Imperial College London

First Year Summer Project Report:

EEERover Design Project

Module: ELEC40006 Electronics Design Project 1

Summer 2022 - 2023

Samuel Khoo (02264291)

Date of Submission: *15th June 2023*

Word count: 9953

Examiner: *Dr Edward A. Stott*

Imperial College
London

Acknowledgements:

Dr Edward A. Stott, Course Lecturer: Senior Teaching Fellow

Mrs Esther Perea Borobio, Course Lecturer: Principal Teaching Fellow

Dr Zohaib Akhtar, Interim Presentation Examiner

All other lab assistants as well as undergraduate and postgraduate teaching assistants

Imperial College London

TABLE OF CONTENTS

INTRODUCTION **3**
- ABSTRACT 3
- BACKGROUND/SPEC 4
- PRODUCT DESIGN SPECIFICATION 6
- PROJECT MANAGEMENT 9
  - *COMMUNICATIONS* *9*
  - *GANTT CHART* *9*

AGE SENSOR **10**

MAGNETISM SENSOR **15**

NAME SENSOR **17**
- HARDWARE 17
- SOFTWARE 22

MOVEMENT **23**
- HARDWARE 23
  - *INITIAL PLANS & TESTING* *23*
  - *FINAL IDEAS* *25*
- SOFTWARE 26
  - *CONCEPT* *26*
  - *TESTING & APPLICATION* *28*
  - *FINAL CODE* *30*

WEB-APP **31**
- UI-DESIGN 31
- SOFTWARE IMPLEMENTATION 32

CHASSIS **35**
- ORIGINAL CHASSIS PLANS 35
  - *Design 1:* *35*
  - *Design 2:* *36*
- FINAL DESIGN 38

EVALUATION **39**
- FINANCES 39
- CONCLUSIONS 40
  - *Review* *40*
  - *Future Work/Developments* *40*
  - *What Could Be Done Differently* *40*

REFERENCES **41**

APPENDIX A: NAME CALCULATIONS **42**
- Resonant circuit: 42
- High pass filter: 42
- Low pass filters: 42

Imperial College
London

INTRODUCTION

ABSTRACT

This report will focus on explaining and demonstrating via results the development process of our web-app controlled rover that is designed to explore a flat plane and identify different aliens present. These aliens emit a combination of unique signals, including radio and infra-red waves. The purpose of this project was to allow us to implement our lab skills all together, whilst also gaining vital insight into project planning and management along with learning about how the industry tends to approach projects. Some of the key functional requirements were; ensuring that the different aliens could be properly detected, being able to effectively and swiftly manoeuvre around the environment, and most importantly receive commands and convey information through a WiFi based web-app.

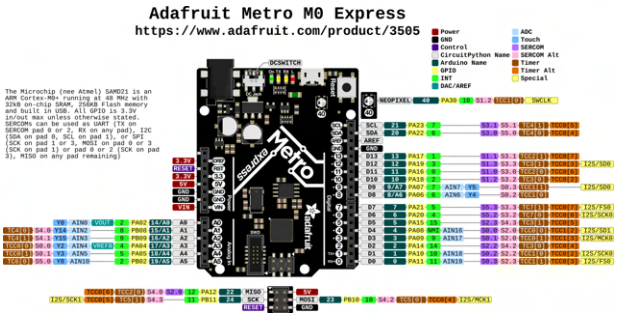
Upon introduction to the project, we held a first meeting where we split the development of the rover into numerous different sections, these were the three main sensors, software, and movement and chassis. The focus of the sections is shown below:

- Sensors - *Research the exact specification for the required criteria, brainstorm and design a schematic for said function. Implement using circuitry and complete testing to ensure efficiency and accuracy.*
- Software - *Develop using the Metro M0 Arduino and/or WiFi module. Ensure that the code runs and is bug free. Implement onto a full code for the final web-app and complete testing on said web-app to ensure reliability and functionality.*
- Movement - *Research and decide with members upon a suitable design/set-up. Work closely with the chassis team to achieve a functional and effective rover body. Develop upon the starter WiFI code to implement the web-app. Lead the software development once the design and hardware of the movement has been figured out.*
- Chassis - *Decide with the movement team upon a design that suits the sensor range and sizing. Then with the use of SolidWorks, create numerous CAD models, with a focus on motor and sensor mounting points, reducing wasted space and having an easily accessible layout.*

BACKGROUND/SPEC

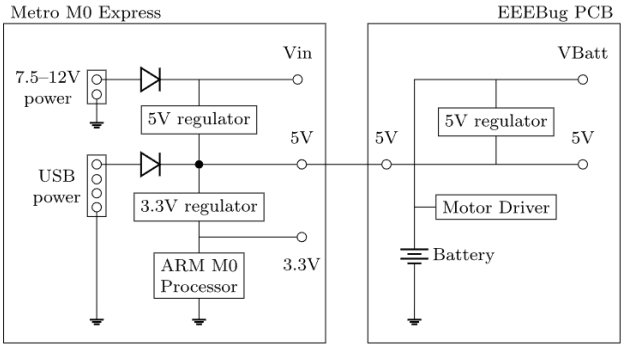
The table below shows the components which we were provided with, some of these components were for testing, while others were useful upgrades to our existing EEEBug rover which we had completed during the labs over the past two terms.

Qty	Description
1	Alien simulator
2	H-Bridge Motor Driver Module
2	Adafruit Metro M0 Microcontroller Module
2	Adafruit WINC1500 WiFi Shield
2†	Magnet

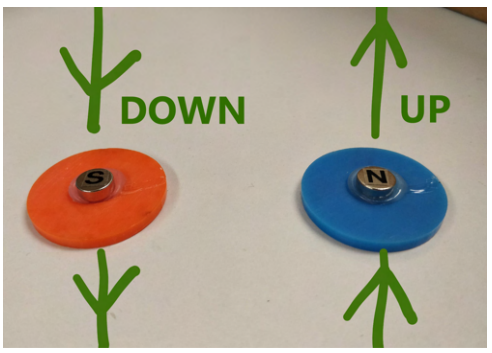


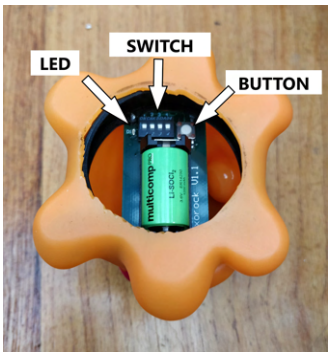
Alongside the table of given parts is the diagram for the Metro M0 Express Arduino microcontroller which we would be using. This board would be paired with an Adafruit WINC1500 WiFi Shield which connects to an existing WiFi network within the lab and allows for us to host our own web-app.

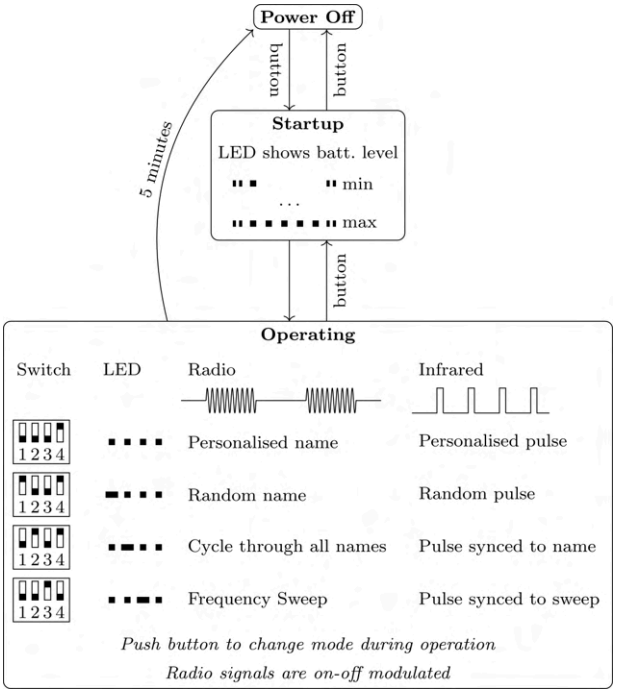
One of the key things to note is that Arduino pins 5,7 and 10 are used WiFi and cannot be used for other purposes. We were also advised that the Arduino layout given was relatively similar to that of the Orangepip Kona328 board which we had been previously using for our EEEBug. The diagram below was also provided to demonstrate how to connect the Metro Express board to our existing EEEBug PCB.



A picture of the alien simulator provided for testing is shown below. This alien module emits radio and infra-red waves which have to be decoded to read information about the alien. The alien configurations shown on the next page were used for testing the name and age sensors. We were also given a set of magnets to use for testing as the alien simulator is unable to generate a magnetic property of its own.

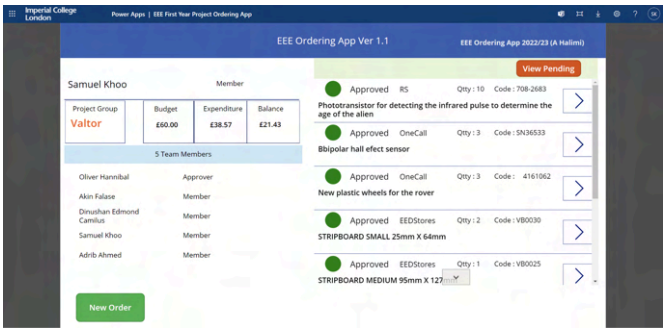






Different configuration of the alien simulator

We were also introduced to an ordering system (pictured below) and given a budget of £60. This was a non-negotiable budget and will be further broken down in the finance section. Additionally, we were advised about the limitations of our project. These included ordering safe items, ensuring that the components we ordered weren't surface mounted, and keeping track of our supplier.



Last but not least, we were also provided with hints for each of the sensors, a starter WiFi control code and a base design of the chassis for the EEEBug. These will be further expanded in their individual sections.

PRODUCT DESIGN SPECIFICATION

One of the key concepts for product development that we were introduced to were the 32 points for product design specification. These are 32 points that should be kept in mind during the development of a product. Some of the points mentioned are not relevant for our use case hence we have merged a few of them together.

1. Performance
    - Low latency communication and minimal lag
    - Ability to navigate around obstacles efficiently

- Precise detection of each type of alien

2. Environment
    - Able to get close enough to the alien to detect the alien
    - Flat plane environment hence smooth wheels without much thread can be used, reducing friction and increasing the top speed of the rover
3. Maintenance
    - Simple design that is easy to work on
    - Order multiple parts online in case of component failures
4. Life In Service & Product Life Span
    - Able to last the demo session and testing we perform
5. Target Product Cost
    - Budget of £60, £15 for emergency fund and £45 for spending
    - Divide the funds equally amongst all the sub groups and divisions
    - An approver is assigned to give the greenlight for all purchases to manage and limit if necessary the purchase of certain components, although he may not submit a purchase himself
    - Due to shipping costs, it may be more efficient to bulk buy certain components or find them all from the same site and/or supplier
6. Competition
    - More accurate detection compared to other rovers
    - More lightweight and compact
    - Faster and more agile
7. Shipping
    - Order components in advance to take into account shipping times
8. Packing
    - No loose wires, careful packaging to prevent damage during storage and prior to testing/demo
    - Clean design that isn't over complicated and difficult to reassemble in the event we need to transfer stuff across boards
9. Quantity
    - Purchase spare sensors in the event of failure of certain components
10. Manufacturing Facility
    - Make use of the resources available in the lab
    - We are only able to test the Wi-Fi and wireless communication related work in the lab where the EEERover Wi-Fi source is set up
11. Customer
    - Dr Ed Stott, specification is provided on GitHub
    - Need to ensure the rover meets the criteria and achieves the desired functionality
12. Size & Weight
    - Relatively small to be able to navigate around obstacles
    - Under 800 grams to prevent triggering of the weight sensor and ‘scaring away’ the aliens
    - Good centre of mass to prevent loss of stability
13. Materials
    - Make use of the materials readily available within the lab
    - 3D printing is an option but it can only be used for components of certain sizes, otherwise long printing times

- Materials shouldn't interfere with the function of the sensors (no magnetic materials as it can interfere with magnet sensor)
- Maximise the durability so that it can withstand testing and repositioning of components
- Parts that come into contact with heat (motors, heatsink, etc) should be able to withstand the maximum temperature

14. Aesthetics Appearance and Finish
    - This is a lower priority goal/target (as with tight time constraints the focus should be on functionality)
    - Ensure a compact design with no overhanging or very loose wires
    - Lightweight design to prevent exceeding weight limit
    - Clear organisation and distinction of different components/sensors to allow for easy maintenance when required
    - Distinguishable from other rovers, so that during control testing and the demo, we won't get confused with our rover and others
15. Ergonomics
    - Controls are intuitive and operate with an easy learning curve
    - The rover should be able to detect the signals from most angles
    - Hence sensors should be placed in an accessible position
16. Quality and Reliability
    - Sensors should be reliable and endure testing
    - The rover should have a decent build quality, sufficient to last through the testing and demo phase
    - Avoid last minute changes, but spare sensors and components should be ready on hand in the event that component failure occurs
17. Shelf-Life/Storage
    - Use of long lasting, non single use materials
    - Product should be able to maintain function even after a week of storage
18. Testing
    - Range of the sensors and range of the Wi-fi connection limitations
    - The accuracy and reliability of the sensors
    - Speed of transmission
    - Speed and agility of the rover
    - Durability and portability
    - Ensure that the logic from the programs written is correct
    - Ensure that the function of the sensors is as desired

19. Processes
    - Chassis redesign along with new wheels
    - Possibly consider higher power motors
    - Schematics for sensor circuitry and layout on a breadboard
    - Maximise the usage of the breadboard and design circuit in the most efficient way possible (no extra unnecessary circuitry)

20. Time Scale
    - Time taken for research, testing and production of the rover should be around one month
    - Divide and allocate the time accordingly to prevent any last minute rushed work
21. Safety
    - Prevent the use of lithium batteries as it can cause fires
    - Ensure that the wiring is done correctly and that the right procedures for wiring is followed
    - Clip wires properly and don't leave any sections of the wire exposed, i.e. stripped wire
    - Power off during storage and when changing components to prevent component failures of amplifiers, capacitors, etc.
    - If soldering is incorporated, we must follow the right procedure for it
22. Company Constraints
    - Budget provided
    - Availability of the materials on hand
    - Manufacturing time and machinery needed for it
    - Existing integrated circuits and availability of certain third party out-sourced components like sensors
23. Market Constraints
    - Availability of certain components
    - Limited to four suppliers listed by the EEE department
24. Patents, Literature and Product Data
    - Include the links for component specifications and manufacturers
    - Include links to online resources used during development and/or testing
25. Legal
    - All online resources should be referenced within the bibliography
    - No plagiarism and avoid anything which could lead to a copyright strike
26. Political and Social Implications (N/A)
27. Installation (N/A)
28. Documentation
    - Usage of a logbook to keep track of the development of the product
    - Explain the theory behind sensors and any code/circuits designed
    - Document the software and development accordingly
29. Disposal
    - Attempt to use environmentally friendly materials and reduce wastage
    - In the event of waste disposal ensure that they are disposed of accordingly (e.g. dead batteries in an electronic waste bin)

PROJECT MANAGEMENT

COMMUNICATIONS

To communicate and organise dates for meetings and progress updates we made use of WhatsApp and polls that could be made on WhatsApp. This would also allow us to get feedback for different questions and queries that each member had. Furthermore, we also conducted weekly meetings and recorded our progress in a logbook. These are both shown below:

Imperial College London

Meeting Log

Thursday, 18 May, 2023 2:22 PM

This section of our logbook focuses on our meeting log and keeps track of all notes and points from our meetings along with their dates.

18/5/2023 - Meeting 1
Within this meeting we decided our groups and split into teams for software and hardware, we also decided to split the workload apart focusing on 3 major sensors and movement + software. Each individual was assigned into a team and we began research for our respective sections. We also decided to keep track of our progress using a logbook and a Gantt Chart which can be found within our logbook.

25/5/2023 - Meeting 2
During our 2nd meeting members within our team all came together to discuss the progress which we had made so far within the first week. Many members of the hardware teams had completed research and begun testing on certain components with the name sensor team working on amplification of their signal and the age sensor team looking into purchasing different sensors and designing circuits for them.

Additionally within the movement and software department, the logic had been completed and we had decided to implement a basic 4 button set up with a rotate button and stop button. This could be further developed into a joystick if time persists. With regards to software implementation, there were difficulties regarding the connection of the Wi-Fi sensor and issues with the rover ignoring comments from the webapp, this will be further worked on within the 3rd week.

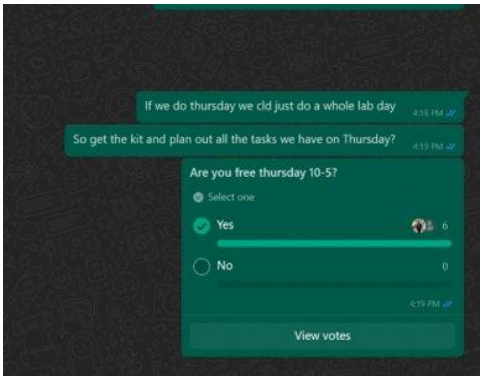
2/6/2023 - Meeting 3
In our 3rd meeting, we had all completed our respective sections regarding/surrounding the interim presentation. At this point of the project majority of the hardware was complete with some fine tuning of the sensors left. The focus was now mainly on the software and since the software for Age and Name was relatively similar, the 2 hardware groups decided to work closely together to solve and incorporate this. The movement department now focused on developing UI and assisting the other 2 groups in the development of software for all the sensors.

Our aim was to take the feedback from the interim presentation and make any modifications as a result. Besides this, we would aim to complete all the software for individual sensors prior before the 9th of June, this give us 1 week to combine and bring everything together, while also allowing us to begin the write ups for our specific components in preparation for the report due on the 15th of June.

9/6/2023 - Meeting 4
For our 4th meeting, we discussed the work beginning on the chassis of the rover and finalised a design which all the different departments/sections felt met their needs for their sensor/aspect of the rover. We had a large focus on the progress of our lab report which was due on the 15th of June. We had now all at least began brainstorming what to include within our lab report and some had even completed drafts. The aim over the week was to complete our draft of the lab report by the 12 or 13th of June. This would provide us with 2-3 days to receive feedback and make micro-changes to our report.

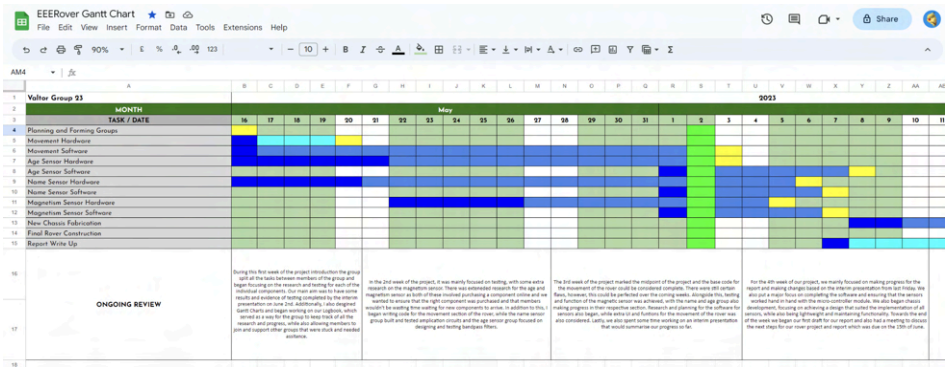
With regards to our progress for software and the final web-app, majority of the back-end code had been completed and just required connecting and finalisation. The front end of implementing the UI was still in the works and the aim was to complete it by the 15th of June.

(Upcoming)
12/6/2023 - Meeting 5



GANTT CHART

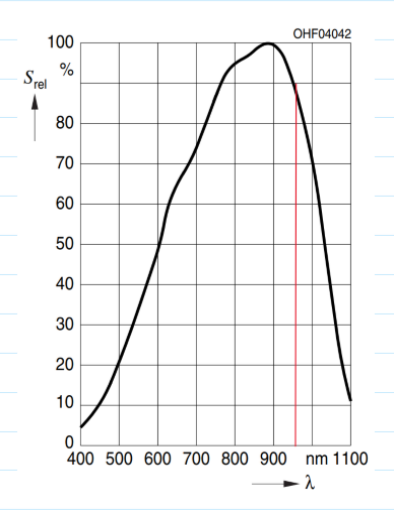
We also decided to make use of a gantt chart as we felt that it would provide us with a visual idea of the timeline and the remaining tasks while also allowing each subgroup of the rover to keep track of progress of other sections and provide extra help if needed. Overall, we felt that it served as a useful tool to enhance the coordination of the team. An image of the gantt chart used is shown below.



AGE SENSOR

From the specification given [1] we know that each alien emits an infrared pulse signal as a means of conveying its age. More specifically, the age of the alien is equal to the period in milliseconds multiplied by 100, giving the alien's age in years. Deciding upon the phototransistor that w ould be used to detect the signal was the initial step we took in order to determine the age.

The phototransistor we had previously used in our EEEBug was too sensitive to visible light and the arena's colour-changing hemisphere. The red-end of the light spectrum from the colour changing hemispheres were between 625-710 nm, whilst infrared has a wavelength of 780 nm - 1.4µm. Therefore a phototransistor that could distinguish between the two spectrums accurately was required. The graph below shows the spectral sensitivity of the phototransistor provided with our EEEBug; notably the wide range of sensitivity makes it difficult to determine the pulse from the alien.



We furthermore discovered several phototransistors that could have potentially served our purpose while researching for an ideal sensor but there were individual drawbacks for most of them which made it more challenging to select a suitable candidate. Each of them are addressed below:

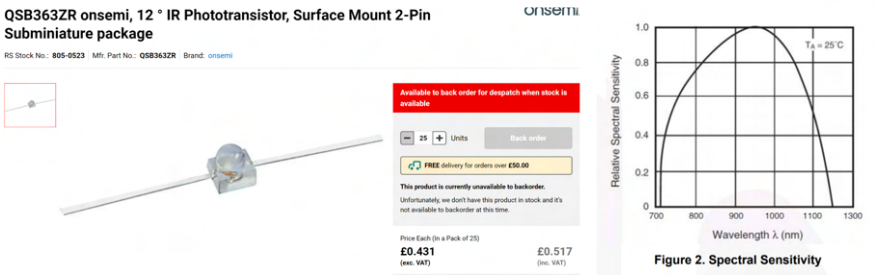
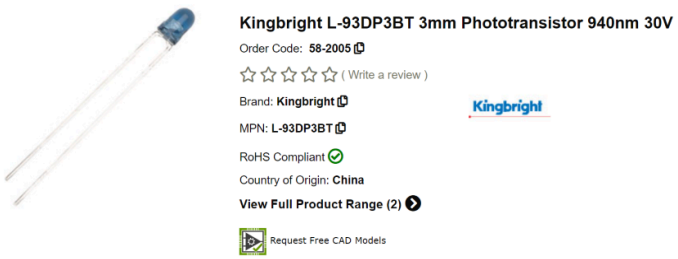
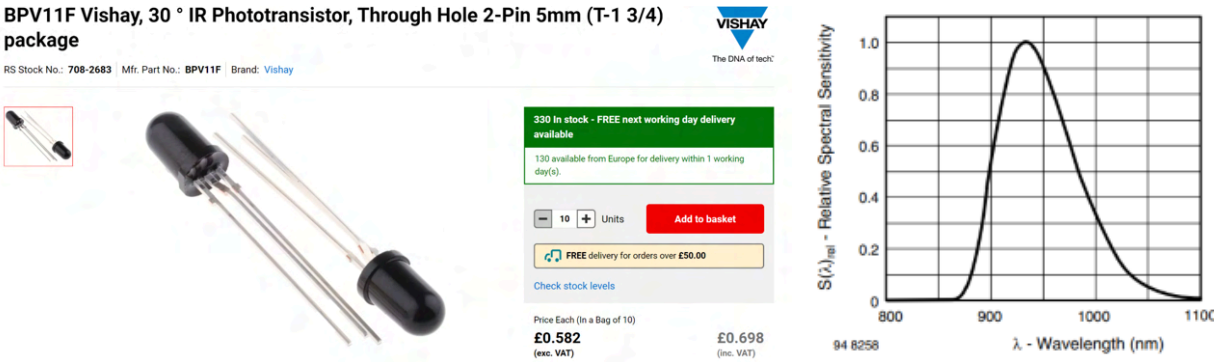


Figure 2. Spectral Sensitivity

The sensor above [5] is sensitive to 950 nm infrared light which would make it an ideal candidate. However, we had to eliminate it as it is a surface mounted device and part of our specification asked us to avoid the use of such sensors. [1]

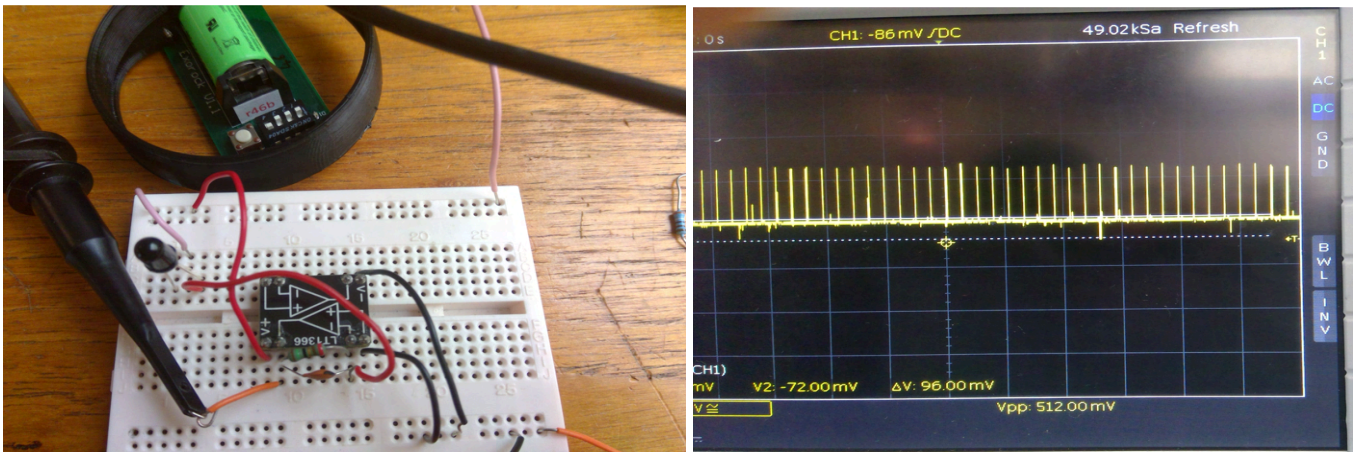


This type of sensor [6] suited our use case as it can detect only a narrow range of wavelengths, shielding it from interference by outside light sources, primarily visible ambient light. The only drawback is that it only responds to light at 940nm, leading us to eliminate this one too.

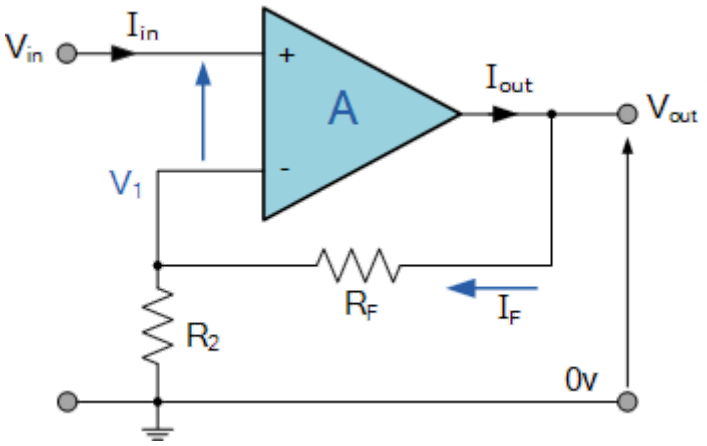


The phototransistor above [7] has a very short range of sensitivity from 900-980 nm. This means it would greatly reduce the interference of external visible light while achieving a sensitivity of 100% at the desired wavelength of 950nm. As a result we decided upon the use of this phototransistor.

The phototransistor outputs a current signal proportional to the light level received. In order to convert this to a voltage signal, we decided to use a circuit called a Trans-Impedandance Amplifier (TIA) [8], which is a configuration involving an operational amplifier and a feedback resistor which outputs a voltage lever directly proportional to the current received. This then outputs a square wave which we could then measure the period/frequency of. The images below show the initial square wave obtained and the Trans-Impedance Amplifier we constructed and used.



The presence of pulses shows the sensor is functional and it is also evidenced that the signal produced has very little noise. Besides, the volts/div are set to 200mV which means that the signal has an amplitude of about 0.5V. In order to increase this, we made the feedback resistor value larger. We changed it from 15k to 100k but this resulted in the absence of any signal. The 1k resistor had no effect while the 30k resistor gave no signal. Since we couldn't get the Trans-Impedance amplifier to output a larger signal and 0.2V is quite small, we decided to add a non-inverting amplifier (NIA) [9] to increase the amplitude of the signal.



gain NIA : $Y = \frac{R_f + R_2}{R_2} X$

$\frac{Y}{X} = 5$

$\frac{R_f + R_2}{R_2} = 5$

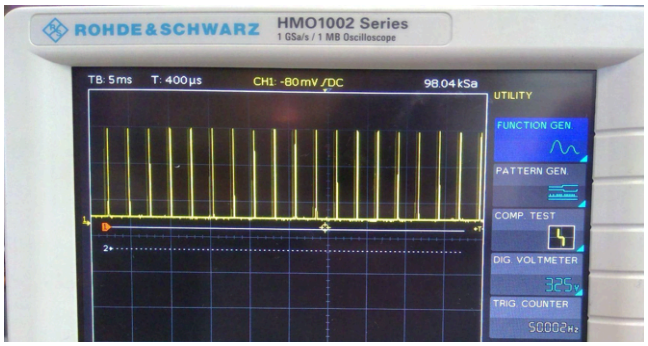
$R_f + R_2 = 5R_2$

$4R_2 = R_f$

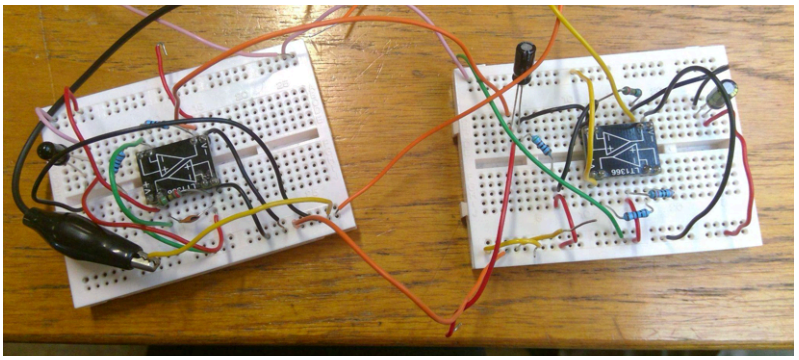
$R_2 = 1000\Omega$

$R_f = 4000\Omega$

Our original signal has a voltage of 0.2V. So, in order to get a 6V signal, we would need to amplify the signal by a factor of 5. The calculations for the resistance values are shown above.

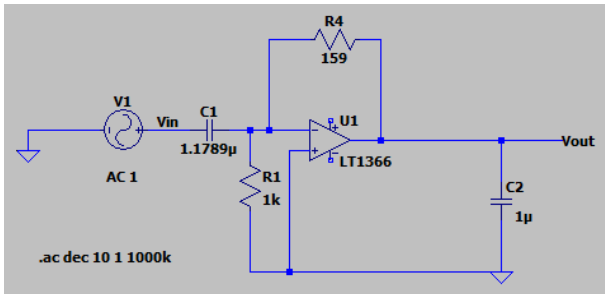


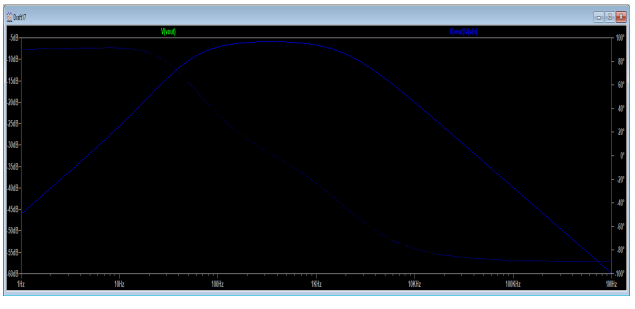
We then connected the output from the transimpedance amplifier to the input of the Non-Inverting Amplifier. This gave us pulses with an amplitude of 1V with the only issue being that the whole signal is offset by +0.5V, causing us to connect the output to the bandpass filter, which gave the following output:



(Note: The peak to peak voltage is still only 0.6V which was one of our key problems prior to the interim presentation).

We had also initially included a band-pass filter which would be used to detect frequencies between 135-1000Hz:

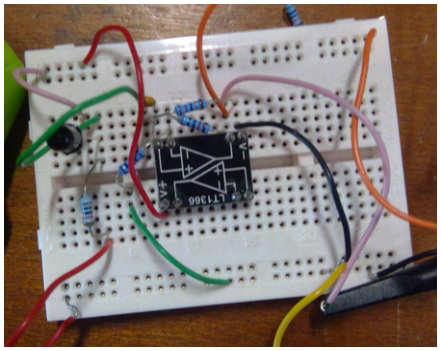


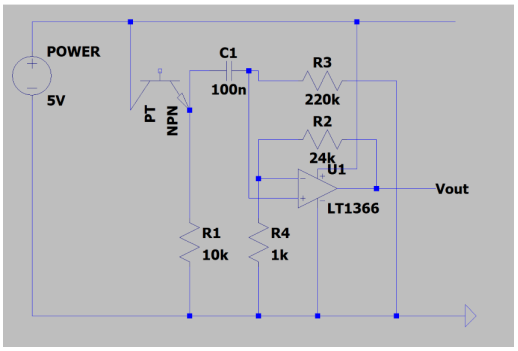


From the circuit above we can see that the left RC combination, the high pass filter, gives a lower bound of 135 Hz and the right RC combination, the low pass filter, gives the upper bound of 1000Hz. The operational amplifier enables the filter to isolate and amplify specific frequency ranges while attenuating other frequencies. As the graph shows, the gradient of the two curves are less steep which meant that it gave a broader range.

However, following the interim presentation, we were advised against using the band-pass filter and to look towards using the Arduino module to electronically filter out the signal/results from the initial circuit instead of using more complex hardware that could result in further confusion. Additionally, the circuit would produce a lower gain as there is a bandwidth limitation where the bandwidth of the band pass filter is smaller than that of the incoming signal, decreasing the gain of the circuit.

We also discovered that our output was being inhibited by the bandwidth and slew rate of the operational amplifier in the transimpedance amplifier, causing our output to fall below the predicted 1V. Following a consultation with lab assistants, we decided to simply develop the signal across a resistor before feeding this into the non-inverting amplifier. This redesigned circuit (pictured below) gave us a much better output signal with an amplitude of 3V at approximately 5 cm to the sides of the alien, while saturating at 5V upon being directly above. Furthermore, this allowed us to get the full range of the phototransistor as we were able to detect the signal at distances up to 30 cm from the alien as long as the sensor was positioned above it.





Our next step was to get this wave to show up in arduino so we could measure the time between pulses.[1] The approach we took uses the pulseIn function to find the on-time and off-time of the pulses seen. We then add these to get the period in microseconds, lastly we divide this value by 10 to give to age in years. The pulseIn function checks the duration between the two consecutive rising edge pulse so that period could be calculated. The code we used for this can be found below. The delay at the end was not necessary for the functioning of the code, but rather so we could read the values of the serial monitor as without it, there were many values coming through per second, so the monitor was moving too fast to read.

```
const int inputPin = 4;  // Pin connected to the pulse input
unsigned long period = 0;  // Variable to store the pulse period
unsigned long age;
void setup() {
  pinMode(4,INPUT);
  Serial.begin(9600);
}
void loop() {
  // Wait for the pulse to start (assuming it starts at LOW)
  while (digitalRead(inputPin) == HIGH) {}

  // Measure the pulse duration
  unsigned long ontime = pulseIn(inputPin, HIGH);
  unsigned long offtime = pulseIn(inputPin,LOW);
  unsigned long pulseDuration = ontime + offtime;

  // Calculate the period in microseconds
  period = pulseDuration;
  age = period / 10;
  // Print the period in microseconds
  Serial.print("Age ");
  Serial.print(age);
  Serial.println(" years");
 // Serial.print("Age is ", 100*period, );
  // Delay for a short interval
  delay(1000);
```

```
Age 150 years
Age 150 years
Age 150 years
Age 147 years
Age 150 years
Age 150 years
Age 150 years
Age 148 years
Age 150 years
Age 150 years
Age 149 years
Age 149 years
Age 150 years
```

We then decided to slightly modify the code by adding a for loop so that we take the pulse duration a hundred times then find the average to get a more accurate reading.

```
for(int i = 0; i < 100; i++)
{
  unsigned long ontime = pulseIn(4, HIGH);
  unsigned long offtime = pulseIn(4, LOW);
  unsigned long pulseDuration = ontime + offtime;
  sum = sum + pulseDuration;
}
```

Link for the age code–
https://github.com/SamuelKhoo2003/Valtor-Rover-Code-2023/blob/main/age_code_v1.ino

Once this was complete, we began work on implementing the full code and ensuring that this sensor works in conjunction with the other sensors. To attain the best results from our sensor we had to shield it from external light and place it top down facing the alien. This was discussed with the chassis teams and changes were made to achieve this.

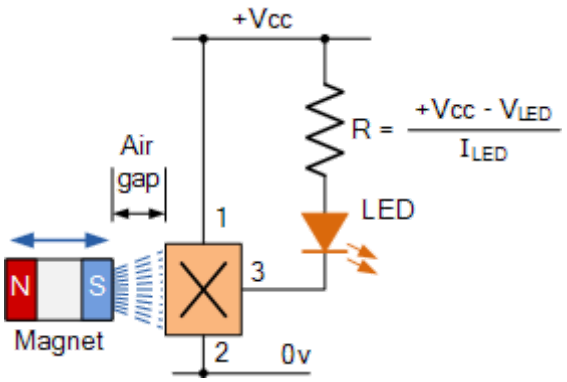
MAGNETISM SENSOR

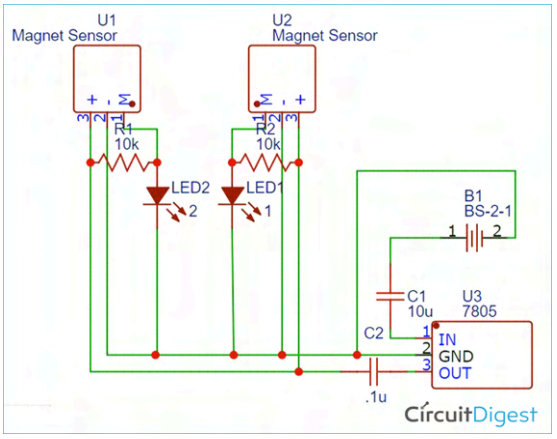
It was explained to us that our task was to determine whether each alien had a magnetic field and, if so, which direction (up or down) it was pointing in. We conducted online research to determine the most effective techniques for detecting a static magnetic field for this purpose. A moving coil is frequently used to accomplish this, but we decided that it would add too much weight and be an expensive solution. Therefore we instead decided to look for sensor that could detect static magnetic fields

From what we had learned in semiconductors, an application of the hall effect could be used to detect whether each alien is up or down. For this particular task, we chose the SS461C hall-effect sensor [10]. This is a bipolar sensor so it can distinguish between north and south poles by being high for one of the poles and then low for the other poles. The sensor could only detect the polarity of the magnet but not the strength but that wasn't required in the specification so we considered it suitable.

Sensor, Hall Effect, Bipolar, TO92 - SS461C

Following some research about an existing magnet detection circuit (pictured below), we placed our bipolar transistor in parallel with a 10k resistor. In doing so, we found that it gives a reading of 5V with one of the poles and 0V with the other. However, we struggled to differentiate between the north and south pole. To combat this, we decided to connect two hall-effect sensors with one connected with inverted ground and supply. The pull-down resistor ensures that when the Hall effect sensor is not actively driven high, the output signal is pulled down to the VCC voltage level. When the sensor detects a magnetic field, the sensor will make the output signal high, overriding the pull-down resistor and indicating the presence or absence of the detected event.





The image to the left [11] was our initial idea for the magnetic sensor, but due to the need to identify the polarity of either up or down, we decided to implement a variation of the circuit shown on the right [12]. Although LEDS are shown in the circuit diagrams above, these were eliminated from our design as we would instead use Arduino code and the analogread pins to receive signals to demonstrate our results.

Through experimenting with a range of values we found that the pull-down resistor helped to achieve a stable output voltage, preventing floating or undefined states when the sensor is not actively driven. By decreasing the resistance value of the pull-down resistor, you can increase the sensitivity of the Hall-effect sensor. A lower resistance value provides a stronger pull-down effect on the output voltage, making it more responsive to changes in the magnetic field and increasing the sensitivity of the magnetic sensor. However, driving the resistance value too low would draw more current, negatively affecting the performance of the sensor.

The main issue we faced was resetting the sensors after a reading, this was because the circuit doesn't reset after detecting the magnetic field. This is currently implemented by unplugging the supply wire so we needed to find a solution to this remotely. After reading through the manual for the Metro M0 Express board [13], we found out the pin A0 is both an analog input and output pin, hence it could supply anywhere between 0 and 3.3V. We also considered the digital PWM pins that weren't used for WiFi, but found that they supplied 0.6V at most.

The resulting code written using this knowledge is shown below:

```
void loop() {
  // Read the state of the input pin
  analogWrite(A0,255);
  int inputState1 = digitalRead(inputPin1);
  int inputState2 = digitalRead(inputPin2);


  // Check if the input pin is HIGH
  if (inputState1 == HIGH) {
    // Output a message
    Serial.println("Down");
  }
   else if (inputState2 == HIGH) {
     Serial.println("Up");
   }
    else {
      Serial.println("No magnetic field detected.");
    }

  // Delay for a short period
  analogWrite(A0,0);
  delay(1500);
}
```

```
No magnetic field detected.
No magnetic field detected.
No magnetic field detected.
No magnetic field detected.
South
South
No magnetic field detected.
No magnetic field detected.
South
North
North
No magnetic field detected.
South
No magnetic field detected.
No magnetic field detected.
```

Using the code above, we had a working circuit which tells us the presence and direction of the magnetic field of the alien using a series of if-else statements that depend on which sensor output is pulled-up. It is to be noted that there exists a delay within the circuit to allow the sensors to properly reset and to allow for the readings to be made.

Following this, we began testing the implementation of the magnet sensor and struggled with the range available with our existing hall-effect sensor. As a result, we purchased a different bipolar hall effect sensor whilst maintaining the same logic behind our code and approach. The new AH49E sensor [14] had higher sensitivity and greater range, better suiting our rovers' needs for alien magnetic field detection. We plan on integrating the sensor when it arrives prior to the demo, reusing the existing logic to complete this.

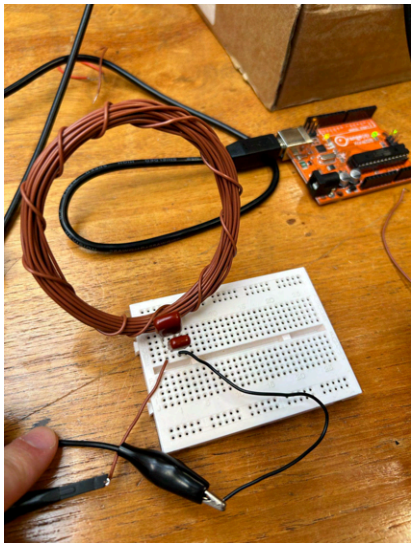
GitHub (magnet sensor code)–
https://github.com/SamuelKhoo2003/Valtor-Rover-Code-2023/blob/main/magnet_code_v1.ino
https://github.com/SamuelKhoo2003/Valtor-Rover-Code-2023/blob/main/magnet_code_v2.ino

NAME SENSOR

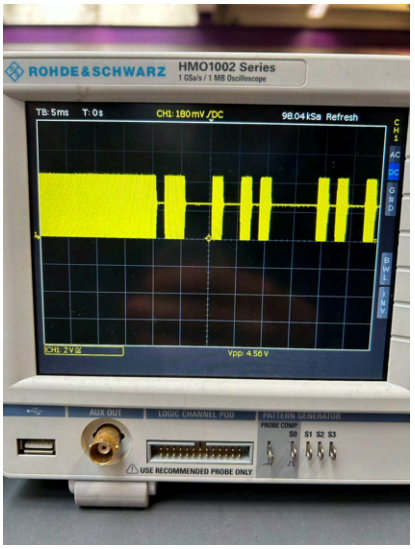
HARDWARE

To determine the alien's name, we had to figure out a way to convert the radio waves emitted from the PCB into an electrical signal that had the name encoded in it.

Reading the radio waves was achieved by coiling some insulated wire around a bottle to create an air-cored antenna and then placing it in parallel with a capacitor to give a tuned LC circuit at the correct resonant frequency - 61kHz. The calculations involved here were simple and involved nothing more than to equate the impedances of the inductor antenna and the capacitor. Using the LCR bridge, we found our initial antenna to have an inductance of 50µ, however we later rebuilt our inductor because the range it could read the radio waves from was not up to standard. To fix this, we added more turns to the coil and made the diameter bigger. The new inductance was 114µH, giving a parallel capacitance of 62nF, although the lab did not have a single capacitor with that value so we merely placed two in parallel which added to produce the correct capacitance and finally we were successful in detecting the radio wave signal which we confirmed by reading the results in our oscilloscope as you can see below:



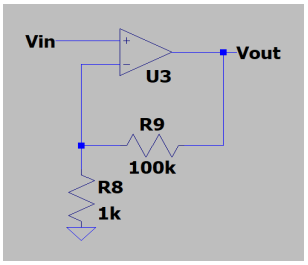
Resonating LC circuit



Detected signal in oscilloscope

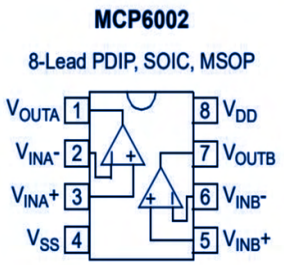
New resonating RLC circuit with new inductor

Given that we could detect the signal our next stage involved amplifying the received signal. One of the options we considered was using a simple non-inverting amplifier. Initially this configuration did not work because we didn't get the desired amplification and couldn't exactly pinpoint why, despite debugging the setup with our teammates. In retrospect, this issue was most likely caused by poor connections within the breadboard itself, something that we would only realise later in the project.



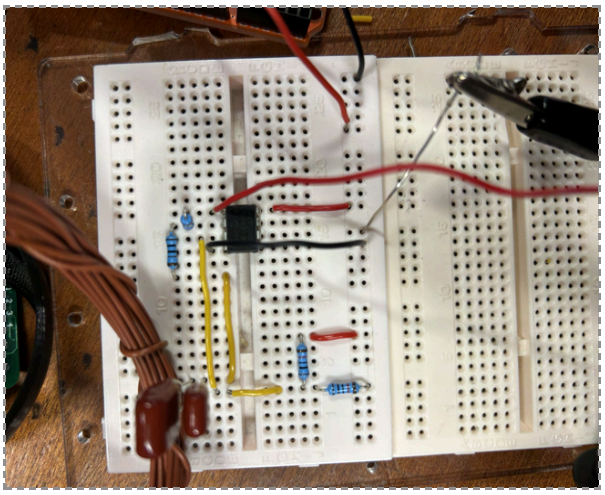
Initial non-inverting amplifier schematic

Unbeknownst to this, we decided to go with a differential amplifier. We used an *MCP6002 operational amplifier (containing two built-in op-amps)* which perfectly fulfilled our needs, providing high, stable gain and minimal noise - a characteristic of a differential configuration.

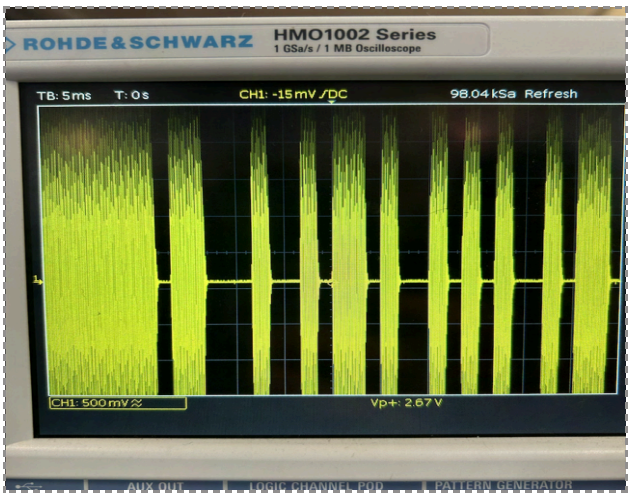


Pinout of MCP6002 opamp

Here below is the differential amplifier configuration in our breadboard and the new amplified signal with high gain as seen on the oscilloscope:



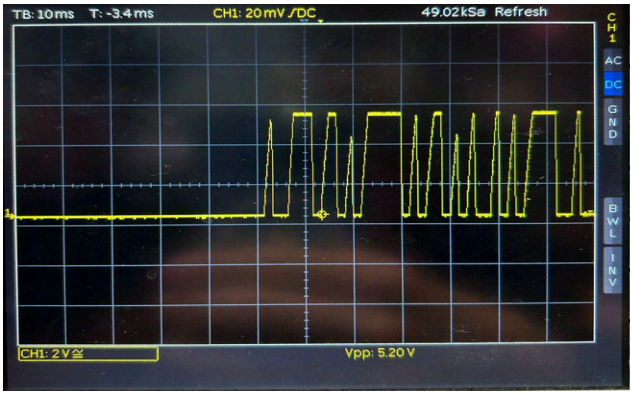
Differential amplifier circuit

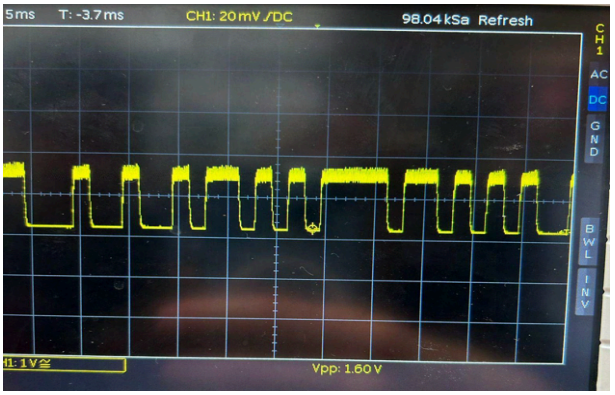


Amplified waveform on oscilloscope

Following the previous step, we had to demodulate the signal, for which we initially just used a diode. Here the diode works as an envelope detector. The process of demodulation using the diode depends on the fact that the amplitude variations of the carrier wave due to modulation can be captured by the diode. The diode only allows the positive half of the modulated carrier wave to pass while blocking the negative half. This results in a pulsating DC waveform that follows the envelope of the modulated carrier.

Even though we did get a signal after demodulation using a diode, the signal had too much noise. To address this issue, we first used a high pass filter configuration to remove the 2.5V DC component before adding the diode. The DC offset was applied to the signal received from the antenna using a potential divider – seen in the overall schematic below – before we passed the signal through the amplifier to prevent clipping of the negative waveform. The high frequency components of the signal were then removed by a low pass filter that was placed after the diode, extracting a much cleaner signal with minimal noise. Determining the cut-off frequency of the low-pass filter was straightforward given that the low and high frequency components of the signal were very far apart in value - 600Hz and 61kHz respectively. These calculations, along with every other for the name circuit can be viewed in Appendix A. In *Low_pass_1*, we found that an RC parallel lowpass filter was more effective than connecting the resistor in series with the diode, and the capacitor to ground, as the overall impedance at higher frequencies was lower compared to the standard configuration, due to the nature of adding parallel impedances producing a lower overall impedance, allowing for more effective attenuation of the carrier signal. At low frequencies the capacitor has a very high impedance, forcing the signal through to the next stage of the circuit. At higher frequencies the capacitor, coupled with the resistor in parallel, opened up a low impedance path to ground for the signal to pass through.

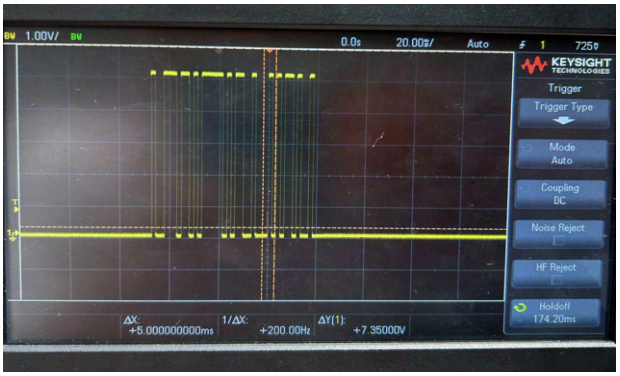


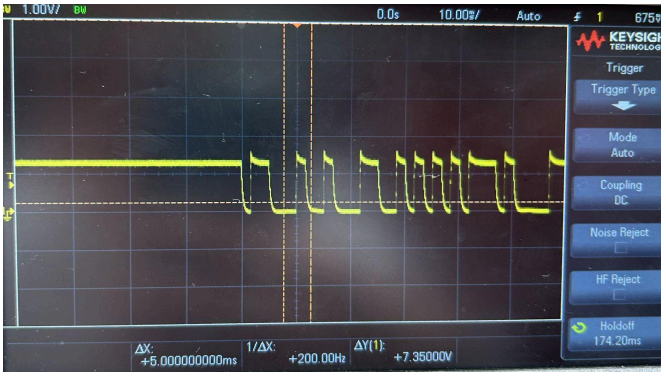


The spikes observed in the figure above on the right are due to the time constant of C4 (seen in the name schematic), calculated by RC, not being low enough. If it happens that the value of C4 is too high, it will discharge more slowly making the varying DC signal produced by *Low_pass_2* less reactive to the distance of the alien from the antenna. This means that it won't have enough time to discharge before the next cycle of modulation begins, giving this pattern of shark teeth. To fix this we made the cut-off frequency of *Low_pass_1* as low as possible - around 2kHz (see Appendix A) - whilst taking care not to get too close to 600Hz as this could prevent the information signal from being transmitted. So, taking into consideration the need for a small time constant to give a faster discharge rate, and a low pass filter that still operated in the correct band of frequencies, we chose these specific values for the resistor and capacitor seen in the schematic.

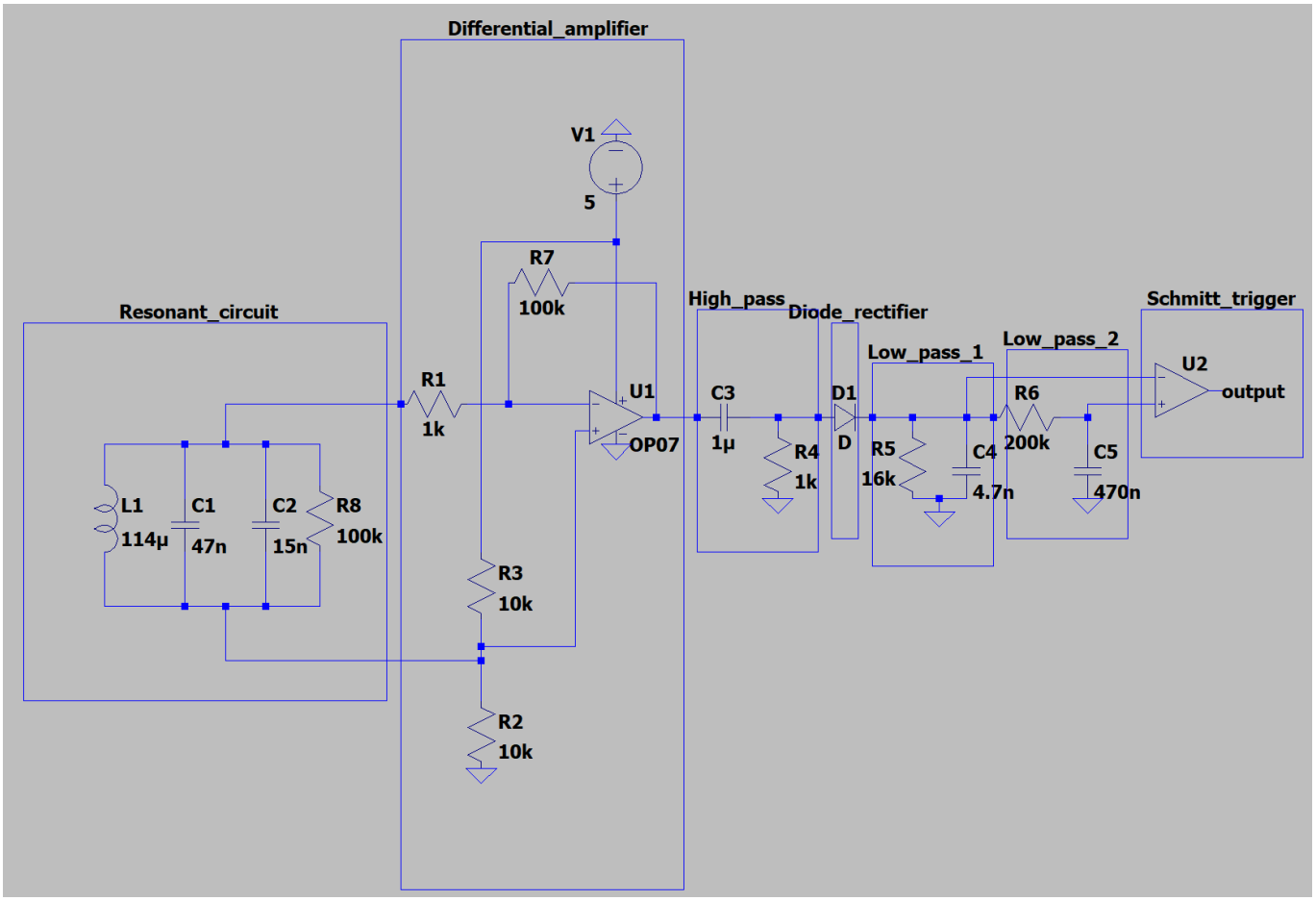
With the second lowpass filter, *Low_pass_2*, we wanted to achieve as low as possible a cut-off frequency in order to effectively eliminate all AC components of the signal and leave behind only a fluctuating DC signal, whilst simultaneously keeping a sensible time constant to create a more accurate real-time representation of the signal being received at any given distance which is important because the DC signal represents the average level of the whole signal. This plays to our advantage as it means we can set the right threshold for the schmitt trigger regardless of how far we hold the antenna from the alien. If we used a simple potential divider to set the threshold, the resulting binary signal would be more error-prone as the signal passed from *Low_pass_1* into the Schmitt trigger is not perfect at both peaks, meaning there is still some noise and the high pass filter is never perfect at attenuating the high frequency 61kHz component of the signal. By determining the average level of the signal at each particular distance, we can be sure that when the signal crosses that threshold, the desired switch from high to low happens, or vice versa.

Feeding the signal into a comparator, for which we used a Schmitt trigger, was the last and perhaps most generic stage of the circuit as we only had to feed two signals into each input rather than decide on a specific setup and choose values for its components. Normally, a Schmitt trigger is largely defined on its positive feedback configuration, however this was unnecessary because the threshold is not set by the output, but rather by one of the inputs which is fed into the positive terminal, which in this case is the average DC signal. When the voltage level of the signal crosses the DC voltage level of the threshold, the output binary signal will change.





We furthermore had an interesting observation while trying to measure the final signal with our oscilloscope. We observed that when our probe setting was set to 10x there was a spike in our waveform, which after research, we found was due to the high frequency noise of the signal. This phenomenon is observed because the resistance of the probe is set to 10MΩ when it is 10x and the capacitance is significantly dropped to around 30 pico farads therefore not attenuating high frequency noise. In contrast when it is set to 1x the capacitance is significantly larger and the resistance is smaller by a factor of 10 which attenuates the high frequency noise giving us a smoother signal.



Name circuit schematic

SOFTWARE

After completing the circuit and making sure that it produced the correct demodulated binary signal from which we could determine the name of the alien, we had to decode it into bytes. In the setup() function below, pinMode(3, INPUT) initialises pin 3 on the metroboard as the receiver pin for the binary signal and Serial.begin(600) initialises the serial communication between the metroboard and the arduino terminal at a specific baud rate of 600 bits per second, although the baud rate is not massively important and could also have been set at some other value.

```
void setup() {
  Serial.begin(600);
  pinMode(3, INPUT);
}
```

In order to write the principal code for reading the binary signal, we merely had to cycle through the bits and store them in an array that together formed the alien's name. In UART, each byte of data is sent as a digital sequence preceded by a start bit and terminated by a stop bit, hence it was simple to differentiate between the idle state of the signal and the desired section of the signal that we wanted to read. Given that the idle state is logic high, whenever a logic low is encountered, we know it is a start bit and that after 8 bits we will encounter a stop bit followed by another start bit, and so on, until we have completed this cycle four times - the exact character length of the alien's name. However, the binary data is transmitted in reverse order because the least significant bit is transmitted first, so each time we read 8 bits of data in the signal (one character), we shifted each bit left by i, where i is the iteration of the for loop, and stored the character in the complementary index of temp[4]. Temp[4] –the variable we created to store the alien's name– is an array of size 4. It is printed after the for loop below is repeated 4 times. The delay at the end of the for loop is in place to allow for a suitable transition from bit to bit before attempting to read each bit. It is important to note that the for loop is specifically written out four times in the actual code, as there are four bits, and this is just an example to showcase the code.

```
delayMicroseconds(2000000 / 600);
for (int i = 0; i < 8; i++) {
  temp[1] |= (!digitalRead(3) << i);
  delayMicroseconds(1000000 / 600);
}
```

```
char temp[4] = {0, 0, 0, 0};
```

Link for name code–
https://github.com/SamuelKhoo2003/Valtor-Rover-Code-2023/blob/main/name_code_v1.ino

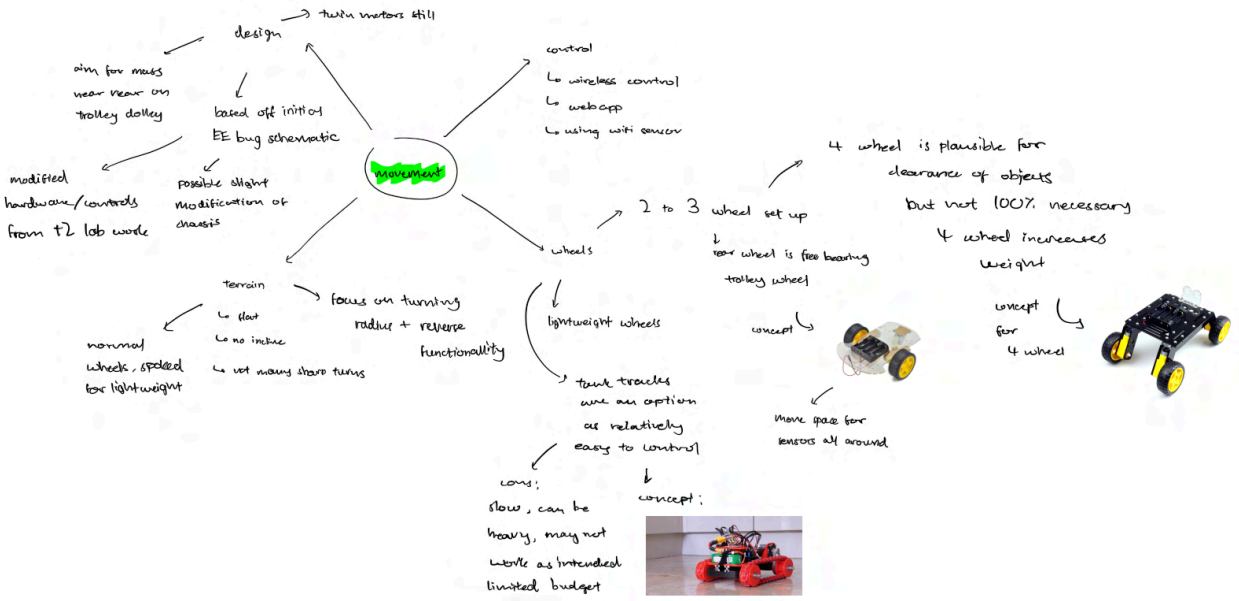
Imperial College London

MOVEMENT

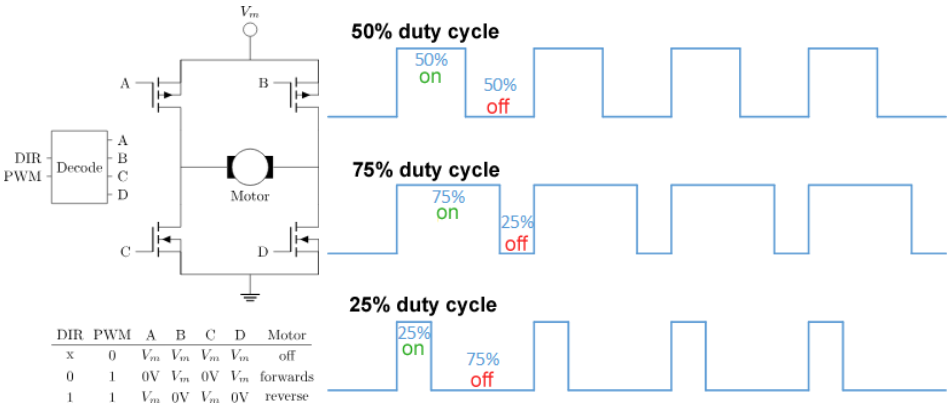
HARDWARE

INITIAL PLANS & TESTING

During the early stages of our project and discussion we had to decide between different movement styles and then choose the approach that best suited our use case. Besides this, we also had to consider the hardware and time frame available for these ideas. To manage and record our ideas and gain a wider perspective of all the possible concepts, we decided to make use of a brainstorming mind map to visualise the different ideas we had for the movement aspect of the rover. This is shown below:



We also looked further into the H-Bridge MOSFET module which had been provided to us. This H-Bridge was designed to work with our existing EEEBug PCB as a plug and play module. We had previously used the 6V DC motors along with the EEEBug PCB and had implemented separate pins to control the left and right motor. This new H-Bridge allows for the left and right motors to be controlled independently with two digital inputs for each side. These digital inputs are DIR and PWM/EN, and would be decoded and used to control four MOSFETs. The inputs to the DIR and PWM for the H-Bridge module can be directly connected to the digital output pins of the Metro M0 Express board given. The analogWrite() function in the Arduino API is used to generate a PWM input that dictates the speed/power output of the motor. [2]



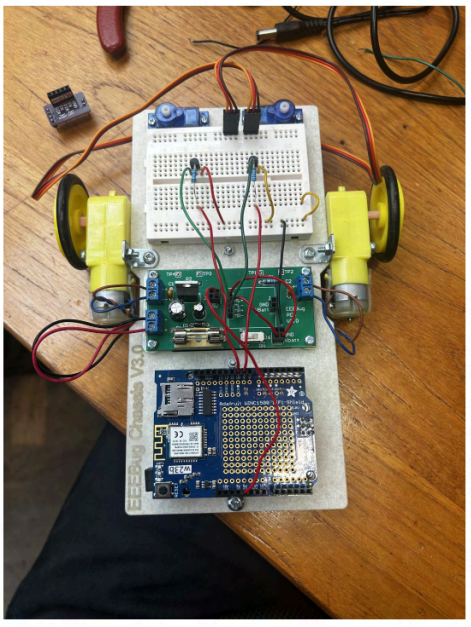
DIR	PWM	A	B	C	D	Motor
x	0	V_m	V_m	V_m	V_m	off
0	1	0V	V_m	0V	V_m	forwards
1	1	V_m	0V	V_m	0V	reverse

The 2 images above show the H-Bridge module, along with a duty cycle that acts as a PWM input.

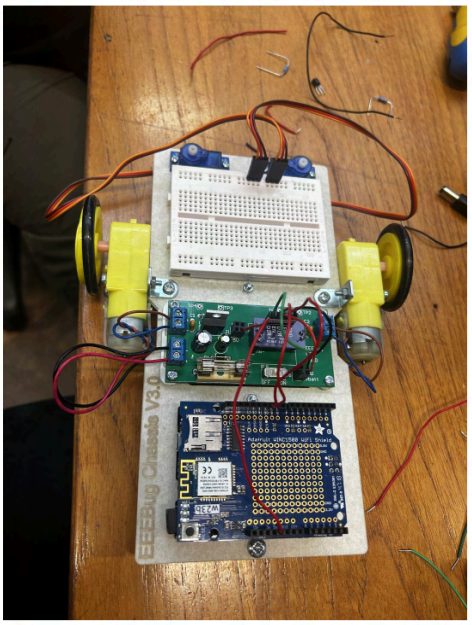
The first steps taken to modify the existing hardware was to experiment with the H-Bridge module and implement it into our existing chassis. This was done within the first week and allowed for the team to garner a good idea of how the rover would operate. As a group we had also decided to stick to a two wheeled design due to the fact that a four wheel setup would increase the weight and power consumption and could potentially add unnecessary turning complexity without using mecanum wheels. Having a four wheel/caterpillar track setup would also require a larger chassis that might prove more difficult to manoeuvre effectively. Although the two wheel design was slower in terms of speed and lacked the angular mobility of that of a mecanum four wheel setup, we felt that the pros far outweigh the cons, while also allowing us to apply prior knowledge much more easily and begin designing the code without needing to fabricate a chassis prior. This would allow us to design our chassis around the sensors instead of the drivetrain.

The before and after of our EEEBug with the H-Bridge module applied is shown below:

Before:



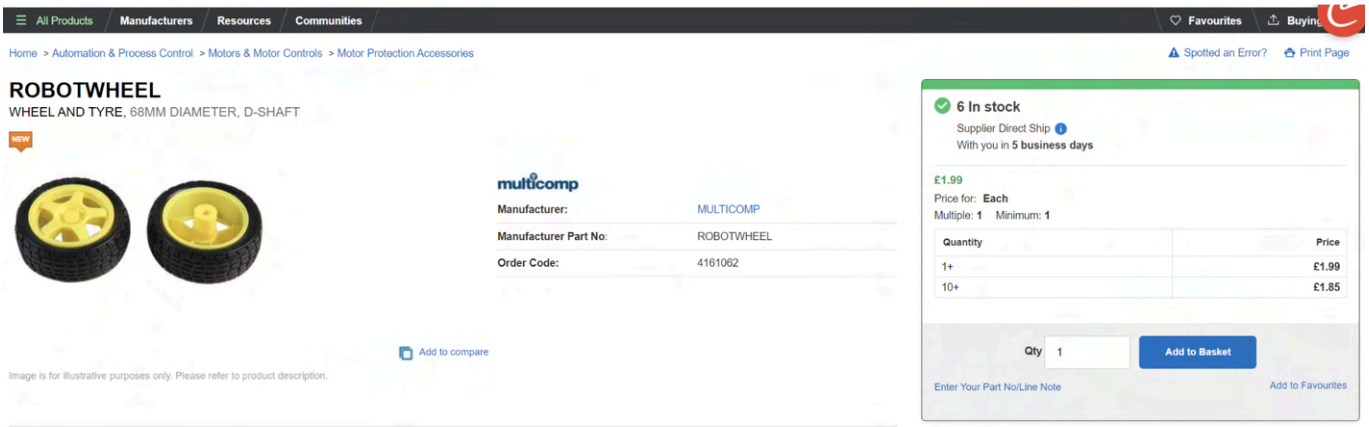
After:



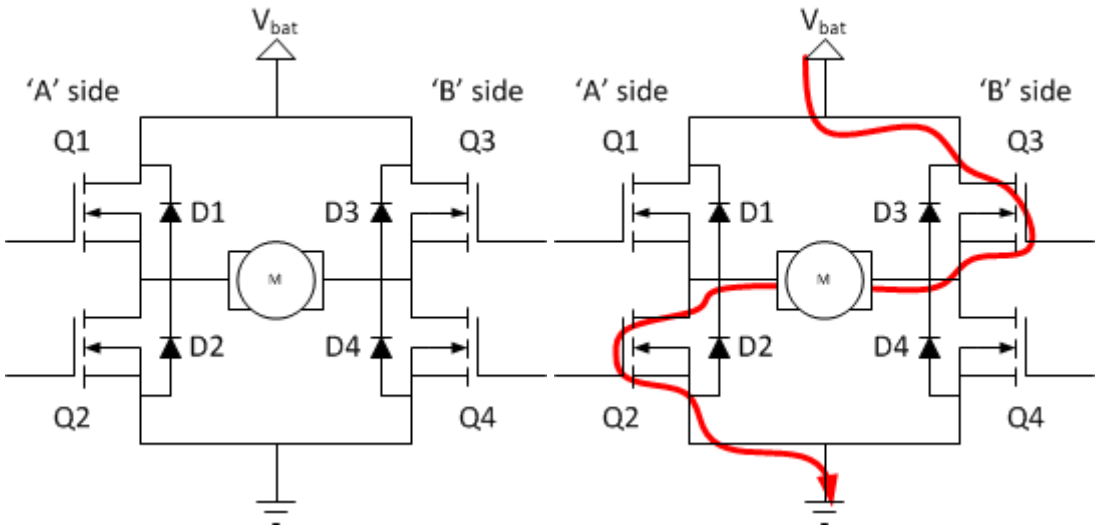
Notably controlling the motors no longer makes use of the breadboard hence freeing up space for the sensors to be implemented.

FINAL IDEAS

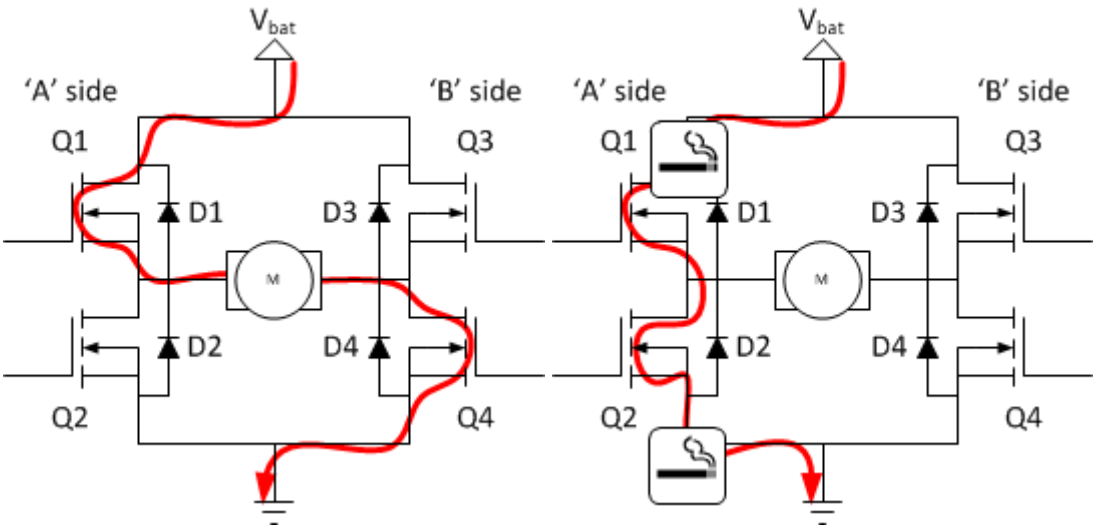
Prior to the interim presentation, we had solidified our final design as a two wheeled rover following the original EEEBug design but with modifications to reduce the wasted space and to include the sensors. Additionally, we also decided to buy new wheels that had a larger surface area and more grip to prevent sliding in the arena and allow for faster movement. They can be seen below:



Further research of the H-Bridge module allowed us to thoroughly understand its function, along with how it was able to switch between the forward and reverse drive modes. The top of the bridge would be connected to a battery/power source while the bottom end would be connected to ground. All of the switching elements can be independently turned on and off. In the figure shown below [4], we will see that Q1 - Q4 (which are normally bi-polar transistors or MOSFETS) are paired with diodes D1-D4. When Q1 and Q3 are turned on the left lead of the motor will be connected to the power source while the right lead is connected to ground, hence driving the motor forward. Alternatively when Q2 and Q3 are turned on, the right lead of the motor will be connected to the power source and the left lead is connected to ground, causing the motor to engage into reverse gear. It is notable that Q1 and Q2 (or Q3 and Q4) cannot be both closed at the same time as this will create a low-resistance path between the power and ground, short circuiting the power supply and possibly damaging the H-Bridge and other components in the process. [4]



One side of the H-Bridge *H-Bridge in reverse configuration*



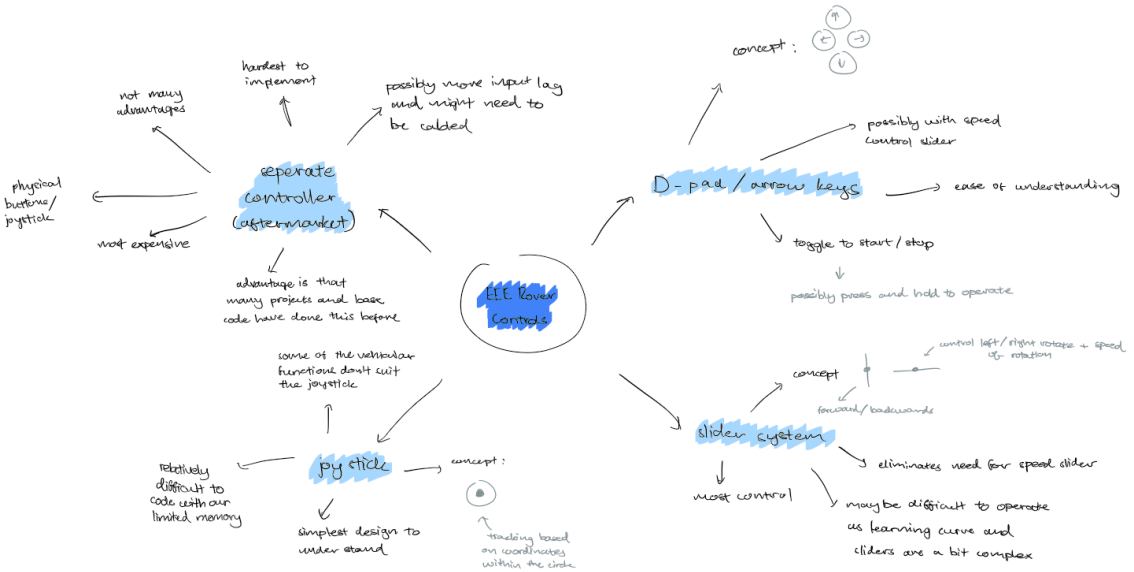
H-Bridge in forward configuration

Short-circuit of power supply

SOFTWARE

CONCEPT

Similar to our approach for the development and planning stage for hardware, we once again began our software concept around how we wanted the layout of the controls to be like and discussed it amongst members and seniors. This is shown below:



A summary of the break down of initial ideas is:

- Separate controller - *Expensive and not many advantages, may need to include wires for the controller to be connected. However it is the only physical option and allows for possibly a separate more efficient web-app/terminal to read off the detected names, age and etc.*

- Joystick - *Arguably the option with the least learning curve for operation and numerous implementations of this had been done previously hence it wouldn't be too difficult to learn to code. But, some of the vehicular functions of our rover hardware don't exactly suit the joystick setup and the code for the joystick would still be long/difficult to debug.*
- Sliders - *This would eliminate the need for a speed slider while also providing the most macro control for the rover. Besides it would be a unique solution that would still be effective in terms of controls. However, it may be difficult to operate as it would involve 2 separate sliders and there would also be a learning curve.*
- D-pad/Arrow Keys - *We could possibly incorporate a speed control slider that would improve the functionality of the D-pad. This was the easiest to understand and was one of the best options towards tackling the controls for the rover. The downsides of this control set up is the fact that it doesn't provide as much macro controllability as the other options and may take longer to operate/reach a destination and it can only either rotate or go backwards and forwards.*

After considering this, we decided to commit to the D-Pad/arrow key design for controls as we felt it best suited our use case while also having a low learning curve hence the operation of the rover wouldn't be limited to any members within the group.

We had also been provided with some starter code to begin work on the WiFi web system. This starter code mainly just focused on an LED ON and OFF switch, while also providing more insight into the server side of code and setting up our own webpage using http function and the WiFiWebServer.h library. This starter code included key server related commands which would be kept the same throughout the entire project and served as reference code for our future code. These commands are shown below:

```
void handleNotFound()
{
  String message = F("File Not Found\n\n");
  message += F("URI: ");
  message += server.uri();
  message += F("\nMethod: ");
  message += (server.method() == HTTP_GET) ? F("GET") : F("POST");
  message += F("\nArguments: ");
  message += server.args();
  message += F("\n");
  for (uint8_t i = 0; i < server.args(); i++)
  {
    message += " " + server.argName(i) + ": " + server.arg(i) + "\n";
  }
  server.send(404, F("text/plain"), message);
}

void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(LED_BUILTIN, 0);

  Serial.begin(9600);

  //Wait 10s for the serial connection before proceeding
  //This ensures you can see messages from startup() on the monitor
  //Remove this for faster startup when the USB host isn't attached
  while (!Serial && millis() < 10000);

  Serial.println(F("\nStarting Web Server"));

  //Check WiFi shield is present
  if (WiFi.status() == WL_NO_SHIELD)
  {
    Serial.println(F("WiFi shield not present"));
    while (true);
  }
```

```
  //Configure the static IP address if group number is set
  if (groupNumber)
    WiFi.config(IPAddress(192,168,0,groupNumber+1));

  // attempt to connect to WiFi network
  Serial.print(F("Connecting to WPA SSID: "));
  Serial.println(ssid);
  while (WiFi.begin(ssid, pass) != WL_CONNECTED)
  {
    delay(500);
    Serial.print('.');
  }

  //Register the callbacks to respond to HTTP requests
  server.on(F("/"), handleRoot);
  server.on(F("/on"), ledON);
  server.on(F("/off"), ledOFF);

  server.onNotFound(handleNotFound);

  server.begin();

  Serial.print(F("HTTP server started @ "));
  Serial.println(static_cast<IPAddress>(WiFi.localIP()));
}

//Call the server polling function in the main loop
void loop()
{
  server.handleClient();
}
```

[3]

From this code we are able to understand that the Metro M0 Express module sets up a WiFi communication using HTTP, the board will connect to the existing EEERover network within the lab and act as a web server, hosting a web page on a specific IP that is related to the group number given (in our case this was 23). When searching up this page, it would return a web page that could be used to wireless control the motors of the EEERover and convey information from the sensors.

It was also important to note that this code works only for the EEERover network within the lab as the EEERover WiFi network allowed for unique IP addresses to be hosted as web pages. When attempting to run this outside of the EEERover network it failed to load up a web page.

Based on this, our initial idea was to develop upon the starter code and add more functions and reorganise the existing front-end of the website to suit our D-pad control style.

TESTING & APPLICATION

Following this the first stage for testing was ensuring that we could drive both motors backwards and forwards, this would focus on altering the LED ON and OFF functions to include motor drive functions that would allow for this desired movement.

```
// Drive motors forward and acknowledge
void driveForward()
{
  if (driveState == INITIALIZING) {
    driveState = FORWARD;
    digitalWrite(7, LOW);
    digitalWrite(6, LOW);
    analogWrite(8, 255);
    analogWrite(3, 255);
    updateDriveState();
  }
}
// Drive motors off and acknowledge
void driveReverse()
{
  if (driveState == INITIALIZING) {
    driveState = REVERSE;
    digitalWrite(7, HIGH);
    digitalWrite(6, HIGH);
    analogWrite(8, 255);
    analogWrite(3, 255);
    updateDriveState();
  }
}
void driveOff()
{
  if (driveState == INITIALIZING) {
    driveState = OFF;
    digitalWrite(7, LOW);
    digitalWrite(6, LOW);
    analogWrite(8, 0);
    analogWrite(3, 0);
    updateDriveState();
  }
}
```

This code proved to be functional and we had started to thoroughly understand how to implement the H-Bridge successfully. Moving on from this we brainstormed and began a draft for the final code for the movement of the rover. The brainstorming process can be seen below:

Planning and Ideas For This Code:
Here are the tables and planning which I completed for this code (this is mainly correlated with hardware and the logic of the H-bridge).

pins 5, 7, 10 used for Wi-Fi

I chose to use pins 0, 1, 2, 3 to control the motors / movement

	gnd	–	GND
left	en	–	0
	dir	–	1
right	en	–	2
	dir	–	3

dir R	en R	dir L	en L	
1	255	1	255	} forward
0	255	0	255	} reverse
X	0	X	0	

The first draft we created for this code faced numerous challenges with an inability to compile properly and facing difficulties when processing the front end. After a long period of debugging, we eventually realised that this was due to the limit of code being 250 lines and my front end being too complex, hence the web page would time out instead of loading. Nevertheless the functions written were useful and would be applicable for the final code.

```
void updateDriveState()
{
  String stateText;
  switch (driveState) {
    case STOP:
      stateText = "Stopped";
      digitalWrite(LED_BUILTIN, 0);
      break;
    case FORWARD:
      stateText = "Forward";
      digitalWrite(LED_BUILTIN, 1);
      break;
    case REVERSE:
      stateText = "Reverse";
      digitalWrite(LED_BUILTIN, 1);
      break;
    case LEFT:
      stateText = "Left";
      digitalWrite(LED_BUILTIN, 1);
      break;
    case RIGHT:
      stateText = "Right";
      digitalWrite(LED_BUILTIN, 1);
      break;
  }
  server.send(200, F("text/plain"), stateText);
}
```

```
// Drive motors forward and acknowledge
void driveForward(){
  driveState = FORWARD;
  digitalWrite(DIR_right, HIGH);
  digitalWrite(DIR_left, HIGH);
  analogWrite(EN_right, 255);
  analogWrite(EN_left, 255);
  updateDriveState();
}
// Drive motors off and acknowledge
void driveReverse(){
  driveState = REVERSE;
  digitalWrite(DIR_right, LOW);
  digitalWrite(DIR_left, LOW);
  analogWrite(EN_right, 255);
  analogWrite(EN_left, 255);
  updateDriveState();
}

void driveStop(){
  driveState = STOP;
  analogWrite(EN_right, 0);
  analogWrite(EN_left, 0);
  updateDriveState();
}

void driveLeft(){
  driveState = LEFT;
  digitalWrite(DIR_right, HIGH);
  digitalWrite(DIR_left, HIGH);
  analogWrite(EN_right, 0);
  analogWrite(EN_left, 255);
  updateDriveState();
}
```

This code wasn't wrong however it was far too lengthy causing the web page to time out. Besides we also realised that having a stop button made the operation of the rover complex. Following discussions with Adrib and Oliver we decided that a hold to operate button style would be favoured for the operation of the rover. One of the highlights of the code was the LED status to visualise whether the problem was the hardware or the code. This proved useful for debugging and was kept throughout the project.

FINAL CODE

This final code was heavily based on the interim code, which hasn't been shown here due to the strong similarity it has with the final code. The 2 major changes made to the interim code was reducing the "magic numbers" which is when integers are chosen with no meaning behind them, by explaining the reasoning behind pin numbers this would make the code much simpler to read and allow for the other sections to understand the existing code much better. Lastly we took advice from Dr Zohaib and modified the turning style of the rover, rotating both wheels at the same time in opposite directions as this would increase the speed of rotation and reduce the turning radius of the rover. Snippets of this code are shown below and the full code can be found in the GitHub link.

Link to code–
https://github.com/SamuelKhoo2003/Valtor-Rover-Code-2023/blob/main/movement_code_v1.ino

```
const int dir_left = 1;
const int dir_right = 3;
const int en_left = 0;
const int en_right = 2;
```

```
void d_forward()
{
  digitalWrite(LED_BUILTIN,1);
  digitalWrite(dir_right, 1);
  digitalWrite(dir_left, 1);
  analogWrite(en_left, 255);
  analogWrite(en_right, 255);
  server.send(200, F("text/plain"), F("FORWARD"));
}
//Switch LED off and acknowledge
void d_reverse()
{
  digitalWrite(LED_BUILTIN,1);
  digitalWrite(dir_right, 0);
  digitalWrite(dir_left, 0);
  analogWrite(en_left, 255);
  analogWrite(en_right, 255);
  server.send(200, F("text/plain"), F("REVERSE"));
}
void d_stop()
{
  digitalWrite(LED_BUILTIN, 0);
  digitalWrite(dir_right, 0);
  digitalWrite(dir_left, 0);
  analogWrite(en_right, 0);
  analogWrite(en_left, 0);
  server.send(200, F("text/plain"), F("STOP"));
}
```

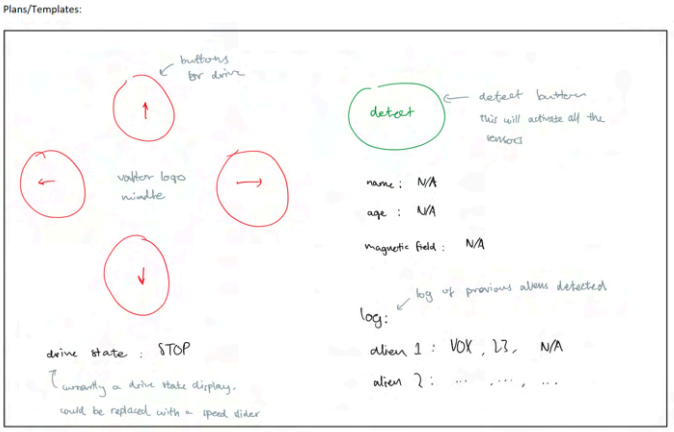
```
// after the interim presentation, I took the feedback to program
// this would reduce the turning radius and speed up turning
void d_right()
{
  digitalWrite(LED_BUILTIN, 1);
  digitalWrite(dir_right, 0);
  digitalWrite(dir_left, 1);
  analogWrite(en_right, 255);
  analogWrite(en_left, 255);
  server.send(200, F("text/plain"), F("RIGHT"));
}
void d_left()
{
  digitalWrite(LED_BUILTIN, 1);
  digitalWrite(dir_right, 1);
  digitalWrite(dir_left, 0);
  analogWrite(en_right, 255);
  analogWrite(en_left, 255);
  server.send(200, F("text/plain"), F("LEFT"));
}
```

Imperial College London

WEB-APP

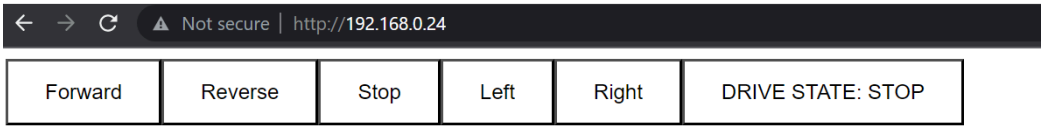
UI-DESIGN

In terms of UI, we needed a simple system that was functional in controlling the rover, while also being able to accurately display the information from the sensors. Together we came up with the following concept for the web-page UI:

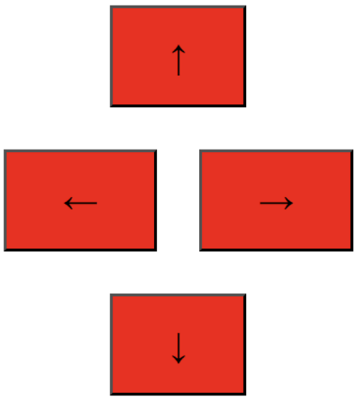


The first version of our UI only included test buttons to ensure that the functions desired were functional, this was because we prioritised the functionality of our code before considering the aesthetic aspect of the web-page, this is shown in the following iterations for controls below:

Development:

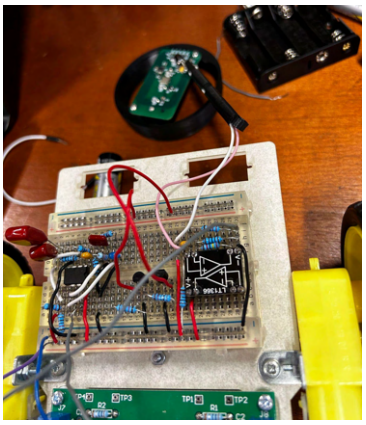


Final:



SOFTWARE IMPLEMENTATION

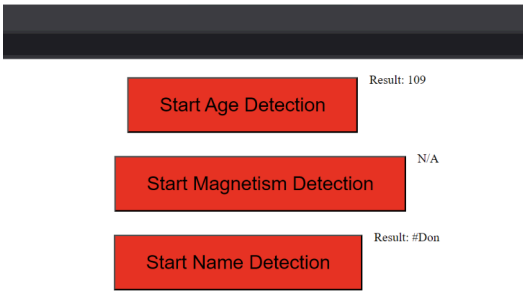
Upon bringing the code for multiple sensors together along with movement, we first focused on combining all the sensors on a single bread-board. This was done to allow for full testing of all the sensors together and to simulate a design that would mirror our final rover. The code written for all the individual sensors was analysed and pins which were recurring or used for WiFi were changed, this is visible below:



```
const char ssid[] = "EEERover";
const char pass[] = "exhibition";
const int groupNumber = 23; // Set your gr
const int age_pin = 4;
const int magnet_input_pin1 = 12;
const int magnet_input_pin2 = 13;
const int name_pin = 8;
const int dir_left = 1;
const int dir_right = 3;
const int en_left = 0;
const int en_right = 2;
```

Following this, I created a basic front-end which allowed the user to read the results from the sensors and turn on and off the sensors. I also paired the code with many serial print functions and code breaks to allow for a swift debugging process. For the most part, I kept the server related code unchanged, adding extra functions and working heavily on the front end to call the right functions. Upon testing with the alien simulator, we were able to confirm that the name and age detect functions correctly displayed their values on the web-page. This was supported via cross-checking with the oscilloscope. The images below show this in progress:

GitHub (all sensors)–
https://github.com/SamuelKhoo2003/Valtor-Rover-Code-2023/blob/main/full_sensors_code_v1.ino




```
void handleDetect() {
  String type = server.arg("type");

  if (type == "age") {
    detect_age();
  } else if (type == "magnetism") {
    detect_magnetism();
  } else if (type == "name") {
    detect_name();
  } else {
    server.send(400, "text/plain", "Invalid detect type");
  }
}
```

Alongside this, we also looked towards developing the existing movement code to include control via pressing the arrow keys or WASD keys on a laptop. This was done by including the Keyboard.h library to recognise certain keys and modifying the front end with a new set of functions called eventListener(), this would look to receive input from the WASD and arrow keys, called the drive functions without the need for mouse commands. However, the webpage would time out due to the length of the code. After discussing with fellow peers, we discovered that a solution to this was by splitting the front end to multiple different sections, loading each one at a time. This would prevent the website from timing out, while achieving the desired function.

GitHub (new movement)–
https://github.com/SamuelKhoo2003/Valtor-Rover-Code-2023/blob/main/new_movement_code_v1.ino

```
void handleRoot()
{
  server.setContentLength(CONTENT_LENGTH_UNKNOWN);
  server.send(200, "text/html", "");
  server.sendContent(webpage0);
  server.sendContent(webpage1);
  server.sendContent(webpage2);
  server.client().stop();
}
```

The image to the left shows the loading of webpages in sections. This helped to prevent the website from crashing. The image below showcases the front end now with the eventListener function and with a clear splitting of sections. In this case, it has been split as styling components, followed by buttons and lastly functions which are called.

```
//Webpage to return when root is requested
const char webpage0[] =
"<html><head><style>.container {display: flex;"
"flex-direction: column;align-items: center;}"
".row {display: flex;justify-content: center;}"
".btn {background-color: red;padding: 20px 40px;font-size: 25px;margin: 15px;}"
".logo {max-width: 300px;height: auto;margin: 20px;}"
".btn:hover {background: #eee;}"
"</style></head><body>";

const char webpage1[] =
"<div class=\"container\"><div class=\"row\">"
"<button class=\"btn\" onmousedown=\"sendCommand('forward')\" onmouseup=\"sendCommand('stop')\" onmouseleave=\"sendCommand('stop')\">▲</button>"
"</div><div class=\"row\"><button class=\"btn\" onmousedown=\"sendCommand('left')\" onmouseup=\"sendCommand('stop')\" onmouseleave=\"sendCommand('stop')\">◀</button>"
"<img class=\"logo\" src=\"file:///C:/Users/samue/Desktop/EEE%20Rover%20Project/finalinterimcode/Screenshot%202023-06-01%20164712.png\" alt=\"Logo\">"
"<button class=\"btn\" onmousedown=\"sendCommand('right')\" onmouseup=\"sendCommand('stop')\" onmouseleave=\"sendCommand('stop')\">▶</button>"
"</div><div class=\"row\">"
"<button class=\"btn\" onmousedown=\"sendCommand('reverse')\" onmouseup=\"sendCommand('stop')\" onmouseleave=\"sendCommand('stop')\">▼</button>"
"</div><br>DRIVE STATE: <span id=\"state\">STOP</span></div>";

const char webpage2[] =
"<script>var x = new XMLHttpRequest();x.onreadystatechange = function() {if (this.readyState === 4 && this.status === 200) {document.getElementById(\"state\").innerHTML = this.responseText;}};"
"function sendCommand(command) {x.open(\"GET\", \"/\" + command);x.send();}"
"document.addEventListener(\"keydown\", function(event) {if (event.key === \"w\" || event.key === \"ArrowUp\") {sendCommand(\"forward\");} else if (event.key === \"s\" || event.key === \"ArrowDown\") {sendC
"document.addEventListener(\"keyup\", function(event) {"
"if (event.key === \"w\" ||event.key === \"s\" ||event.key === \"a\" ||event.key === \"d\" ||event.key === \"ArrowUp\" ||event.key === \"ArrowDown\" ||event.key === \"ArrowLeft\" ||event.key === \"ArrowRigh
"</script></body></html>";
```

This new application was successful and we managed to gain control of the rover using physical laptop keys. This greatly increased the controllability of the rover and allowed for multiple sources of input, allowing for other ways to control the rover in the event where one of the inputs failed.

Lastly, we combined the movement and sensors sections together in a single arduino code file, we continued to make use of the sectioning method to split our front end apart. This helped prevent the website from timing out and allowed us to achieve full functionality of the code. Snippets of this code along with the combination of the D-pad and detect sensor buttons are shown below:


```
//Register the callbacks to respond to HTTP requests
server.on(F("/"), handleRoot);
server.on(F("/forward"), d_forward);
server.on(F("/reverse"), d_reverse);
server.on(F("/stop"), d_stop);
server.on(F("/right"), d_right);
server.on(F("/left"), d_left);
server.on(F("/detect"), handleDetect);
server.onNotFound(handleNotFound);
```

GitHub (draft of final code)–
https://github.com/SamuelKhoo2003/Valtor-Rover-Code-2023/blob/main/rover_final_v1.ino

During the testing phase of this front end, we found that the readings were occasionally slow and could time out, besides the movement had a notable degree of input lag. This, alongside redesigning the front end to make it more user friendly, are the next steps we plan to take before the demo.

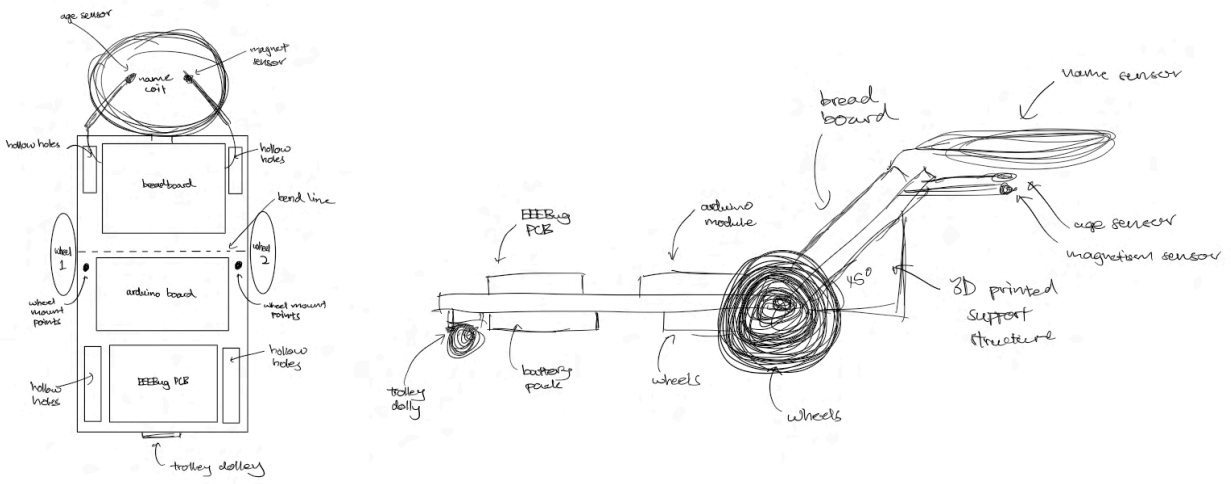
CHASSIS

ORIGINAL CHASSIS PLANS

Following the decision to retain the two wheel set-up, we tested the range of the sensors, initially running with the idea that for optimum efficiency of the rover we had to place the sensors as close as possible to the aliens. This meant placing the sensors above the alien for more consistent, reliable results which led us to consider the following designs.

Design 1:

The first design we considered was an inclined surface at the front end of the rover with the possibility of having a trailing wheel to provide stability and help distribute the weight. The sensors would have a direct view of the top of the alien to maximise their ability to take readings. The sensors would be placed perpendicular to the alien above it which would supposedly give us the best chance of capturing the information from within it. Notably this design considered the arrangement of the Arduino board and battery pack along the chassis to balance the weight of the rover and prevent it from tipping over. On the other hand, this is not important if the arm is 3D printed because it is extremely lightweight relative to the rest of the vehicle – as is the antenna and age and magnetism sensor. A concept of this design is shown below.



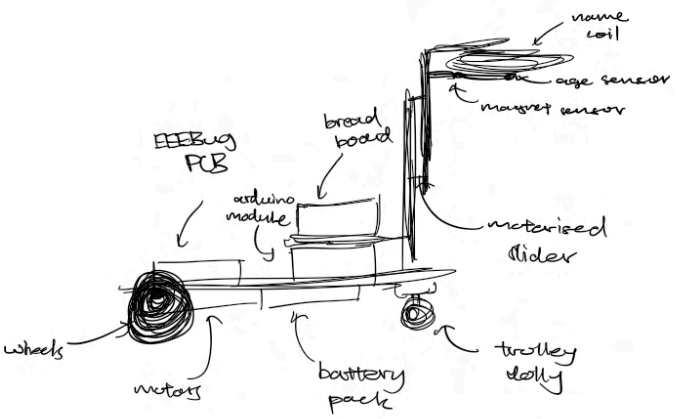
Advantages	Disadvantages
Covering the alien would reduce the opportunities for other rovers/groups to take readings.	Relies on the rover reversing away from the alien, hence increasing the time needed to move between aliens.
The positioning of the sensors allows for the most accuracy.	Having the sharp inclined ramp may prove difficult to produce and might have implications on the structural integrity of the rover.
This arm is quick and easy to 3D print, so more time can be spent testing several versions of it, producing a more optimised final product.	The ramp angle may make it difficult to mount wires or components onto the board; this makes it harder for us to work on our rover.

Design 2:

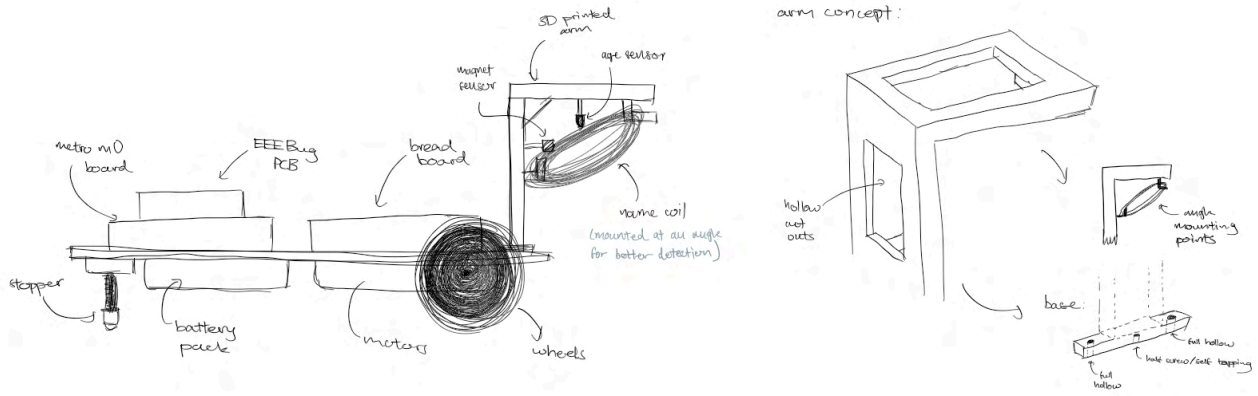
The second design focused on the concept of using an arm to hover over the alien to take the readings. We had thought about the possibility of making the arm motorised due to the different heights of the aliens, however it was decided to scrap this plan as the antenna didn't have great range when placed directly above the alien, even when touching its head, so a mechanical arm would have been futile in this regard. The drawing below is a rough sketch of this solution. One key thing to note is how the breadboard and arduino module are layered on top of each other, a design choice that came to life in the final design because of the significant space it saved and the weight it removed from the rover.

Advantages	Disadvantages
Motorised arm would allow for the sensors to be as close to the alien as necessary regardless of the alien's height.	More important to focus on the core parts of the project and improve the sensors to eliminate the need for a motorised arm in the first place. Difficult to justify the need for a motorised arm which would be time-consuming to develop.
Having the breadboard and arduino board layered over each other reduces the weight and size of the rover, increasing the manoeuvrability in the process.	By layering the components it may be more difficult to work on and maintain, but overall the PCB is small enough to leave sufficient access to the metroboard.
Having a shorter chassis could allow us to rotate on the spot more effectively instead of reversing away from the alien in the demo.	Having an arm too far out in front of the rover can compromise the centre of mass of the rover.

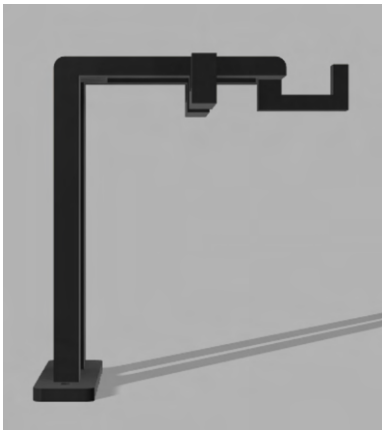
Motorised arm:



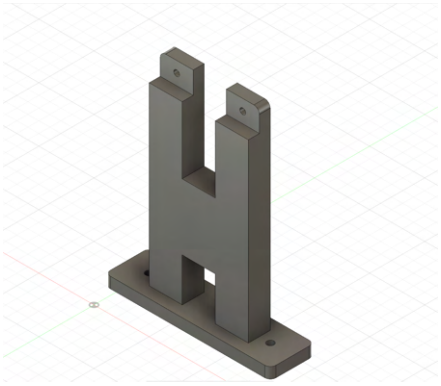
Non Motorised arm:

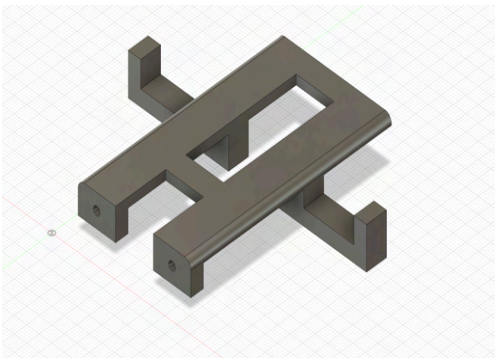


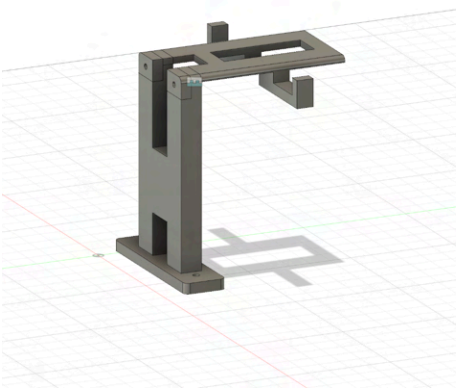
Arm V1:



This was the first design of the arm that came to life in Fusion 360, but quickly after printing it became clear that we had made a few errors. The screw holes were not spaced correctly according to the laser cut chassis, but this would be easily fixable. The main issue was that the bridge joining both pillars at the top got in the way of the age and magnetism sensor so we had to remove it. This led us to design a second arm in CAD, as shown below.





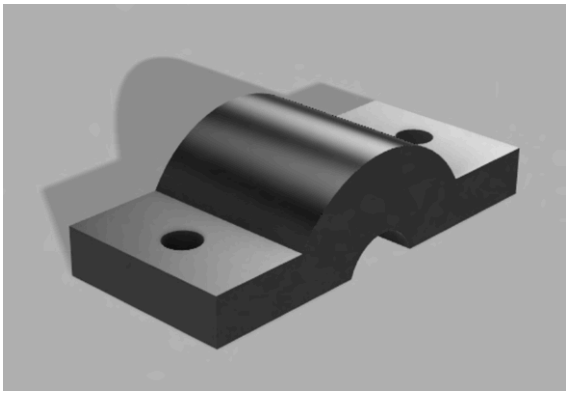


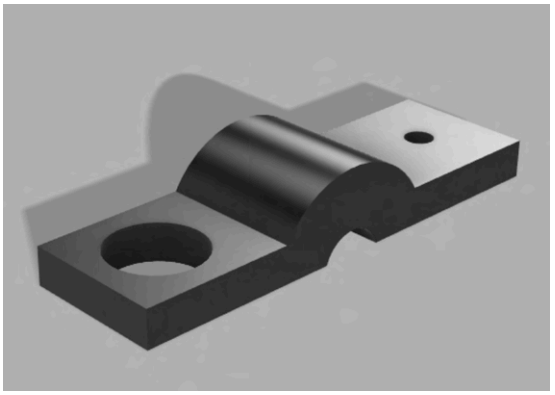
This was our second design on fusion 360 which seeked to solve a variety of issues with the first. It removed a bridge at the top of the arm and made it faster to 3D print and remove the support material after.

In the end, we couldn't justify the time it would take to 3D print multiple fabrications of the arm, especially after we discovered that the best position to place the antenna was at an angle to the alien. This led us onto the final design.

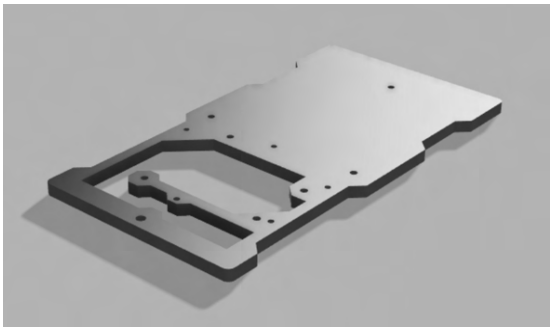
FINAL DESIGN

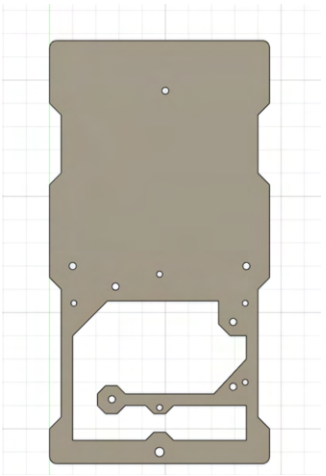
We decided on the chassis in design 2 due to its benefit of saving weight, space, and size. For an optimum approach to the alien, we removed the arm and positioned the antenna at an angle of around 45°. In order to attach it to the chassis, we designed two tighteners, one to strap the antenna onto the chassis, and the other at the top end of the antenna to support the age and magnetism sensor. They can be viewed below respectively.

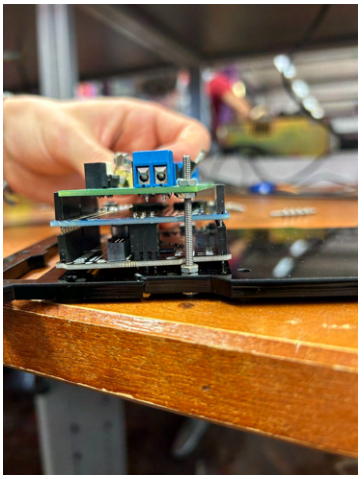




The chassis is made from 5mm black acrylic. The cut area visible below is there to make the vehicle lightweight. In order to position the motor PCB above the metroboard, we used a threaded rod with a diameter of 2.5mm onto which we screwed four nuts in total – two on either side of each board. They were cut to a length of 35mm (the one in the image is 40mm) to prevent them from protruding too much.





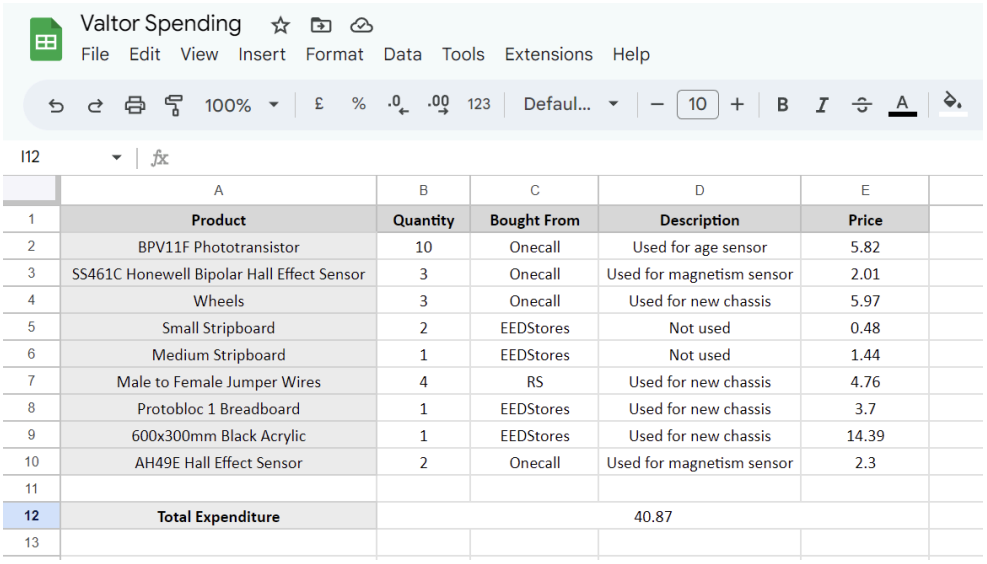


EVALUATION

FINANCES

A core focus within our project requirements was achieving a low-cost rover. The budget at the beginning of the project was £60 to spend on extra components on top of the parts that had been provided within the EEEBug upgrade kit. Besides the ordering system that had been introduced we could also use external vendors. However, there was a risk of the parts ordered not arriving and high shipping costs.

We decided to split the budget into 75% for spending and 25% for an emergency fund, therefore making our new budget £45. We decided to minimise costs by making use of scraps and supplies available for us to use within the labs. Besides the app, we also made use of a spreadsheet to keep track of our spending. Our total expenditure came to £40.87 which was within our budget. However upon review, the use of a black acrylic board and purchasing strip boards which we ended scrapping proved to be avoidable costs. Besides, a new breadboard which wasn't essential was also purchased. The image below is a picture of the spreadsheet that summarises our spending.



	A	B	C	D	E
1	**Product**	**Quantity**	**Bought From**	**Description**	**Price**
2	BPV11F Phototransistor	10	Onecall	Used for age sensor	5.82
3	SS461C Honewell Bipolar Hall Effect Sensor	3	Onecall	Used for magnetism sensor	2.01
4	Wheels	3	Onecall	Used for new chassis	5.97
5	Small Stripboard	2	EEDStores	Not used	0.48
6	Medium Stripboard	1	EEDStores	Not used	1.44
7	Male to Female Jumper Wires	4	RS	Used for new chassis	4.76
8	Protobloc 1 Breadboard	1	EEDStores	Used for new chassis	3.7
9	600x300mm Black Acrylic	1	EEDStores	Used for new chassis	14.39
10	AH49E Hall Effect Sensor	2	Onecall	Used for magnetism sensor	2.3
11					
12	**Total Expenditure**			40.87	
13					

A large portion of our budget went towards purchasing new materials and wires for our chassis. This was chosen as using jumper cables allowed for more organisation and standardised cables, while a large black acrylic would have adequate space for at least 2 laser cuts of the chassis. Other items we obtained and services we used include the 3D printing service which was free of charge and components like op-amps, resistors and capacitors which were all found within the lab and could be used with no extra cost.

CONCLUSIONS

Review

Overall Valtor feels that the rover we have constructed and tested is able to navigate around the environment efficiently, while also detecting incoming signals from the various aliens present and differentiate them accordingly. Through our results and testing, we know that our sensors are accurate and are able to work simultaneously with each other with no interference. The chassis design chosen focuses on implementing a lightweight structure whilst having an ease of access to numerous components. Additionally the 3D printed arm attachment works as intended to position the sensors in their most ideal position to read results from the alien model. With regards to software, the code written managed to correctly convert the information from the sensors accurately and display readings for the end user to read. However, we faced many difficulties with the front end of the website due to the 250 line limitation on the Metro board. Besides, there is still a degree of input lag over the WiFi network given. Nevertheless the code written was functional and manages to perform the tasks and convey both commands and information with the limited memory provided.

Future Work/Developments

After assembling our rover and running tests, we felt that the weakest aspects of our rover functionally were the speed at which the rover is able to manoeuvre around the circuit, the strength of our magnetic sensor and the range of our name sensor. These could be further developed by purchasing and looking into more hall effect sensors that are of higher sensitivity, therefore providing us with a greater range of detection. The range of the name sensor could be improved by double amplifying the signal received or using an exposed thicker copper coil wire. With regards to speed, we could purchase higher power motors or alter the power system pushing 12V instead of 6V to the motors. However altering this would also increase the weight of the rover, a key feature of the rover which we wanted to focus on.

In addition to this, we felt that the construction of our rover could also be improved with the circuits being soldered onto boards instead of being constructed on the breadboard. This would prevent components from possibly losing connection when being transported and would allow for a much cleaner finish. Aesthetically we could also design an enclosure that keeps out any dust or particles and conceals any exposed wires. Other points of interest we also considered were developing an entire app instead of a web-app as this would allow for more control and less storage limitations.

What Could Be Done Differently

Upon our group meeting prior to the submission of the project report, we felt that beginning work on the software of the rover earlier would have been more desirable and provided us with more time to debug any errors. We also felt that the late development of our chassis meant that we had a restricted time frame to test fit all the components on the final board and perform full tests with our 3D print arm. Extended research and further discussion with teaching assistants would also be an approach we would all take if we were to start the project again. This would allow for clearer plans and less wasted time (e.g. our incorrect assumption and decision to develop a band-pass filter). Lastly, we also felt that we could consider a platform where numerous individuals could edit code at the same time, this would reduce the time needed to debug errors and increase workflow.

REFERENCES

[1] (Dilip-Raja, Frequency Counter using Arduino, 2016) https://circuitdigest.com/microcontroller-projects/arduino-frequency-counter-circuit#:~:text=Arduino%20Frequency%20Counter%20Code%20Explanation%3A&text=Htime%20%3D%20pulseIn(8%2CHIGH,negative%20levels%20in%20Micro%20seconds (Accessed June 15, 2023)

[2] (Dr Ed Stott, EEERover, 2023) https://github.com/edstott/EEERover/blob/main/doc/README.md (Accessed June 15, 2023)

[3] (Dr Ed Stott, EEERover, 2023) https://github.com/edstott/EEERover/blob/main/metro-starter-arduino/metro-starter-arduino.ino (Accessed June 15, 2023)

[4] (Modular Circuit, 2012) https://www.modularcircuits.com/blog/articles/h-bridge-secrets/h-bridges-the-basics/ (Accessed June 15, 2023)

[5] (Store)https://uk.rs-online.com/web/p/phototransistors/8050523 (Accessed June 15, 2023)

[6] (Store)https://www.rapidonline.com/kingbright-l-93dp3bt-3mm-phototransistor-940nm-30v-58-2005 (Accessed June 15, 2023)

[7] (Store)https://uk.rs-online.com/web/p/phototransistors/7082683 (Accessed June 15, 2023)

[8] (Wikipedia, 2012) https://en.wikipedia.org/wiki/Transimpedance_amplifier (Accessed June 15, 2023)

[9] (Store)https://www.electronics-tutorials.ws/opamp/opamp_3.html (Accessed June 15, 2023)

[10] (Store)https://uk.farnell.com/honeywell/ss461c/sensor-hall-effect-bipolar-to92/dp/1811311 (Accessed June 15, 2023)

[11] (Store)https://www.electronics-tutorials.ws/electromagnetism/hall-effect.html (Accessed June 15, 2023)

[12] (Store)https://circuitdigest.com/electronic-circuits/magnetic-polarity-detector-circuit-using-hall-sensor#:~:text=Here%20two%20Hall%20Effect%20sensors,South%20Pole%20of%20a%20magnet. (Accessed June 15, 2023)

[13] (Metro M0 Manual) https://www.mouser.com/pdfdocs/adafruit-metro-m0-express-designed-for-circuitpython.pdf (Accessed June 15, 2023)

[14] (Store)https://onecall.farnell.com/diodes-inc/ah49ez3-g1/hall-effect-sensor-40-to-85deg/dp/3373778 (Accessed June 15, 2023)

APPENDIX A: NAME CALCULATIONS

Resonant circuit:

In an RLC circuit, you can ignore the resistor and equate the impedances of the inductor and capacitor when finding the resonant frequency to give:

$$\frac{1}{j\omega C} = -j\omega L$$

Given that the resonating frequency is 61kHz and the inductance is 114µH, we can calculate the capacitance as follows:

$$C = \frac{1}{\omega^2 L}$$

$$= \frac{1}{4\pi^2 \times 61 \times 10^{3^2} \times 114 \times 10^{-6}}$$

$$= 60nF$$

In reality we used a 62nF capacitor as this was the closest capacitance we could make with the values available in the labs. When calculated, the resonant frequency checks out to be around 60.5kHz, which we deemed close enough.

High pass filter:

The cut-off frequency for this simple, first-order filter is calculated by equating:

$$\omega = \frac{1}{RC}$$

Plugging the correct values into the equation, we can rearrange the equation to give a cut-off frequency of 159Hz:

$$f = \frac{1}{2\pi \times 1 \times 10^3 \times 1 \times 10^{-6}}$$

$$= 159Hz$$

Low pass filters:

Both cut-off frequencies are calculated in the same manner as shown above. For the first low pass filter, the cut-off frequency is:

$$f = \frac{1}{2\pi \times 16 \times 10^3 \times 4.7 \times 10^{-9}}$$

$$= 2116Hz$$

For the second low pass filter, the cut-frequency is:

$$f = \frac{1}{2\pi \times 200 \times 10^3 \times 470 \times 10^{-9}}$$

$$= 1.69Hz$$