Constructing the EEERover

ELEC40006 Electronics Design Project - Dr Edward Stott

Team Helionix

June 2023

Maximilian Adam	Lucas Ng	Rolando Charles
CID: 02286647	CID: 02014979	CID: 02236652
Amin Mohamed	Krischad Pourpongpang	Harun Aslam
CID: 02216105	CID: 02251262	CID: 02032283

Imperial College London

Abstract

This report presents the design of a remotely-controlled rover that can explore a remote planet and survey the alien creatures that live there. It highlights the design choices and processes we made to implement a triangular Omni wheel rover able to detect and analyse radio waves, infrared radiation and static magnetic fields. The rover is controlled remotely via a WiFi Web server. Its modular and flexible design provides a good base for further structural, hardware and software improvements.

To achieve this we went through many iterations of planning, designing and prototyping for each module. Before implementing these modules into our final design.

Contents

1 System Design & Project Management

1.1 Background & Context

The EEERover is a remotely-controlled rover that can explore a remote planet and survey the alien creatures that live there. Using a variety of electromagnetic signals, the rover must find out the name, age and magnetic polarity of each alien [1].

1.2 Roles & Submodules

The following modules and submodules were defined in the preliminary stages of the planning process:

1. **Sensing**: research, design, implementation and testing of each of the 3 sensor submodules:
    - **Name**, design circuit to receive RF signals and write code to interpret these signals
    - **Age**, design circuit to receive IR signals and write code to interpret these signals
    - **Magnetism**, design circuit to detect a static magnetic field and write code to interpret it.
2. **Construction**: overall conceptualisation, design and construction of the rover. This was further split into 3 submodules:
    - **Printed Circuit Board**: redesign of EEEBug PCB for motor control, integration of sensor circuits
    - **Chassis**: design and manufacturing, as well as mounting of sensors
    - **Wheels**: design and manufacturing, as well as physical construction of Omni wheels
3. **Software**: 3 submodules of software design were defined:
    - Planning and design of the movement control system and user interface.
    - Verify all code produced to complete integration and testing.
    - Design how information is sent and received to and from the rover.

Management and progress tracking was carried out by keeping shared files and documents updated as often as possible. Biweekly team meetings were essential to our workflow. The key points from these meetings were summarised and used as a timeline and plan for development.

Subsequently, the following technical and non-technical responsibilities were assigned:

Member	Roles	Submodules
Rolando Charles	Project Manager	PCB and Chassis
Amin Mohamed	Treasurer and Electrical Engineer	Name (RF)
Krischad Pourpongpang	Electrical and Mechanical Engineer	Magnetism and Chassis
Harun Aslam	Electrical Engineer	Name (RF)
Lucas Ng	Software and Electrical Engineer	Age (IR) and Software
Maximilian Adam	Mechanical and Software Engineer	Chassis, Wheels and Software

1.3 Methods and Tools

To ensure that project development was organised, efficient and inclusive to all team members, various third party tools were used:

- **Autodesk Fusion360**, All mechanical components were designed in Fusion360. To ensure every team member had collaborative access a common project folder was created to store all design files. This repository was further split into key areas of the rover:
  - Chassis
  - Connecting components
  - Wheels

  Creating an easy-to-use environment to design in.
- **GitHub**, software implementation required frequent modifications and improvements. To keep track of progress a GitHub repository was created and frequently updated [2].
- **Microsoft Teams**, to communicate and share documents, resources and ideas.
  5 separate channels were created according to the defined submodules and academic deliverables.
  The platform was also used to store all important files related to our project and allow us to work on them collaboratively.
- **Gantt chart**, To have a complete overview of progress on our goals for this project we employed the use of a collaborative Gantt chart:

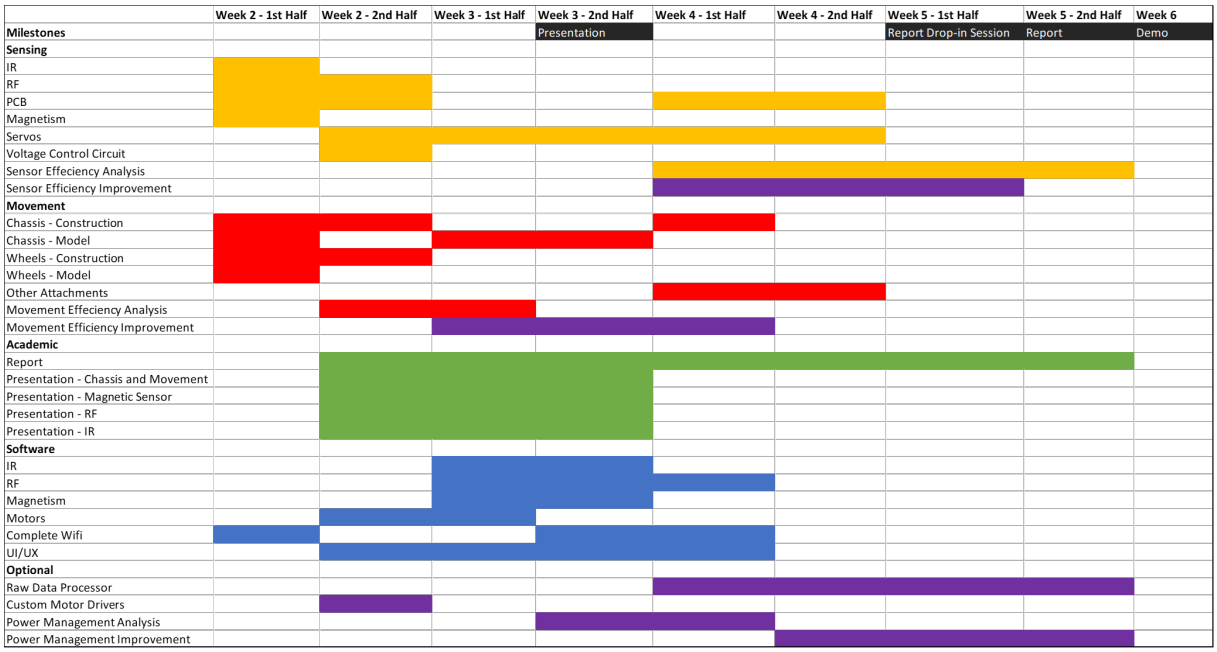


Figure 1: Gantt Chart

  This was helpful to keep on track and have a time reference for the project.
- **Budgeting spreadsheet**, A collaborative expense spreadsheet was created and updated by each team member as components were purchased, see Appendix. This was essential to eliminate unnecessary costs and allow for contingencies.

1.4 Design Requirements & Specifications

We identified qualitative and quantitative top-level specifications and assigned each to an individual sub module. Additionally, these specifications were fleshed out to split each design element into smaller design problems. Furthermore, we identified additional problems that were added to our project plan during testing.

1.4.1 Qualitative Specifications

Top-level qualitative requirements include how the overall EEERover should function and what it should look like as well as non-technical information such as cost. These specifications are summarised in Table 1.

Table 1: Design specifications and their subdivisions

Requirement	**Specifications**
Performance	1. Accurate and consistent in measuring the characteristics of the aliens 2. Lightweight 3. Manoeuvrable and can be controlled remotely 4. Responsive control 5. Stable and fast connection over WiFi 6. Mean time to failure should be at least time taken to complete the course
Environment	1. Consists of a smooth surface with uncrossable obstacles where aliens are randomly scattered across the arena 2. The rover must circulate smoothly through the arena and avoid obstacles 3. Different signals can be identified throughout the environment, the rover should be able to differentiate and navigate towards them
Testing	1. Tests and trials must be reproducible 2. Make sure no involuntary collisions occur with obstacles 3. Recreate an environment similar to the one that will be used during the demonstration 4. Test each submodule on its own 5. Keep track of results and compile tables summarising them
Materials	1. Chassis should be lightweight, cost effective and sturdy 2. Sensors and electrical components should be cost-effective, reliable, compact and easy to replace 3. Locomotive components should also be easy to replace 4. Materials should be environmentally friendly or recyclable

Target Product Cost	1. Below £60 2. Majority of budget should be allocated to buying circuit components 3. Some components can be recycled by desoldering from the EEEBug provided 4. Contingencies should be considered in case of mistakes and for prototyping
Aesthetics, Appearance, & Finish	1. The EEERover should look simple and design should be fairly intuitive 2. User interface should be easy to use and intuitive for non-experts 3. Appearance could also impact its performance, (e.g. loose wires could become a hazard)
Processes	1. The rover can be designed in CAD 2. Breadboards can be used for prototyping circuits 3. A custom PCB can be designed to house all of the electronic components or components can be soldered directly on a Veroboard 4. The chassis could be laser-cut or 3Dprinted, however the PCB could function as a chassis too 5. Motor and sensor mounts can be 3D printed

1.4.2 Quantitative Specifications

Quantitative specifications describe the technical characteristics of the individual submodules to achieve the desired product. Table 2 summarises these specifications. Note that non-essential submodules such as PCB and wheel design were identified later and assigned with the appropriate justifications.

Table 2: Qualitative requirements and their subdivisions

Submodule	**Requirements**
Name	1. Receive RF signal of a 61 kHz carrier frequency. 2. Modulation scheme employed should be two-level amplitude-shift keying 3. Decode the name which is encoded using ASCII character codes in UART packets with 1 start bit and 1 stop bit
Age	1. Receive infrared radiation with a wavelength of 950 nm 2. Decoding pulses in the 135 to 1000 Hz frequency range, with a pulse width of about 50 µs

Magnetic Field	1. Magnetic field directions should be detectable from an appropriate distance 2. The environment's magnetic field should be taken in consideration when determining the direction of the alien's magnetic field
Chassis	1. Lightweight 2. Compact 3. Durable and shatter proof 4. Resistant to deformation
Wheels	1. Lightweight 2. Durable, will not fall apart 3. Able to integrate with the rest of the rover 4. Sufficient degree of friction to manoeuvre
Software	1. Easy to use, with intuitive and logical controls 2. Minimal lag and WiFi network reliability

2 Sensor Implementation

2.1 Name

2.1.1 Planning

The sensor module was designed to achieve the following [3]:

1. Acquiring the raw signal
2. Amplifying it so it could be measured
3. Demodulating the signal such that it can be interpreted by the Arduino
4. Decoding the UART package to display the name

2.1.2 Antenna Design

Electromagnetic induction is the principle behind the transmission and detection of radio waves. The antenna works because the radio waves are incident on the surface area of the coil which induces a current in the wire.

Equations that govern the behaviour of the antenna include:

$$\varepsilon = -N\frac{d\Phi}{dt} \tag{1}$$

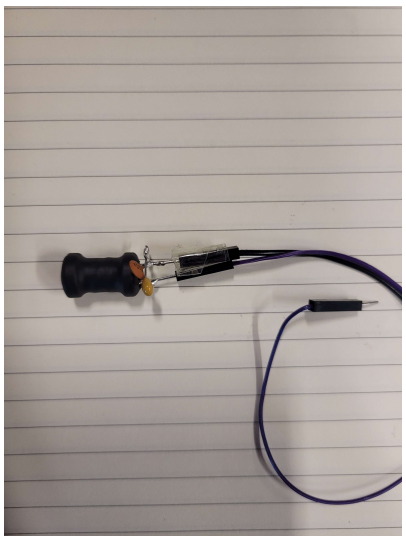
$$\Phi = BA \tag{2}$$

To acquire the raw signal two different designs were considered:

1. **Coiling a large inductive loop**, as shown in Figure 2
   Pros: the coil worked with a decent range, shown in Figure 3
   Cons: Upon testing, the coil of wire was too heavy, as shown in Table 3
2. **Using a large size and value inductor**, as shown in Figure 2
   Pros: Lightweight and small, easy to manoeuvre, good range, as shown in Figure 3
   Cons: directional, and while the inductive loop is too, the smaller surface area is less forgiving

Table 3: Physical properties of the inductor and coil

	Inductance	Weight	Diameter
Inductor	10 mH	62 g	8.4 cm
Coil	55 µH	207 g	1.2 cm



(a) Inductor as an antenna

(b) Inductive loop as an antenna

Figure 2: The two antennae

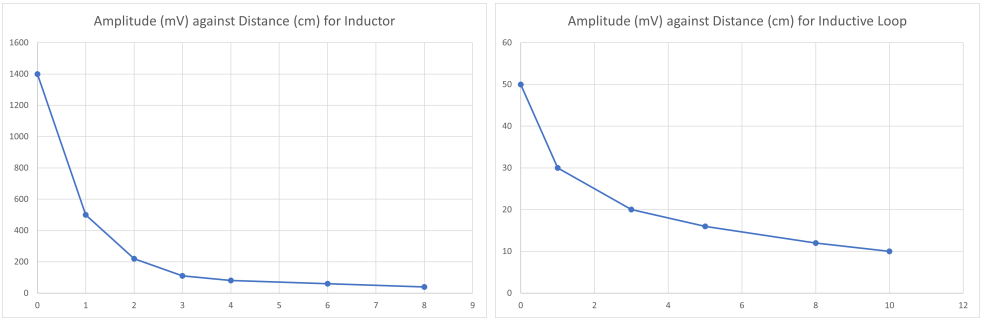
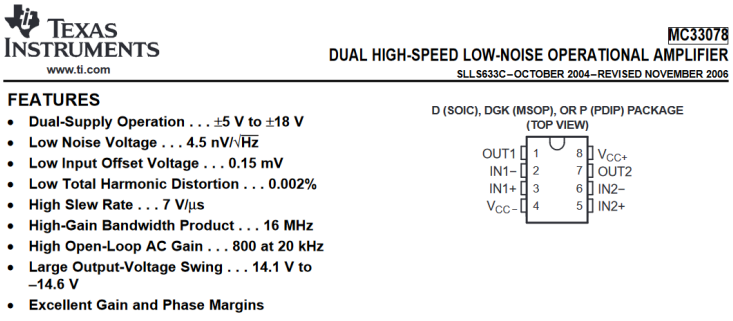


Figure 3: Plotting Antenna performance of (a) Inductor and (b) Coil of wire

We settled on the inductor alone due to it being easy to tune, cheap, reliable and lightweight.

2.1.3 Circuit Design - Amplification

Cascading two inverting amplifier stages proved to be sufficient in amplifying the raw signal. The benefits of this set up included the choice for increased bandwidth and increased gain.



TEXAS INSTRUMENTS
www.ti.com

MC33078
DUAL HIGH-SPEED LOW-NOISE OPERATIONAL AMPLIFIER
SLLS633C – OCTOBER 2004 – REVISED NOVEMBER 2006

FEATURES

- Dual-Supply Operation . . . ±5 V to ±18 V
- Low Noise Voltage . . . 4.5 nV/√Hz
- Low Input Offset Voltage . . . 0.15 mV
- Low Total Harmonic Distortion . . . 0.002%
- High Slew Rate . . . 7 V/µs
- High-Gain Bandwidth Product . . . 16 MHz
- High Open-Loop AC Gain . . . 800 at 20 kHz
- Large Output-Voltage Swing . . . 14.1 V to −14.6 V
- Excellent Gain and Phase Margins

D (SOIC), DGK (MSOP), OR P (PDIP) PACKAGE
(TOP VIEW)

Figure 4: Key Features of the MC33078 [4]

We chose to use the MC33078, brief specification in Figure 4 operational amplifier due to:

- High GBP, which is important as the carrier frequency is high at 61 kHz
- High slew rate , which is important as the signal was to be amplified to 6-8 Vpp. Therefore, given a carrier frequency of 61 kHz, the minimum slew rate needed is 0.5 V/ µs

However, the operational range as shown in Figure 4, states an operational range supply voltage of ±5 V to ±16 V. As the Arduino board is incapable of supplying -5 V DC we used the ICL76605CPA inverter [5], to solve this problem.

This also reduced distortion caused by the 0.7 V drop across the diode as now our signal had a potential range of 10 V instead of 5 V.

2.1.4 Circuit Design - Demodulation

To demodulate the amplified signal three methods were considered:

1. **Diode alone**
2. **Basic envelope detector**
3. **Utilising an averaging circuit**

Below, Figures 5, 6 and 7 show a direct comparison of the first two circuits and, in opposition to what the simulation suggests, the diode alone demodulates the signal as shown in Figure 7. It was noticed that:

- Either the diode's non ideal behaviour or another external factor aided in demodulation
- The envelope detector increases the amplitude of the high frequency oscillations.

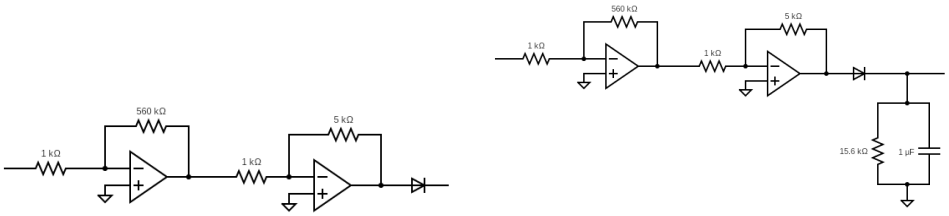


Figure 5: (a) Diode alone (b) Envelope Detector inverted

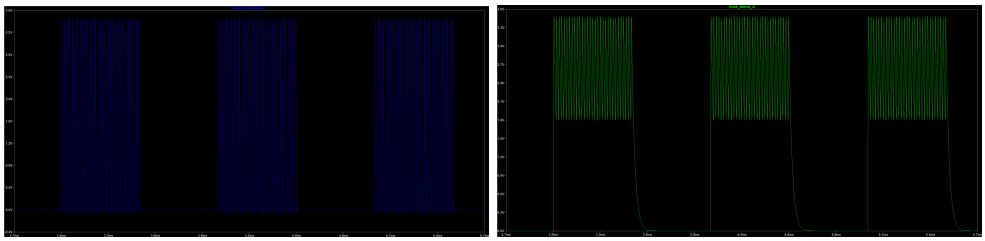


Figure 6: LTSpice Simulation of (a) Diode alone (b) Envelope Detector

Figure 7: Oscilloscope waveform of (a) Diode alone (b) Envelope detector

Through consultations, it was discovered that the oscilloscope probes had an input impedance of 1 MΩ and capacitance of 16 pF [6]. The cutoff frequency caused by this was:

$$\frac{1}{2\pi(1M\Omega)(16pF)} = 10kHz \tag{3}$$

The ideal cutoff frequency of an envelope detector is given by $\sqrt{f_c f_m} = 6049$ kHz, where $f_c = 61$ kHz and $f_m = 600$ Hz. The ideal cutoff frequency is in the same order of frequency as the oscilloscope probes which caused the response in Figure 7.

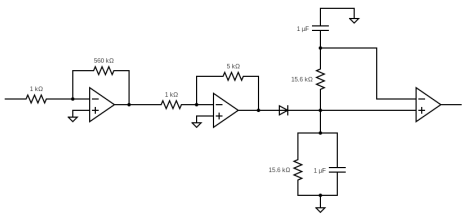


Figure 8: Averaging Circuit Schematic

Near the max range of the antenna, where the raw signal amplitude was low, the high frequency oscillations became a bigger proportion of the amplitude. If too large, it could affect whether the Arduino reads a 0 or a 1.
The purpose of the averaging circuit shown in 8 was to allow for a variable reference voltage from which a comparator can be used with the demodulated signal.

This circuit features two RC networks, each serving a distinct purpose and characterised by different time constants.
One acts as a standard envelope detector, filtering out the high frequency carrier oscillations. The equation for the optimised time constant, RC, for the envelope detector as shown in equation 3, is given below:

$$\frac{1}{2\pi RC} = \sqrt{f_c f_m} \tag{4}$$

The second RC network is the averaging circuit. The circuit, ideally, removes all frequencies apart from DC, which is the average. We experimented with the time constant and settled on around 100 ms as it provided a clean signal as shown in figure 9.

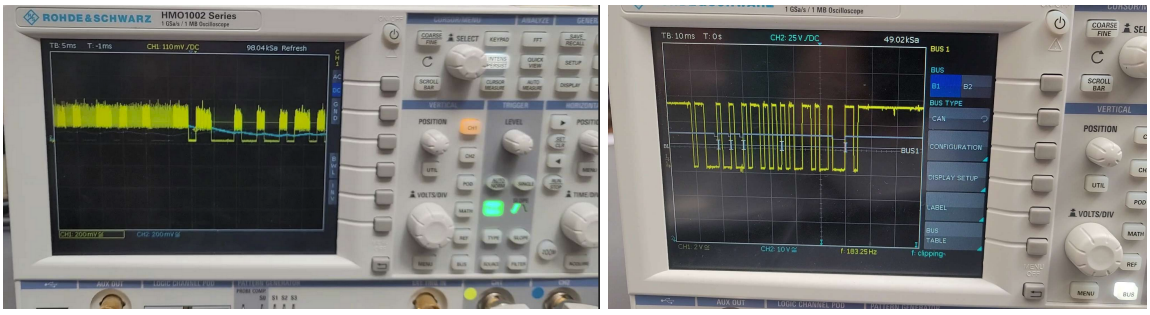


Figure 9: (a) Output of envelope detector and averaging circuit (b) Ouput of comparator

We decided to use the diode alone as it performed sufficiently well and used fewer components.

For future improvements, a new version of the PCB with the averaging circuit included would not only increase signal range, but the clarity of the signal at a given range.

2.1.5 Code

A function was written to return the name of the alien when called, while eliminating errors and inconsistencies in the reading:

- Before returning a string, the function reads the name until it reads the same name 5 times.
- Clearing the serial buffer before every call. This is necessary as, it ensure the serial input being read is from the time the function is called.

Code can be found in the Appendix [2].

2.2 Age

2.2.1 Planning

Aliens have an age pulse that can be detected optically. These pulses have the following characteristics:

1. Aliens emit infrared radiation at a wavelength of 950 nm
2. The frequency range of the pulses is 135–1000 Hz, with a pulse width of 50 µs
3. Aliens spend the first century of their lives underground, so the minimum period would be 1 ms. A pulse period of 1 ms corresponds to 1 century.

The sensor module was designed to achieve the following:

1. **Amplification** of IR pulses for a minimum detection range of 5 cm. This would provide some flexibility in mounting the sensor on the rover.
2. **Filter** out noise, in particular, that which is produced by the rectification effect of the 50 Hz AC source
3. **Preserve** signal waveform shape and characteristics for accurate and consistent measurements

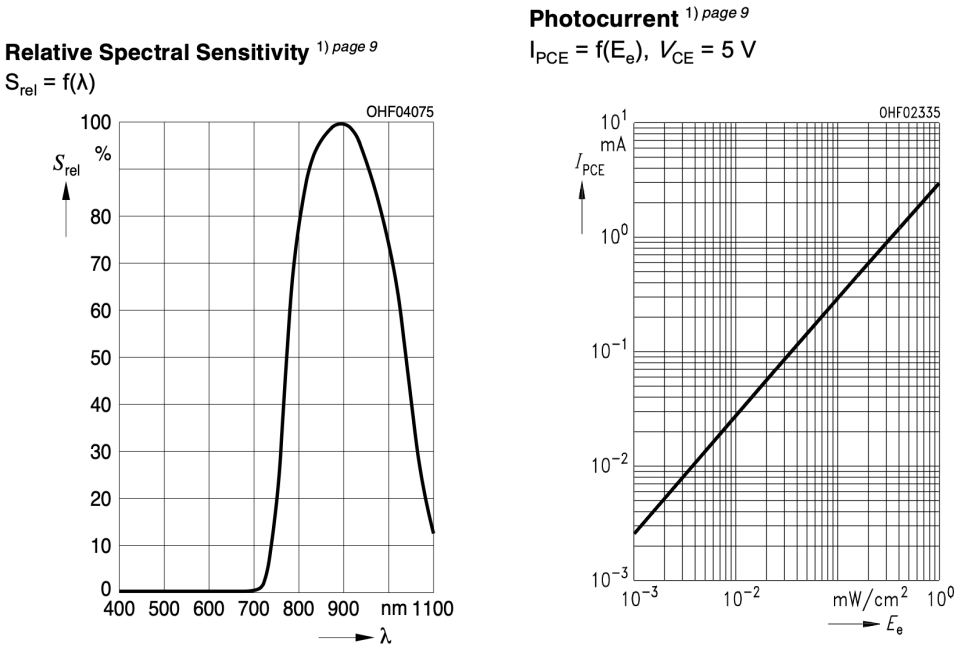
2.2.2 Sensor Selection

The following types of infrared sensors were evaluated:

1. **Photo-diode**: high photo-current and response time, but limited sensitivity in certain IR wavelengths [7]
2. **Photo-transistor**: similar to photo-diodes, but with higher gain. Infrared photo-transistors (with a dark lens that reduces sensitivity to visible light) were also considered [8]
3. **IR Receiver**: costly, lower spatial resolution [9]
4. **Thermopiles**: detects IR radiation via temperature differences, slow response time and low spatial resolution [10]

The existing SFH300-3/4 photo-transistors [11] that are available in the lab were selected for initial testing. These photo-transistor were found to be suitable for the following reasons:

1. **High relative photo-sensitivity** at 950 nm (Figure 10a). Given the lack of ambient light at frequencies higher than 100 Hz in the Figure 12b, a high pass filter was believed to be sufficient for removing noise.
2. **Sufficiently high photo-current** of approximately 0.01 mA in the light conditions in the lab (Figure 10b)



(a) Relative spectral sensitivity (b) Photo-current

Figure 10: SFH300-3/4 photo-transistor datasheet

2.2.3 Circuit Design

To convert the photo-current from the photo-transistor to an output voltage at the emitter, a 10 kΩ load resistor in series was used. In the light conditions of the lab, the 10 kΩ load resistor generated a sufficiently high output voltage for measurement and further amplification.

To remove low frequency noise, a second order high pass filter (Figure 11) was designed with a corner frequency of approximately 100 Hz and a -40 dB stop-band at 10 Hz. High frequency noise was deemed to be sufficiently small, such that it could be ignored (Figure 12b).

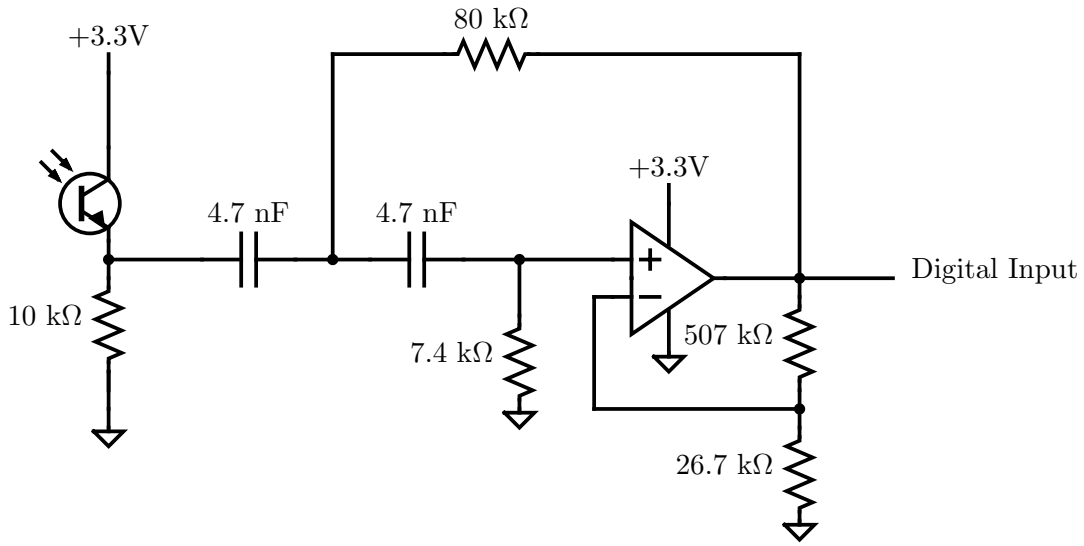
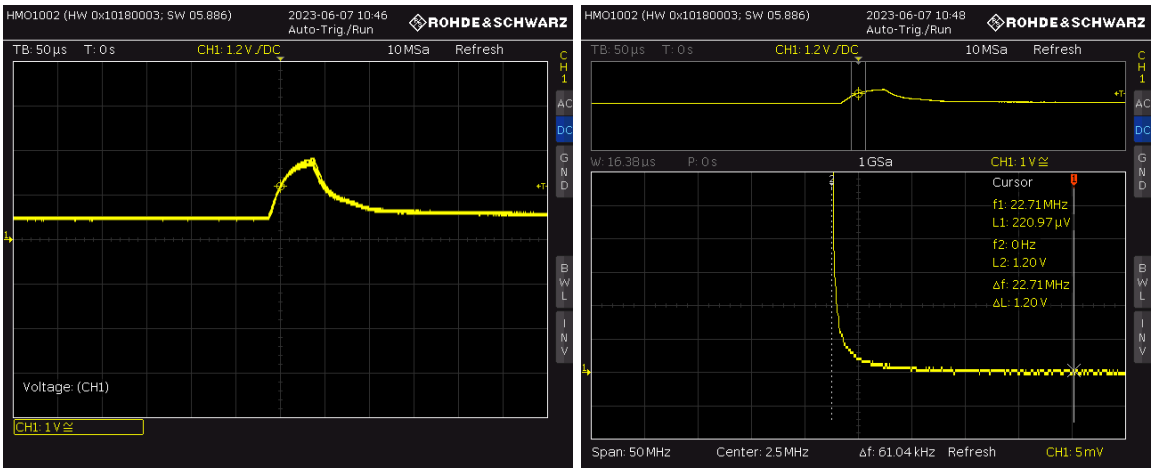


Figure 11: IR circuit

In initial tests, the `analogRead` function on the Metro Express M0 was too slow to detect the 50 µs pulses. Hence, the analog output of the filter was converted to a digital signal for faster detection. This was achieved by amplifying the signal to the rails of the LT1366 operational

amplifier, which were set to +3.3V and ground. Digital inputs on the Metro MCU board are 0 V for logic low and 3.3 V for logic high. Thus, the amplified signal can be effectively used for digital processing. To amplify the signal to the rails, the filter was designed with a gain of 26 dB.



(a) IR pulse (b) Fast Fourier Transform

Figure 12: Signal before amplification

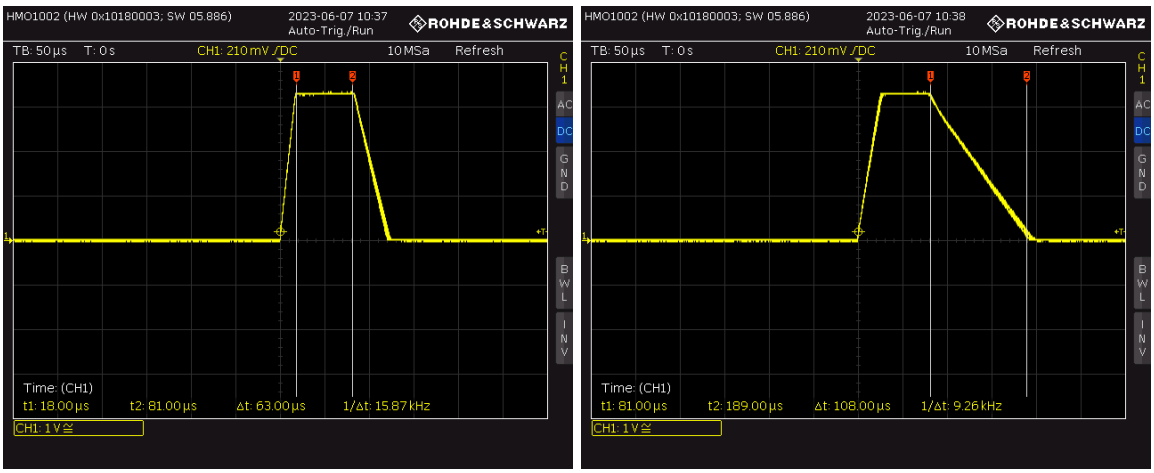
2.2.4 Testing

Figure 13b shows an image of the waveform produced at the output of the IR detection circuit. The square wave input is largely preserved, but the edges of the waveform are smoothed. This is likely due to the rise and fall time of the photo-transistor, as a result of its internal capacitance [12], as well as the low slew rate of the LT1366 operational amplifier [13].

(a) Multiple IR pulses (b) IR pulse waveform

Figure 13: Signal after amplification and filtering

Initially, the `pulseIn` Arduino function was used, which measures the time that a digital signal is HIGH or LOW. However, this resulted in some variation in the age measurements as the slope of the waveform changed with distance (Figure 14). This variation was approximately 50 µs, which is within the acceptable margin of error of 100 µs (1 decade).



(a) IR pulse at distance of 5 cm (b) IR pulse at distance of 15 cm

Figure 14: Length of IR pulse and variation in lag time

2.2.5 Code and Further Testing

To minimise this error, a different method of age measurement was adopted. The `micros` and `digitalRead` functions were used to create an edge detection routine that would measure the time between 2 rising edges. This would be less sensitive to changes in the rise and fall time of the waveform. This brought the error down to < 10 µs (1 year).

Testing was done with the set-up shown in Figure 15. The amplitude of the pulses remained above 2.0 V up to a distance of 25 cm (Figure 16). Given that the threshold for a digital HIGH is approximately 2.0 V [14], detection is possible up to this range.

To determine the consistency of the age measurements, the standard deviation of the time between pulses at each distance was obtained over 200 pulses (Figure 17). Using this method, the age of the alien could be determined consistently and accurately up to a distance of 20 cm. Beyond this distance, the standard deviation became too large for meaningful measurement. The amplitude of some peaks was lower than the HIGH signal threshold, hence those peaks were not detected.

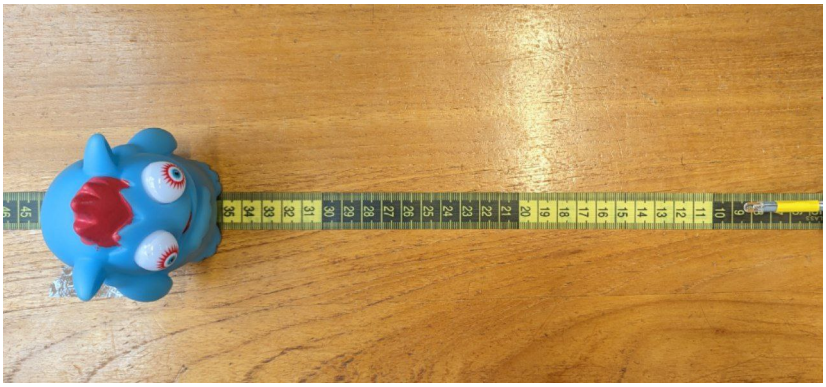


Figure 15: Distance measurement set-up

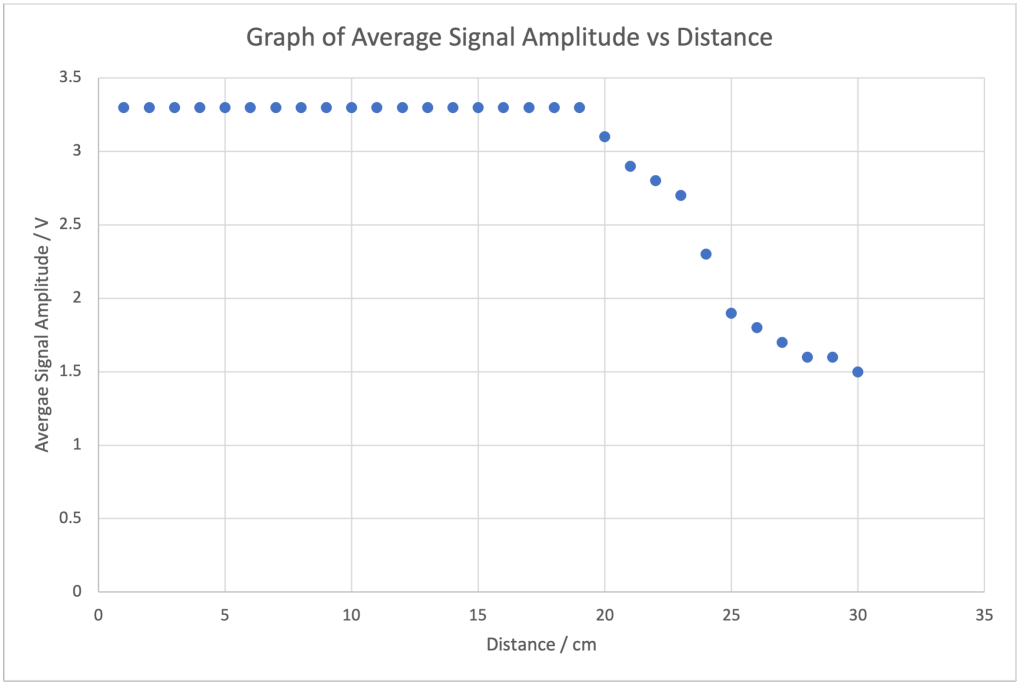


Figure 16: Graph of average signal amplitude vs distance

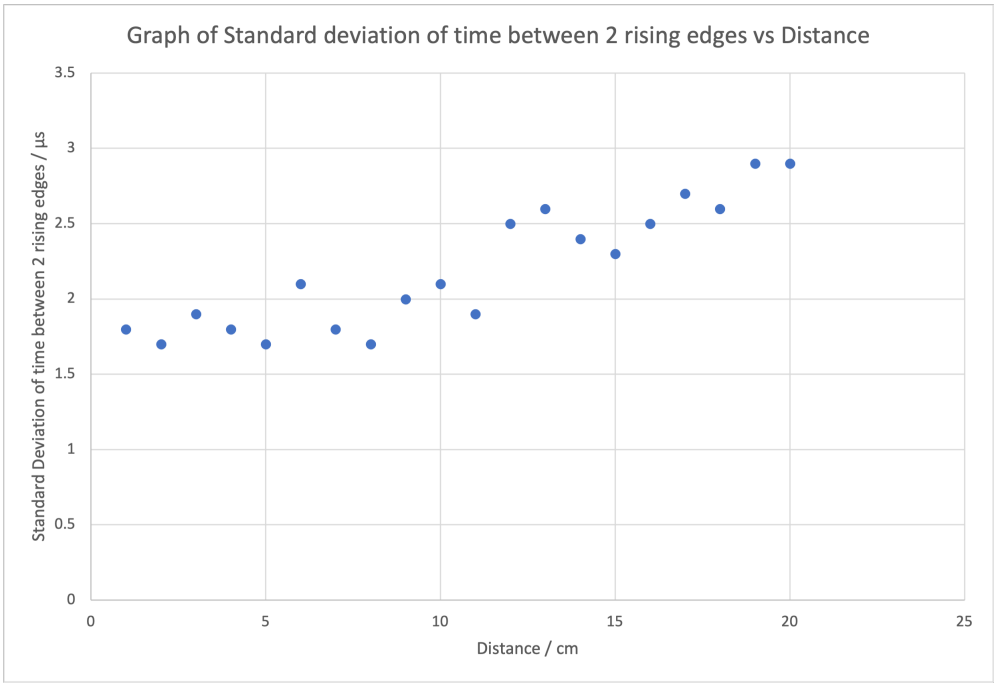


Figure 17: Graph of standard deviation of time between 2 rising edges vs distance

There are some possible areas of improvement. To reduce sensitivity to visible light, an IR-specific photo-transistor could be used [8]. An operational amplifier with a higher slew rate could also be employed to minimise the rise and fall time of the digital IR pulses [15]. However, given the high range, accuracy and consistency of the existing implementation, we considered it to be sufficient in fulfilling the submodule's requirements.

2.3 Magnetic Field

2.3.1 Planning

Aliens have magnets that are placed at highest point possible inside the alien, about 1cm from the top. The magnets have a static magnetic field pointing either UP or DOWN. This sensor module was designed to achieve the following:

1. Magnetic field directions should be detectable from an appropriate distance
2. The environment's magnetic field should be taken into consideration when determining the direction of the alien's magnetic field. There is distortion of magnetic field in different parts of the room [16].

2.3.2 Sensor Selection

We evaluated the following types of magnetic field sensors:

1. **Linear Hall effect sensor**: output voltage increases and decreases from bias based on direction of magnetic field it detects [17].
2. **Digital Hall effect sensor**: Outputs High or Low based on magnetic field direction [17]. It has a high threshold so that the result is not determined by external magnetic fields, so it can't measure small signals [18]. 2 sensors are needed to detect both magnetic field directions.
3. **Magnetometer**: It tends to have higher accuracy and sensitivity than the other sensors [19]. Moreover, it also has 3-axis measurement capability, which means precise direction of magnetic field. However, it has extra functions and complex software that makes it hard to use.
4. **Reed switches**: Circuit will be changed from its initial state ON or OFF, when a specific polarity of magnetic field is applied [20]. It has a high threshold so that the result is not determined by external magnetic fields, so it can't measure small signals. Two sensors are needed to detect both magnetic field directions.

The best options would be either linear hall effect sensor or the magnetometer. Both methods require only one sensor to determine field direction, and allows detection of small signals from a good range. However, linear sensors match our requirements better. We are just required to detect if the magnetic field is upwards or downwards in 2 dimensions, not 3. The magnetometer also comes with other additional functions that we do not need and can be harder to use.

We chose to use the SS495A Honeywell Linear sensor model [21]. This sensor is not affected by temperature over a large range of magnetic flux densities, and a acceptable sensitivity of 3.125 mV/G, shown in Figure 18.

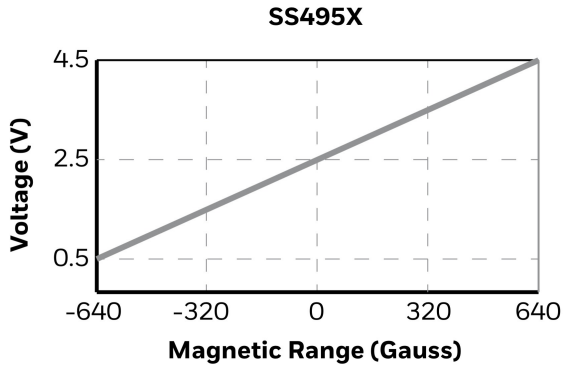


Figure 18: shows the sensor linear over a large range +-640 Gauss, while taking temperature into consideration

2.3.3 Circuit Design

We biased the V+ input of the operational amplifier by using a potentiometer to vary the resistance, so that output/analog input is approximately 1.7 V. This makes the `analogRead` output 500, when there is no magnetic field. This biased the output to the middle of the range of 0 to 1024. The gain of 43 dB is used so that the difference between output of the sensor and V+ is amplified. This increases the sensor range to around 8 cm. A supply voltage of 3.3 V is used for the operational amplifier, as analog input accepts a maximum of 3.3 V. The schematic is shown in Figure 19.

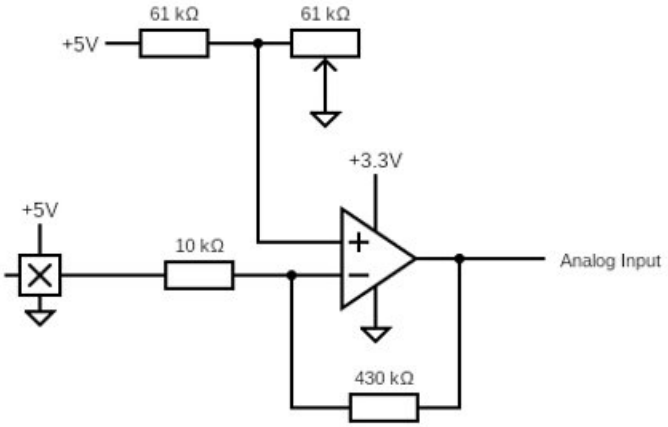


Figure 19: Magnetic field circuit

2.3.4 Code

There are two functions sections to the code:

1. **UP or DOWN**: used to compare a reference value to the magnetic field around the alien. If the difference is greater than the threshold `analogRead` 20, then magnet is present, else prints `NO Magnet`. If the value is less than reference and greater than threshold, then it is magnet field `UP`, else `DOWN`. Threshold is needed to ignore noise.
2. **Environment magnetic field**: due to magnetic field strength varying in the environment, a calibrate function is used to take average values of 20 magnetic field readings. This average value is taken before reaching an alien, and used as reference to determine if the alien is magnetic field UP or DOWN.

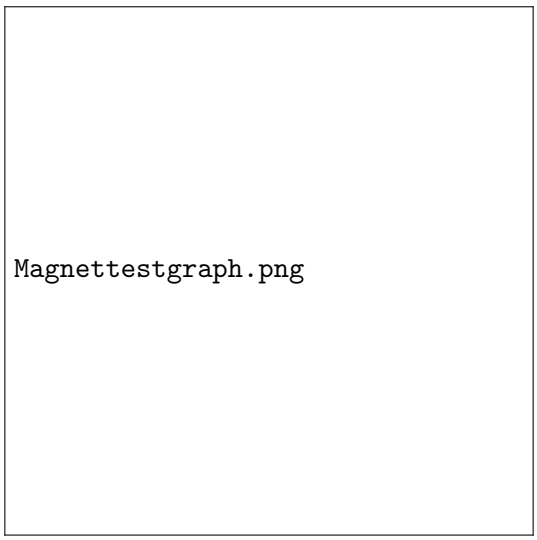


Figure 20: Testing

We recorded data at different distances between the sensor and the magnet. Figure 20 shows that the largest change in `analogRead` is 500, when the magnet is 0-1.5 cm away from the sensor, which represents the strongest magnetic field strength. At 8 cm, the change in `analogRead` is 21. Due to noise from the environment, we set a threshold in the code so that only a change in `analogRead` greater than 20 would suggest there is a magnet present. This threshold would eliminate noise, and ensure the result is determined only by the magnet in the alien.

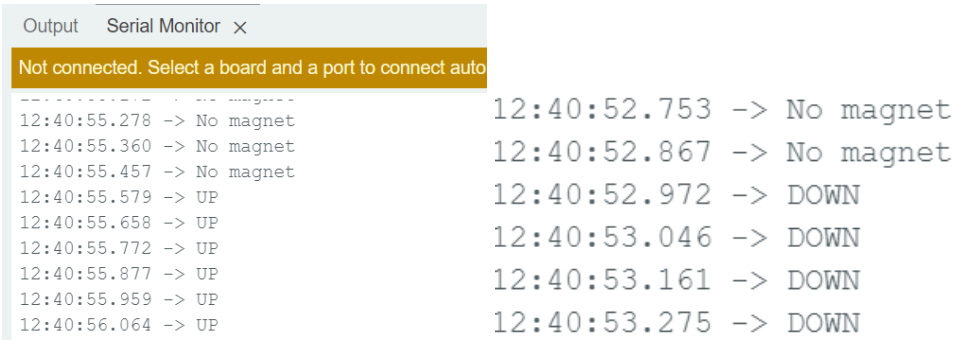
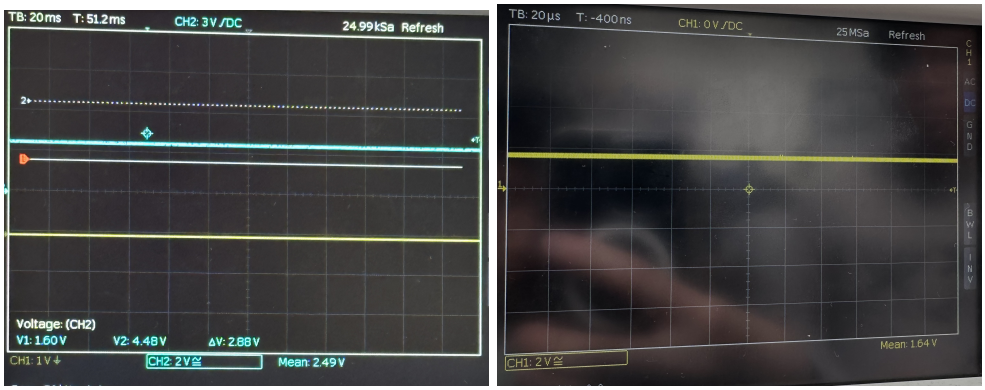


Figure 21: Readings when alien magnetic field is UP and DOWN, and moved towards sensor

2.3.5 Testing and Results

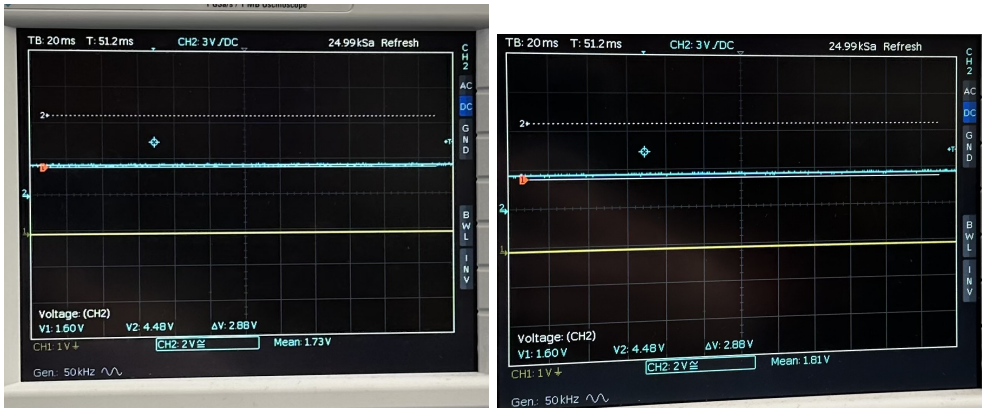
When no magnetic field is present, the sensor reading is shown in Figure 22a, which is not in the middle of 3.3V analog input. The amplified circuit reading is shown in Figure 22b, which is now in the middle of 3.3V, so it is in the middle of the range of `analogRead`.



(a) Sensor bias (b) Amplified circuit bias

Figure 22: No magnetic field present case.

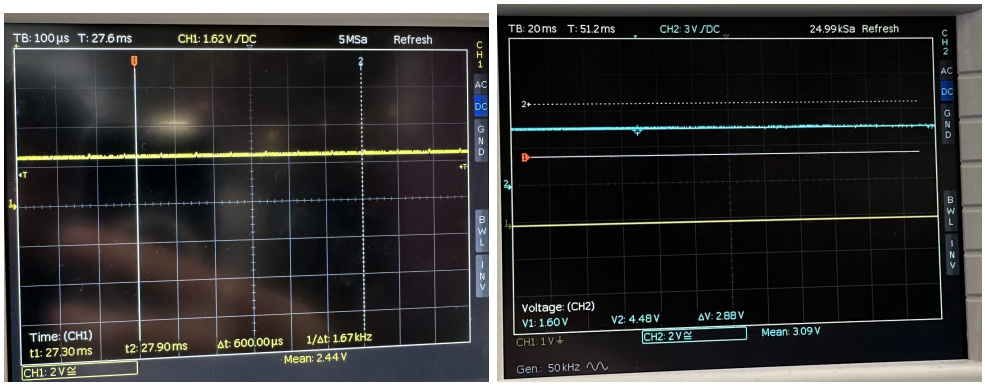
The sensor alone before amplification gives no change in readings, when the measured range is greater than 3 cm. The amplified circuit gives readings at different locations, 8 cm in Figure 23a and 6 cm in Figure 23b, which shows visible change from the bias point.



(a) 8cm reading (b) 6cm reading

Figure 23: Measurements at 6 and 8 cm

At a range of 2 cm, the sensor shows some change from the bias point, but it is a very small change. This is shown in Figure 24a. Figure 24b shows this range with the amplified circuit.



(a) Sensor reading (b) Amplified circuit reading

Figure 24: Readings at a distance of 2 cm

Hence the bias point should be set to 1.6 V, such that it is in the middle of the range of

`analogRead`. The differential amplifier circuit allows for an `analogRead` midpoint bias, as well as an increased range by amplifying any change from the bias point.

2.3.6 Future Improvements and Applications

To further improve magnetic field detection, the magnetometer can be used. With the 3-axis measurement, it can determine the exact direction in 3D, as well as accurate field strength. This sensor is made to take into consideration anomalies and local magnetic field disturbance, which makes the rover ready for more challenging environments [22]. It can be applied in other parts of the rover as well, such as mapping and navigation [23].

3 Rover Design and Construction

3.1 PCB Design

With the EEEBug PCB design in mind, our custom Helionix Control Board PCB provides numerous technical and design improvements.

3.1.1 Motivations

While not required by the design brief, we decided to design our own custom PCB. This choice was supported by the following advantages:

1. **EEEBug improvements**: by having a single circuit board containing both voltage and motor control circuits we avoided the use of multiple EEEBug PCBs as these would be needed to control 3 or 4 wheels with the given custom H-bridges [3].
2. **Compactness and modular design**: our board was designed to be stackable, like most Arduino-based MCU shields are [24]. This means it can be easily manufactured and exchanged for other boards as we improve the sensing circuitry throughout the duration of the project. Furthermore, by stacking the board we avoid the need for additional space around the MCU thus permitting a smaller chassis design as can be seen in Figure 25.

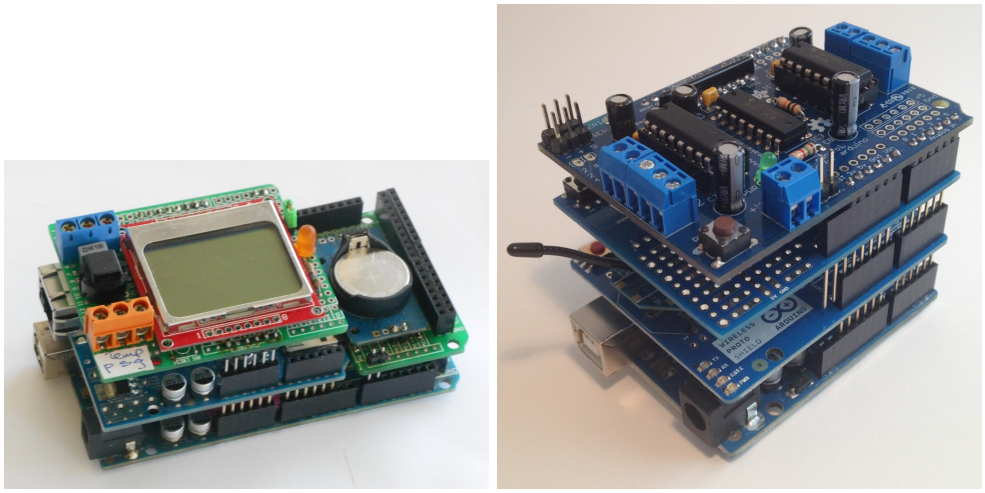


Figure 25: Examples of Arduino-based stackable shields

3. **Interference minimisation**: by laying out and routing components according to the relative sensitivity we can effectively minimise interference due to track lengths and skew [25] [26]. This was particularly important when designing the RF circuitry.
4. **Wider input voltage range**: compared to the EEEBug design we used a LM78M05CT linear voltage regular, see Figure 26, which provides a steady 5V output from a wide range of input voltages [27].

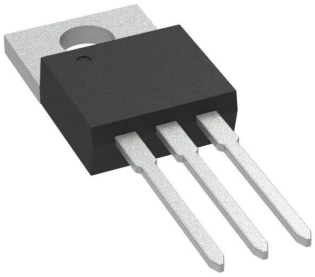


Figure 26: LM78M05CT

While these were all valid reasons to consider custom PCB design there were also other factors to consider:

- **Cost**: custom PCB manufacturing is relatively cheap but it does involve a non-negligible cost. We budgeted for this accordingly, see Appendix.
- **Time**: manufacturing and shipping can take anywhere between 5 days and 10 days so accounting for this was essential during the whole design process.

3.1.2 Design Process

All of our PCB board designs were made with KiCad 7.0 [28] following these steps:

1. **Analysis** of the original EEEBug PCB schematic from the original EEEBug design, see Figure 27.

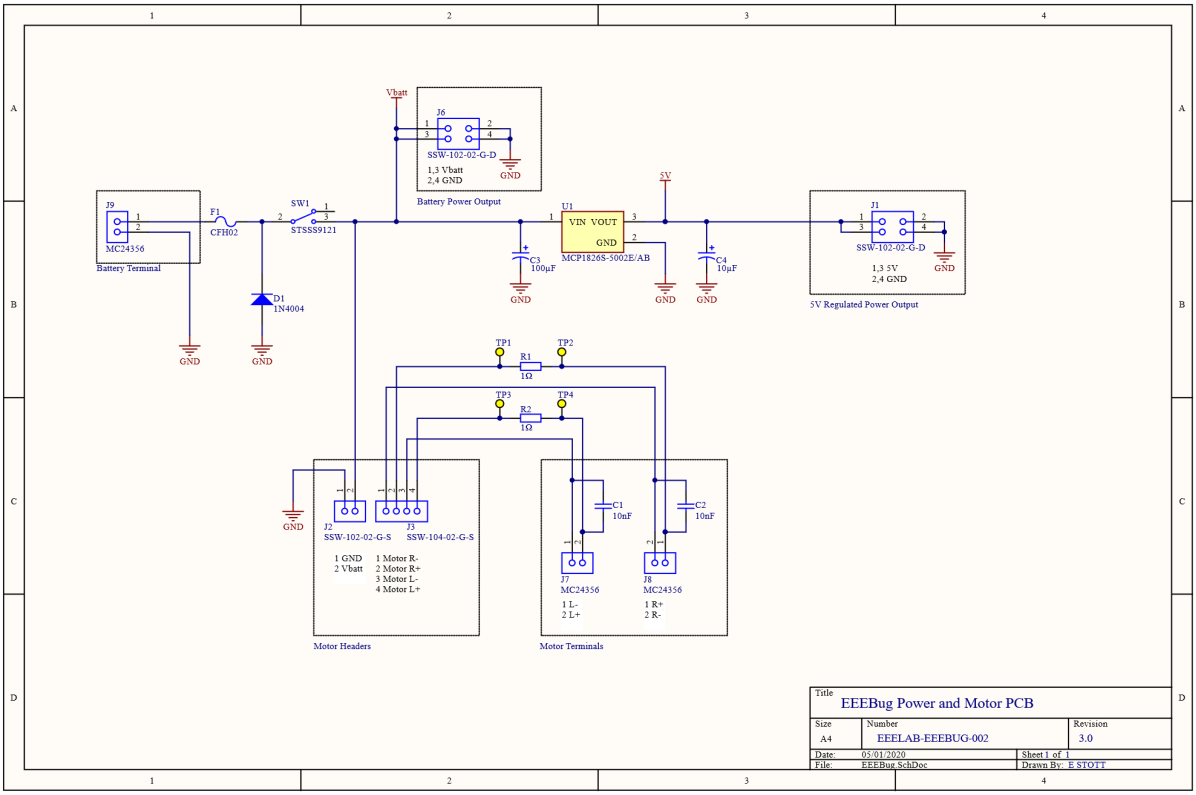


Figure 27: EEEBug Board - Schematic

2. **Modifications**: we wanted to use a 9 V battery as power source so the schematic had to be modified appropriately, see Appendix for the BOM. We also needed the board to support both custom motor drivers [3] so inserts for these had to be included in the design.

3. **Additional circuitry**: sensing circuitry from the other submodules was included, see Section 2

4. **Schematic**: the complete board must designed from an Arduino shield KiCad template schematic freely available from KiCad's Project Templat Selector as you can see from Figure 28

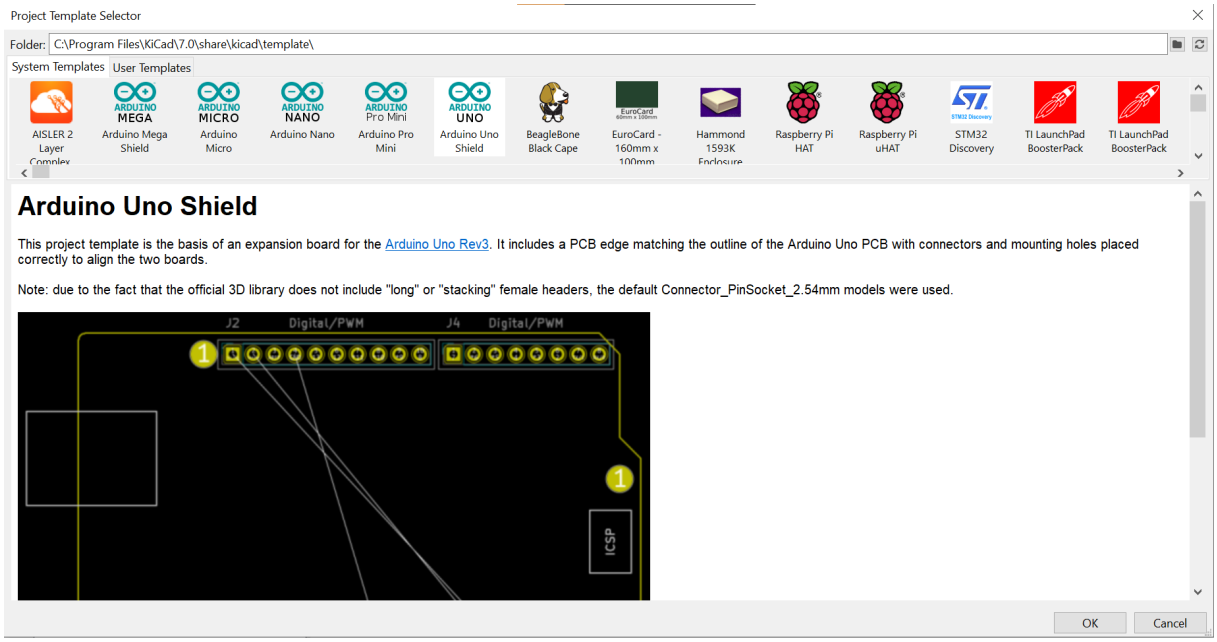


Figure 28: Arduino Uno Shield KiCad Template

5. **Footprinting**: many footprints had to be sourced online as they were not available in the standard KiCad footprint library. See Appendix for a comprehensive list.

6. **Laying out**: the PCB had to be physically layed out, this meant positioning components to make tracks and power routing as simple as possible as well as positioning connectors in useful positions to avoid overlapping wires. We also included test points for multiple voltage levels to be used during the testing process.

7. **Routing**: components had to be connected correctly and tracks and vias had to be the correct size to avoid ground loops, solder bridges and other common issues [29].

8. **Manufacturing and soldering**: all of our boards were designed as 4 layer boards for quick access to ground and 5 V power planes. The board also produces 3.3 V through the Metroexpress MCU and 9 V directly from the battery. Orders were put in at JLC PCB [30] and soldering was done in labs once delivered.

9. **Testing** was necessary to make sure all connections were sound and no short-circuits occurred. This was also a useful diagnostic tool to design the final version of the control board.

3.1.3 Prototype Version

The first PCB design we came up with consisted of custom voltage and motor control circuits, first-version sensing circuits, test points, mounting holes and Arduino header connectors. The schematic is shown in Figure 29.

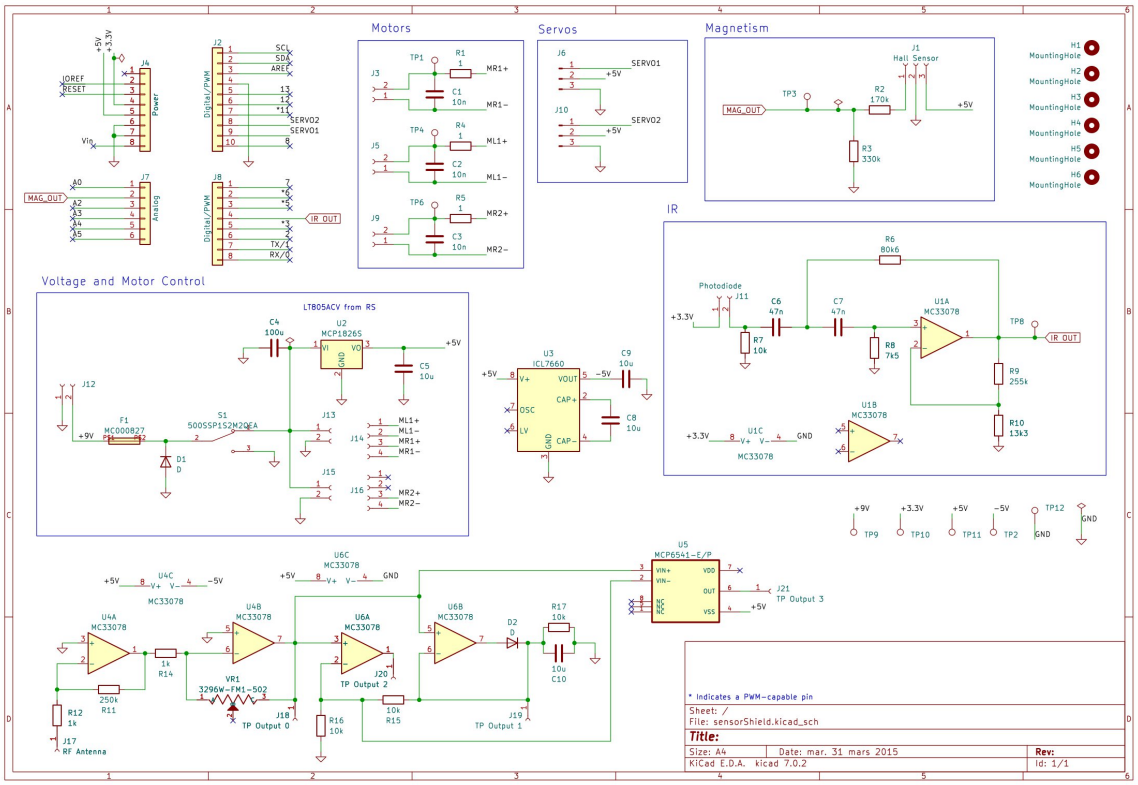


Figure 29: Prototype Board - Schematic

The PCB layout and routing is shown in Figure 30, and the 3D rendered views of the board are shown in Figure 31.

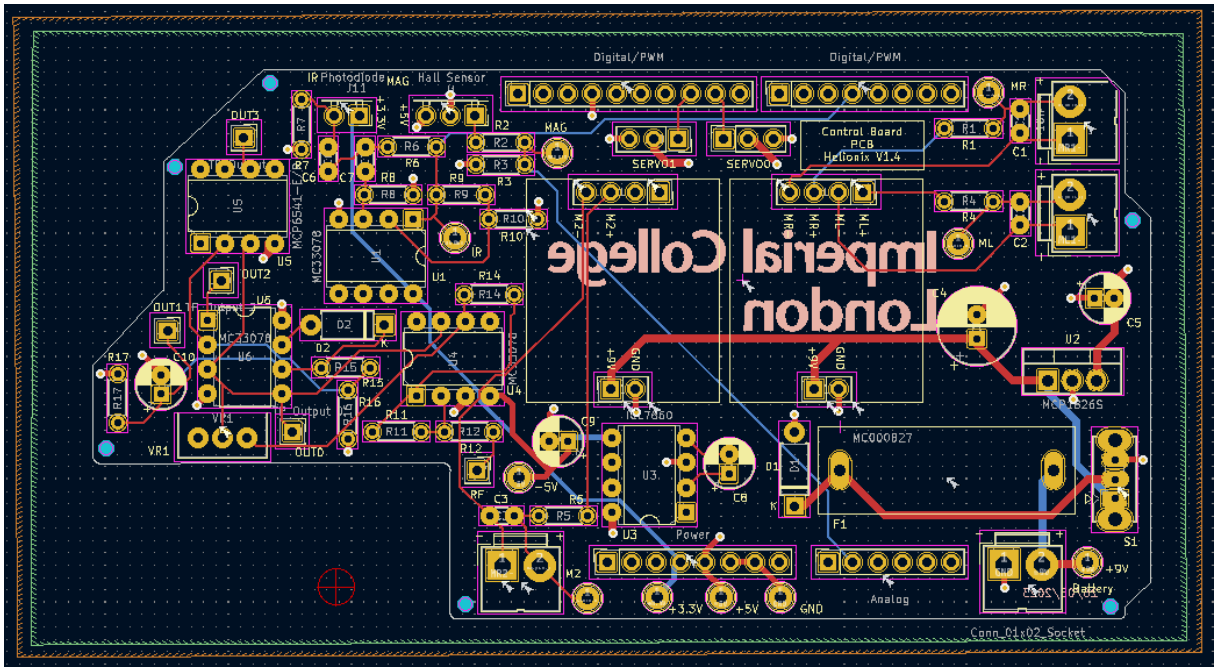
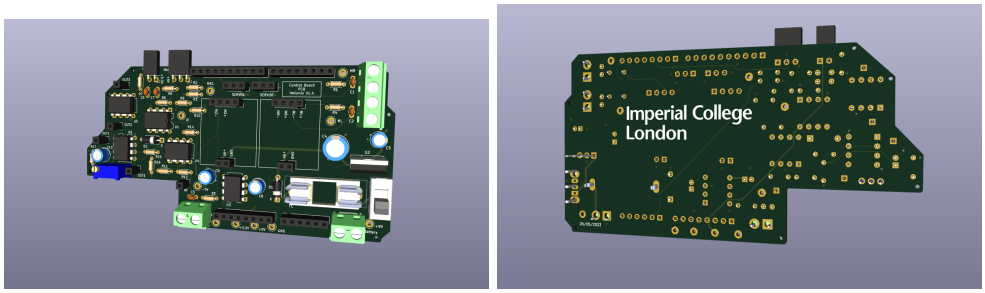


Figure 30: Prototype Board - PCB



(a) Front side of prototype board (b) Back side of prototype board

Figure 31: 3D Views

The multi-faceted outline that can be seen in the 3D rendered views was substituted during the manufacturing process with a simple rectangular outline to avoid additional costs. The unsoldered and soldered PCBs are shown in Figure 32.

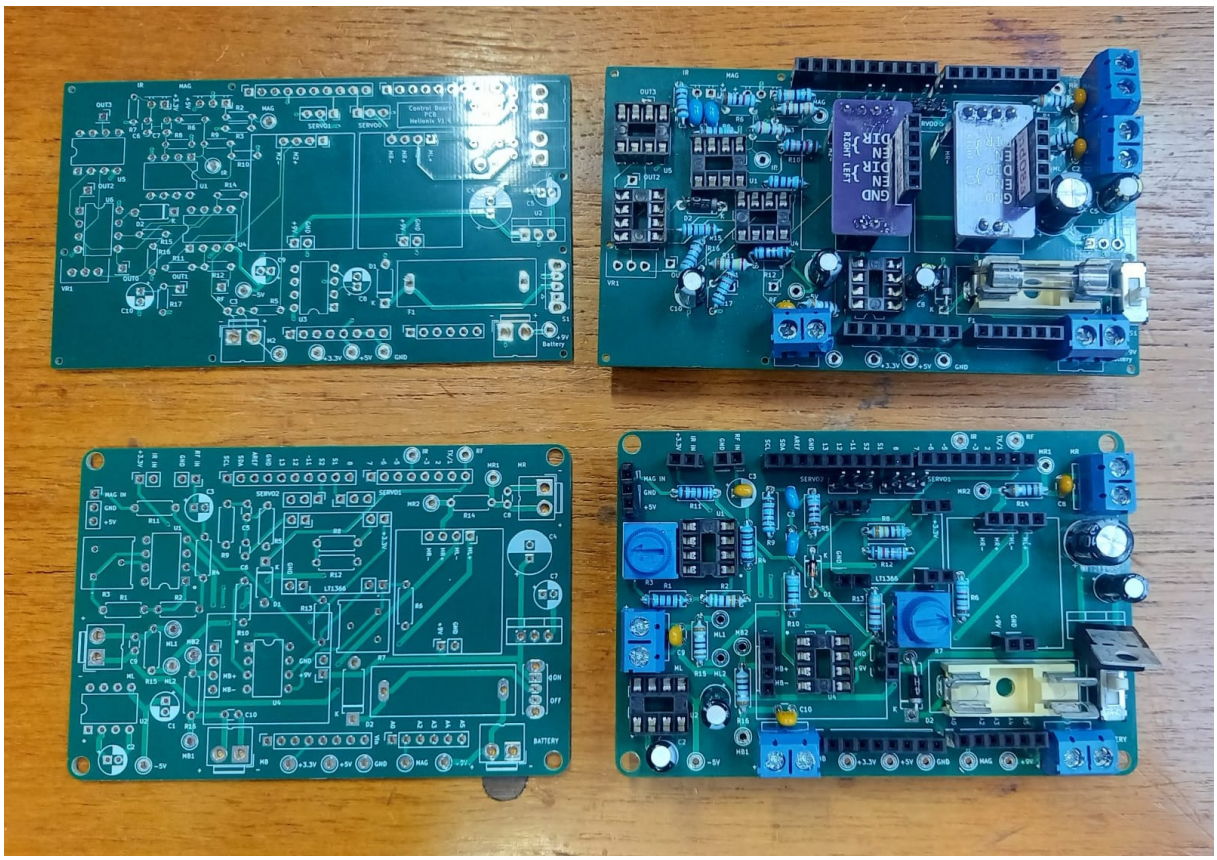


Figure 32: Prototype Board - PCB

Testing showed various areas of improvement which were addressed in the final version of the control board.

3.1.4 Final Version

For the final version, the sensing circuitry was updated to its current, most-functional version, see Section 2. Alongside this major improvements were made to the board:

- **Removed the switch's third pin connection to ground**, this was shorting the battery and blowing the fuse in the prototype version.

- **Changed resistor package size**, from a smaller and more difficult to solder size to a larger one.
- **Included inserts for the LT1366 operational amplifier**, as well as an optional DIL-8 socket to exchange this operational amplifier for any other in case of future improvements. This operational amplifier is shared between the IR and magnetic sensors circuitry.
- **Increased track and via sizes** to 0.6 mm and 1.0 mm tracks and 0.8 mm/0.4 mm vias to avoid overly thin tracks and to increase thermal relief and reduce resistance [31].
- **Shortened the RF routing**, to avoid interference before signal processing it was crucial to shorten the tracks between the RF antenna connection and the closest components [26] [25].
- **Components layout**, such that the space for the inserts can be used and is not wasted. Connectors were also moved to correspond to the physical mounting of motors and sensors and were given more space to include annotations regarding orientation and labelling.
- **Servo connector**, direct male pin connectors were added to the board for easy prototyping of up to 2 servos.
- **Rounded corners and bigger mounting holes**, for overall structural improvements, see Figure 34
- **Annotations and pin labelling**, such that the board can easily be understood by other users without having to refer to this report. Furthermore, the schematic was also annotated in a more exhaustive manner, see Figure 33

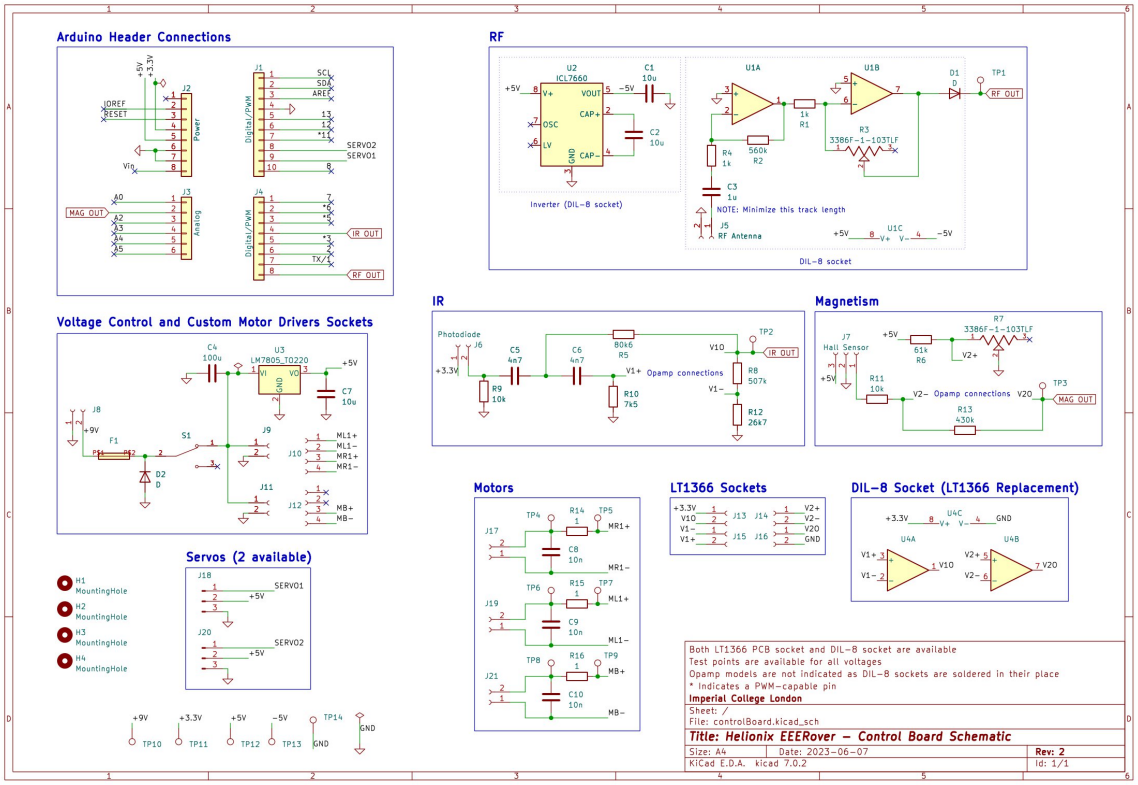


Figure 33: Final Board - Schematic

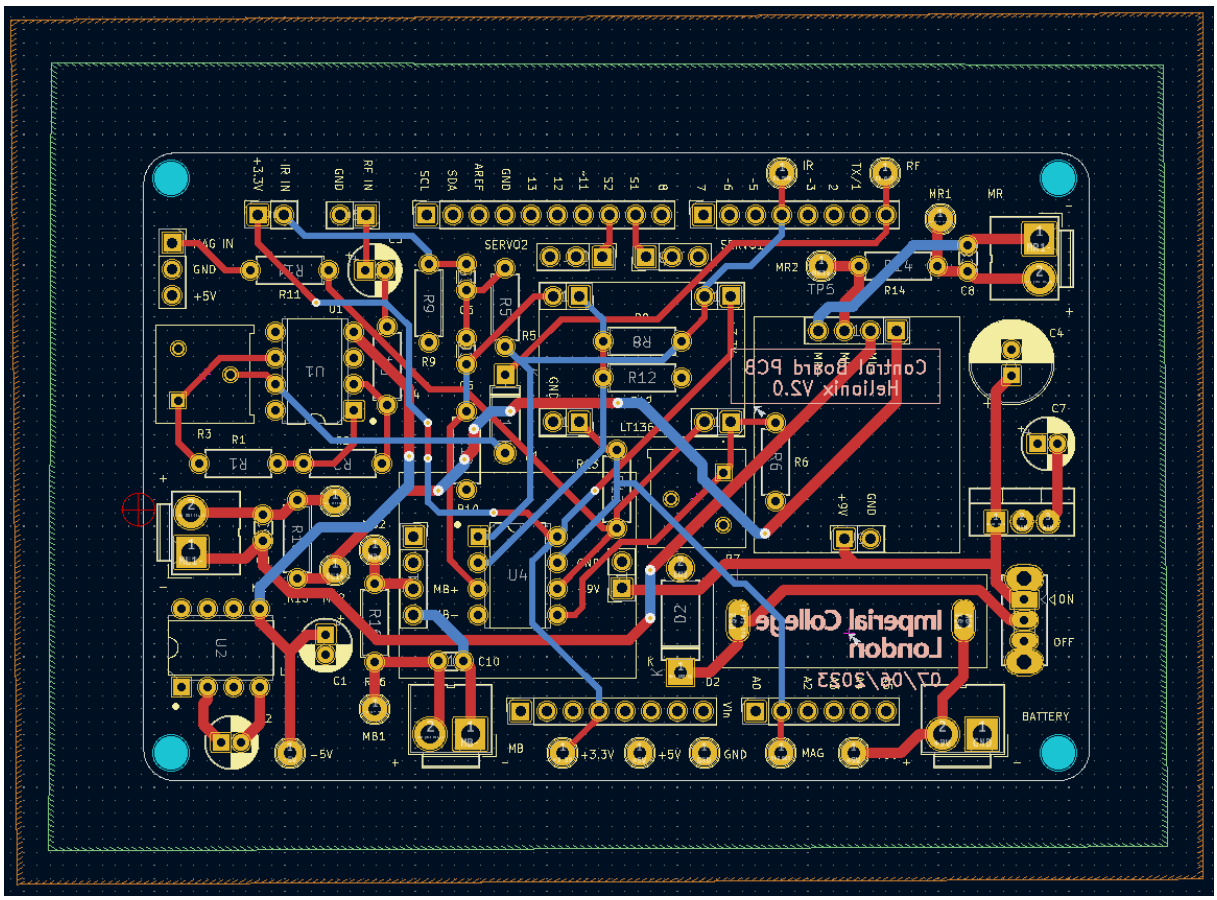
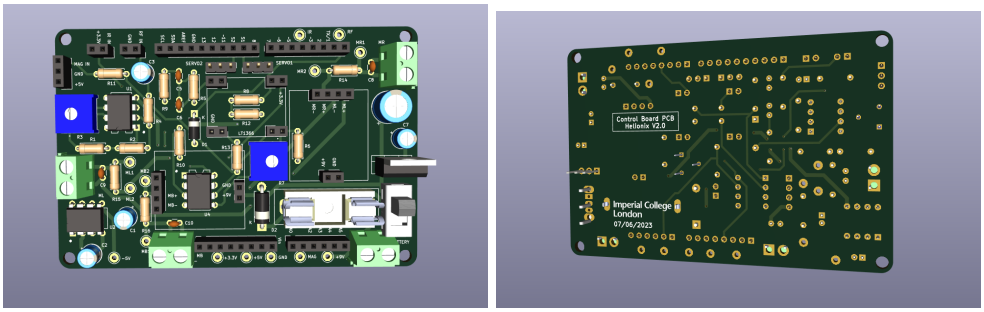


Figure 34: Final Board - PCB

The 3D rendered views of the board are shown in Figure 31. The unsoldered and soldered PCBs are shown in Figure 32.



(a) Front side of final board (b) Back side of final board

Figure 35: 3D Views

3.2 Chassis

Chassis and wheel design were parallel as most design decisions influenced both submodules, they are presented in this order for no particular reason.

3.2.1 Planning

We defined 6 key areas of our chassis design:

- **Weight**, Due to the existence of the shy alien, and to obtain high manoeuvrability, we kept our final design as lightweight as possible.
- **Surface Area**, the rover should be as small as possible so that collisions with other rovers are minimised and the rover is able to reach any hard to get to aliens. Simultaneously we ensured that we had enough room to fit all circuits, the power supply and other hardware onboard.
- **Durability**, the rover should not to fall apart / get damaged at any time to avoid delays, extra prototyping cost and failure in the final demo.
- **Stiffness**, while under stresses the chassis could flex, which would introduce errors in movement such as sliding as well as compromise the positioning of the sensors if a fixed sensor housing option was implemented.
- **Adaptability**, if further ideas and improvements occur after our chassis had been completed it should be easy to integrate new hardware onto it.
- **Integration**, chassis must be compatible with all other components in our design, there must be a cohesive design language between the wheels, circuit boards and other hardware attached to it.

3.2.2 Initial Design Choices

1. **Structure**, we decided to use a triangular base as this only requires three Omni wheels rather than a four-wheel drive which has a higher power consumption or a two-wheel drive which requires more precise motor control.Additionally, chassis surface area is reduced compared to any other rectangular shape.
2. **Modular design**, as prototyping was essential to integrating the submodules we chose to make all mounts and attachments as modular as possible to be quickly exchanged or improved.
3. **3D print**, for fast and inexpensive iterations we decided to use FDM techniques such as 3D printing for all our initial prototypes until we were satisfied with a final design, to be laser cut for structural integrity.

3.2.3 Prototyping and Testing

After analysing our planning the first chassis prototype was designed in Fusion360:

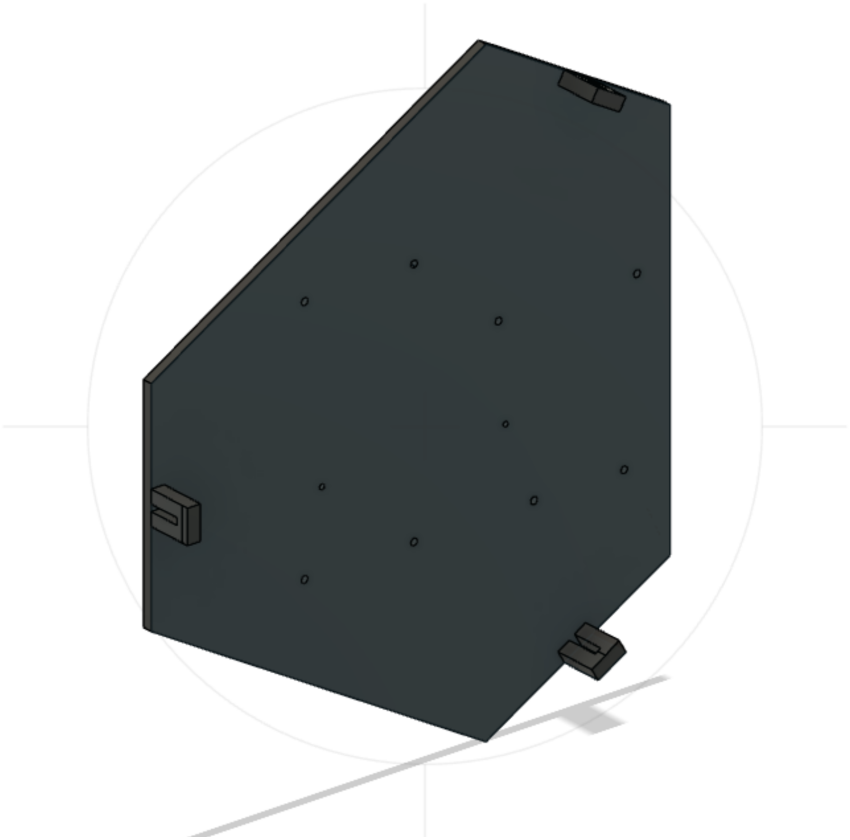


Figure 36: Chassis V0 in Fusion

In Figure 36, small rectangular blocks centred on each edge of the cut-off triangle act as motor mounts, allowing a screw to be drilled through them. Holes to allow a breadboard, Adafruit and motor drivers to attach had also been implemented.

We then proceeded to print and assemble the V0 chassis as seen in Figure 37.

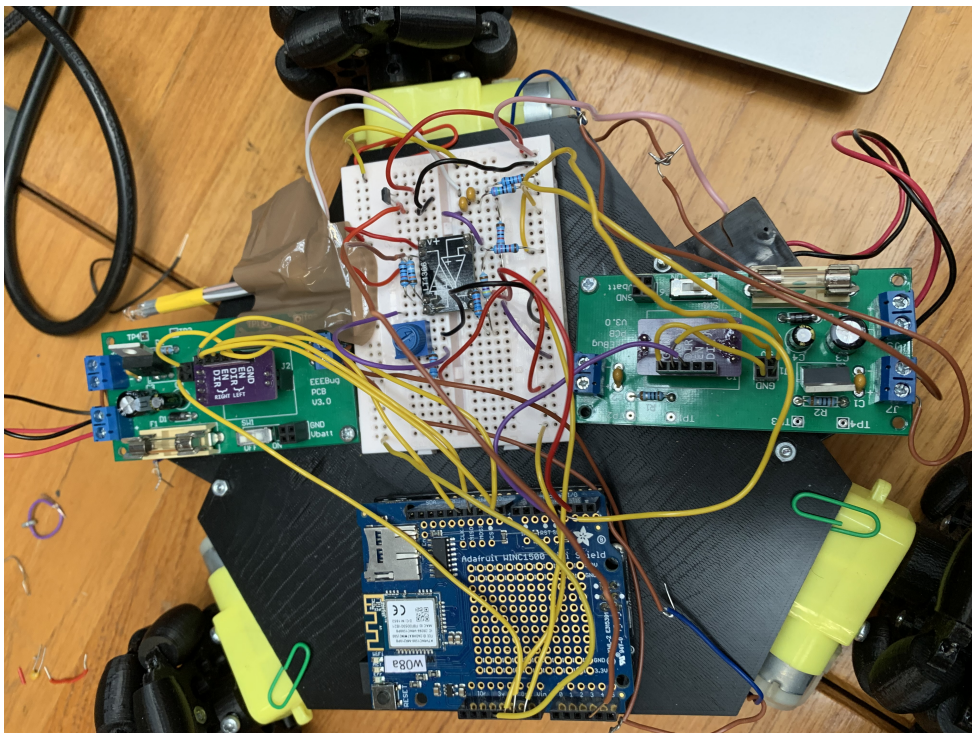


Figure 37: Chassis V0 Printed

After printing and assembly the V0 chassis was tested and the following was learnt:

- **Shape**, Confirmed that the corner cut triangular shape is applicable to an Omni wheel rover design.
- **Motor wobble**, when driving it was observed that motors attached to the mounts would slide upwards as the motors were unable to be properly fastened to them. This could lead to severe loss of control and possible collisions.
- **Hard to modify**, During testing it became apparent that adding any extra hardware to our chassis would be very difficult to do, and drilling any new holes was difficult, inaccurate and irreversible.
- **Oversized**, the initial prototype had a significant excess of surface area for our needs, this lead to chassis sagging, wasted space and decreased manoeuvrability.

Building upon what we learned in our initial round of testing we iterated at speed creating a V1 and V2 chassis designs in Fusion360:

(a) Chassis V1 render (b) Chassis V2 render

Figure 38: V1 and V2 Chassis

These new designs implemented the following features:

- **Compactness**, V2 as seen in Figure 38a greatly reduced surface area, having been optimised to more accurately reflect spatial needs.
- **Stability**, motor mounts were moved further under chassis shown in Figures 38a and 38b mounts were then redesigned so that motors could be properly fastened, eliminating motor sliding therefore making the rover more stable.
- **Attachments**, at the time or these designs being created we had our sensors implemented through being attached to a servo arm with 2 degrees of freedom, mounting capability was added to the rover for this.

3.2.4 Final structural design

Taking in all that was learnt and building upon it a final chassis was designed:

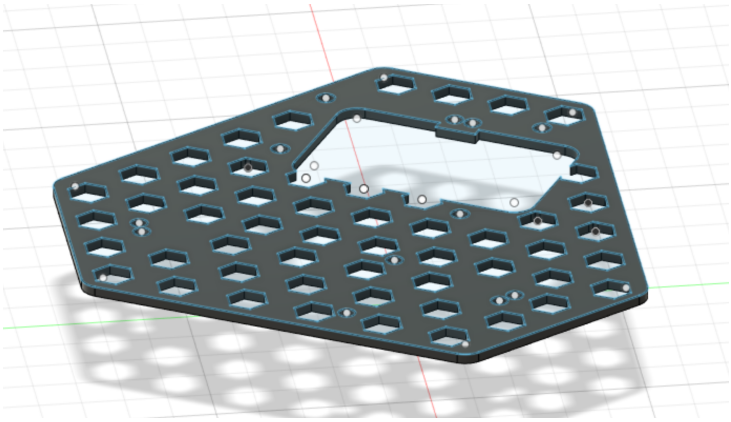


Figure 39: Final Chassis in fusion

Figure 39 contains a mounting for the circuit board stack, holes to allow motor mounts to attach and hexagonal cutouts to reduce weight and allow easy mounting of further external attachments.

Building upon all lessons learned in the prototyping process the final design made further improvements in:

Weight, Through making the design have hexagonal cutouts seen in Figure 39 overall weight was significantly reduced.

Adaptability, through the hexagonal cutouts any future sub modules could be easily designed to attach to the chassis. The motor mounts were also separated from the main chassis body allowing to iterate upon their design separately without a complete chassis redesign.
These design choices would eliminate future time waste and expenses.

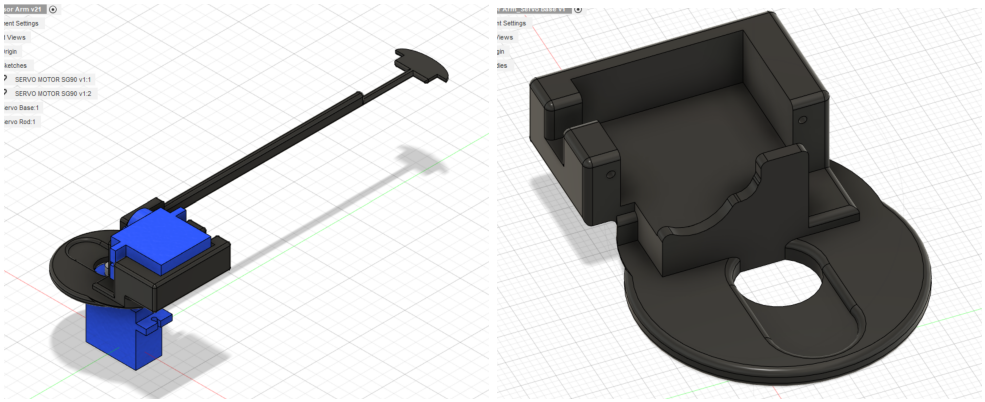
3.2.5 Mounts and attachments

Throughout the process of designing the chassis many separate components were removed or implemented and improved. The following is a comprehensive list of the design process of the individual chassis components:

Sensor arm

Our initial design had all our sensors attached to the end of an arm which was given two degrees of freedom through the combination of two servos.
Figure 40a shows the full sensor arm module where the lower servo would be slotted into a

cutout in the chassis, the arm is directly attached to a servo and the base in Figure 40b is likewise attached to a second servo through its circular cutout.



(a) Sensor Arm (b) Sensor Arm Base

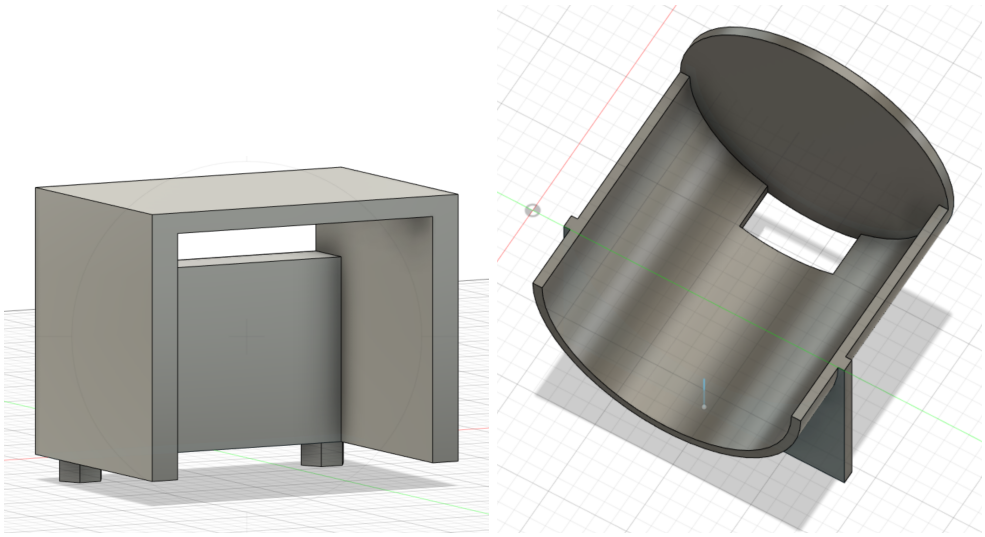
Figure 40

This design allows accurate positioning of our sensors so reliable and accurate readings can be taken every time.
There were two major issues:

- **Time**, the time taken for positioning the arm correctly for each alien would greatly increase the overall run time.
- **Vulnerability**, by exposing all our detection equipment through a single fault point away from our rover, a single collision could destroy all our sensing equipment.

These issues resulted in us removing the sensor arm from our final design.

Sensor Housing



(a) Sensor Housing V1 (b) sensor Housing V2

Figure 41

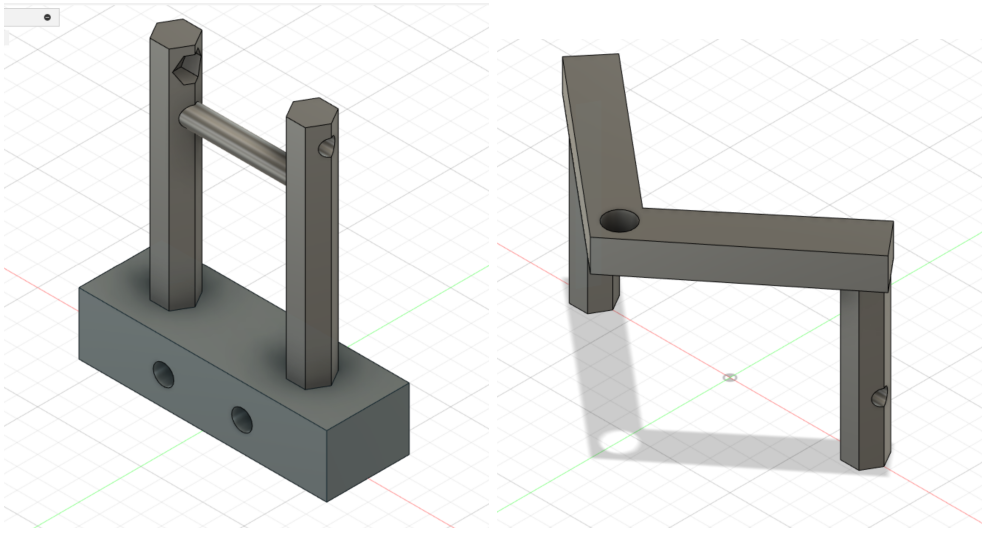
Figure 41a has all sensors attached to the roof of the housing with wiring running out the hole in the back. Housing slots into the chassis using hexagonal cutouts.

Figure 41b guides the alien into the centre of the housing with sensors attached to the roof, wiring runs through the cutout hole. Housing is screwed to chassis using rear plate.

These designs could somewhat act as platforms for our sensors but were rejected due to:

- **Weight** they would add to the rover.
- **Distance of sensors** to aliens was in excess of our tolerance levels (around 10 cm).

This led to a complete redesign, separating the RF and IR sensors from the magnetic sensor, slimming down the design and adapting it to more easily integrate with the chassis.

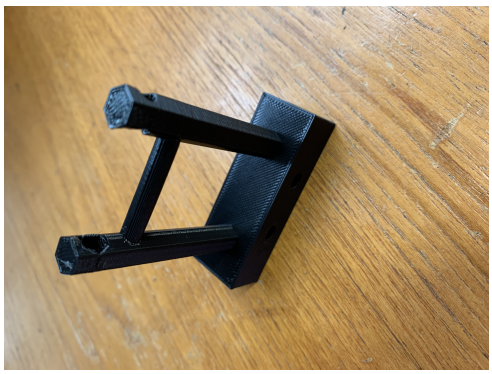


(a) Bottom module (b) Top module

Figure 42

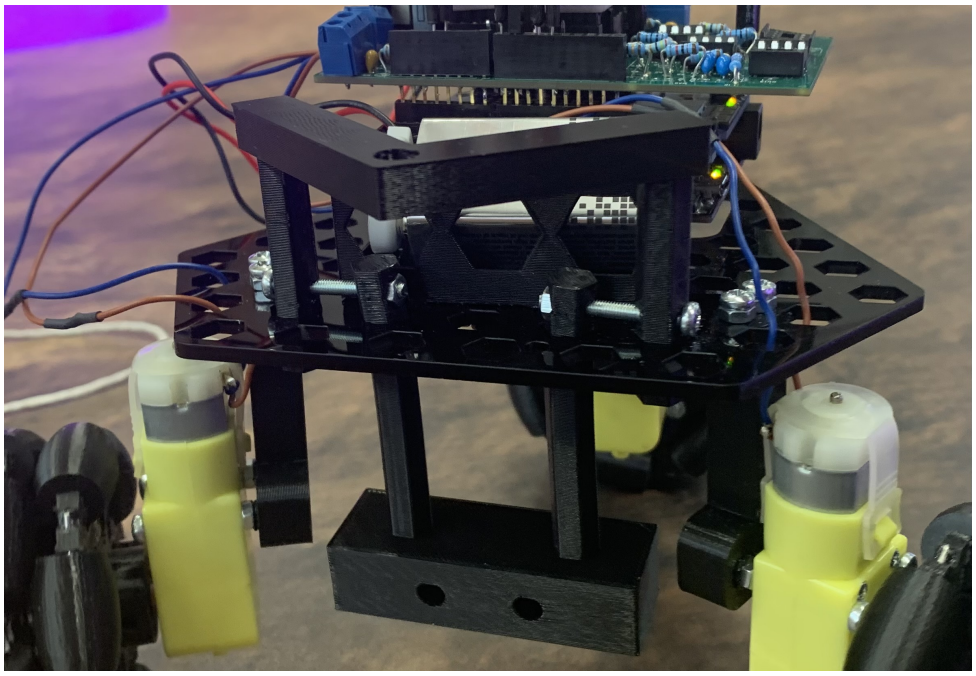
Figure 42a holds both RF and IR sensors in the cutout holes, measured to intercept the alien on its stomach, connects through its hexagonal extrusions through the chassis with the top module seen in Figure 42b. The top module holds the magnetic sensor above the head of the alien allowing the distinguishing of polarity.

Figure 43a and Figure 43b show the printed solution with Figure 43c displaying the sensor holder integrated onto the rover with screws holding the top and bottom modules together, see Appendix for BOM.



(a) Printed bottom module

(b) Printed top module

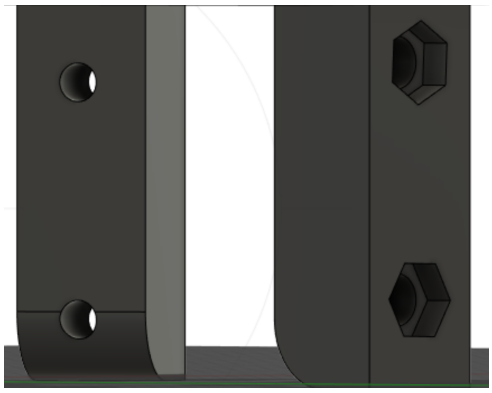


(c) Implemented sensor housing

Figure 43

Motor Mounts

After removing the mounts from the chassis to be developed independently and attached later initial designs were created in Fusion360 and printed:



(a) Motor Mount V1 in Fusion360

(b) Motor Mount V1 printed

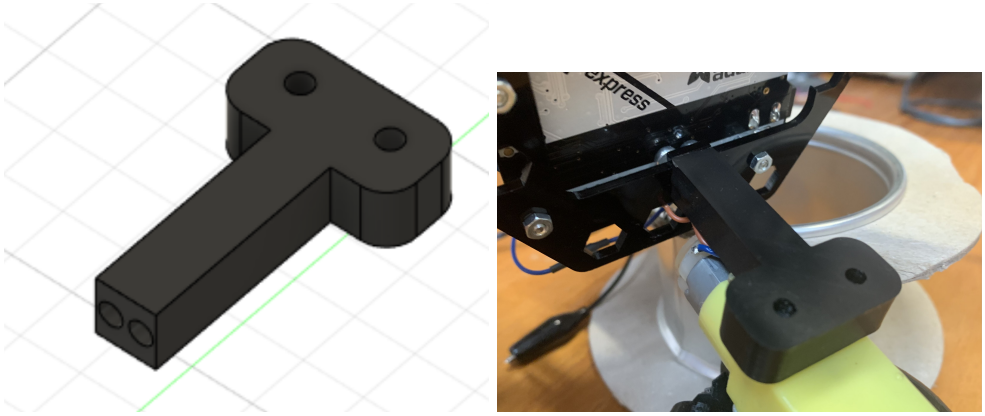
Figure 44

Figure 44a shows both sides of each mount which attaches to the motor on the side with 4mm circular holes. 44b displays the printed design, the side containing the bronze heat inserts is the top connection through which the mounts were attached to the chassis.

After implementation and testing the following was observed:

- **Drift**, as the wheels were not centred about the centre of each edge of the cutoff triangle the rover would drift while moving.
- **Heat inserts**, performed as desired so will be used in all further designs

To rectify the drift of the rover the mounts were redesigned so that each motor would now attach vertically:



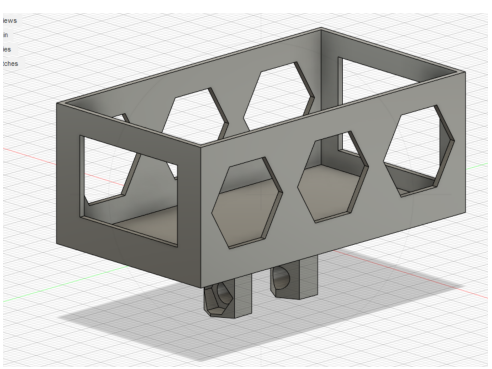
(a) Final Motor Mount in Fusion360 (b) Final Motor Mount printed

Figure 45

Figure 45a contains the motor mount we used in our final design, attaches to the chassis through the use of heat inserts in the two close together holes in the foreground. The holes with greater spacing provide screw holes to attach to the motor as seen in 45b.

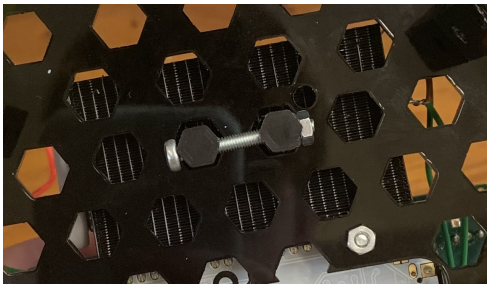
Battery Holder

To fix the battery (see Appendix) in place a simple battery holder was designed in Fusion360 and implemented into our rover, taking advantage of the chassis's hexagonal cutouts.



(a) Battery Holder in Fusion360

(b) Printed Battery Holder



(c) Connecting screw

Figure 46

Figure 46a shows the design of the battery holder, hexagonal extrusions go through the chassis and a screw then connects the extrusions as seen in Figure 46c, attaching the holder to the chassis as seen in Figure 46b.

3.2.6 Future improvements

The following are improvements to the chassis and attachments of the rover that could be implemented to improve the overall design:

- **Altering top module of sensor support**, in the current design for the hall effect sensor to pick up a suitable signal it must be suspended from the top module.
  Two ways to improve this are currently being considered:

  - **Shielding**, have a circular attachment which will shield it while dangling.
  - **Shortening**, by shortening the hexagonal legs of the top module the hall effect sensor will be able to be shielded in the hole at an appropriate height.

  These improvements are not top priority as sensing while having no shielding is within tolerance levels for noise for our environment.

- **Altering battery holder**, the volume of the battery holder is over size and the holder shifts about its mounting slightly, adding unnecessary weight and a possible hazard. To fix a small redesign and print of the holder would suffice, however due to time limitations and the minimal effect of these issues it is low priority.

- **Better shielding**,the sensor mounts' shielding is not perfect as it does not fully isolate external penetrating signals, if there was more time we would redesign the sensor mounts to incorporate a Faraday cage [32] that would significantly isolate external noise so that rover could operate in more extreme environments.

3.3 Wheels

3.3.1 Planning

When choosing what wheels to use for our design we took in many considerations in concurrence with the project brief:

- **High manoeuvrability** to navigate complex terrains and potential obstacles with speed.
- **Low cost**, to lower costs we chose to 3D print our wheels and looked to designs where we could achieve this.
- **Ease of implementation**, we analysed whether each wheel type was feasible to build with the time and resources we had.
- **Power**, motor drivers were limited to 10 V so we analysed whether we would have enough power to drive certain wheel types.
- **Weight**, weight limit is approximately 0.8 kg, to eliminate weight we looked to a wheel design where as little material as possible is used, this in turn would further lower printing costs.

These considerations led us to Omni and Mecanum wheels. The key outstanding area of these wheels was their manoeuvrability.

(a) Example of a Mecanum wheel (b) Example of a Omni Wheel

Figure 47

Figure 47a contains a Mecanum wheel, its key property being the small rollers inset along the edge of the wheel at 45 degrees to the vertical, 4 working together allow more dynamic movement such as sliding sideways. Similarly Omni wheels have rollers inset into the rim as seen in Figure 47b however they are position parallel to the vertical, dynamic movement similar to the Mecanum wheel can be achieved using 3 wheels.

After further comparison of their capabilities, we opted for Omni wheels.

Reasons for choosing Omni wheels were:

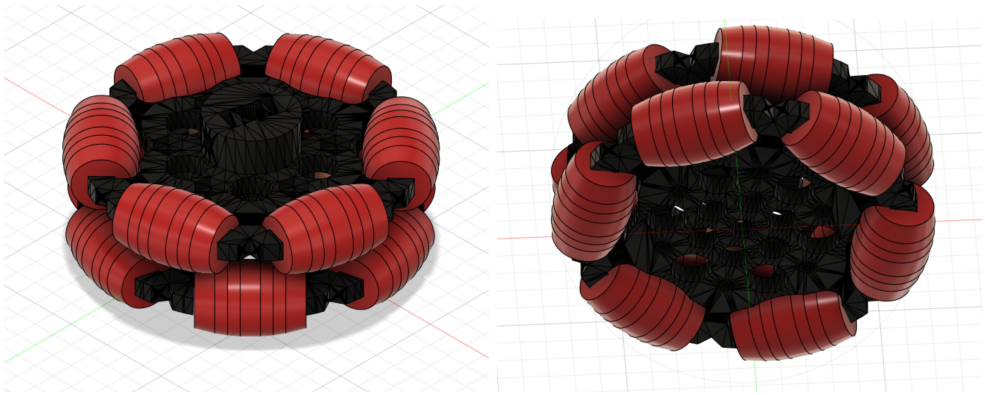
- **Ease of implementation**, taking into account the 45 degree offset of the rollers in Mecanum wheels added complications to printing and construction.
- **Weight**, a functional rover can be created with just three Omni wheels, providing a leaner design than the four-wheel requirement of a Mecanum setup.

- **Time**, the time required to print and construct the wheels would be significantly extended for a Mecanum design.

3.3.2 Design

We started the design process, simultaneously looking at the options of designing our own wheels from scratch versus leveraging a publicly available design to adapt to our needs.

A design was found that met our criteria in all capacities, deciding to go ahead with it with some slight adaptations to the axle hole. [33]



(a) Side of motor mount (b) Outer side of Wheel

Figure 48: Wheel views

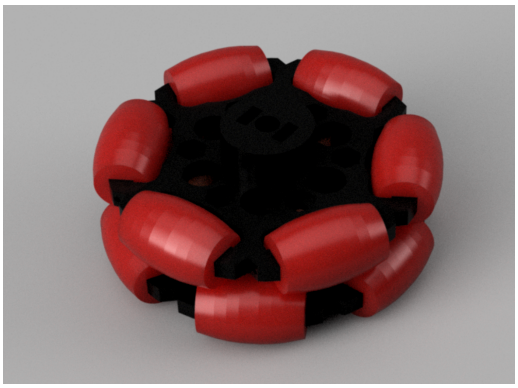
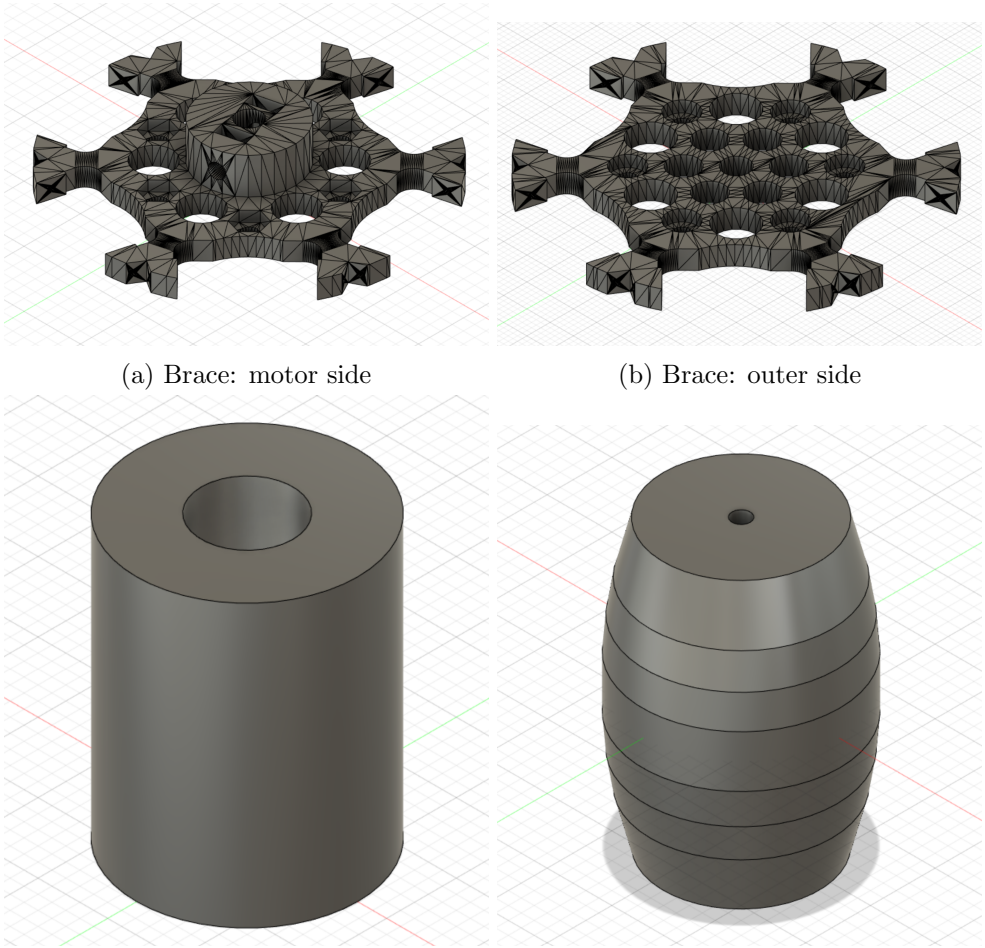


Figure 49: Render of Wheel

Figure 48a offers a view from the rear of the Omni wheel, showing the support structures and the assembly of the smaller wheels, the motor connection in the centre of the wheel can also be seen.
Figure 48b shows the outer side of the wheel, holes that will contain the nuts that fasten the screws connecting the parts of the wheel surround the centre.
Figure 49 displays the final render of our wheels.

3.3.3 Construction and Integration

From the STL files, we 3D printed the necessary components for our vehicle's wheels. These files provided the exact specifications for each part, measured to ensure that we could connect the wheels to the motor mounts correctly. The following images display all unique parts:



(a) Brace: motor side

(b) Brace: outer side

(c) Spacer

(d) Roller

Figure 50: All unique parts

Figure 50a connects to the motors axle through the central hole, holds 6 rollers as seen in 50d in the exterior gaps. Similarly Figure 50b holds the rollers in the same manner. Both bracers are then attached using the spacers in 50c and screws. For a comprehensive list of components, see Appendix.

We prototyped a single wheel to make sure that the design was applicable to our rover.

In this initial round we used a simple method:

1. Drill an approximately 4mm hole in each roller.
2. Strip paper clips coating with a wire coating remover.
3. Use metallic part of paper clip as axis of a roller.
4. Slot each roller into a slot in either bracer
5. Connect bracers using the screws and spacers.

However, this method provided varied results. There was a high degrees of axis wobble on the rollers. Some weren't rotating due to the bending of the paper clip axes. The paper clips holding the rollers in place also became loose.
Learning from the testing we implemented many improvements to ensure precise construction of high quality Omni wheels with very little roller axis wobble:

- **1.1 mm drill** to ensure consistent and precise size of roller axis holes.

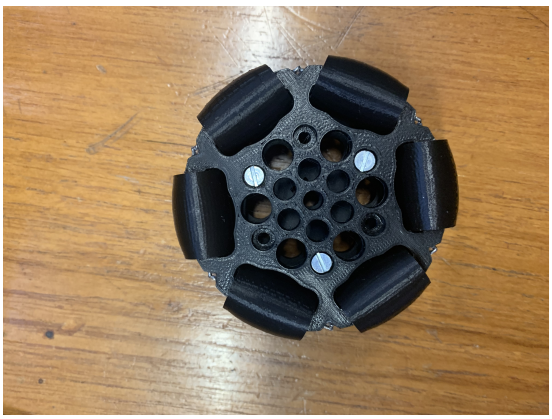
- **1 mm diameter steel rods** to use as the axis for each roller, ensuring each roller could smoothly roll without the risk of the axis getting bent.
- **File** to ensure each roller fit precisely into its bracer slot with no friction with the bracer making it unable to rotate.
- **Super glue** to glue the end of each steel rod that pokes through the roller hole to the bracer, fastening it.

And an improved method, by making the hole of each roller slightly smaller that desired diameter and making the slot for each roller in the brace slightly small. Both changes were done in fusion 360 by editing the STL files.

1. Use 1.1 mm drill to make the axle hole of each roller precisely and consistently.
2. Insert 1 mm diameter rod of 35 mm length into the hole of newly drilled roller at high pace until roller can freely spin around it.
3. File bracer until roller is able to slide into slot with negligible friction.
4. Slot rod through roller and hole in bracer.
5. continue until all slots in the bracers required for each wheel are full, then test each roller can freely move (reassemble a roller if not).
6. If each roller can freely move glue each axis to the bracer.
7. Connect the bracers with the screws, nuts and spacers, see Appendix for BOM.

Figure 51c shows the completed wheel.

(a) Motor side



(b) Outer side

(c) full wheel

Figure 51

3.3.4 Future Improvements

While the wheels have satisfied all requirements for our terrain, there are possible improvements which could be implemented given more time and resources:

- **Friction**, due to the rollers having a low level of friction in environments with smoother surfaces sliding can occur, decreasing manoeuvrability and speed.
  To rectify this a possible solution is to reprint the rollers in Fibersatin.
- **Number of rollers**, to increase the level of control that can be achieved the wheels could be redesigned to contain more than 12 rollers per wheel while also introducing higher level of redundancy.

4 Software Design

4.1 User Interface

The Metro Express M0 provided included a Adafruit WINC1500 WiFi Shield. The microcontroller connects to a WiFi network and acts as a web server, returning a web page when it receives an HTTP request. It also acts on certain paths in the HTTP request when the client interacts with the webpage [34].

In designing a user interface and control system, the following options were considered:

1. Keyboard buttons: WASD buttons control movement in each direction, very intuitive and logical
2. Website buttons: easy to use and implement, but does not enable precise control and movement
3. Physical joystick (Xbox controller): tactile and intuitive, with a high degree of manoeuvrability

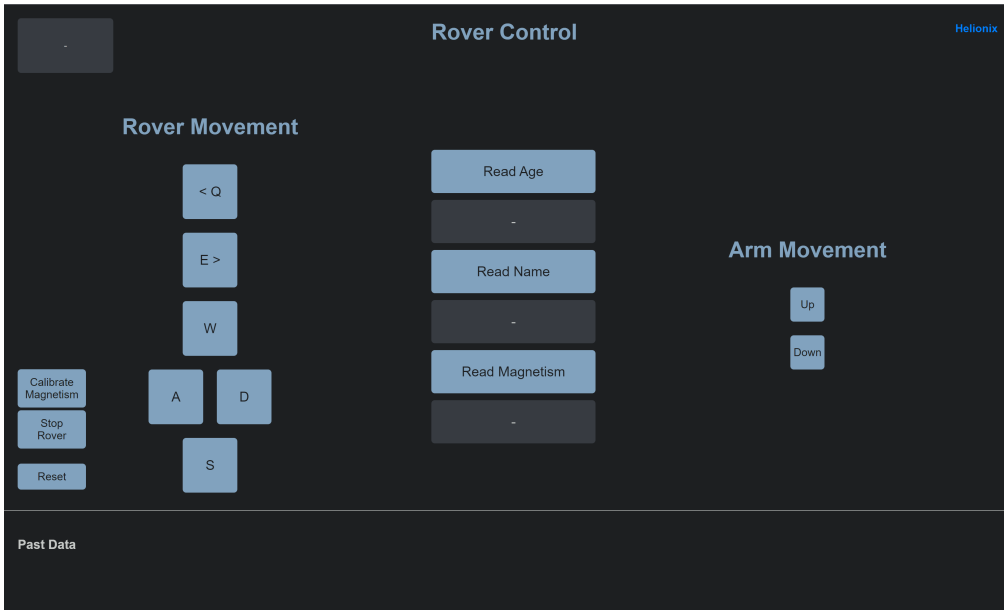


Figure 52: Web interface Prototype

The keyboard and website buttons were implemented in the initial design (Figure 52), with the following features:

1. WASDQE corresponds to forwards, backwards, right, left, clockwise and anticlockwise respectively. Multiple buttons can be pressed at the same time for movement along multiple axis.
2. The SHIFT button is used to reduce the speed of the rover when it is pressed, for finer movement control.
3. The web buttons can be used for control on mobile phones.
4. The 3 read buttons are used to obtain the various characteristics of the aliens.
5. The box in the left corner is used to display the current status of the read request, while the "Past Data" box displays the values of past data readings.

In the final design, the keyboard buttons, website buttons and physical joystick control were implemented in parallel on the user interface, giving users the option to choose one (Figure 53).

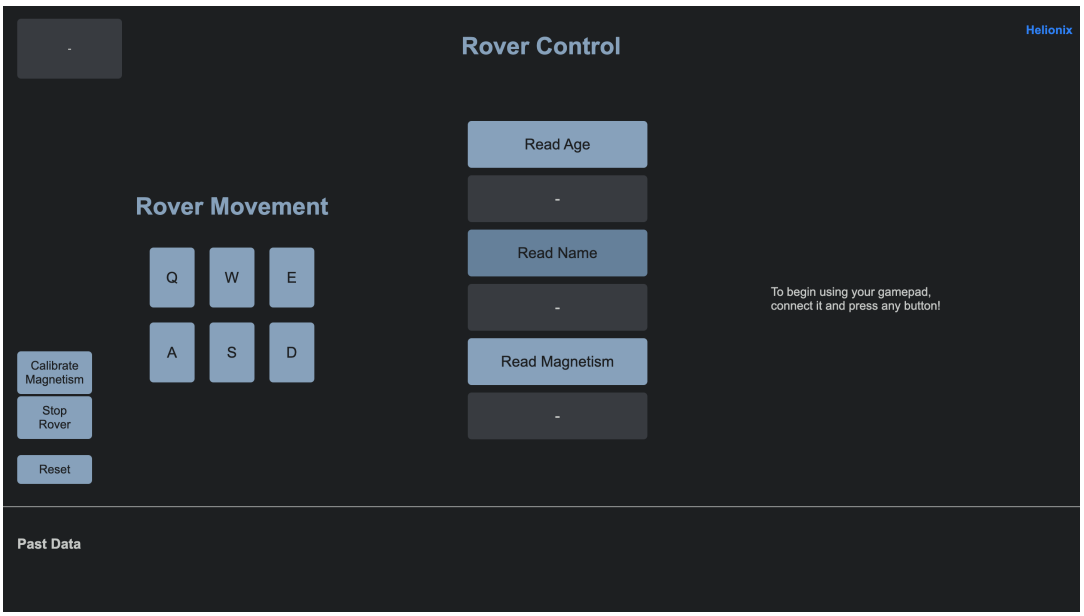


Figure 53: Final Version of Web interface

4.2 Movement Control

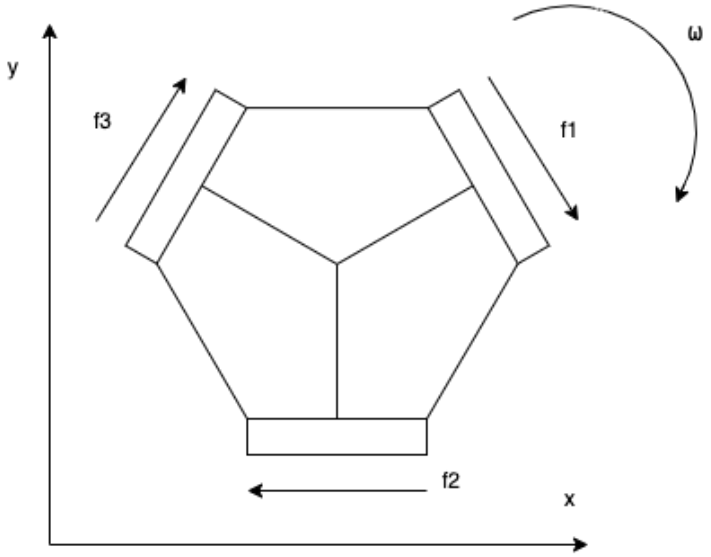


Figure 54: Omni wheel control diagram

A matrix equation was obtained from Figure 54. a_x, a_y and ω are state variables that are initialised at start-up. These variables are changed when a movement control button is pressed. These variables are multiplied by the inverse matrix and scaled to obtain PWM values for motors f1, f2 and f3.

$$\begin{bmatrix} a_x \\ a_y \\ \omega \end{bmatrix} = \begin{bmatrix} \sin 30^\circ & -1 & \sin 30^\circ \\ -\cos 30^\circ & 0 & \cos 30^\circ \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} f1 \\ f2 \\ f3 \end{bmatrix}$$

For instance, a keydown event on the W button increments a_y by a fixed amount, while leaving the other variables unchanged. Likewise, a subsequent keyup event on the W button decrements a_y for the same amount, ensuring that the rover comes to a stop along the y-axis. This allows for simultaneous control along multiple axes using a keyboard.

4.3 Remote Communication

4.3.1 Adafruit Web Server

To control the rover using a web server, the `WiFiWebServer.h` library was used, which handles HTTP requests made by browsers [34]. After constructing the server object with WiFiWebServer server, several handler functions were attached to handle incoming requests:

```
server.on(F("/"), handleRoot);
server.on(F("/forward"), forward);
server.on(F("/backward"), backward);
server.on(F("/left"), moveleft);
server.on(F("/right"), moveright);
server.on(F("/rotatecw"), rotate_cw);
server.on(F("/rotateccw"), rotate_ccw);
server.on(F("/stop"), stop);
```

`server.on` calls the given function when a `GET` request with the given string is encountered. For instance, `handleRoot` would send the HTML file of the webpage stored in the Adafruit's SPI flash.

The WiFi101 library resulted in some limitations. In particular, the buffer length was set to 1400 [35]. This limited the number of bytes that could be sent in a single `server.send`. To work around this issue, the text file containing the website was split into chunks, each of which was sent using `server.sendContent`. To initialise this process, `server.setContentLength` and `server.send` were used.

4.3.2 Web Client

To send commands in JavaScript from the web client to the webserver, `XMLHttpRequest` was used [36]. JavaScript event listeners were added to website buttons and keyboard buttons to enable detection and movement:

```
document.getElementById('readAgeBtn').addEventListener('click',function() {
    fetchData('age', 'ageBox');
});
document.addEventListener('keydown', function(event) {
    console.log('Key pressed: ', event.keyCode);
    if (event.keyCode === 87 && !isWKeyPressed) {
        isWKeyPressed = true;
        moveRover('forward');
    }
});
```

In JavaScript, an `XMLHttpRequest` object was initialised. Inside a handler function, a HTTP `GET` request could be issued:

```
function moveRover(direction){
    var xhr = new XMLHttpRequest();
    xhr.open('GET', "/"+ direction);
    xhr.send();
}
```

To reduce lag and improve network reliability, a mobile hotspot was used, instead of the EEERover WiFi network. During tests, the EEERover network was too slow and unreliable, with significant lag time between movement GET requests.

4.3.3 Controller Support

To enable support for controllers, we used the Gamepad API [37]. The `Navigator.getGamepads` method returns an array of all devices currently visible to the webpage, as `Gamepad` objects. The `Gamepad` object can be queried to find out which buttons and axes are pressed.

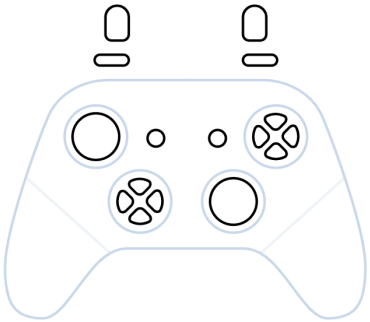


Figure 55: Xbox One Controller Buttons and Axes

We decided to make use of the Xbox One joystick axes to control the rover (Figure 55). The `reportOnGamepad` function queries the current position of the joystick axes and sends an HTTP `POST` request containing this information. The `POST` request method was used as it allows the client to send key-value tuples in the body of the request. This allows for variable speed control using the joysticks, which is a significant advantage over keyboard buttons. This function was called periodically using `window.setInterval`.

Initially, this control system was tested with intervals of 10 ms. This worked poorly in the artificial environment in the lab. Occasionally, there was significant delay between sending and receiving movement requests. This was likely due to the high frequency of `POST` requests, combined with significant noise and interference, as a result of being surrounded by electronic equipment.

To reduce connectivity issues, the frequency of HTTP `POST` requests was reduced. A POST request would only be sent if the change in joystick position is larger than a fixed threshold. The call frequency of `reportOnGamepad` was also increased to 20 ms.

4.3.4 Future Improvements

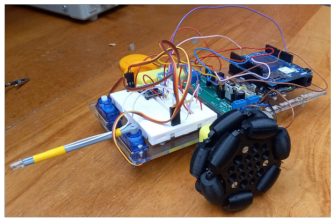
To further reduce connectivity issues and improve responsiveness, multiple solutions exist. One alternative would be to send commands over Universal Datagram Protocol (UDP) instead of Transmission Control Protocol (TCP), which is used by HTTP. Unlike TCP, UDP is a connectionless protocol with minimal overhead. It is suitable for time-sensitive applications because dropping packets is preferable to waiting for packets delayed due to re-transmission [38].

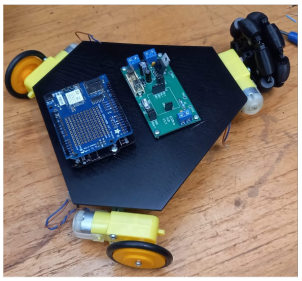
5 Integration & Testing

5.1 Integration Timeline

Week 2: Preliminary Integration

Mounting of Age (IR) sensor, Metro Board and Omni Wheel Prototype on EEBug Chassis



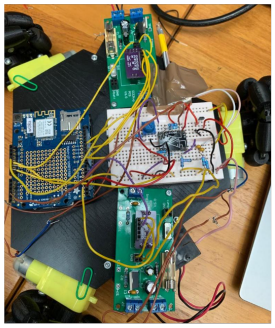


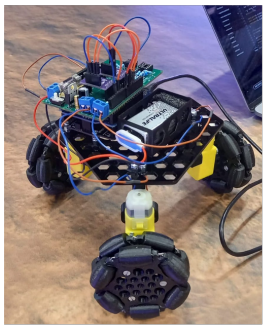
Week 2.5: Chassis V0

Construction of chassis V0, mounting of prototype wheels

Week 3: Chassis V0 Sensor Integration

Mounting of Age (IR) and Magnetism sensors, Metro Board, EEBug PCB and Omni Wheels on chassis V0



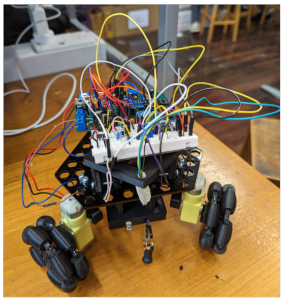


Week 4: Chassis V3

Mounting of PCB prototype board, Metro Board, Motor Mounts and battery holder, Omni Wheels on chassis V3

Week 4.5: Chassis V3 Sensor Integration

Mounting of Age (IR), Magnetism and Name (RF) sensors on a breadboard, sensor housing on chassis V3



Week 5: Full integration

Integration of final PCB with sensors and sensor mounts

5.2 Integrated Tests

After full integration of the various submodules, functional tests were conducted. This included Name (RF), Age (IR) and magnetism detection tests on aliens of various heights (Figure 56).

Figure 56: Different alien types

This is particularly important for validating the performance of the magnetic sensor, which hangs above the head of the alien. This also ensures that the Name (RF) and Age (IR) sensors were able to detect their respective signals through aliens of different body types. For instance, the brown test alien is the smallest (Figure 57). The rover was successful in measuring all 3 characteristics from each of the alien types.

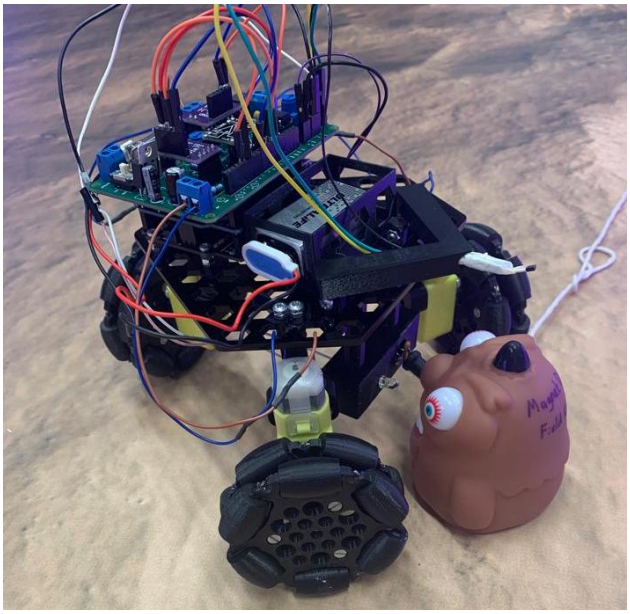


Figure 57: Test alien for magnetism

6 Appendix

6.1 Code

Github Link: `https://github.com/rolandocharles/Helionix-EEERover` [2]

6.2 Bill of Materials

Table 4 shows the components list and their associated cost. However, many of the components were sourced from the lab or recycled from EEEBug PCBs such that their cost is purely an estimate. The estimated total cost is £129.1.

Table 4: Bill of Materials

Submodule	**Part**	**Quantity**	**Unit Cost**	**Total Cost**
Name (RF)	ICL76605CPA +ve to -ve Converter	1	1.74	1.74
	MC33078P Operational Amplifier	1	6.55	6.55
	10uF Capacitor	2	0.09	0.18
	1uF Capacitor	1	0.82	0.82
	1k Resistor	2	0.05	0.1
	560k Resistor	1	0.05	0.05
	Diode	1	0.05	0.05
	Variable Resistor	1	0.98	0.98
	10mH Inductor	1	0.77	0.77
	680pF Capacitor	1	0.08	0.08
Age (IR)	4.7 nF Ceramic Capacitor	2	0.05	0.1
	LT1366 Operational Amplifier (shared with magnetism submodule)	1	7	7
	SFH300-3/4 Phototransistor	1	0.66	0.66
	7.5kΩ Resistor	1	0.05	0.05
	510kΩ Resistor	1	0.05	0.05
	10kΩ Resistor	1	0.05	0.05
	27kΩ Resistor	1	0.05	0.05
	82kΩ Resistor	1	0.05	0.05
Magnetic	LT1366 Operational Amplifier (shared with IR submodule)	1	0	0
	SS495A Hall Effect Sensor Linear	1	5.76	5.76
	10kΩ Resistor	1	0.05	0.05
	430kΩ Resistor	1	0.05	0.05
	61kΩ Resistor	1	0.05	0.05
	100kΩ, Through Hole Trimmer Potentiometer	1	0.76	0.76
PCB	Female Header Pin Connectors (Various Sizes)	6	0.46	2.76
	1x3 Way Male Header Pin Connectors	2	0.23	0.46
	Physical PCB board (final version)	1	6.1	6.1
	CFH02 Fuseholder	1	0.31	0.31
	1A Fuse	1	0.9	0.9
	1N4004 Diode	1	0.42	0.42
	LM78M05CT 5V Linear Regulator	1	0.33	0.33
	100µF Electrolytic Capacitor	1	0.34	0.34
	10µF Electrolytic Capacitor	1	0.09	0.09
	STSSS9121 SPDT Slide Switch	1	0.5	0.5

	1Ω Resistor	3	0.05	0.15
	10nF Ceramic Capacitor	3	0.08	0.24
	DIL-8 Socket	3	0.6	1.8
	2 Way Screw Terminal Block	4	0.18	0.72
Mechanical Parts	Motor mounts (final version)	3	0.29	0.87
	M3 Heat Inserts	12	0.06	0.72
	M3 Nuts	15	0.03	0.45
	M3x25mm Screws	9	0.05	0.45
	M2.5 Nuts	16	0.03	0.48
	M2.5x12mm Screws	4	0.05	0.2
	Sensor mount (final version)	1	1.52	1.52
	Battery holder(final version)	1	0.216	0.216
	DC Motor Assembly	3	2.96	8.88
	Base (final version)	1	4.35	4.35
Wheels	Motor Brace	3	0.32	0.96
	Outer Brace	3	0.22	0.66
	Spacer	9	0.02	0.18
	Roller	36	0.14	5.04
	M3x16mm Screws	9	0.05	0.45
	M3 Nuts	9	0.03	0.27
	35mm Rods	36	0.18	6.48
Battery	Press Stud Connector for PP3	1	0.46	0.46
	PP3 Duracell Plus MN1604 9V	1	3.84	3.84
Other	Metro Express M0	1	24.6	24.6
	Wifi Shield	1	24.9	24.9
	Jumper Wires	10	0.2	2

6.3 Footprints

Table 5 contains the list of components whose symbol, footprint and/or 3D model was not available in the KiCad standard library. These were all sourced from SnapEDA [39].

Component	Symbol, footprint or 3D model
CFH02 Fuseholder	`MC000827`
Variable Resistor	`TRIM_3386F-1-103TLF`
2 Way Screw Terminal Block	`282837-2--3DModel-STEP-269445`
STSSS9121 SPDT Slide Switch	`500SSP1S2M2QEA--3DModel-STEP-520916`

Table 5: Missing footprint table

6.4 Thanks

We would like to thank:

Amine Halimi for processing and 3D printing / laser cutting all of our parts.
Steven Wright and Vic Boddy for help in troubleshooting the radio sensor.

7 References

References

[1] D. E. Stott, *EEERover Project Brief*, 2023. [Online]. Available: https://github.com/edstott/EEERover/blob/main/doc/brief.md.

[2] *Helionix github repository.* [Online]. Available: https://github.com/rolandocharles/Helionix-EEERover.

[3] D. E. Stott, *EEERover Project Brief*, 2023. [Online]. Available: https://github.com/edstott/EEERover/blob/main/doc/README.md.

[4] *Mc33078 data sheet*, Texas Instruments. [Online]. Available: https://www.ti.com/lit/ds/symlink/mc33078.pdf?ts=1686611173437.

[5] *Icl7660 data sheet*, mouser. [Online]. Available: https://www.mouser.co.uk/datasheet/2/609/ICL7660_MAX1044-3122848.pdf.

[6] *RTC1000 Oscilloscope)*, Rohde & Schwarz. [Online]. Available: https://scdn.rohde-schwarz.com/ur/pws/dl_downloads/dl_common_library/dl_brochures_and_datasheets/pdf_1/RTC1000_dat_en_3607-4287-32_v0501.pdf.

[7] *BPW20RF Silicon Photodiode*, Vishay Semiconductors. [Online]. Available: https://docs.rs-online.com/4d00/0900766b80e2fb21.pdf.

[8] *BPV11F Silicon NPN Phototransistor*, Vishay Semiconductors. [Online]. Available: https://docs.rs-online.com/8e17/0900766b80e22c14.pdf.

[9] *IR Receiver Modules for Remote Control Systems*, Vishay Semiconductors. [Online]. Available: https://www.vishay.com/docs/82460/tsop45.pdf.

[10] *ZTP-148SR Thermopile IR Sensor*, Amphenol Advanced Sensors. [Online]. Available: https://4donline.ihs.com/images/VipMasterIC/IC/AMPH/AMPH-S-A0004027521/AMPH-S-A0004027521-1.pdf.

[11] *SFH 300 FA*, OSRAM Opto Semiconductors GmbH, 2014. [Online]. Available: https://look.ams-osram.com/m/35ebcc342238d580/original/SFH-300.pdf.

[12] *Characteristics of Phototransistors*, EGG Optoelectronics. [Online]. Available: https://johnloomis.org/ece445/topics/egginc/pt_char.html.

[13] *LT1366/LT1367/LT1368/LT1369 Dual and Quad Precision Rail-to-Rail Input and Output Op Amps*, Linear Technology Corporation. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/1366fb.pdf.

[14] *Atmel-42181G–SAM-D21$_D$atasheet*–09/2015, Atmel. [Online]. Available: https://cdn-shop.adafruit.com/product-files/2772/atmel-42181-sam-d21_datasheet.pdf.

[15] *MCP6291/1R/2/3/4/5 1.0 mA, 10 MHz Rail-to-Rail Op Amp*, Microchip Technology Inc. [Online]. Available: https://www.farnell.com/datasheets/809135.pdf.

[16] *Magnetic distortion in motion labs.* [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0966636208003858?ref=pdf_download&fr=RR-2&rr=7d7beb8bcc68dd83.

[17] *Hall effect sensor variants*, RF wireless world. [Online]. Available: https://www.rfwireless-world.com/Terminology/Unipolar-hall-sensor-vs-Bipolar-hall-sensor-vs-Omnipolar-hall-sensor.html.

[18] E. Stott, *Strength of magnetic field.* [Online]. Available: https://edstem.org/us/courses/39506/discussion/3150370.

[19] Barley$_L i$, *Magnetoresistive vs. hall effect in sensor applications.* [Online]. Available: https://forum.digikey.com/t/magnetoresistive-vs-hall-effect-in-sensor-applications/1185.

[20] *Reedswitch.* [Online]. Available: https://en.wikipedia.org/wiki/Reed_switch.

[21] *Linear Hall-effect sensor ICs SS490 Series*, Honeywell. [Online]. Available: https://www.farnell.com/datasheets/3154557.pdf.

[22] *Sensitive magnetometry in challenging environment.* [Online]. Available: https://pubs.aip.org/avs/aqs/article/2/4/044702/997296/Sensitive-magnetometry-in-challenging-environments.
[23] *Magnetometer.* [Online]. Available: https://en.wikipedia.org/wiki/Magnetometer.
[24] *Using multiple shields*, 2016. [Online]. Available: https://forum.arduino.cc/t/using-multiple-shields/393230/4.
[25] *PCB Design Guidelines For Reduced EMI*, Texas Instruments. [Online]. Available: https://www.ti.com/lit/an/szza009/szza009.pdf.
[26] *PCB Design Techniques to Reduce EMI*, Altium. [Online]. Available: https://resources.altium.com/sites/default/files/uberflip_docs/file_731.pdf.
[27] *LM78M05CT*, Texas Instruments. [Online]. Available: https://www.ti.com/lit/ds/symlink/lm341.pdf.
[28] *KiCad Version 7.0.0 Released*, KiCad. [Online]. Available: https://www.kicad.org/blog/2023/02/Version-7.0.0-Released/.
[29] *Guide to common pcb issues failures.* [Online]. Available: https://www.mclpcb.com/pcb-guide/.
[30] *Jlc pcb.* [Online]. Available: https://jlcpcb.com/.
[31] *Pcb track width and track resistance.* [Online]. Available: https://pcbdesignworld.com/article/pcb-track-width-and-track-resistance-importance-calculation-and-design-tips.
[32] *Faraday cage for emc improvement of electronic devices.* [Online]. Available: https://ieeexplore.ieee.org/document/7412272.
[33] remrc, *Omni wheels for robotics*, Thingiverse, 2021. [Online]. Available: https://www.thingiverse.com/thing:5027191.
[34] K. Hoang, *WiFiWebServer.* [Online]. Available: https://github.com/khoih-prog/WiFiWebServer.
[35] *WiFi101*, Arduino Libraries. [Online]. Available: https://github.com/arduino-libraries/WiFi101/blob/master/src/socket/include/socket.h.
[36] *XMLHttpRequest*, MDN Web Docs. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest.
[37] *Gamepad API*, MDN Web Docs. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Gamepad_API.
[38] *UDP (User Datagram Protocol)*, MDN Web Docs. [Online]. Available: https://developer.mozilla.org/en-US/docs/Glossary/UDP.
[39] *Snapeda.* [Online]. Available: https://www.snapeda.com/.