ESP32 powered AI litter collection robot

Table of Contents

Preparation

Project overview

One of the biggest problems in the world today is pollution which includes litter outside and in buildings. To help solve this problem, I am going to create a small autonomous robot capable of detecting and moving litter. This robot will be capable of removing plastic from internal and external environments to prevent damage to the infrastructure and ecosystems of these spaces. To achieve this, I will need to follow the following specifications:

- No 3d printed components can be larger than 220 x 220 x 270 mm as these are the dimensions of the print volume for the 3d printer I will be using. I will use a 3d printer as it is quick and allows for precise, replicable manufacturing of the components in my design.
- The cost of all the components should be around £50 but no more than £100. This will ensure the robot is cheap to manufacture and can be easily replicated.
- The robot will need to include a camera for image recognition which decreases the options for microcontrollers that can be used.
- The microcontroller used also needs plenty of free GPIO pins to connect the various components necessary for the design.

Components

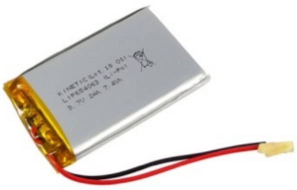
For this project, I have chosen to source all the electrical components from the website Pi hut rather than directly buying from a direct electronics component's seller such as Digi key. This reduces the number of options I have for components to use for the project but will allow me to receive the components quicker and with easier to access documentation.

Microcontroller

I will use an esp32 microcontroller to create this project, rather than a single board computer such as a raspberry pi. This will result in a lower processing power but the positive of a smaller form factor to fit into the chassis and lower power and cooling requirements. However, this also means I will only be able to power the components with 3.3V rather than 5V due to the limited power output of esp32 boards. This will limit my options in terms of components, most notably motors, but should still allow for a successful design.

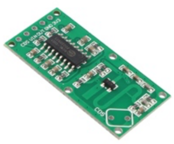
The specific board I have chosen to use is the Xiao esp32 s3 sense. This is due to its small form factor, as well as having a built-in camera which leaves all the GPIO pins free to wire components to, making it ideal for my project. As well as this, it has an inbuilt module and attachment to be powered by a 3.7v lithium polymer battery and recharge the battery. This makes It easier for my project as the battery doesn't have to be swapped out as often.

Battery



I have chosen to use a 3.7v LIPO battery with 1200mAh of capacity. This battery will connect directly to the board, which reduces the need for extra components. It also maintains a small for factor coming in at only 5.3 x 35.5 x 64.5 mm making it ideal for a small form factor chassis.

Distance sensor



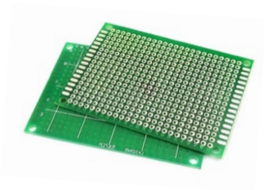
Due to the limited voltage output of the esp32 s3 chip I am using, I cannot use a standard 5v HC-SR04 ultrasonic distance sensor without the use of a voltage converter. Since using a converter can lead to unreliability in the operation of the sensor, I will instead use an ultrasonic distance sensor that uses a RCWL-1601 board instead. This means that it can be powered with voltages between 3 – 5V, making it compatible with my board.

Motors

For the powering of the wheels, I will use 2 TT Motors (which can be powered by 3 – 6V), one for each wheel. Unfortunately, as the battery only outputs 3.7V, and servo motors require an input of 5V, I will use a third TT motor to drive the claw mechanism. This means I will need to use a claw mechanism that can be continuously driven, rather than a pivot-based design.

To drive these motors, I will use 2 HR8833 DFROBOT motor drivers one of which will allow me to power both the wheel motors and the other for just the claw driving motor. This motor driver is much smaller than most which will help limit the weight and size of my project.

Other Components



Other components I will use are:

- Proto board to allow me to easily solder the parts together without the need to design an order a custom PCB.
- Ceramic and electrolytic capacitors to smoothen the signal between the battery and the motor driver as well as between the motor driver and board. This will result in more reliable and consistent operation of the components.

- A simple OLED display to show information relating to the operation of the robot.

- Simple push buttons to allowing changing settings of various parts of the system when a WIFI – connection is not available.

Full components List

Component	Price
Xiao ESP32 S3 Sense	£13.50
3.7V 1200mAh LIPO battery + JST connector	£8.70
RCWL-1601 ultrasonic distance sensor	£3.90
3V TT motor * 3	£7.80
HR8833 Motor Driver * 2	£8.80
Proto board	£4.40
Ceramic and Electrolytic capacitors	£3.80
OLED display	£4.00
Push buttons	£2.20
Total Price	£57.10

Using these components the total price is £57.10 which makes it below the budget of £100 due to the use of the affordable ESP32 motors as well as the use of TT motors rather than servos.

CAD design

I have chosen to use Fusion 360 to design the 3d printed components for this project as it is a software I am familiar with and allows for the parts to be easily exported to be 3d printed.

Claw Mechanism

I chose to first design the claw mechanism as this will heavily influence the size and shape of the final body design to house the motors, circuitry and other components.

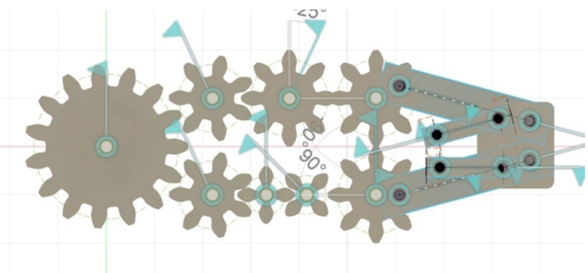
Claw Design Ideas

I started by experimenting with various claw designs to determine the most optimal design bearing in mind the following factors:

- It must be driven by continuous rotation as that is the form of motion delivered by the TT motor I will be using.
- Have a small total size to avoid the size of the robot becoming too large. In attempt to achieve this, I will try to minimise the number of components involved in the design.

Design 1

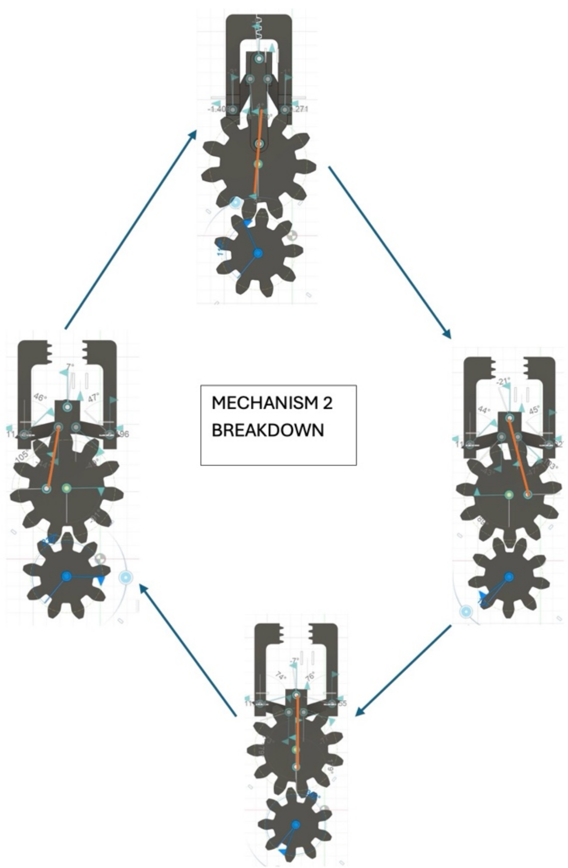
Initially, I tried experimenting with a dual pivot driven claw, like the one showed below. However, this required a lot of components to keep the desired form factor.



Furthermore, the complex geometry made it difficult to simulate, and it would likely get jammed if 3d printed due to the large number of moving parts

Design 2

Due to the difficulty of designs that only use rotary motion, I decided to use a slider-based design instead. To achieve this, I created a mechanism only drives one slider along the y-axis. This slider then connects to a slider for each claw along the x-axis. This results in a grabbing motion shown in the diagram below as well as the video. This design uses significantly less components while maintaining a small form factor which makes it ideal for my design. However, the rotary motion driving the claw means it is only in a tight grip for a small portion of the rotation.



Video Of Mechanism

Chosen claw design

For my design I will use the second design as it is significantly more efficient than the first.

Next, I will design the body which will be heavily influenced by the size and form factor of the chosen claw design.

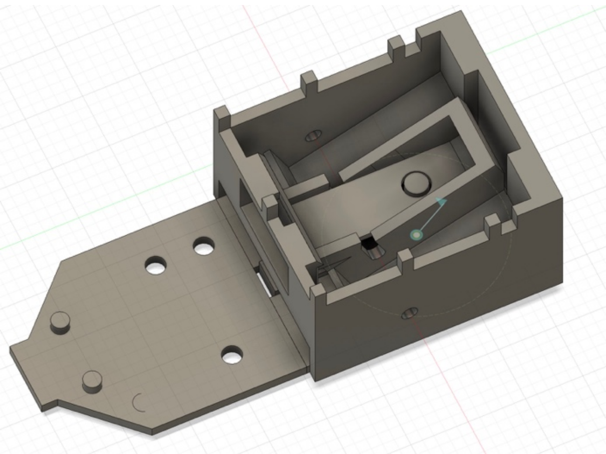
Body design

For the body design to be successful, I will need to follow the following specifications:

- It can be no larger than 220 x 220 x 270 mm as otherwise it won't fit in the print space, I have available.
- It will need space to store the 3 TT motors which are 22.6 * 22.3 * 69.4 mm but the size of the space for each motor will have to be a millimetre or two larger to allow for tolerance when 3d printing.
- I will need to leave an open space that can hold the 50 * 70 mm perf board which the components will be soldered to.

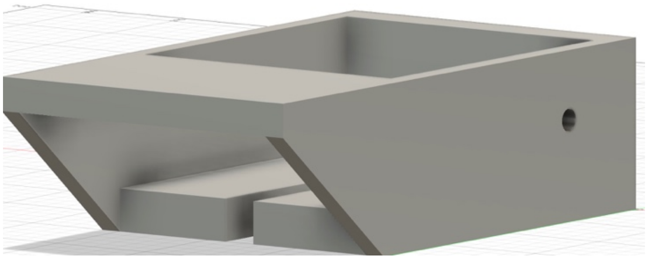
Body design prototyping

Design 1



Initially, I tried to create a design where the motor holders were rotated along the X-axis to minimise the size of the chassis. However, this resulted in the design having a large vertical size, which would increase the cameras distance away from the claw and would also result in the design being difficult to 3d print due to creating an excessive number of overhangs requiring lots of filament being needed to support the overhangs.

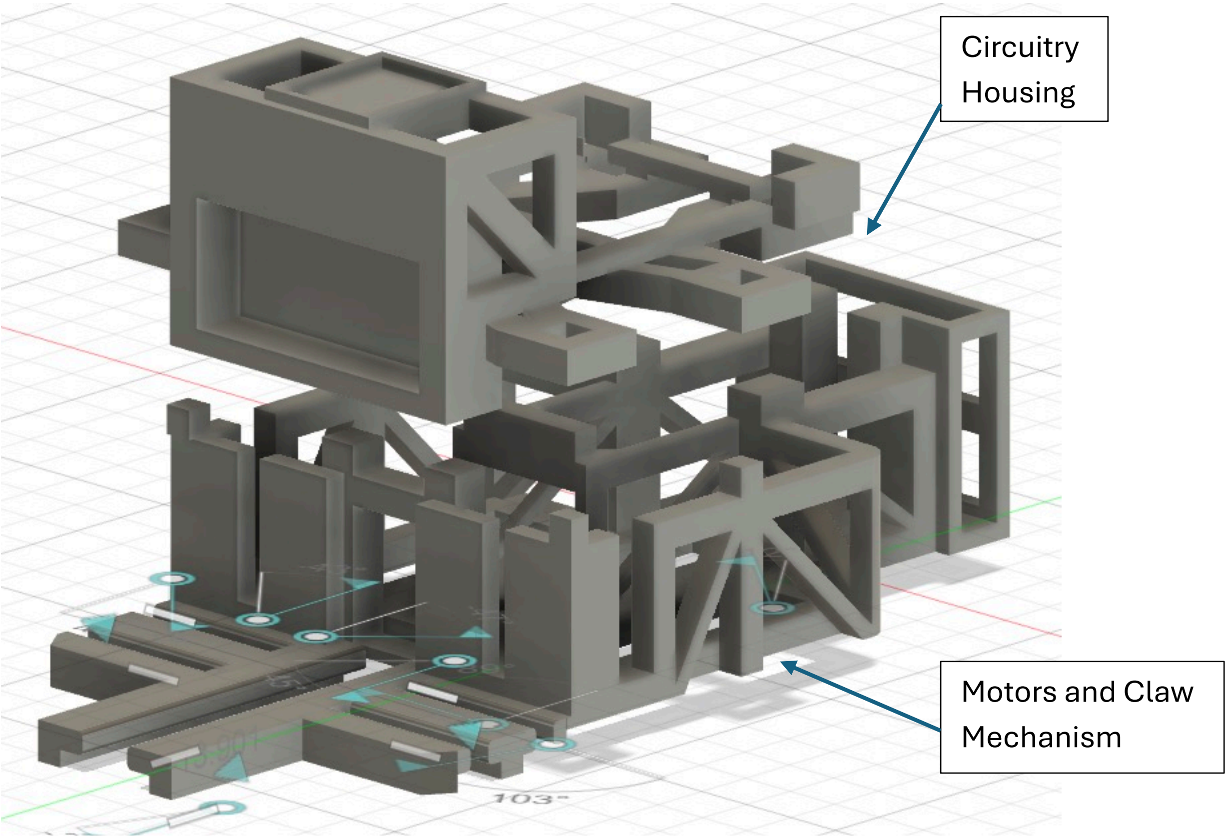
Design 2

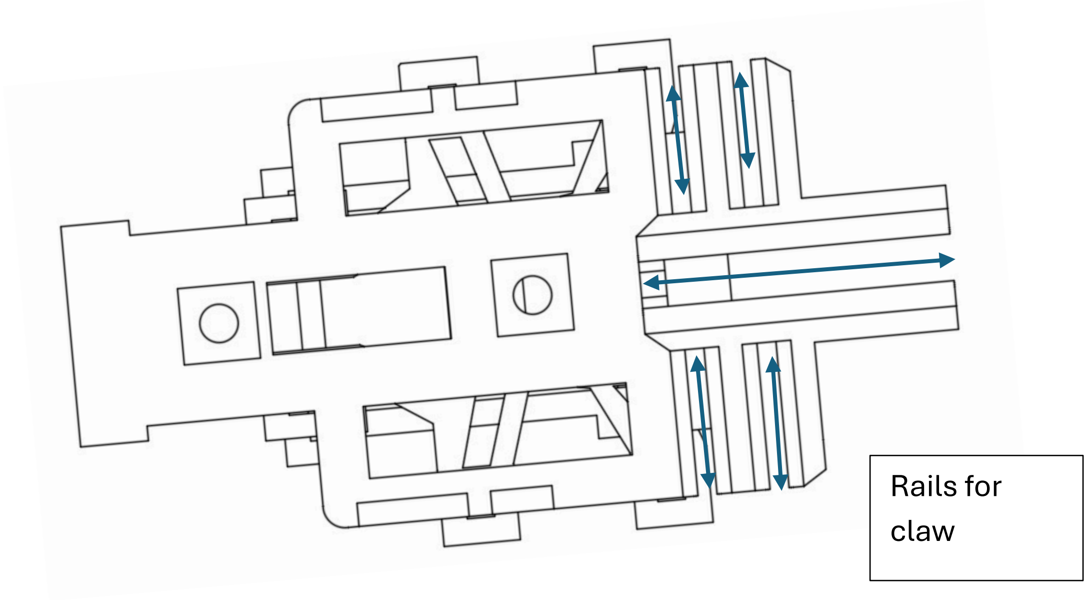


I then tried a design which the motors were held flat and the camera could be held in front of the mechanism. However, this design didn't allow space to store the circuitry.

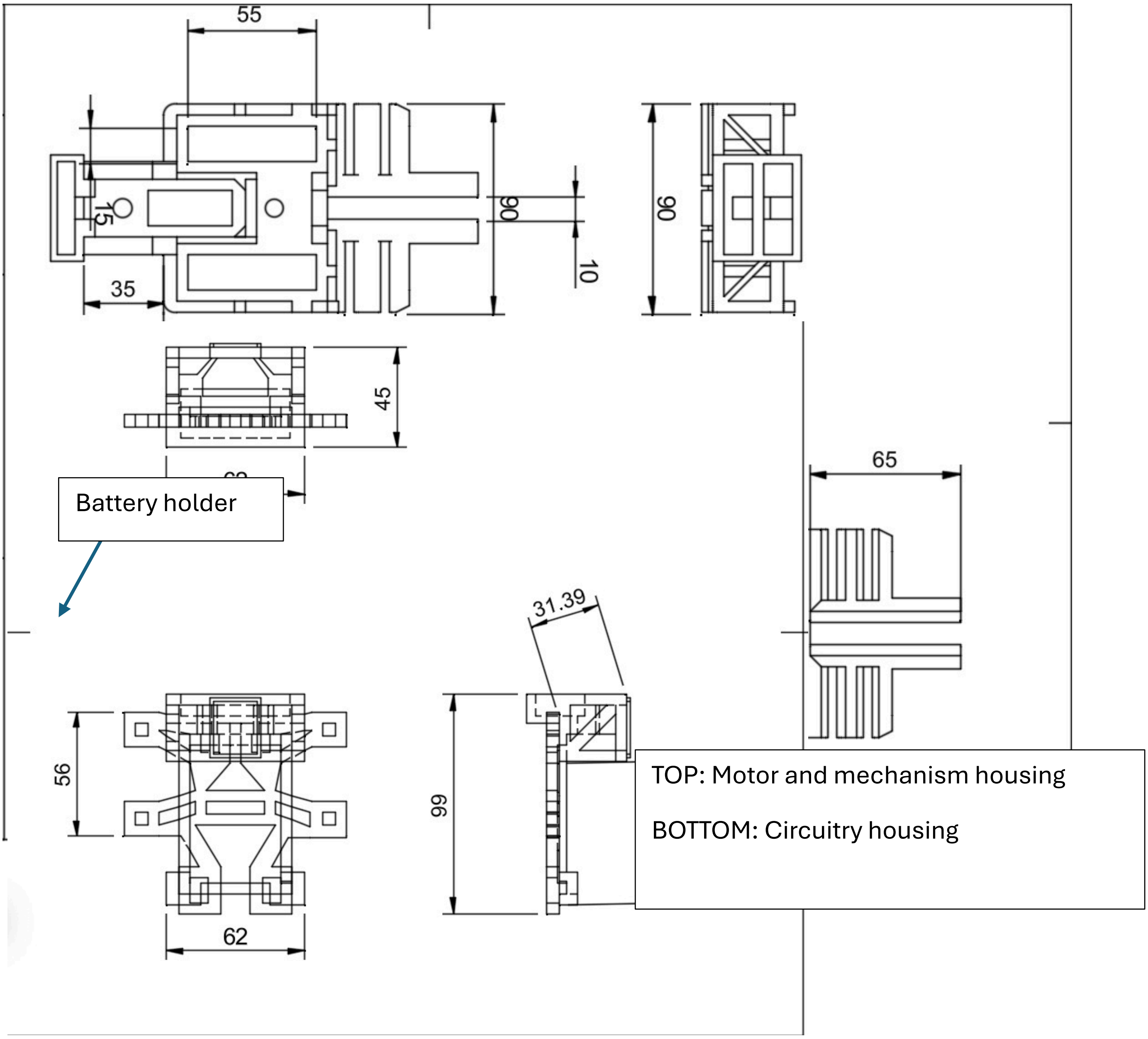
Design 3

For the third and final implementation of the design I split the design into the main body and a separate part to hold the circuitry and ultrasonic distance sensor. I also implemented railings into this design and adjusted the mechanism to fit.



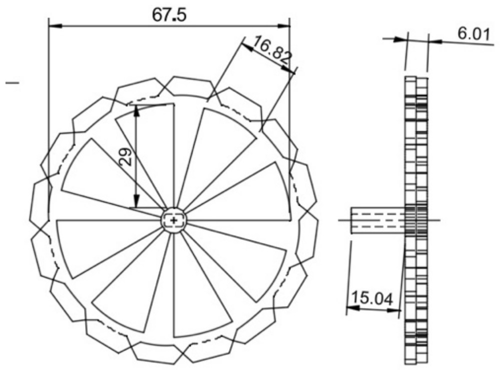


Rails for claw

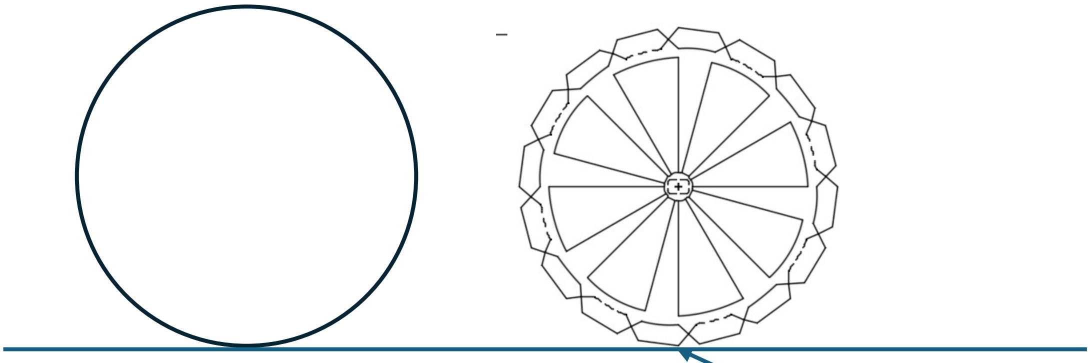


55
15
35
90
10
90
45
65
Battery holder
31.39
56
99
62
TOP: Motor and mechanism housing
BOTTOM: Circuitry housing

Wheel Design

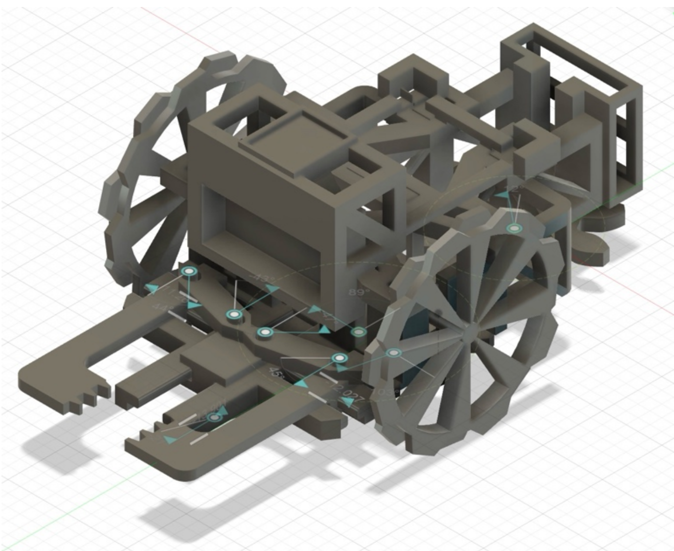


I chose to use a wheel with a ridged circumference in order to increase the friction and so also increase the robots grip and ability to manoeuvre across trickier terrains such as carpet or grass

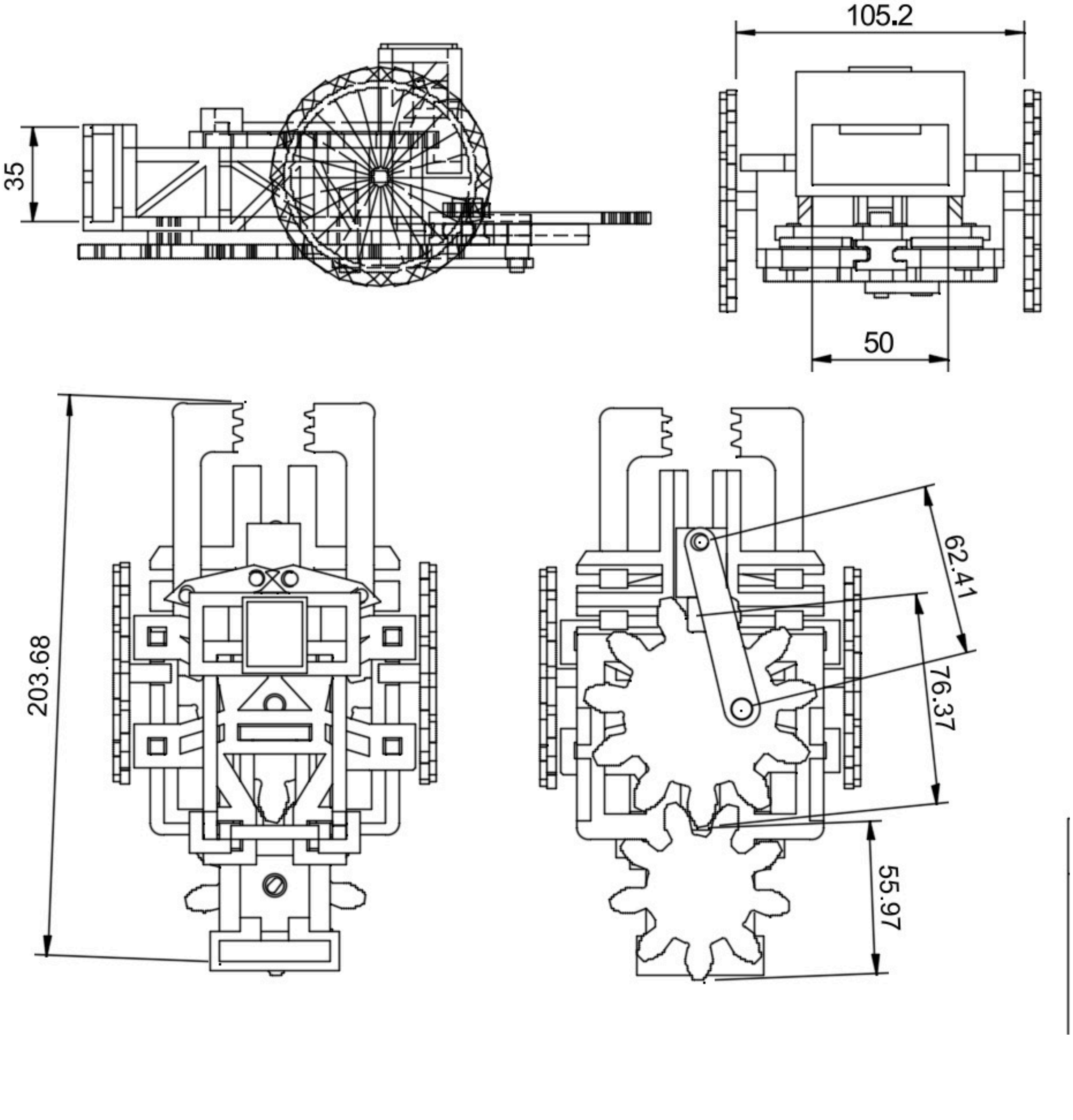


More points of contact than normal wheel = increased grip

Full assembly

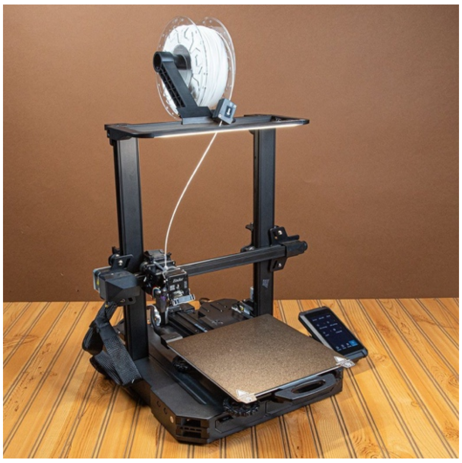


The picture to the left displays the entire assembly of the finished design. The total size is 110 x 70 x 200mm which means it can all be easily printed in the available print space using a minimal amount of filament

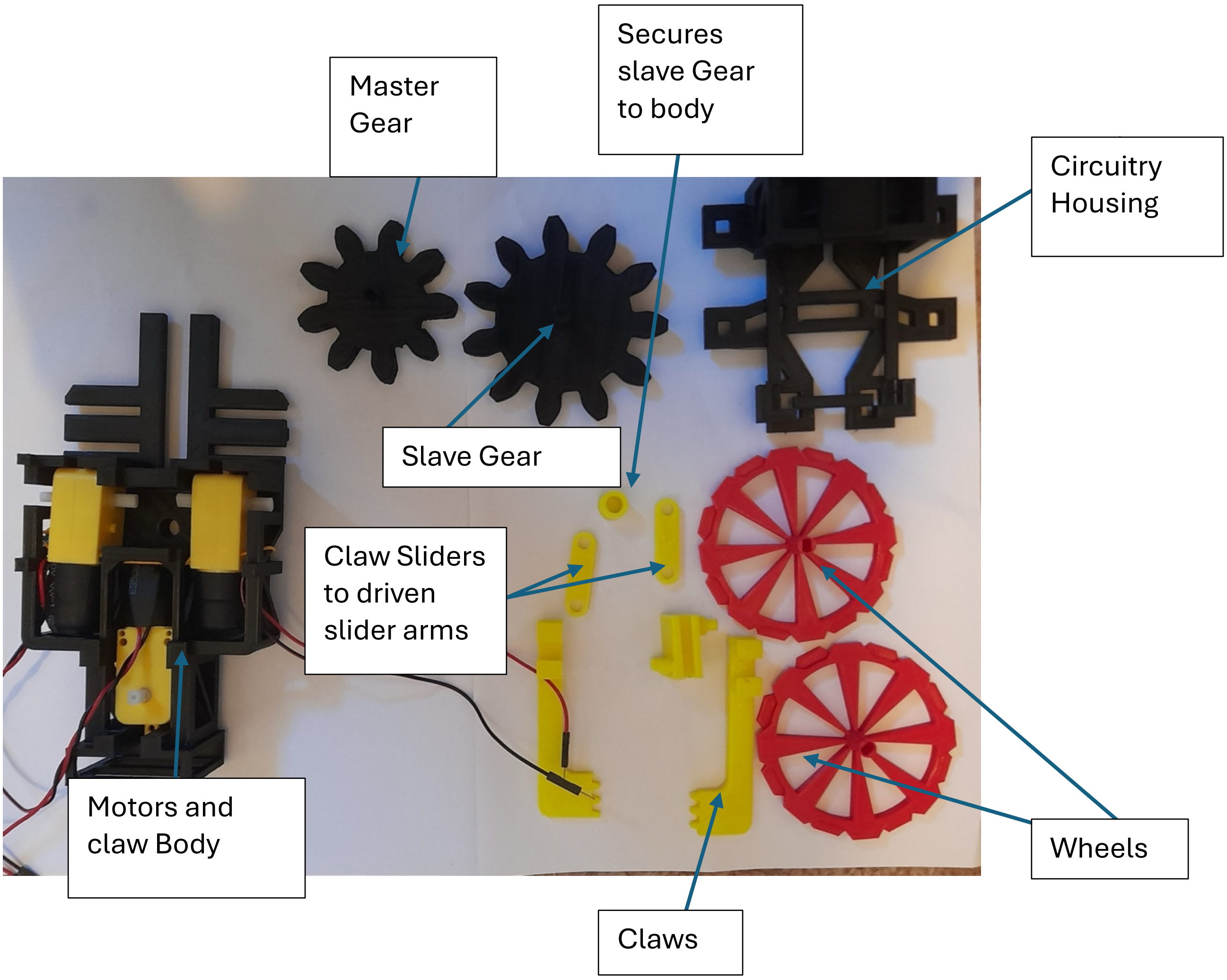


Full Body Drawing

3d printing



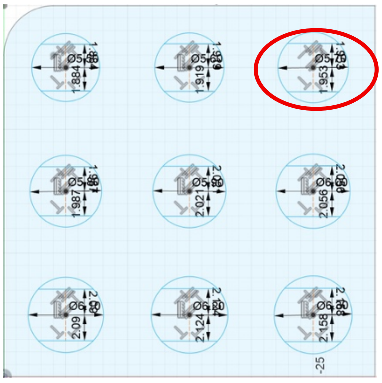
I chose to 3d print the design as it is an easy, quick and precise manufacturing method. The 3d printer I will use is the ender 3S1.



Material

I printed my design using PLA as it is the filament type I had available at the time but if I had more time, I would have printed the design using PETG as it offers a higher toughness which is desirable in my robot due to the mechanical parts which operate the claw mechanism.

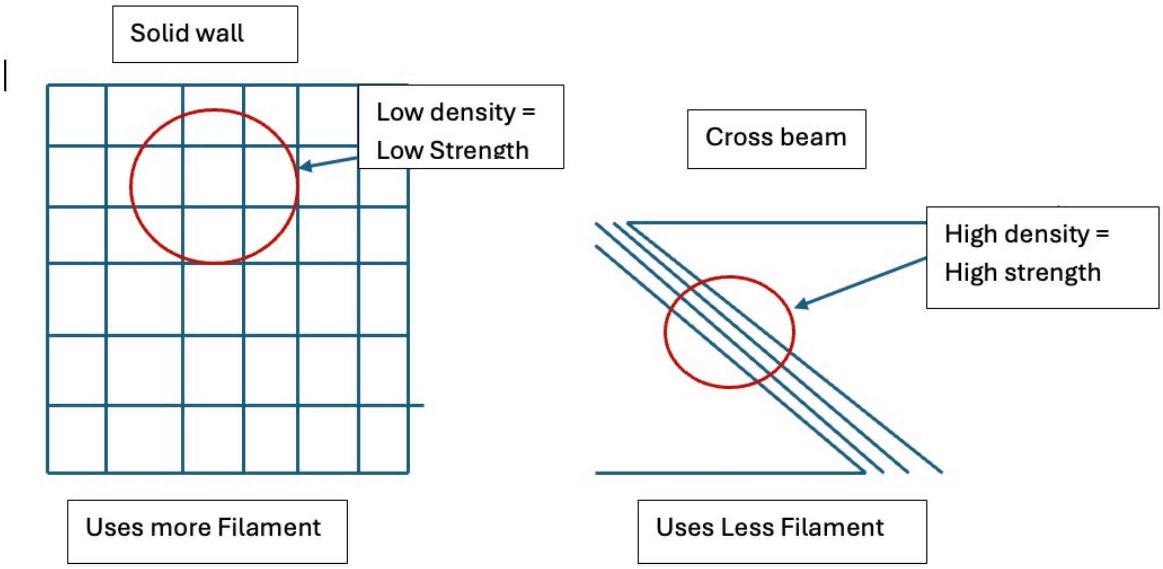
Tolerance Testing



I first printed a 30 x 30 mm square with holes of different sizes in order to find the correct size to print the holes on the master gear and wheel for attachment to the TT motor shaft.

From this, I determined I would need to use a hole with diameter of 5.7mm and a width of 3.9mm.

Cross Beams

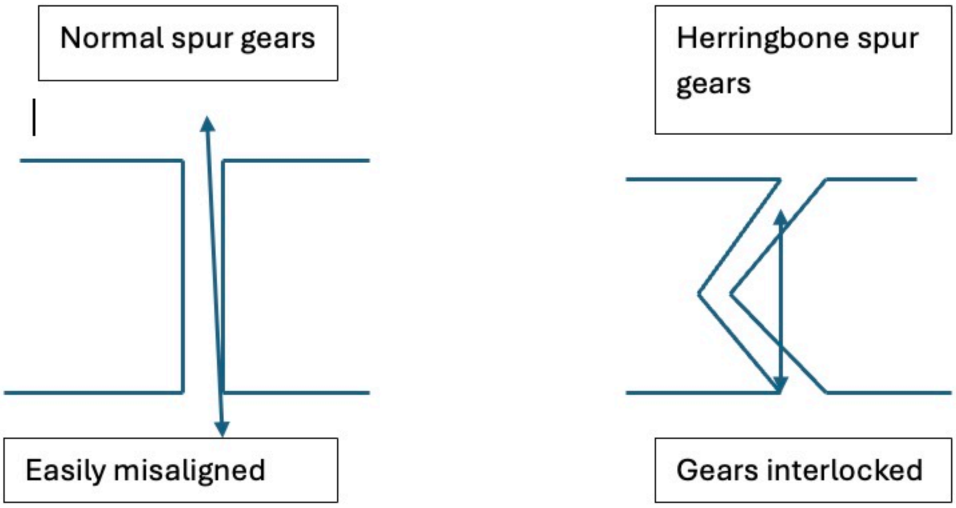


I used crossbeams to support the main body and parts of the Circuitry Housing. This is because Crossbeams use a lower amount of filament which reduces the amount of material and reduces the print time. It also results in a frame that is often stronger than its solid counterpart due to the high density of filament in the beam as well as the triangular structure it creates, which is capable of withstanding more force.

Issues with design

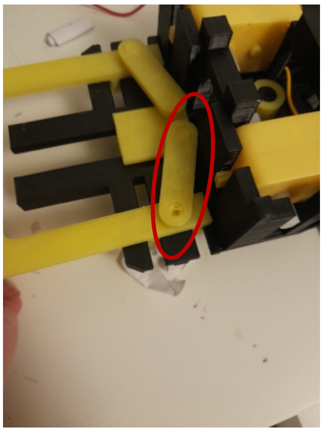
Now that I have printed the design, I have been able to identify 3 main problems across chassis and the mechanism.

Misaligned gears



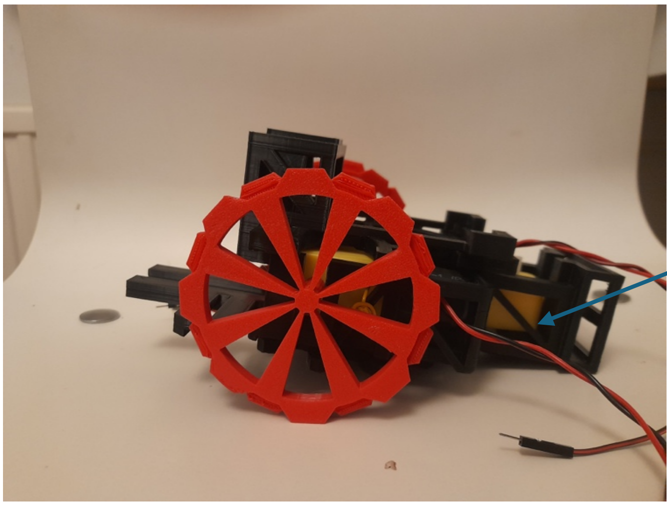
My current design uses normal spur gears which easily become misaligned which results in the mechanism jamming and stopping. To fix this I will reprint the gears using the herringbone design in the future. This will help the mechanism run more smoothly.

Low Grip Strength

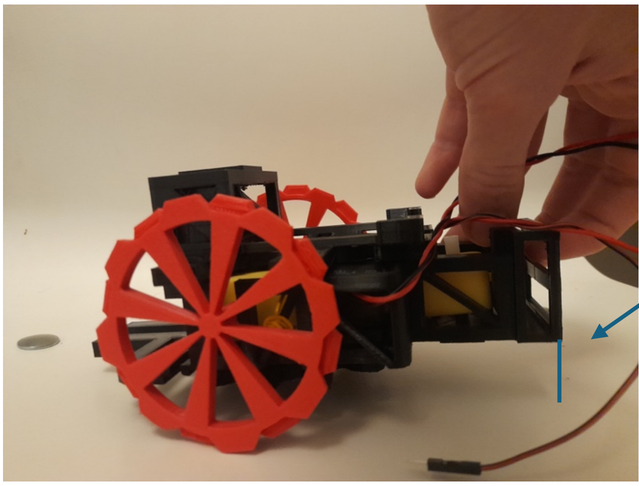


The design has a low grip strength because there is only one arm running between the main slider and each claw. In a future design, I will solve this by adding a second arm between the slider and each arm. This will lead to a more stable positioning of the claw and therefore a more powerful grip strength.

Scraping of mechanism



Master Gear scrapes along the floor which can lead to problems with the driving of the claw

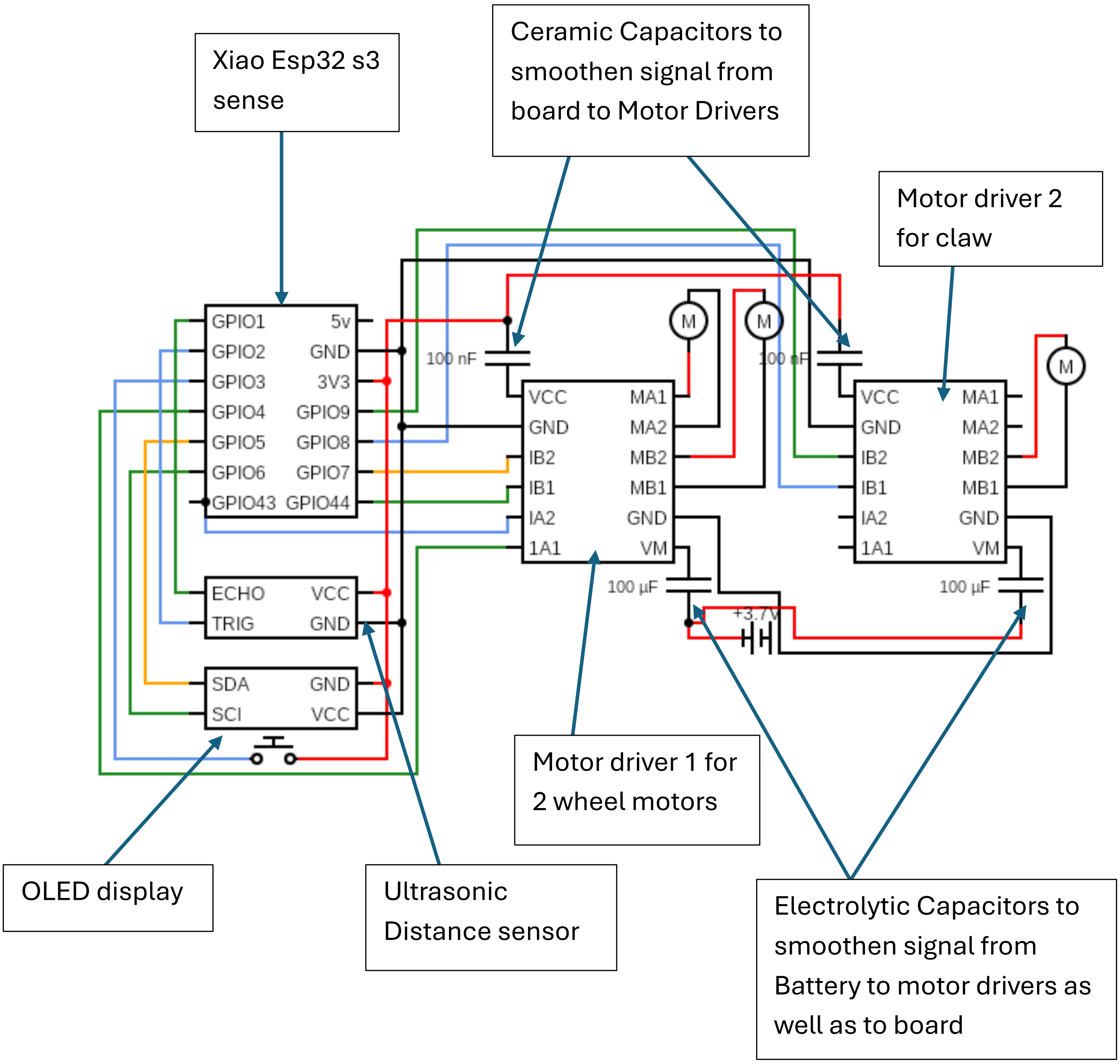


By lowering the rear of the chassis, it will prevent the gear from scraping along the floor.

In the future I will redesign the Chassis to prevent these issues from occurring.

Electronics

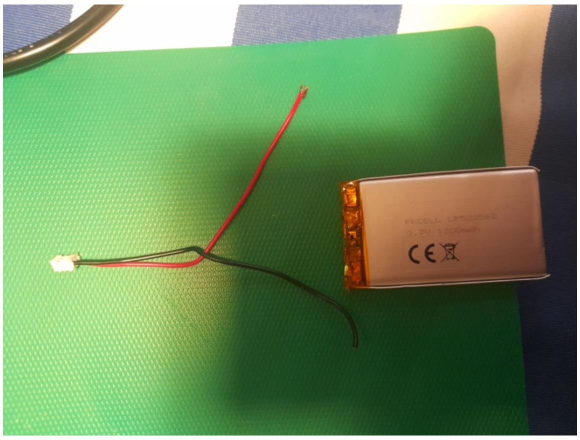
Schematic



GPIO connections

GPIO pin	**Connection**
GPIO1	Distance Sensor Echo (Input)
GPIO2	Distance Sensor Trigger (Output)
GPIO3	Button (Input)
GPIO4	IA1 Motor Driver 1(speed of left wheel)
GPIO5(SDA)	OLED SDA (output)
GPIO6(SCL)	OLED SCL (input)
GPIO7	IB2 Motor Driver 1(direction of right wheel)
GPIO8	IB1 Motor Driver 2(speed of claw)
GPIO9	IB2 Motor Driver 2(direction of claw)
GPIO43	IA2 Motor Driver 1(direction of left wheel)
GPIO44	IB1 Motor Driver 1(speed of right wheel)

Soldering



In the future, I will solder the components together onto a sheet of proto board, but I am currently unable to do so because the wires on the batteries have come disconnected from the positive and ground terminals on the batteries. I am unable to resolder the battery as the terminals are inaccessible below the tape used to cover them. To solve this issue, I will have to order a new battery.

Code

Once I have fixed the issues with the soldering, I will use the following code to control the components:

```
//Motor setup
const int left_speed = 4;
const int left_direction = 43;

const int right_speed = 44;
const int right_direction = 7;

const int claw_speed = 8;
const int claw_direction = 9;

pinMode(left_speed, OUTPUT);
pinMode(left_direction, OUTPUT);

pinMode(right_speed, OUTPUT);
pinMode(right_direction, OUTPUT);

pinMode(claw_speed, OUTPUT);
pinMode(claw_direction, OUTPUT);
```

Defines the pin number of each component

Defines each pin as an output

```
void left_wheel(float speed){
  if(speed < 0){
    analogWrite(left_speed,abs(speed));
    digitalWrite(left_direction,LOW);
  }else{
    analogWrite(left_speed,speed);
    digitalWrite(left_direction,HIGH);
  }}

void right_wheel(float speed){
  if(speed < 0){
    analogWrite(right_speed,abs(speed));
    digitalWrite(right_direction,LOW);
  }else{
    analogWrite(right_speed,speed);
    digitalWrite(right_direction,HIGH);
  }}

void claw(float speed){
  if(speed < 0){
    analogWrite(claw_speed,abs(speed));
    digitalWrite(claw_direction,LOW);
  }else{
    analogWrite(claw_speed,speed);
    digitalWrite(claw_direction,HIGH);
  }}
```

Takes PWM value between -255 to 255

Converts PWM value to positive and sets direction to reverse

Sets speed to PWM value and rotation in positive direction

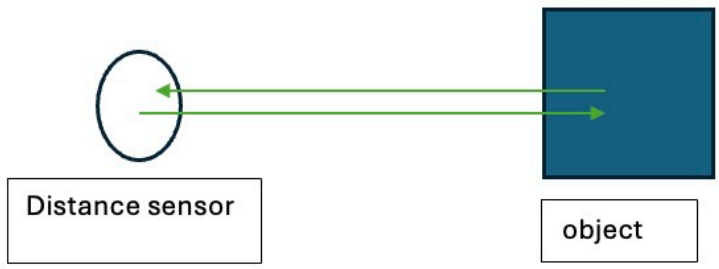
Repeats same function twice for other motors

```
void get_distance(){

  //Send ultrasonic signal
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  float duration = pulseIn(echoPin, HIGH);  //Listen for
  float distance = (duration*.0343)/2; // divide by spee
  return distance; //Distance in cm
```

Sends signal down trigger to distance sensor to prompt echo

Listens for duration wave takes to return to sensor.

Distance = half of the time the beam took to travel multiplied by the distance the wave travels per second

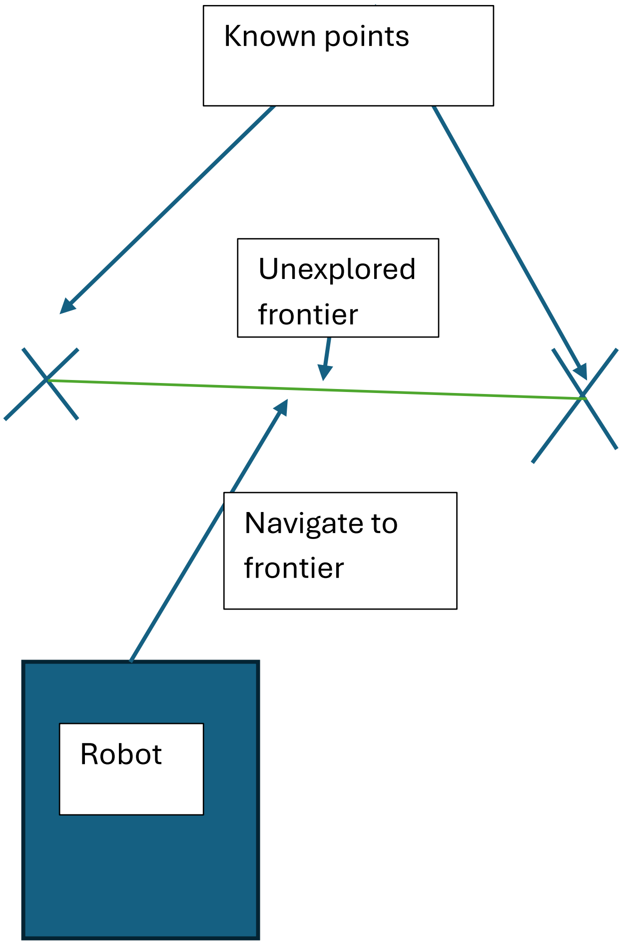


Litter recognition

I will use an edge impulse model trained on the TACO dataset which features a wide array of labelled images of litter making it ideal for the training of my model.

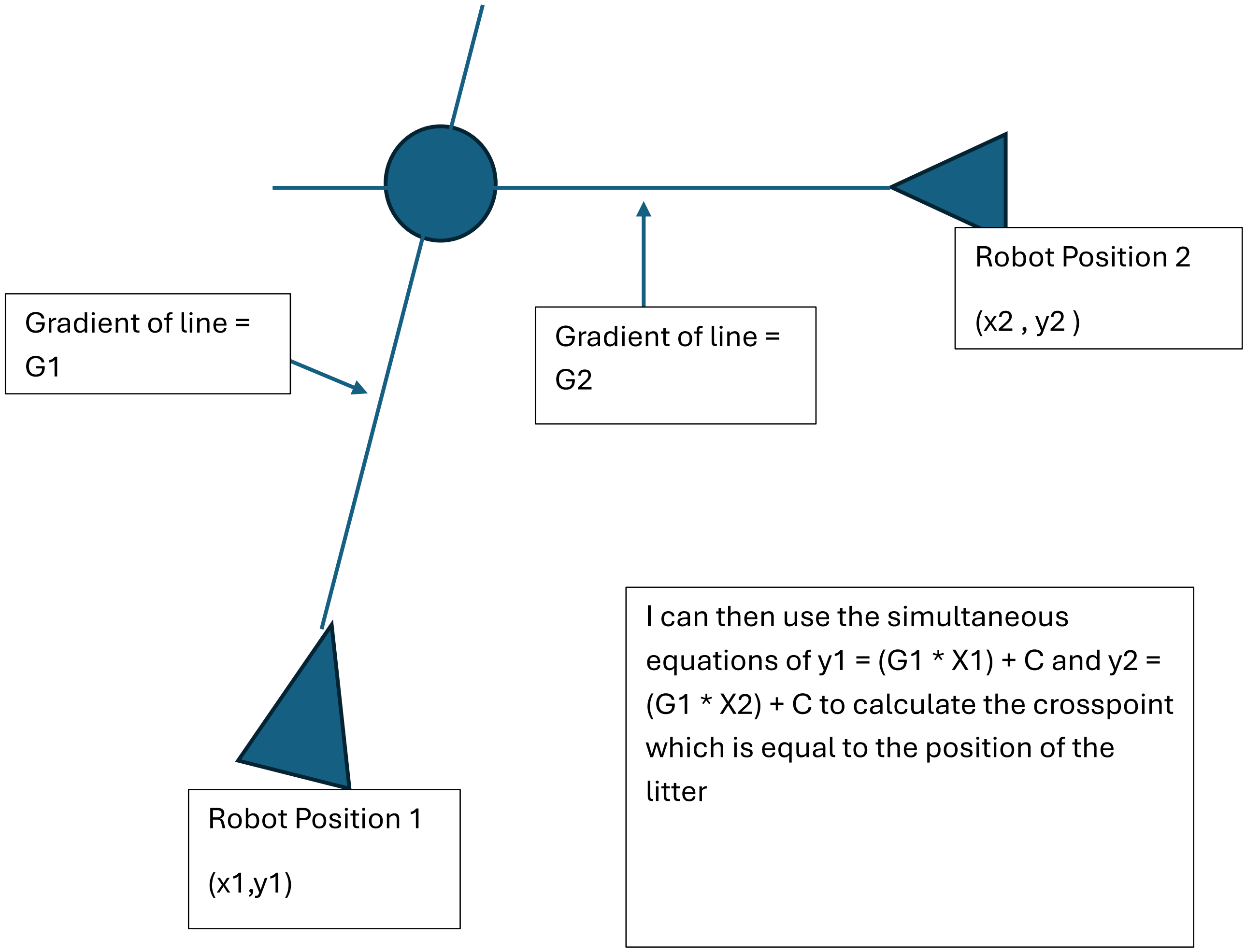
Slam algorithm

In order to allow the robot to navigate I will use a SLAM (or Simultaneous Localisation Algorithm which operate by navigating to unexplored frontiers in order to document and navigate its environment.



Litter positioning

As most of the pieces of litter will be too irregular in shape to be detected by the distance sensor, I will instead use a mathematical approach to calculate the position of the plastic based on the angles, which it can be seen from at different positions



I can then use the simultaneous equations of y1 = (G1 * X1) + C and y2 = (G1 * X2) + C to calculate the crosspoint which is equal to the position of the litter

Next Steps

Next, I will solder the parts together and then implement the code to control the parts. I can them implement the code to identify and locate the litter as well as navigate the environment. Then, I will need to make the robot capable of using the claw to connect and move litter.