Electronics Design Project Report

ELEC40006 – Team Zenithar (Group 28)

Zakariyyaa Chachia	EEE/EIE Year 1
Hassan Choudhury	EEE/EIE Year 1
Joe Elguezabal	EEE/EIE Year 1
Gurjan Singh Samra	EEE/EIE Year 1
Ryan Voecks	EEE/EIE Year 1
Letong Xu	EEE/EIE Year 1

Submission to Edward A Stott

Submitted on 15th June 2023

Word count: 8488

Imperial College London

Contents

Abstract 2
1 Introduction 2
2 System Design, Implementation, and Testing 3
  2.1 Product Design Specification 3
  2.2 Submodule Definitions 7
  2.3 Project Organisation and Teamwork 8
  2.4 Radio Receiver 10
  2.5 Infrared Sensor 14
  2.6 Magnetic Field Sensor 17
  2.7 Drive Train and Chassis 19
  2.8 Website and Motor Control 22
  2.9 Full System Design 26
3 Discussion 28
4 Conclusion 28
References 29
Appendices 30

Abstract

The aim of this project is to develop a rover to take readings of electromagnetic signals, in an arena simulating a foreign planet with aliens. Remote-controlled robotics are hugely important in modern logistics and manufacturing, as well as space exploration and research. Electromagnetic signals are fundamental to the operation of communications systems. Doing research and development in both these areas therefore contributes to important technological and scientific advancements.

Each simulated alien has an age and name encoded using electromagnetic signals, and may also have a magnetic field. The ideal rover moves between the aliens as quickly as possible, and takes readings very quickly from a far distance. Our rover is capable of both tasks, moving between aliens at 0.37m/s, and taking readings in under 400ms from a distance of 7cm. This meets the product design specification criteria, but does not meet some of the ambitious goals we set.

The design of the rover can be easily built upon to either improve or add additional functionality. The speed can be increased to reduce the time taken to survey aliens, and more sensors can be mounted on the sensor package to gather a wider range of signals.

The source code for the project is available at https://github.com/Zakattack1045/Zenithar.

1 Introduction

The aim of the project is to design a remote-controlled rover to take readings of electromagnetic signals in an arena, simulating a remote planet filled with electromagnetic signal-emitting aliens. Remote-controlled robots are important in many real-world tasks, such as modern warehouse and factory operation, and search and rescue missions. They are also an essential part of every space operation. Electromagnetic signals are also hugely important in communications, as well as inspecting foreign objects in space.

Each simulated alien emits a 61kHz carrier wave amplitude shift-keyed two-level modulation UART signal, corresponding to its name. It also emits infrared pulses with a period corresponding to its age. Each alien may also have a magnetic field pointing upwards or downwards, or no magnetic field at all. A rover must be capable of traversing the arena to find the characteristics of all the aliens, and the project must be delivered within a budget of £60. Some aliens are 'shy', meaning that they will not emit a signal when a rover which is too heavy is nearby. Therefore, the rover must also be lightweight.

To develop a solution, we came up with a quantitative product design specification, and divided the problem into submodules. Each team member then chose a submodule to work on, and these modules were integrated together to form the final rover. Collaboration was key to ensuring cross-compatibility between submodules, and the production of a high-quality solution. Each submodule was tested independently, and as part of a collection of submodules, to ensure that it was working correctly.

2 System Design, Implementation, and Testing

2.1 Product Design Specification

This specification is written to describe the desired functionality of the rover, following the guidance given in the project brief (1) by the client, Ed Stott. Wherever possible, these goals have been quantified, to increase objectivity. In the following specification, a requirement from the project brief is an objective which must be, while a target determined by our team is an objective which should be met.

2.1.1 Performance

The rover must be able to identify the name, age, and magnetic field direction of each alien in the arena. It should be able to make automatic measurements within 400ms at a range of 4cm from the centre of the alien. For each alien, the name and magnetic field direction must be determined exactly, and the age must be found to the nearest decade. The rover must send the data it records to a website to be displayed. This website must also be able to be used to control the movement of the rover. The rover should have a top speed of at least 0.5m/s.

2.1.2 Environment

The environment is a 3.6m x 2.4m flat rectangular arena, with a hard, smooth, floor, and rigid walls. An unknown number of aliens will be distributed around the arena. The arena contains 4 hemispherical domes with a radius of around 30cm, made of hard, translucent, white plastic. These domes contain LEDs which cycle through the RGB spectrum, causing the dome to change colour. The arena also contains 4 smaller hemispherical domes with a radius of around 15cm, made of hard, transparent plastic. An unknown area of the arena has underfloor weight sensors, which prevent nearby 'shy' aliens from emitting signals when the total dynamic weight of rovers in this area is too great.

2.1.3 Life in service

The rover should be able to operate continuously from fully charged batteries for at least 30 minutes. The rover should also be able to function entirely as specified throughout an entire day of use, without repair.

2.1.4 Maintenance

The rover will be powered by AA alkaline batteries, which should be replaceable. All modules should be easily removable for improvement and repair.

2.1.5 Target product cost

The product must be delivered for a total cost of no more than £60. This includes all the materials and processes required for research, development, and the construction of the final design. Components and materials should be purchased from the official supplier websites, listed in Appendix 1. Any components which are part of the 2022-23 EEEBug (2) or the 2022-23 EEEBug

Expansion Kit (3), along with any other components, materials, and processes which are freely available in the EEE Level 1 Lab, do not count towards this cost.

2.1.6 Competition

There are 30 other teams designing rovers to meet the same project brief. The rover will survey aliens in the arena at the same time as several other groups. The rover should be able to withstand interference from other rovers. The rover should record the data for all the aliens in less time than any other team's rover.

2.1.7 Shipping

The product must be delivered by hand to the client in the EEE Level 1 Labs on June 20th, 2023.

2.1.8 Packing

The product does not require any packaging as it will not be shipped.

2.1.9 Quantity

Only one final design must be constructed and delivered to the client.

2.1.10 Manufacturing facilities

All the facilities in the EEE Level 1 Labs are available freely for use. These include equipment for soldering, laser cutting, 3D printing, cutting, drilling, and other mechanical alternations. 3D printing filament is available for free, but other materials must be purchased.

2.1.11 Size

The rover should have dimensions not exceeding a base of 20cm x 20cm, and a height of 12cm.

2.1.12 Weight

The rover must have a static weight which is less than the threshold for the weight sensors in the arena. The rover should also have a maximum dynamic weight which is less than this threshold. The threshold weight is unknown, although there is a unitless scale in the arena which displays whether the rover's dynamic weight is over the threshold.

2.1.13 Aesthetics, appearance, and finish

The rover should be made aesthetically pleasing, although never at the cost of functionality. Wires should have colours which indicate their purpose. The website's user interface should be simple, clean, and easy-to-use. It should have no unnecessary components.

2.1.14 Materials

Several breadboards, and many basic electronic components are freely available in the EEE Level 1 Labs, along with 3D printer filament. Many more electronic and mechanical components are available at cost from official supplier websites, and acrylic for laser cutting is available in several sizes for purchase from the EEE Department Stores.

2.1.15 Product life span

The product should be able to be stored without degradation for 5 weeks, then used continuously for an entire day without loss of functionality.

2.1.16 Standards and specifications

Circuit diagrams presented should follow IEEE standards (see Appendix 2). The product must meet the specification of the client entirely.

2.1.17 Ergonomics

The rover should be easy to carry by hand, and easy to turn on and off. The rover should automatically connect to the website and run the required software on startup. The rover should be easily controllable using the website, and the maximum input response time should be less than 50ms.

2.1.18 Customer

The customer is the client, and the end user is our team.

2.1.19 Quality and reliability

The rover should have a high build quality, with no noticeable damage on the components in the final design. All mechanical fastenings should be tightened, and there should be no loose wires or components. The rover and website should function fully without glitches for a period of 30 minutes.

2.1.20 Shelf Life

The product should be able to be stored in a cardboard box or on a shelf, without degradation, for 5 weeks.

2.1.21 Processes

The processes which may be used include (but are not limited to) circuit simulation, soldering, 2D and 3D computer aided design, 3D printing, laser cutting, drilling, cutting, and mechanical fastening.

2.1.22 Timescale

The design process started on May 17th, 2023. final product must be delivered by 9am on June 20th, 2023.

2.1.23 Testing

Each module should be tested independently to ensure that it works fully. The final product should be tested rigorously to ensure that it meets quality and reliability standards. Testing methods include running software edge cases, comparing signals with an oscilloscope, and timing rover movement.

2.1.24 Safety

The product should be safe to use, according to the requirements set by the British Standards Institution (see Appendix 2).

2.1.25 Company constraints

The EEE Level 1 Labs are open on weekdays from 9am-6pm. The labs, and their facilities and resources, are shared with other EEE students. The time taken for laser cutting, 3D printing, and component collection may be longer than expected due to servicing of other groups.

2.1.26 Market constraints

Due to the number of competing teams, manufacturing processes and component delivery could be slowed.

2.1.27 Patents, literature, and product data

See references for a list of literature used in designing this product. All data from the development and testing of the product will be stored in a private folder on the Imperial College OneDrive, accessible only to team members. No patents will be filed for this product.

2.1.28 Political and Social Implications

The rover should not cause damage to any other rovers while in the arena.

2.1.29 Legal

There are no legal concerns for the product or its development.

2.1.30 Installation

The product will be delivered fully assembled and will not require installation.

2.1.31 Documentation

A report detailing the product development process and final design will be submitted to the client by June 15th, 2023.

2.1.32 Disposal

The final design should be easy to disassemble, and most components should be reusable.

2.2 Submodule Definitions

To design the system, the problem was divided into submodules, which form the full rover. The following modules, and their interfaces and requirements, were defined using the project brief (1). The microcontroller being used is an Adafruit Metro M0 Express.

2.2.1 Radio Receiver

To find the name of each alien, hardware is required to receive a 61kHz carrier radio wave modulated with two-level amplitude-shift keying, and a data rate of 600 bits per second. The name is encoded using ASCII in UART packets with a start bit and a stop bit. Each alien's name is four characters long, including an initial '#' symbol (1). The radio receiver hardware must detect the wave and convert it into a 3.3V binary signal to input to the RX pin of the microcontroller. Software must then be used to take the UART input signal and return a three-character name as a string.

2.2.2 Infrared Sensor

Each alien emits infrared pulses at an interval in milliseconds equal to their age in centuries. Each pulse lasts 50µs, and the period of the pulses is between 1.0ms and 7.4ms. The infrared pulses have a wavelength of 950nm (3). Hardware is required to take detect the infrared pulses, and software must be used to determine the frequency of these pulses from hardware readings. The software should return the age of the alien.

2.2.3 Magnetic Field Sensor

Each alien either contains no magnet, or has a magnet mounted approximately 10mm below the highest point of its top. If an alien has a magnet, it will either have a field pointing 'up' (north face upwards) or 'down' (south face upwards). Hardware is required to take readings of the magnetic field. Software must then use these readings to determine and return the magnetic field direction, or lack thereof.

2.2.4 Movement

To move between aliens, a drive train is required. This drive train will have several wheels, driven by motors. These motors will be connected to a motor controller PCB, which takes power inputs directly from the batteries, and signal inputs from the microcontroller. The motors and motor controller PCB must be mounted on the chassis. Software is required to control the motors using signal inputs to the motor controller PCB.

2.2.5 Chassis

To physically connect the components, a chassis is required. This chassis must have holes for the sensors, motors, microcontroller, motor controller, and battery packs to be mounted using screws. The chassis must be strong and durable enough to withstand collisions with other rovers, and light enough not to trigger the weight sensors in the arena.

2.2.6 Website

A website is required which has a user interface to display the characteristics of encountered aliens and control the movement of the rover. The website should input instructions for movement from the user, then send this data to the microcontroller. The website should receive and display data from the microcontroller describing the characteristics of aliens.

2.2.7 Full System Software

The microcontroller should use the results of the radio receiver, infrared sensor, and magnetic field direction submodules to determine when an alien is detected and send its information to the website. The microcontroller also needs to take movement instructions from the website, interpret these, and output the correct signals to the motor controller.

2.3 Project Organisation and Teamwork

To design the rover, team members were allocated specific submodules to work on, with consistent communication to ensure compatibility between modules. Additional members were allocated to modules which were found to be more complex. *Table 2.3.1* on the next page shows the major contributions of each group member during the week commencing on each day.

Team Member	**15th May**	**22nd May**	**29th May**	**5th June**	**12th June**
Zakariyyaa Chachia	Researching and testing basic website design	Developing code to control the motors from the website	Adding keyboard inputs for motor control and writing presentation	Adding swing turning and game controller inputs to the website	Integrating motor control and sensor software and writing report
Hassan Choudhury	Researching and designing infrared sensor circuits	Building and testing infrared sensor circuits	Developing software for infrared sensor and writing presentation	Testing infrared sensor and integrating it with radio sensor	Testing sensors and writing report
Joe Elguezabal	Researching and designing infrared sensor circuits	Testing infrared sensor circuits and Hall effect circuits	Testing Hall effect and infrared sensor circuits and writing presentation	Testing motors and developing drive train	Testing sensors and writing report
Gurjan Singh Samra	Researching Hall effect sensors to order	Building Hall effect sensor circuits and researching magnetism	Designing a stripboard layout for sensors and writing presentation	Writing software for and testing a magnetometer	Designing a stripboard layout for the final design and writing report
Ryan Voecks	Developing and testing radio antenna	Building and testing amplifiers	Developing software for radio and infrared sensors and writing presentation	Integrating sensor software and designing final chassis	Integrating motor control and sensor software and writing report
Letong Xu	Conceptualising basic chassis	CAD modelling of an initial chassis design	Preparing chassis for laser cutting and writing presentation	Testing initial chassis and designing final chassis	Assembling and testing final chassis design and writing report

Table 2.3.1: major contributions of team members during each week

These contributions are quite different to those projected during the first week (see Appendix 3) and the plan made before the initial report on 2nd June (see Appendix 4) was only slightly more accurate. This is to be expected, due to the nature of unforeseen challenges.

2.4 Radio Receiver

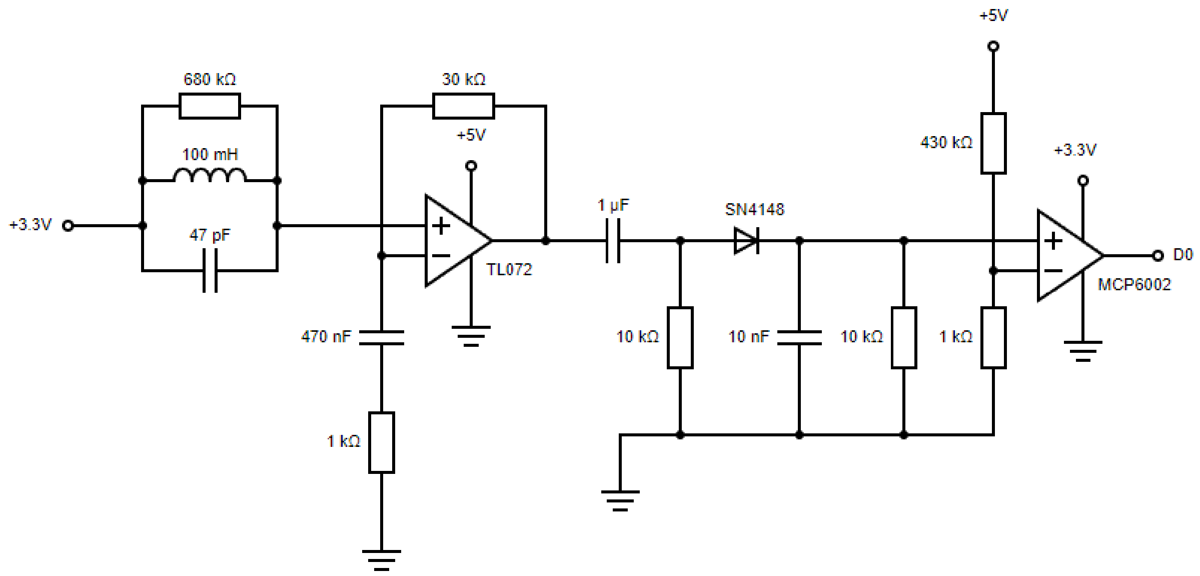


Figure 2.4.1: circuit diagram for final design of radio receiver

The rover must have a radio receiver capable of converting 61kHz carrier wave two-level amplitude shift-keyed signals into a 3.3V binary signal to input to the RX pin of the microcontroller. The final design consists of 4 submodules: an antenna, an amplifier, a demodulator, and a comparator. *Figure 2.4.1* above shows the circuit diagram for this design. The associated software reads ASCII characters from a 600 bits per second UART signal on the RX pin and returns the name of the alien. The receiver has an effective range of 7cm from the edge of the base of an alien, in all directions. The radio receiver was tested 50 times using the random name function on the alien simulator, with the software result compared to the oscilloscope UART reading. The receiver gave the correct name every time.

2.4.1 Antenna

The antenna must convert a 61kHz radio wave into a corresponding electrical signal. A simple, yet effective, and very commonly used type of antenna is a linear half-wavelength dipole (4). This is effectively a straight wire of length equal to half the wavelength of the carrier wave. For a 61kHz radio wave, this is 2460m, which is infeasible. Instead, a loop antenna may be used, with a total length equal to its diameter. Two different loop antennas, shown in Appendix 5, were tested with tuned parallel RLC circuits to adjust the resonant frequency. These tests showed that both have similar range, but also revealed that the inductor in the LC resonator was acting as the antenna, and not the custom-made coils. Various inductors were tested as antennas. For each inductor, the parallel capacitor which resulted in the greatest signal strength at a distance of10mm from the base of an alien was used. A 680kΩ resistor was used in parallel with the LC resonator, as this was found to increase signal clarity. The results of these tests are shown in *Table 2.4.1* on the next page.

Inductance / mH	Capacitance / pF	Signal amplitude / mV
1.0	10000	4.0
3.3	560	21.6
4.7	390	16.5
10	390	18.6
22	220	59.2
33	180	103.4
47	100	141.4
100	47	198.6
100 with ‘curled’ antenna	47	187.7
100 with ‘loop’ antenna	22000	5.2

Table 2.4.1: signal strength for various antennas

From *Table 2.4.1*, we observe that a 100mH inductor has the largest signal amplitude and should be paired with a 47pF capacitor. This was used in the final design.

2.4.2 Amplifier

Initial prototypes attempted to use op-amps to form a precision half-wave rectifier for the demodulator, but this proved difficult due to the relatively low slew rate of freely available op-amps, and requirement of many designs for a dual rail voltage supply. Instead, an amplifier was used to increase the signal voltage before being input to the demodulator. Due to the low slew rate of freely available op-amps, transistor-based circuits were investigated. The circuit which performed best out of these is shown below in *Figure 2.4.2*.

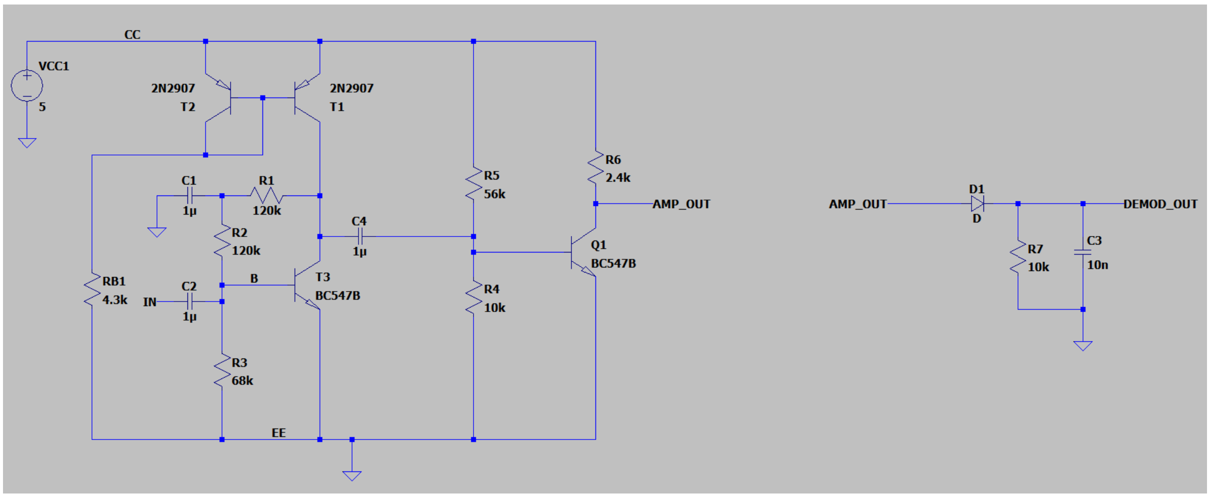


Figure 2.4.2: transistor amplifier and envelope detector circuit

The circuit above is a dual-stage common-emitter amplifier with an active load, coupled into an envelope detector. Although it worked well in simulations (see Appendix 6), with a gain of over 2000, it did not significantly amplify the signal when the physical circuit was built and tested. This is because the current mirror used to form the active load is ineffective without matched transistors, and the circuit has a low enough input impedance to significantly reduce the amplitude of the antenna signal.

Instead, a non-inverting amplifier with a gain of 30 was used, based on a TL072 op-amp. This is because the TL072 is the freely available op-amp with the highest slew rate. The antenna was biased at 3.3V to allow for 1.7V of headroom with 0V and 5V supply rails. More headroom would not increase the range, because the output is input to a comparator after being demodulated. When tested, this circuit had a small signal gain of 30, although increasing the resistance between the non-inverting and output terminals of the op-amp did not further increase this gain. Using an op-amp with a higher slew rate may allow for increased amplification, which may increase the range of the radio receiver. This was not done because the radio receiver needed to have a range slightly shorter than the magnetic field sensor for the automatic alien detection software to function.

2.4.3 Demodulator

The amplifier output is capacitively coupled into the demodulator, so that a simple envelope detector can be used. The envelope detector consists of a diode and RC low-pass filter (5). This filters out the high frequency carrier wave, leaving only the positive part of the modulating wave. The final demodulator design worked well.

2.4.4 Comparator

To convert the demodulated signal into a 3.3V binary signal, a comparator was used on the output of the envelope detector. The threshold voltage was set at 11.6mV, which is sufficiently high to avoid being triggered by noise, and low enough for the receiver to have good range. An MCP6002 op-amp was used instead of the TL072. This was because it was found to work better for a comparator in testing and allowed for the comparator to have an output of 3.3V.

2.4.5 Software

```
void readCharacter(){
  static char temp_name[3];
  char current_char = Serial1.read();

  if (current_char == '#'){
    UART_counter = 1;
  }

  //checks whether input is a letter
  else if ( (current_char >= 65 && current_char <= 90) || (current_char >= 97 && current_char <= 122) ){
    if (UART_counter > 0){
      temp_name[UART_counter - 1] = current_char;
      UART_counter++;
    }

    if (UART_counter == 4){
      current_name = "";
      for (int i = 0; i < 3; i++){
        current_name.push_back(temp_name[i]);
        UART_counter = 0;
      }

    }
  }

  //resets counter if invalid character is read
  else {
    UART_counter = 0;
  }
}
```

Figure 2.4.4: program to determine name

Figure 2.4.4 above shows the software run on the microcontroller to determine the name of an alien. This function is called whenever a serial input character is available. It determines where the name should start using the '#' symbol, then forms the name from consecutive characters. If an invalid character is read, the name detection is reset. The program stores the name in a global string variable called current_name.

2.5 Infrared Sensor

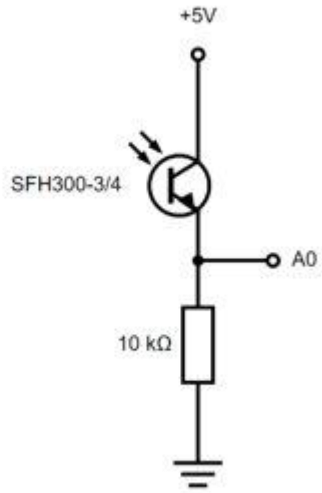


Figure 2.5.1: final design for infrared sensor circuit

The age of the alien is determined by finding the period of infrared pulses it emits. The final design uses very simple hardware consisting of a single phototransistor and resistor. The software polls the analogue pin in the circuit every 100µs for a period of 30ms, then uses the data to determine the period of the pulse. This circuit gives accurate readings at a range of 17cm, although only within a field of view angle of 20 degrees, so the phototransistor must be aimed at the alien. The infrared sensor was tested 50 times using random pulse function on the alien simulator and comparing the results to period values calculated by an oscilloscope. The age was correct every time.

2.5.1 Hardware

Early prototypes for the infrared sensor involved much more complicated hardware. The first used a band-pass filter on the output of a phototransistor to filter out signals outside the expected range of 135Hz to 1000Hz. However, this did not work as there was significant background noise, and the reactive components in the filter produced additional noise.

Another prototype involved using a Schmitt trigger on the output of the phototransistor. However, this worked poorly due to the IC being used having a relatively high, unmodifiable threshold voltage. This approach would likely still have had poor performance if a Schmitt trigger with a different threshold voltage were used, because the average voltage on the phototransistor output varies with lighting conditions.

The final design, consisting of a single phototransistor and resistor, relies heavily on software processing, but using only two components has several advantages. The circuit is smaller, lighter, cheaper, and easier to assemble and test than more complicated alternatives. Testing was done to determine the optimal resistor value for the circuit. The results of this are shown in *Table 2.5.1* on the next page.

Resistance/ kΩ	Range / cm	Noise
3.0	7	Negligible
5.6	8	Minimal
7.5	10	Minimal
10.0	17	Minimal
15.0	20	Moderate
20.0	25	Severe

Table 2.5.1: results of using different resistance values

In the table above, noise is a measure of the consistency of results at a short range of 5cm. For minimal or negligible noise, more than 99% of results are from the software give an age, which is important for reliability. For moderate or severe noise, at least 25% of the results do not give an age. For any level of noise, no erroneous ages were given. The 10kΩ resistor was chosen for the circuit because it provided the longest range of the resistors which had excellent short-range consistency of results.

2.5.2 Software

The first approach attempted for the software was to take voltage readings every 100µs for 30ms and determine the age using these readings. *Figure 2.5.2* below shows how this was accomplished.

```
//constructing list of pulse periods
for (int i = 1; i < 300; i++){
  if (voltage_readings[i] > voltage_readings[i - 1] + 20 && current_period > 7){
    if (current_period < 100){
      period_lengths.push_back(current_period);
    }
    current_period = 0;
  }
  current_period++;
}
```

Figure 2.5.2: code snippet showing construction of list of pulse periods

Readings are taken every 100µs because detectable pulse widths are just over 100µs using the selected hardware. After 300 voltage readings have been taken across 30ms, a function is called which uses these readings to calculate the time between each pulse, part of which is shown above. If a voltage reading is significantly greater than the previous reading, it is treated as a spike. The theoretical range of possible periods should be between 10 and 74 units in the program above. The program allows periods between 8 and 99 units to allow room for error, making troubleshooting easier with extreme values. Once the periods have been determined, they are sorted using a quicksort algorithm, and the period is confirmed if the median 3 are all the same. A complete version of the code in *Figure 2.5.2* can be found in the final design microcontroller software (9).

The biggest flaw with this software approach is that it uses the delay function in the loop, which prevents the microcontroller from handling other tasks for at least 30ms. A solution to this problem is shown in *Figure 2.5.3* on the next page.

```
void loop() {

  current_IR_reading = analogRead(A0);

  if (current_IR_reading > previous_IR_reading + 20){
    current_time = micros();

    if (current_time - previous_time > 949){

      recent_periods[recent_periods_counter] = roundHundred(current_time - previous_time);

      previous_time = current_time;
      recent_periods_counter = (recent_periods_counter + 1) % 5;
      GetAge(recent_periods);

    }
  }

  previous_IR_reading = current_IR_reading;

}
```

Figure 2.5.3: loop for real-time period calculator

Instead of calculating the period from an array of samples, this program compares each reading to the previous reading immediately and records the period as the difference between the times of each spike. It then returns an age the five most recent periods calculated are all the same. This approach allows for other functions to be run in the loop, as it does not require stalling the processor for 30ms.

For the final design, the software approach based on sampling was used. This is because every loop iteration, a server handle is required to get information about motor control from the website. This handle takes 1ms, and the second approach only works if the loop can iterate at least every 100µs.

2.6 Magnetic Field Sensor

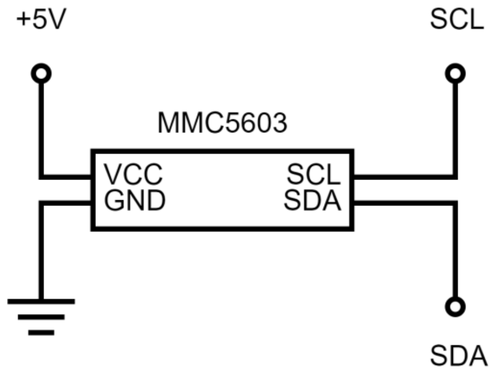


Figure 2.6.1: final design for magnetic field sensor circuit

For each alien, the rover must determine whether there is a magnet underneath the highest point, and if so, in which direction it points. The final design consists of a single Adafruit MMC5603 triple-axis magnetometer, connected to the microcontroller over the I2C bus on the SCL and SDA pins. The magnetometer takes readings of the magnetic field strength and direction in 3 perpendicular axes. Software is used to determine the presence and field direction of a nearby magnet using only readings from the axis which is perpendicular to the Earth's surface. The other two axes' readings change when the magnetometer is rotated while being kept perpendicular to the Earth's surface, due to the Earth's magnetic field. The final design has a range of 10cm. The magnetic field sensor was tested on 50 consecutive aliens with random field directions (or lack thereof). The result was correct every time.

2.6.1 Early Prototypes

Initial designs focussed on using Hall effect sensors to detect the presence and direction of a magnetic field. See Appendix 7 for information on Hall effect sensors. Using a spinning coil to detect magnetic fields was ruled out due to the complexity of mounting, rotating, and electrically connecting the coil. The first Hall effect sensor ordered and tested was a TO-92. This sensor had both low sensitivity, and discrete outputs, which reduced its effective range. The sensor also had a latching output and would not reset until it was brought near a magnet, and so was not useful for observing aliens without a magnet field.

The second Hall effect sensor ordered and tested was an SS443A unipolar Hall effect sensor. This was tested to determine whether a unipolar Hall effect sensor might have a longer range. Again, the sensor had discrete outputs, which reduced its range to just 11mm, even with 25V across it. Two unipolar Hall effect sensors would also be required to determine the direction of a magnetic field. *Table 2.6.1* on the next page shows the results of testing the SS443A unipolar Hall effect sensor.

Supply voltage / V	Range / mm
3	3
4	7
5	8
6	8
7	8
8	8
10	8
12	8
15	8
20	10
25	11

Table 2.6.1: variation of SS443A range with supply voltage

An arm could be built onto the rover on which to mount a Hall effect sensor, but this would add weight, and require very precise driving, increasing the time taken to survey aliens. The arm would also be difficult to attach securely, and the varying height of the aliens could prove problematic for the very limited range of the Hall effect sensors.

2.6.2 Triple-Axis Magnetometer

To increase the range of the magnetic field sensor, an Adafruit MMC5603 triple-axis magnetometer was ordered from a non-supplier website (see Appendix 8) at a cost of £6.00. This was done because no official suppliers offered magnetometers. The triple-axis magnetometer uses the I2C bus to communicate with the microcontroller over the SDA and SCL pins. This provides accurate readings of magnetic fields to a resolution of 0.1μT, which gives it a much longer useful range than the Hall effect sensors tested. All the data processing was done through software, as shown in *Figure 2.6.2* below.

```
void getMagneticField(){

  sensors_event_t event;
  mmc.getEvent(&event);

  double y_val = event.magnetic.y; //take reading of y-axis magnetic field

  double field_measure = y_val - y_init;

  //use difference between initial value and measured value to determine magnetic field direction
  if (field_measure > 20){
    field_direction = "Down";
  }
  else if (field_measure < -20){
    field_direction = "Up";
  }
  else {
    field_direction = "None";
  }

}
```

Figure 2.6.2: software function to determine magnetic field direction

This program is based off example code from an Adafruit library (6). The Adafruit MMC56x3 library must also be installed for the magnetometer to be used. To determine the magnetic field direction, the software takes a reading of the magnetic field on the y-axis during setup, then compares the current magnetic field on the y-axis to the initial value when the function is called. It then uses the difference between these values to determine the presence and direction of a magnetic field. Only the y-axis was used, as this axis is perpendicular to the Earth's magnetic field, and so does not change when the rover turns. The specific values used as thresholds were found through testing and set to maximise the range, while ensuring results were correct.

2.7 Drive Train and Chassis

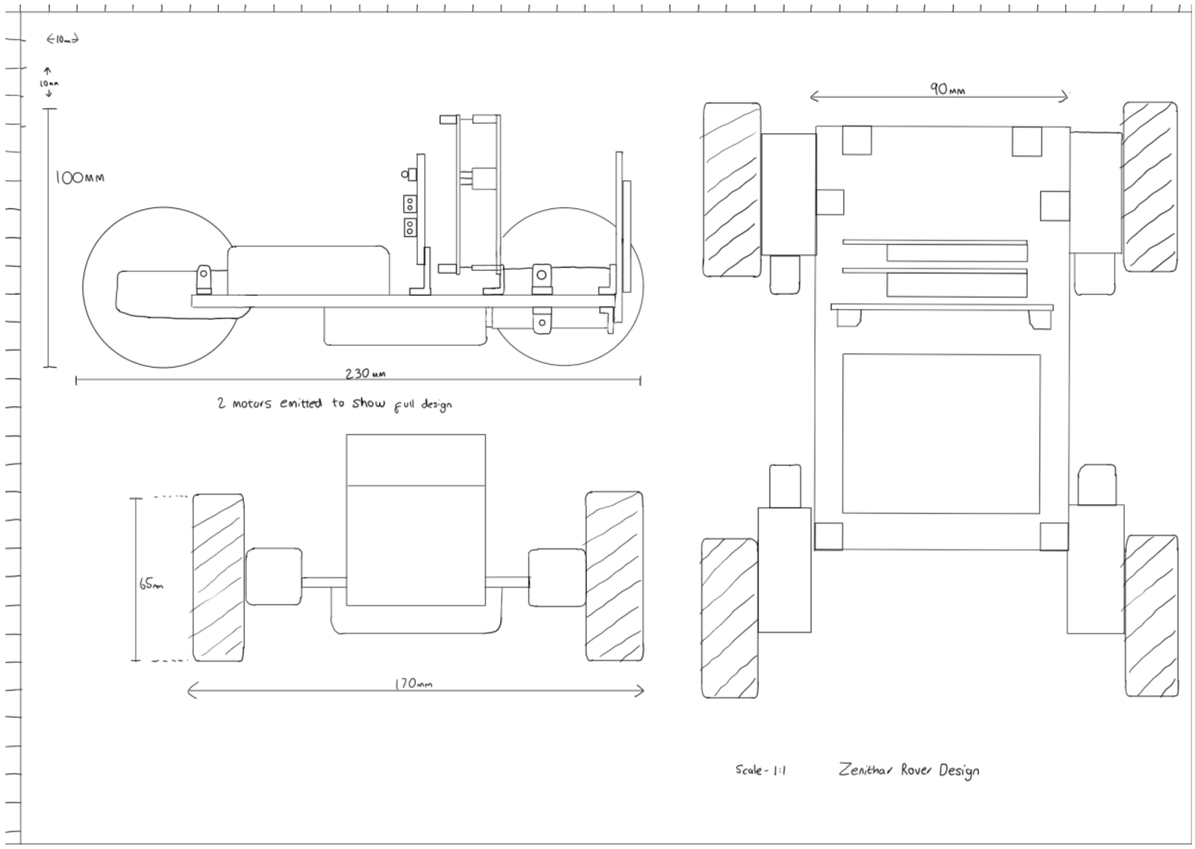


Figure 2.7.1: engineering drawing of final design

To facilitate the movement and structural integrity of the rover, a chassis and drive train were designed. The final rover design uses a single vertical sensor package built on a stripboard, made up of the radio, infrared, and magnetic field sensors. This stripboard is mounted on a chassis using right angle brackets. Also mounted vertically on the chassis are a microcontroller and a motor controller. The drive train uses 4 powered wheels, each driven by a motor. Motors on the same side are connected to the same junction on the motor controller PCB, as they move in the same directions at the same speed for all motor control instructions. To significantly increase the speed and torque of the motors, two 6V battery packs are also mounted on the chassis, used in series to provide 12V across each motor. The front two wheels have zip ties on the tyres to reduce sideways friction, enabling faster turning. The final design moves at 0.368m/s, and turns at a speed of 129°/s.

2.7.1 Drive Train

Initial testing suggested that the EEEBug's drive train, formed of two motors connected to 47mm diameter plastic wheels with tyres, was too slow. There were two ways to increase the speed of the rover: increasing the diameter of the wheels, and increasing the speed of the motors. A set of four plastic wheels with rubber tyres with a diameter of 67mm were ordered from the supplier websites for a total cost of £6.24. These were the wheels with the greatest diameter available within the project budget. When tested, these only slightly increased the speed of the original EEEBug. To overcome any limitations from the torque of the motors, a drive train with four powered wheels was developed. Further testing also revealed that increasing the voltage across the motors from 6V to 12V significantly increased their speed and torque.

A prototype chassis was designed for a four-wheel drive system, powered by two 6V battery packs in series. See Appendix 9 for the CAD model of this prototype chassis. This chassis was laser cut from 3mm acrylic, then the system was assembled and tested. The new drive train had much greater speed and torque than the original.

For the next prototype, one of the motor controller PCBs was removed to save weight, keeping only one to control all 4 motors. This worked because the motors on the same side always rotate in the same direction, at the same speed.

One issue that the drive train suffered from was poor turning due to using 4 wheels with rubber tyres. To reduce the sideways friction while turning, zip ties were added to a pair of tyres. Testing was then done on a range of different wheel configurations, shown in *Table 2.7.1* below.

Front wheels diameter / mm	**Front wheels tyres**	**Rear wheels diameter / mm**	**Rear wheels tyres**	**Linear speed / ms^{-1}**	**Turning speed / $°s^{-1}$**
38	Rubber	38	Rubber	0.280	85.5
38	None	38	Rubber	0.266	82.4
52	Rubber	52	Rubber	0.372	105.0
52	None	52	Rubber	0.349	123.7
38	Rubber	52	Rubber	0.304	91.6
52	Rubber with zip ties	52	Rubber	0.368	128.6
52	Rubber	52	Rubber with zip ties	0.370	138.5
52	Rubber with zip ties	52	Rubber with zip ties	0.368	133.3

Table 2.7.1: linear and turning speed for different tyre configurations

The values for wheel diameter in the table above are for the plastic part of the wheel. Adding rubber tyres increased the diameter of the 38mm wheels to 47mm, and diameter of the 52mm wheels to 67mm. The combination selected for the final design was 52mm wheels with rubber tyres and zip ties for the front wheels, and 52mm with rubber tyres and no zip ties for the rear wheels. This is because it had both good linear and rotational speed, and pivoted around the back of the rover, which is preferrable for turning away from aliens quickly.

2.7.2 Chassis

To accommodate the four-wheel drive train and additional batteries, a custom chassis had to be designed. A 600x300mm sheet of 3mm acrylic was purchased from official suppliers for £8.07 to laser cut chasses from. This sheet had enough area to test at least 4 different prototypes.

The first prototype chassis was designed around having a sensor package on every side and had screw holes to mount stripboards on all sides. It was also designed to accommodate two 6V batteries on the underside, 4 motors mounted on the sides, 2 horizontal motor controller PCBs, and a horizontal microcontroller. *Figure 2.7.2* below shows this chassis mostly constructed, missing only the sensor packages.

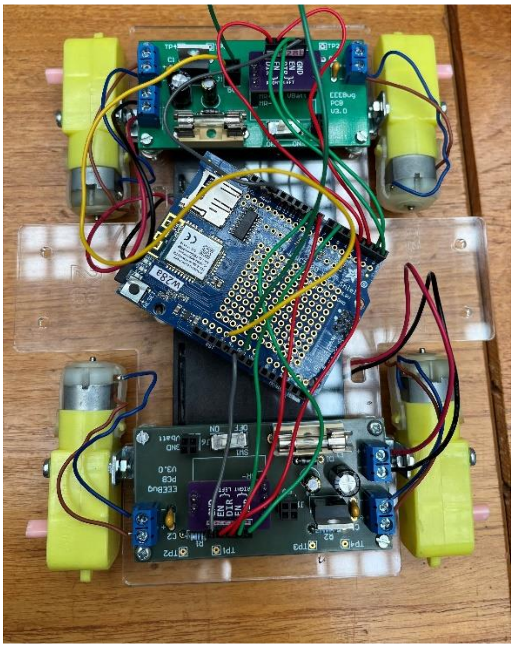


Figure 2.7.2: partial assembly of prototype chassis for drive train testing

This chassis was strong and effective for mounting components, but was only slightly under the weight limit, without the stripboards attached. It triggered the weight sensor when turning due to the high friction caused by the rubber tyres.

The final chassis design (see Appendix 10 for CAD sketch) made several improvements upon the previous prototype, mostly to reduce weight. Developing the magnet field sensor revealed that it would be too expensive to have four sensor packages to the excess cost of additional magnetometers from unofficial suppliers. Therefore, the final chassis was designed to mount only a single stripboard. One of the motor controller PCBs was removed, as it was unnecessary, and both the microcontroller and motor controller PCB were mounted vertically to reduce the area of acrylic required. One of the battery packs was also mounted on the topside of the acrylic to further reduce the area, and the motors were moved as far towards the centre as possible. *Figure 2.7.3* on the next page shows the final chassis with all the components except the sensor package installed.

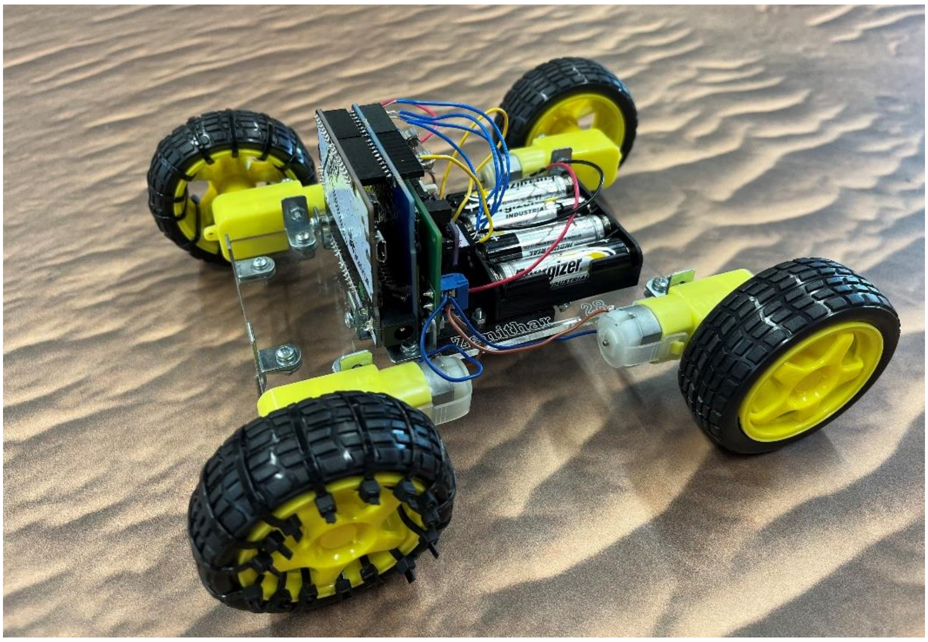


Figure 2.7.3: final chassis assembly without sensors

The final chassis was significantly under the weight limit when stationary, but still triggered the weight sensor when turning due to the reaction of the weight sensor to quick changes in force. The addition of zip ties also caused the weight sensor to trip when the rover was driving straight, so the rover must stop at each quiet alien. Full system design determined that this was a requirement anyways for all aliens, for the infrared sensor to work.

2.8 Website and Motor Control

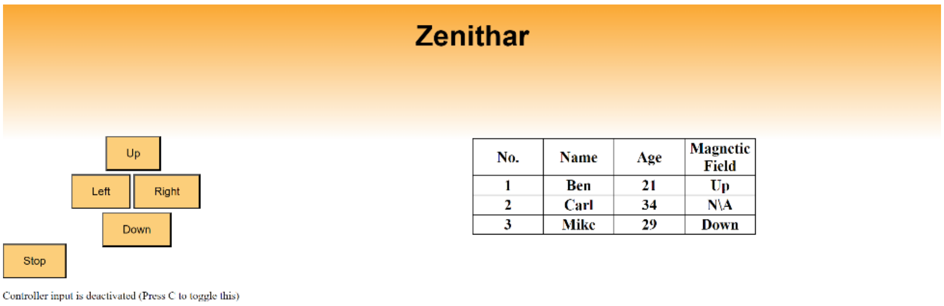


Figure 2.8.1: final website design

A website or app is required to control the rover and display alien characteristics. The final design uses a website, due to the greater ease with which the code can be written and tested. This website has a user interface for controlling the robot through clicking or tapping, and a table which displays the data for each detected alien in an easily readable format. The website also has compatibility with keyboard and game controller inputs to allow for more precise and intuitive control of the rover's movement. The website is hosted externally, from a user's device, to reduce the processing done by the microcontroller, and overcome the 1500 byte limit for data sent by the microcontroller.

2.8.1 Website Hosting and Data Display

Initially, hosting the website from the microcontroller was attempted, but this was found to be infeasible due to the 1500 byte limit on data sent wirelessly by the microcontroller. Therefore, it was decided to host the website externally from a laptop. To do this, the IP address of the microcontroller had to be found and included in the webpage's HTTP requests.

Another early issue was a browser safety feature called CORS, which blocks any POST requests sent to an external sever on the same network. This was resolved by using the Firefox web browser with an extension to disable this.

The first functionality built into the website was the ability to receive and display data from the microcontroller. *Figure 2.8.2* below shows how the data is received and displayed.

```
xhttp.onreadystatechange = function () {
    if (this.readyState == 4 && this.status == 200) {
        var data = this.responseText.split(",");
        var table = document.getElementById("Data");
        var row = table.insertRow(table.rows.length);
        var cell = row.insertCell(0);
        cell.innerHTML = table.rows.length - 1;
        for (let i = 0; i < data.length; i++) {
            cell = row.insertCell(i + 1);
            cell.innerHTML = data[i];
        }
    }
};
```

Figure 2.8.2: code to receive and display data on website

The data is sent to the website as a string, with name, age, and magnetic field direction separated by commas. The data is split by the comma then added dynamically to a table (7). A separate function run on the website is used to determine whether an alien is new, and its data is only added to the table if it is.

2.8.2 User Inputs and Motor Control

To control the motors, the website sends an HTTP request (8) to the microcontroller, following a path. Based on the path the microcontroller receives, it runs a different function. These functions control the motors.

To implement variable speed of the motors, a POST request is required, as POST requests are sent via the path, but also contain data. The requests are generated by the website, then interpreted by the microcontroller as shown in *Figure 2.8.3* below.

```
void forwards() {
  if (server.method() == HTTP_POST) {
    speed = int((server.arg("plain")).toFloat() * 255);
  }
  Serial.println("Forward");
  Serial.println(speed);
  analogWrite(0, speed);
  digitalWrite(1, 1);  //M1 DIR

  analogWrite(2, speed);  //M2 EN
  digitalWrite(3, 1);     //M2 DIR
}
```

Figure 2.8.3: code to interpret POST requests

The write statements are to pins that control the direction and speed of the motors. The HTML page also takes in keyboard input, using the WASD keys to control the rover. The code in *Figure 2.8.4* below shows how keyboard inputs are read.

```
document.addEventListener("keydown", keyDownTextField, false
function keyDownTextField(e) {
    var keyCode = e.keyCode;
    if (moving == false) {
        if (keyCode == 87) {
            up(1);
            moving = true;
        }
```

Figure 2.8.4 code run on website to read keyboard inputs

Game controller compatibility was also desirable, to allow for more precise control of speed and direction using analogue inputs. HTML has a built-in library to interface with game pads, which was used to read and send game controller inputs to the microcontroller. The analogue sticks gave float values ranging from 1 to -1. *Figure 2.8.5* on the next page shows the code required to interpret the inputs.

```
function movement(x,y){
    if (Math.abs(x) < axisTollerance) x = 0;
    if (Math.abs(y) < axisTollerance) y = 0;
    console.log("X : " + x);
    console.log("Y : " + y);
    xhttp.open("POST",address+"/movement");
    xhttp.setRequestHeader("Content-type","text/plain");
    var output = (x *255) + "," + (y*255);
    xhttp.send(output);
}
```

Figure 2.8.5: code for interpreting game controller inputs

The code above shows how the gamepad float can be converted into an analogue PWM signal, by simply multiplying by 255, since PWM values are in the range of 0 to 255. *Figure 2.8.6* below shows how these values can be used by the microcontroller to determine the required motor control signals.

```
if (server.method() == HTTP_POST){
  dataRecieved = server.arg("plain");
  for (int i = 0; i < dataRecieved.length(); i++){
    if (dataRecieved[i] == ','){
      upSpeed = dataRecieved.substring(i+1).toFloat();
      sideSpeed = dataRecieved.substring(0,i).toFloat();
    }
  }
  if (upSpeed < 0 && sideSpeed < 0){//North West
    Serial.println("North West");
    analogWrite(0,-sideSpeed);
    digitalWrite(1,1);//Turning left side
    analogWrite(2,-upSpeed);
    digitalWrite(3,1);
  }
  else if (upSpeed < 0 && sideSpeed > 0){//North East
```

Figure 2.8.6: microcontroller code to enable analogue turning

In the function above, the joystick position values are stored in separate variables. These can be used to execute the correct move instruction. There are a total of 8 possible ways in which the rover can be moving with this script, however more can be added for swing turns at different speeds.

When code making use of variable speeds was tested, very little variation in speed on the wheels was observed. The PWM signals also significantly reduced the torque. This meant that implementing more complex swing turning and speed control would not work properly, so the final design only uses 8 possible motion instructions: forwards, backwards, pivot clockwise, pivot anticlockwise, turn forwards to the right, turn forwards to the left, turn backwards to the right, and turn backwards to the left.

The code for determining the movement type was moved to the website, which runs much faster, to reduce input delay for the rover controls.

The latency for all methods of input was tested by analysing slow-motion videos of the controls being used. The results of this testing are shown in *Table 2.8.1* on the next page.

Input type	**Test 1 / s**	**Test 2 / s**	**Test 3 / s**	**Average / s**
Controller	0.73	0.51	0.56	0.60
Keyboard	0.51	0.50	0.41	0.47
Mouse/touch	0.49	0.54	0.45	0.49

Table 2.8.1: input delay for each method of rover control

The input delay for the keyboard and mouse was lower than that for the game controller. The keyboard and game controller provide the most intuitive interfaces, so the keyboard will likely be used for the final demo.

2.9 Full System Design

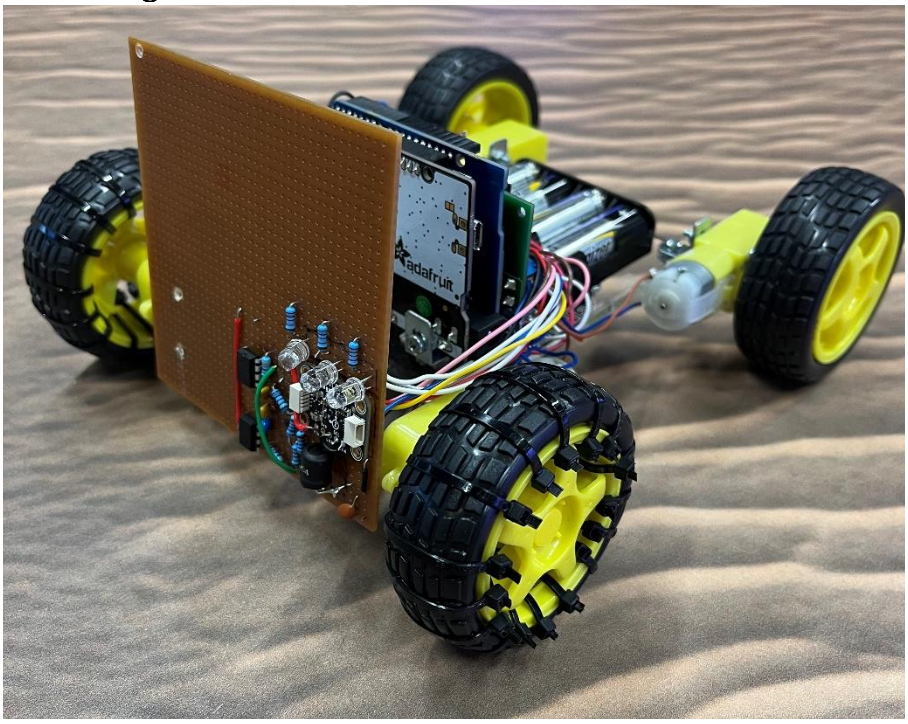


Figure 2.9.1: fully assembled final rover design

To form the final rover, all the submodules were integrated through software and hardware. The sensors are mounted together on a single stripboard at the front of the rover. The sensor package has a single radio receiver, a single magnetometer, and three infrared sensors, each with their own connection to the microcontroller. Using more infrared sensors expands the field of view of the sensor package. There are 4 motors, 2 on each side, connected to a motor controller PCB. This PCB receives 12V power from two 6V battery packs connected in series. The microcontroller receives power from the motor controller PCB and sends signals to control motor speed and direction.

The microcontroller executes a simple software loop, calling the server handle function and RX serial read function. The server handle function checks for data sent from the website corresponding to inputs and translates this data into movement in the motors. The RX serial read functions reads characters over UART, then runs functions to get the magnetic field direction and age once a name is known. It then sends a string containing the alien's characteristics to the website, which determines whether the alien is new, and displays the characteristics of detected aliens in a table. The final design works fully in testing, correctly identifying 20 consecutive aliens.

2.9.1 Hardware Integration

To ensure a strong and durable rover design, each component must be mounted securely to the chassis using screws. This was accounted for fully in the chassis design process.

Each electrical component must also be wired as needed. For the sensors, the output of the radio receiver must be connected to pin 0 (RX) on the microcontroller, the magnetometer pins SCL and SDA must be connected to the pins with the same name on the microcontroller, and the outpost of the infrared sensor must be connected to pins A0, A1 and A2. The DIR and EN pins on the H-bridge of the motor controller PCB must be connected to pins 1, 2, 3 and 4 on the microcontroller. Two opposite battery pack terminals must be connected, and the remaining two must be input to J9 on the motor controller PCB. The positive and negative wires for each motor must be connected to their respective terminals on J7 and J8 of the motor controller. GND and 5V nodes on the motor controller, microcontroller and sensor package must be connected, and the 3.3V node on the sensor package must be connected to 3.3V on the microcontroller. GND must be connected from the motor controller to its H-bridge.

2.9.2 Software Integration

To ensure that the rover interprets movement controls and reads alien characteristics correctly, the software from each submodule had to be integrated.

The first challenge with this was that the infrared sensing software relied on polling the analogue pins every 100μs. Polling less frequently may miss a pulse, since each pulse is 100μs long. However, the server handle function takes 1ms to execute, meaning that the infrared detection cannot run in the same loop as the server handle. Therefore, it was decided to have the main software loop execute a server handle, then call a function which reads UART from the RX pin on the microcontroller.

The server handle function checks for any inputs from the website, then calls different motor control functions depending on the input. The serial read function determines when a full name has been read and calls a function to find the magnetic field direction, as well as a function which polls the analogue pins for 30ms and calculates the age of the alien. If all 3 characteristics are detected, this data is sent to the website, which then displays it if it the same data has not already been displayed. Using these two functions allows for the rover to be controlled with a polling rate of 1kHz, although it will become unresponsive for around 40ms when it detects an alien.

3 Discussion

The final design of our rover performed excellently on all the tests done on each module. Although assembled, testing and debugging of the final rover was not complete by the time of the report submission. Instead, tests were performed on groups of submodules. The rover can detect the name, age, and magnetic field direction simultaneously with a range of 7cm from the radio receiver's inductor. The rover can move at a speed of 0.37m/s and turn at a rate of 130°/s. Rover control has an input delay of 0.5s using a keyboard, and the website can display alien characteristics and interpret rover control signals from mouse, keyboard, and game controller inputs.

Although the rover fulfils all the essential criteria of the product design specification, it does not meet some of the more ambitious goals. The product design specification suggests that the rover should move at 0.5m/s, which is significantly greater than the 0.37m/s achieved. This speed cannot be attained using the chosen motors and wheels. The best way to overcome this barrier would be to buy new motors with a higher rpm. This option was explored, but was deemed to be too expensive, and add significant complication due to the need for additional mounting brackets for the motors. The rover is also slightly larger than the target in the product design specification, with dimensions of 230x170x100mm. The constraint on the size of the robot is the large space taken up by the two 6V battery packs. This issue could be solved by purchasing batteries which have the same output voltage, but a lower volume. Furthermore, the robot remains under the weight limit while stationary, but goes over the dynamic weight limit while moving. This is due to the friction between the tyres and the arena surface, and could only be solved by using smoother wheels. However, this is likely to reduce linear speed.

Despite missing some of the ambitious targets of the product design specification, there are some which the rover exceeds. The rover's sensor range of 7cm is much greater than the targeted 4cm, and the sensor readings are performed within 400ms, as desired. The cost of the entire project was also just £32.25, far less than the budget of £60. Although this required ordering a magnetometer for £6.00 from an unofficial supplier, doing so significantly increased the range of the magnetic field sensor. Based on research into sensors being sold, it is unlikely that a similar result could be achieved using official suppliers.

4 Conclusion

Our rover was designed with the goal of determining the characteristics of all the simulated aliens in the arena in as quick a time as possible. Each alien emits 61kHz carrier frequency amplitude-shift keyed two-level modulation UART signals, encoding their name. They also emit infrared pulses with a period proportional to their age and may have a magnetic field aligned with their body.

The ideal rover would move at over 1.0m/s between aliens and determine their characteristics from as far as 20cm. It would turn at more than 360°/s, stay under the dynamic weight limit when moving, and have 10ms input delay. Although these features are not unachievable, they require significant additional research, testing, and expense. With more powerful motors, more power dense batteries, and a lighter chassis, a 1.0m/s top speed could feasibly be achieved. 20cm sensor range may not be achievable using magnetic sensors due to the inverse square law defining the strength of the magnetic field, but could be done using advanced image processing and electromagnetic sensors. This in turn would require a significantly more powerful processor.

Although constructing an ideal robot is beyond the current abilities of the team, there are some recommendations posed for future work, to improve the rover. The first of these, which will be done immediately, is to finish debugging and testing the full system, so that the rover can be used as intended. Beyond this, an easy to implement improvement would be to use smaller and lighter batteries with the same total voltage, to allow for a smaller and lighter chassis to be designed. A more complicated, but similarly important, improvement would be to replace the current motors with ones with a higher rpm, as this would significantly increase the speed of the rover. However, this would also require significant redesign of the chassis. Another two simple improvements to the rover are replacing the Adafruit Metro M0 Express with a more powerful microcontroller and replacing the front two wheels with omni-wheels, to reduce sideways friction and dynamic weight while turning. We expect that making the suggested changes would enable the rover to fully meet the ambitious goals of the product design specification.

References

1. Ed Stott. *Project Brief: EEERover*. https://github.com/edstott/EEERover/blob/main/doc/brief.md [Accessed 15th June 2023]
2. Ed Stott. *EEE1 Lab Skills*. https://github.com/edstott/EEE1labs/tree/main/skills [Accessed 15th June 2023]
3. Ed Stott. *EEERover Technical Guide*. https://github.com/edstott/EEERover/blob/main/doc/README.md [Accessed 15th June 2023]
4. Balanis C. *Antenna Theory*. 3rd ed. Hoboken, New Jersey: John Wiley & Sons, Inc.; 2005.
5. Leung K. *Amplitude modulation (AM)*. [Lecture] Imperial College London. 9th March 2023.
6. Adafruit. *Adafruit MMC56x3 Magnetometer Library*. https://github.com/adafruit/Adafruit_MMC56x3 [Accessed 15th June 2023]
7. Patel, V. *Dynamically Add/Remove rows in HTML table using JavaScript*. https://www.viralpatel.net/dynamically-add-remove-rows-in-html-table-using-javascript/ [Accessed 14th June 2023]
8. Mozilla. *XMLHttpRequest: send() method*. https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest/send [Accessed 14th June 2023]
9. Zakariyyaa Chachia. *Zenithar Project Source Code*. https://github.com/Zakattack1045/Zenithar [Accessed 15th June 2023]

Appendices

1. **Official supplier websites**
   The official supplier websites are the EEE Department Stores (http://www.ee.ic.ac.uk/storesweb/contents.html), OneCall Farnell(https://uk.farnell.com/), Rapid (https://www.rapidonline.com/), and RS Electronics (https://uk.rs-online.com/web/)

2. **Standards**
   The IEEE/ANSI 315-1975 standards for graphic symbols for circuit diagrams are available for purchase from https://standards.ieee.org/ieee/315/515/. The BSI standards for the safety of electrical and electronic devices are available at https://www.bsigroup.com/en-GB/industries-and-sectors/electrical-and-electronic/.

3. **Initial plan**

Team Member	**15th May**	**22nd May**	**29th May**	**5th June**	**12th June**
Zakariyyaa Chachia	Website development	Motor software and testing	System software integration	Testing	Report
Hassan Choudhury	IR research	IR hardware	IR software	Testing	Report
Joe Elguezabal	IR research	IR hardware	IR software	Testing	Report
Gurjan Singh Samra	Magnetic sensor research	Magnetic sensor hardware	Magnetic sensor software	Testing	Report
Ryan Voecks	Radio research	Radio hardware and software	Chassis construction	Testing	Report
Letong Xu	Website	Chassis design	Chassis construction	Testing	Report

The table above shows the team's initial plan for the project.

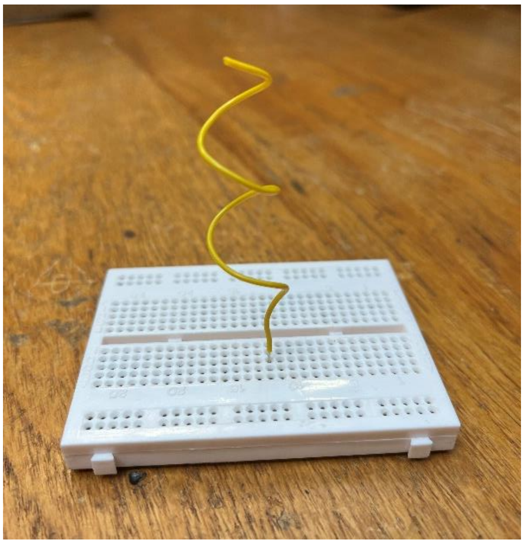
4. Mid-project plan

Team Member	5th June	12th June
Zakariyyaa Chachia	Develop keyboard and game controller inputs for rover control, and improve website GUI	Troubleshoot website software problems and write report
Hassan Choudhury	Develop software to connect sensor data to website, then test and optimise sensors	Practice driving and fix any last-minute problems
Joe Elguezabal	Optimise software for simultaneous single-package sensing and driving	Test and troubleshoot full system and write report
Gurjan Singh Samra	Order ratiometric Hall effect sensor and integrate into sensor package	Test and troubleshoot full system and write report
Ryan Voecks	Develop and implement perfboard sensor package, then develop software for multi-package sensing	Test and troubleshoot sensor system and write report
Letong Xu	Submit chassis file for laser cutting and assemble rover	Assess chassis performance, and write report

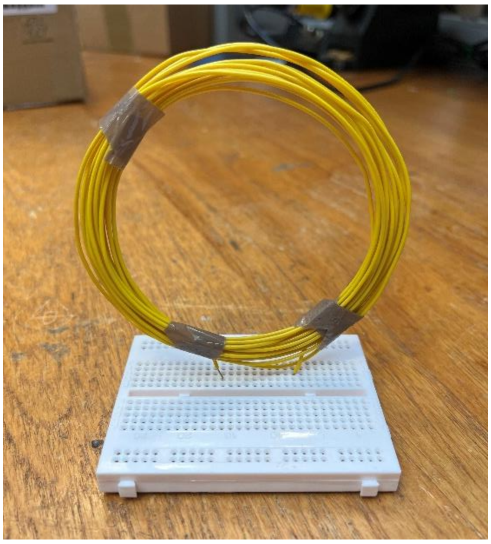
The table above shows the plan for the project made for a presentation on the 2nd of June.

5. Loop antennas

The figure below on the left shows the 'curled' antenna tested, and the figure below on the right shows the 'loop' antenna tested.

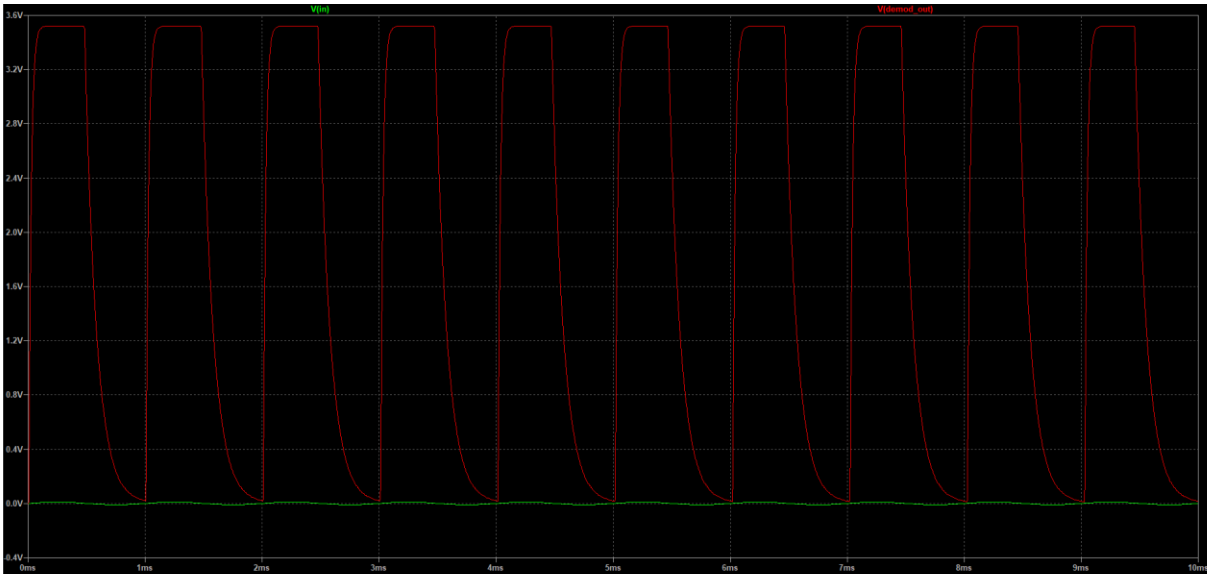


'Curled' antenna



'Loop' antenna

6. **Transistor amplifier and demodulator simulation results**
The figure below shows the results of a simulation run on the dual-stage common emitter amplifier with active load demodulator, using an input with 10mV amplitude.



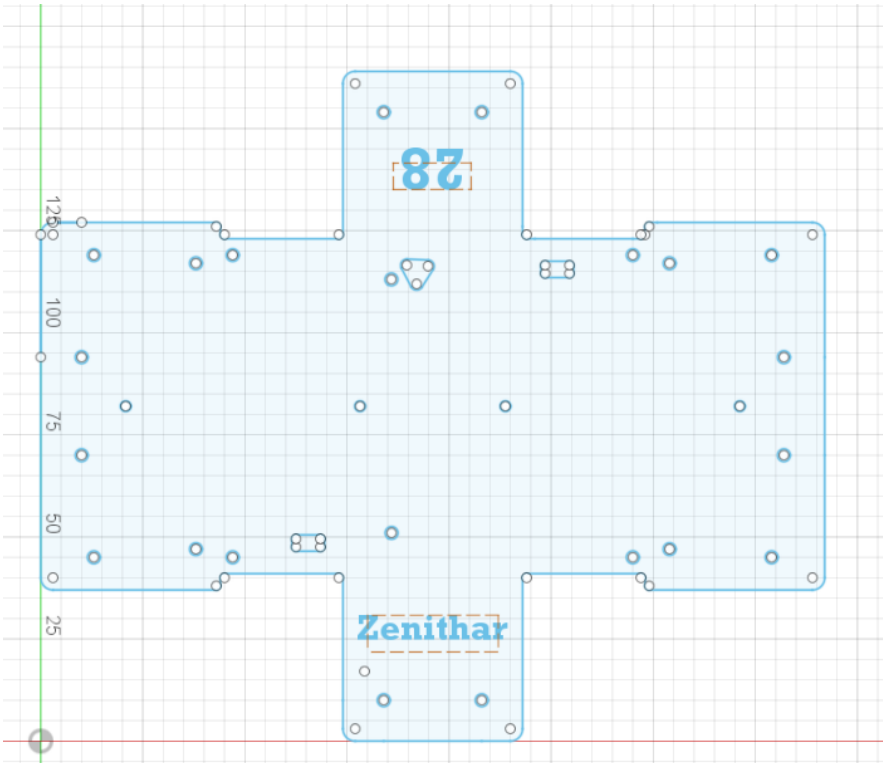
Results of transistor amplifier simulation

7. **Information on Hall effect sensors**
A Hall effect sensor makes use of the forces acting on charged particles in a magnetic field to determine the strength and direction of a magnetic field.
There are three common types of hall effect sensor: unipolar, bipolar, and ratiometric. A unipolar sensor can only detect a magnetic field in one direction, while a bipolar sensor can detect a magnetic field in both directions. Ratiometric sensors have a voltage output which varies with the strength of a magnetic field, while unipolar and bipolar sensors often have discrete outputs.

8. **Ordering a magnetometer**
The Adafruit MMC5603 magnetometer was ordered from Pimoroni at https://shop.pimoroni.com/products/adafruit-wide-range-triple-axis-magnetometer-mlx90393.

9. CAD model for prototype chassis

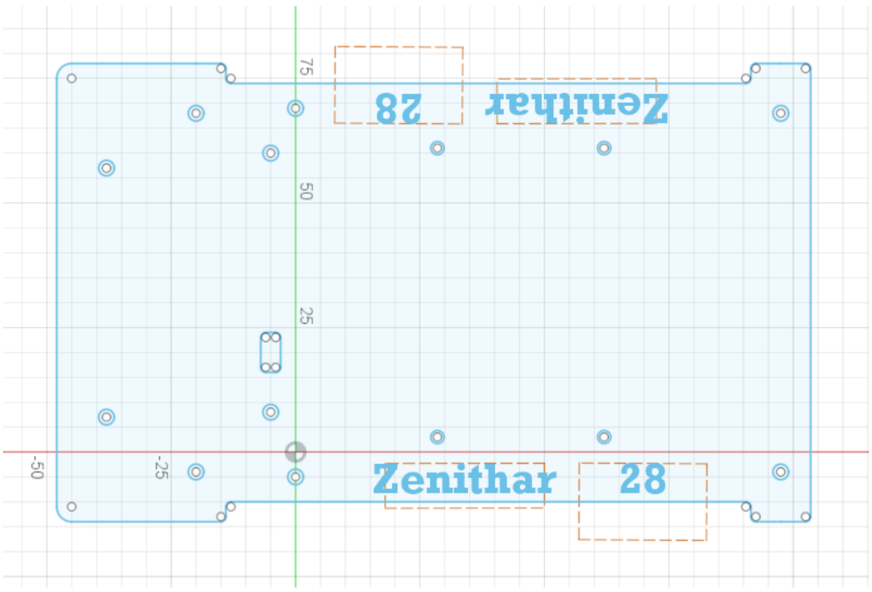
The figure below shows the Fusion 360 sketch for the prototype chassis.



Fusion 360 sketch for first prototype chassis

10. CAD model for final design chassis

The figure below shows the Fusion 360 sketch for the final design chassis.



Fusion 360 sketch for final design chassis