EG1311 B18 Team 6
Project Report

Name	Student ID
CHAI CHI KEAN	A0271597N
LEE QIN LING AUDREY	A0283137A
RODERICK KONG ZHANG	A0286550Y
SEAN POON	A0281583X
SI THU LIN AUNG	A0271776R
TAN JIA MIN	A0281575W

Table of Contents

1. Introduction

This report aims to provide a comprehensive overview of our learning experiences, as well as the step-by-step process of creating the final robot design. Subsequently, we will delve into the specific challenges we encountered while designing the robot, and discuss the solutions used to address these challenges. Finally, we will summarise the key insights gained throughout this journey. This section of the report aims to introduce the main objective of this project, as well as the various constraints imposed.

1.1 Aim

The aim of this project is to design a self-powered robot capable of navigating a complex obstacle course, consisting of a starting area, a bump, a ramp, and a wall. Additionally, the robot must successfully deliver a ping-pong ball over the wall (see Fig. 1).

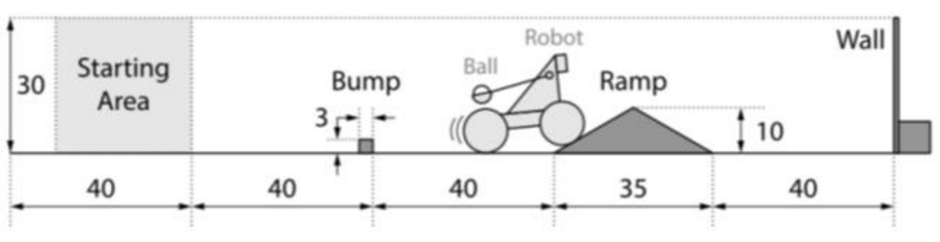


Fig. 1. Schematics for the obstacle course

1.2 Project Constraints

There are five constraints that were imposed for the project:

- The robot must fit within a 30 x 30 x 30 cm cubic cage at the start.
- The robot must be made using only the materials that are provided.
- The team may not interact with the robot after it leaves the starting area.
- The team may not attach anything to the ping pong ball.
- The robot has up to 30 seconds to complete the course.

2. Ideation and Prototyping

This section of the report discusses the initial hardware and software designs of our robot. Initially, a four-motor design with a catapult used to launch a ping-pong ball was considered, but it was heavy and exceeded the prescribed dimensions. Taking into consideration the practicality and functionality of the robot components, the design was simplified to a two-motor configuration with a cannon, significantly reducing its weight and size. In the subsequent sub-sections, we will discuss the design and considerations of the components in our two-motor prototype.

2.1 The Body

For our initial prototype, we identified that the main purpose of the robot body was to support the weight of the Arduino, the breadboard, and the catapult. We decided to use cardboard as the main material as it serves this purpose while being lightweight. Additionally, it provides the flexibility for any design adjustments, which is advantageous for an initial prototype, as compared to less flexible materials such as acrylic. The body features a rectangular base with walls to prevent the internal components from falling off, and a roof for the ultrasonic sensor and catapult attachment (see Appendix B and C for more details).

2.2 The Wheels

In the initial phase, acrylic was chosen as the material for our wheels due to its structural integrity and weight-bearing capability. A trial-and-error approach yielded two designs – the jagged wheels, and the round wheels; both with a 100 mm diameter to ensure that the robot crosses the bump (see Appendix F), and the ramp without stalling at the apex. The jagged wheels were intended to overcome the initial bump as the round wheels lack sufficient grip.

2.3 The Catapult

We figured that the servo motor may not have sufficient power to launch the ping-pong ball over the wall. To address this issue, two ice-cream sticks were linked together to increase the maximum height attainable, allowing the ball to simply drop over the wall without needing to be propelled. The catapult also included a wide and deep box at the end to securely hold the ball in place and prevent it from falling out when traversing the obstacle course.

2.4 The Code

The code consists of five primary functions (labelled in red in Fig. 2). Fig. 2 illustrates our initial thought process for the code in a flowchart. The complete source code can be found in Appendix D.

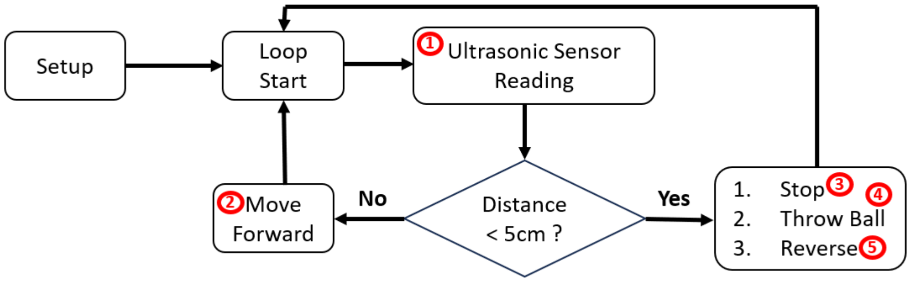


Fig. 2. Flowchart for code (flawed)

2.5 The Circuit

In the circuit, a single H-bridge was used for two motors to allow for motor reversal. Both motor wires are connected to the H-bridge output pins to alternate between high and low voltages, to reverse the motors. To reduce weight, both sides of the breadboard were removed (see Appendix E for the complete circuit).

3. Prototype Testing

In this section, we will address some of the challenges that our group faced during the multiple rounds of testing.

3.1 Stability Issues

Equipped with only 2 motors, our robot experienced stability issues, including lateral sways, as well as deviations from its intended path, particularly after overcoming the obstacles. These problems mainly stemmed from the misalignment of the motors, uneven weight distribution, shifting of components during the obstacle runs, and the significantly wider body compared to the area of contact between the wheels and the ground. Placing the catapult on the roof also raised the robot's centre of gravity, causing it to be even more unstable. Additionally, the use of rubber bands on the wheels (in later prototypes), while necessary for traction, distorted their circular shape due to the use of hot glue, further compromising the stability of the robot.

3.2 Wheel Positioning and Traction

The wheels were initially too close to each other, causing both wheels to contact the bump (first obstacle) simultaneously. Since both front and back wheels were rotating clockwise, a resultant upward force acts on the front of the robot, causing it to rotate counterclockwise, hence toppling over (see Fig. 3).

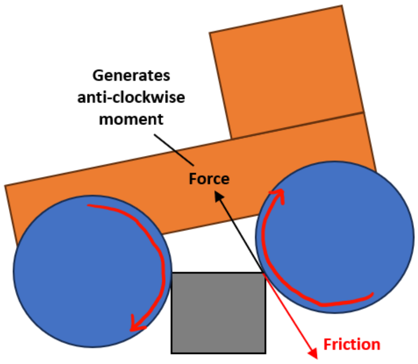


Fig. 3. Issue with wheels

The initial testing also revealed issues with both the jagged and the round wheel prototypes. Both lacked sufficient traction, preventing them from successfully completing the obstacle course. Their failure could be attributed to the smooth surfaces that were created during laser cutting.

3.3 Catapult Size and Design

The oversized catapult design, which was also longer than the robot's body, worsened the stability issues we mentioned earlier. The design raised the robot's centre of gravity, leading to issues when scaling the ramp (second obstacle) – the robot would frequently topple backward due to its weight generating a counterclockwise moment (see Fig. 4).

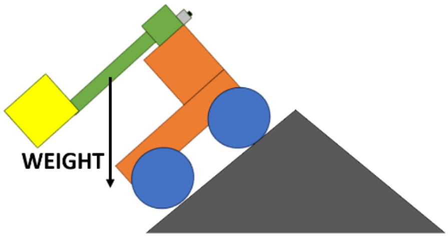


Fig. 4. Issue with the catapult

3.4 Motor Power and Efficiency

Another factor that was overlooked was the torque demands on each motor. As each motor was responsible for driving two wheels, the increased friction acting on each motor imposed higher torque demands on them to sustain motion. Initially, our motors were powered by a 5V supply directly from the Arduino, which proved to be insufficient in generating the required torque. When the torque requirements exceed the maximum capacity of each motor, it reaches the stall condition, causing the motor to stall and resulting in a complete stop in the robot's movement (see Fig. 5).

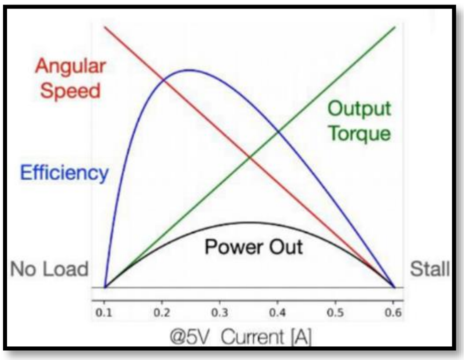


Fig. 5. Relationship between torque and power

3.5 Code Bug

In our initial test run, we encountered a bug in the code. The code worked relatively well for the forward part of the obstacle course but failed when attempting to reverse back to the starting area. The issue was that after the ultrasonic sensor detects the wall and initiates the function to reverse the robot, it would eventually move forward again when the detected distance exceeds five centimetres. This loop continues indefinitely, preventing the robot from reversing as intended.

In the subsequent test runs, we also needed to adjust the ultrasonic sensor's detection range, especially when using a 9V power supply. As the robot's speed increased in these subsequent runs, it was imperative for the sensor to detect the wall at a much longer distance to guarantee the robot's timely halt. The issue primarily stemmed from the robot's inertia.

4. Prototype Improvements

In the previous section, we outlined the challenges encountered during the construction of our robot. In this section, we will go through the steps taken to resolve these issues.

4.1 Addressing Stability Issues

To address the problem with the weight distribution, we conducted multiple tests to obtain the optimal placement of the internal components. To enhance stability, we widened the robot's base by attaching wheel couplers onto each wheel and reduced its overall body size. Additionally, to eliminate unwanted swaying, we used adhesive to secure the internal components, and switched the body material to corrugated polypropene for its increased rigidity and durability, as compared to cardboard. We also reconsidered the practicality of the catapult design and ultimately chose to remove it, considering it to be impractical for a robot operating with only two motors.

4.2 Addressing Wheel Issues

To improve the grip of the wheels, we wrapped the thicker rubber bands around the wheels, by hot gluing them into place. This simulates the traction of car tyres. We also adjusted the front and back motor positions to ensure sufficient spacing between them, preventing both the front and back wheels from simultaneously contacting the bump (see Fig. 3). To accommodate the increased motor spacing, we modified the robot's main body and created dedicated wheel housing areas to ensure that the robot fits within the 30 cm limit.

4.3 Addressing Catapult Issues

After reconsidering the practicality of our catapult design, we opted to replace it with a cannon design. The cannon's mechanism involves pre-loading the ball and using an ice-cream stick to secure it. When the robot reaches the wall, the servo, equipped with another ice-cream stick will turn to flick the ice-cream stick secured to the ball, releasing the tension on the rubber band where the ball sits. This effectively converts the elastic potential energy stored in the rubber band to kinetic energy, which is then transferred to the ball, propelling it over the wall.

4.4 Addressing Motor Issues

Recognising our robot's need for more power, we opted to directly supply the motors with a 9V power source instead of relying on the 5V supply from the Arduino. This adjustment increased the maximum torque the motors can generate, effectively preventing them from ever reaching their stall conditions. Detailed changes to the circuit can be found in Appendix E.

4.5 Addressing Code Issues

To eliminate the code bug, we implemented a Boolean variable to track whether the robot had previously come to a stop. If the variable is true, it never moves forward again. After extensive testing, we determined that the ultrasonic sensor needed to detect the wall at a distance of 11 cm to ensure that the robot would come to a timely stop (see Fig. 6).

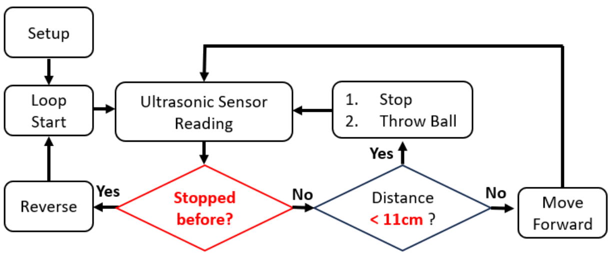


Fig. 6. Revised Code Flowchart (changes in red)

5. Lessons Learnt

In this section, we will delve into the key insights and valuable lessons learnt by our group over the course of this project.

5.1 Embracing All Ideas

During the brainstorming and ideation phase, we discovered the importance of considering ideas methodically instead of dismissing them on a whim. While some concepts may initially appear unfeasible, we learnt that through iterative development and adjustments, what seemed impossible can evolve into a brilliant idea. A prime example of this was our two-motor and cannon designs, which, despite their initial impracticality, became integral components of our final robot following multiple rounds of refinement. Furthermore, by not dismissing ideas, we have significantly broadened our pool of options to work with.

5.2 Proper Prototyping

During the prototyping phase, our group learnt the importance of understanding the design parameters for our robot. In our initial test run, we overlooked certain parameters and relied on intuition instead of established physics principles. This led to issues such as the wheels being placed too closely and the catapult being too large for the robot. In the subsequent test runs, however, we applied the concepts taught to analyse our prototype measurements which helped us to minimise the instances where we needlessly introduced problems that were preventable.

5.3 Understanding Constraints, Practicality and Functionality

During the project, there were instances where we overlooked these fundamental principles, resulting in redundant work being done. For instance, overlooking the step of measuring the robot's size before cutting it out, led to it exceeding 30 cm in length. Additionally, the inclusion of materials or components without adequate consideration of their practicality and functionality resulted in unnecessary effort and increased weight. These examples emphasise the need to consistently question ourselves on whether our ideas align with the three principles to avoid redundancy.

5.4 Learning from Failures

Reflecting on past prototypes and understanding why they failed is pivotal to avoid recurring mistakes. When experiencing failure, it is important that we refrain from making hasty conclusions and instead adopt a systematic approach to identify the root cause of the problem. For instance, during our initial tests, our robot would often either experience glitches or just simply stop functioning. We initially attributed this problem to connection issues and reworked on the wiring multiple times when the main issue was the motor's inadequate power supply from the 5V source, causing it to stall.

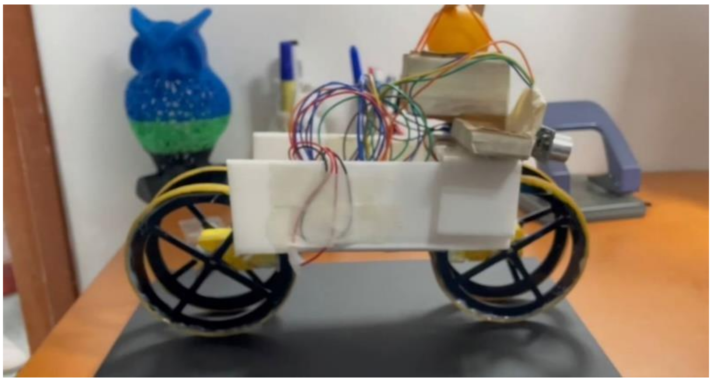
6. Conclusion

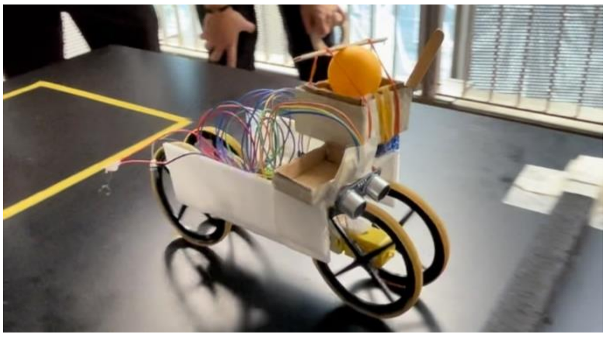
Our journey in building a self-powered robot capable of traversing through an obstacle course and delivering a ball over the wall, was plagued by numerous design and operational challenges. Yet, these challenges also provided us with multiple opportunities to apply the design thinking model to come up with innovative and practical solutions. This journey also offered valuable lessons such as the importance of proper prototyping, adaptability, and teamwork, to solve complex engineering problems.

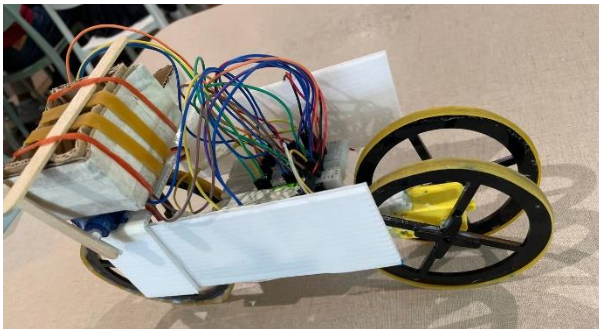
Appendix

Appendix A (Robot Photographs)

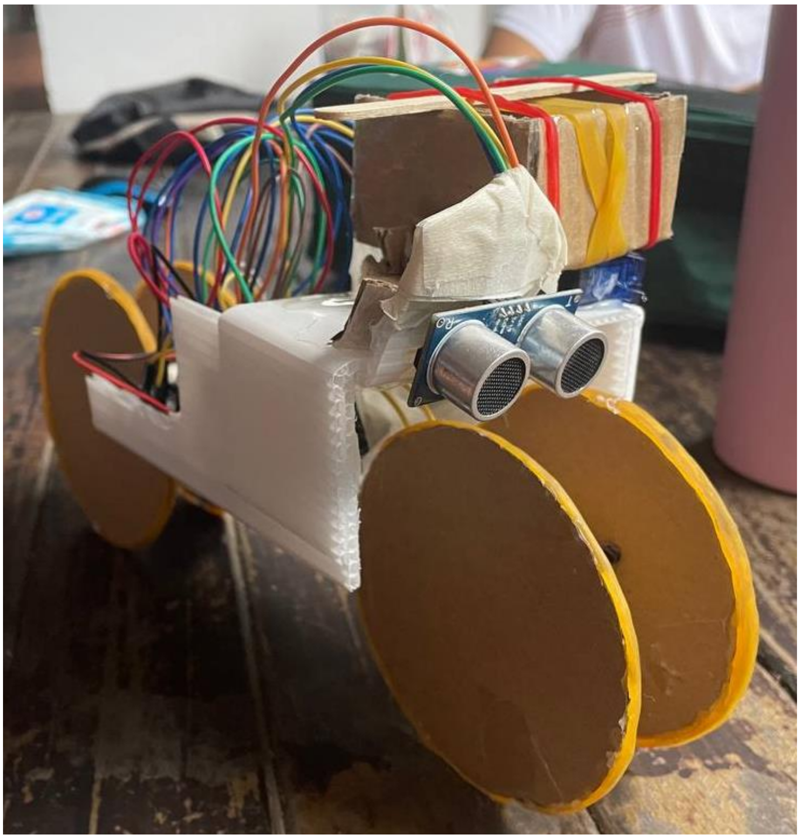
Final Robot 1 (Used in the first two graded runs)





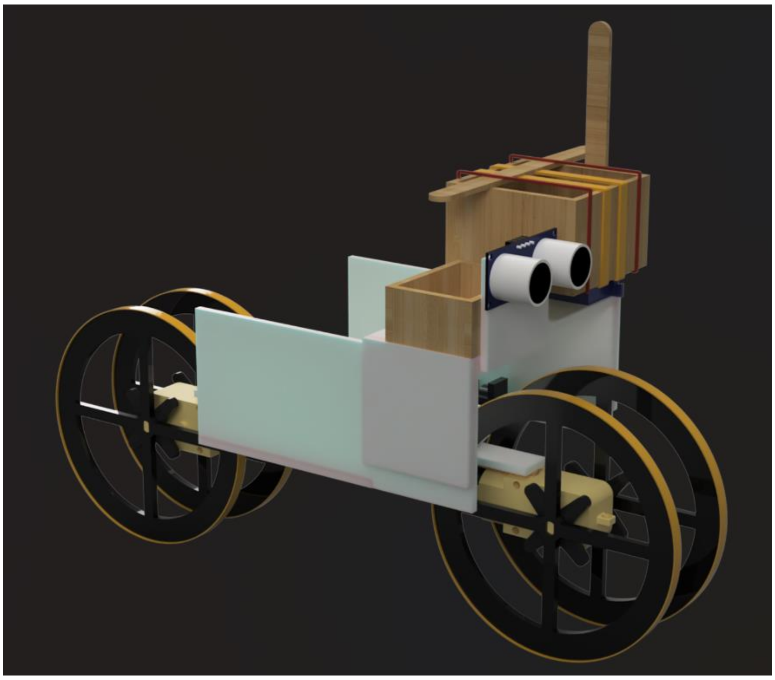


Final Robot 2 (Used in the final graded run to reduce weight)

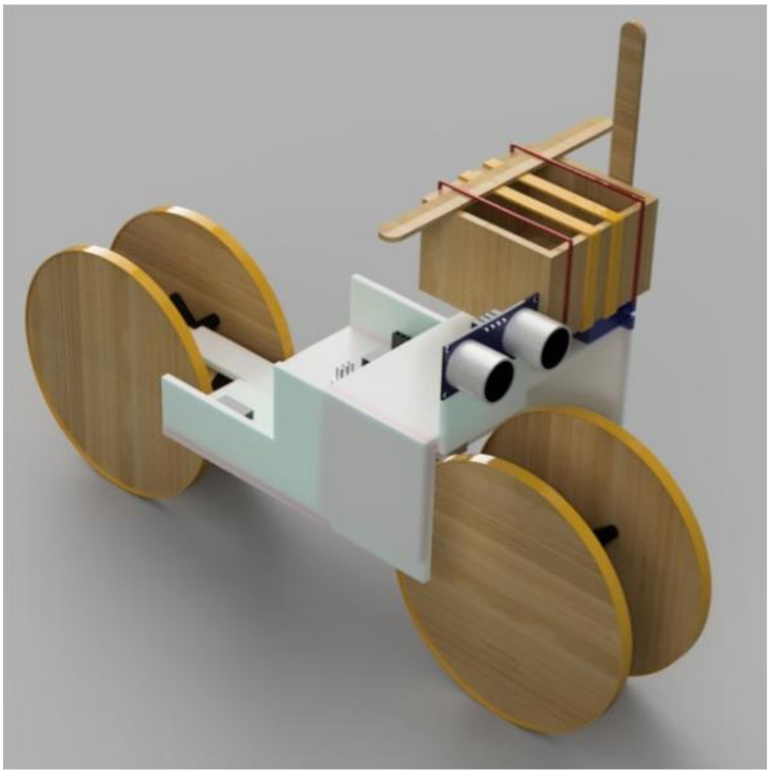


Appendix B (CAD Rendering – Final Robots)

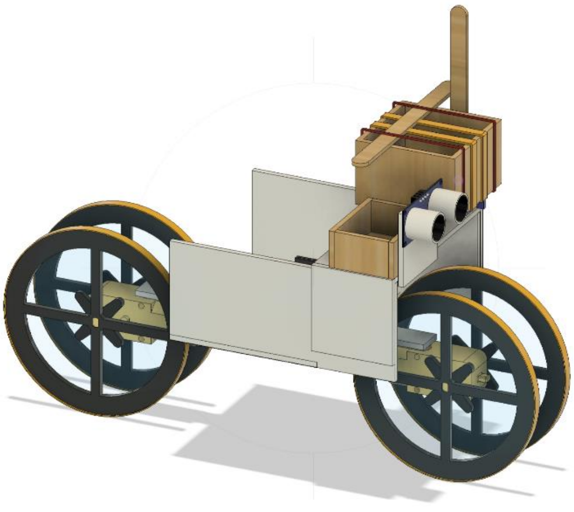
Final Robot 1 (Used in the first two graded runs)



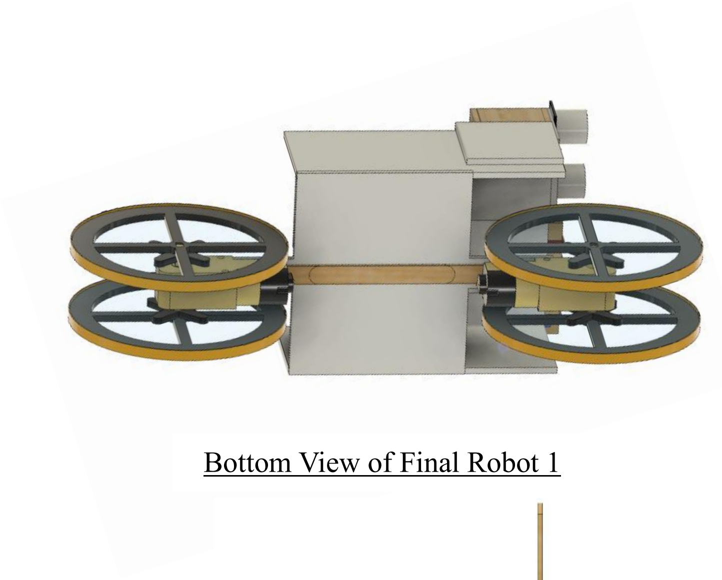
Final Robot 2 (Used in the final graded run to reduce weight)



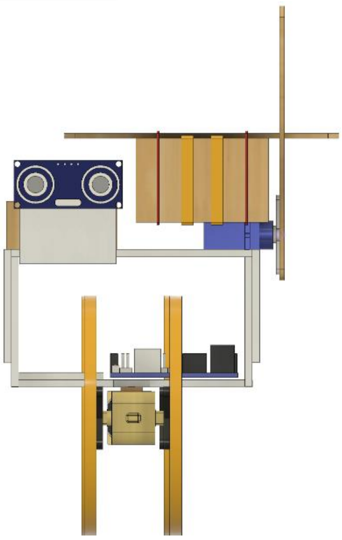
Final Robot 1 (Used in the first two graded runs)



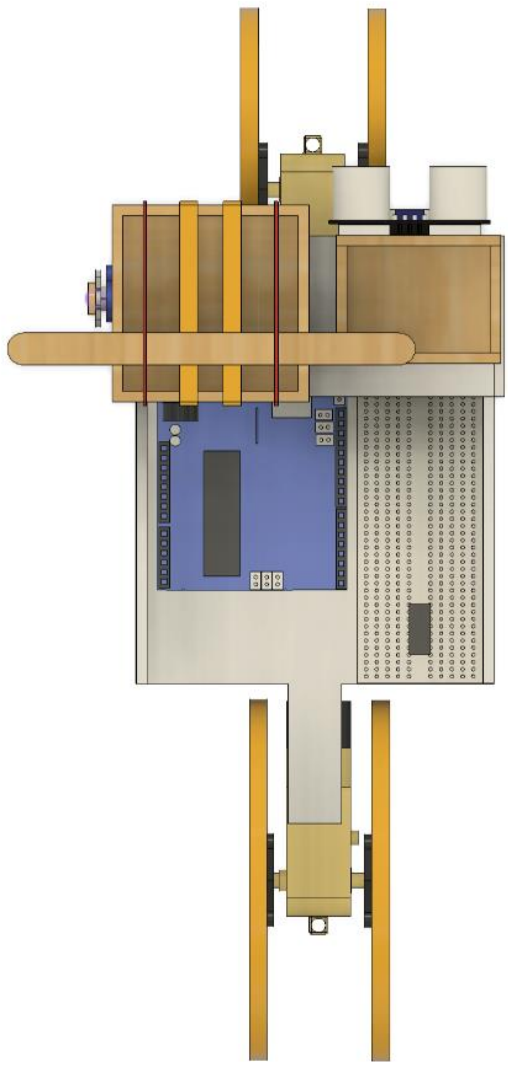
Isometric View of Final Robot 1



Bottom View of Final Robot 1

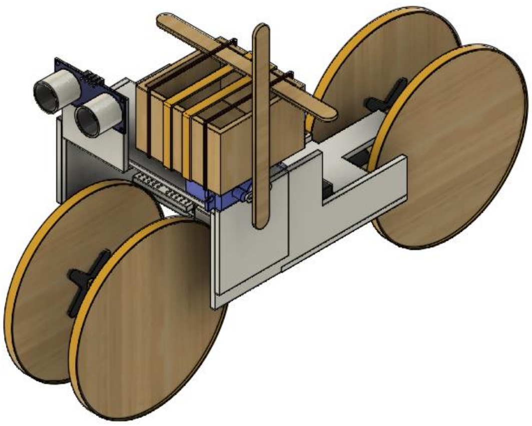


Front View of Final Robot 1

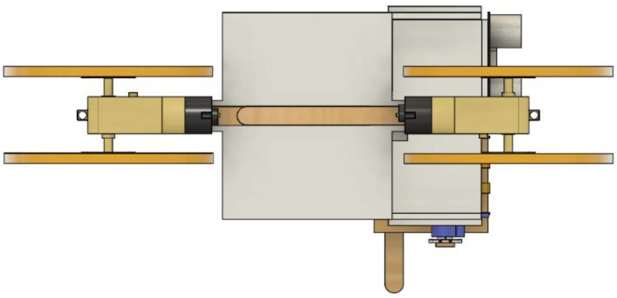


Top View of Final Robot 1

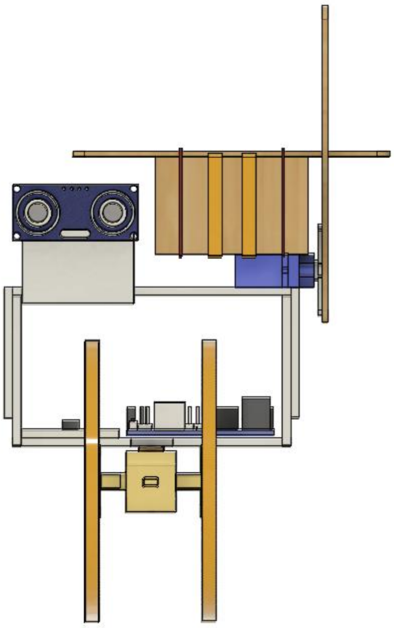
Final Robot 2 (Used in the final graded run to reduce weight)



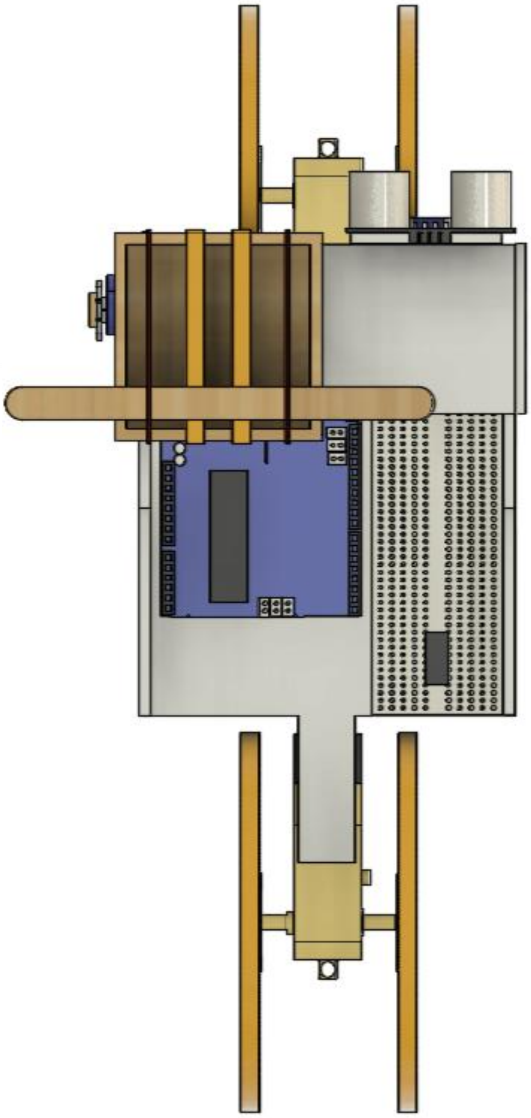
Isometric View of Final Robot 2



Bottom View of Final Robot 2



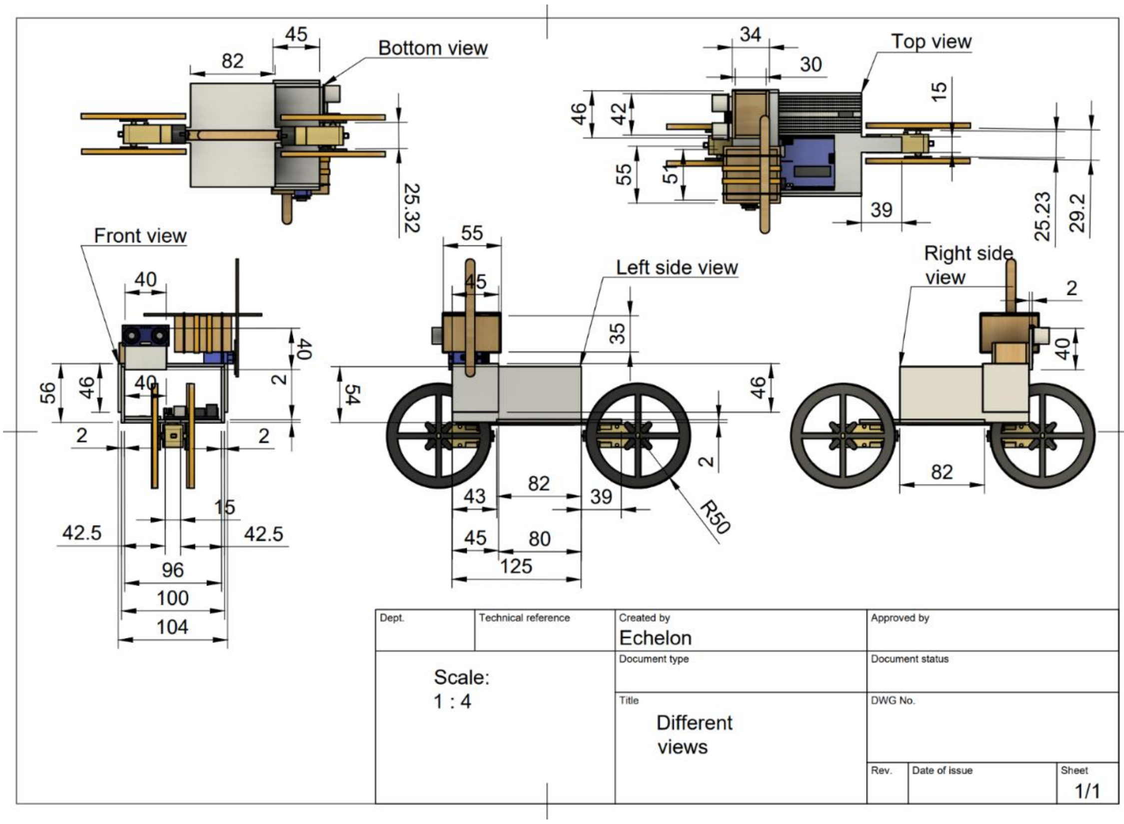
Front View of Final Robot 2



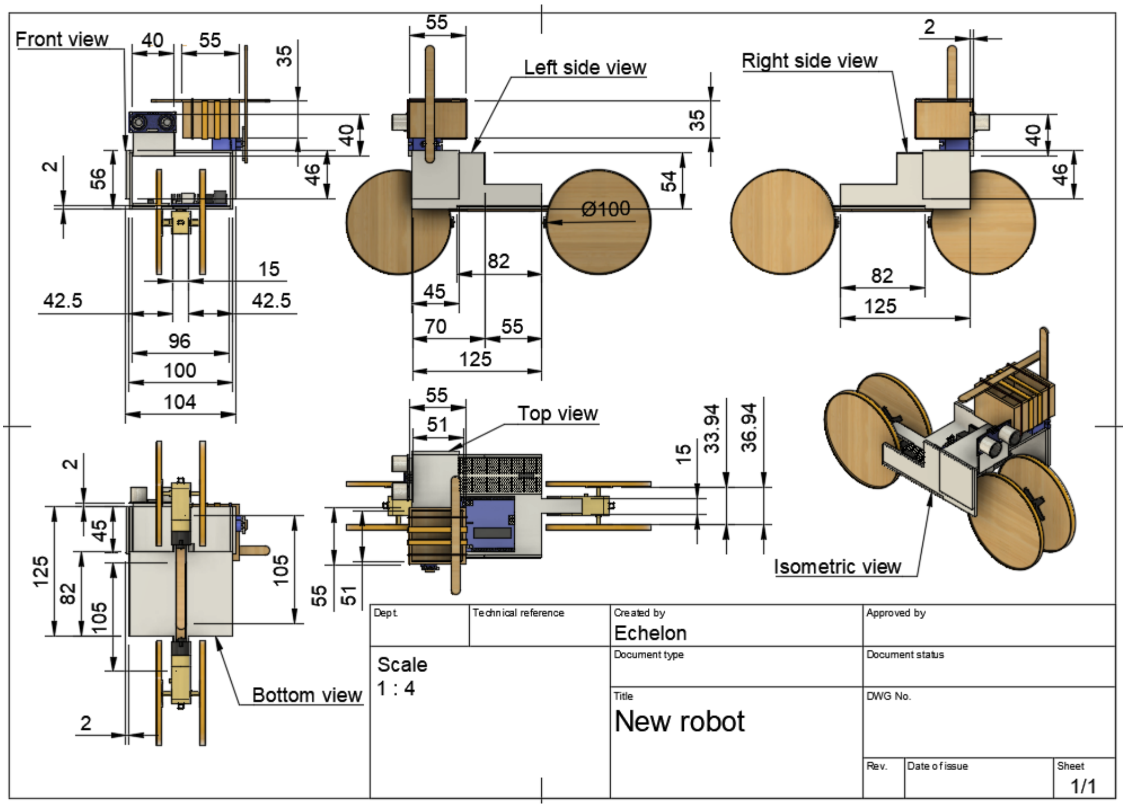
Top View of Final Robot 2

Appendix C (2D CAD Drawings – Final Robots)

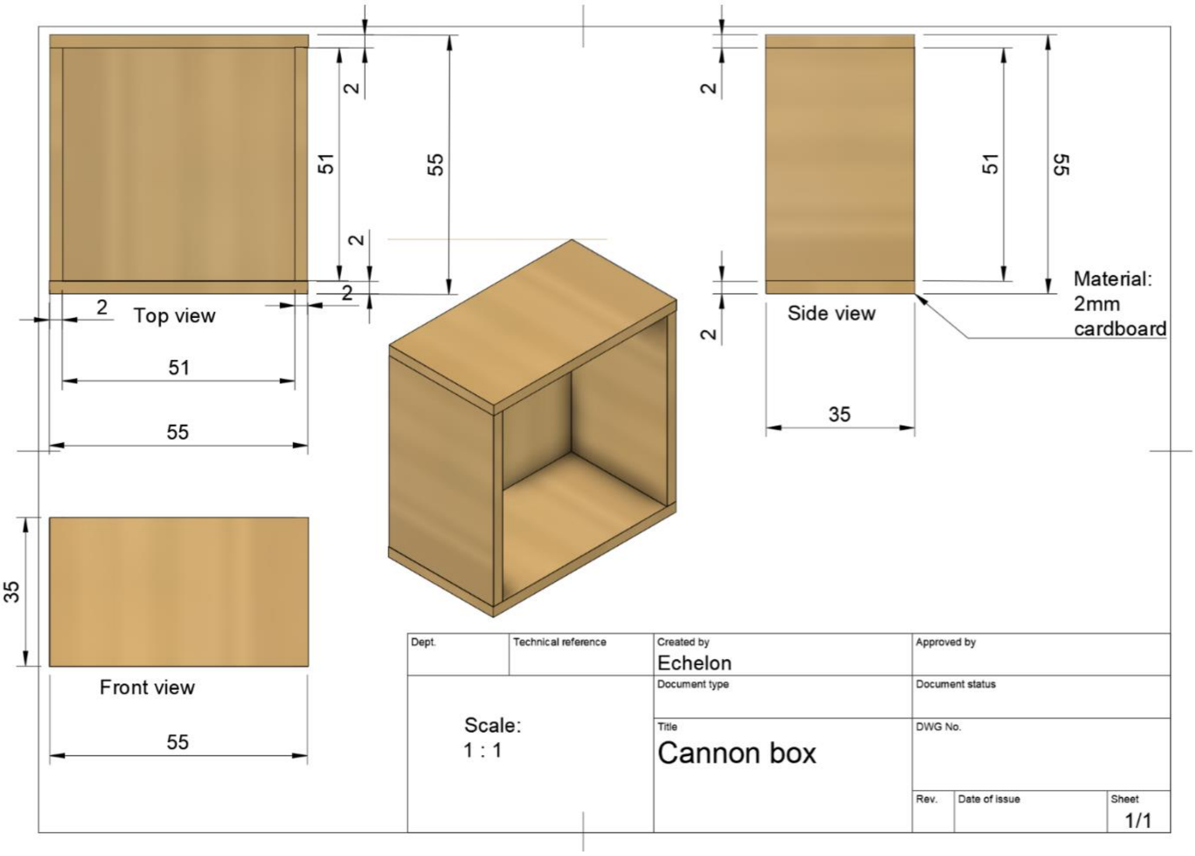
Final Robot 1 (Used in the first two graded runs)



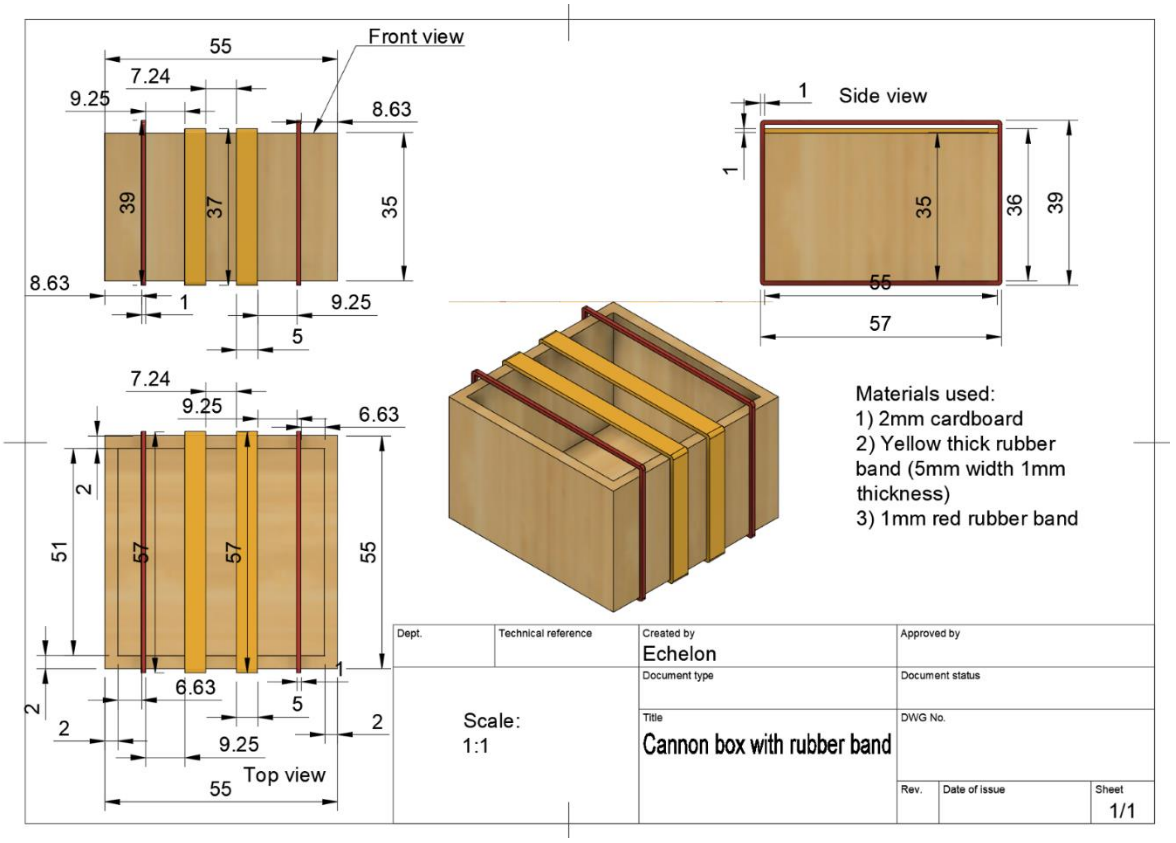
Final Robot 2 (Used in the final graded run to reduce weight)



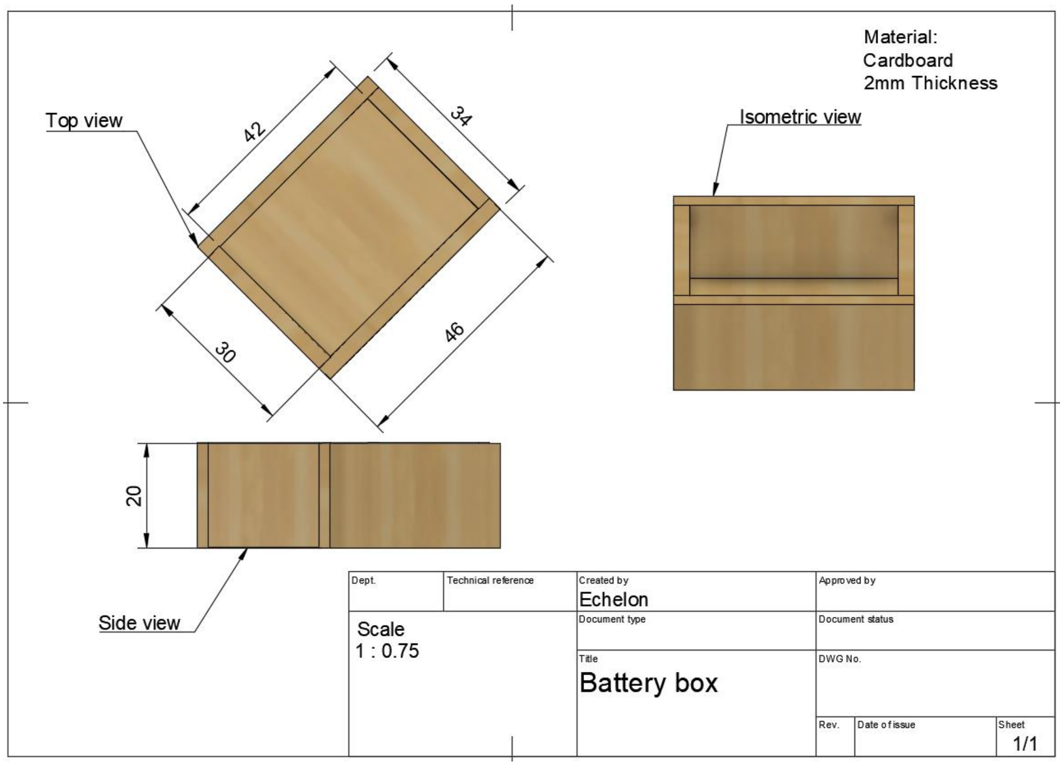
Cannon Box (without rubber bands)



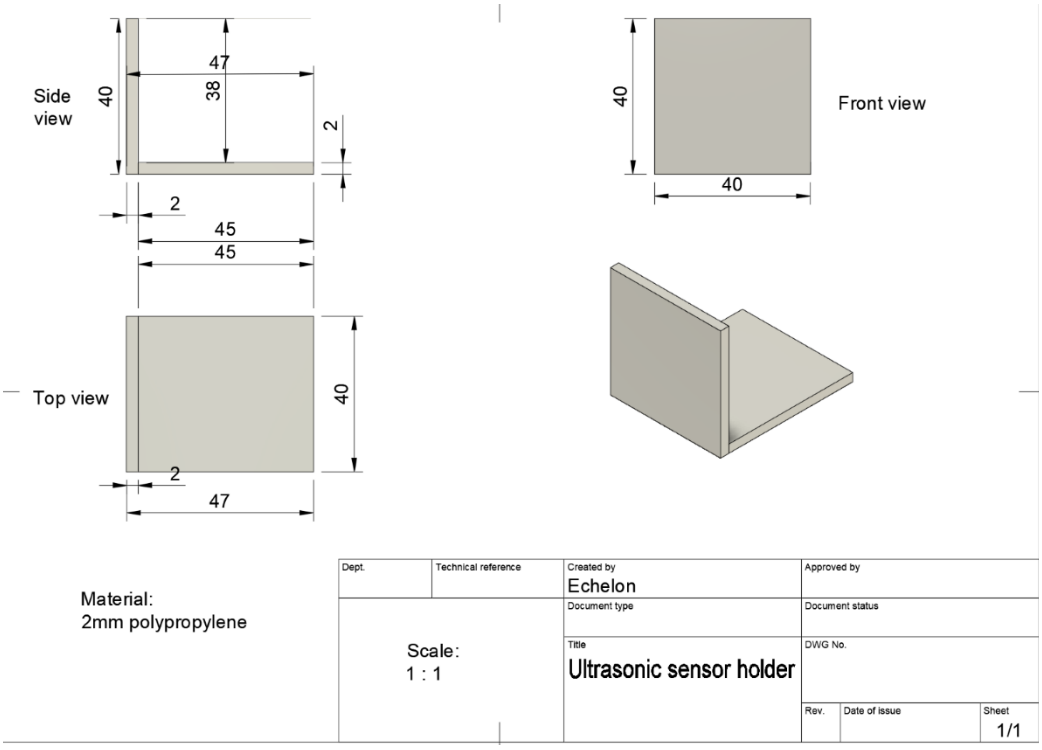
Cannon Box (with rubber bands)



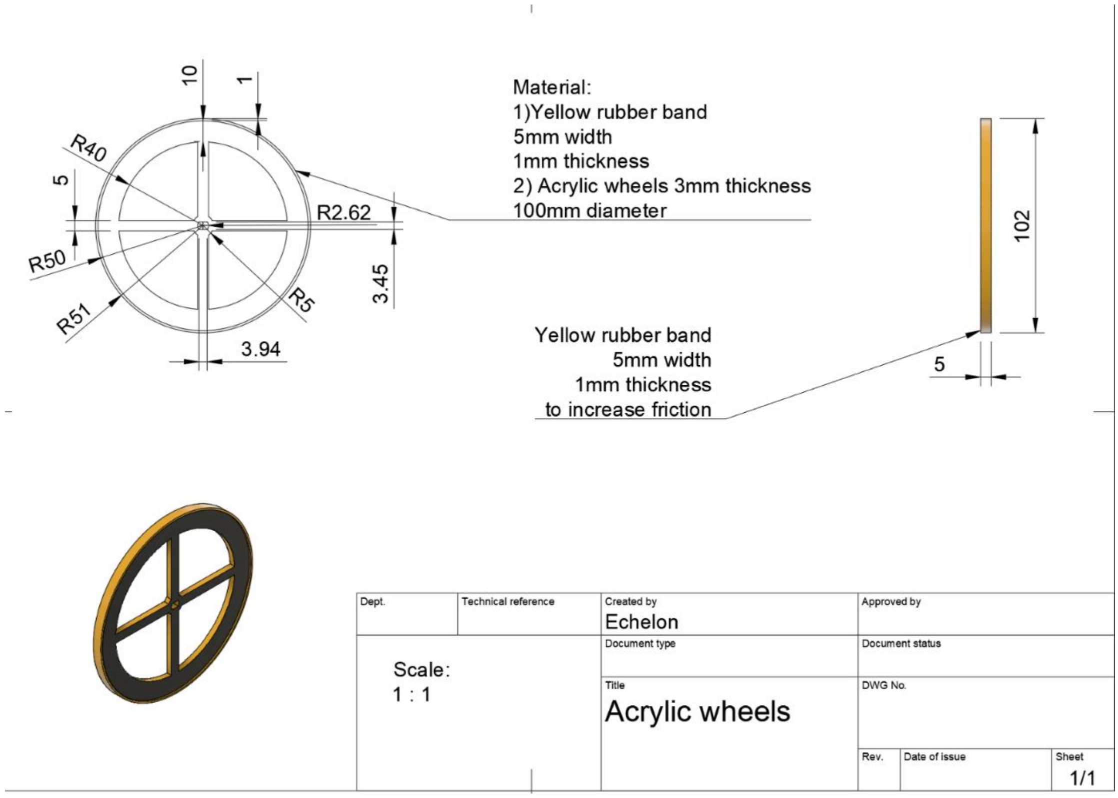
Battery Holder (Used only in Final Robot 1)



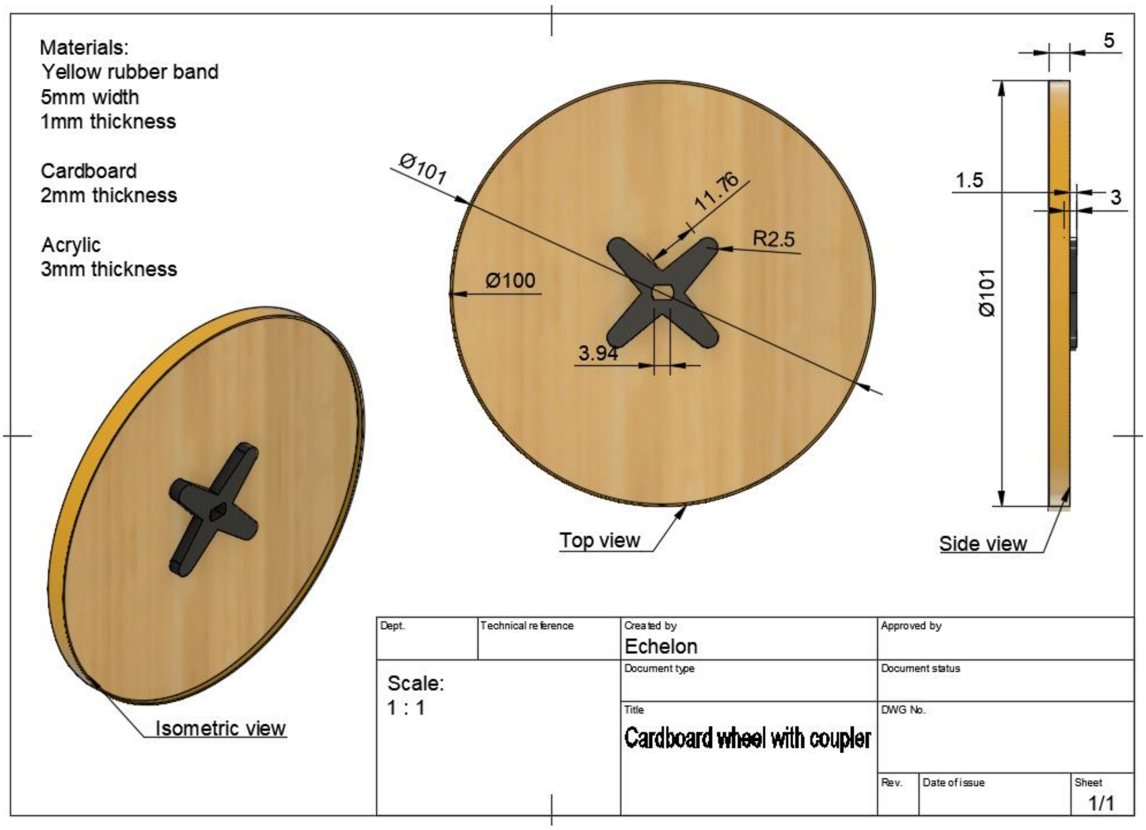
Ultrasonic Sensor Holder



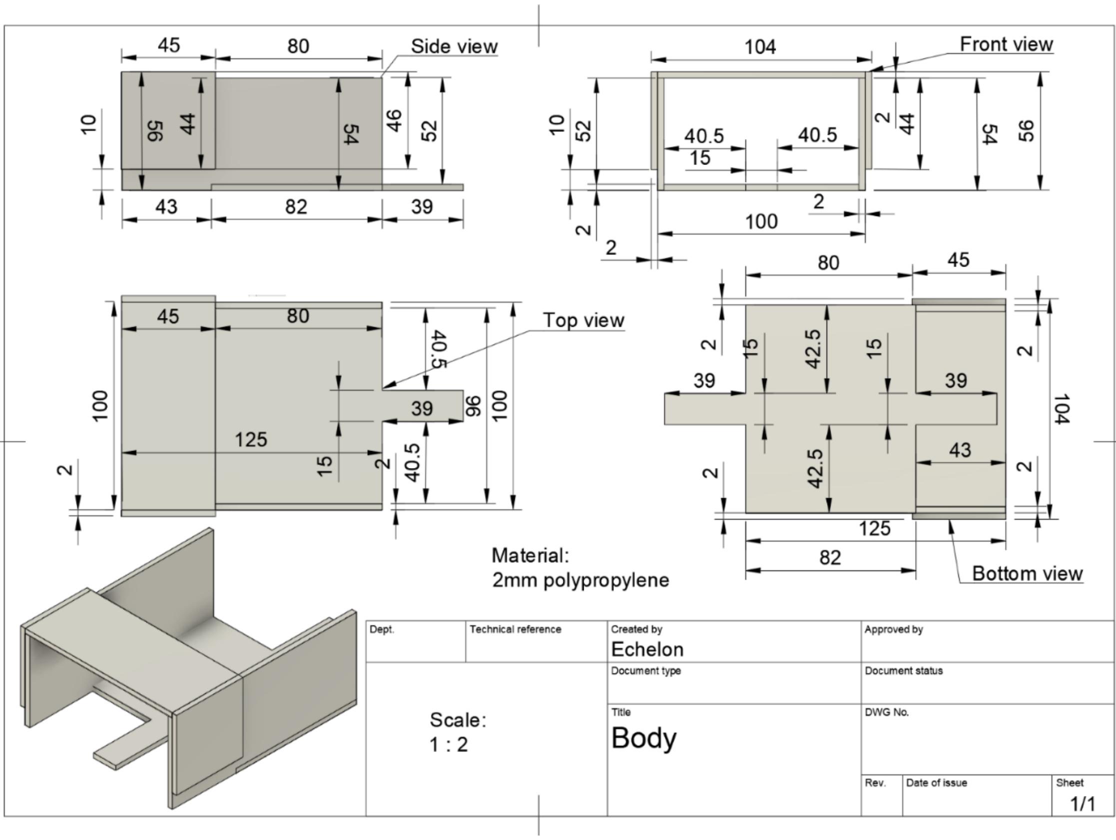
Acrylic Wheels (Final Robot 1)



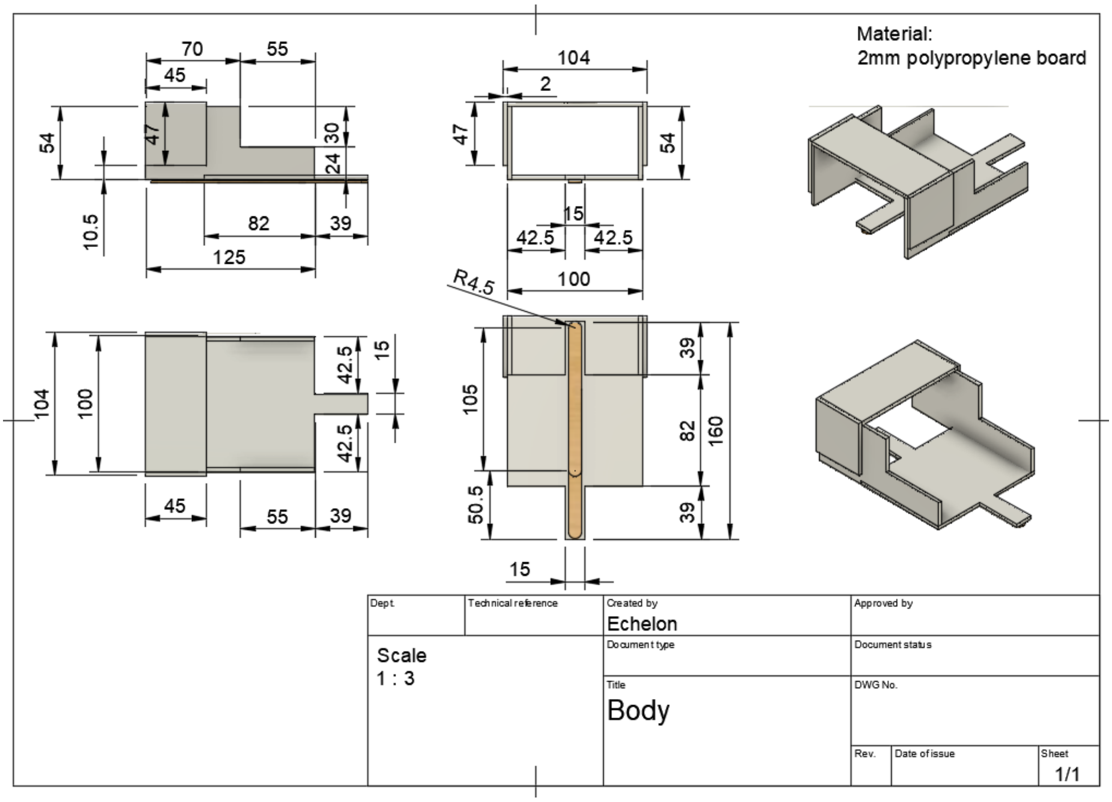
Cardboard Wheels (Final Robot 2)



Main Body (Final Robot 1)



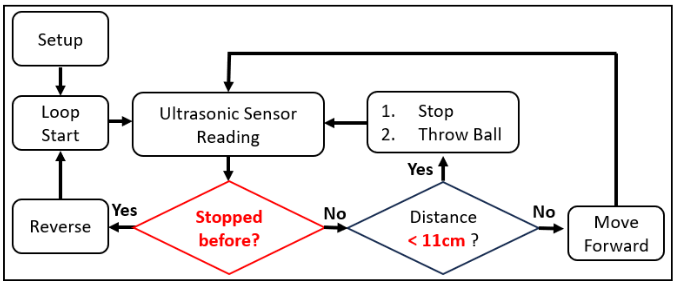
Main Body (Final Robot 2)



Appendix D (Arduino Code)

```
1  #include <Servo.h>
2  Servo servo;
3
4  //Below are the variables and pins used
5  int MOTOR_1A = 4;
6  int MOTOR_1B = 5;
7  int MOTOR_2A = 6;
8  int MOTOR_2B = 7;
9  int TRIG = 13;
10 int ECHO = 12;
11 int servo_pin = 3;
12 bool stopped_once = false;
13 float sound_speed = 0.0345;
14
15 //Function below moves the robot forward
16 void forward()
17 {
18   digitalWrite(MOTOR_1A, HIGH);
19   digitalWrite(MOTOR_1B, LOW);
20   digitalWrite(MOTOR_2A, HIGH);
21   digitalWrite(MOTOR_2B, LOW);
22 }
23
24 //Function below stops the vehicle
25 void stop()
26 {
27   digitalWrite(MOTOR_1A, LOW);
28   digitalWrite(MOTOR_1B, LOW);
29   digitalWrite(MOTOR_2A, LOW);
30   digitalWrite(MOTOR_2B, LOW);
31 }
32
33 //Function below moves the robot backwards
34 void backward()
35 {
36   digitalWrite(MOTOR_1A, LOW);
37   digitalWrite(MOTOR_1B, HIGH);
38   digitalWrite(MOTOR_2A, LOW);
39   digitalWrite(MOTOR_2B, HIGH);
40 }
41
42 // Below this section are the main codes
43 void setup()
44 {
45   Serial.begin(9600);
46   pinMode(MOTOR_1A, OUTPUT);
47   pinMode(MOTOR_1B, OUTPUT);
48   pinMode(MOTOR_2A, OUTPUT);
49   pinMode(MOTOR_2B, OUTPUT);
50   pinMode(TRIG, OUTPUT);
51   pinMode(ECHO, INPUT);
52   digitalWrite(TRIG, LOW);
53   servo.attach(servo_pin,300,2700);
54   servo.write(180);
55 }
56
57 void loop()
58 {
59   digitalWrite(TRIG, HIGH);
60   delayMicroseconds(10);
61   digitalWrite(TRIG, LOW);
62   int time_of_pulse = pulseIn(ECHO, HIGH);
63   float distance = (time_of_pulse * sound_speed) / 2;
64   Serial.println(distance);
65   if ((distance < 11) && !(stopped_once)){
66     stop();
67     delay(1000);
68     servo.write(90);
69     delay(1000);
70     servo.write(180);
71     stopped_once = true;
72   }
73   else if (stopped_once) {
74     backward();
75   }
76   else {
77     forward();
78   }
79   delay(10);
80 }
```

Code Flowchart

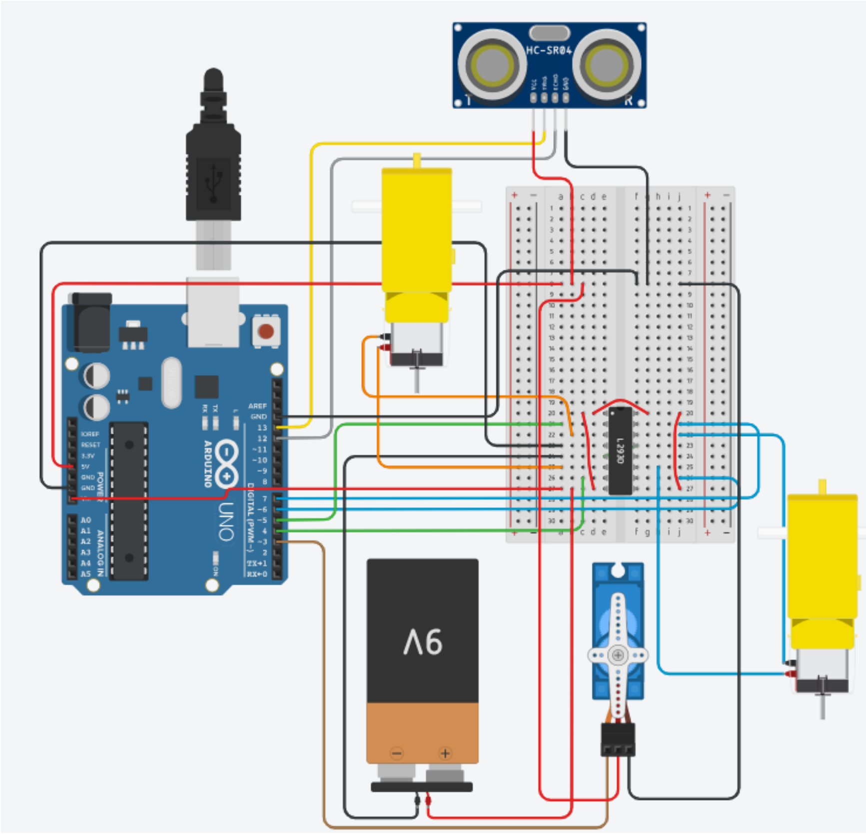


Explanation of Code

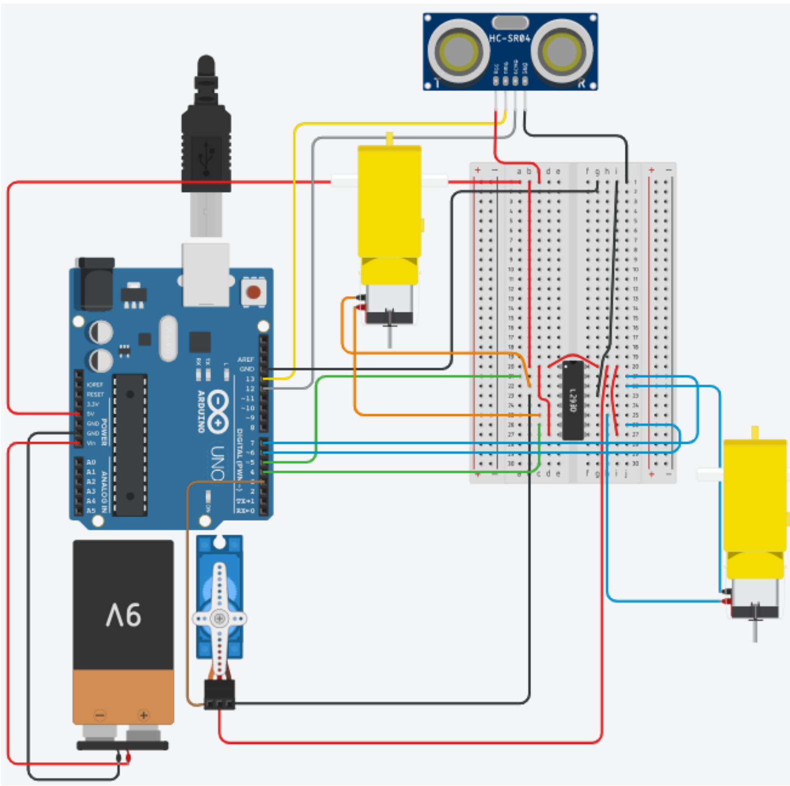
Lines	Purpose
4 – 13	Set the pin numbers and some of the variables that are used in the code. All the variables are compiled here for ease of change when required.
15 – 22	Function to make the robot move forward by setting one of the wires of the motor as HIGH and the other as LOW (ground wire).
24 – 31	Function to make the robot stop moving by setting all the wires to LOW, which stops all current flow into the wire.
33 – 40	Function to move the robot backwards by setting one of the wires of the motor as HIGH and the other as LOW (ground wire). Notice that this is the **opposite** of the forward function to ensure that **current** now flows in the **reverse direction**.
42 – 55	Sets up the relevant output and input pins (only once).
59 – 63	The ultrasonic sensor's main function. Sends out a pulse and the time for the pulse to hit the wall and back is stored. Distance calculation can be done with this duration that was measured.
65 – 72	When it reaches within 11 cm of the wall, the robot will stop and launch the ball. The Boolean "stopped_once" is updated to true, making "!stopped_once" false, preventing the code from entering this condition ever again.
73 – 75	Robot will only move backwards if it has stopped once before.
76 – 78	If the robot has not reached the wall, move forward.

Appendix E (Electrical Circuits – TinkerCAD Diagrams)

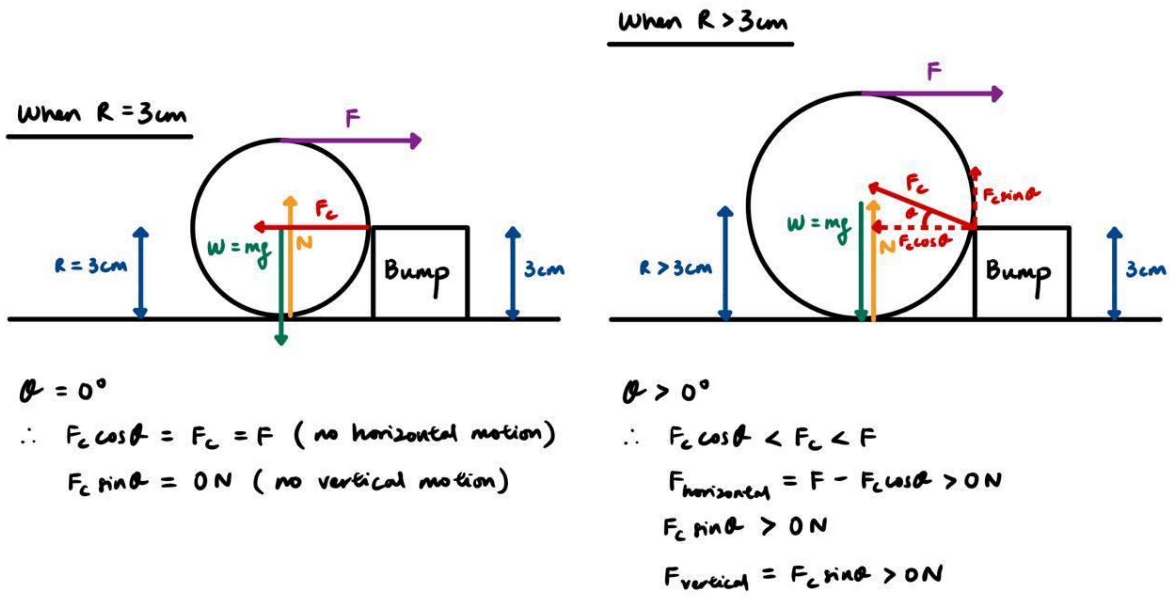
Final 9V Circuit Design (Used for graded runs)



5V Circuit Design (Used in previous prototypes)



Appendix F (Explanation for Increased Wheel Diameter)



The normal contact force from the bump on the wheel provides for the vertical movement of the wheel, while impeding the horizontal movement of the wheel. When R ≤ 3 cm, the horizontal component of this normal contact force is maximum and equal to F, and the vertical component is zero, hence there is no lift and forward momentum for the wheel to mount the bump. However, when R > 3 cm, the horizontal component of this normal contact force is less than F, while the vertical component of this normal contact force provides the lift for the wheel. Therefore, a large enough wheel radius > 3 cm (diameter > 6 cm) is required for the wheel to mount the bump.