improve antenna performance. Since GPS relies on accurately timed signals from satellites, the time between these signals can be used to determine the device's speed and thus combining multiple signals from different satellites allows the device to triangulate its position. The problem that arises is the fact that the antenna is not mounted on the ground but rather some distance above it. Since the satellite signals can bounce off the earth's surface, they can hit the underside of the patch antenna causing the signal to get to the GPS module. While this may not seem like a bad problem, we are getting more signals than without the signal bouncing off the earth. But since that signal takes a fraction of time longer to bounce off the ground then hit the antenna the accurate timing of the signal produced by the GPS is altered so the GPS will think the device has moved more or less than it had. This causes a large amount of error in the position the GPS reports compared to its true position so is undesirable. By adding a ground plane, a simple sheet metal disk, underneath the



GPS Signal Reflecting Off Ground

antenna we can reflect the signals that would have hit the underside of the antenna away from the antenna such that they can't be received. This means the antenna only will receive signals from the sky and not ones reflected off the earth. The exact size of the ground plane is mostly down to the maximum size that would fit on the lid of my electronics enclosure. However, a forum post I found suggested that if I could fit a 70mm diameter ground plane, there would be a noticeable increase in GPS positioning accuracy.



Test Results By Stack Exchange User Comparing Ground Plane Size

It seemed possible to fit this onto my lid however I was a bit concerned about having to mount the LoRa antenna very close to the GPS antenna as I was unsure if they may create unwanted interference leading to unreliable data. Since I am no expert in this field, I decided to contact Rocket Lab to advise on whether this would be an issue or not.

CONTACTING AN EXPERT

My stakeholder Matt Darley put me in contact with one of their communications experts Matthew Lawrence. The email I sent to him went as follows:

Hi Matt,

I'm building a drop test vehicle for Matt Darley and need to mount a GPS antenna along with a 915MHz LoRa antenna. The antennas would ideally be mounted to a round lid with a diameter of ~150mm.

I was looking at using a 1/4 wave antenna (https://www.digikey.co.nz/en/products/detail/linx-technologies-inc/ANT-916-CW-RH-SMA/1962849) since it is quite low profile.

However it specified it required a proximity ground plane in the data sheet. The GPS antenna I plane to use it is this one (https://www.digikey.co.nz/en/products/detail/abracon-llc/AECW0401G4ZS-3000S/13574216?s=N4IgTCBcDaIIFEDCB1ADAFjQRgOIYC0QBdAXyA) and it doesn't explicitly specify it requires a ground plane but researching around suggests a ground plane increases antenna performance?

So I am wondering if it would be possible to mount both antennas to the same ground plane and if so do they need to be in the centre, specific distance apart, etc so that they don't interfere with each other?

Would there be any best practices for mounting both of these antennas?

Attached is the initial mock up for reference (note: the tube diameter is incorrect in the drawing and has increased to ~150mm)

RESPONSE

Soon I got the following response:

Hi Jasper,

I'll preface this with saying that it's great that you have considered the comms/ RF aspect and that this will likely work in its current state as LoRa CSS is very resilient.

For this application; yes it is possible to mount both antennas to the same ground plane.

A ground plane is best practice. It minimises wasted radiation in the reverse direction. The GPS and LoRa antennas are at different frequencies. The signals may 'mix' with each other but receiver filtering will deal with this.Place them sufficiently within the ground plane such that side-beams are covered by the edge of the plane. (The Lora antenna ground plane should be min $\lambda/4$ radius)

You should reconsider the position of both antennas relative to the carbon tube as carbon fibre is not radio transparent.

For the GPS Antenna you're reducing the Sky FOV from 180 degrees to 120 degrees or so.Consider RF transparent material (e.g. fibreglass) for that section or,Changing antenna position higher up the tube or, radially mounted antennas (in electron we have two antennas 180 degrees apart and combined to give 360 degree coverage)

For the LoRaWAN antenna you're likely blocking the FOV to the ground/ roving station.Consider mounting the LoRa antenna in the nose cone if it is RF transparent with the ground plane "above" the antenna to direct more RF energy downwards.
If not RF transparent, you could hang an 8cm wire ($\lambda/4$ antenna) somewhere off the bottom of the vehicle and that should provide close to a 50Ω impedance match.

Let me know your thoughts.

Cheers,

Matt

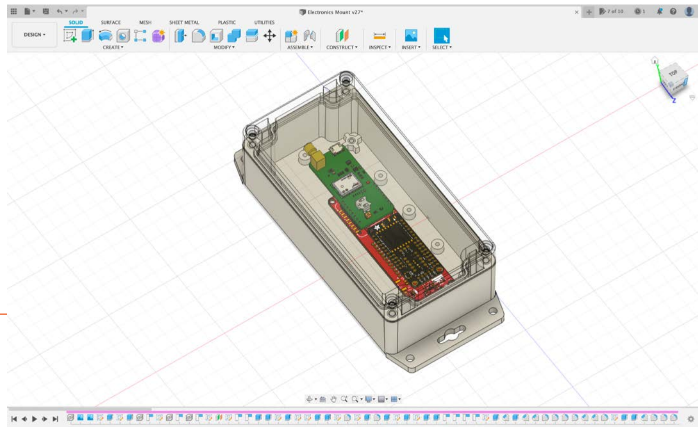
CHANGES TO THE DESIGN

This was super helpful and reassured me that using the LoRa would be an excellent option for communication with the ground station. He also brought up a good point about the fact that the roving ground station would have a better view of the antenna if it were to be placed on the underside of the vehicle.

He also confirmed that a ground plane would be necessary for the ¼ wave antenna that I had chosen and given me the specific size it should be which was incredibly helpful as I was able to trust an expert rather than interpreting data from the internet. It was also good to know that if I had to resort to mounting the LoRa antenna anywhere close to the GPS antenna the inbuilt filtering of the GPS module would deal with any signal mixing that could occur. Since I would want to move the antenna down to the underside of the vehicle and a nose cone would be required to reduce impact force on the bottom of the vehicle, mounting the antenna inside the nose cone would be a good idea so long as the nose cone was constructed from an RF transparent material.

ORDERING PARTS

With these changes made, I began finalising and double-checking that all of the parts I would need would be compatible and fit into my enclosure. Basically doing a sanity check that I haven't made any mistakes due to shipping for parts taking around a week and a half so re-ordering parts could significantly set back progress. The main reason for designing a lot of my ideas in CAD was the fact that it didn't cost anything other than my time. Since I would be spending a decent amount of money on this project it was important that everything would work as or as close to expected as re-ordering parts over and over since they weren't suitable, is not very cost-effective and could increase the total projects cost significantly. I then sent my shopping cart over to Rocket Lab and they got the parts all ordered and on their way down to me.



Electronics Inside Enclosure to Confirm Sizing

UPDATING CAD MODEL

While these parts were being shipped, I began working on turning the quick mock-up CAD model into a fully functional version that would be able to be 3D printed and assembled. A lot of my knowledge around designing for 3D printing came from an extensive background in owning various 3D printers and using Fusion 360 since around year 9 or 10 and building a lot of functional projects involving 3D printing.

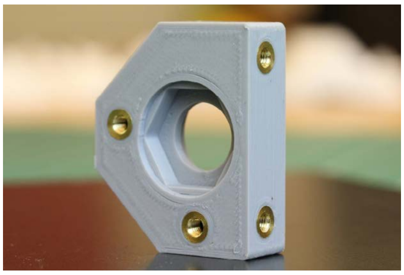
ELECTRONICS MOUNT

The first part I began 3D modelling was the mount that would hold the Sparkfun RP2040 board with the GPS module and LoRa feather board and would need to interface with the mounting holes on the enclosure. I opted to purchase thermoset threaded brass inserts that I could insert into the 3D-printed mounting bracket as well as into the waterproof enclosure mounting holes. The use of these threaded inserts is common practice for structural 3D prints due to the increased pull-out strength compared to self-tapping screws into 3D printed holes.

While my application was not critical that it be overly structurally strong, the fact that the electronics may need to be removed and then reassembled multiple times since I was customising the waterproof enclosure as I went. The threaded insert increases the life of the hole and ultimately the number of times

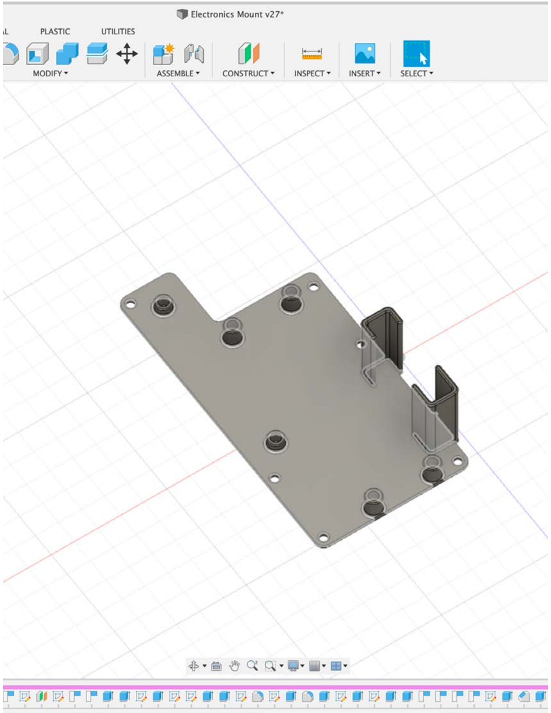
Thermoset Brass Threaded Inserts

the fastener can be inserted and removed without stripping the threads.



Threaded Inserts in 3D Print

Another bonus was that I could use Loctite on all of the fasteners to ensure that they wouldn't back out due to vibrations causing parts to come loose. Since Loctite could only be used on metal-to-metal fasteners, using the threaded inserts ensured I could use Loctite as the Loctite bottle suggests it may react with certain plastics causing the plastic to become brittle.



Final Electronics Tray Mount

3D Printed Tray Mount With Threaded Inserts Installed

Test Prints of Each Iteration of the Design

POWER BUTTON

A power button would be needed. However, to ensure that it would be difficult to accidentally turn off the device, a software-controlled power button would be needed. This would allow me to set the amount of time the power button would need to be held to turn off the device. Therefore I could set it to something like 8 seconds to ensure that if it was held for this long, it would be most likely that the button press would be intentional. It also means that I can use the externally mounted power button to act as the power button and as a multifunction button.

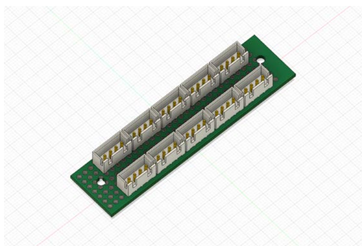
SparkFun Soft Power Switch

When the electronics are plugged into power, they would start logging data so I could instead have a double tap to change modes such that the device could be put into and out of a charging mode and vice versa for a data logging mode. I found several DIY circuits that I could build to solve this problem but stumbled on the Spark X page on Sparkfun's website. Sparkfun described these products as:

"products that are rapidly produced to bring you the most cutting edge technology as it becomes available. These products are tested but come with no guarantees."

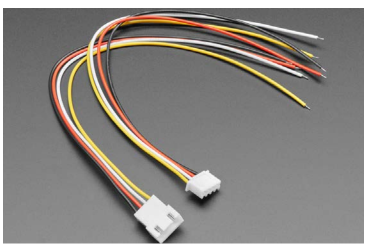
There I found exactly what I was looking for, the Spark X soft power switch, which had all of the functions described above and would go in between the battery and the power lead to the RP2040 board. It also allowed the battery to be charged when power was supplied via the USB port and allowed the use of an external power button. The main drawback described on the product page was that using the power button on the board could lead to unpredictable behaviour due to the user's finger potentially shorting a capacitor. However, they say that this problem is mitigated with the use of an external power button which means that it should work fine for my application.

To attach all of the cables that would exit the enclosure, I decided to create a very simple breakout board. I chose the breakout board rather than inline connectors as it would allow me to cable manage the cables coming off the electronics boards such that removing cables that exit the enclosure wouldn't strain the connections to the electronics. It also would make removing the lid from the enclosure easier as all of the cables to disconnect would be collated in the same spot.



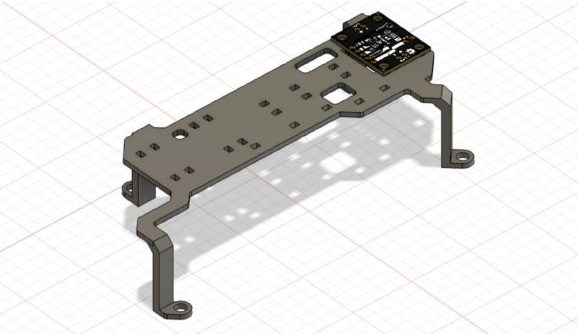
JST XH Conenctor Breakout Board in CAD

I decided to use JST XH connectors since they were common in the RC hobby and could be bought in NZ for cheap. They are also large making them relatively robust and easy to work with. I purchased ten female XH sockets and the matching plug. These were soldered to a small bit of prototyping board and had the connectors joined with a small piece of wire. Holes were drilled for mounting the breakout board and standoffs were added to the electronics mount.

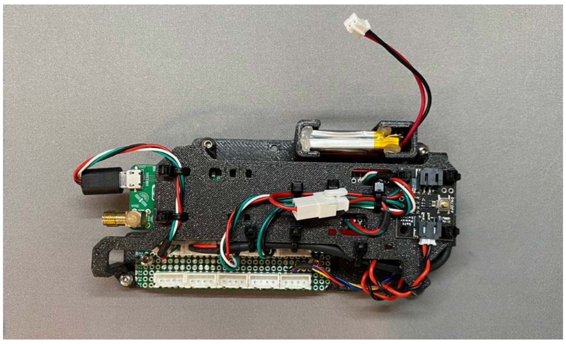


JST XH Conenctors - Cable to Cable

I had originally planned to mount the soft power switch mentioned above in a similar fashion to the battery but once the cable breakout board was added, there was very little space for cables. Therefore I decided to create a cable management cover that would go over the electronics and allow the soft power switchboard to be mounted and would have holes for zip ties to manage the cables coming off the electronics. This went through several revisions to add all of the appropriate holes and ensure the legs wouldn't snap off while removing it from the 3D print (which I managed to do on several occasions).



Cable Managent With Soft Power Switch Board in CAD



Cable Managent Cover Installed With Cables Managed

Multiple Versions 3D Printed to Refine the Design

CABLE GLAND MOUNT

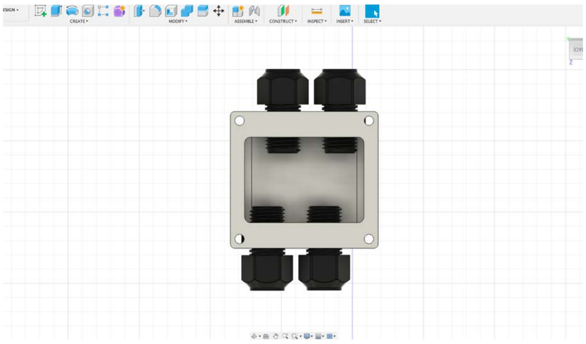
To ensure a waterproof seal around the entry and exit of various cables into the enclosure I decided to use cable glands. These are essentially a large rubber grommet that clamps down around the jacket of the cable sealing against water ingress. These are commonly used in outdoor applications so do come with an IP67 ingress protection rating making them suitable for use in my project.

Cable Gland Exploded View

The size of the cable gland is directly related to the size range of the wire that can be sealed. For my application I would need the smallest possible external size as space would be limited. Also the shortest length cable gland would be ideal since mounting perpendicularly to the side of the enclosure and accounting for the minimum bend radius of the cable, the total length the cable and cable gland would extend would be large.

For this reason I decided to mount the cable glands parallel to the top lid of the enclosure to save space. This also meant that the total length of the enclosure could be made as short as possible which keeps the total vehicle length short, improving the handle-ability of the vehicle.

Therefore I moved into Fusion 360 to design a small mount that could hold the four cables that I would need to enter the enclosure. The main challenge I faced with modelling this part was that the cable glands couldn't be mounted directly across from each other as the cables may not be able to bend past each other. For this reason they had to be slightly offset from each other. I also forgot to update the CAD model to include the slightly larger body tube of 150mm diameter vs the original 120mm version (due to the fact I could only buy 150mm tubes at the local plumbing store and that the vehicle could be made shorter and wider while holding the same amount of ballast). Therefore I had modelled the cable gland mount with the notion that it wouldn't be able to be rectangular but rather have the corners cut off so that it would fit inside the tube.



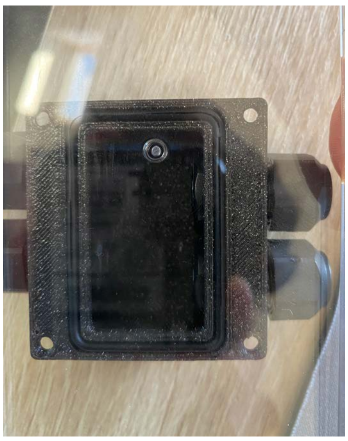
Offset Cable Glands so Cables Don't Hit Each Other

This meant that the cable glands would need to be mounted closer together than they need to be which took a number of 3D printed prototypes to verify all of the tolerances and clearances. Having 3D printed a final version of the mount, I covered the entire mount with a clear nail polish gel.

While I had printed the mount at a very fine layer height to minimise the chances of small holes in the 3D print which could allow water ingress, the nail polish was a reassurance that it would be 100% waterproof.

BACK PLATE AND SEAL

The mount would also need a backing plate of sorts to go on the underside of the lid of the enclosure so I made a very simple design that would allow a threaded insert to be installed and distribute an even mounting pressure.



O'ring Squish Against Lid

The original plan was to use a small piece of O-ring cord I had gotten from work but while dry-fitting the mount using the bracket I had made, I found that I was getting very uneven mounting pressure. Unfortunately, the culprit was the fact that the surface that the O-ring interfaced with was 3D printed so was not 100% flat. This combined with the relatively stiff O-ring meant that it was very difficult to get an even mounting pressure to ensure the coring would be compressed along its entire length.

Mounting Bracket

Since the lid of the enclosure was clear, it could easily be seen where the O-ring was not being compressed and thus not making a watertight seal. I decided to forget this option in favour of some red RTV silicone. I was able to fill the O-ring channel and the entire face of the mount with the red RTV silicone which once dry would form a watertight seal. I opted for this over caulking silicon as the RTV silicone is not much of an adhesive as caulking silicone is. This meant it acted like a very weak glue so parts could be disassembled if changes need to be made. To ensure the mount would be lined up in the middle of the enclosure lid, I designed a small hole drilling jig that used two edges of the lid to align the lid. Thus holes could be accurately drilled to ensure everything was lined up correctly.

Red RTV Silicone Seal

Bolt Holes Filled With Silicone

Silicone Around Edge For Complete Seal

The centre hole was then dremeled out and edges chamfered with a knife so that the cables couldn't be cut by the sharp plastic edge. The final assembly of this part was done and I was sure to apply RTV silicone in the bolt holes to ensure they would be watertight.

Different Versions 3D Printed to Refine Shape and Tollerances

ANTENNA EXTENSION MOUNTING

My original plan to have the SMA antenna connectors enter and exit the enclosure was to have them pass through a cable gland. However, due to the small diameter of the cable that the SMA connector is attached to, it was virtually impossible to find a cable gland that could allow the SMA connector to pass through but clamp down on the cable to make a watertight seal. Therefore I would need to take a different approach so switched to using an SMA extension with a bulkhead fitting on one end. This combined with a 90-degree adapter fitting allowed me to connect to the SMA connector on the GPS board.

SMA Female (Left) SMA Male (Right)

SMA 90 Degree Apdater

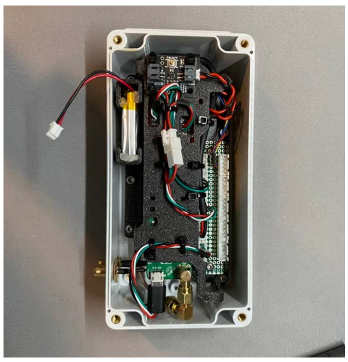
The Adafruit LoRa feathering board I purchased had the option to use either a full-sized SMA connector like the GPS board or a very small u.fl connector. I opted for the u.fl connector as once the feather board was stacked on top of the Sparkfun RP2040 board, there would be no room for a full-sized SMA cable to be routed without interfering with the GPS board. Soldering this on was quite difficult due to its tiny size but I managed to do it without damaging the connector. Continuity was checked to ensure the solder joints were making a good connection.

U.Fl Connector

I found a waterproof bulkhead u.fl to SMA connector which would allow me to mount the bulkhead alongside the GPS SMA bulkhead. Unfortunately, the thickness of the enclosure was much thicker than the maximum panel mounting thickness of 3.2mm as specified in the bulkhead data sheet. Therefore I would need to route out a slot where the SMA bulkhead connectors would be mounted such that the enclosure's wall thickness at that point would be 3.2mm or less. To do this accurately I designed and 3D printed a custom jig to allow me to first drill the SMA bulkhead connectors mounting hole. The second jig had the slot size I would intend to route out with the Dremel and was large enough to allow the Dremel routing jig to not fall off the edge of the 3D printed jig.

Hole Routing Jig

Since the edge of the enclosure was slightly tapered, I slowly routed the slot deeper and deeper until the bottom hole's wall thickness was 3mm. This would mean the top holes' wall thickness would be slightly less due to the enclosure's tapered walls but still thick enough to



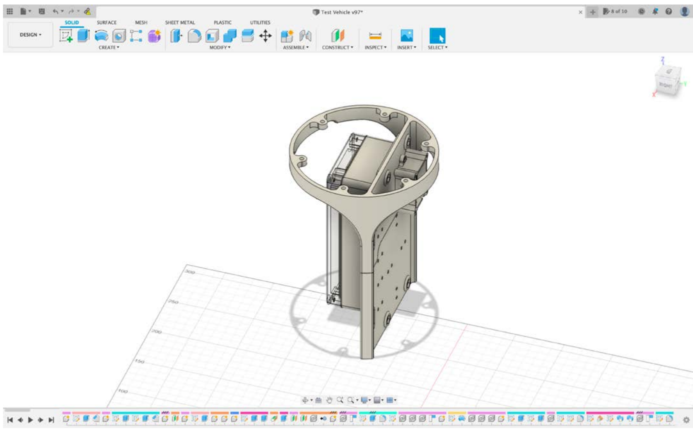
SMA Connectors Exiting Bottom Left

not break if the bulkhead connector got knocked around. The mounting location was just under the GPS module as it was the only spot I could fit the SMA extension cable from the GPS. As this was not the original plan, I had to add a cut-out in the electronics mounting bracket. Luckily SMA bulkheads were the last thing to be added to the enclosure so once the cutout was added, the electronics mount would be the final version.

ENCLOSURE MOUNT

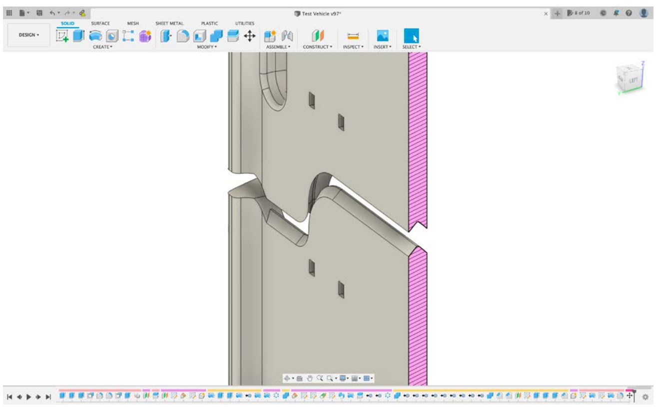
The first version of the mount that would mount the electronics enclosure to the lid included a large ground plane that would span the entire lid to provide a ground plane for the GPS and LoRa antenna. However, with the relocation of the LoRa antenna to the nose cone, this large ground plane would no longer be necessary.

The first version was intended to be multiple parts that would use 3D-printed snap-fit joints with zip tie mounts to avoid the use of screws. Since I decided to switch to a threaded pipe cap I no longer needed to print the lid so the electronics enclosure could mount to a bracket that simply used screws to mount to the lid.

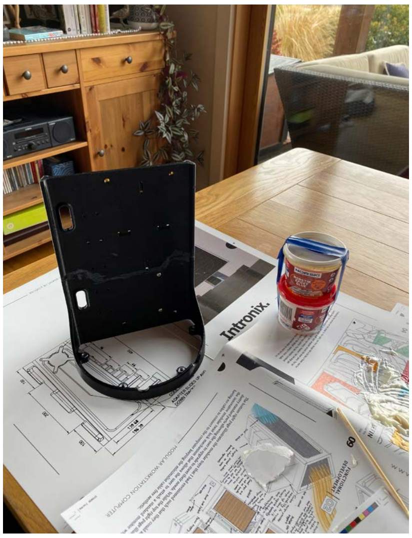


Electronics Mount Design on Fusion 360

Since my 3D printer, the Prusa Mini, has a maximum build height of 180mm, I had to split my mount into two pieces that could be epoxied together. Since a lot of things had to be changed in this second version, it was easier just to start from scratch than to alter the existing 3D model. In doing this I added quite a few extra details that I had missed from the first version such as captive nut mounts for the electronics enclosure mount, an extra cable pass-through hole, cable management zip tie mounts and a mount for the altimeter holder which I was yet to finalise.



Interfacing Method That Allows Halfs to Self Align



Epoxying Both Halfs Together

The joint between the two pieces was one that I had seen in a 3D-printed RC car whereby the joints' edges were chamfered to cause the parts to self-align. I employed this method as I knew it would work and was quite strong even without glue however in my case I epoxied the halves together to ensure it wouldn't come apart.

ALTIMETER MOUNT / ENCLOSURE

The Sparkfun MS5673 is the altimeter that I settled on as it has an accuracy of ± 13 cm so far more accurate than I would need. However, this would need an air-tight enclosure within the body tube of the vehicle. It would then have a vent tube that went out the lid so that the air pressure outside of the vehicle could be passed inside to the altimeter. This vent tube snakes around the inside of the electronics compartment in an attempt to prevent water from reaching the altimeter once the vehicle lands in the water as the module is not waterproof.

SparkFun MS5637 Altimeter


Altimeter Enclosure Deisgn in Fusion 360

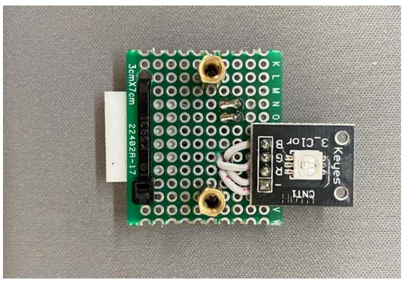
Having the vent tube was quite similar to what is done in model rocketry however they don't need to be waterproof so hence my vent tube needs to snake around to prevent water ingress. In terms of the enclosure that it would go in, I chose a really simple design that had large flat surfaces so that red RTV silicone could be applied to easily create an airtight seal. The cable to the altimeter was terminated with a JST XH connector so that it would be easy to swap the altimeter board for a new one if it got water damaged. Obviously, the red RTV silicone would need to be reapplied but this creates a very reliable seal so is better than using an O-ring in this specific case.

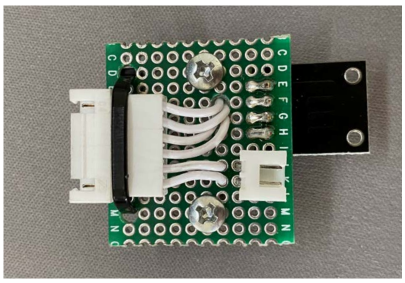

Panel Mount LED

STATUS LED & BREAKOUT

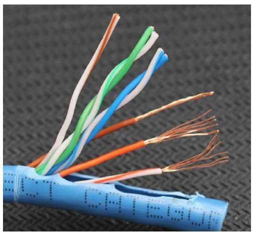
In order to make it as obvious as possible to the user the state that the vehicle is in i.e charging, data logging, etc, I decided to add a RGB LED. I looked into options and found that a tri-colour panel-mount LED would work well as they came in a IP67 variety. However, coming in at the $20 - $30 range, this seemed like a very expensive option.

Doing some digging in my drawer of electronics bits and pieces I found a SMD RGB LED breakout board that could be suitable.


LED Break Out Underside


LED Breakout Connectors

Unfortunately since it was a breakout board and not designed to be panel mounted, I would have to build my own mounting solution. I planned to mount the LED breakout to a piece of proto board since this could then be mounted to some threaded standoffs.



Twisted Pairs in an Ethernet Cable

Since this would be mounted near the power button, I decided to add a connector so that the power button would attach to the LED's mounting PCB. Therefore I could then use a 6 pin connector which would mean that one ethernet cable could run to the electronics enclosure rather than a power button cable and LED cable.

LED WINDOW

I also found a piece of acrylic that I already had. Using the dremel, I was able to remove material so that the acrylic would poke through a circular hole. This was then glued to the lid and caulking silicone was applied around its perimeter to help hold it in as well as seal around it to prevent water ingress. The LED board could then be mounted and a cable made to connect to the electronics enclosure.

Acrylic Piece in Drilled Hole

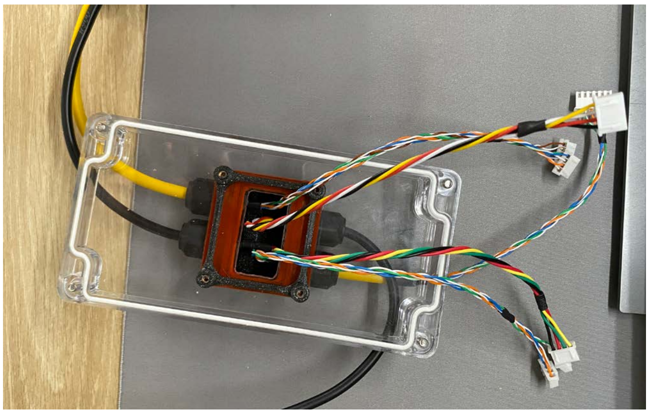
Acrylic Piece and Mounting Hole

Side View of Cut Acrylic Piece

Since an ethernet cable has four twisted pairs i.e. 8 conductors total, I removed one twisted pair. However, since the cable only needed a small amount of cable jacket to pass through the cable gland, I had to apply red RTV silicone to the part where the cable jacket ended as the jacket could slide over the twisted pairs. This was then heat shrunk over to create a waterproof seal. Most of the jacket of the cable was removed as the twisted pairs can bend more easily to reach the LED as well as placing less strain on the connector on the LED breakout board.

CABLES INTO ENCLOSURE

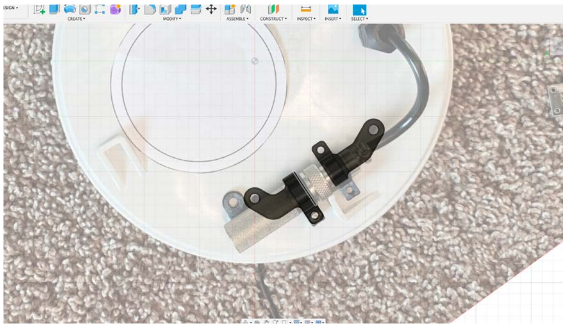
The other cables from the altimeter, GPS USB, Microcontroller USB and temp & humidity sensor had a JST XH connector crimped on the end. The two USB connections used a single ethernet cable and split into two four-pin connectors while the altimeter and temp and humidity sensor used a single four-conductor



Cables Terminated With JST XH Connectors

26 AWG wire. The cables first had the crimps crimped on the conductors then were passed through the cable gland where they were inserted into the plug housing. I had made many of these types of cables at work so I am quite experienced in making cables, however, I didn't own the tools so had to go and purchase the necessary tools. I probably could have used the tools at work, however, since I worked on this project in my free time and the fact that it was very prototyped in nature, having the tools made it much more convenient to make alterations or fix changes when I needed to.Temperature and Humidity Sensor Mount

Final Sensor Mount Design

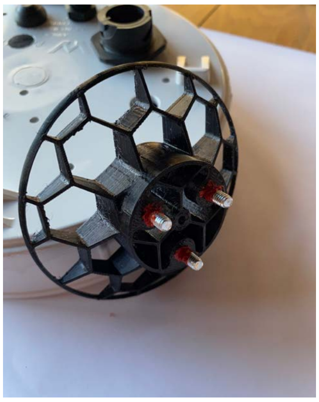


Using a Canvas in Fusion 360 to Ensure the Mount Would Line Up

To mount the temperature and humidity sensor onto the pipe lid, I used the existing electronic enclosure mounting holes. A simple clamp system that used a nut and bolt to clamp around the body of the temp and humidity sensor was used as it allowed the sensor to be removed and reinstalled without removing the entire mount from the lid. This mount went through two iterations to fine-tune the angle the sensor would be mounted so that it wouldn't interfere with the knobs on top of the lid or anything else mounted on the lid. Once the design was finalised, it was 3D printed and then installed with red RTV silicone to seal the bolts through the lid.

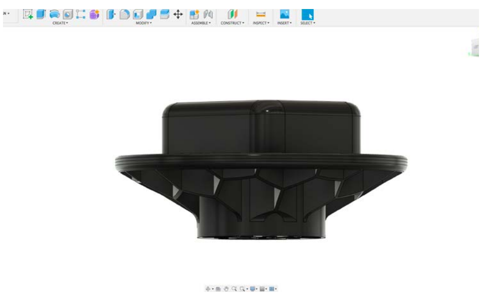
GPS PEDESTAL MOUNT

With the temperature and humidity sensor mounted, I was slightly concerned about the metal enclosure of the sensor interfering with the GPS's signal. I figured it may act like a ground plane preventing signals from hitting the GPS antenna so I decided the GPS antenna should be mounted above the temp and humidity sensor to avoid reducing the GPS antenna FOV (field of view) of the sky.



3D Printed Stand

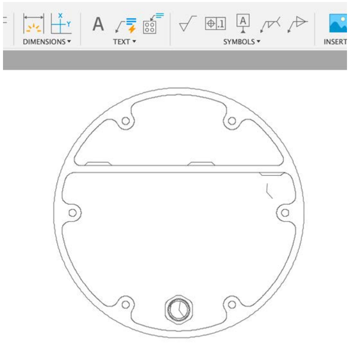
The first thought was a simple cylindrical pedestal that the 70mm ground plane would mount to and thus the GPS antenna to that ground plane. However, this would take up significant space on the lid so I decided to look for alternatives. That's when I found that it was quite common for GPS to be mounted on a Y-shaped pedestal in boating.



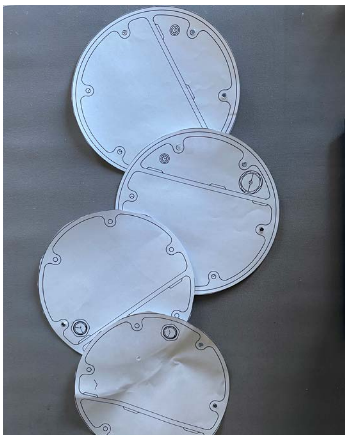
GPS Stand Design on Fusion 360

I decided to incorporate this design but make the inside a honeycomb structure to give the mount strength while being lightweight. This mount could have a ground plane salvaged from some spare sheet metal from an old printer. It then had a thin 3D printed cover where the GPS antenna could be double-sided taped to the ground plane. This whole mount could be screwed to the lid while making sure to seal any holes with red RTV silicone.

Template for Drilling Mounting Holes



Drawing on Fusion 360



Various Templates Used For Mounting

MOUNTING TEMPLATES

For every hole I drilled in the top lid, the CAD model on Fusion 360 was updated, and then a drawing was created that I could then print on paper and cut out. These templates allowed me to ensure I was drilling the holes accurately and once mounted, everything would be in the correct location. This was especially important as I had a lot of things to mount to the lid so the tolerance between parts was small necessitating precision assembly.

Outdoor Lora Receiver

GROUND STATION

Other DIY LoRa Enclosures

Going into this project I initially didn't have a plan for the device that would receive data from the vehicle via the LoRa connection. Initially, I thought of a simple box with an antenna that connected to a laptop such that the box contained the LoRa receiver and some simple circuitry to output serial data to the laptop would work. The laptop would then capture the data ready for it to be processed and manipulated.

However, the downside was always needing to be connected to a laptop. So I decided it would work best to have the receiver record data to an SD card that could then be sent to the computer in real-time or after a test had finished.

This meant the device could be set up stand-alone or tethered to a computer making it more flexible in its operation. Also, this would make it a more robust solution since if the vehicle is ever transmitting data, the receiver should be able to record that data even if a computer isn't connected.

Xiaomi Walkie Talkie

Having a look around at other solutions for LoRa base stations, a common form factor was that similar to a walkie-talkie. This led me to research different walkie-talkie designs as they would have the same antenna, screen and buttons that my design would have. One walkie-talkie I found was a model from Xiaomi. I decided to base the shell design to mimic the Xiaomi walkie-talkie.

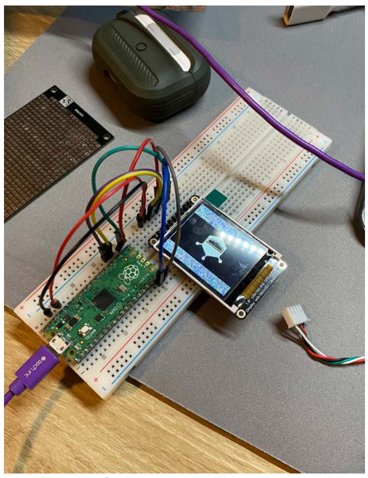
Knowing that I would need a small screen, buttons, more prototyping board and a suitable antenna, I ordered some parts from Core Electronics Australia. The size wasn't a constraint for the antenna so I decided to go with a ½ wave antenna due to it not requiring a ground plane. As for the screen I just chose a cheap model that had a micro SD card slot on the back so I wouldn't need to buy a separate breakout board. While those parts were being shipped, I started a first version that consisted of an antenna mount and tray to hold the electronics.

First Design - Failed Print

After finalising the dimensions with the parts that had arrived, I was able to 3D print the first version. Unfortunately, the print failed towards the end but enough was printed for me to test fit the parts. Also, the flexible buttons I had designed didn't print so well due to their orientation which caused there to be nothing supporting the overhangs hence the button became fused to the body thus rendering them not usable. This along with a lot of other assembly and fitment issues, I decided to look for an alternative design that could be 3D printed in multiple pieces. Therefore if a print failed, it could be reprinted which is much quicker than re-printing larger pieces. I found a design of a similar ground station device that used a LoRa radio and STM32 microcontroller.

Existing STM32 Based Enclsoure

First Version of Ground Station



Protoyping Circuits on Breadboard

Since I had mostly different parts to work with, I decided to take the design language from that design and apply it to a custom version for my application.

Battery From Power Bank

Since I had a few Raspberry pi picos laying around (same RP2040 chip in the vehicle) I figured they should be sufficient compute-wise. I started laying out my buttons, pi pico and LoRa radio onto a proto-board to figure out how they would be arranged to make the overall size as small as possible. I originally wanted to mount the screen such that it would plug directly into the protoboard, however, due to limited space and the fact that it would make the overall design very thick, I decided to mount it above the proto-board.

Test Prints

BATTERY

To power the whole ground station, a battery and charging board with a USB connector would be necessary however since I would need to buy this online, shipping would take a week or two. Therefore I figured I could simply take the insides out of a power bank and mount them inside a custom enclosure instead. Since it already had a charging port, battery and 5v USB output, it was basically all of the parts I would need but could be bought from the local hardware store.

4 Pin GX12 Connector

The mounting for the circuit board from the power bank was quite difficult since it had very tight tolerances. This combined with the intricate small features that were needed to mount the board made it especially difficult. However, with a lot of careful measuring with a set of digital callipers and small test 3D prints, I was able to replicate the features in the original injection moulded power bank shell.

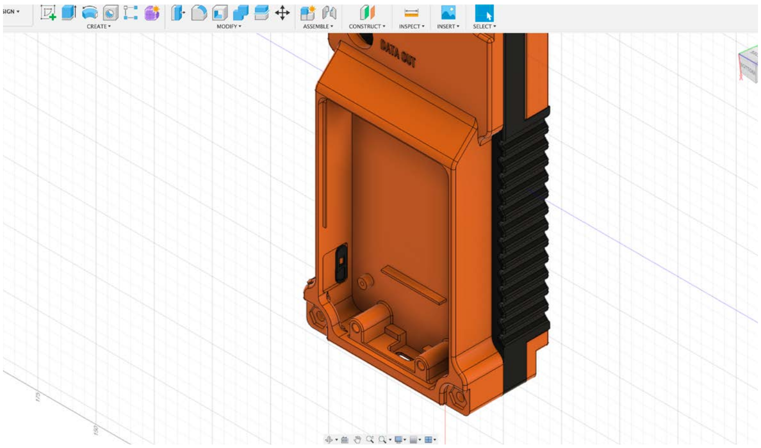
Battery Cover (Right)

USB CONNECTION

USB to GX12 Cable For Programming

I would need a way to connect the ground station to a computer via USB but due to the power banks micro usb port only being able to charge the battery and not pass data another connector would need to be added. I decided to use a four pinned GX12 connector since I already had one on hand from an older project but it also was a screw-on connector so it would be very difficult to accidentally remove the cable.

As I still needed to write code for the microcontroller, I wired the power such that power would be supplied to the Raspberry pi pico board only when plugged into USB power. This meant the power bank wouldn't power the electronics at this stage but I put an inline JST XH connector on the power into the raspberry pi pico board.



Battery Enclosure Design

I couldn't find a suitable power switch for the entire device at this stage so used the inline connectors so that one could be added in the future and when the code was finalised, the ground station could be powered off the power bank. This means the USB connection to the computer would only receive data and not provide power to the device.

BATTERY COVER

My plan for the method of attaching the power bank battery cover was to use a toolless snap-fit design. I had previously 3D printed a Raspberry Pi enclosure that employed the sort of snap-fit lid design I was after. I decided to copy the lid design from that enclosure such that it would fit into my design. 3D printing a few versions let me refine the mechanism, however, due to the back lid needing to be thicker than that on the Raspberry Pi enclosure, I kept snapping the latch off the lid while trying to install the lid. Since I didn't want to make the cover thinner to allow it to flex, I decided to switch to using some countersunk screws to hold the cover on but kept the other ends tabs so that only two screws would be needed rather than four.

Snap Fit Raspbbery Pi Enclosure (Top)

Cover iterations (Below)

Adding all of the small details and tweaks to accommodate the new parts to the CAD model on Fusion 360 allowed me to 3D print a final version. For these, I purchased some PETG filament to use instead of the PLA filament I had used for the prototypes. The main reason for the filament change was the fact that PETG was much stronger and less brittle than PLA so was more hard-wearing. The downside is that it is considerably more expensive than PLA so hence prototyping with PLA and then printing final versions with PETG.

SOLDERING

The next step was to solder all of the breakout boards onto the proto-board as well as the control buttons. I dug through my collection of old electronics to find a suitable connector for the screen and then soldered it on. The trickiest part came of using thin wire to make all the necessary connections between the components. While I was doing this, I was sure to make a reference of all the pins that were connected so that it would be easy to define what was connected to what in the code. That was by far the most time-consuming assembly aspect of the ground station. With that completed, I could use several M3 bolts to hold the entire assembly together. After checking all of the connections with a multimeter for continuity, I plugged it into power and nothing had exploded so I warranted it a success.

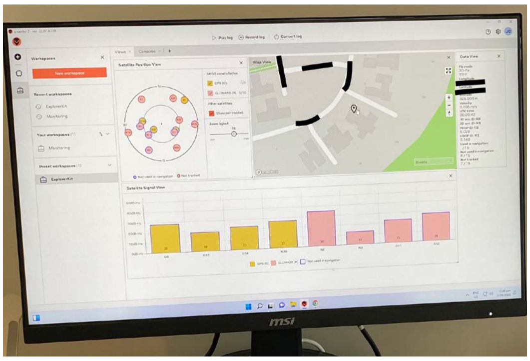
Electronics Soldered to Proto-board

Backside With Buttons

Closeup of Pi Pico and LoRa Breakout

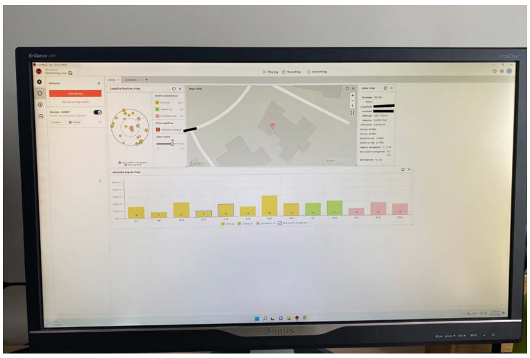
GPS TESTING

I first began by testing the GPS module as this was just a simple case of downloading Ublox Center on a PC and then plugging it in via the USB connection. The GPS module showed up in the software indicating that my wiring was all correct. Interestingly even without a ground plane and having the antenna located inside, the GPS was able to pinpoint my location down to the correct end of my house. This was a good sign as even in sub-optimal conditions, the GPS provided sufficient tracking accuracy.



GPS Test Without Ground Plane

Next, I tested the module with the antenna on the ground plane. This showed that 23 satellites could be tracked which was much greater than without the ground plane. Also, the signal strength was higher than without which definitely will improve the positioning accuracy when the GPS is in suboptimal conditions.



GPS Test Without Ground Plane

Since the test was done inside, the signal strengths are quite low so the GPS position does drift slightly but I wasn't able to place it outside as I only have a desktop PC to run the Ublox software. However, I might be able to run the program on my Macbook with Bootcamp to fix this issue. Overall the GPS module seemed to be operating correctly so I moved on to further testing.

SENSOR TESTING

Next, I wanted to test that my altimeter was working correctly so using the SparkFun provided example code, ran it on the RP2040 chip. This outputted the temperature readings and pressure over serial to my computer. Blowing into the altimeter causes the pressure and temperature to rise which indicates the sensor is working properly.

To test the temperature and humidity sensor, I used the Adafruit example code and had it simply output over serial. The temperature and humidity were displayed correctly thus indicating that the sensor was working correctly.

SCREEN TEST CODE

I was quite undecided on the approach I wanted to take to design the user interface for the ground station. There were various Arduino libraries available to use but all required the UI layouts to be written in code. That's when I found another option called Squareline Studio which was an app where you could arrange elements to create your design all without writing any code. The UI could be exported as code to upload to the RP2040 Microcontroller making for an easy-to-use system. However, after playing around with the software and exporting a simple UI layout, I found that it would become very difficult to add the other functionally that I needed to run things like the LoRa radio. For this reason, I reverted to a very popular tft_Espi library which proved relatively easy to implement. I only managed to get to the stage of drawing basic shapes on the screen since I had to learn to use the entire library. Thus I tried some of the inbuilt examples to test that the screen worked and displayed graphics properly.

Screen Test 1

Screen Test 2

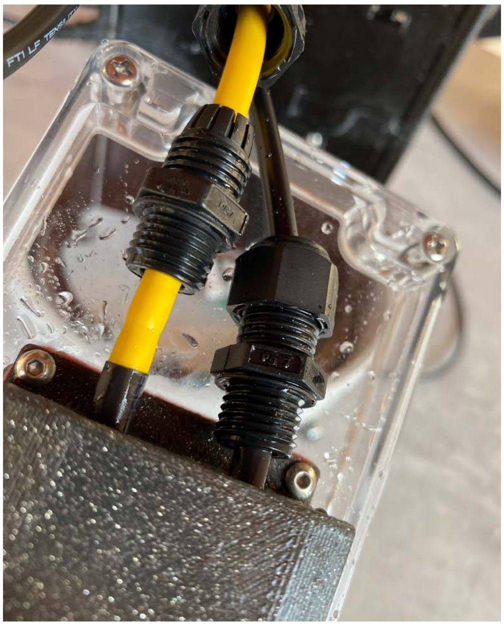
Screen Test 3

ENCLOSURE TESTING

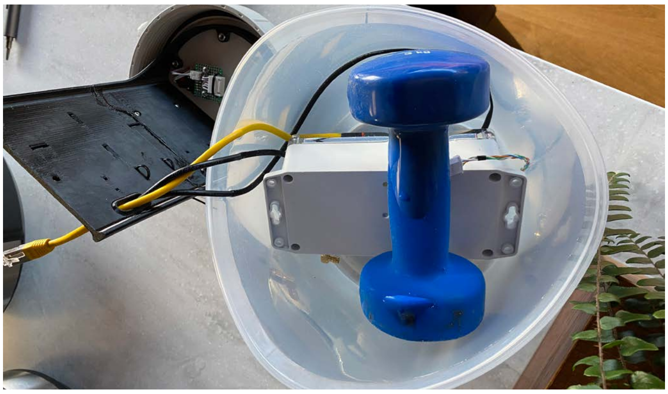
Enclosure Submerged

To ensure that my electronics enclosure was completely watertight, I decided to remove all of the electronics from the enclosure but leave the cables attached. Therefore I would be able to test for leaks in any of the modifications I made.

For the first test, I submerged the enclosure such that it was fully below the water's surface. This test didn't last for long as I could already see water leaking into the enclosure. After some careful observation, I was able to orientate the enclosure such that I could see the water dripping in. Thus I concluded that water was leaking in from around the cable glands. They have a built-in seal that I meant to provide a water-tight seal, however, I think that the slightly rough surface of the 3D print was causing the seal to not seal properly. I remedied this issue by adding Red RTV silicone around the cable glands and then tightening them up.



Cable Glands Wet Around Thread Indicating a Leak



Enclosure Submerged With Weight

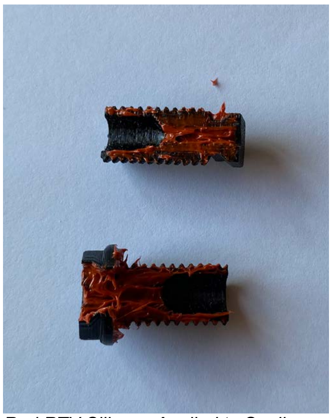
The second test resulted in no leaks and was left for around an hour of submersion so I could safely conclude that the enclosure still conformed to the IP67 rating after my modifications had been made.

SPLIT CABLE GLAND

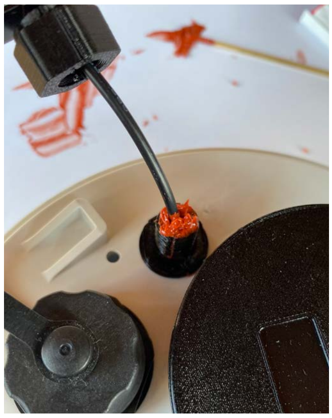
To pass the GPS SMA cable through the top lid I needed a cable gland. I couldn't use a standard one due to the SMA connector being too large to fit through a cable gland that would seal around the cable it was connected to.

A commercial solution to this problem is the use of a split cable gland. These split into two pieces to allow the cable through then clamp back down around the cable. I decided to employ this strategy by taking the design of a regular cable gland and then splitting it up such that the cable could pass through a hole in the lid, then have the cable gland fit over the cable and screw in just like a regular cable gland. This could then be filled with red RTV silicone to make a watertight seal.

Exploded View of Split Cable Gland Design



Red RTV Silicone Applied to Sealing Surfaces

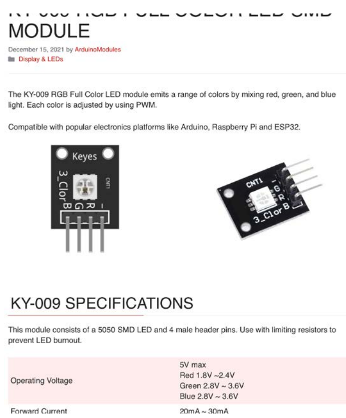


Red RTV Silicone Around Cable Exit

POWER BUTTON CODE

With a lot of things tested, I decided to start writing the code that would control the behaviour of the soft power switch. The board had some example code that showcased all of its possible functions. However, I decided just to strip out the logic from that code that would apply to my situation. Therefore I only kept the powering off and button press sense logic. This was because I wanted the power button to be held for longer than 8 seconds which could be defined in the code and when the power button was being pressed, the red led should flash to indicate to the user that the button is being pressed. It was relatively easy to adapt the example code to provide this functionality but I ran into a problem where the red LED wasn't lighting up consistently but the green and blue LED's worked fine.

After a lot of head scratching and testing on breadboards, I found some information on the specific LED breakout board that I had and found out that it required resistors to bring the LED driving voltage down to 1.8 - 2.8v. This made sense as the green and blue LEDs required a driving voltage of 2.8 - 3.6V so the 3.3V supplied by the RP2040 microcontroller was below its maximum voltage range. At the time of writing this, the correct resistors are still being shipped so will be added to the LED breakout board in series such that the red led is supplied with a voltage less than or equal to 2.8V.



LED Module Info Showing Voltages

This should remedy the strange behaviour of the LED making it light up consistently. The current code will light up the blue LED when the electronics are powered, when the button is pressed, the red LED blinks to indicate a button press and holding down the power button makes the red LED blink until the microcontroller sends the signal to the soft power button to cut power to the electronics. A single press of the power button turns the electronics on such that it is easy to turn the device on but hard to accidentally turn off. This makes the likelihood of the device not being powered on when it is dropped from the helicopter as low as possible thus mitigating the chances of mistakes.

Below are some code snippets from the initial code I wrote.

```
void loop(void) {
  while (digitalRead(POWER_BUTTON) == LOW)
  {
    if (digitalRead(POWER_BUTTON) == LOW && powerPressedStartTime == 0)
    {
      //Debounce check
      delay(debounceDelay);
      if (digitalRead(POWER_BUTTON) == LOW)
      {
        #ifdef DEBUGGING
          Serial.println("User is pressing power button. Start timer.");
        #endif

        powerPressedStartTime = millis();
      }
    }
    else if (digitalRead(POWER_BUTTON) == LOW && powerPressedStartTime > 0)
    {
      //Debounce check
      delay(debounceDelay);
      if (digitalRead(POWER_BUTTON) == LOW)
      {
        if ((millis() - powerPressedStartTime) > 2000)
        {
          #ifdef DEBUGGING
            Serial.println("Time to power down!");
          #endif
          fastPowerDown();
        }
      }
    }
    else if (digitalRead(POWER_BUTTON) == HIGH && powerPressedStartTime > 0)
    {
      //Debounce check
      delay(debounceDelay);
      if (digitalRead(POWER_BUTTON) == HIGH)
      {
        #ifdef DEBUGGING
          Serial.print("Power button released after ms: ");
          Serial.println(millis() - powerPressedStartTime);
```

My Adapted Code Based on the Provided Examples

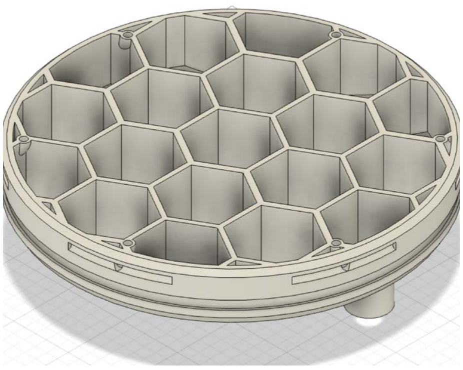
```
void setup(void) {
  //Power Switch Input Setup
  pinMode(POWER_BUTTON, INPUT_PULLUP);
  pinMode(FAST_OFF, INPUT_PULLUP);
  powerPressedStartTime = millis();
  while (digitalRead(POWER_BUTTON) == LOW)
  {
    //Wait for user to stop pressing button before
    delay(100);
    if (millis() - powerPressedStartTime > 500)
      break;
  }

  if (millis() - powerPressedStartTime < 500)
  {
    fastPowerDown();
  }

  // Display Red LED to indiacate system is on and not data logging
  pinMode(LED_RED, OUTPUT);
  digitalWrite(LED_RED, HIGH);
  powerPressedStartTime = 0; //Reset var to return to normal 'on' state

  //Turn on
```

My setup Function

TUBE DIVIDER

I planned to use sand as the mass that can be added and removed from the vehicle in order to adjust the descent rate of the vehicle. Therefore I would need a divider to separate the electronics and sand inside the body tube.

I designed and 3D printed divider that could be glued in to separate the two compartments. The divider is slightly domed in order to direct the force of the sand on top of it towards the walls of the body tube. This means the divider can be made of less material since most of the load on the structure is transferred so will be far stronger than a simple flat divider. In order to give this dome some strength, I opted to add a honeycomb mesh which also helped make the dome easier to 3D print. The small hole in the side is to mount an SMA extension cable such that another one can be connected to reach the LoRa antenna that will be in the nose cone.



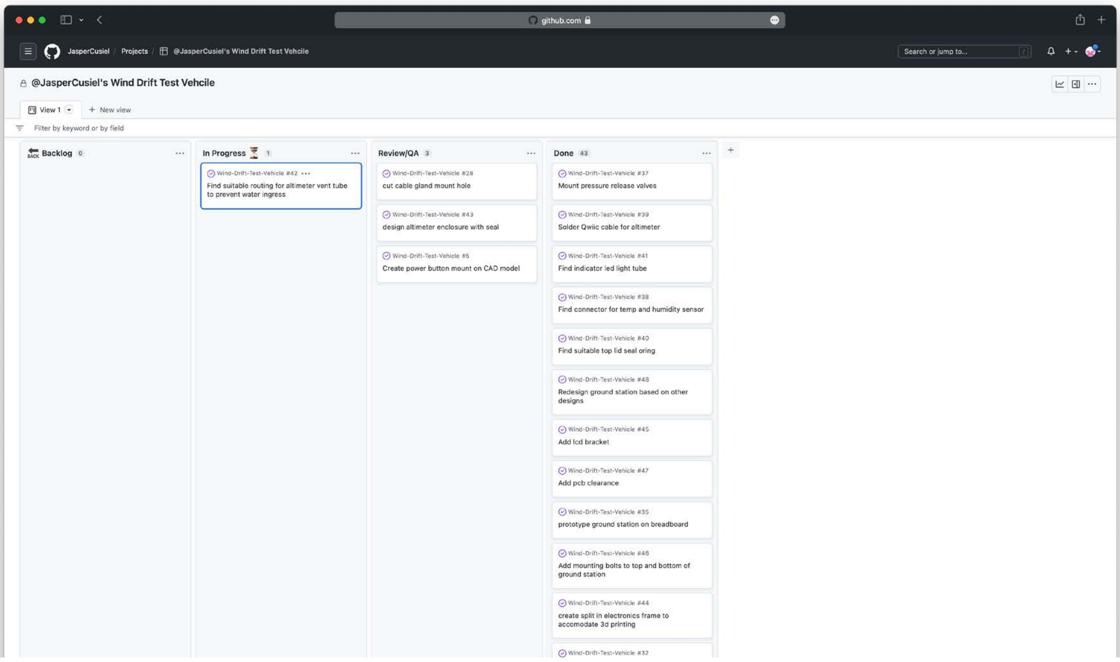
Divider Design on Fusion 360

Divider Domed Side With SMA extension Installed

Honey Comb Side

The side has a channel for a silicone tube to wrap around which helps seal as well as retain the divider. The small channels above the silicone tube channel have holes that allow glue to be injected such that the divider can be glued in. Lastly, both top and bottom edges are chamfered to allow a bead of caulking silicone to act as a backup seal.

PROJECT MANAGEMENT (GITHUB)



Github Project (Almost All Current Issues Closed)

To manage my progress throughout this project, I decided to use GitHubs' project management feature over something similar like Trello as I would be using Github anyway to upload my code. Therefore it made sense to use Githubs' ecosystem as it also integrated with Visual Studio Code, the IDE I used to write the code.

The general layout I followed was having four columns to manage issues, the backlog to hold new issues, an in-progress column to hold the issues I was currently working on, a review/QA column for verifying the fix for the issue, and a final column for completed issues.

While I tried to document as many issues as possible initially, I found that it became more time-consuming to document every little detail due to almost every aspect of the project being custom so changes were being made very frequently. For this reason, I decided to pivot to just recording major issues rather than breaking them down into very small pieces. I know this is not great practice but due to me not knowing every little detail as the project progressed, it became very difficult to plan ahead in detail.

Although, GitHub still proved to be very helpful for keeping track of the bigger picture and helped me to keep making progress on the project. Currently, the project is relatively empty as I need to start writing the main data logging code so more issues should continue to crop up.

CONCLUSION AND REFLECTION

Due to the scope of the project and the planning required to coordinate the testing of the vehicle with Rocket Lab, I feel that I did manage to complete a very large portion of the project and look forward to testing it out in the field in December 2022. I would say this project would definitely be competing to be one of the most complex projects I have built to date. I learnt a lot about integrating systems as well as gaining more depth of knowledge into various topics such as GPS, LoRa and C++ coding (was starting from scratch here). Combining this with knowledge from various projects I have built, experience from work and talking with experts, I was able to produce a very high-quality project.

After a lot of exploring different avenues to find an awesome, genuine project to work on, I only landed on this project for Rocket Lab in Term 2 which left a very small window of time to get this project underway. However, due to the size and scope of the project being much larger than initially anticipated, I was very pleased with the outcome to date. I love the idea of a much more challenging project that I thought would be genuinely interesting and engaging since it would lead me to want to work on the project in any spare time.

After some field testing and feedback to refine the design, I intend on building version two. While doing the research for the project I came across a lot of custom-made PCBs that consolidated many components onto a single board. It would be awesome to find a much smaller enclosure and build a PCB that had the GPS, LoRa, power switch and altimeter on a single PCB. This would just mean that I could buy all of the individual components and solder them onto this board. Thus the electronics could be much more tightly integrated and much smaller. This would allow the total physical size of the electronics and thus vehicle size to shrink and would make it more cost-effective to build multiple as you don't have to pay extra for the convenience of breakout prototyping boards. Another thing I would be interested in adding is an RTK (real-time kinematics) GPS module as these can achieve accuracy down to ± 1 cm which would increase the recorded data significantly. With those changes, I think version two of the vehicle would create a very polished and professional solution.

Overall this project was both challenging and interesting; I really enjoyed the process of creating a solution to a genuine problem and the learning experience that came with it. I definitely look forward to completing the project and working on a second version.

PHOTOS OF FINAL OUTCOME