ELEC40006 - Electronics Design Project 1

Imperial College London

Department of Electrical and Electronic Engineering

Group Nimbion

EEE Rover

Authors:

Guanxi Lu (CID: 02233412)

Haocheng Fan (CID: 02262158)

Idrees Mahmood (CID: 02061101)

Noam Weitzman (CID: 02049854)

Adam Eljaafari (CID: 0 2204948)

Tofunmi Esuruoso (CID: 02318963)

Project Marker:

Dr Edward Stott

Date of Submission:

June 15th, 2023

Abstract

The EEE Rover Project aims to develop a remotely-controlled rover that can explore a unknown planet and survey the alien creatures that live there about three characteristics: Name, Age and Magnetism.

The project includes several stages, including group discussions, individual work on submodules, separate testing, compilation and assembly, as well as final testing and reviewing.

A robust rover system is developed which is capable of collecting and processing data, displaying information and interact with user through a webpage.

This report presents the design and implementation process of the project, which includes three main aspects: Sensor Units, Remote Control Units, and Mechanical Design.

The EEE Rover not only develops a robust, well-functioning, and user-friendly rover design, but also witness the collaborative work of all members of Group Nimbion.

Table of Contents

PART 1: Introduction

Section 1: Background and Aims

1. Aim of the Project

This project aims to design a remotely controlled rover to explore a remote planet and survey the alien creatures that live there. There are three characteristics to be figured out from the alien: Name, Age and Magnetism. For the three characteristics above, three different sensors are required for the survey: Radio wave, Infrared and Magnetism.

As stated in project brief [1], the prototype of the design is evaluated according to the criteria below:

(1) Ability to find the characteristics of the alien
(2) Cost and weight effectiveness
(3) The manoeuvrability of the rover to negotiate the environment
(4) Robustness and reliability of the construction
(5) Logicality and user-friendliness of the remote-control interface

2. Operating Environment

The design prototype is tested in an artificial environment in the lab.

2.1 Demo Arena

The demo arena is a flat, smooth area in the lab which consists of several aliens and obstacles. A view of the arena is shown in Figure 1.1.1 [2].

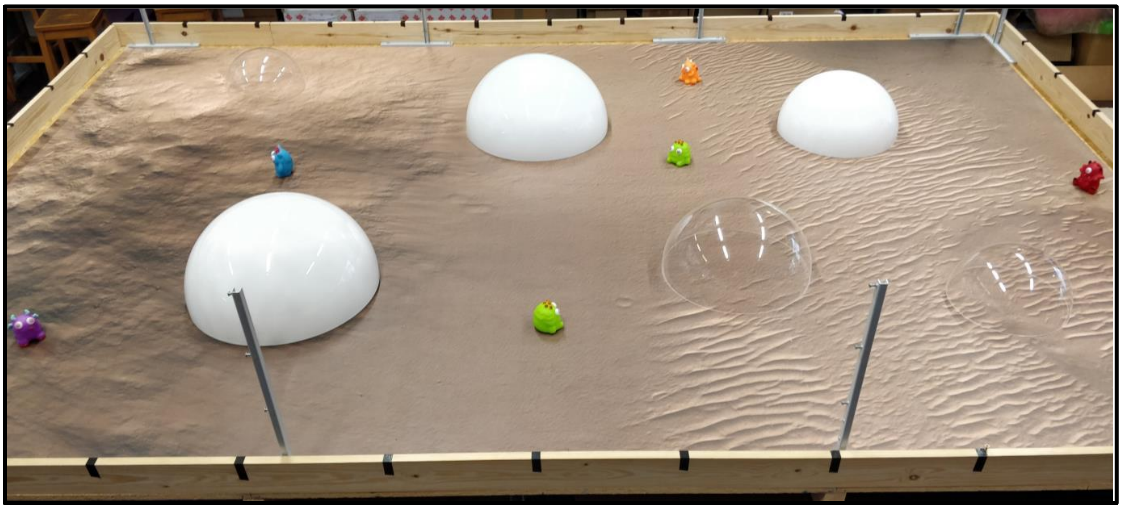


Figure 1.1.1: Demo Arena

2.2 Alien Simulator

As viewed in Figure 1.1.2 [2], an alien simulator comprises an electronic device and a rubber case. The electronic device, placed at the bottom of the simulator, replicates the characteristics of actual aliens.

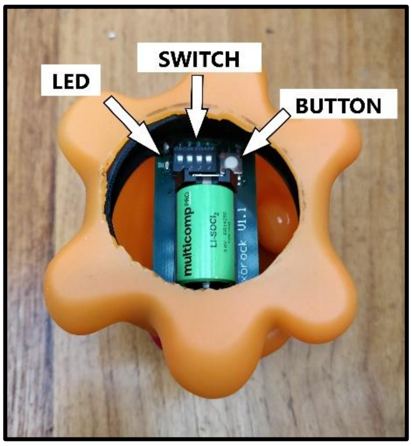


Figure 1.1.2: Alien Simulator

Figure 1.1.3: Electronic Device

As illustrated in Figure 1.1.3 [3], the electronic device consists of a LED, four switches and a button. Every time the button is pressed, LED indicates the battery level, then it transmits the signal determined by the four switches.

Four combinations of switches meaningfully represent four modes, while each mode represents a unique configuration of radio wave and infrared signals, as shown in Figure 1.1.4 [3].

In normal operations, the simulator is set to the first mode, where it simultaneously transmits its personalised name and pulse. At the same time, the other four modes are used for testing purposes.

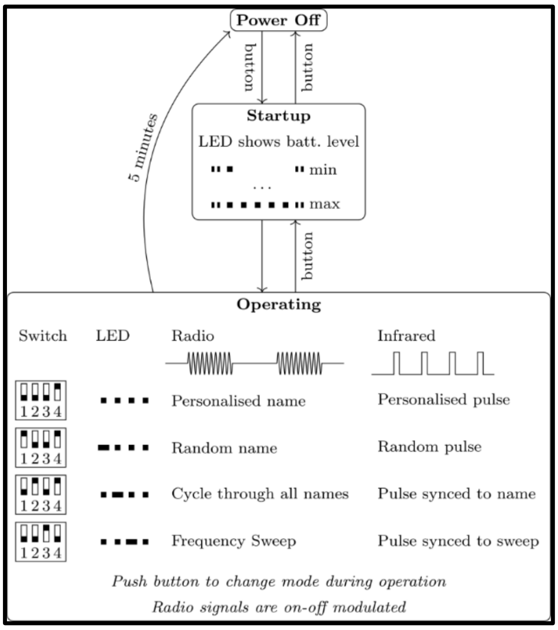


Figure 1.1.4: Operating modes of alien simulator

2.3 Magnetic

A small magnetic will be placed inside the alien approximately 10 mm below the top to simulate its magnetic property. Magnetic field lines go from top to bottom for S polarity, while for N polarity, magnetic field lines go from bottom to top. This is illustrated in Figure 1.1.5 [3].

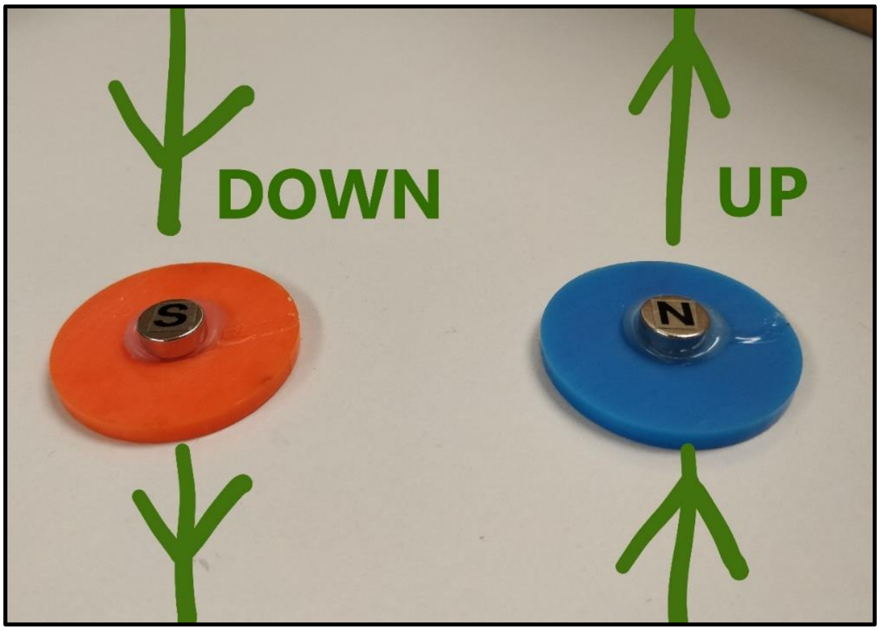


Figure 1.1.5: S and N polarities of magnetics

3. Design Criteria

A Product Design Specification (PDS) defines the product to be designed. Table 1.1.1 below highlights the X crucial elements that defines the EEE Rover.

Element	Criteria
Environment	1. The rover is expected to manoeuvrer in a defined demo area. 2. The rover should be able to avoid the obstacles. 3. The rover should work under normal temperature and pressure.
Materials	1. For sensor parts, electronic components and wires are needed, and all components are fixed on a stripboard. 2. Mechanical parts should mainly use plastic and acrylic, and the tyre is made of rubber.
Target Product Cost	1. Overall target should strictly below £60. 2. 30% of the budget should be reserved for emergency. 3. The project should aim at using as least budget as possible.
Processes	1. Sensing units should be designed and simulated on software before they are built on breadboard. 2. All sensing units in the final design should be soldered on stripboard. 3. Mechanical parts are manufactured with 3D printing or laser cut. 4. Programs are built on PlatformIO and uploaded to the microcontroller. 5. All units are assembled at the end of the design.
Performance	1. The Radio Wave, Infrared and Magnetism sensors should detect the signal correctly. 2. Sensors be able to detect the signal within 15 cm distance. 3. The rover should be controlled remotely with keyboard/mobile phone, and the delay should be within a desirable range. 4. The rover should be able to move forward, backward, left and right. 5. The webpage should display all the information gathered correctly.

Table 1.1.1: PDS Summary

A Product Design Specification document is attached in Appendix 3.

Section 2: Project Management

1. Responsibilities and Roles

The Nimbion team consists of six members. Each member is assigned different technical and non-technical roles according to their strengths and preferences. The aim is to ensure everyone receives an average workload and maximises their strengths in various parts of the project. Furthermore, flexibility is available in case some of the members are absent.

Table 1.2.1, attached below, describes each member's responsibility based on the actual situation.

Name	Technical responsibilities	Non-technical roles
Guanxi Lu	Radio wave sensor	Project manager, Report editor
Haocheng Fan	Radio wave sensor, Magnetism sensor	Report author and reviewer
Idrees Mahmood	Infrared sensor, Web server	Report author and reviewer, Treasurer
Noam Weitzman	Remote control, Chassis design	Report author and reviewer
Adam Eljaafari	Remote control	Report author and reviewer
Tofunmi Esuruoso		

Table 1.2.1: Responsibilities and roles

2. Meeting Routine

Flexibility is granted to team members, allowing them to work at their preferred pace and select their own working schedules. To ensure coordination and keep everyone informed about the project's progress, regular team meetings are essential and must be conducted.

Usually, the project manager organises one meeting each week, with the date and time determined two days ahead. In a regular meeting, each member presents their progress the following week and answers questions from other group members. In the meantime, the project manager evaluates whether the progress matches the objective. Afterwards, group members discuss the objective for the next week and, if necessary, the project manager would reallocate the resources available to ensure the completion of the objective. The project manager would state the nearest due or event at the end of the meeting.

3. Time Management and Planning

Each week in the meeting, group members discuss the objective for the coming week, and planning is scheduled with a Gannt chart. Also, the team would amend the plan with actual progress. With this routine, group members can be completely aware of the project, and the workload can be divided fairly throughout the period.

The Gannt chart can be found in Appendix 1.

4. Budget Management

Budget management is strictly monitored within the group, and group members have made every effort to maximise the efficiency of the spending and avoid waste. The following approaches are taken to achieve efficiency:

(1) A discord channel is created, and every order should be discussed within the channel before approval.
(2) A group member (Idrees Mahmood) serves as treasurer and reviews all the orders before approval.
(3) Before every order, group members must read the datasheet carefully to ensure that the component meets the requirement.
(4) For every order, group members must confirm that the component is in stock.
(5) For every component needed, group members should compare against all the suppliers to ensure that the choice is the lowest price.
(6) The group aims to leave at least 30% of the budget in case of emergencies.

The final expenditure is £38.71, which accounts for 64.52% of the total budget. A table of ordering history is shown in Appendix 2.

PART 2: Design Process and Details

Section 1: Name Characteristics - Radio Wave

1. Introduction and Objectives

The alien will reveal its Name by radio communication. The data rate of transmission for radio communication is 600 bits per second. The digital input signal is transmitted with a carrier frequency of 61k Hz, modulated with binary amplitude-shift keying (ASK, on-off modulation).

The alien's name is a four-character word, including an initial “#” symbol. The name is encoded with ASCII character codes in UART packets with one start and one stop bit. The lowest Significant Bit (LSB) is transmitted first.

The relevant sensor aims to receive the modulated signal, demodulate it to recover the modulating signal and convert it to the character represented by ASCII code.

2. Component List

Section	Components	Quantity
Receiver	Copper Coils	1
	10 nF Ceramic Capacitor	1
Rectification Circuit	1N914 Diode	1
	10 kΩ Resistor	1
	20 nF Ceramic Capacitor	1
Comparator	74HCT14 Schmitt Trigger Inverter	1
Amplifier	TL072 Dual Op-amp	1
	1 kΩ Resistor	2
	20 kΩ Resistor	1
	6.2 kΩ Resistor	1
Power Supply	ICL76605CPA +VE TO -VE Converter	1
	10 µF Electrolytic Capacitor	2
Others	Wires	Several
	Stripboard Medium 95mm × 127mm	2

Table 2.1.1: Component List

3. Design Details

3.1 Overview

The design of the radio wave sensor consists of six submodules, as illustrated in Figure 2.1.1.

(1) Receiver: Receive the signal at the carrier frequency with a resonance circuit.
(2) Rectification circuit: Rectify the high-frequency input waveform and produce a signal indicating the amplitude of the input signal.
(3) Comparator: Generate a square wave by comparing the demodulated signal with a threshold.
(4) Decoding program: Decode demodulated binary signal, represented by a square wave,

into bytes with the UART communication built-in of the microcontroller.
(5) Amplifier: Amplify the signal generated by the antenna.
(6) Power supply: Supply powers to components like Op-amps and Schmitt Trigger.

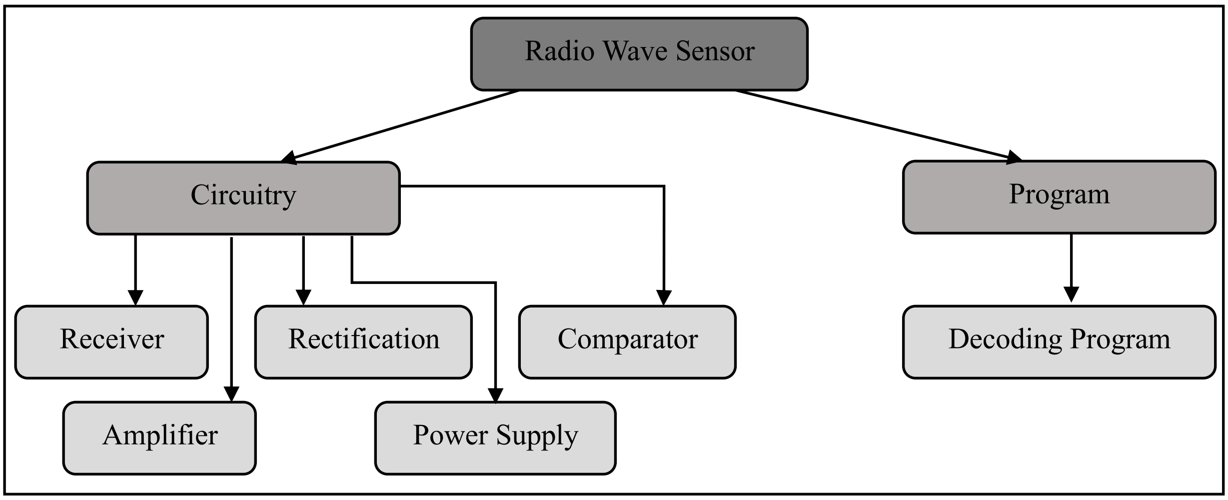


Figure 2.1.1: Submodules of radio wave sensor

The design process can be concluded into five stages:

(1) General discussion and divide the objective into submodules.
(2) Circuitry design and simulation on LTspice.
(3) Build up the circuit and primary test with the oscilloscope.
(4) Program design and secondary test with the program.
(5) Further tests and improvements.

The whole circuitry design includes the receiver, rectification circuit, comparator, and amplifier. A direct view of the circuitry design is below in Figure 2.1.2.

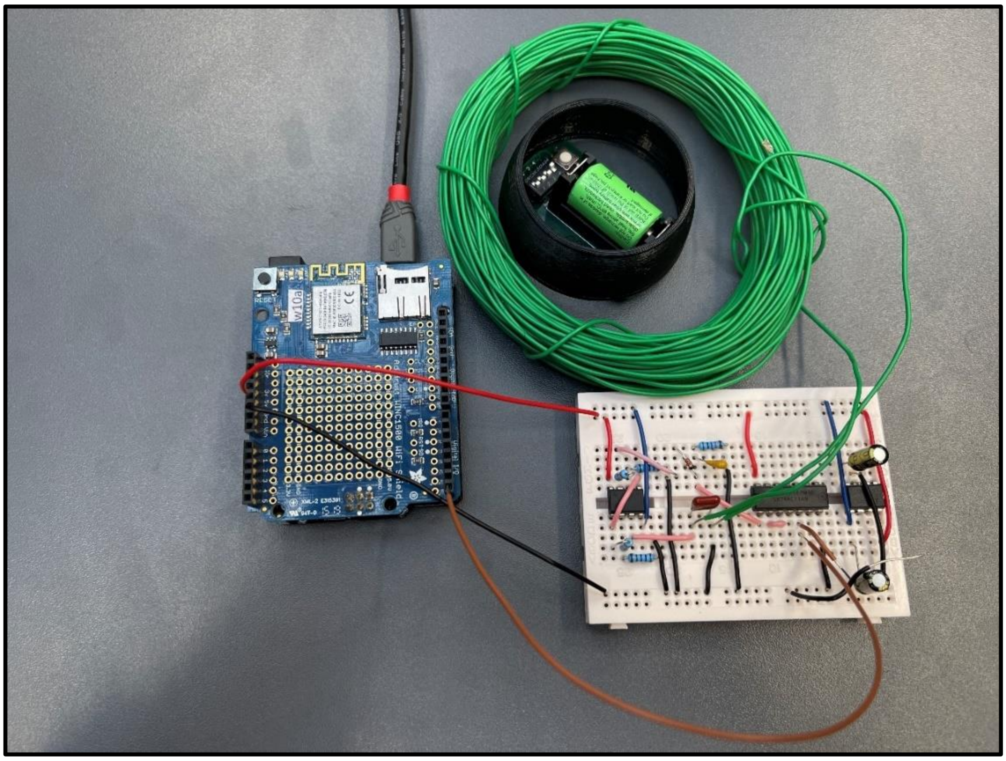


Figure 2.1.2: Circuitry design in breadboard

3.2 Receiver

3.2.1 Principles

3.2.1.1 Amplitude Shifting Keying (ASK)

The process of modulating a digital signal is called keying. Amplitude Shifting Keying (ASK) modulates the signal in magnitude. The simplest and most common ASK is on-off keying (OOK), which uses the presence of a carrier wave to indicate a binary one and its absence to indicate a binary zero [4]. This can be represented by the following equation: $s(t) = A(t)\cos(2\pi f_c t)$, where $A(t) \in \{0,\ A\}$.

3.2.1.2 Resonant frequency

Resonant frequency reflects a frequency at which the response amplitude is a relative maximum value of the system. For a RLC circuit, the resonant frequency is the frequency at which the impedance of the circuit is at a maximum [5], which is equivalent to the purely real impedance. Since the impedance of the resistor is R, the impedance of the capacitor is $1/j\omega C$, and the impedance of the inductor is $j\omega L$, we can derive that the resonant frequency is $\omega_0 = \frac{1}{\sqrt{LC}}$, so $f_0 = \frac{1}{2\pi\sqrt{LC}}$.

3.2.2 Configuration

The receiver is an LC resonance circuit. The resonance circuit is set to receive the signal at or near a particular corner frequency, in this case, 61k Hz, where the input signal has maximum magnitude. The capacitor and inductor are connected in parallel, as illustrated in Figure 2.1.3.

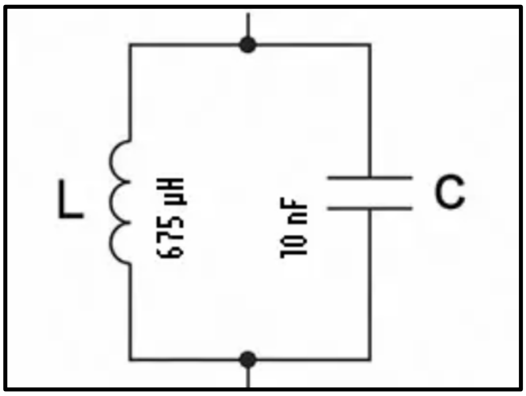


Figure 2.1.3 Parallel LC setup

Apart from approaching the resonant frequency, another critical factor is the circuit's total impedance. The impedance of the circuit should be minimised. It can be derived from the formula at high frequencies to receive the input signal with a more desirable quality; the capacitance should be large while the inductance should be small to achieve low impedance.

The inductor in the circuit is implemented with a tuned coil antenna, which aims to be constructed in a large diameter to improve signal quality. An LCR bridge can be used to measure the antenna's inductance.

In our configuration, the inductance of the antenna is 675 µH. In this case, a 10 nF capacitor enables us to approach resonant frequency at 61 kHz, and the total impedance is 30.69 kΩ.

3.2.3 Discussion

In preliminary experiments, the antenna was replaced with an ordinary inductor. The value of the inductor used in this version is 1 mH, and that of the capacitor is 6.8 nF. Accordingly, the resonant frequency is 61.03 kHz, and the total impedance was 353 kΩ. A photo of the configuration can be viewed in Figure 2.1.4.

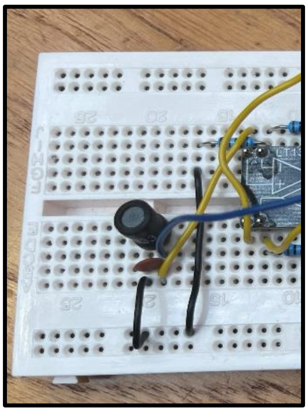


Figure 2.1.4: Antenna replaced with inductor

With coils functioning as an antenna, the receiving ability is improved, and signals can be received from a more significant distance.

3.2.4 Performance

The input wave can be viewed from the oscilloscope, as shown in Figure 2.1.5. The signal can be described as different packages of waves from a comparatively large time domain. The magnitude of the signal also has a strong negative correlation to the distance from the alien to the sensor. However, typical magnitude values vary from a few tens of millivolts to a few hundreds of millivolts.

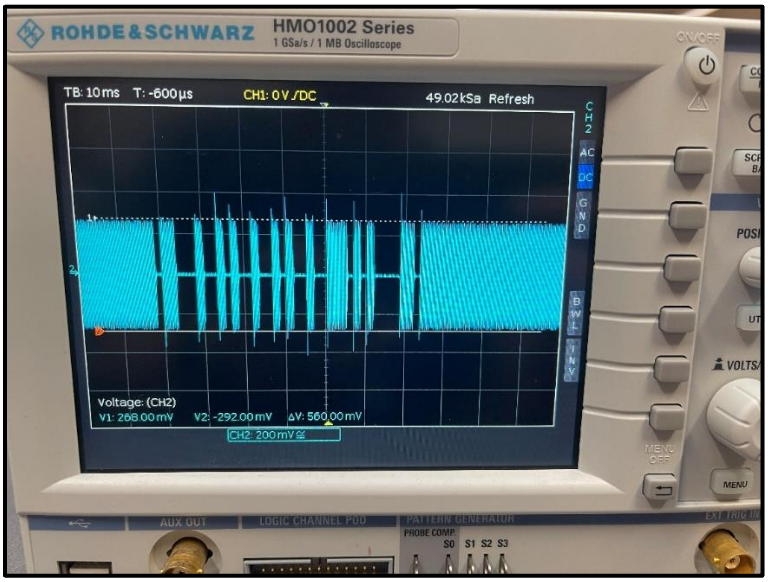


Figure 2.1.5: Input signal generated by resonance circuit

3.3 Rectification circuit

3.3.1 Principles

3.3.1.1 Envelope Detector

An envelope detector takes the high-frequency input signal and provides an output signal as the demodulated envelope of the original signal.

A simple example of the envelope detector is the diode detector, which consists of a diode and a parallel RC circuit. In this setting, the diode enhances one-half of the receiving signal over the other half. At the same time, the RC circuit removes high-frequency components of the signal and hence smooths the receiving signal, even though there are often some ripples remaining in the output circuit. A diagram of the setting above is shown in Figure 2.1.6.

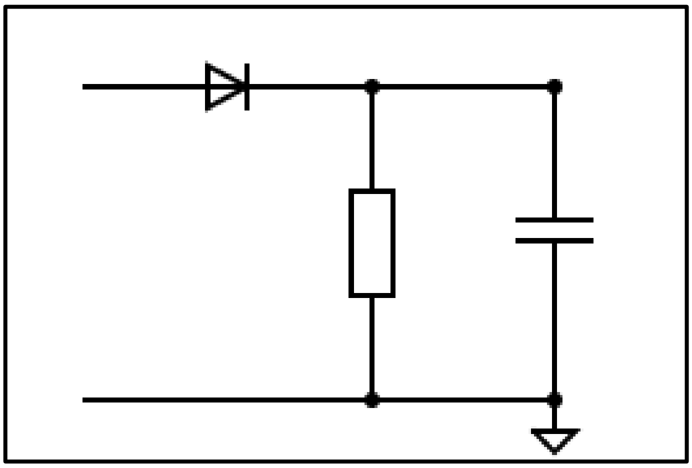


Figure 2.1.6 Diode detector

Constraints exist for the values of RC. If RC is too small, it could lead to significant ripples in the output waveform; if RC is too large, the voltage of the output signal will not follow the envelope. From this sense, it is required that $RC \gg 1/\omega_c$, and $RC < 1/2\pi B$.

3.3.2 Configuration

The rectification circuit applies a diode detector set-up. For R and C values, following the constraints mentioned in 3.3.1.1, $1.64 \times 10^{-6}\ \Omega \cdot \mathrm{F} < RC < 2.65 \times 10^{-4}\ \Omega \cdot \mathrm{F}$. A 10 kΩ resistor and a 20 nF capacitor were chosen, giving a RC value of 200 μΩ·F.

3.3.3 Performance

The input and output waveform can be viewed from the oscilloscope, as shown in Figures 2.1.7 (a) and 2.1.7 (b). The input waveform is the waveform generated by the resonant circuit after going through amplification, while the output waveform is the output of the diode detector.

In theory, the output waveform should provide a signal as the demodulated envelope of the original signal. However, in practice, the output signal depends on the magnitude of the input signal. For the case illustrated in Figure 2.1.7 (a), the input signal is small, and the output signal is the demodulated envelope of the input signal. When the input signal is significant, the magnitude of the output signal is restricted.

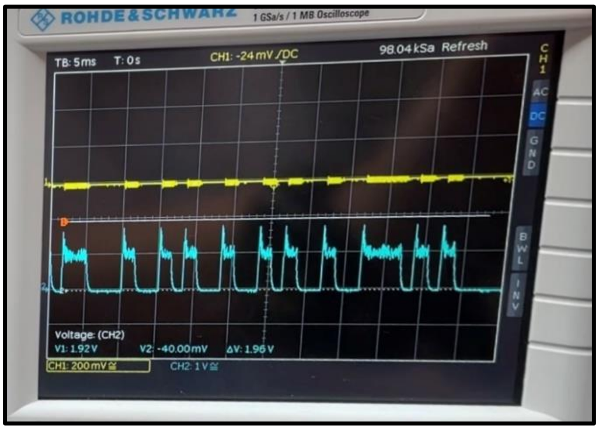


Figure 2.1.7 (a): Small input

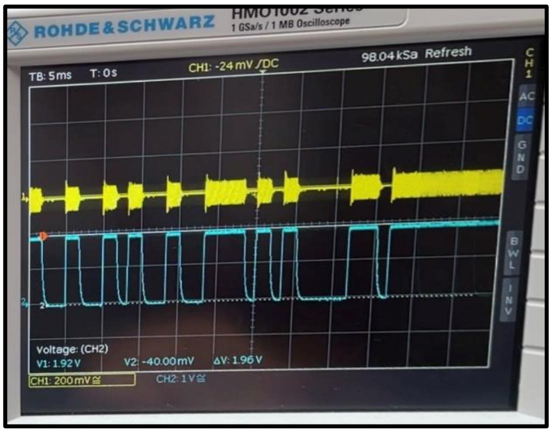


Figure 2.1.7 (b): Large input

3.4 Comparator

3.4.1 Principles

3.4.1.1 Schmitt Trigger

The schmitt trigger is a comparator circuit that converts analogue input signals into digital input signals with positive feedback.

For a comparator configuration illustrated in Figure 2.1.8, the positive feedback logic can be explained as follows: when the value of Y rises, the value will increase, leading to Y rising even more until it reaches the maximum value. Conversely, when the value of Y drops, it will keep dropping until reaching the minimum value. In this way, the Schmitt trigger can be viewed as a switch.

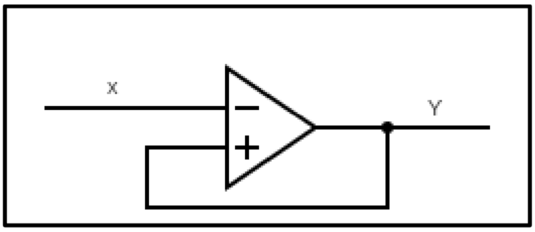


Figure 2.1.8: Comparator

For transistor-based Schmitt triggers, the high threshold and low threshold are determined by built-in resistors. In the case of a Schmitt Trigger implemented by two emitter-coupled BJTs, as illustrated in Figure 2.1.9 [6], the high threshold is determined by a voltage divider formed by R_{C2} and R_E, while the low threshold is determined by a voltage divider formed by R_{C1} and R_E. The formula for determining the value of high threshold and low threshold is shown as follows:

$$V_{HT} = \frac{R_E}{R_E + R_{C2}} V_+,\ V_{LT} = \frac{R_E}{R_E + R_{C1}} V_+ \quad .$$

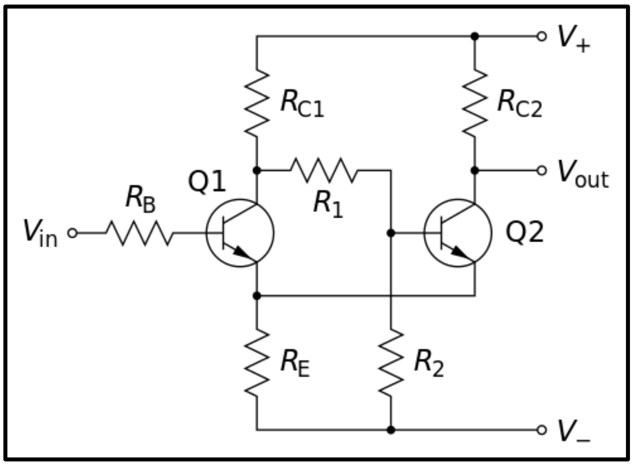


Figure 2.1.9: Schmitt trigger implemented by two emitter-coupled BJTs stages

3.4.2 Configuration

The comparator design applies a transistor-based Schmitt trigger instead of building up a positive feedback Op-amp circuit. 74HCT14 Schmitt Trigger Inverter was chosen, whose logic diagram is illustrated in Figure 2.1.10 [7].

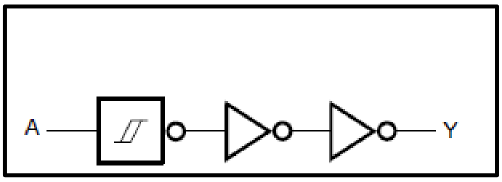


Figure 2.1.10: Logic diagram of 74HCT14 Schmitt Trigger Inverter

Since the Schmitt trigger inverter will invert the waveform after comparison, the design inverts twice, and in this case, the Schmitt trigger inverter functions as a Schmitt trigger buffer. A view of the breadboard is shown in Figure 2.1.11.

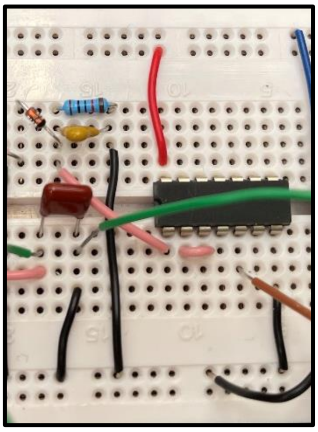


Figure 2.1.11: Breadboard view of Schmitt trigger buffer

3.4.3 Discussion

The preliminary design was implemented with a positive feedback circuit. However, there was a problem with finding the threshold voltage, which led to a more complex circuit, higher weight, and a lack of robustness. With transistor-based Schmitt trigger being introduced, input waveform can be automatically compared with the threshold voltage, improving the precision, and reducing complexity.

3.4.4 Performance

The input and output waveform can be viewed from the oscilloscope, as seen in Figure 2.1.12. The rectification circuit generates the input waveform, while the output waveform comes through the comparator. It is demonstrated that the analogue input waveform is converted into the digital waveform, which can be represented by a square wave, where low voltage represents binary 0 and high voltage represents binary 1.

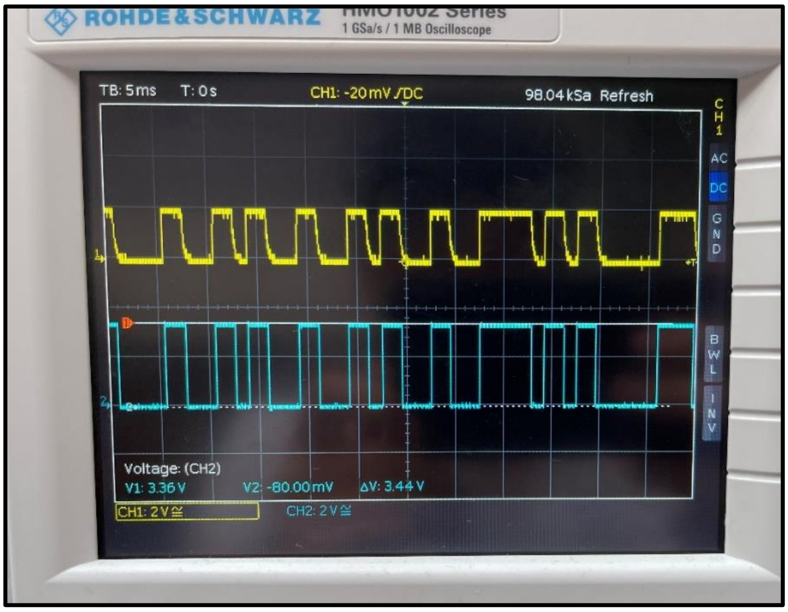


Figure 2.1.12: Input and output signal of the comparator shown in oscilloscope

3.5 Decoding program

3.5.1 Program Overview

The decoding program converts demodulated digital signals to characters. ASCII characters can be automatically converted when the demodulated signal is connected to the 0 digital pins with the usage of the internal library of the microcontroller. The information signal is formed with a loop of 4 characters, with an initial "#" symbol. The decoding program mainly consists of two parts: the anti-noise function and the loop function.

3.5.2 Anti-noise Function

The Anti-noise, as shown in Program 2.1.1, use function `Find_MostFre`. In this function, with variables `maxCount`, `countMap` and `mostFrequent`, the string with highest frequency in the ten-string sampling is selected.

```
void Find_MostFre(std::vector<String> samples, String& mostFrequent)
{
  std::map<String, int> countMap; // Map to count occurrences of
values
  int maxCount = 0;  // Maximum count of occurrences

  for (const auto& sample : samples) {
    countMap[sample]++;  // Increment the count for the current
sample
    if (countMap[sample] > maxCount) {
      maxCount = countMap[sample];
      mostFrequent = sample;
    }
  }
}
```

Program 2.1.1: Anti-noise Function

3.5.3 Loop function

The loop function, as shown in Program 2.1.2, is used to create strings that represent the name of an alien and print the name when a specific condition is satisfied. The characters converted from digital signals are pushed into a string in the loop part. When "#" is detected, the string will be pushed to a vector and cleared immediately. When "#" is not detected and the name is incomplete, the function will continue pushing the current character to the string. When the size of the vector is ten, which means ten samples are collected, the anti-noise function is called, and the result for the alien's name is printed.

```
void loop() {

  if (Serial1.available() > 0) {

    char asciiCharacter = Serial1.read();
    if (asciiCharacter == '#') {
      samples.push_back(outputString);
      outputString = "";  // Clear the output string

    } else {
      outputString += asciiCharacter;  // Append the character to the
output string
    }
    delay(10);
  }

  if (samples.size() == 10) {
    Find_MostFre(samples, MostFre);  // Call the function to find the
most frequent value
    Serial.println(MostFre);
    samples.clear();
    delay(100);
  }

}
```

Program 2.2.2: Loop Function

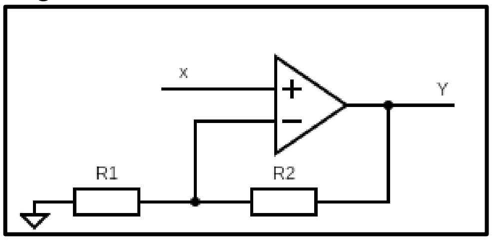
3.5.4 Discussion

Anti-noise function `Find_MostFre` improves tolerance against noise during the detection process. In the initial program, noise or other environmental factors may result in wrong characters, influencing the sensor's performance. With the addition of the anti-noise function, the result displayed on the website will not be influenced even if wrong characters occasionally appear.

3.6 Amplifier

3.6.1 Configuration

The amplifier design adopts the non-inverting amplifier model, as illustrated below in Figure 2.1.13. For the case below, a relationship can be derived as $Y=\frac{R2+R1}{R1}X$, so the input signal X is amplified by $\frac{R2+R1}{R1}$ times. In the configuration, two Op-amps are used to increase the gain by $\frac{20k+1k}{1k}\times\frac{6.2k+1k}{1k}=151.2$ times, as viewed from Figure 2.1.14.



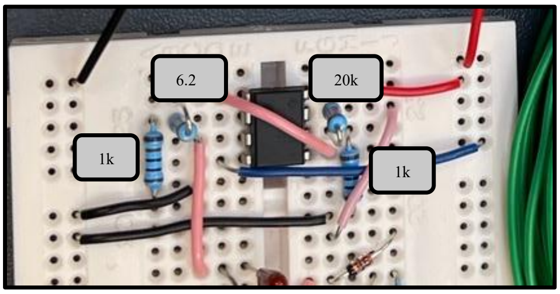


Figure 2.1.13: Non-inverting Amplifier

Figure 2.1.14: Amplifier Configuration

3.6.2 Discussion

The amplification circuit replaces the original LT1366 Op-amp with TL072 Op-amp, whose pin figuration is illustrated in Figure 2.1.15 [8].

TL072 is superior to LT 1366 for two reasons. Firstly, the LT1366 Op-amp has low gain-bandwidth product of 0.4 MHz [9], which can only amplify by 6.56 times at 61 kHz. For TL072, the gain-bandwidth product is 5.25 MHz [8], around ten times that of LT1366, enabling a larger amplification scale.

Secondly, the LT1366 Op-amp has a low slew rate. The slew rate is defined as the maximum rate of change of an Op-amp's output voltage and is given in units of volts per microsecond. A low slew rate can bring about a negative influence on comparator and rectification circuits. The slew rate of LT1366 is 0.13 V/µs, much lower than that of TL072 (20 V/µs).

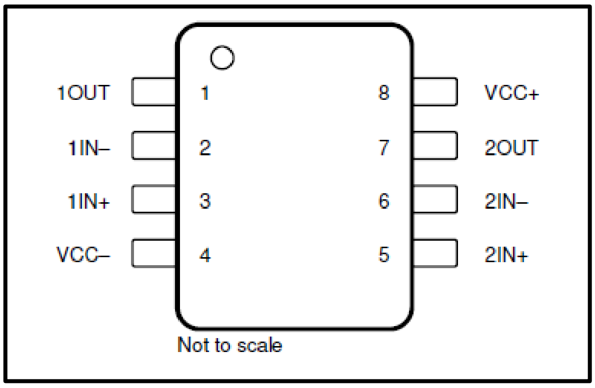


Figure 2.1.15: TL072 Pin Figuration

3.6.3 Performance

The amplification process is crucial since the input waveform received from the alien is usually small. The input and output waveform can be viewed from the oscilloscope, as seen in Figure 2.1.16. The input waveform is amplified by $\frac{3.12V}{28mV} = 111.4$ times, which is less than the theoretical value, but still able to pass through the diode in rectification circuit.

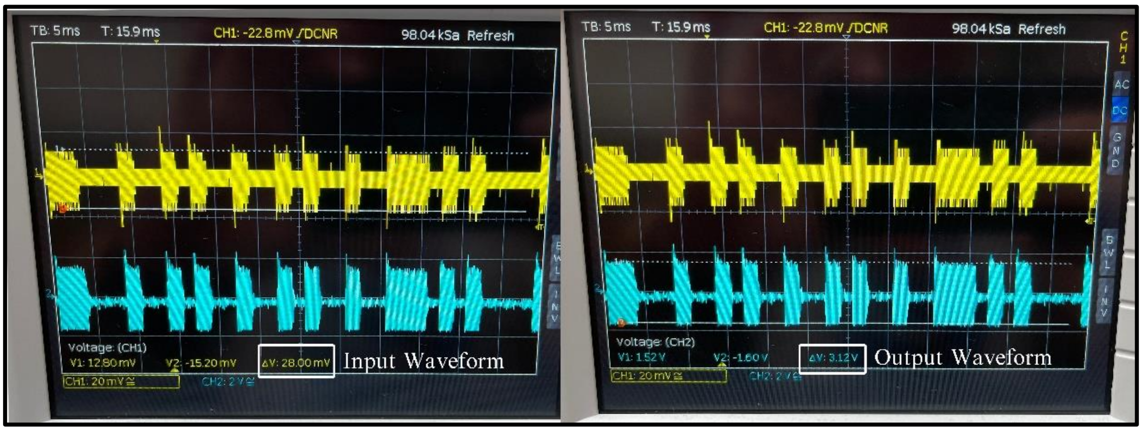


Figure 2.1.16: Input and Output waveform shown in oscilloscope

3.7 Power supply

3.7.1 Configuration

The logic level of the microcontroller is 3.3V. However, the negative voltage cannot be supplied directly, so an inverter is required. In this case, ICL76605CPA +VE TO -VE Converter is adopted.

With reference to datasheet [10], approaches can be taken for improving power efficiency. By adding two large capacitors, as illustrated in Figure 2.1.17, three conditions can be achieved:

(1) The driver circuitry consumes minimal power.
(2) The output switches have extremely low ON resistance and virtually no offset.

(3) The impedances of the pump and reservoir capacitors are negligible at the pump frequency. Hence theoretical power efficiency can be achieved.

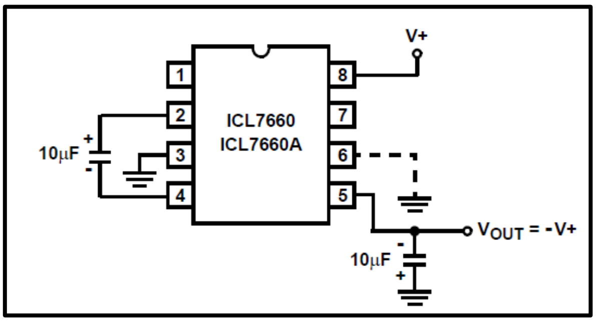


Figure 2.1.17: Power supply configuration

4. Outcomes and Performance

4.1 Simulation

The circuitry design was first developed in LTspice and simulation. Figure 2.1.18 and Figure 2.1.19 demonstrate the circuitry design in the simulation and the results of the simulation. In Figure 2.1.19, the tiny green waveform represents the signal received by the resonance circuit, the grey waveform represents the signal after amplification, the blue waveform represents the signal going through rectification, and the red waveform represents the signal passes through the comparator, which is a digital signal that the decoding program will process.

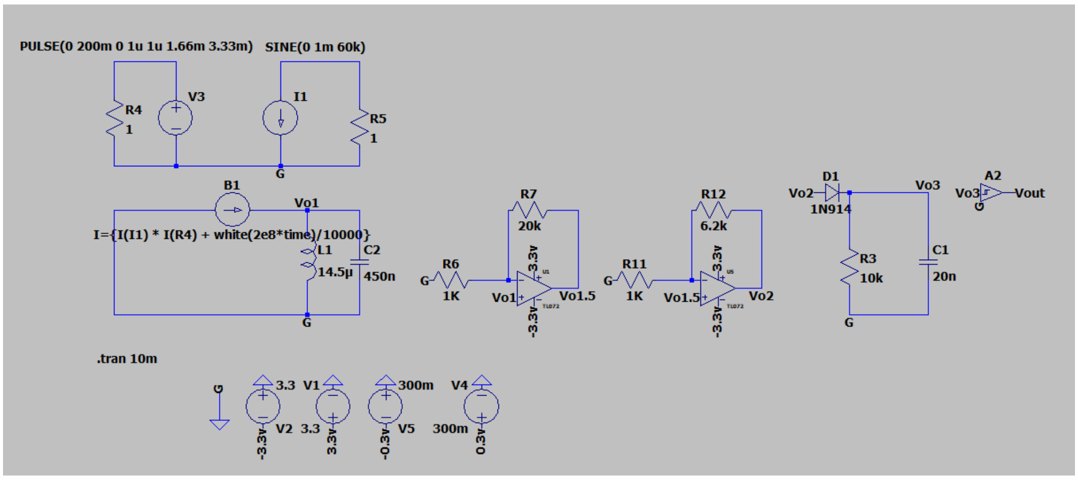


Figure 2.1.18: Circuitry design in LTspice

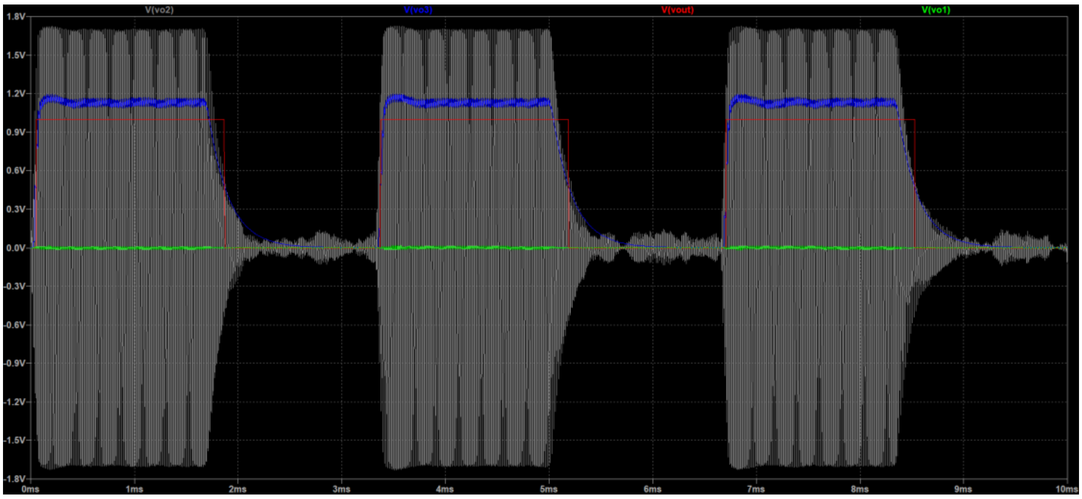


Figure 2.1.19: LTspice Simulation

It needs to be emphasised that design in simulation does not precisely reflect reality, and further improvements are made subject to actual situations. The theoretical design simulated the input signal as a square wave modulated by a sinusoidal wave, and the noise was added. The input signal was received by an LC resonance circuit with L = 14.5 µH and C = 450 nF. However, in practice, we used a different setup due to the restriction of materials, which generated a similar waveform.

4.2 Oscilloscope Test

After the simulation, the circuitry was constructed, as shown in Figure 2.1.20. The circuit was then reviewed, and connections were optimised so each submodule could be viewed clearly.

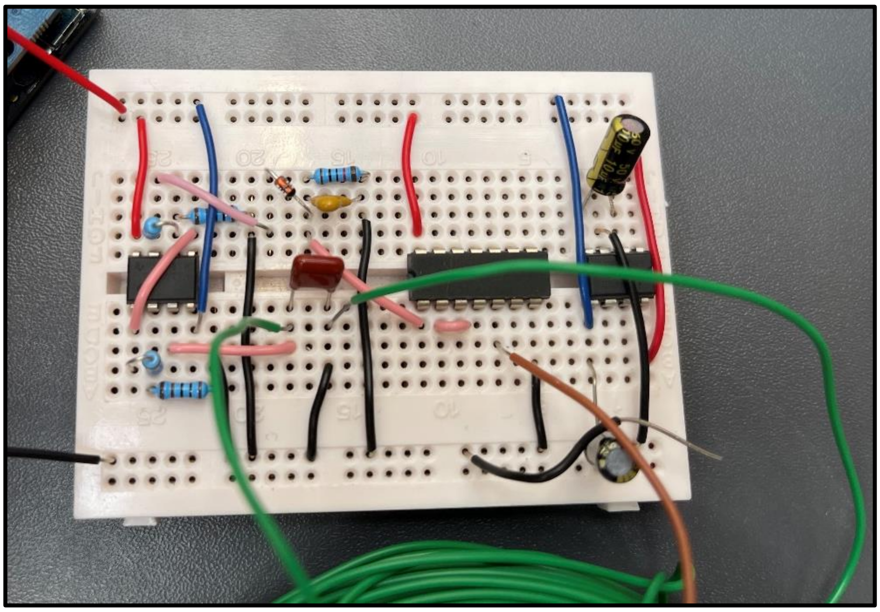


Figure 2.1.20: Complete circuitry

The oscilloscope tests the circuit to see if the input and output waveform matches the objective and simulation results. As seen in Figure 2.1.21, CH1 (yellow) displays the input signal, while CH2 (blue) displays the output signal. The input and output signal matches the simulation results in Figure 2.1.19.

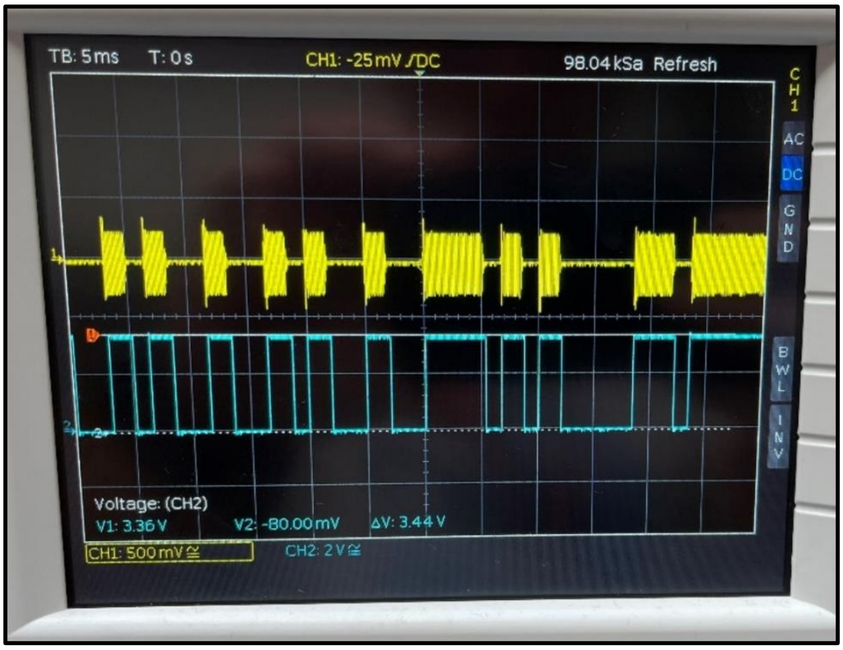


Figure 2.1.21: Input and output signal of the radio wave sensor displayed in oscilloscope

4.3 Final Test

The final test is carried out on the computer, whose aim is to prove the correctness of the program. Two tests are conducted, with different mode of the alien simulator.

4.3.1 Personalised Name

In this mode (switch 0001), the alien simulator transmits a signal representing its personalised name. The result will constantly be a three-character (exclude "#") name. The result of the program, as shown in Figure 2.1.22, displays a three-character name of "Ira", which matches the expected outcome.

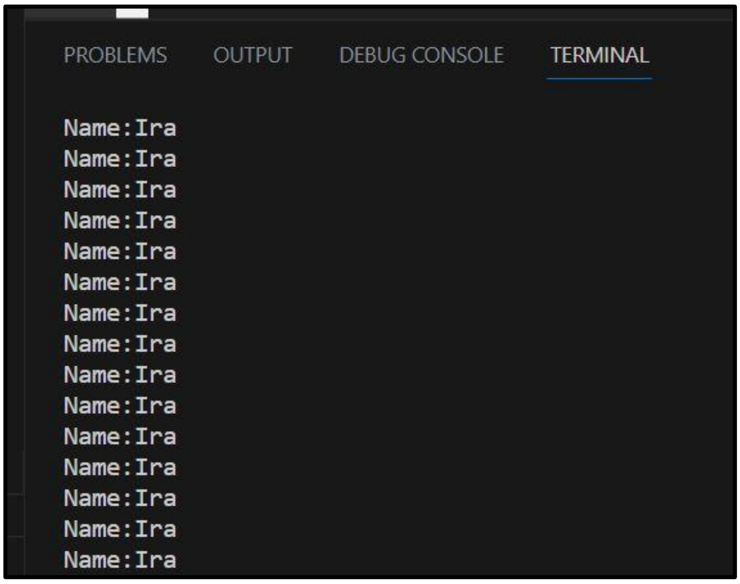


Figure 2.1.22: Result demonstrates a three-character name "Ira"

4.3.2 Cycle Through All Names

In this mode (switch 0101), the alien simulator transmits signals representing all possible names. The result will be cycling through all possible names from A to Z. The result of the program, as shown in Figure 2.1.22, displays various three-character names which match the expected outcome.

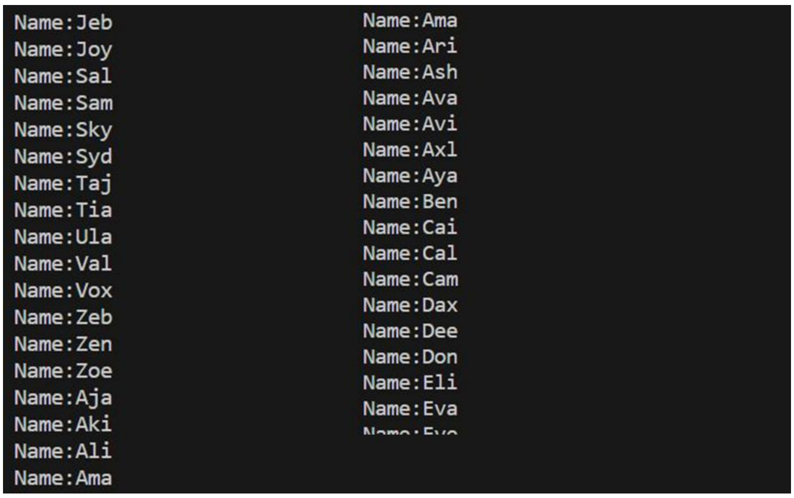


Figure 2.1.23: Result cycles through all possible names

Section 2: Age Characteristics – Infrared

1. Introduction and Objectives

The age of the alien is indicated by a pulsing infrared signal: the period of the pulse increases at exactly 1ms per century of lifetime. The frequency of the pulse ranges from 135 to 1000 Hz, while the pulse width of the infrared signal is 50 µs.

The relevant sensor aims to obtain the period of the infrared pulse, then convert it into the real age. The environment is expected to be visually noisy, from which sense a filter which is only sensitive to the wavelength of the alien emission (950 nm).

2. Component List

Section	Components	Quantity
Infrared Sensor	Wires	Several
	MCP6002 Operational Amplifier	1
	SFH 309 FA-4/5 ams OSRAM, 24 ° IR Phototransistor	1
	1 kΩ Resistor	4
	10 kΩ Resistor	2
	10 µF Electrolytic Capacitor	1
	BC547 NPN Bipolar Transistor	1

Table 2.2.1: Component List

3. Principles

3.1 Phototransistor

Phototransistors, when biased correctly, produce a photocurrent when light is incident on the exposed collector-base junction. It creates a photocurrent, which is amplified by the DC current gain of the phototransistor (hFE) and can be interpreted by extra circuitry which is designed as follows. [11].

4. Design Details

4.1 Overview

The design of the infrared sensor consists of three submodules, as illustrated in Figure 2.2.1.

(1) Infrared Sensing Unit: Receive the analogue infrared wave from alien simulator.
(2) Signal Amplification and Processing: Amplify the signal received, and process through a high pass filter.
(3) Decoding Program: Calculate the age with the processed signal.

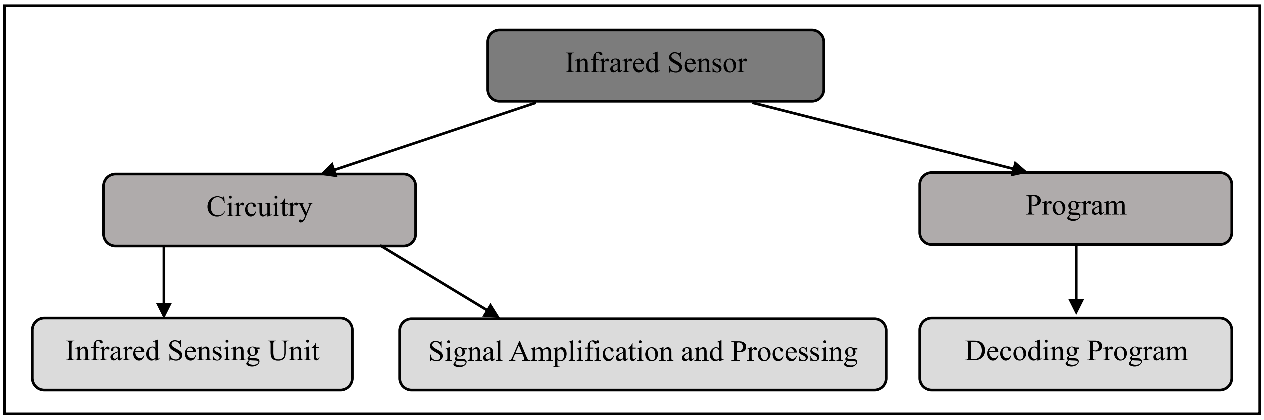


Figure 2.2.1: Submodules of radio wave sensor

The design process can be concluded into five stages:

(1) General discussion and divide the objective into submodules.
(2) Circuitry design and review.
(3) Build up the circuit and primary test with the oscilloscope.
(4) Program design and secondary test with the program.
(5) Further tests and improvements.

The whole circuitry design includes the infrared sensing unit, amplification circuit and a high pass filter. A direct view of the circuitry design is below in Figure 2.2.2.

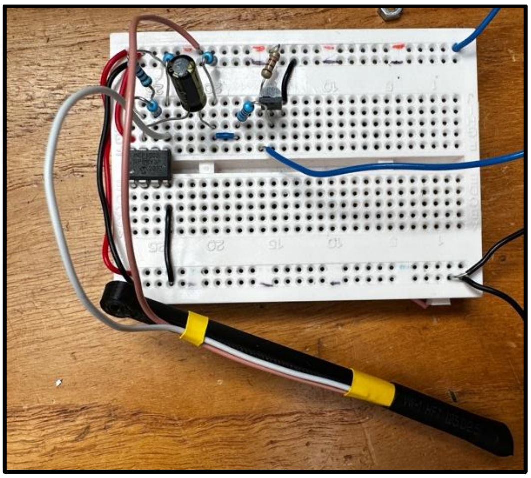


Figure 2.2.2: Circuitry design in breadboard

4.2 Infrared Sensing Unit

4.2.1 Selection of Infrared Sensor

Two types of sensors are commonly used to detect changes in light intensity, which are photodiodes and phototransistors. Despite being similar in semiconductor design, they are applied in distinct applications and circuit configurations.

A comparation between phototransistor and photodiode is stated as in Table 2.2.2.

Perspective	Phototransistor	Photodiode
Circuitry	Has a built-in Amplifier and requires less extra circuits.	Requires extra amplification
Sensitivity	Highly Sensitive.	Low sensitivity.
Response Time	Longer response time.	Instant response.
Price	Comparative low.	Comparative low.

Table 2.2.2: Comparation between phototransistor and photodiode

Phototransistor is more desirable since it removes the requirement for extensive amplification and the dependency on light intensity. SFH-300-3/4 phototransistor was adopted due to its relatively high sensitivity, low cost and large package size.

4.2.2 Circuit Test

The phototransistor is separately tested with alien simulator, where circuit configuration is demonstrated in Figure 2.2.3. The response of the phototransistor is demonstrated in Figure 2.2.4. It is found that the phototransistor responds quickly on the rising age but does not hold for 50 µs of HIGH period. Further, the falling edge is also slow, which suggests a delay in phototransistor response of at least 20 µs.

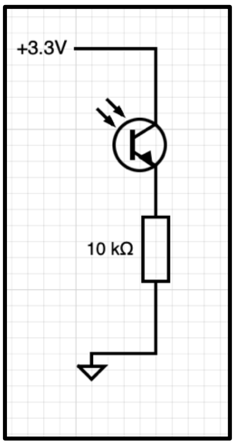


Figure 2.2.3: Circuit configuration for phototransistor testing

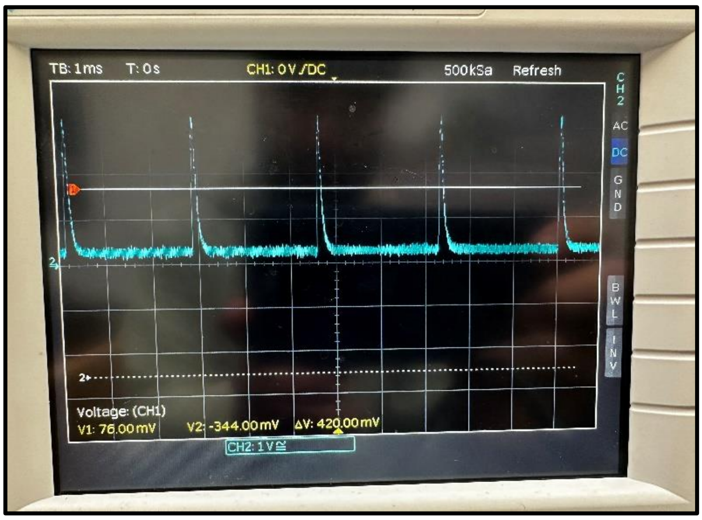


Figure 2.2.4: Phototransistor response

4.3 Signal Amplification and Processing

4.3.1 Signal Processing Circuitry

The distance between the alien and phototransistor must be small for it to work, so amplification is added to ensure stable detection in a larger distance. MCP6002 Op-amp configured as an inverting amplifier is adopted to achieve amplification, which highlights high gain-bandwidth product.

The addition leads to a DC offset caused by visual noise, which needs to be filtered out. A passive RC high pass filter is placed after the amplifier, whose values can be calculated from $f_c = \frac{1}{2\pi RC}$, where the corner frequency is around 15 Hz.

However, the signal still needs to be further amplificated, which can be achieved by a BC547 NPN transistor configured as a CE amplifier, which provides high gain with very simple set-up. The CE amplifier configuration may lead to some downsides such as phase delay and saturation, which does not hinder the design.

In the end, a heat shrink is used on the phototransistor to block off surrounding light. The resulting circuitry is demonstrated in Figure 2.2.5.

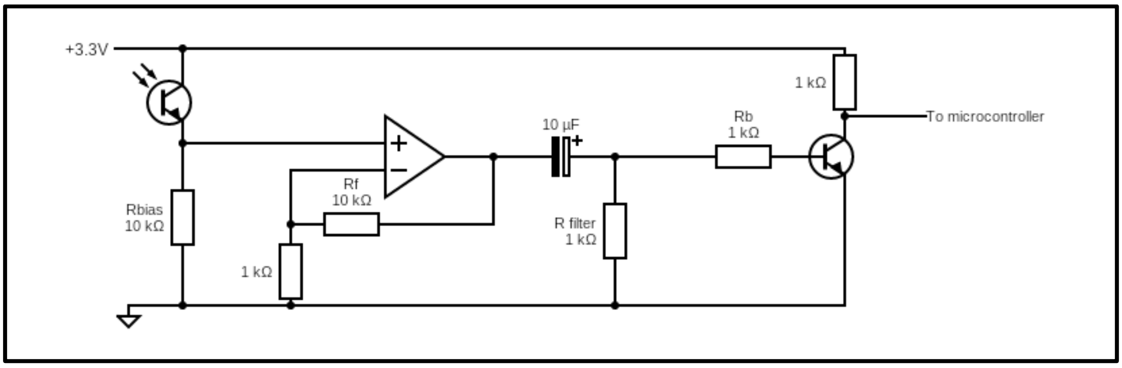


Figure 2.2.5 Resulting circuitry

4.4 Decoding Program

4.4.1 Calculation of age

From the circuitry, a defined HIGH period can be measured, which is fed into microcontroller. A formula can be derived as $Age = \frac{T(\mu s) - 50}{10}$.

Program 2.2.1 demonstrates the calculation of age, and at the same time filters out a range of unlikely values.

```
long getAge()
{
    // returns a long with the obtained age
    unsigned long age = 0;
    age = pulseIn(6, HIGH);
    std::vector<long> samples;
    if (age)
    {
        age = (age + 50) / 10;
        if (age > 740 || age < 100)
        {
            Serial.println("age invalid");
            return 0;
        }
        Serial.println("age found");
        samples.push_back(age);
    }
    else
    {
        Serial.println("no age found");
        return 0;
    }

    long mostFrequent = Find_MostFrel(samples);
    return mostFrequent;
}
```

Program 2.2.1: Decode program to calculate age

5. Outcomes and Performance

5.1 Oscilloscope Test

After design, the circuitry was constructed, and an oscilloscope test was conducted to see if the waveform matches the objective.

The waveform is shown in Figure 2.2.6, which is a steady and clear waveform for program to analyse the HIGH period and obtain the age.

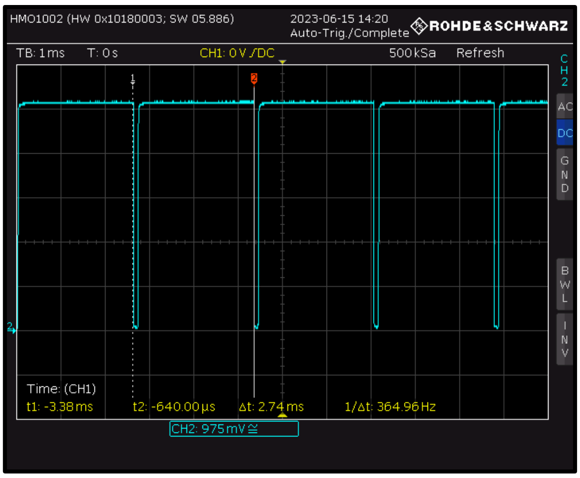


Figure 2.2.6: Waveform displayed in oscilloscope

5.2 Final Test

The final test is carried out in the computer, whose aim is to prove the correctness of the program. Figure 2.2.7 displays the result of the test. It is found that the age of the alien is 270 years, and the result is clear and consistent.

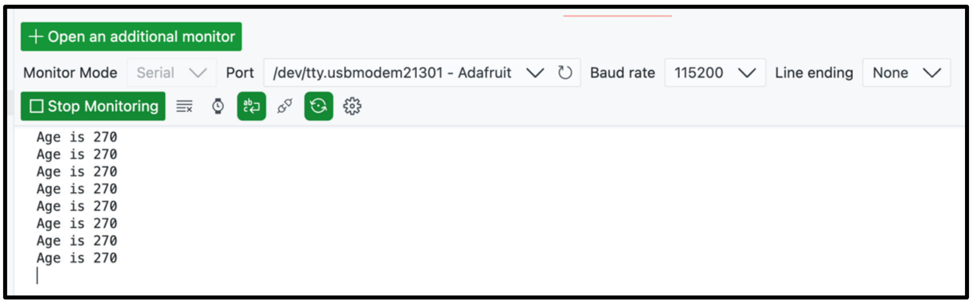


Figure 2.2.7: Result of final test suggests that the age is 270 years

Section 3: Magnetism Characteristics

1. Introduction and Objectives

A magnet is fixed to the body of the alien approximately 10 mm below the top to simulate its magnetic property. There are three types of static magnetic fields:

(1) Pointing away from top of their heads
(2) Pointing towards their heads
(3) Magnetic field is zero

The magnetism sensor aims to detect the direction of the magnetic field. A Hall effect sensor is implemented to carry this out.

2. Component List

Section	Components	Quantity
Magnetism	SS49E Linear Hall Effect Sensor IC	1
	Wires	Several
	1 kΩ Resistor	2
	10 kΩ Resistor	2
	50 kΩ Resistor	2
	TL072 Dual Op-amp	1

Table 2.3.1: Component List

3. Principles

3.1 Hall Effect and Hall Effect Sensor

The Hall effect generates a voltage when a magnetic field is applied perpendicular to the current flow in a conductor or semiconductor. When a magnetic field is present, these charges experience a force, called the Lorentz force [12].

The hall effect sensor is a type of sensor that detects the presence and strength of a magnetic field using the Hall effect. Figure 2.3.1[13] illustrates the working principles of a Hall effect sensor.

When a magnetic field with a perpendicular component is applied, the paths between collisions are curved. Thus, moving charges accumulate on one face of the material. The movement of charges leaves equal and opposite charges exposed on the other face. The result is an asymmetric charge density distribution on the Hall sensor's thin strip, arising from a force perpendicular to both the straight path and the applied magnetic field.

The separation of charge establishes an electric field that opposes the migration of further charge, so a steady electric potential is established for as long as the charge flows. This voltage difference is proportional to the strength of the magnetic field.

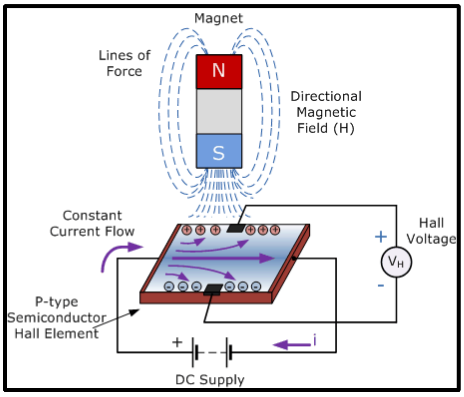


Figure 2.3.1: Principle of Hall Effect Sensor

4. Design Details

4.1 Overview

The design of magnetism sensor consists of two submodules, as illustrated in Figure 2.3.2.

(1) Circuitry: Obtain an analogue output value from Hall effect sensor.
(2) Program: Process the analogue output and identify the polarity.

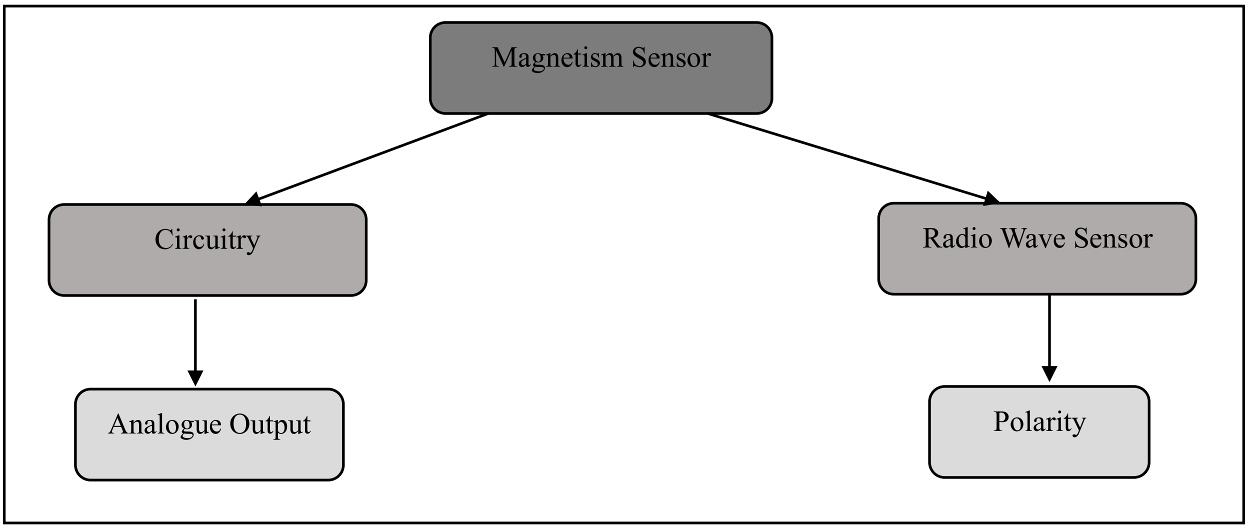


Figure 2.3.2: Submodules of magnetism sensor

The design process can be concluded into five stages:

(1) General discussion and divide the objective into submodules.
(2) Decide the type of sensor.
(3) Build up the circuit, primary test with oscilloscope.
(4) Program design, and secondary test with program.
(5) Further tests and improvements.

A direct view of the circuitry design can be viewed below in Figure 2.3.3.

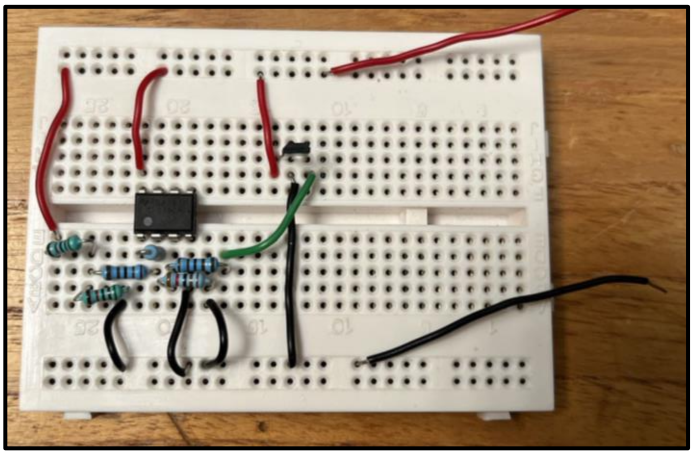


Figure 2.3.3: Circuitry design in breadboard

4.2 Circuitry

4.2.1 Hall Effect Sensor Selection

The type of the Hall effect sensor needs to be determined depending on the application's requirement. There are two types of Hall sensors: digital output and analogue output types.

For the digital type of Hall effect sensor, a specific threshold voltage needs to be defined to qualify the direction of the magnetic field. The sensor may not detect the direction of the magnetic field when the field is weak.

In this case, an analogue output type is desired, whose analogue output is presented as different voltages. The design adopts SS49E Linear Hall Effect Sensor IC, whose structure is shown in Figure 2.3.3 [14].

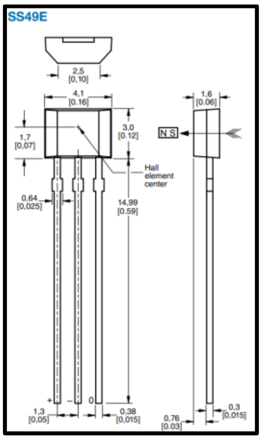


Figure 2.3.3: Structure of Hall Effect Sensor

4.2.2 Configuration

The circuit design, as shown in Figure 2.3.4, includes a Hall effect sensor, a potential divider and an Op-amp. When there is no magnetic field, the output voltage is about 1.67 V, the median value of the Op-amp supply voltage. When the direction of the magnetic field passes through the metal strip from top to bottom, the potential difference increases, and the direction of the field is North. Conversely, when the direction is from bottom to top, the direction of the field is South.

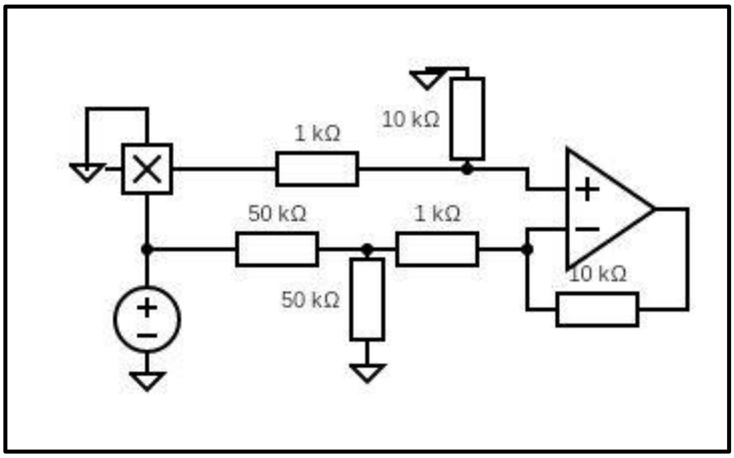


Figure 2.3.4: Circuit design

4.3 Program

4.3.1 Program Overview

The program identifies the direction of the magnetic field from the value of the analogue output. The program consists of three parts: an anti-noise function, a direction identification function, and a loop function.

4.3.2 Anti-noise Function

Anti-noise function selects and prints out the polarity which appears the most frequently in 10 samples using `Find_MostFreS`. The program of anti-noise function is demonstrated in Program 2.3.1.

```
void Find_MostFreS(std::vector<String> samples2) {
  std::map<String, int> countMap; // Map to count occurrences of
values
  int maxCount = 0;  // Maximum count of occurrences

  for (const auto& sample : samples2) {
    countMap[sample]++;  // Increment the count for the current
sample
    if (countMap[sample] > maxCount) {
      maxCount = countMap[sample];
      mostFreS = sample;
    }
  }
}
```

Program 2.3.1: Anti-noise Function

4.3.3 Direction Identification

The direction identification program is demonstrated in Program 2.3.2. When the magnetic field is zero, analogue output values are within a range of 1750 to 1730. The range is taken as a reference for direction identification. When the output from the sensor is higher than the reference level, the Polarity function gives the result as “N”. When the output from the sensor is lower than the reference level, the Polarity function gives the result as “S”.

```
String polarity="Neutral";

String Polarity(){

  double sensorValue= analogRead(A0);

  if (sensorValue>1750){
    polarity="N";

  }
  else if(sensorValue<1730) {
    polarity="S";
  }
  //Serial.println(sensorValue);
  return polarity;
}
```

Program 2.3.2: Direction Identification

4.3.4 Loop function

In the loop function, a vector is used to save analogue output values from the Hall effect sensor. When the size of the vector is 10, the anti-noise function is called, and the result is displayed on the interface. The program of the loop function is demonstrated in Program 2.3.3.

```
void loop(){

  samples.push_back(Polarity());
  delay(10);

  if(samples.size()==10){
    Find_MostFreS(samples);
    Serial.println(mostFreS);
    samples.clear();
    delay(100);
  }

}
```

Program 2.3.3: Loop Function

4.4 Discussion

In the preliminary version, as shown in Figure 2.3.5, the analogue output of the Hall effect sensor is directly connected to the microcontroller. In this case, when sensing from a greater distance, the variance of output values is not apparent, which may lead to detection errors.

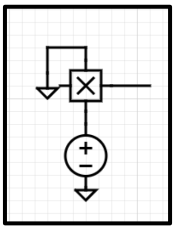


Figure 2.3.5: Preliminary version of the design

A differential amplifier is introduced to solve this problem. It can increase the range of the sensor. The Op-amp amplifies the difference between the threshold voltage, 1.67 V, and the actual voltage level. The threshold voltage is passed to the Op-amp through a potential divider and then amplified ten times.

Besides, an anti-noise function like the one in the radio wave sensor is introduced to improve noise tolerance. With the anti-noise function, only the polarity which appears the most frequently is printed out, leaving room for fluctuations.

5. Outcomes and performance

5.1 Oscilloscope Test

The analogue output is measured with an oscilloscope to test the circuitry. In theory, the Op-amp value, when the magnetic field is absent, should ideally be the midpoint of the potential difference across it, which in this case is 1.67V. This corresponds to an analogue output of 1398, considering the microcontroller's resolution of 12, where 5V corresponds to an analogue output of 4096. However, in reality, the measured reading for zero magnetic field is approximately 1750. This deviation could be attributed to factors such as the influence of surrounding magnetic fields or other unknown factors.

Figure 2.3.6 (a), Figure 2.3.6 (b) and Figure 2.3.6 (c) show the wave in the oscilloscope corresponding to three different polarities. In Figure 2.3.6 (a), the magnetic field is zero, and the voltage level is 1.56 V. This voltage level is set to the reference level. In Figure 2.3.6 (b), the polarity is “N”, and the voltage level is 2.36 V, higher than the reference level. In Figure 2.3.6 (c), the polarity is “S”, and the voltage level is 0.84 V, which is lower than the reference level.

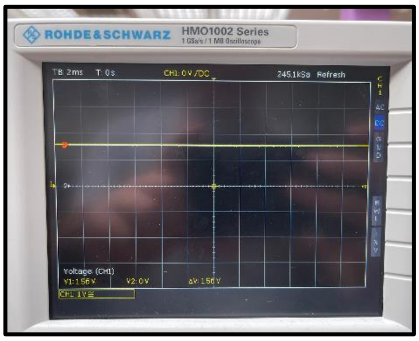


Figure 2.3.6 (a): Neutral

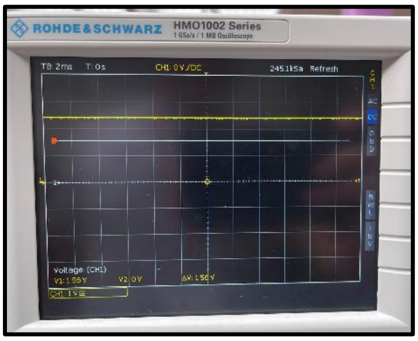


Figure 2.3.6 (b): North

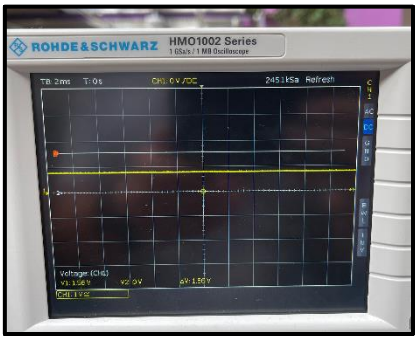


Figure 2.3.6 (c): South

5.2 Final Test

The final test is carried out on the computer, aiming to prove the program's correctness. Figure 2.3.7 shows the result of the testing. The polarity changes after the magnetic is placed differently. Although there are some fluctuations in the magnetic field strength, the polarity result is accurate and consistent.

Figure 2.3.7: Result shows that polarity changes after the magnetic is placed differently

Section 4: Remote Control and Web Server

1. Introduction and Objectives

A motor driver module, an integrated H-bridge with decode logic, is provided for EEE Rover. H-bridges allow current to be driven in either direction through the central board by connecting one terminal to the positive supply and the other to the ground [3].

Two digital inputs: direction (DIR) and PWM, are decoded for H-bridges using MOSFETs. The objective for controlling the movement is to figure out the decode logic and utilise it in Arduino programs. Distinct forward, backwards, left, and suitable functions must be developed to control the movement.

The remote control should be developed to transmit sensor data and control commands. Also, for data detected by the sensor to be displayed, a webpage must be established with a user-friendly user interface (UI).

2. Component List

Section	Components	Quantity
Motor Control	H-Bridge Motor Driver TC78H620 Module	2
	Wires	Several
	ESP32 Dev Kit V1	1
	16 Pin Female Header	2
	DC Motor Assembly	2

Table 2.4.1: Component List

3. Principles

3.1 Pulse-width modulation (PWM)

Pulse-width modulation (PWM) is an approach to control average power delivered by an electrical signal [15]. PWM controls the speed by varying duty cycle, which is quantified by the function `analogWrite(pin, value)`, with the relationship illustrated in Figure 2.4.1 [16]. For the `value` parameter in the `analogWrite` function, its value should be within the range of 0 - 255, where a high value indicates more energy received by the motor, hence leads to increased speed.

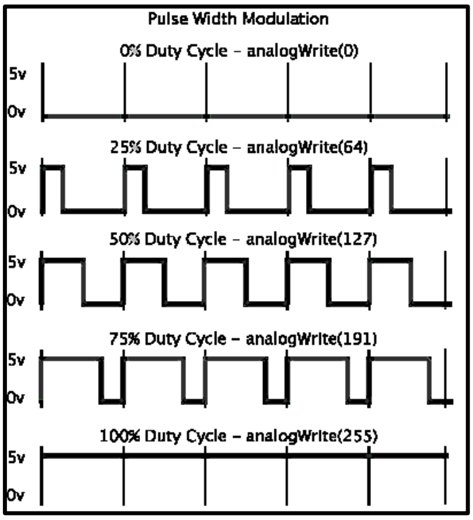


Figure 2.4.1: Relationship of duty cycle with analogWrite value

4. Design Details

Remote Control and Web Server section consists of three submodules, as illustrated in Figure 2.4.2.

(1) Decode Logic for Movement: Develop four different functions for movements in four directions.
(2) Web Server: Establish a connection between wireless devices with high bandwidth and fast respond time.
(3) Webpage Design: Design UI for the control of rover and display data collected from sensors.

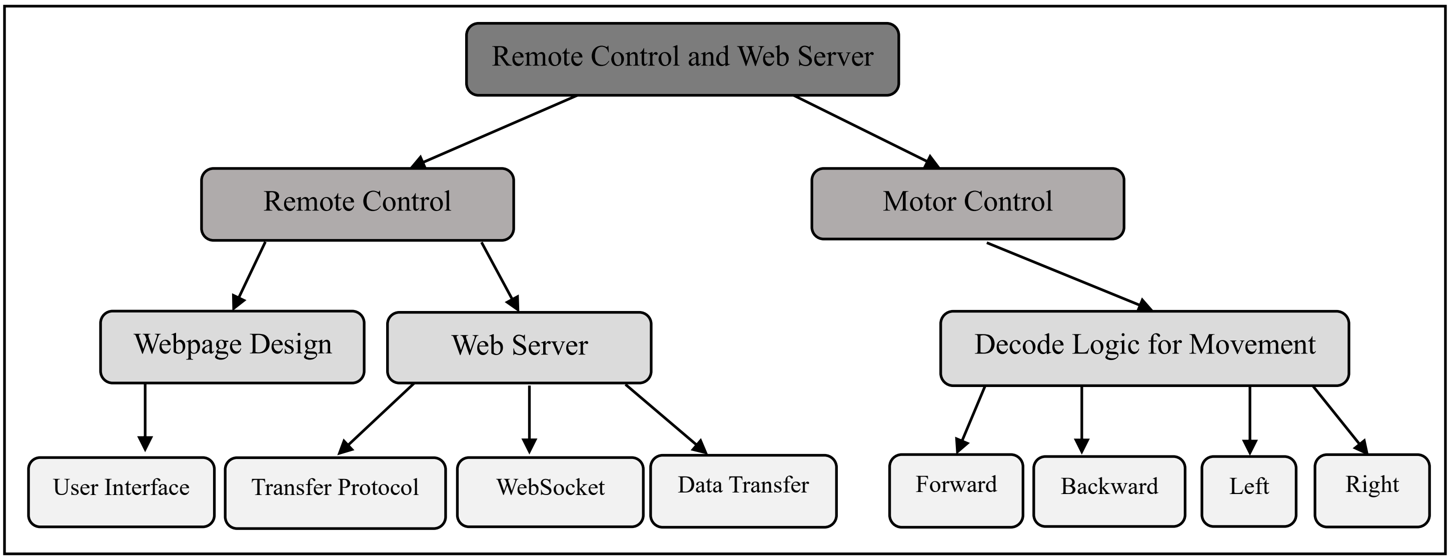


Figure 2.4.2: Submodules of Remote Control and Web Server

The design process can be concluded into five stages:

(1) General discussion and divide the objective into submodules.
(2) Figure out decode logic for movement.
(3) Develop different functions for movement.
(4) Determine transfer protocol and create connections between devices and the rover.
(5) Design the webpage.

4.1 Decode Logic for Movement

4.2.1 Overview

The movement of the rover is caused by the rotation of the two motors. For each motor, its operation is determined by two digital input signals, DIR and PWM, as illustrated in the Figure 2.4.3 [3].

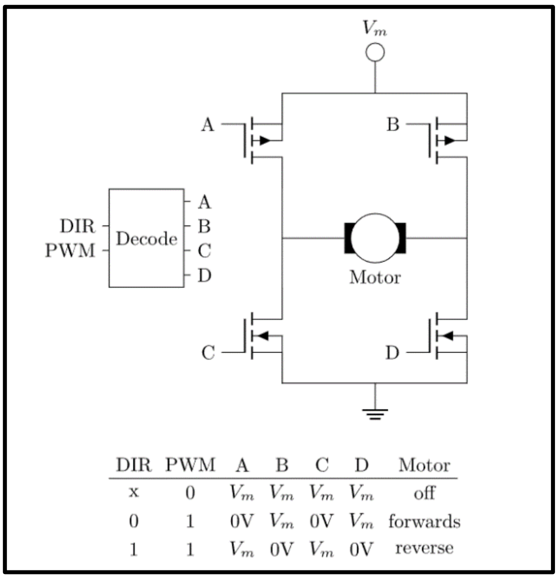


Figure 2.4.3: Decode logic for motor operations

Since the rover consists of two motorised wheels, the operations of two motors define the moving direction of the rover, with a relevant relationship illustrated in Figure 2.4.4.

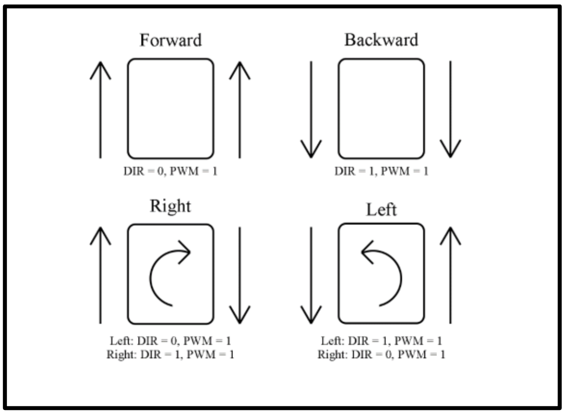


Figure 2.4.4: Decode logic for movement of the rover

In this sense, a function can be applied to determine the DIR value based on the desired direction of movement, while the duty cycle can be manipulated in the respective functions corresponding to different moving directions.

4.2.2 DIR Determination

DIR stands for direction and functions as binary input. DIR controls the direction of rotation of the motor, where a LOW value corresponds with a "positive" direction. The value of DIR can be set using `digitalWrite(pin,value)` function.

Based on the truth table illustrated in Figure 2.4.3, the function `control_dir` which derives the value of DIR is demonstrated in Program 2.4.1.

```
void control_dir()
{
  if (leftmotor >= 0)
  {
    digitalWrite(left_dir, LOW); // left wheel moving forwards
  }
  if (leftmotor < 0)
  {
    digitalWrite(left_dir, HIGH); // left wheel moving backward
  }
  if (rightmotor >= 0)
  {
    digitalWrite(right_dir, LOW); // right wheel moving forward
  }
  if (rightmotor < 0)
  {
    digitalWrite(right_dir, HIGH); // right wheel moving backward
  }
}
```

Program 2.4.1: `control_dir` function

4.2.3 Forward movement

The function for forward movement is demonstrated in Program 2.4.2. An unsigned constant `delta_v` = 254, at the edge of the constraint, is introduced to set the value for `analogWrite` of each motor, which should be within the range of 0 – 255.

In analogWrite, the `control_dir` function is called to determine the DIR value for the movement, which in this case is LOW for both motors. Subsequently, `analogWrite` is called for the PWM pins, resulting in the rover moving forward.

A constraint is set in the function to constrain the PWM values within 0 – 255. Otherwise, it is found that the rover will not return to a stationary state if the buttons are pressed too frequently.

```
void forward()
{
    Serial.println("forward");
    leftmotor = delta_v;
    rightmotor = delta_v;
    control_dir();
    analogWrite(left_pwm_pin, abs(constrain(leftmotor, -255, 255)));
    analogWrite(right_pwm_pin, abs(constrain(rightmotor, -255,
255)));
}
```

Program 2.4.2: `forward` function

4.2.4 Backward Movement

The function used to implement backward movement, as demonstrated in Program 2.4.3, is like the `forward` function. Negative value of `delta_v` is sent to both motors, resulting in moving backwards.

```
void backward()
{
    Serial.println("backward");
    leftmotor = -delta_v;
    rightmotor = -delta_v;
    control_dir();
    analogWrite(left_pwm_pin, abs(constrain(leftmotor, -255, 255)));
    analogWrite(right_pwm_pin, abs(constrain(rightmotor, -255,
255)));
}
```

Program 2.4.3: `backward` function

4.2.5 Leftward Movement

The function used to implement leftward movement, as demonstrated in Program 2.4.4. Positive value for `delta_v` is sent to left motor, while negative is sent to right motor.

```
void moveleft()
{
    Serial.println("Moving left");
    leftmotor = delta_v;
    rightmotor = -delta_v;
    control_dir();
    analogWrite(left_pwm_pin, abs(constrain(leftmotor, -255, 255)));
    analogWrite(right_pwm_pin, abs(constrain(rightmotor, -255,
255)));
}
```

Program 2.4.4: `moveleft` function

4.2.6 Rightward Movement

The function used to implement leftward movement, as demonstrated in Program 2.4.5. Negative value for `delta_v` is sent to left motor, while positive is sent to right motor.

```
void moveright()
{
    Serial.println("Moving right");
    leftmotor = -delta_v;
    rightmotor = delta_v;
    control_dir();
    analogWrite(left_pwm_pin, abs(constrain(leftmotor, -255, 255)));
    analogWrite(right_pwm_pin, abs(constrain(rightmotor, -255,
255)));
}
```

Program 2.4.5: `moveright` function

4.3 Web Server

4.3.1 Overview

EEE Rover needs to be able to transmit data wirelessly between itself and other connected devices such as a phone or computer. This data involves motor control commands being sent to the rover or sensor data being transmitted from the rover.

Connections are established between the rover and wireless devices with high data bandwidth and fast response time for precise control of the motor's movement. A UI (User Interface) is also designed so that users can control the motor with buttons, and receive data detected by sensors.

4.3.2 Transfer Protocol

Various protocols allow for data transfer across Wi-Fi, among which the most notable protocol is the Hyper Text Transfer Protocol (HTTP). HTTP is the main protocol used for accessing webpages built on HTML across internet, which provides convenience when developing the UI. The microcontroller unit uses an Arduino Wi-Fi library imported in the source code is connected to Wi-Fi through Program 2.4.1.

```
    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED)
    {
        delay(500);
        Serial.println("Connecting to WiFi..");
    }

    Serial.print("Connected to the WiFi network: ");
    Serial.println(ssid);
    Serial.print("IP Address: ");
    Serial.println(WiFi.localIP());
```

Program 2.4.1: Wi-Fi set up

4.3.3 WebSocket

An alternative to HTTP is the WebSocket protocol, which not only provides full compatibility with HTML, but also enables a continuous connection allowing for fast asynchronous data transmission.

Using WebSockets requires a client and server. In this case, the microcontroller is configured as a server and a webpage, with a UI, acts as a client running on a separate device connected to Wi-Fi. Program 2.4.2 illustrates the startup code and how devices are connected.

```
    ws.onEvent(onWsEvent);
    server.addHandler(&ws);

    server.begin();
}

void loop()
{
    ws.cleanupClients();
}
```

Program 2.4.2: Server startup code and device connection

The clients, which include all webpages running on connected devices, use embedded JavaScript. Program 2.4.3 demonstrates the program that opens and closes the connection to the server.

```
const ws = new WebSocket("ws://172.20.10.14:80/ws");

ws.onopen = function () {
    console.log("Connection opened");
}
ws.onclose = function () {
    console.log("Connection closed");
}
```

Program 2.4.3: Open and close the connection to the server

4.3.4 Data Transfer

JavaScript Object Notation (JSON) is a prevalent method for transferring data over the internet, which is introduced for data transfer of EEE Rover. Both microcontrollers and JavaScript possess the ability to decode and encode JSON data, making JSON the foundation of communication between these connected devices.

The JSON messages consist of two sections: the header and the data. Program 2.4.4 shows an example illustrating the inclusion of message header and the corresponding data, which is the content of the message. In this way, various functions, from sending control commands to toggling the built-in LED on and off, can be achieved through communication.

```
json = {
    "header" : "message",
    "data" : "This is a message from ...."
}
```

Program 2.4.4: Inclusion of message and corresponding data

A few headers are included for the decode of JSON data which is available on the GitHub repository [17]. As an example, Program 2.4.3 illustrates how the ESP32 decodes the JSON data for message header. On the client, the decoding process is similar, but uses JavaScript functions instead.

```
    StaticJsonDocument<200> doc;
    // Serial.println("Allocated JSON memory");
    if (type == WS_EVT_CONNECT)
    {
        Serial.println("Websocket client connection received");
    }
    else if (type == WS_EVT_DISCONNECT)
    {
        Serial.println("Client disconnected");
    }
    else if (type == WS_EVT_DATA)
    {
        // Serial.println("Deserializing JSON data");
        DeserializationError error = deserializeJson(doc, data);
        if (error)
        {
            Serial.print(F("deserializeJson() failed: "));
            Serial.println(error.c_str());
            return;
        }
        Serial.println("Obtaining header");
        const char *header = doc["header"].as<const char *>();
        if (strcmp(header, "message") == 0)
        {
            // Serial.println("Attempting to access message data");
            const char *message = doc["data"].as<const char *>();
            Serial.print("Message received: ");
            Serial.println(message);
        }
```

Program 2.4.5: Arduino JSON decoding

4.3.5 Discussion

For enabling data transfer using HTTP, the XMLHttpRequest protocol is used. XMLHttpRequest protocol is built upon the HTTP stack, which provides great versatility, strong compatibility and ensures precise data transmission.

One widely used approach for data transfer using HTTP is through a protocol known as XMLHttpRequest. This protocol, built upon the HTTP stack, provides great versatility and ensures precise data transmission. It also offers strong compatibility across various devices.

An example of interaction between the HTTP web server and the microcontroller is shown in Program 2.4.6 and Program 2.4.7, where the former is for the client and the latter is for the server. The client sends a request to the server to toggle the microcontroller's built-in LED and to also get the LED state. The server will successfully change the LED state and feed this back to the client to update it on the UI.

```
void toggleLED()
{
  digitalWrite(LED_BUILTIN, !digitalRead(LED_BUILTIN));
  if (digitalRead(LED_BUILTIN))
    server.send(200, F("text/plain"), F("ON"));
  else
    server.send(200, F("text/plain"), F("OFF"));
}
```

Program 2.4.6 Interaction between client and HTTP server

```
        var xhttp = new XMLHttpRequest();
        xhttp.onreadystatechange = function () {
            if (this.readyState === 4 && this.status === 200) {
                document.getElementById("led-state").innerHTML =
this.responseText;
            }
        };

        function toggleLED() {
            xhttp.open("GET", "/toggleLED", true);
            xhttp.send();
        }
```

Program 2.4.7 Interaction between HTTP server and client

However, during testing, it was observed that requests between devices were relatively slow while using xhttp. As a result, the WebSocket protocol, was chosen instead, which is explained in 4.3.3. This protocol not only maintains compatibility with HTML but also addresses the speed concerns encountered with XMLHttpRequest and allows for quicker data transmission.

This decision also influences which microcontroller is used. Most microcontrollers are not compatible with WebSockets, so the ESP32 microcontroller device was chosen. It is lighter, provides ample speed for our processing tasks, takes up a smaller footprint and has extensive documentation on WebSockets.

4.4 Webpage Design

The requirements for the user interface are summarised as follows:

(1) Control the movement of the rover using W, A, S and D buttons on keyboard by sending motor commands to microcontroller.
(2) Receive sensor data from microcontroller.
(3) Store the sensor data in the table and display it.

All these requirements can be configured into a webpage shown in Figure 2.4.5. Pressing the keyboard buttons calls JavaScript functions which send requests to ESP32, while Sensor Data and LED State are constantly listening to any messages sent from ESP32 so that data can be updated.

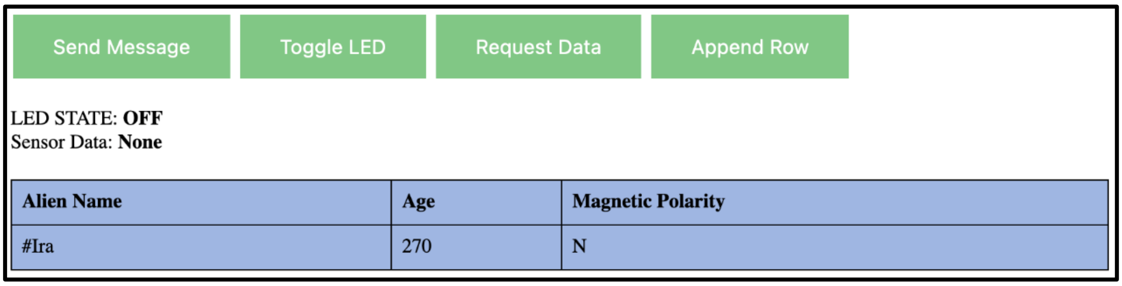


Figure 2.4.5: Screenshot of Webpage

5. Outcomes and Performance

Regarding the motor control aspect, the decode logics have been verified to accurately represent the motor's operation through the truth table. When coupled with the remote-control functionality, the rover is capable of moving in the intended direction.

As for the web server component, the final design integrates the WebSocket protocol, JSON formatting and a webpage to create a cohesive module. This module enables users to interact with the ESP32 over Wi-Fi, allowing them to gather data and control the EEE Rover. Extensive testing has demonstrated the web server's successfully reception of data from the ESP32, as well as the seamless transmission of control commands in the opposite direction. The transmission speed meets expectation, indicates that the delay falls within an acceptable range, thereby not impeding the controls of the EEE Rover.

Section 5: Mechanical Design

1. Introduction and Objectives

Mechanical design is based on the previous design of the EEEBug. Although the EEEBug provides a robust option for the base of the rover, it is necessary to build a design suitable for accommodating the sensor parts while in the meantime enhancing the rover's performance.

The objective includes reducing the weight of the chassis, maximising the use of space, enhancing stability, accommodating the sensors, and increasing the speed of the rover.

2. Component List

Section	Components	Quantity
Chassis	3mm Black Acrylic 600×300 mm	1
	M2 Screws	12
	M2 Nuts	12
	M3 Nuts	4
	M3×10mm Pozi Pan Head Screws	2
	M3×25mm Pozi Pan Head Screws	2
	Pozi Pan Heads Self tapping Screws No4 X 3/8in （9.5mm）	6
	M4 x 10 Pozi Pan screws	2
	M4 nuts	2
	M4 screws	4
Wheel	70×25mm Wheel and Tyre	2
	Guitel Hervieu Castor Wheel, 12kg Capacity, 25mm Wheel	1
Battery	PP3 Duracell Plus MN1604 9V Battery	1
	Press Stud Connector For PP3	2
	LM7805CT T0220 5V +VE O/P 1A	1
	PP3 Battery Case	1

Table 2.5.1: Component List

3. Design Details

3.1 Overview

The mechanical design consists of four submodules, as illustrated in Figure 2.5.1.

(1) Base Design: Design a new base that maximises space usage and accommodates new components.
(2) Wheel Configuration: Select and arrange the wheels.
(3) Power Improvement: Improve speed and manoeuvrability.

(4) Support Platform: Support for the microcontroller and enable the sensors to be close to the alien simulator for more precise detection.

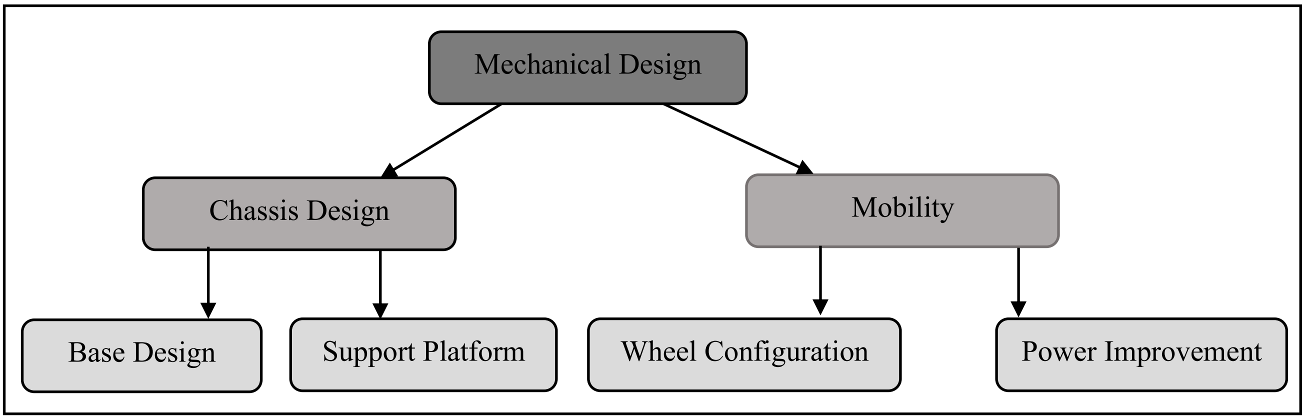


Figure 2.5.1: Submodules of mechanical design

The design process can be separated into six stages:

(1) General discussion and divide the objective into submodules.
(2) Discussion of the structure of the rover.
(3) Selection of the components.
(4) Designing and constructing the first chassis.
(5) Assemble sensing units and test.
(6) Second design, construction and assembling.

3.2 Base Design

3.2.1 Weight Reduction

The existing chassis design of the EEEBug [18] serves as the foundation for the base design of EEE Rover. To reduce the weight, it was decided to reduce the length of the base. The approach is to place the microcontroller above the PCB board, supported by four rods. Also, the dimensions of the stripboards, which integrates the sensors and microcontroller, are measured, whose values are as follows:

(1) Sensing Unit: 69.36 mm × 68.01 mm
(2) Microcontroller: 80 mm × 40 mm

These dimensions are measured and considered while designing the base to maximise the usage of space.

3.2.2 Balancing

A challenge for the new design is balancing the weight. The support platform in the front of the rover needs to be balanced. Otherwise, the rover may tip over. The wheels are positioned closer to the rear end of the chassis to counterbalance this weight distribution and optimise speed.

3.2.3 Safety Measures

The edges of the base are made with rounded corners. This design improves safety and user-friendliness.

3.2.4 Outcomes

The design is completed using Fusion 360, highlighting the reduction of weight, accommodation of necessary components and safety improvements. A third-angle projection sketch is shown below in Figure 2.5.2.

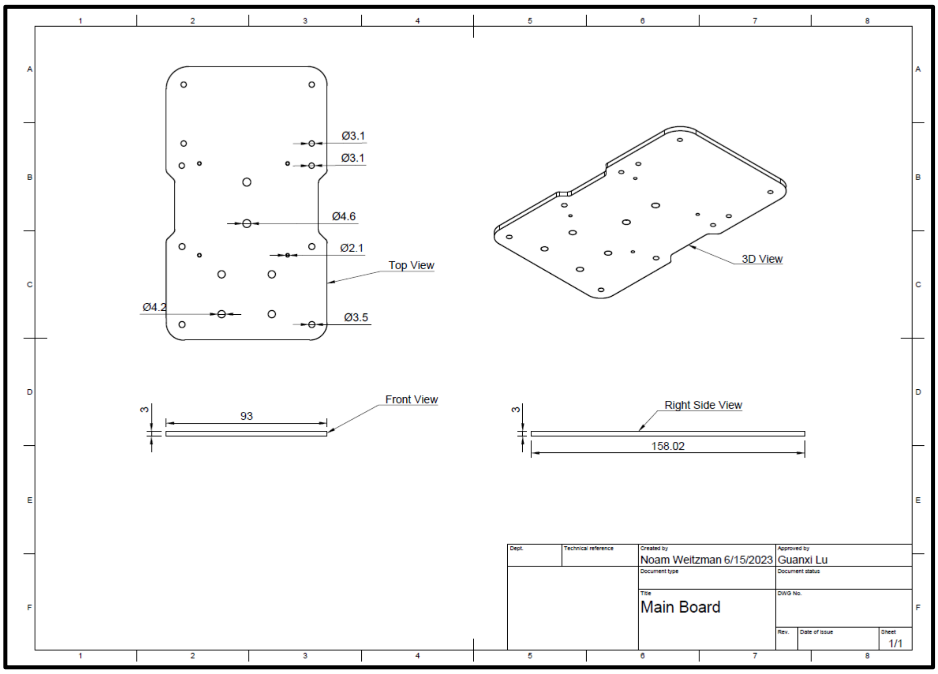


Figure 2.5.2: Sketch for Base

3.3 Support Platform

A support platform was added in the front of the rover, allowing it to hold the copper coils used for the radio wave sensor. The platform would enable the coil to be above the alien during measurement, providing optimal results.

The support platform consists of three parts: supporting rods for the microcontroller, two supporting pillars and a Y-shape platform. The rods support the microcontroller above the PCB board, and the supporting pillar fixes the platform to the chassis while the platform holds the coils.

The pillar is designed as a triangle to allow better performance since it allows two connection points with the base. The novel design of the Y-shape platform makes it feasible to hold the coil with minimised weight.

The third angle projection drawing of the rods, pillar and Y-shape platform is shown in Figure 2.5.3, Figure 2.5.4, and Figure 2.5.5, respectively.

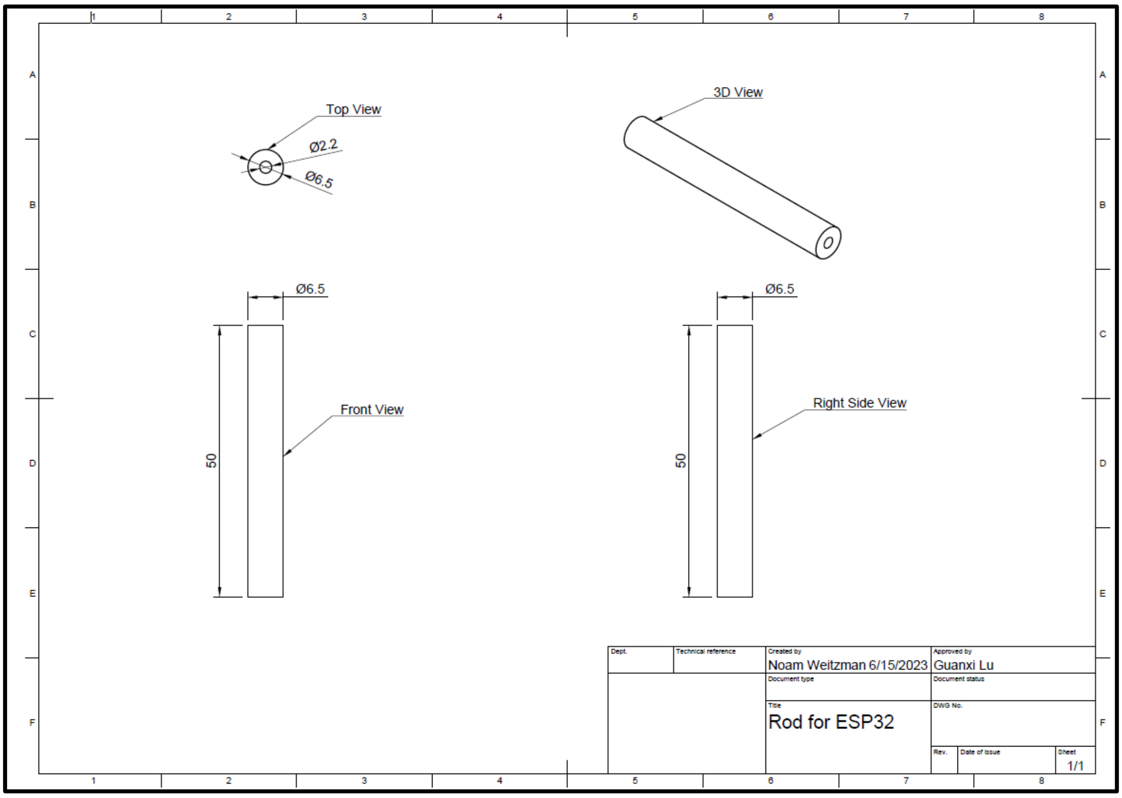


Figure 2.5.3: Sketch for rods

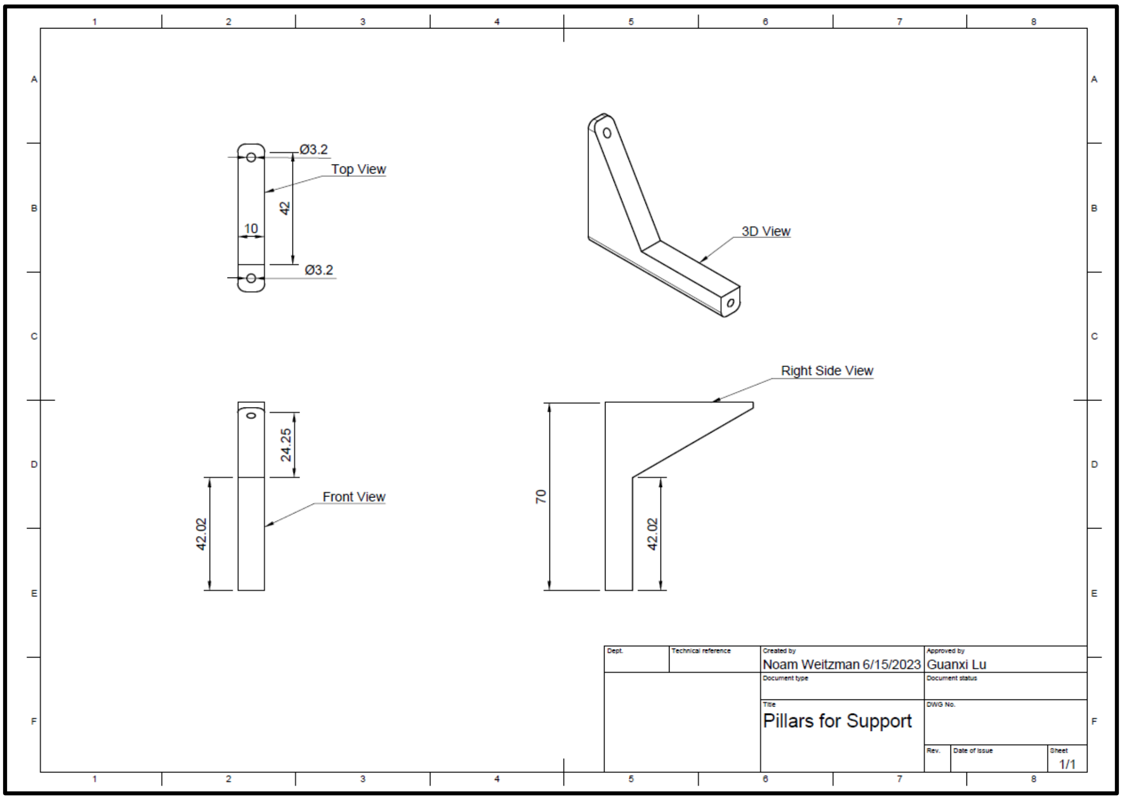


Figure 2.5.4: Sketch for pillar

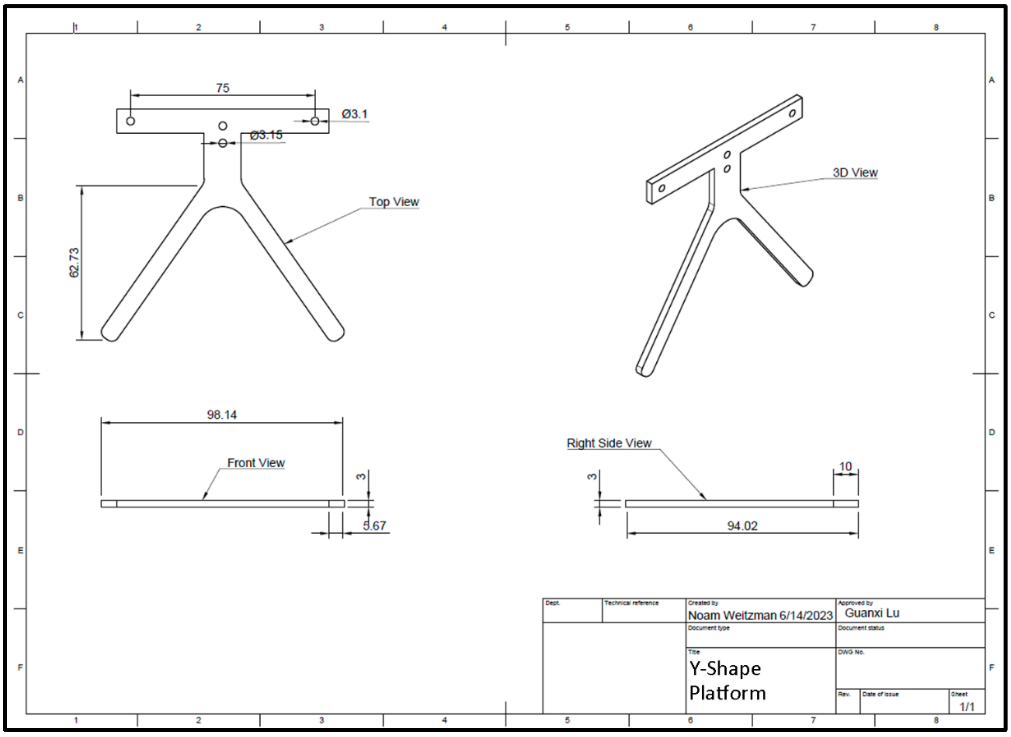
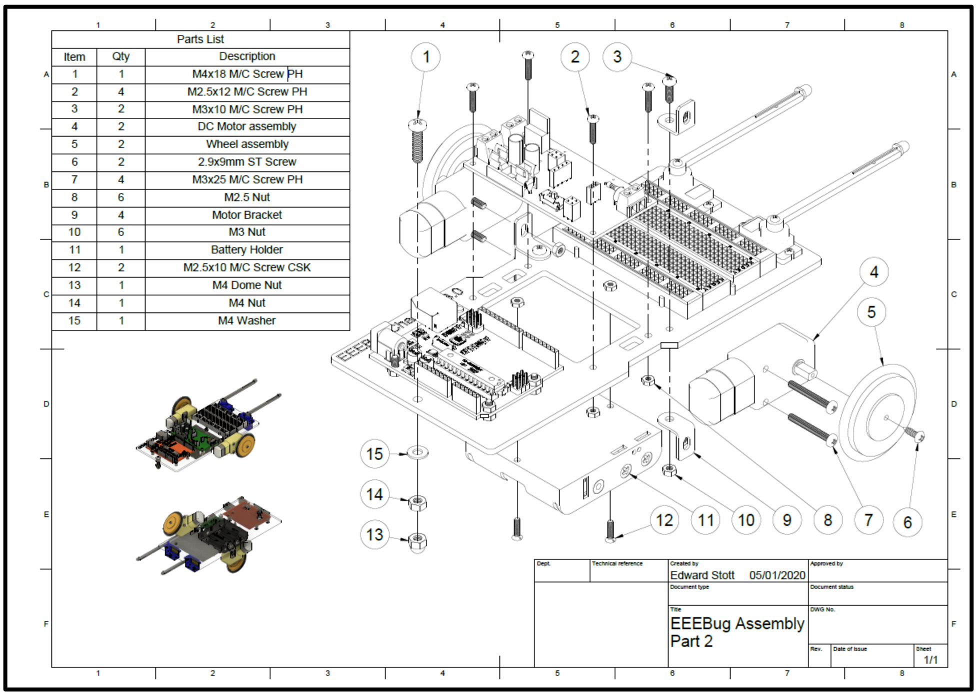


Figure 2.5.5: Sketch for Y-shape Platform

3.4 Wheel Configuration

In the design of EEEBug, the two wheels driven by motors enables movement, and a M4x18 M/C Screw with M4 Dome Nut are combined to balance the rover, as illustrated in Figure 2.5.6 [19]. This would lead to the problem with the screw that can lead to extra frictional force during movement.



Parts List

Item	Qty	Description
1	1	M4x18 M/C Screw PH
2	4	M2.5x12 M/C Screw PH
3	2	M3x10 M/C Screw PH
4	2	DC Motor assembly
5	2	Wheel assembly
6	2	2.9x9mm ST Screw
7	4	M3x25 M/C Screw PH
8	6	M2.5 Nut
9	4	Motor Bracket
10	6	M3 Nut
11	1	Battery Holder
12	2	M2.5x10 M/C Screw CSK
13	1	M4 Dome Nut
14	1	M4 Nut
15	1	M4 Washer

Figure 2.5.6: Assembly drawing of EEEBug

In the design of EEE Rover, a non-motorised 25 mm castor wheel substitutes the screw and dome nut, resulting in reduced frictional resistance. The original wheels are replaced by two 70×25mm wheels, which increases the speed significantly by converting the torque from the motor more efficiently. Larger width of the wheels also helps maintain stability.

3.5 Power Improvement

In the design of EEEBug, the power unit consists of four AA 1.5V batteries. The design of EEE Rover adopts the selection of a PP3 9V battery, which provides 1.5 times more power than the original set-up with less weight. An LM7805CT voltage regulator is also introduced to regulate the DC voltage.

3.6 Discussion

One option for wheel configuration is to use four motors instead of two. Even though it can enhance the speed greatly, the weight optimisation of the rover is significantly compromised since four AA 1.5V batteries and two more motors need to be added, which leads to a substantial increase in weight.

The chassis and the y-shape platform are fabricated using laser cutting instead of 3D printing. The decision is primarily driven by the fact that the prototype created with a 3D printer does not possess the required strength to support all the components adequately, leading to instability.

4. Outcomes and performance

As shown in Figure 2.5.8, a preliminary chassis was designed and put into production to test the design idea. Compared to the base of EEEBug, the preliminary design reduced the size and rearranged the space by putting the microcontroller above the PCB board.

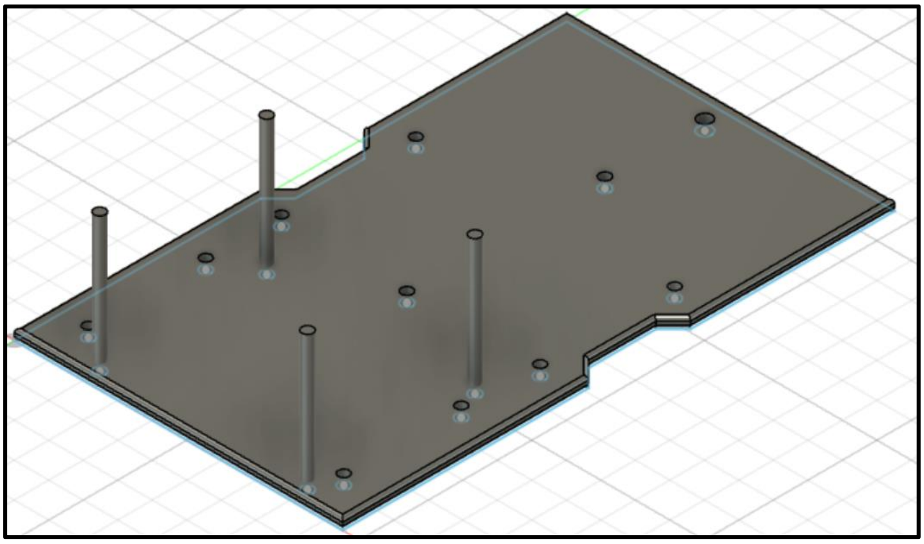


Figure 2.5.8 Preliminary chassis

Both 3D printed and laser cut versions are compared to decide the superior option, the photo of which can be viewed in Figure 2.5.9.

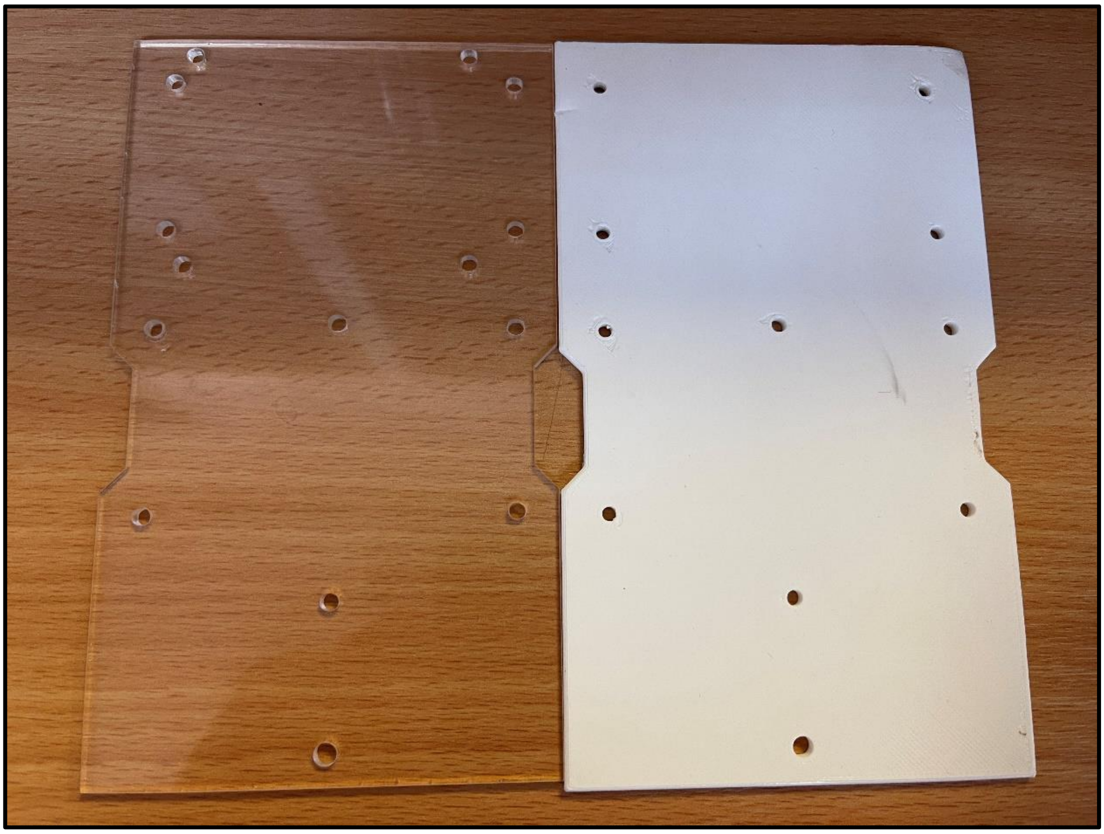


Figure 2.5.9: 3D printed and laser cut versions of chassis

Following the resolution of uses encountered in the initial design, including the replacement of the initial breadboard with a stripboard and a change in the microcontroller, a second chassis was manufactured. This final design prioritises optimised space utilisation and incorporates the addition of a castor wheel and a support platform for the coils. Upon completion of the design review, the power units, wheels, and support platform were integrated into the chassis, resulting in a fully assembled mechanical unit.

It is demonstrated that the mechanical design provides stable support for the components, with reduced weight and improved moving speed. Space usage is also optimised, and measures are taken to improve safety and user-friendliness. A photo of the mechanical unit is shown in Figure 2.5.10.

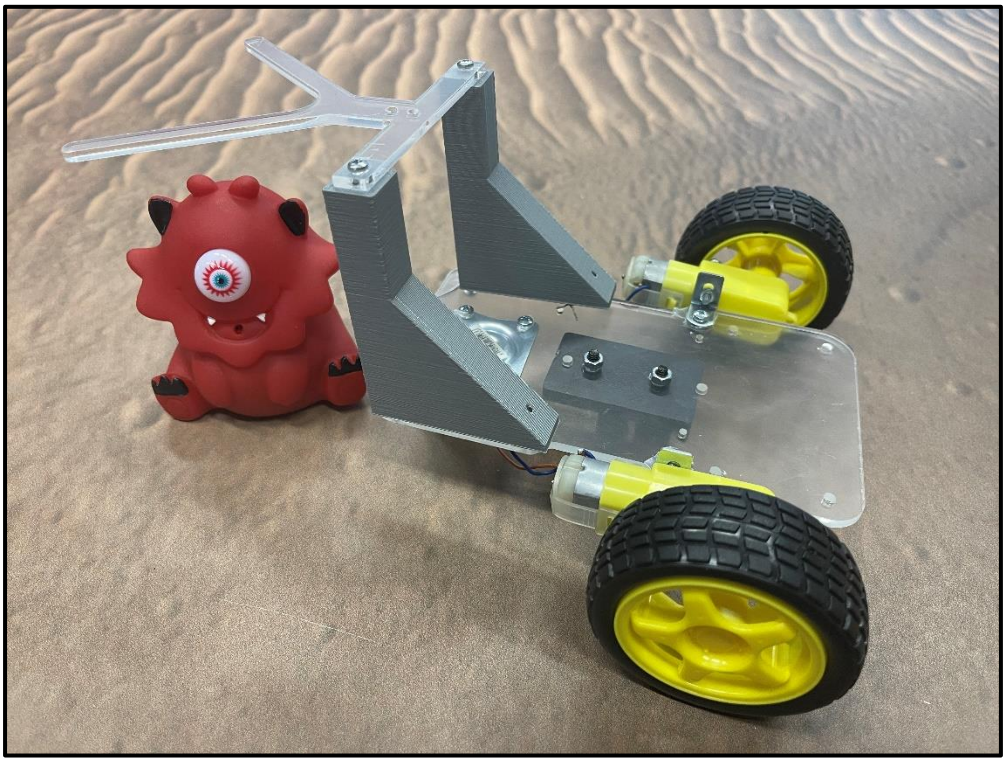


Figure 2.5.10 Mechanical Unit

Section 6: Overall Compilation and Integration

1. Overview

After all sensor units were constructed and tested and the mechanical design was built, the parts were assembled and constituted the complete EEE Rover.

To integrate the sensors, all sensing units were soldered into a single stripboard, which was fixed into the chassis, together with the microcontroller and other units. The assembly was tested, and minor modifications were made to perfect the design.

2. Sensor Integration

2.1 Stripboard Design

The sensor circuits are soldered on a stripboard. A circuit is first designed and reviewed using VeroRoute. Afterwards, the components are placed before being soldered and tested. For a stripboard, each column is connected in series by default, and in some cases, the default connection needs to be broken for the design.

In the software design illustrated in Figure 2.6.1, several techniques are introduced to minimise space utilisation and improve clarity.

(1) Components connected to the ground or power supply are placed in a single column and connected in series by default.
(2) The output node of a stage is aligned to the input node of the subsequent stage, avoiding extra soldering.
(3) Different colours of wires are used to indicate distinct potential levels.

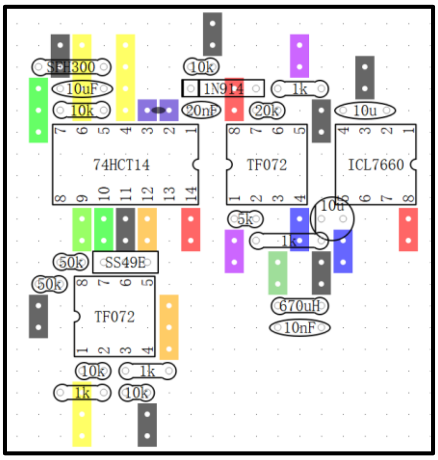


Figure 2.6.1: Stripboard software design

While soldering, checks are undertaken to ensure connectivity. Each time a connection is made, a multimeter set to continuity mode is used to ensure a solid electrical connection between the component's leg and the board.

2.2 Discussion

Stripboard was chosen as the platform to accommodate the three sensors for the following advantages it offers:

(1) Stripboards are lighter than breadboards, which assists with weight reduction.
(2) Stripboards provide more stable connections than breadboards, and components are less likely to fall off.
(3) Compared to printed circuit boards (PCB), the design process of stripboard is less complicated and has a shorter manufacturing period. It provides flexibility if design errors happen since soldering mistakes can be easily rectified.

3. Program Integration

3.1 Overview

The programs for the radio wave sensor, infrared sensor and magnetism sensor, remote control and web server are compiled as the complete program.

A few issues are considered during the process of compilation:

(1) For ESP32 to read and decode the data from the pin, extra time is required. Since the reading process will stop the motors from turning, sensor data should only be read when requested.
(2) Multiple sets of data are expected to be sent to the clients to be visually analysed for validity.
(3) The code size, including the built-in libraries, should be, at most, the flash size for ESP32.
(4) The code cannot run too long while gathering sensor data to prevent the ESP32 from timing out on the WebSocket connection and crashing.

3.2 Code compilation

All the code for each sensing unit are put into separate functions, for example, Polarity can be detected through `getPolarity` function, as shown in Program 2.6.1.

```
String getPolarity()
{

    Serial.println("Getting polarity");
    std::vector<String> samples;
    for (int i = 0; i < 10; i++)
    {
        int sensorValue = analogRead(magnet_pin);
        Serial.println("Sensor value: " + String(sensorValue));
        if (sensorValue > 1750)
        {
            samples.push_back("N");
        }
        else if (sensorValue < 1730)
        {
            samples.push_back("S");
        }
        else
        {
            samples.push_back("C");
        }
    }
    String mostFrequent = Find_MostFreS(samples);
    return mostFrequent;
}
```

Program 2.6.1: `getPolarity` function

All the functions are integrated in the `getData` function, as shown in Program 2.6.2. The `getData` function then compiles all the information into a JSON string and sends it to all connected clients with the header `sensorData`. In this way, the clients can parse this information and present it on the UI, as explained in 4.3.

```
void getData()
{
    for (int i = 0; i < 5; i++)
    {

        String name = getName();
        String polarity = getPolarity();
        long age = getAge();

        Serial.println("Data: Data compiled preparing to send");
        StaticJsonDocument<400> doc;
        doc["header"] = "sensorData";
        doc["data"]["name"] = name;
        doc["data"]["polarity"] = polarity;
        doc["data"]["age"] = age;

        String json;
        serializeJson(doc, json);
        Serial.println("Data: Sending sensory data to clients");
        ws.textAll(json);
    }
}
```

Program 2.6.2: `getData` function

A time tracker is added within the sensor functions to break the loop in case too much time is taken gathering the data, as illustrated in Program 2.6.3. Before the while loop is executed, the current time is recorded so that the time that has passed is reviewed at the end of every iteration. If that time exceeds a particular value, the Boolean variable timeout becomes TRUE, and the while loop is terminated. In this case, the function returns “timeout”. All the examples maintain the sampling method to record multiple values into a vector and take the most recurring value, improving noise tolerance.

```
    int time = millis();
    bool timeout = false;
    Serial.println("Beginning time: " + String(time));
    while (samples.size() < 5 && !timeout)
    {
        if (MySerial.available() > 0)
        {
            //Serial.println("Serial for name found: ");

            char asciiCharacter = MySerial.read();
            Serial.println("Ascii character found: " +
String(asciiCharacter));
            if (asciiCharacter == '#')
            {
                name+=asciiCharacter;
                samples.push_back(name);
                Serial.println("Name found: " + name + "At time: " +
String(millis() - time));
                name = ""; // Clear the output string
            }
            else
            {
                name += asciiCharacter; // Append the character to
the output string
                //Serial.println("Appending character to name " +
name);
            }
            delay(10);
        }
        if (millis() - time > 2000)
        {
            timeout = true;
        }
    }
    if (samples.size() == 0)
    {
        Serial.println("No name found");
        return "";
    }
    else if (timeout == true)
    {
        Serial.println("timeout");
        return "timeout";
    }

    mostFrequent = Find_MostFreS(samples);
    return mostFrequent;
}
```

Program 2.6.4: Implementation with time tracker

PART 3: Conclusion and Future Work

Section 1: Conclusion

Based on the findings presented in previous sections, it can be concluded that the EEE Rover project has successfully delivered a prototype that fulfils the expected functionalities.

In terms of the sensing unit, the radio wave, infrared, and magnetism sensors effectively detect corresponding signals. The associated programs efficiently process and decode these signals, accurately extracting Name, Age and Magnetism characteristics.

Regarding the mechanical unit, the designed structure ensures stable component placement and proximity of sensor to the alien simulator. Improvements in wheel configuration and power units have enhanced the rover's movement speed, resulting in improved efficiency.

The control unit allows wireless control of the rover, promptly responding to movement commands with minimal delay. When a request is sent, the user-friendly interface correctly displays the relevant characteristics.

What's more the design prioritises human and alien safety by maintaining a low profile while minimising the risk of injury to user or alien.

To summarise, as depicted in Figure 3.1.1, the EEE Rover offers an optimised, well-functioning, and user-friendly product. More information about the project can be accessed via the GitHub webpage: https://gitfront.io/r/Idrees/i34eARepTwR6/Group-Nimbion/.

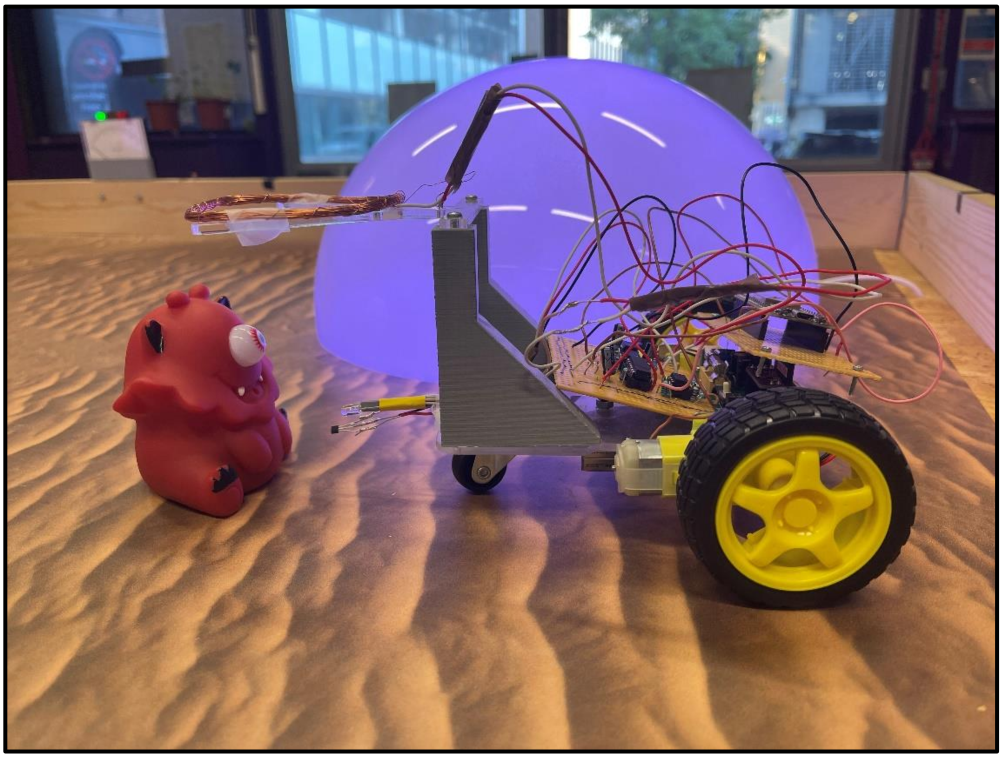


Figure 3.1.1: EEE Rover complete design

Section 2: Future Work

1. Sensing Unit

For the sensing unit, a few improvements can be made to further improve the precision and reduce detection time.

In terms of radio wave sensor, some delay can be caused by components. The delay can result in some glitches in the detection and may lead to failure in decoding. Also, the design of antenna can be improved so that the receiving ability is enhanced. For the infrared sensor, the detection range can be improved by a different configuration. A specialised infrared phototransistor that only accepts specific wavelengths can improve the detection. With regards to the magnetism sensor, a more advanced Hall effect sensor helps with accuracy and range of detection.

The sensing units can also be integrated onto a PCB board, which provides solid connections, a much smaller footprint and easy manufacturing.

2. Mechanical Design and Remote Control

There are opportunities for further improvement in the mechanical design and remote-control aspects, aiming to enhance moving speed, optimize weight, and introduce additional functions.

In the realm of mechanical design, the incorporation of more powerful motors can lead to increased speed capabilities. Additionally, the inclusion of Mecanum wheels can significantly improve the control experience.

In terms of remote control, the integration of a Virtual Joystick feature can provide mobile control functionality, while efforts should be made to further optimize the latency between input and response. Moreover, novel UI designs can be implemented to enhance the overall user experience.

References

[1] Ed, “EEERover Specifications and Technical Guide,” GitHub, Jun. 10, 2023. https://github.com/edstott/EEERover/blob/main/doc/brief.md

[2] E. Stott and E. Perea Borobio, “Engineering Design Project,” 2023. Accessed: Jun. 15, 2023. [Online]. Available: https://static.us.edusercontent.com/files/LmQXnReaXBE6HDaC2lPSOq37

[3] Ed, “EEERover Specifications and Technical Guide,” GitHub, Jun. 10, 2023. https://github.com/edstott/EEERover/blob/main/doc/README.md

[4] “Amplitude-shift keying,” Wikipedia, Nov. 11, 2021. https://en.wikipedia.org/wiki/Amplitude-shift_keying

[5] Wikipedia Contributors, “RLC circuit,” Wikipedia, Oct. 08, 2019. https://en.wikipedia.org/wiki/RLC_circuit

[6] “Schmitt trigger,” Wikipedia, May 27, 2020. https://en.wikipedia.org/wiki/Schmitt_trigger

[7] “74HC14 HEX INVERTERS WITH SCHMITT TRIGGER INPUTS,” 2013. Available: https://www.diodes.com/assets/Datasheets/74HC14.pdf

[8] “TL07xx Low-Noise FET-Input Operational Amplifiers.” Accessed: Jun. 15, 2023. [Online]. Available: https://www.ti.com/lit/gpn/TL074H

[9] “LT1366/LT1367/LT1368/LT1369 Dual and Quad Precision Rail-to-Rail Input and Output Op Amps” Accessed: Jun. 15,2023. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/1366fb.pdf

[10] “ICL7660, ICL7660A CMOS Voltage Converters,” 2010. Accessed: Jun. 15, 2023. [Online]. Available: https://www.renesas.com/www/doc/datasheet/icl7660.pdf

[11] “Characteristics of Phototransistors and Photodarlingtons,” johnloomis.org. https://johnloomis.org/ece445/topics/egginc/pt_char.html

[12] Wikipedia Contributors, “Hall effect,” Wikipedia, Aug. 16, 2019. https://en.wikipedia.org/wiki/Hall_effect

[13] J. Dargie, “Magnetic Sensing,” Standex Electronics, Jan. 12, 2021. https://standexelectronics.com/magnetic-sensing/magnetic-sensing-technologies-reed-switches-vs-hall-effect-switches/ (accessed Jun. 15, 2023).

[14] “SS39ET/SS49E/SS59ET Series Linear Hall-effect Sensor ICs.” Available: https://www.farnell.com/datasheets/2007294.pdf

[15] Wikipedia Contributors, “Pulse-width modulation,” Wikipedia, Jun. 21, 2019. https://en.wikipedia.org/wiki/Pulse-width_modulation

[16] “Basics of PWM (Pulse Width Modulation) | Arduino Documentation,” docs.arduino.cc. https://docs.arduino.cc/learn/microcontrollers/analog-output

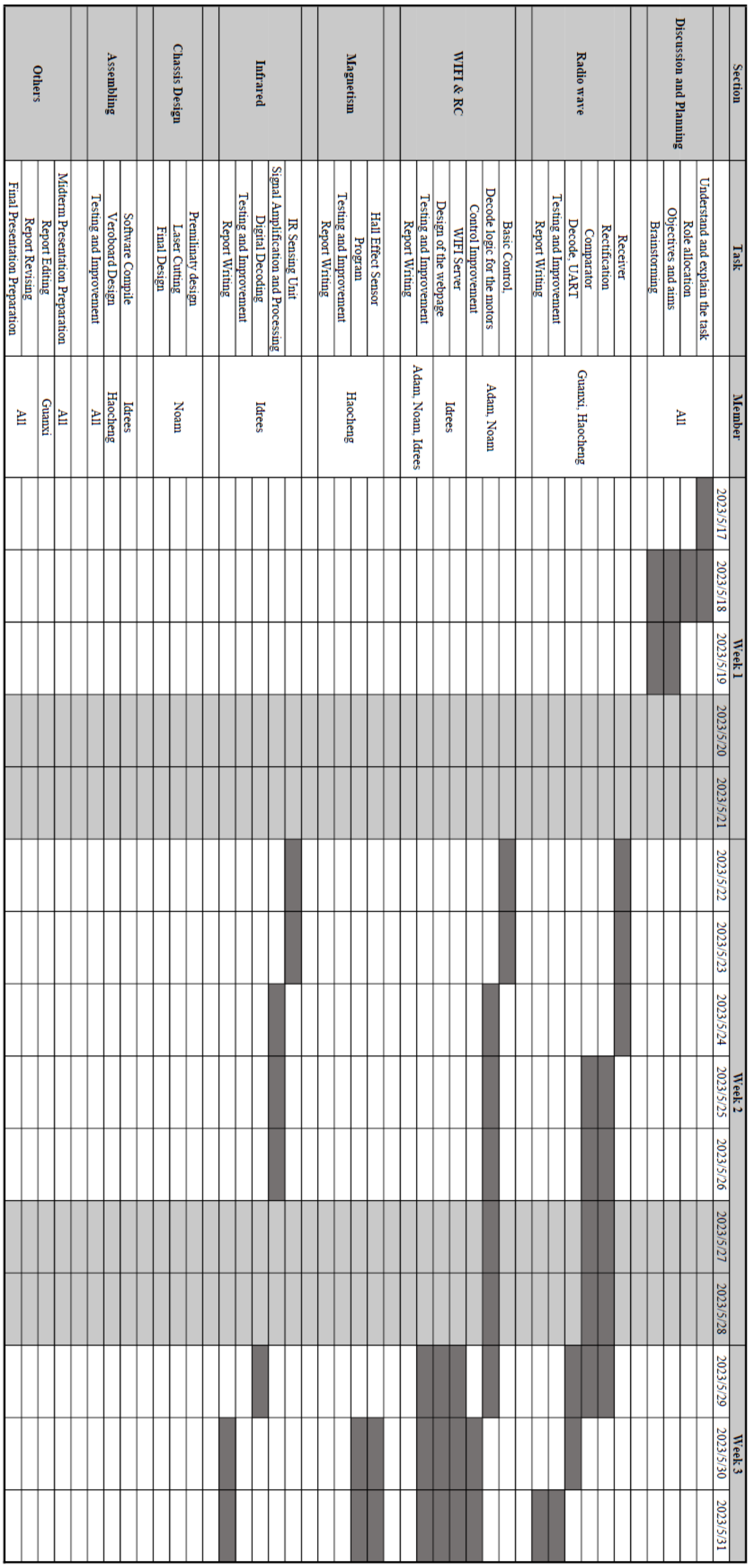
[17] BenoitBlanchon, “Documentation,” ArduinoJson. https://arduinojson.org/v6/doc/ (accessed Jun. 15, 2023).

[18] Ed, “EEERover Specifications and Technical Guide,” GitHub, Jun. 10, 2023. https://github.com/edstott/EEERover/blob/main/mech/chassis.dxf (accessed Jun. 15, 2023).

[19] E. Stott, “Assembly Drawing,” https://github.com/edstott/EEE1labs/blob/main/skills/graphics/EEELAB-EEEBUG-010_1-0.pdf (accessed Jun. 15, 2023).

Appendices

Appendix 1: Time Management - Gantt Chart



Section	Task	Member	Week 3				Week 4							Week 5							Week 6	
			2023/6/1	2023/6/2	2023/6/3	2023/6/4	2023/6/5	2023/6/6	2023/6/7	2023/6/8	2023/6/9	2023/6/10	2023/6/11	2023/6/12	2023/6/13	2023/6/14	2023/6/15	2023/6/16	2023/6/17	2023/6/18	2023/6/19	2023/6/20
Discussion and Planning	Understand and explain the task	All																				
	Role allocation																					
	Objectives and aims																					
	Brainstorming																					
Radio wave	Receiver	Guanxi, Haocheng																				
	Rectification																					
	Comparator																					
	Decode, UART																					
	Testing and Improvement																					
	Report Writing																					
WIFI & RC	Basic Control,	Adam, Noam																				
	Decode logic for the motors																					
	Control Improvement																					
	WIFI Server	Idrees																				
	Design of the webpage																					
	Testing and Improvement	Adam, Noam, Idrees																				
	Report Writing																					
Magnetism	Hall Effect Sensor	Haocheng																				
	Program																					
	Testing and Improvement																					
	Report Writing																					
Infrared	IR Sensing Unit	Idrees																				
	Signal Amplification and Processing																					
	Digital Decoding																					
	Testing and Improvement																					
	Report Writing																					
Chassis Design	Premilinaty design	Noam																				
	Laser Cutting																					
	Final Design																					
Assembling	Software Compile	Idrees																				
	Veroboard Design	Haocheng																				
	Testing and Improvement	All																				
Others	Midterm Presentation Preparation	All																				
	Report Editing	Guanxi																				
	Report Revising	All																				
	Final Presentation Preparation																					

Appendix 2: Ordering History

Order Date	Description	Supplier	Unit Price (£, Ex VAT)	Quantity	Total Price (£, Ex VAT)	URL
2023/5/19	SFH 309 FA-4/5 ams OSRAM, 24 ° IR Phototransistor, Through Hole 2-Pin 3mm (T-1) package	RS	0.352	10	3.52	https://uk.rs-online.com/web/p/phototransistors/6548019
2023/5/22	ICL76605CPA +VE TO -VE CONVERTER	EEDStores	1.74	1	1.74	http://www.ee.ic.ac.uk/storesweb/Page24.html
2023/5/24	74HCT14 SCHMIDT TRIGGER INVERTER	EEDStores	0.41	2	0.82	http://www.ee.ic.ac.uk/storesweb/Page24.html
2023/5/25	Honeywell Through Hole Hall Effect Sensor, Ratiometric Output, 4.5 → 10.5 V dc, 10.5V	RS	5.49	1	5.49	https://uk.rs-online.com/web/p/hall-effect-sensors/2166231
2023/6/6	70x25mm Wheel And Tyre	OneCall	1.56	4	6.24	https://onecall.farnell.com/unbranded/mc02755/wheeltyre-70-x-25mm/dp/MC02755?st=wheel%20and%20tyre
2023/6/8	STRIPBOARD MEDIUM 95mm X 127mm	EEDStores	1.44	2	2.88	http://www.ee.ic.ac.uk/storesweb/Page26.html
2023/6/12	Guitel Hervieu Castor Wheel, 12kg Capacity, 25mm Wheel	RS	2	1	2	https://uk.rs-online.com/web/p/castor-wheels/2391411
2023/6/12	BATTERY PP3 DURACELL PLUS MN1604 9V	EEDStores	3.84	1	3.84	http://www.ee.ic.ac.uk/storesweb/Page5.html
2023/6/12	PRESS STUD CONNECTOR FOR PP3	EEDStores	0.46	1	0.46	http://www.ee.ic.ac.uk/storesweb/Page5.html
2023/6/12	LM7805CT T0220 5V +VE O/P 1A	EEDStores	0.33	1	0.33	http://www.ee.ic.ac.uk/storesweb/Page24.html
2023/6/12	74HCT14 SCHMIDT TRIGGER INVERTER	EEDStores	0.41	1	0.41	http://www.ee.ic.ac.uk/storesweb/Page24.html
2023/6/14	3mm Black Acrylic 300 x 200	EEDStores	9.93	1	9.93	http://www.ee.ic.ac.uk/storesweb/Page35.html
2023/6/15	74HCT14 SCHMIDT TRIGGER INVERTER	EEDStores	0.41	1	0.41	http://www.ee.ic.ac.uk/storesweb/contents.html
2023/6/10	ESP32 Dev Kit Module V1	Kunkune	5.9	1	5.9	https://kunkune.co.uk/shop/esp32-esp8266/esp32-unsoldered-30-pins-nodemcu-development-board-wifi-bluetooth/
				Total	38.71	

Appendix 3: Product Design Specification

Product Design Specification			
Product: EEE Rover	Author: Group Nimbion	Date: 19/05/2023	Version: 1.0
1. Performance	Be able to sense the name, age, and magnetism of the alien, can be controlled remotely.		
2. Environment	Manoeuvrable negotiate the demo arena, work in normal temperature.		
3. Life in Service	One year, be able to work under high-intensity test.		
4. Maintenance	Batteries need to be replaced when run out.		
5. Target Product Cost	Strictly below £60.		
6. Competition	Superior to 80% other products in terms of speed and sensitivity of the sensor.		
7. Shipping	Should be comparative light and small to reduce shipping expense.		
8. Packing	Should be in regular shape to avoid extra packing costs.		
9. Quantity	One prototype.		
10. Manufacturing Facility	The prototype is manufactured in the laboratory with facilities including 3D printer and laser cutter.		
11. Customer	Dr Edward Stott.		
12. Size	Less than 25 cm × 20 cm × 15 cm.		
13. Weight	Less than 800 g, be able to pass sensitivity test.		
14. Materials	Majority plastic, also rubber for wheel and metal for electronic components.		
15. Product Life Span	Be able to last throughout intense test.		
16. Aesthetics, Appearance and Finish	Easy to recognise, with components tidied up and structures able to be identified; novel UI design.		
17. Ergonomics	Batteries are accessible to customer, easy to be held with one hand.		
18. Standards and Specifications	Accord with standards specified by project introduction.		
19. Quality and Reliability	High in quality from both hardware and software perspective, reliable in sensing and controlling.		
20. Shelf Life (storage)	One year, as long as batteries last.		
21. Testing	Be able to be tested with respect to sensing and remote control in demo arena.		
22. Processes	Circuit design, stripboard design, embedded programming, 3D printing, soldering, assembling.		
23. Time Scale	5 weeks in total, complete sensor design in three weeks before interim presentation.		
24. Safety	Avoid sharp parts in surface, wires should be wrapped.		
25. Company Constraints	Design and manufacture under budget and use existing resources.		
26. Market Constraints	Compete against products designed by other intelligent engineers.		
27. Patents, Literature and Product Data	Use components designed by other manufacturers, avoid violating other patents.		
28. Legal	Comply with both laws and rules of United Kingdom and Imperial College London.		
29. Political and Social Implications	Get rid of any social symbols, avoid offence to any group; promote environmental friendliness.		
30. Installation	No need to be installed, can be used directly.		
31. Documentation	A technical report with design and manufacture details.		
32. Disposal	Batteries should be disposed according to regulations, other parts should either be able to recycle, or be disposed appropriately.		