Imperial College London

YEAR 1 FINAL GROUP PROJECT

IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

JABA Rover

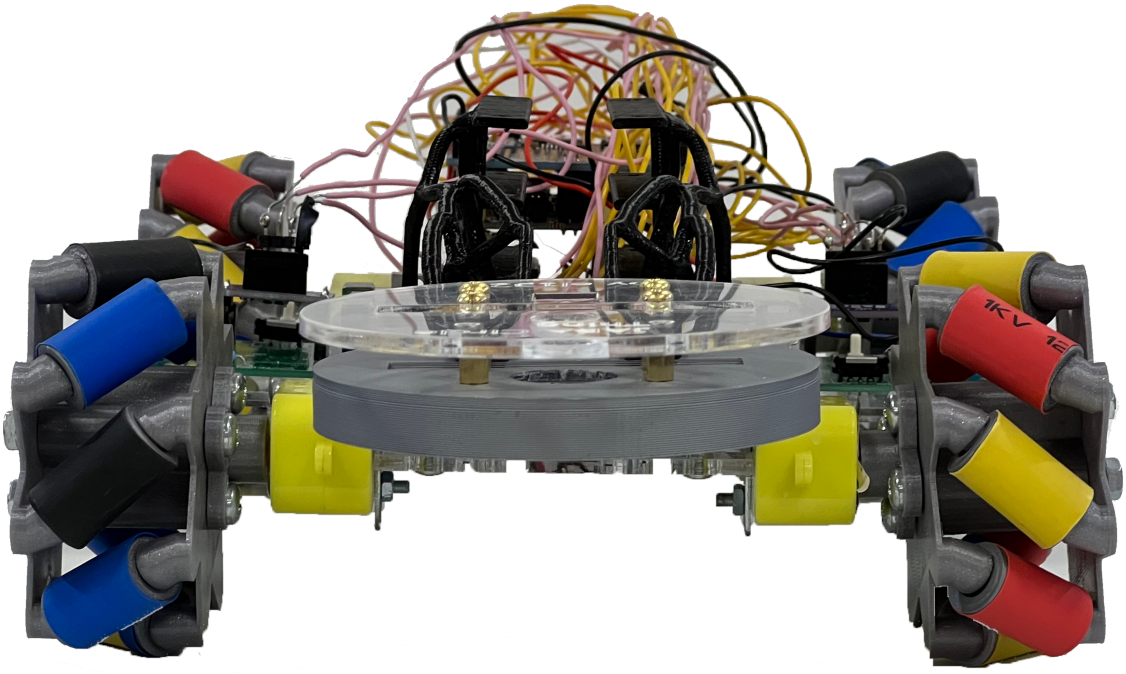
Authors:
02021051 Devgan Sharma Dhruv
02029103 Gibson Thomas
02108888 Liu Chang
02035682 Liu Renwei
02050238 O' Malley Corey
02044991 Rinaldi Luigi
02053685 Varvarigos Anastasis

June 19, 2022

Contents

Abstract

The purpose of the project was to create a functioning rover to cross the landscape of the Moon, name rocks based on their characteristics and meet the requirements set out in our Product Design Specification (PDS). The problem was broken down into its core components: sensors, software, and chassis to which a team was allocated. Circuits were designed on LTSpice and rigorously tested using Picoscope. These were then mounted to a chassis that was developed using Computer Aided Design (CAD) software such as Fusion 360. During the assembly of the rover various practical skills were used including 3D printing and laser cutting. Software was used to gather the information provided by the sensors to provide users with tangible data and information about the rock. This was achieved through a native app that used Electron to combine React (framework) and Node (runtime machine) in conjunction with User Datagram Protocol (UDP) to allow communication between the rover and the app. The end product was a rover that could not only identify the rocks accurately but also has great manoeuvrability due to innovative mecanum wheels providing it with a small turning radius and omni-directional movement.



Chapter 1

Introduction

The project consisted of both hardware and software applications. In the first domain, there was a careful research in the sensors used and the components, in order to acquire the proper solutions to the tasks given. This was done through the understanding of data-sheets. After that, using information from the courses, the practical labs during the year and also from various sources online, the team was able to find solutions to the problems given. The second domain was the development of a server for the control of the rover, which would also be responsible for receiving the information gathered from the sensors, and interpreting them correctly to be able to distinguish each Exorock.

Chapter 2

Design and Planning

2.1 Gantt Chart

The Gantt Chart is included in appendix A.1.

2.2 Teamwork

Initially, when the brief was released, we collaborated as a team for ideas on how to tackle each section. Then we split into groups and delegated tasks depending on the individuals' strengths and passions. Members that worked well together, with similar personalities, were set on the same task to increase efficiency. Frequent team meetings were arranged to help keep everyone focused on their assigned roles. These project management methods employed ensured smooth sailing throughout the project and were paramount to our completion of the project.

2.3 Budgeting

A.7

Chapter 3

Product Design Specification

3.1 Weight and Size

When designing and creating projects for use in space the most important aspects are weight and size, even the cheapest transport (the newest falcon heavy) costs over £1000 for each Kilo of payload [1] and this is without including size, making this a high priority.

Another thing to consider when building the rover is how the weight of the chassis and sensors may reduce maneuverability and speed, factor which are important when trying to move across the surface quickly.

The size of the rover must generally adhere to two aspects, functionality and cost of transport. The rover should be as small as possible in order to reduce transport costs, but still should be large enough to have a sturdy frame and easily accommodate the microcontroller along with each sensor in a place that can easily scan a rock effectively.

3.2 Performance

The performance of the sensors can be broken down into a few aspects, speed of detection, range, and accuracy. The speed of detection is important as it allows the rover to scan larger numbers of rocks and cover a larger area with the same amount of battery life. Range is more complicated as it needs to be large but directed as otherwise it will be easy to scan multiple rocks and add interference. Finally, the accuracy should be great enough such that given one of the frequencies it cannot be easily measured as the other or incorrectly.

Other measures of performance include maneuverability, power efficiency and speed, in which how the rover moves as well as for how long on a single charge are important as they increase the overall ability to complete the tasks at hand without breaks.

3.3 Environment

The environment that the rover is running in is very harsh, with no atmosphere, high amounts of UV, and rough terrain. Because of this the rover requires high maneuverability to be able to overcome this, also the sensors must be able to effectively work with high UV particularly the infrared sensors due to the large amounts of noise encountering the rover without an atmosphere. Another factor is the temperature as it will have to run in very cold and very hot conditions [2] ranging from -208 C to 120 C.

An effect of the lack of atmosphere from the moon may also introduce large amounts of noise as rocks radio signal will pass to the rover easier than that of in an atmospheric room and so will more likely interfere.

3.4 Cost

The cost of the rover is limited to a £60 budget and such each sensor should follow this budget with their importance, for example radio will have a larger budget due to being able to receive more information than the others. Some other areas to consider the costs are material costs to produce a newer chassis and previous prototypes produced during development. Following these and producing a high-performance rover is what is most important, but the overall cost should be kept to a total minimum allowing for cheap production.

3.5 Aesthetics

Aesthetics although aren't important to the rover's functionality it can provide a focus on ease of use and accessibility, which can be important factors when being used by a customer and when in active use to reduce problems occurring while in the field. Some parts also may provide other additional functionality gains such as a shielded cover of the internals can help the rover to be able to reduce interference from outside signals.

3.6 Testing

When testing the rover many parts should be measured to allow easier modifications in the future as well as finding issues that may occur after long periods of usage. For example, the power consumption should be measured to find the amount of active use time that the rover may run without having to stop for recharging.

Each system should also be tested both independently as well as together as, some systems may interfere with each other or cause general issues with battery and computational power. On top of this how accurate and reliable is each sensor for running, when multiple sources are acting in near proximity or large amounts of artificial noise is introduced.

3.7 Disposal

Disposal is very important as the rover will be placed onto the moons surface, and such should either be easily retrieved able or will effectively breakdown after being on the moon for a long period of time.

If on earth, disposal should be considered in the sense of recyclability, reusability as well as a reducing in the amount of non-recyclable parts that are used to build the rover.

3.8 Materials

Materials have a few considerations it needs to be, lightweight, affordable, abundant, easily workable, and environmentally friendly each of these need to be considered as they will affect the overall ability of the rover as well as its external factors such as cost and ease of manufacture.

3.9 Process

Many separate processes need to be completed to create the rover, these can be separated into manufacturing process as well and running processes. Included in the manufacturing processes are 3D printing of components, soldering of wire connections, laser cutting and design, coding the website, overall component linking and integration. While on the other hand running process include the websites communication with the Arduino, each sensors data processing, the Arduinos

data response, and the processing of the data on the computer to output the correct rock information.

3.10 Ergonomics

Ergonomics concerns the way that people interact with the project, this includes the role of each person within the group having the job that they want as well as can effectively complete. Another main focus of this is the way that a user can interact with the rover easily with a good UI as well as fast reaction times and an easy way to see the outputted data from the rover after receiving a response from the rover.

Further specifications can be found in A.1

Chapter 4

Rover Chassis

4.1 Mecanum Wheels

4.1.1 Kinematics

A mecanum wheel is a wheel that contains 45° small rollers. There are 2 types of mecanum wheels, one with rollers pointing to forward-left and one with rollers pointing to forward-right. The rollers convert the velocity into diagonal direction and thus provide the horizontal velocity and vertical velocity, as shown in Figure 4.1.

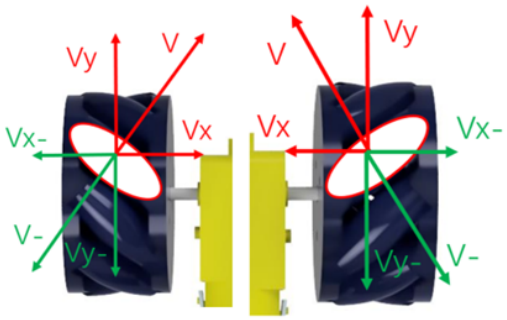


Figure 4.1: Kinematics for 2 Types of Mecanum Wheels Spinning Forward (red) and Backward (green) [3]

When the mecanum wheels are oriented in specific ways (e.g., in Figure 4.2), it is possible to utilize this advantage and achieve omni-directional movement, i.e., the rover can move to any direction without turning.

It is achieved by vector sum. For non-turning all angle movement, it is necessary to have a sum vector that only provides the pure direction vector. The minimum requirement for achieving this is 3 mecanum wheels, where the vectors that are not needed on each wheel are cancelled out. The EEERover is designed to have 4 mecanum wheels for aesthetics and easiness.

For example, in Figure 4.2 the movement of the chassis is toward left indicated by the big red arrow. The small red arrows on each wheel represents the forward or backward motion powered by motor. The rollers convert each motion into 45° velocity vector. The component vectors of this 45° velocity is represented by green arrows similar to Figure 1. The 4 y-axis component vectors are cancelled out in pairs, the 4 x-axis component vectors all point to left, which sums up to be the final movement of the chassis.

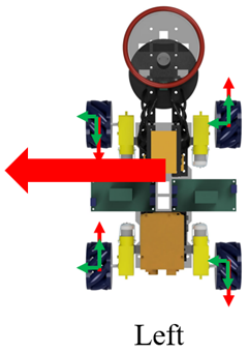


Figure 4.2: Mecanum Wheel Chassis Orientation and Left Movement

4.1.2 Control

The basic motor control using the motor drive PCB is provide a PWM signal using `analogWrite()` function toward `EN` pin and a digital signal using `digitalWrite()` function toward `DIR` pin. PWM, which ranges from 0 to 255 in code, enables the speed control in `EN` pin by generating a square wave which results in controlled average voltage. Digital signal in `DIR` pin determines the direction of the motor rotation. A sample code for single motor control is as follows:

```
void frontright(int sped, bool isForward){
    digitalWrite(frdir, isForward);
    analogWrite(frpwm, abs(speed));
}
```

Each motor drive PCB contains two sets of output, which can be used to control two motors. To achieve 4 motor control, two motor drive PCBs are installed.

There are two methods for controlling mecanum wheel chassis - the discrete control which utilizes the two-state buttons (e.g., keyboards) and the continuous control which utilizes joysticks. The two control methods both involve operating vector sum inside code.

As Adafruit `Metro M0 Express` is limited in its computing power. The control of the overall chassis is implemented in a closed loop operated by another Orangepip. The data is communicated through an `I2C` bus where `M0` is the master and `Orangepip` is the slave. The closed loop system is illustrated in Figure 4.3.

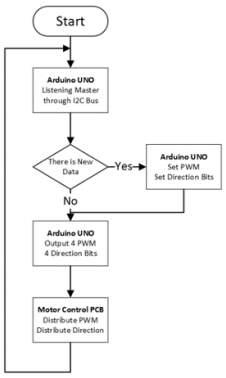


Figure 4.3: Closed Loop Chassis Control System

Discrete Control

Discrete control implements the omni-directional movements by setting each movement into a single function. Each function is called when Orangepip receives instruction from M0. The movements and the respective motion for each wheel is shown in Figure 4.4.

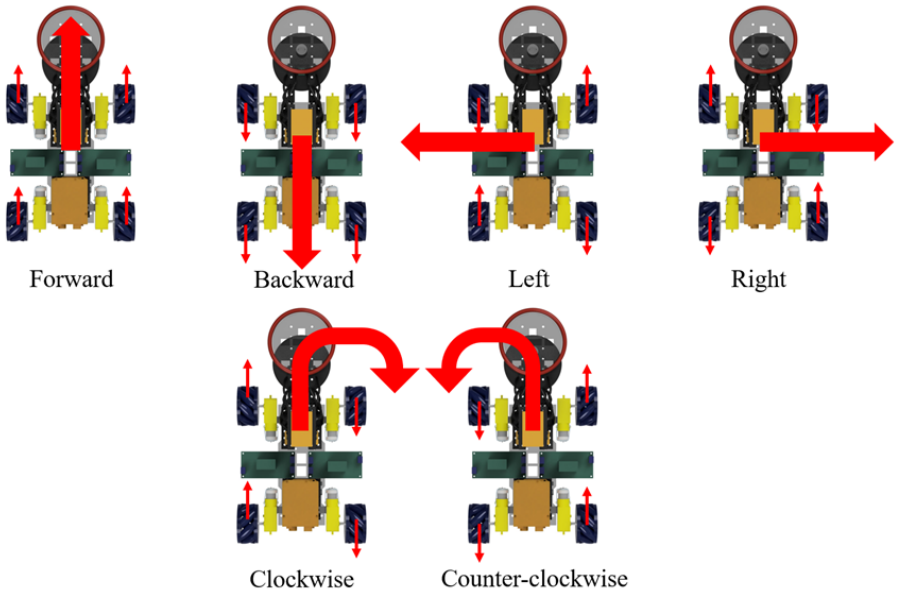


Figure 4.4: Mecanum Wheel Chassis Movement with Wheel Motion

A typical function is shown as follows:

```
void left(int sped, bool isForward){
  frontleft(sped, !isForward);
  backleft(sped, isForward);
  frontright(sped, isForward);
  backright(sped, !isForward);
}
```

The instruction sent from M0 through I2C bus contains two information, a Char type (8 bits) representing the direction, and 3 Char types each represents one digit in the speed (000-255), see Figure 4.5. The code determines the function to call based on the direction Char.

When the interrupt from I2C bus is called, the code calls the respective function determined by the direction Char. It sets the speed for function from the 3 speed Char as well.

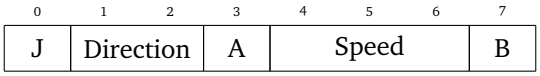


Figure 4.5: Discrete Control Encoding

Continuous Control

For a 4-wheel mecanum chassis, the speed for each wheel for omni-directional driving can be determined as the following equation [4].

$$\begin{cases} v_{frontleft} = v_y + v_x \\ v_{frontright} = v_y - v_x \\ v_{backleft} = v_y - v_x \\ v_{backright} = v_y + v_x \end{cases} \tag{4.1}$$

For rotation, v_r is introduced. Similarly to the principle in clockwise and counterclockwise in figure 4, v_r acts positively for the left wheels and negatively for the right wheels.

$$\begin{cases} v_{frontleft} = v_y + v_x + v_r \\ v_{frontright} = v_y - v_x - v_r \\ v_{backleft} = v_y - v_x + v_r \\ v_{backright} = v_y + v_x - v_r \end{cases} \tag{4.2}$$

The y-axis and x-axis components, as well as the rotation, are determined from the joystick. The idea of proportional control is implemented in control end (6.1.4), where the computer transfers the location of joystick (ranges from -1 to 1) into a speed (ranges from -255 to 255).

However, this method causes the overflow of speed, as it is possible to have a value of 510 in speed where 255 is the maximum. To counter for this overflow, a denominator v_{denom} is introduced. It is calculated as follows: $v_{denom} = max(|v_y| + |v_y| + |v_r|, 255)$
After calculation as follows, the overflow is countered by changing everything into proportion.

$$\begin{cases} v_{rontleft} = \frac{v_y + v_x + v_r}{v_{denom}} \cdot 255 \\ v_{rontright} = \frac{v_y - v_x - v_r}{v_{denom}} \cdot 255 \\ v_{backleft} = \frac{v_y - v_x + v_r}{v_{denom}} \cdot 255 \\ v_{backright} = \frac{v_y + v_x - v_r}{v_{denom}} \cdot 255 \end{cases} \tag{4.3}$$

The instruction sent from M0 through I2C bus contains 6 information (in Char/Chars, detail in figure 4.6) – the direction for rotation, the direction for forward and backward movement (y-axis), the direction for left and right movement (x-axis), the speed for forward and backward movement, the speed for left and right movement, and the speed for rotation. These contributes to the sign and the value of v_y, v_x and v_r.

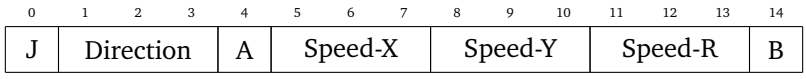


Figure 4.6: Analogue Control Encoding

An interrupt routine running in Orangepip works out the respective values in Eq 4.3 and send them to each motor control function. However, as Eq 4.3 indicates, the value from continuous control is determined by sign mark. In order to be compatible with discrete control at the same time, a logic function is implemented to achieve as in Table 4.1.

Speed Is Positive	Is Forward	Final Direction
1	1	1
1	0	0
0	1	0
0	0	0

Table 4.1: Truth Table for Motor Direction

This is an AND gate, and the compatibility is achieved simply by setting isForward to True when using continuous control. A sample code is as follows:

```
void frontright(int sped, bool isForward){
  digitalWrite(frdir, (sped>0)&&isForward);
  analogWrite(frpwm, abs(sped));
}
```

4.1.3 Testing

First tests are conducted to study the discrete control. The tasks and the results are in Table 4.2 The movement is very straight in direction (see figure 5). There is a small deviation, however it can be easily countered by the driver.

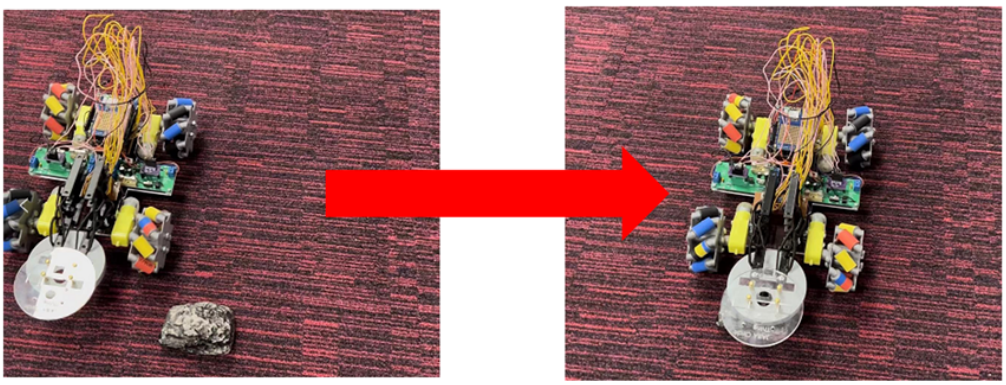


Figure 4.7: Left Movement

The continuous control provides a true omni-directional capability. A simple testing of omni-directional capability is to make the rover draw a circle. The process is split as figure 4.8, and the results are in table 4.3.

It is observed that during task B, task D, task F, and task H the speed of the overall movement is reduced by significant amount. The reason is the driving motor is reduced to 2 – the power is halved.

Task	Result
Forward	Passed
Backward	Passed
Left without Turn	Passed
Right without Turn	Passed
Clockwise Rotation	Passed
Counterclockwise Rotation	Passed

Table 4.2: Discrete Control Testing

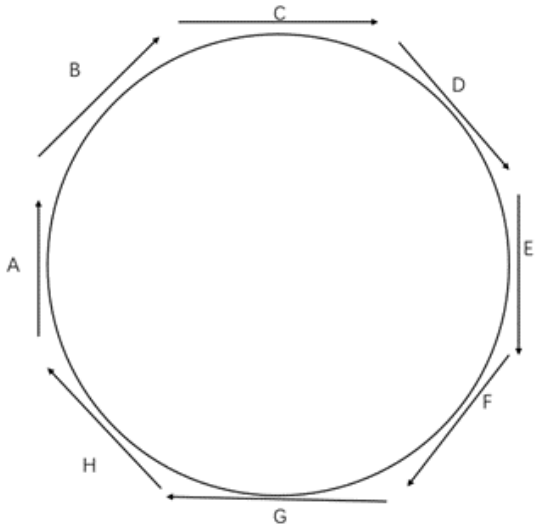


Figure 4.8: Circle-Drawing Process

Task	Result
A	Passed
B	Passed
C	Passed
D	Passed
E	Passed
F	Passed
G	Passed
H	Passed

Table 4.3: Continuous Control Testing

4.2 Frame Design

After the EEEBug chassis is proven to be unable to hold the payloads we have, a total of 3 major iterations are conducted for designing the frame of the rover. The design focused on three main parts – the size, the availability for holding payloads, and the redundancy.

The size is constrained by the obstacle distance. To manoeuvre between obstacles that are distanced at 500mm minimum, the rover is required to be smaller than 500mm. The payload availability is constrained by the size of breadboards, stripboards, PCBs, and sensors mostly. The rover is required to provide space to hold all the payloads. The redundancy is considered to remove errors and provide mounting places for potential new payloads.

4.2.1 Chassis Iterations

Initially, the design is mostly focused on providing a simple testing platform for mecanum drive. Mounting holes are either replicated at a different position or enlarged to account for errors. The first major iteration chassis is sized in around 200mm. It is designed with mounting holes for holding all the mecanum drive components as well. The sensor payloads are installed on the second layer sized in around 200mm.

After testing with the first major iteration chassis, problems are encountered which includes unstable motor mounting, mounting hole deviation, and component overlapping. The second major iteration chassis is designed to solve most of the problems. Additionally, it utilizes the engraving in laser cutting to provide intuitive information and aesthetic enjoyment. Most of the testing is conducted on the second major iteration chassis.

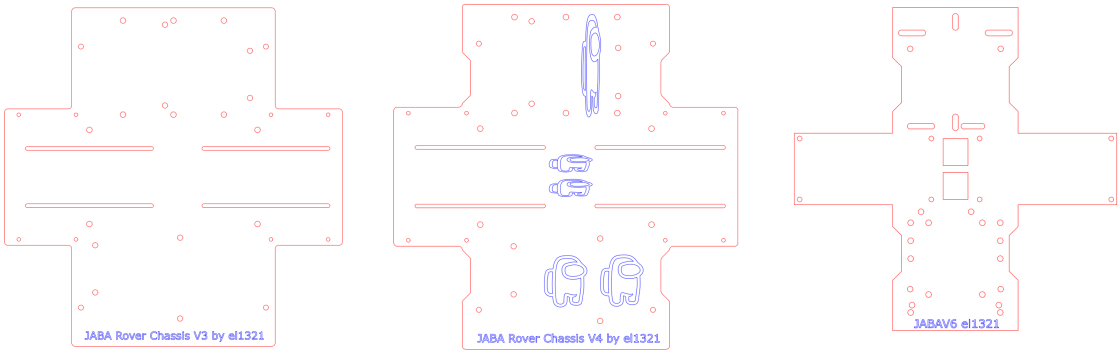


Figure 4.9: First, Second, and Third Major Iteration Chassis Laser Cutting Drawing

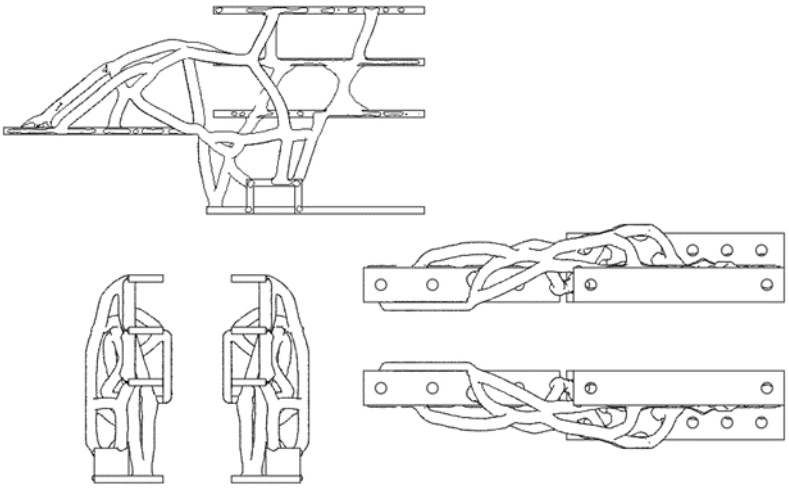


Figure 4.10: Sensor Holder

During the testing of second major iteration, problems are encountered which mostly is the manoeuvrability. A smaller chassis is a new demand. The third major iteration chassis compressed everything into a minimal design. Mecanum drive components are designed to be mounted on the chassis, and all the sensor payloads are mounted on a newly designed 3D printed holder as figure 4.10. The third iteration minimizes the Rover's size to a great extent. It is eventually becomes the final iteration.

4.2.2 Design Techniques

The design of the frame involves laser-cutter, additive manufacturing, and generative design. They speed up the design workflow and reduced the dependency on knowledge of structural mechanics.

Laser Cutter

Laser Cutter involves a laser with adjustable focus, a 2-axis printer head, and a vent. It scans the vector file and works on either engraving or cutting [5]. It takes a very short time to cut either acrylic or wood, and the edges are clean and smooth. The advantage of being fast makes it a very useful tool for prototyping.

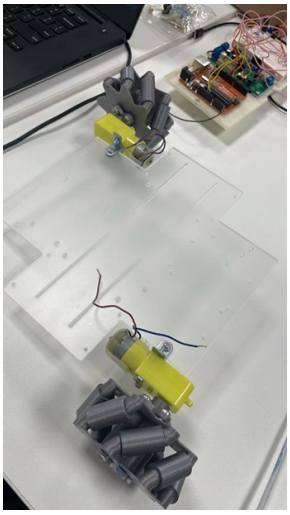


Figure 4.11: Prototyping with Laser Cut Acrylic

Additive Manufacturing

The usage of additive manufacture is helpful in prototyping as well. 3D printers are used a lot in manufacturing the mecanum wheel, the sensor holder, and the sensor plate. It takes 4 hours to print all the components for one mecanum wheel, 4 hours for the sensor holder, and 2 hours for the sensor plate. Although it takes a long time to manufacture components, it is capable for manufacture complex components with its 3-axis capability.

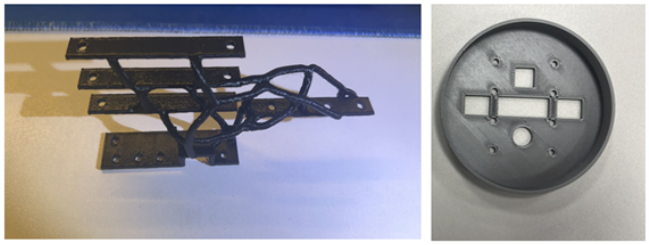


Figure 4.12: Sensor Holder and Sensor Plate Manufactured through Additive Manufacturing

Generative Design

Generative design is the design method utilized in designing the sensor holder (left in figure 4.13). The designer sets the instruction and AI runs all possible permutations and learns after iterations. [6] Eventually the design is calculated and generated. It is especially useful and easy for individuals that do not have knowledge of structural mechanics.

For example, to generate the sensor holder. Only the limitations (use PLA as material, use additive design, hold 10 newton force, provide space for stripboards, etc.) are needed. The structure is generated by AI.

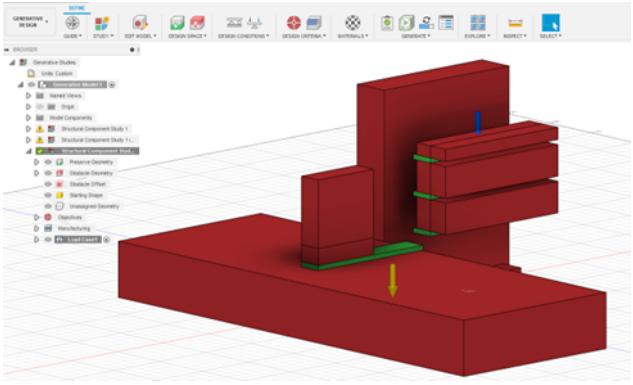


Figure 4.13: Generative Design in Autodesk Fusion 360

4.3 Evaluation

4.3.1 Mecanum Wheels

Chassis

In order to reduce cost, mecanum wheels are manufactured through 3D printer. It significantly lowered the cost from 16 to 0. However, 3D printed wheels do not have enough grip to manoeuvre smoothly. Heat shrink pipes are used to increase gripping force. However, they are suboptimal as they aren't as frictional as a real mecanum wheel. Rubber bands and epoxy can be utilized in future improvement to provide a greater grip.

Moreover, as mecanum wheel converts the forward velocity into two components, half of the power is cancelled out during movements. Thus, the speed is reduced by half comparing to normal rovers. This can be improved by introducing a better motor as the power is limited by the regulator (5V max).

Control

The continuous control is not able to achieve the full speed as the discrete control does. The reason is the inaccuracy in joystick. There is always an x-axis and y-axis component in the output and the speed will be reduced in v_denom in order to counter for the overflow. It can be improved by setting a "full speed region" where the minor axis data is ignored.

The movement is clean in its movement, however there are still deviations. The driver needs to counter for the deviation by using feedback from naked eyes and joystick control. This process can be implemented in the closed loop. Instead of the proportional control the rover currently adopted, a better and easy control method PID control can be used. This will improve the problem of deviation.

A field-centric control method can be implemented instead of robot-centric control method. This moves the centre axis from robot to the field. For example, the forward on robot-centric control method will make robot move to its own forward, whereas the field-centric control method will make robot move to the forward direction of the field, no matter how robot is oriented. This improves the driving intuitiveness and relates the robot with the field.

4.3.2 Frame Design

Chassis

Chapter 5

Sensors - Building and Testing

5.1 Operational Amplifiers

Operational amplifiers (op-amps) are required for infrared (IR), radio and acoustic sensors. The op-amp selected for each sensor may vary depending on the requirement of each circuit.

5.1.1 Op-amp Requirements

Gain Bandwidth Product

The Gain Bandwidth Product is the frequency at which there is 0dB gain, op-amps with a large Gain Bandwidth Product perform better at high frequencies. The IR sensor is dealing with low frequencies therefore, a lower gain bandwidth is acceptable. The radio and acoustic sensor deal with significantly higher frequencies, hence may require a more specific op-amp.

High Slew Rate

The slew rate is the rate of change of the output voltage caused by a step change in input. If the slew rate is too low, there will be distortion in the output which can change the shape of the graph. Furthermore, it could cause a decrease in the output amplitude. (See Figure 5.1). The required slew rate can be calculated using:

$$SlewRate = 2\pi fA \quad \text{(this is the maximum value of the derivative of } Asin(2\pi ft)).$$

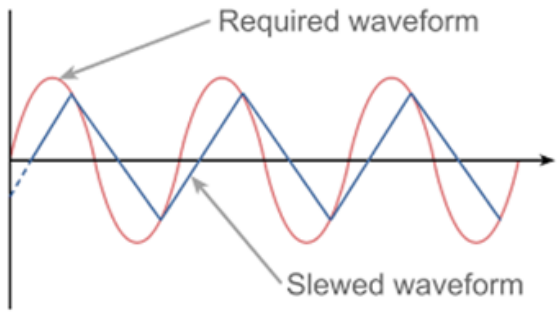


Figure 5.1: Slew rate distortion

This is more important for the acoustic and radio sensors due to higher frequencies. The maximum slew rate requirements are given below:

IR: $0.01794V/\mu s\,(571Hz)$ and $0.01109V/\mu s\,(353Hz)$

Radio: $2.7964V/\mu s\,(89kHz)$ and $1.9166V/\mu s\,(61kHz)$

Acoustic: $1.2568V/\mu s\,(40kHz)$

Rail-to-Rail Input and Output

It is useful for the op-amp to be able to operate over the entire range between the negative and positive power supply voltages as it will allow Schmitt triggers to provide a binary output which will be easier for the Arduino to read. This is important for all sensors.

Supply Voltages Range

The negative supply rail must be set to 0. This allows Schmitt triggers to operate between the full range that the Arduino can measure (0 to 3.3V). This is important for all sensors.

Comparing Op-Amps

	LT1366 [7]	**MCP6002 [8]**	**NE5532AP [9]**
Gain Bandwidth	400 kHz	1 MHz	10 MHz
Slew Rate	0.13 V/μs	0.6 V/μs	9 V/μs
Supply voltages range	±5	V_{SS} = GND V_{DD} = 1.8V to 5.5V	±22

Table 5.1: Op-Amp comparison

All op-amps would be suitable for IR sensor. The acoustic and radio sensors likely require a combination of NE5532AP and MCP6002. The NE5532AP being used for an initial amplification stage and the MCP6002 for a Schmitt trigger.

5.2 Radio

The aim of the radio senor is to detect a radio signal of 61KHz and 89Hz both modulated at 151Hz and 239Hz accurately through the rock casing. Using this information to distinguish between 4 rock types: Gaborium, lathwaite, Adamantine and Xirang.

Mineral	**Property 1**
Gaborium	61kHz radio modulated at 151Hz
Lathwaite	61kHz radio modulated at 239Hz
Adamantine	89kHz radio modulated at 151Hz
Xir	89kHz radio modulated at 239Hz

Table 5.2: Table of material properties [10]

Using the properties of minerals table 5.2 given in the brief there is no need to identify both the carrier and modulate frequencies as the second properties will provide enough information to determine the rock correctly. This will reduce the circuitry required therefore simplifying the circuit. Furthermore, fewer components required hence cheaper and lighter rover.

Before designing the circuit, the main requirements and challenges were identified:

Requirements:

- Detect signals up to a 10cm range
- High accuracy
- Reliability
- Output of 0 to 3.3V

Challenges:

- Interference from other radio signals
- Detection through rock casing
- Parasitic behaviour of components may lead to correlation error between simulation and reality

Initially the circuit was broken down into 4 parts:

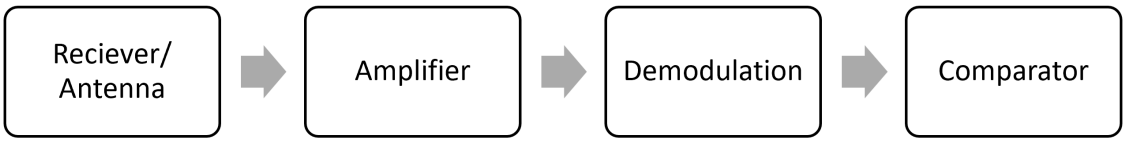


Figure 5.2: Circuit high level diagram

5.2.1 Antenna

An antenna was required to pick up the radio waves emitted from the rock. Radio waves are a type of electromagnetic wave. Electromagnetic waves contain magnetic components (Fig A.5 [11]). When an electromagnetic wave is incident on an inductor which is a wire coil with a magnetic metal core a current will be induced due to the changing magnetic field as stated by Lenz's Law. Therefore, an inductor can be used as an antenna.

A resonant circuit can be used as a tuned antenna. A well-matched inductor and capacitor will only pass-through signals within a specific range reducing interference. This is useful as the antenna can be tuned to be more sensitive for carrier frequencies of 61kHz and 89kHz.

Typical inductors are impractical for high frequencies as they suffer hysteresis and eddy currents which give rise to losses through heat and noise [12]. Hysteresis is a result of ferromagnetic core's inability to change magnetisation in phase with the magnetic field. As the magnetic field changes, the molecules collide resulting in energy loss due to friction and heating. Eddy currents are circular loops of current within a conductor which disrupt current flow causing power loss. They are the result of a moving magnetic field incident on a stationary conductor or vice-versa and form perpendicular to a magnetic field [13].

The most practical way to make an antenna is using an air coil. An air coil is an inductor without a ferromagnetic core to achieve the desired inductance. The inductance of the air coil can be calculated using the following formula [14]:

$$L = \frac{\mu_r \mu_0 N^2 \pi r^2}{l}$$

L = inductance of coil (H), μ_r = relative permeability of the core, μ_0=permeability of free space = $4\pi \cdot 10^{-7}(H/m)$, N = number of turns, A = coil area (m^2), r = coil radius (m)

5.2.2 Amplifier

The output of the rock has been AM modulated with a carrier frequency of 61k or 89k and modulated frequency of 151Hz or 239Hz. Amplitude modulation (AM) causes the amplitude of the carrier frequency to vary in line with the intensity of the modulation. The metro M0 will be unable to read the output of the amplifier at the carrier frequency therefore, the signal must be demodulated [10].

Envelope detection is a form of demodulation. It uses a diode to rectify the signal followed by a low pass filter to remove any high frequency noise that remains. The circuit on Figure 5.3 is an example of how to make an envelope detector.

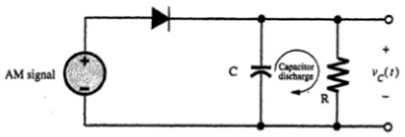


Figure 5.3: Envelope detector [15]

To calculate the values of the RC circuit the following formulae were used (see testing for actual values):

$$\frac{1}{\omega_c} << RC < \frac{1}{2\pi B}$$

$\omega_c =$ angular frequency, $B =$ Bandwidth

5.2.3 Testing

The first step was to design the antenna circuit as this would determine the amount of amplification that would be needed. The amplitude of the signal was expected to be very small. Constructed an air coil with diameter of 10cm and 13 coils, which measured 29.6μH using LCR bridge.

$$f = \frac{1}{2\pi\sqrt{LC}}$$

$$C = \frac{1}{4\pi^2 f^2 L}$$

A resonant of frequency of 75KHz was required as this is equidistant from 61KHz and 89KHz. The aim was to ensure the antenna would amplify the high frequencies better than the lower frequencies, therefore, provide clean pulses of high frequency waves.

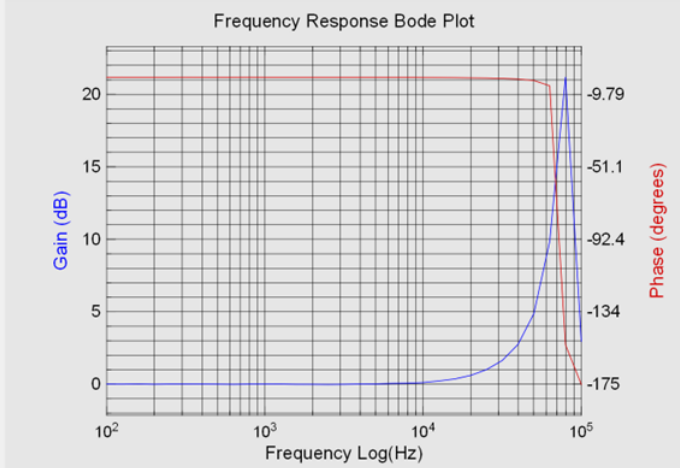


Figure 5.4: Frequency Response of Antenna LC

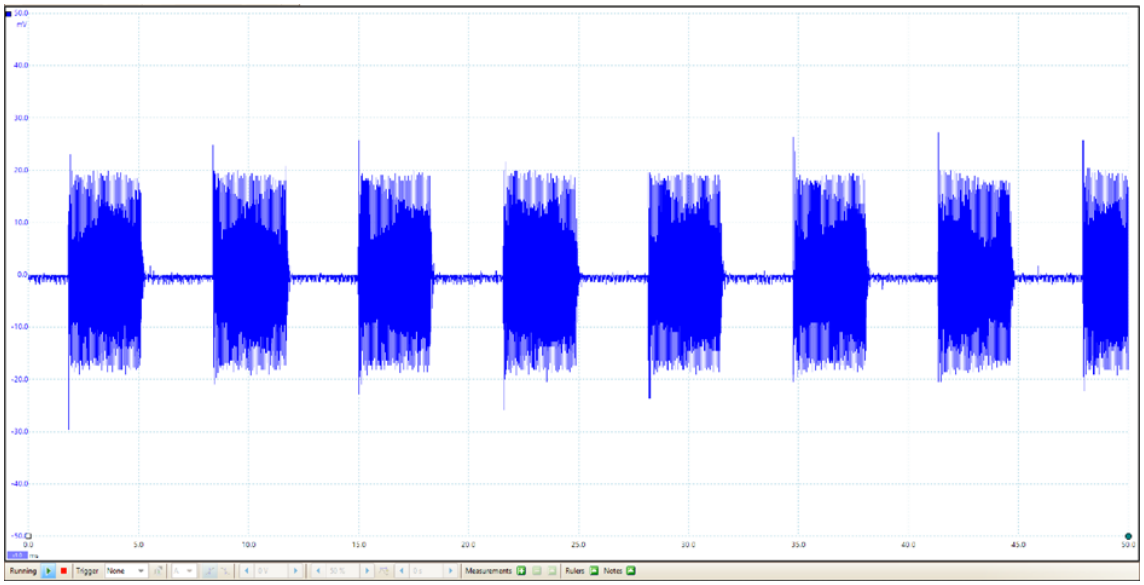


Figure 5.5: Oscilloscope measurement with no amplification

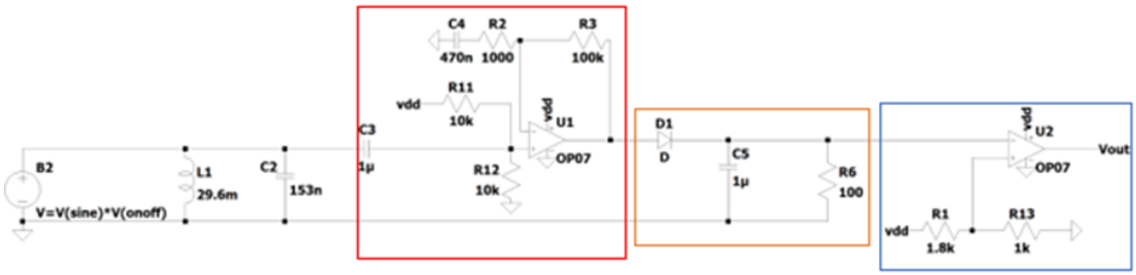


Figure 5.6: Component level circuit schematic

Antenna had a gain of 8.5dB at 61KHz and 10dB at 89KHz. The initial reading of the radio signal with no amplifier showed that the maximum amplitude was only 20mV. 5.5 Following the construction of the antenna the first design of the circuit was made:

The MCP6002 was originally used in this circuit during testing its inadequacies become noticeable. The circuit consisted of 3 stages:

Stage 1 (Non – Inverting Amplifier): Used to amplify the circuit as much as possible theoretical gain of 101. This will not hold true due to the Gain Bandwidth Product (GBP) of the op-amp. The capacitor before the non-inverting is used to block DC voltage. The potential divider is used to bias the circuit, this prevents op-amp saturation. Saturation is caused when the signal falls out of the supply voltage range. The 470nF capacitor is used to maintain the biasing.

Stage 2 (Demodulation): The envelope detector is a half wave rectifier that charges a capacitor to the peak voltage of the input signal. When the output of the diode is increasing the capacitor charges and discharges when output is decreasing. Usually followed by a low pass filter to remove noise within the circuit. This caused the circuit to behave differently to the simulation and narrowed the range of operation even further. By removing the low pass and tuning the comparator a better range was achieved.

Stage 3 (Comparator): The threshold voltage was set to 1.78V. Had to be very precise due to the limitations mentioned previously.

The circuit performed well with accurate measurements when tested with Picoscope the percentage error was 0.1%. However, the range was less than 1cm (dependant on positioning over

the rock) which provides little leeway with the positioning of the sensor on the rover and risks touching the rock during the demo. A minor improvement was made by changing the $1k\Omega$ resistor in the first amplifier improving gain, however, due to the GBP an increase of only 0.7cm was seen.

Taking as much information from the short comings of the first circuit a new version of the circuit was developed. The key takeaways were:

- Op-amp needs to be upgraded with better GBP.
- Develop a more reliable demodulation circuit.
- Comparator at the end worked well when set to the correct threshold voltage.

After the original design behaviour was far worse than expected the LTSpice simulation was improved to more accurately represent the input signals. Due to the AM modulation the signal is not trivial to represent. The signal was accurately represented by multiplying together a square pulse of frequency 151Hz a sine wave of 89kHz (Figure 5.7). These create the following wave, which accurately depicts the one measured in practice.

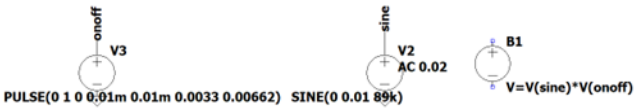


Figure 5.7: LTspice signal generator

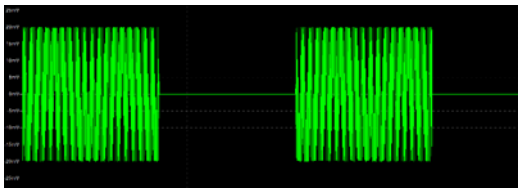


Figure 5.8: Signal Generated in LTSpice

Final Circuit

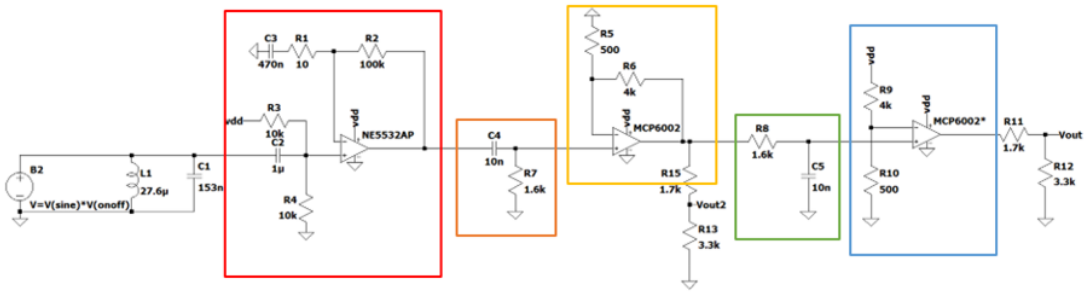


Figure 5.9: Final Circuit schematic

Stage 1 (non – inverting amplifier): Similar to the first circuit with an updated op-amp which has a significantly better GBP.

Stage 2: This is used to remove the bias introduced in the first amplifier

Stage 3: The second amplifier reverts back to the MCP6002. This is because the NE5532AP does not operate rail-to-rail. By removing the biasing, the op-amp was purposefully saturated. As

the supply rails were set to V_+ = 5V and V_- = 0V however, input was below 0V. Hence, everything between 0V and 5V was amplified and the negative part was removed, essentially rectifying the signal. The high frequency part of the signal is less amplified due to restriction in GBP and slew rate.
The carrier frequency was detected from the output VOUT2 through the use of software. This is explained later in the software section 6.3.1.

Stage 4 (Low-pass filter): Removes the high frequency carrier frequency. To leave a distorted square wave.

Stage 5 (Comparator): Any signal above 0.55V was set to 5V. Removing noise seen after stage 4. The comparator requires some tuning, originally it was set to 1V which resulted in extra spikes in the square wave, this was due to downward spikes due to capacitors discharging crossing the threshold voltage which gave extra square waves. When testing on the Arduino this became evident due to large frequencies being detected. The solution involved reducing the threshold voltage so that the distortion did not affect the output.

After the comparator a potential divider is used to set the maximum voltage to 3.3V this is done to avoid damaging the Metro M0's voltage regulator.

5.2.4 Evaluation

There was a large improvement in range to 12.2cm, however, due to distortion issues mentioned previously it was unreliable below 2cm. After improvements to the circuit the range was between 0 to 12.2 cm. Measurements contain on average a 1% percentage error which is an increase over the previous method likely due to the large amount of amplification causing slight distortion to the results.

The change to the op-amp rectifier was beneficial from a range perspective, however, the envelope detector was a cheaper option as it involves very basic components. However, with the new circuit carrier frequency was detected and used to validate the results.

The final design met all the specification points as it is able to accurately detect the modulated frequency at a reasonable range with the casing on. As an improvement the coil could be redesigned to be smaller and incorporate new designs that would reduce parasitic capacitance within the antenna such as basket-weaving or spiderweb (appendix) [12].

5.3 Infrared

5.3.1 Design and planning

The square wave, low frequency, infrared output by the rock initially looks straightforward to analyse, however three main challenges were quickly identified:

- When the rock casing is placed over the rock core, nothing is received at all, hence amplification is required.
- The original EEEBug phototransistor supplied senses visible light which could be problematic, this was thought possible to solve by wrapping bin lining around the tip or by ordering alternative transistors.
- The signal is very directionally sensitive and can only be detected when pointed at a specific angle above the core's emitter.

5.3.2 Components

Phototransistors act as current amplifiers, similar to Bipolar Junction Transistors, but the base is exposed semiconductor that absorbs light to generate electron-hole pairs, which produces a current

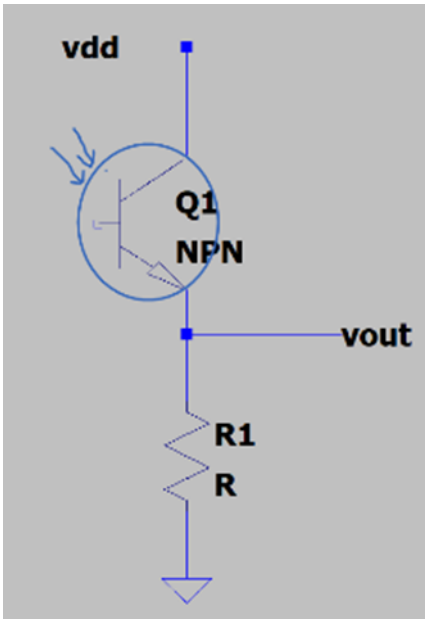


Figure 5.10: showing the circuit diagram for the transistor created in LT SPICE

across the junctions [16]. Constructing a circuit as shown in fig 5.10 generates a current which then causes a voltage drop across the resistor, therefore increasing the resistor will increase the voltage drop. This voltage can then be amplified and analysed. The phototransistor SFH 300 FA was deemed appropriate for the project as it filters out visible light thank to the black coating on the receiver. The wavelengths detected by the component range between 730 and 1120nm (fig A.2), which coincide with the infrared requirements.

5.3.3 Prototyping and Testing

It was initially thought only amplification would be required, however, after testing the response to a large amount of amplification then the signal can be received through the rock casing, but also some noise that is amplified can be detected as well. This was theorised to be the lights flickering in the room, which was supported by the peaks disappearing when the transistor was covered. This occurred at 100Hz therefore a high pass filter with a corner frequency of 194Hz was added before the op-amps. Two stages of amplification were required to achieve a reasonable amplitude, this resulted in small peaks of noise still getting through, since the pure signal is wanted, a comparator biased around 4.5V was added, which gave a clean output.

There was still the problem that the sensor had to be directly perpendicular to the pulse to be picked up, especially since the new transistor had poor angular sensitivity, as shown in fig A.3. Therefore, an array of phototransistors (A.4) was constructed in parallel, where the currents add at the end of the node so if generally pointed over the rock always some signal is found. This drastically increased the horizontal range and also slightly increased the amplitude at a given distance.

5.3.4 Final implementation and building

Stage 1: current source that consists of an array of phototransistors emitting the approximate output signal, it simulates an infrared pulse at approximately 353Hz

Stage 2: high pass filter with a corner frequency of 194Hz to block out noise from light.

Stage 3: two non-inverting amplifiers using the MCP6002 op-amp, with a gain of 1001 and 1352 respectively. There is no need to bias the input as the signal always stays above the negative voltage rail due to the non-inversion, hence there is no distortion.

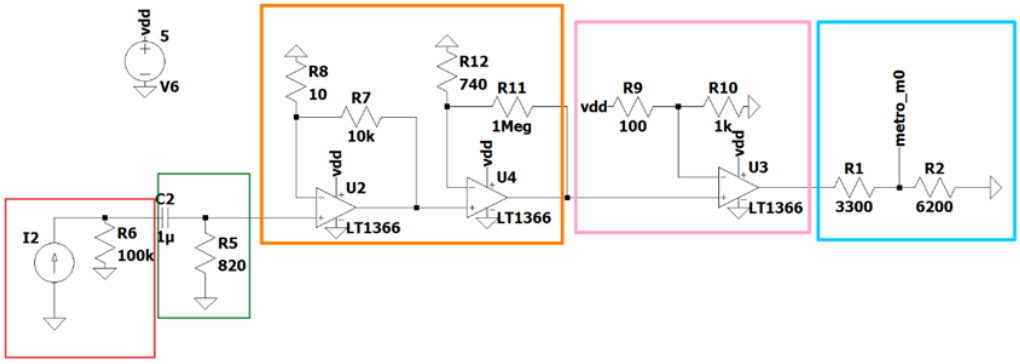


Figure 5.11: Final Circuit schematic

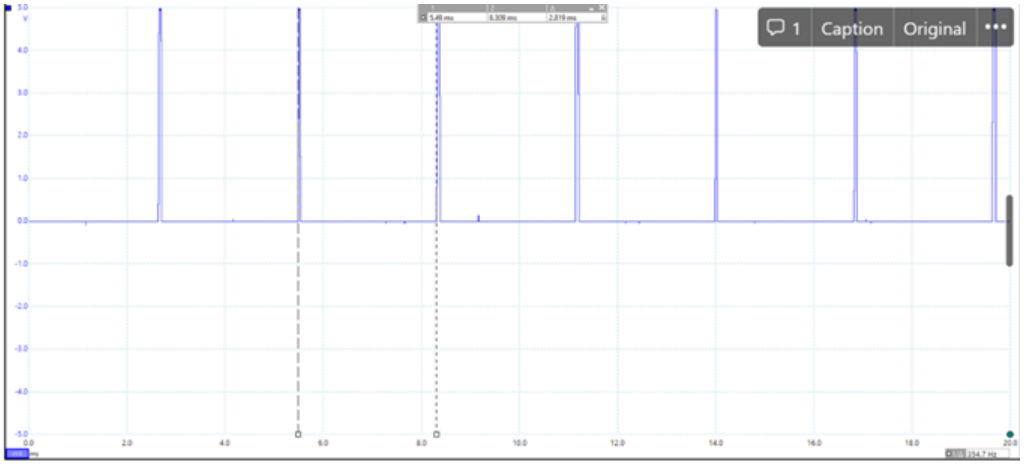


Figure 5.12: showing the output via Picoscope

Stage 4: comparator with a threshold of 4.5V to ignore noise that occasionally peaks to 4V

Stage 5: potential divider at the output that limits the pulses voltage to not exceed 3.3V, otherwise the Metro M0 board could be damaged when the signal is sent through for processing

5.3.5 Evaluation

Overall, the design works effectively for the specification set. When the array is placed over the rock, a high voltage level is measured (fig 5.12) and the time between these rising edges can be identified by the Arduino that determines the frequency of the incident wave and hence the correct mineral can be determined (fig A.6).

All problems originally outlined were overcome, however improvements could be made. The max range of about 5cm over the rock casing's hole limits the rover's structure. Further amplification could resolve this, but more noise cancellation methods would be required.

Even with the array the infrared can only be detected in a thin beam. To maximise this area of detection more transistors could be added and arranged in a more efficient manner like being spread out more. But this was believed unnecessary as it would increase cost and the size of the rover to not much benefit

5.4 Acoustic

5.4.1 Design and Planning

Some of the EXOROCKS emit a 40kHz sound. In order to detect this sound, the 400SR160 Transducer Ultrasonic Receiver was selected as the ideal sensor. This receiver was selected as it has a very high sensitivity, a wide bandwidth and is designed to detect this acoustic frequency range.

5.4.2 How the transducer works

The receiver is designed to oscillate mechanically at 40kHz. When the sound is around the central frequency, the amplitude of the sine wave is large due to resonance as shown in the figure 5.13.

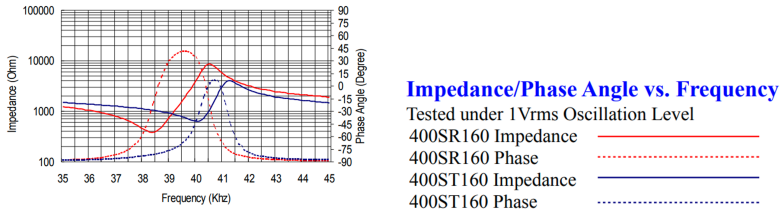


Figure 5.13: Acoustic bode plot [17]

5.4.3 Prototyping and Testing

Our target is to measure the frequency of the Square Wave from the Schmitt Trigger. When the 40kHz sound is detected the amplitude will be significantly larger than the amplitude of the existing noise. As shown in Figure 3, below 2.5 Volts, which is considered amplified noise, the Schmitt Trigger goes to zero. Otherwise, the Schmitt Trigger will output approximately 5 Volts, creating a Pulse Wave. For the simulation in LTSpice, the receiver was replaced with the sine(2.5 20m 40k), because it matches the output of the transducer in the lab.

Circuit:

Part 1: Potential divider This part of the circuit works as a potential divider. As is shown in the graph above the impedance of the transducer is 10kOhm when the sound has a frequency of 40kHz and drops as the sound has different frequencies. The output voltage of a potential

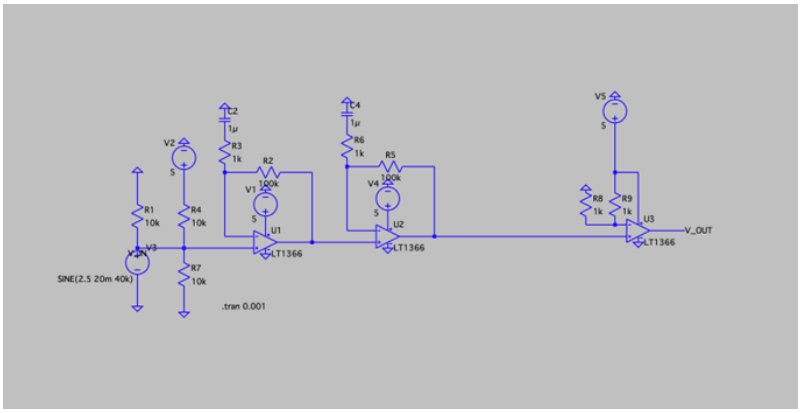


Figure 5.14: LTSpice Schematic of the receiver

divider is:

$$V_{out} = V_{in} * R_{transducer}/(R_{transducer} + R1).$$

Since the impedance is the same as R1 when the sound is around 40kHz the Vout = ½ Vin but in any other case the Vout is much smaller.

Part 2: Biasing The Vin+ pin of the first amplifier is biased at 2.5Volts, which is half the voltage of the power supply. This is done, in order to provide enough head and leg room for the signal not to be cut, i.e. distorted.

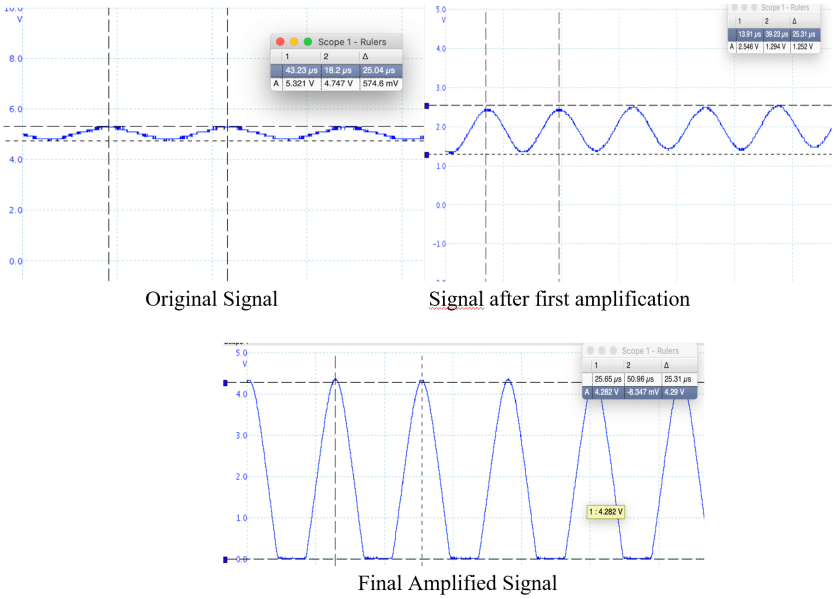


Figure 5.15: Signals during each stage of amplification

Part 3: Amplification The amplitude of the AC component depends on the frequency and the amplitude of the soundwave. As explained above when the frequency is different from the central frequency of the receiver the amplitude tends to zero. When the soundwave has the desirable frequency the amplitude of the AC component is 500mV, which is too low for the Arduino to detect. We therefore perform two stages of amplification, since the MCP6002 that is being used does not have a high enough gain bandwidth product (GBP).

Part 4: Schmitt Trigger After the two stages of amplification, the noise (detected without the presence of a 40kHz sound) was also amplified. This brought the need for a Schmitt Trigger, so as for the noise not to be mistaken for the 40kHz sound, since this way the amplitude of the noise will always be less than the threshold voltage set. The problem is caused due to the fact that the transducer is designed in a way that its noise has the same frequency. In the figure 5.16, we can see that when the Rock is close to the transducer, and the Voltage exceeds the threshold of 2.5 Volts, the Schmitt Trigger goes to 5 Volts, otherwise it drops to 0 Volts. This way the Arduino can detect the existence of the 40kHz sound, as it measures the frequency of the pulse wave.

The shape of the Schmitt Trigger however, appears triangular. This happens for two reasons. Main reason: due to the slew rate, which is basically how quickly the opamp can change voltage.

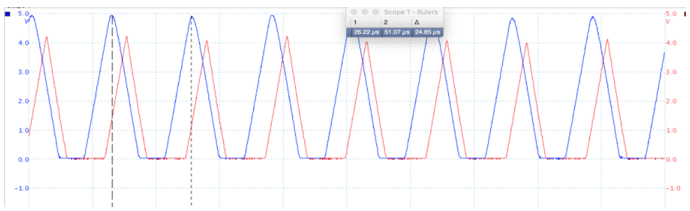
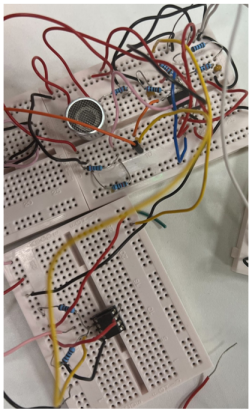


Figure 5.16: Amplified signal (Blue)-Schmitt Trigger output (Red)

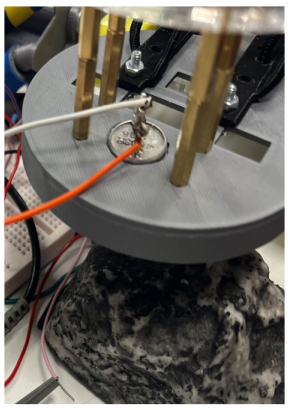
As described in the IR section of the report, the higher the slew rate, the more the final waveform appears to be square. The Opamp we have used has a small slew rate. To solve this we can attempt to switch opamps.

Second reason: The Schmitt Trigger has a threshold voltage of approximately 2.5 Volts, which is very close to the upper voltage (5 volts). Therefore, the pulse does not have enough time to become a square wave as the period of time for which the signal is greater than 2.5 volts is relatively small. Therefore, once it reaches the 2.5Volts, it shows a small increase and then almost immediately starts to fall.

5.4.4 Implementation- Improvements



Picture of prototype circuit.



Final Implementation of Receiver on holder.

Figure 5.17: Pictures of the built circuit

The circuit at this point works very well, and is able to detect the existence of the 40kHz sound from a distance of approximately 10cm, without the need of placing it directly above. After the final testing for the performance, the circuit will be soldered on acrylic. This way the size is reduced and the probability of an error due to a faulty contact is minimized.

5.5 Magnetic

5.5.1 Design and Planning

The objective is to identify the direction of the static Magnetic Field, inside the EXOROCK. During the first research, three types of Sensors were considered. The first one onsisted of two Hall Effect Sensors (one for each direction), which only output voltage when the correct polarity of the magnetic field is present. Despite these sensors being low-cost, the model was rejected due to the need for the output to be amplified greatly to detect the magnetic field from a distance, while also being very susceptible to noise. The second option was the use of A1262: 2D, Dual-Channel, Ultrasensitive Hall-Effect Latch, which offers values for two components (axis) of the magnetic field. As an extension to this, a better implementation is apparent with the *Adafruit Triple-axis Magnetometer - LIS2MDL*, which has very high sensitivity and allows the 3-axis display of the components of the magnetic field in a vector diagram.

The magnetic sensor is directionally independent. As a result, the magnetic field from all directions can be detected. This is not the case for the Hall Effect Sensors, which are able to detect in one direction only, thus introducing the need to align the Sensor with the source, a constraint which can be avoided.

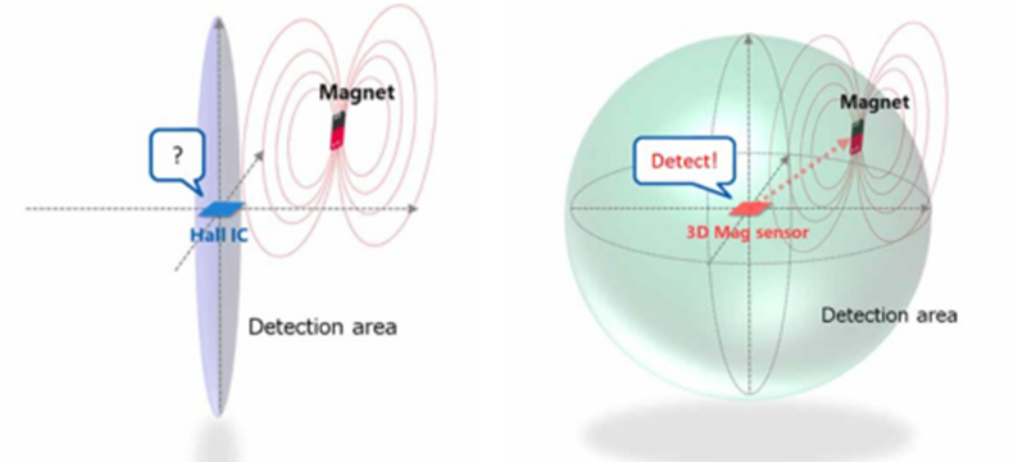


Figure 5.18: Difference between the ability to detect the magnetic field, between the Hall Sensor and the Magnetic Sensor [18]

5.5.2 Prototyping and testing

The *Adafruit Triple-axis Magnetometer - LIS2MDL* from the beginning showed great accuracy in the measurements and did not show the need for calibration. The measurements have the Gauss unit for each direction, with a sensitivity of $\pm 1.5\, mGauss = 0.15\, \mu Tesla$, suitable for the strength of the magnet in the rock. The magnet offers all three components of the magnetic field (x,y,z). The z value establishes the direction of the Magnetic Field of the rock (UP, DOWN, Non-Existent).

The testing revolved around the use of a magnet, glued to a plastic base with a similar thickness to the rock.

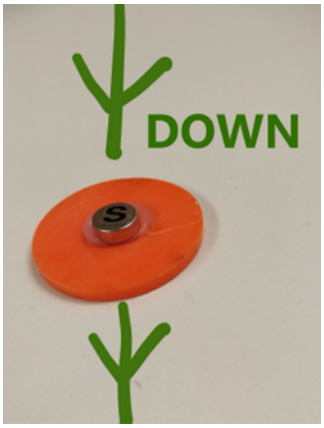


Figure 5.19: Magnetic Field of Rock simulator. By switching sides on the magnet, values for the UP direction are taken

5.5.3 Testing

Step 1: Use of Sensor without the magnet sample.
The Magnet Monitor measured the z value to be approximately 83 uT, basically a combination of the earth's magnetic field(20-60 uTesla) and other surrounding sources. At the beginning of the program, the average magnetic field after 20 iterations was computed.

$$x_{avg} = -11.65,\ y_{avg} = 42.97,\ z_{avg} = 83.45$$

Step 2: Use of Sensor with the magnet sample
The presence of the magnet caused a change in the z values

$$-113.70 \quad -118.95 \quad 308.40 \quad 311.25$$

The initial plan was to compare the Z values to two predefined thresholds for Z_{UP} and Z_{DOWN}. However, the average value showed slight changes, depending on the surrounding environment. This brought the need for adjustment in the code during each use. For that reason, the difference from the average value was measured instead.
The variation of the Z values from the magnet without the existence of the magnet led to the definition for the existence of the magnet to $25\mu T$. This would reduce the error in the classification of the direction of the magnetic field.

Implementation and Building

Due to the use of SPI pins from the wifi module, upon discussion, I2C communication protocol was used. The precision in the measurements is largely dependent on the placement of the sensor, ideally positioned horizontally and directly above. For this purpose, a 3d printed Holder was designed and implemented, shown in Figure 5.20 [19].

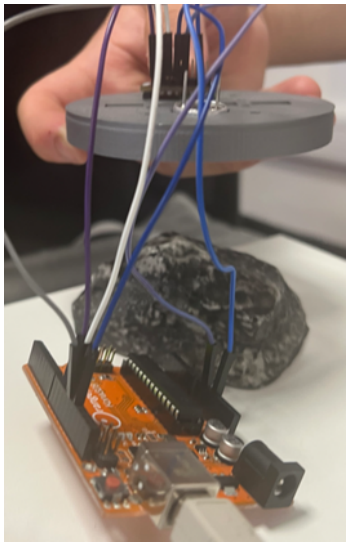


Figure 5.20: Depiction of Holder and positioning of the magnet above the Rock.

5.5.4 Implementation of 3d Graph

As mentioned in the Design and Implementation section, the strength of each component of the magnetic field is investigated using this sensor. To get a more intuitive view of how the magnetic field changes with the presence of a magnet, a Python program for a 3D plot was implemented, see Figure 5.21.

In the plot, each vector has the same starting point (0,0,0) and a different endpoint depending on the direction and the amplitude measured by the sensor. The Arduino periodically reads the measurements from the magnetometer and sends them back using serial communication. The python script reads each different measure and plots them in the graph. As expected, the closer the magnet to the sensor, the greater the magnitude of the vector, as depicted in the graph. The measurements in figure 5.21 were taken by placing the magnet below the sensor, the same way the rover is going to approach the rock, to simulate the steepest change in Z. It is also possible to investigate the effect of the magnet on the other two axis by placing the magnet in different positions relative to the sensor since the direction of the magnetic field changes at each point around the magnet as in figure 5.22.

The goal is to have live updates of the magnetic field in the webserver. In order to achieve this, wireless communication is necessary. As it will be shown later in chapter 6.3.2, the optimal

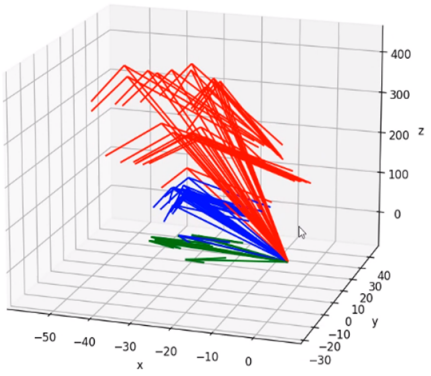


Figure 5.21: 3d Vector Graph of each component. UP direction is represented in red, DOWN in green, and No Magnet in blue.

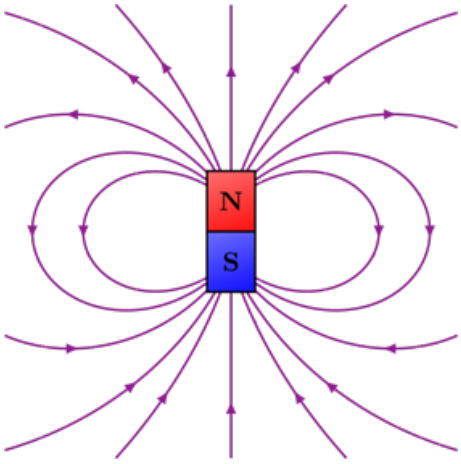


Figure 5.22: Magnetic Field of a Magnet

approach is the UDP method. However, there is a problem with the integration of the 3D plotter. There are two possible solutions. The first is to have the Python Script pull data from the Arduino, create the graph and save it as a picture/gif which is going to be shown on the website. The second is to send the data directly to the server and use a React JavaScript library to plot the graph. With further testing, it will be determined which method is better for our system.

5.5.5 Evaluation

The final design, after much testing, proved very effective and accurate in the detection of the direction of the magnetic field from a fair distance (up to approximately 12cm away). Even though the thickness of the rock affected the amplitude of the measurements, the design was still able to distinguish between each case. The extra implementation also allowed the depiction of the Magnetic Field direction and magnitude. The next task is the integration of the 3d Plotter in the server.

Chapter 6

Software

6.1 User Interface

6.1.1 Initial Attempt

The first approach to building the user interface (UI) to drive the Rover was based on the starter code provided in [20]. Building on the code, the website was expanded to have functions to control the motors, allowing for moving and steering of the Rover.

Very soon the problem of file size was encountered, where, after adding more functionality to the web-page in the form of more functions or additional components, the web server would crash when trying to serve the web-page. After some research the culprit for this phenomenon was found in the application note of the `WINC1500PB` [21] which is the micro-controller used by the Wi-Fi module provided:

> **(2.2.2 TCP Packet Size):** A packet is sized as per the lowest MTU (Maximum Transmission Unit) on the path, which is typically1500 bytes (for Ethernet) of data including TCP (20 bytes) and IP headers (20 bytes). The packet size is defined in `main.h` as :
>
> ```
> #define MAIN_WIFI_M2M_BUFFER_SIZE 1460
> ```

This means that the web-page must be smaller than `1460 bytes` in order to fit within one single TCP packet. Alternatively the web-page could be split into chunks and served separately. Both of these solutions add more complexity than required and can be avoided by hosting and serving the website locally on the client's device.

6.1.2 Electron + React

Free from the `1.5KB` limit on the website size, alternative solutions can be explored for the building of the UI. The solution chosen was to build the user interface using the React framework [22], a JavaScript library made for building website UIs. This would still require a server to be ran in order to serve the website, however the Electron framework [23] was used to make the website into a cross-platform desktop app. Electron runs both a front-end, which in this case is comprised of the React website, and a back-end, which uses the Node.js run-time environment. This opens up the possibility of using alternative methods of communication.

6.1.3 UI Design

To control the movement of the rover as well as receive the information response in a clear and concise format a Clean UI that would display the information to the given user is needed. To do

this react components were used, formed from JSX, a form of JavaScript and HTML that uses states to update the DOM of the react website giving, flexible, reusable and easy to create UI elements.

Therefore, a layout of each page of the react app using basic block shapes is formed and worked out with what components are needed to make each of the pages and what would be included within each.

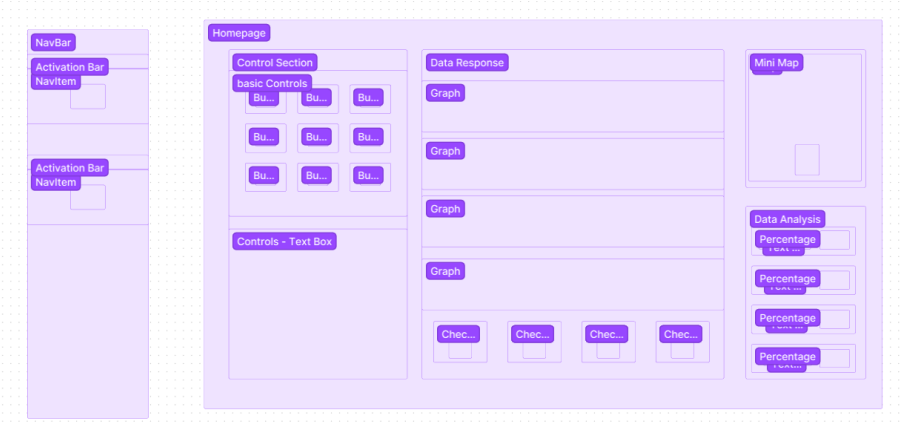


Figure 6.1: Concept Design Images

The first element created was a page navigation bar, to do this a new component that utilises the react-router library is created, this gives a series of buttons, each with a page associated with them. Additionally, adding CSS styling with SVG icons ties the look together.

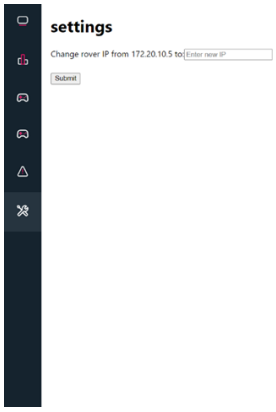


Figure 6.2: Navbar Of Electron App

Next, moving onto creating some of the motor control components a basic discrete control is formed from a directional pad of buttons, (with Mecanum wheels this includes 7 buttons, each cardinal direction, stop and turn left and right) along with a test connection button which would send a single test message and log the response.

After the controls had been added to the motor control page, a console that would log the responses of each sent message and the received response was created. For this nested DIVS that contained a Span for each message was used, this gives us a large amount of control with the style of each response and how to include a large amount of data about each message.

Also using the basis of this console component to create more consoles for each data type and each response giving us an effective way to debug each section individually. On top of this it also allows communication with the backend to save the data to give error logs in case of an issue or to check back at each sent message and the received response.

After adding graphs for each form of data, these were added to the home page along with motor control buttons, and some test connections. These would update live using Apex charts for Magnetic, Infrared and Radio, giving a live view of each type, along with the data analysis,

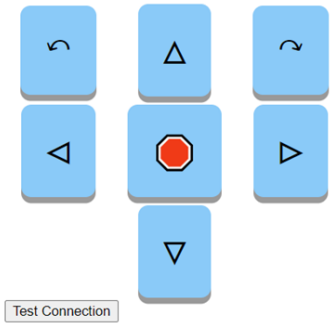


Figure 6.3: Discrete Motor Controls

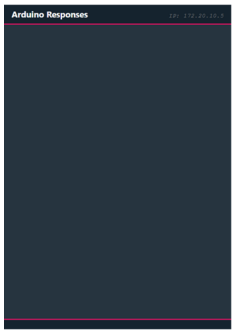


Figure 6.4: Console For Message Responses

this data can produce a small set of boxes containing which rock we think is the most likely to be scanned at a given moment.

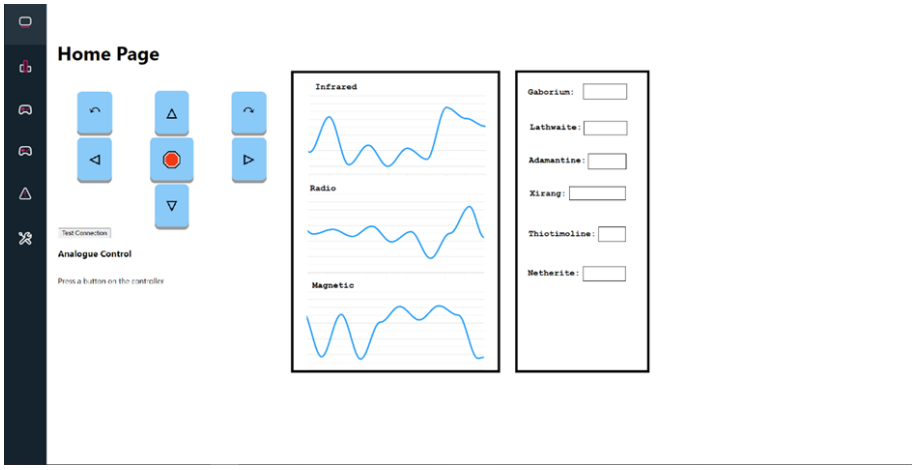


Figure 6.5: Working Basic Homepage

6.1.4 Controller Logic

For the Controls two separate methods were implemented, first discrete controls that used a simple react component and secondly an API that allowed the use of either a PS4 controller or a Xbox controller, and encoded the movement into directional vectors.

The discrete controls take a direction from the given coordinate button and use that to send a move message for each, at the maxium speed. This allows for both turning in each direction as well as movement in each cardinal direction, but no fine grain adjustment.

To allow for more complex movement the use of a controller is needed, this was done using

the Gamepad-API. This polls the joystick at a set rate to retrieve button and joystic inputs. This is important as this can then be used to limit the amount of messages being sent by the client App to the Rover. The joysticks give a coordinate response each time it was polled, to process this we first calculate the magnitude of the two coordinate systems as a vector, if this value was over a set threshold value from the last sent message, it updates the new location and sends a UDP message with the given direction to the rover.

This process runs for both joysticks individually, allowing for both a turning vector and a directional vector when processing motor speed and direction.

6.2 Communication Protocol

6.2.1 Network Topology

The communication between the client and the Rover happens over the `EEERover` network, where the `Metro M0` micro-controller connects to the network through the `WINC1500` Wi-Fi shield. The `M0` is then connected to and `OrangePip` which acts as the motor controller, as detailed in the Mechanical section?. The client-side app will therefore send the codes which will then be relayed to the motor-controller to actually drive the wheel. (Justify this in the chassis design section) The `M0` also measures the signal coming from the sensors and has to relay those to the client-side app to be displayed in the UI.

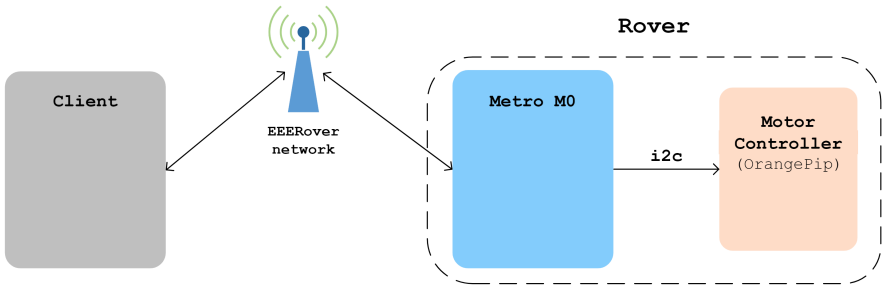


Figure 6.6: Devices in network

6.2.2 HTTP

As a first approach `HTTP` `POST` and `GET` requests were used to communicate with the Rover. Having generated the correct code to drive the Rover in the desired direction, the `fetch` API [24] was used to make a `POST` request to the Rover, with the direction code as the request body. This method worked well when the website was hosted on the Rover, where the domain of the host (the `M0`) matched with the domain the requests were being made to. Once the website was moved to being self-hosted on the client device, the website was making requests to a domain different to its own, resulting in a `CORS` (Cross-Origin Resource Sharing [25]) error, since the browser considered this a security violation. To amend this issue two changes had to made:

1. Add an `Origin:` header to the request being made by the client, whose value is the clients' domain.
2. Add a `Access-Control-Allow-Origin:*` header to the response coming from the Rover which indicates that requests from all domains are allowed.

Once the requests were being successfully made the Rover motor controls were working very well, with the `Xbox controller` allowing for very fine-grained control of direction and speed. However it was noticed that the controls did not feel very responsive. By timing the requests made, it was

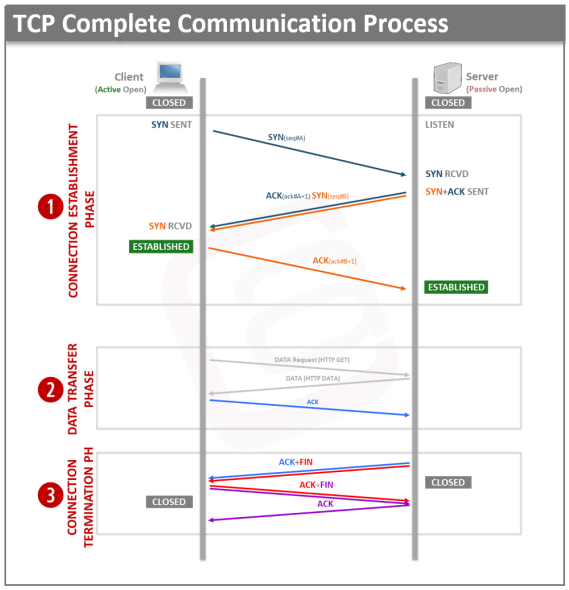


Figure 6.7: TCP communication process [26]

found that the latency averaged `100ms` reaching a maximum of `150ms` if many requests were being made in rapid succession, such as in the case of the `Xbox controller`. This meant that requests could not be sent at intervals smaller than about `200ms`, and even in that case if many successive requests were sent they would form a backlog, slowing down the process even more.

Researching the `HTTP` protocol it was discovered that in order to make a successful request and receive a response, many data packets are exchanged between the server and the client, as can be seen in Figure 6.7. If an alternative communication protocol which used fewer packets were to exists, it would reduce the latency significantly and improve the responsiveness of the Rover control.

6.2.3 UDP

User Datagram Protocol (`UDP` [27]) is a simpler communication protocol compared to the Transmission Control Protcol (TCP) used by HTTP, however it doesn't guarantee that the data sent has arrived to the desired destination nor does it provide any confirmation of data being received. UDPs' main advantage stems from the fact that it send very small data packets, with a `Header` of just `12 bytes`, and it only sends one packet, without making a three-way handshake as TCP does. This all means that communication, if successful, will be much quicker.

After verifying UDP communication works on the `Metro M0` using the `WiFiWebServer` library [28], the protocol had to be implemented in the client-side app. Since `UDP` is a low level protocol it cannot be implemented in the `React` front-end framework and has to be implemented in the `Node.js` back-end. `Node.js` provides the `dgram` library [ADD CITATION] to create `UDP sockets` over which messages can be sent and received. The relevant methods from the `socket` class are:

- `send(message)`: Sends the message
- `on`:
  - `on('message', callback(message))`: When a message is received it calls the callback passing the received message as a parameter.
  - `on('error', callback)`: Calls the callback if any error is encountered
- `close`: Closes the socket

In order to distinguish between different types of messages a single `char` is prefixed to the message according the Table 6.1.

Prefix	Type of Message
m	Move
d	Data
t	Test
e	Error

Table 6.1: UDP message encoding

In order to measure the latency of the protocol, and to introduce a basic message success verification, the M0 sends a UDP message back containing the same message it received. This then allows the client-side app to verify that the message it sent out was indeed received and processed correctly, as well as to time the amount of time between the sending and receiving. The process in outlined in Figure 6.8.

With the new UDP protocol the latency is greatly improved compare to HTTP averaging about 20ms, with peaks of 30ms. This unfortunately does not mean that messages can be sent to the M0 every 30ms since while testing it was discovered that as the frequency of messages sent increases the micro-controller cannot process them quickly enough, and a backlog of messages forms, where the time to process each messages increases greatly reaching even 3 seconds or more. After more testing the ideal period between messages was found to be about 130ms, a great improvement compared to the HTTP method. At this frequency the latency is very small, about 20ms, meaning once the input is given from the client-side app it is almost instantly converted into motor movement.

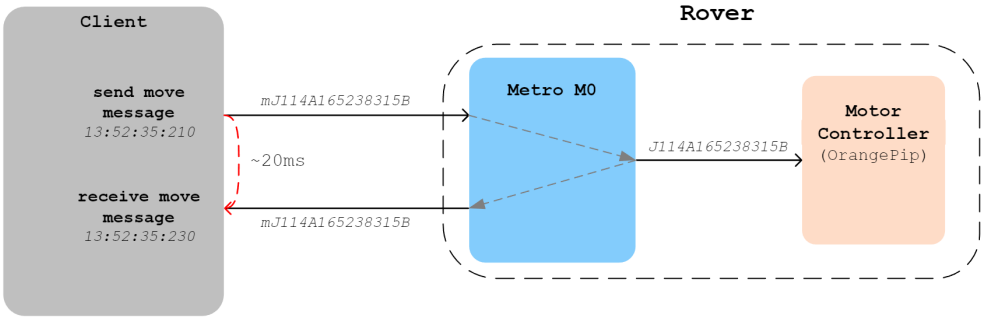


Figure 6.8: UDP move message example

6.2.4 Data Communication

Another great advantage of the UDP method as opposed to HTTP is that it allows the M0 to send data to the client-side application at any time, without the client having to request it first. This is compared to HTTP where in order for data to be sent from the server (M0) to the client, the client must request it first. Using this approach to receive a continuous data stream from the Rover would lead to continuous polling of the M0, placing a burden on the micro-controller and therefore limit the frequency of the measurements. Some basic testing revealed this would have to be about once a second. With UDP the M0 can send data over at will, and testing showed that even when sending data every 100ms the performance was not affected.

6.3 Sensor Measurements

6.3.1 Radio, Infrared and Acoustic

The Radio, Infrared and Acoustic sensors all produce square wave signals at a certain frequency which needs to be measured. To achieve this without slowing down the micro-controller Interrupt Service Routines (ISR [CITATION arduino website]) are used. Interrupt routines are functions

which get executed whenever an interrupt is triggered, stopping whatever process was being executing in that moment. The `Metro M0` has 16 external interrupts, meaning all pins expect for `D4` can be used as inputs for the sensors, the pins used are described in Table 6.2.

On the *rising edge* of the sensor input the interrupt is triggered, calling an ISR which increments a counter for that particular sensor. In the main loop the `millis()` function [CITATION] is used to check if a set amount of time has passed, if so, the value of the counter is divided by how much time has elapsed to determine the frequency of the signal. The signal is then sent to the client-side application.

Radio

The Radio signal technically encoded two different frequencies: *carrier frequency* and *demodulated frequency*. The final output of the sensor is the demodulated signal from which the frequency is measured, however the modulated signal is amplified enough to be able to be measured by the `M0`, allowing for measurement of the Carrier frequency. This is done by using an additional two interrupts:

- *Falling edge* on the demodulated signal input
- *Rising edge* on the carrier signal input

Since the carrier signal is only large enough during the *on-time* of the demodulated signal, that is the only period during which it should be measured. To achieve this a Boolean variable is used, when true the counter is incremented on the rising edge. This Boolean variable is set to true on the *rising edge* of the demodulated signal and is then set to false on the *falling edge*. Using this method carrier frequency can be measured to a sufficient degree of accuracy, and allows for redundancy in the measurements.

6.3.2 Magnet

The magnet sensor can be interfaced with either `SPI` or `i2c` but since the `WINC1500` Wi-Fi shield is already using the `SPI` bus, `i2c` was chosen to avoid any conflict.

6.3.3 Data encoding

The 4 sensors each produce different data which is converted to a string format according to the Table 6.2 and sent to the client-side app according to the format in Figure 6.9.

Sensor	Data produced	Number of Chars	Start Char	Pins Used
Radio	Signal 151Hz - 351Hz Carrier 61kHz - 89kHz	3 2	R	8 6
Infrared	352 Hz - 571 Hz	3	I	2
Acoustic	40 kHz	2	A	3
Magnet	3 axis 0 - 400	4	M	SDA, SCL

Table 6.2: Sensor Data Encoding

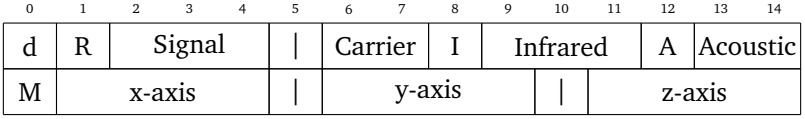


Figure 6.9: Data encoding for UDP packet

6.4 Discussion

6.4.1 Evaluation

Overall the app functions effectively with an efficient protocol to send messages with a low average latency of around 50ms along with controller support for fine grain control for the user, with both movement and turning being a focus.

Along with this the app has other features that make it very easy to interact with for example a settings page lets the user update the IP for both the local machine to send UDP packets as well as the remote IP that the UDP packets are being sent to.

Additionally included with are the Error logs that are saved and logged in a folder for the app, giving users the ability to view and monitor the activity of the packets that are being sent and received in-case of any bugs that may occur.

6.4.2 Future Work

- To increase processing speed of m0 data to allow more packet communication giving higher speed of controller polling
- Change the return data to communicate using chars instead of pure binary as a single char can act as an 8 bit binary number and stringing together will use less bandwidth
- Update the UI to look cleaner and update faster, with an alternative to using the JSON file format for error logs

Chapter 7

Future Work

Given more time, the model could be modified to look more streamlined and professional. For example, wires could be tidied up, re-cut and soldered to more appropriate lengths that fit within the rover and to reduce risk of faulty contacts. In addition, it is planned to transfer all circuits to PCB's to be mounted onto the chassis for a more permanent design. Ordering new quicker motors would be beneficial as the current ones limit the maximum speed. To be more space and power efficient, it was proposed to remove the motor controller Orangepip and operate everything with just the Metro board.

Chapter 8

Conclusion

The expectations for the project were to create a driveable rover capable of accurately classifying multiple rocks with a range of characteristics. These requirements were not only met but surpassed, as the rover can identify all specified rocks solely from the first property, but also detects precisely the second properties too. With the sensors working in multiple environments consistently, the rover is definitely reliable. It is fit to navigate harsh landscapes with large wheels, which would be effective at gripping the rocky terrain, and simple remote controls. After cutting down on the weight and size of the rover, the final implementation is quick but still robust being able to cope with 10N of applied force with our 3D printed supports. While wire organisation currently is messy, it will be cleaned up before the demo but still the rover is aesthetically pleasing and could be marketed well to potential buyers.

Appendix A

Appendix

A.1 PDS

A.2 Company Constraints

Some of the constraints that we may face could come from both the budget and the access to materials that we have as a small student group, when trying to use certain materials or components. Another limit that comes with the project is the amount of people working together, for example our group has 7 people limiting the quality of certain aspects such as the website, as with a larger group we could have a larger number of people adding features.

A.3 Quality and Reliability

The main aspects of quality that are important are those of the sensors and movement systems such that the main task of identifying each rock can be completed easily and accurately with minimal issues. The reliability is also quite important as each sensor needs to have a consistent response from each type of rock, that can be processed easily from the Arduino or computer.

A.4 Life in services

Since the rover should run on the moons surface the life in service is vital to the quality of the product as it needs to last in service for as long as possible to increase the cost to usage ratio. A good example of this is the moon rover opportunity which lasted for 15 years of service while only was designed to last for 93 days. [29]

A.5 Maintenance

As the rover will be purposed for use on the moon surface it should be built to need minimal maintenance and have a large life in service, during the prototyping phase on the other hand it should be easy to quickly change and modify such that new designs can be tested quickly while planning the final form.

A.6 Competition

The main competition that are group face, is the other groups that are also producing rovers, some aspects of competition we may face are the limited supply of certain components as well as limited internet bandwidth of the rover network. Finally, the main competition will take place during the trials in which they are timed against each other.

A.7 Shipping

The shipping isn't easily covered by the scope of the project but is still affected by somethings such as the size and mass of the rover itself can change the cost of package as well as its difficulty to produce especially when the rover is quite delicate.

A.8 Packing

The packing of the rover faces two main issues, the size and the shape of the rover, generally for a rover to be well packaged it needs to be of a small size along with a simple and strong shape such that a packaging can be easily designed for its transportation.

A.9 Quantity

We only need to produce a single rover, but we may produce a few prototype rovers during the design and implementation phases, as well as the website should be easily usable and downloadable for any device.

A.10 Manufacturing Facilities

We only have the facilities that are available to use within the lab, for example a laser cutter, 3D printer as well as oscilloscopes and power supplies, but we are also limited by the amount of time that these may be in use from the other groups also needing time to develop their own parts.

A.11 Size

The size of the rover is very important to the overall effectiveness of the rover's movement and most likely its relation to its weight when being considered with the different materials used during construction are greatly affected by the size of the rover. Another aspect of the size is how stable the rover is, generally a larger rover will have a wider base and such a lower center of gravity making it more stable for use when moving over rough terrain. As such a balance needs to be found with it being stable enough to scan easily without movement issues while also maintaining a lightweight and fast pace around the test area.

A.12 Life Span

If used in space/moon, due to a lack of atmosphere, the main contribution to deterioration would be radiation decay from UV, which may deteriorate plastics and coatings like on the infrared sensor. However, there would be less oxidation damage coupled with this [30].

A.13 Standards and Specifications

As the rover is to be used in space it should follow international standards such as the international space law.

A.14 Customer

The customer for the rover would be that of an aerospace company or government agency that would send it onto the moon surface to do rock scouting. Another example of a customer may be that of a school or education branch who would like to use if for education purposes

A.15 Shelf Life

The shelf life of the rover needs to last for a decent amount of time as it may be left stationary or without use for long periods of time before usage or during transit, as well as during its usage on the moon's surface where it may be unused for days or weeks as it isn't required but needs to stay ready for use.

A.16 Timescale

After receiving the report, we had around a month to complete both the design and implementation of our design as well as the report of the rovers' design process and creation. After the report is submitted, we also take place in a timed trial run of the rover on the 24th of June.

A.17 Safety

Although the rover is small, it should still have reasonable precautions taken place, for example warnings should be used to show areas where the battery is located and may cause a hazard if damaged along with reasonable warnings with the controls to give reasonable guidance to a user.

A.18 Patents Literature and Data

A large amount of research should be taken into account when finding what other patents may interfere with the design and production of the rover, some examples include, the mars rovers produced by NASA, [31] [32].

A.19 Political and Social Implications

The general stability of the space market is very strong as it doesn't fluctuate much with many other businesses, the main effect is that of defense and government spending.

A.20 Legal

A product that is to be sold to other companies for large scale use, requires long term management and warranty such that the product can be integrated into the required environment effectively as well as the product should have available support for if the rover breaks down or needs maintenance.

A.21 Installation

In general, the installation should be quite easy only requiring the rover to be placed into the required position, for the moon surface this may require more precise work, but it is not covered in the scope of the project.

A.22 Documentation

For documentation there should be extensive included guides that explain how to use and navigate the app for using the rover along with how to setup the rover such that it can be used easily, and quickly get into a position of effective use.

A.23 Market Constraints

The main market constraints that we face are the lack of aerospace companies as well as their low demand, so to create an effective product it must be highly targeted to each company with a pitch or interview demonstrating its effectiveness, as well as be high quality such that a company would rather reach out to our third party than create a similar product in house.

A.24 Images

Figure A.1: Team Gantt Chart

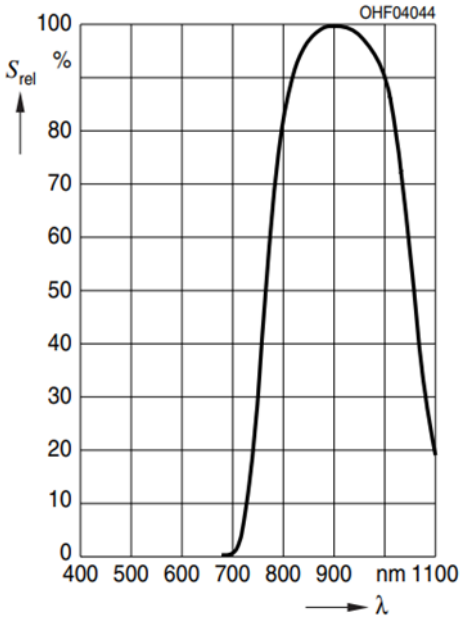


Figure A.2: showing the transistor's wavelength characteristics [CAPTION]

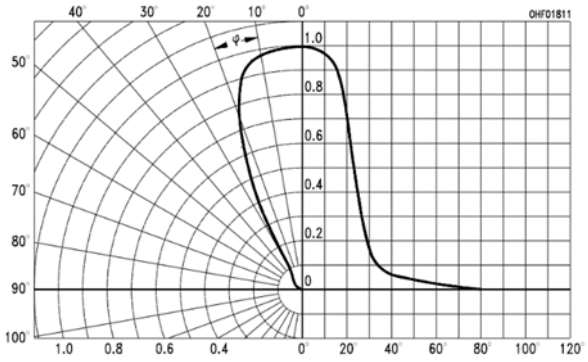


Figure A.3: ir angle [CAPTION]

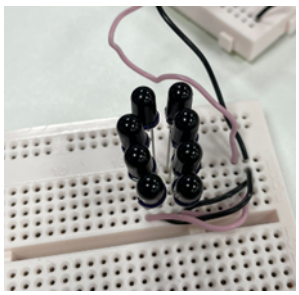


Figure A.4: ir array

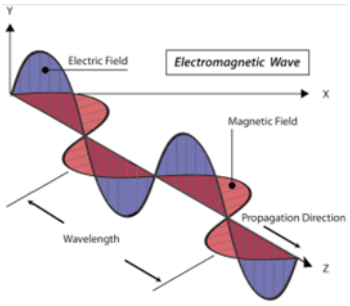


Figure A.5: Electromagnetic Wave

```
infrared freq: 570.00Hz
infrared freq: 570.00Hz
infrared freq: 570.00Hz
infrared freq: 570.00Hz
infrared freq: 575.00Hz
infrared freq: 570.00Hz
infrared freq: 570.00Hz
infrared freq: 575.00Hz
infrared freq: 570.00Hz
infrared freq: 575.00Hz
infrared freq: 570.00Hz
```

Figure A.6: Arduino measuring the frequencies

Product	Quantity	Bought From	Description	Unit Price / £	Total Price /
3D Printing Sensor Holder	2	Lab	Chassis and Frame	0	£0.00
3D Printing Sensor Plate	1	Lab	Chassis and Frame	0	£0.00
3D Printing Mecanum Wheel	4	Lab	Chassis and Frame	0	£0.00
Heat Shrink Pipe for Wheels (15mm)	32	Amazon	Chassis and Frame	0.22	£7.04
Motor	4	EEEBug Kit	Chassis and Frame	0	£0.00
Motor Mount	4	EEEBug Kit	Chassis and Frame	0	£0.00
Motor Drive PCB	2	EEERover Kit	Chassis and Frame	0	£0.00
Power PCB	2	EEEBug Kit	Chassis and Frame	0	£0.00
Orangepip (Arduino UNO)	1	EEEBug Kit	Chassis and Frame	0	£0.00
Adafruit Metro M0 Express with WIFI Shield	1	EEERover Kit	Chassis and Frame	0	£0.00
Battery Holder	1	EEEBug Kit	Chassis and Frame	0	£0.00
Battery (1.5V)	4	Lab	Chassis and Frame	0	£0.00
STRIPBOARD MEDIUM 95mm X 127mm	2	EEEStore	Chassis and Frame	1.7	£3.40
STRIPBOARD SMALL 25mm X 64mm	1	EEEStore	Chassis and Frame	0.24	£0.24
3mm Clear Acrylic 600 x 300mm	2	EEEStore	Chassis and Frame	5.19	£10.38
1K ohm resistor	8	Lab	Chassis and Frame	0.06	£0.48
Wire	20	Lab	Chassis and Frame	0	£0.00
M3 Screws and Nuts	48	Lab	Chassis and Frame	0.06	£2.88
SFH 300 FA	8	Farnell	Infrared	0.54	£4.32
MCP6002	2	Lab	Infrared	0.61	£1.22
Resistors	10	Lab	Infrared	0.06	£0.60
Capacitors	12	Lab	Infrared	0.09	£1.08
Wire	10	Lab	Infrared	0	£0.00
NE5532AP	1	Cricklewood Electronics	Radio	1.8	£1.80
MCP6002	2	Lab	Radio	0.61	£1.22
Resistors	14	Lab	Radio	0.06	£0.84
Capacitors	7	Lab	Radio	0.09	£0.63
Wire	15	Lab	Radio	0	£0.00
LIS2MDL Magnetometer	1	Pimoroni	Magnetic	5.7	£5.70
Wire	4	Lab	Magnetic	0	£0.00
400SR160 Transducer	1	Farnell	Acoustic	7.93	£7.93
Resistors	9	Lab	Acoustic	0.06	£0.54
				Total:	£50.30

Figure A.7: Group Budget

Bibliography

[1] B. Venditti. The cost of space flight before and after space. [Online]. Available: https://www.visualcapitalist.com/the-cost-of-space-flight/

[2] Space. [Online]. Available: https://www.space.com/18175-moon-temperature.html

[3] perfect mecanum wheels with any size. [Online]. Available: https://grabcad.com/library/perfect-mecanum-wheels-with-any-size-1

[4] H. Taheri, B. Qiao, and N. Ghaeminezhad, "Kinematic model of a four mecanum wheeled mobile robot," *International Journal of Computer Applications*, vol. 113, no. 3, p. 6–9, 2015.

[5] V. Boddy, a talk with Mr. Victor M Boddy.

[6] "What is generative design: Tools software," Dec 2021. [Online]. Available: https://www.autodesk.com/solutions/generative-design

[7] *LT1366*, Linear Technology. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/1366fb.pdf

[8] *MCP6002*, Microchip Technology Inc. [Online]. Available: http://www.mouser.com/datasheet/2/268/21733e-41017.pdf

[9] *NE5532AP*, Texas Instruments. [Online]. Available: https://www.ti.com/lit/ds/symlink/ne5532a.pdf

[10] E. Stott, "Eeerover/readme.md at main · edstott/eeerover," 2022. [Online]. Available: https://github.com/edstott/EEERover/blob/main/doc/README.md

[11] K. B. R. Cafe, "Understanding electromagnetic wave physics," Mar 2020. [Online]. Available: https://www.rfcafe.com/references/electrical/electromagnetic-wave-physics.htm

[12] E. Engineering360, "Rf inductors information," 2022. [Online]. Available: https://www.globalspec.com/learnmore/electrical_electronic_components/electronic_components_passives/inductors/rf_inductors

[13] E. Classroom, "Hysteresis loss and eddy current loss and their difference," May 2021. [Online]. Available: https://www.electricalclassroom.com/hysteresis-loss-and-eddy-current-loss/

[14] A. A. Circuits, "Coil inductance calculator - electrical engineering amp; electronics tools," 2022. [Online]. Available: https://www.allaboutcircuits.com/tools/coil-inductance-calculator/

[15] K. Leung, "Ee1: Introduction to signals and communications," 2021.

[16] E. ElProCus, "Phototransistor : Construction, circuit diagram amp; its applications," Feb 2021. [Online]. Available: https://www.elprocus.com/phototransistor-basics-circuit-diagram-advantages-applications/

[17] F. Farnell, 0AD. [Online]. Available: https://www.farnell.com/datasheets/3109335.pdf

[18] A. Kasei, "Why use tri-axis magnetic sensors ?: What makes akm different ?: Tri-axis magnetic sensors: Products: Asahi kasei microdevices (akm)," 0AD. [Online]. Available: https://www.akm.com/eu/en/products/tri-axis-magnetic-sensor/what-makes-akm-different/why-use-tri-axis-magnetic-sensors/

[19] B. Siepert, "Adafruit lis2mdl triple axis magnetometer," 0AD. [Online]. Available: https://learn.adafruit.com/adafruit-lis2mdl-triple-axis-magnetometer/arduino

[20] E. Stott, *EEERover*. [Online]. Available: https://github.com/edstott/EEERover

[21] Atmel, "At14596: Tcp client and server operation using atwinc1500," Atmel, Tech. Rep., Aug. 2016. [Online]. Available: http://ww1.microchip.com/downloads/en/appnotes/atmel-42739-tcp-client-and-server-operation-using-atwinc1500_at14596_applicationnote.pdf

[22] *React Framework*, React. [Online]. Available: https://reactjs.org/

[23] *Electron*, Electron. [Online]. Available: https://www.electronjs.org/

[24] *Fetch API*, Mozilla. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API

[25] *Cross-Origin Resource Sharing*, Mozilla. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS

[26] "Tcp communication process," ATech Academy. [Online]. Available: https://i1.wp.com/aurumme.com/atech/wp-content/uploads/2017/04/TCP-3-way-handshake-process-ATech-Waqas-Karim-8.png

[27] ipv6.com, "User datagram protcol." [Online]. Available: https://www.ipv6.com/general/udp-user-datagram-protocol/

[28] K. Hoang, *WiFiWebServer Arduino library*. [Online]. Available: https://github.com/khoih-prog/WiFiWebServer

[29] M. Greshko. National geographic. [Online]. Available: https://mars.nasa.gov/mer/

[30] D. A. Gunn, "How long will the man-made objects on the moon last?" Apr 2020. [Online]. Available: https://www.sciencefocus.com/space/how-long-will-the-man-made-objects-on-the-moon-last/

[31] Rivellini. NASA. [Online]. Available: https://patentimages.storage.googleapis.com/bf/a4/dd/5d6267c2487d9f/USD673482.pdf

[32] Lindemann. NASA. [Online]. Available: https://patentimages.storage.googleapis.com/b4/41/8a/3615edd16bffec/USD487715.pdf