EG1311 DESIGN AND MAKE

B13 TEAM 6 REPORT

Justin Poon Jun Jie	A0308159Y
Pho Jia Xuan	A0301314B
Samuel Lee Wen Jin	A0307792W
Zhang Bingran	A0311931N
Zhu Wenbo	A0307272J

Table of Contents

1. The Objective

The aim of the project is to develop a self-powered robot capable of navigating an obstacle course consisting of a starting area, a bump, a ramp, a wall as well as launching a ping pong ball over the 30 cm-high wall (Figure 1). The robot must be able to fit within a 30 cm^3 starting area and be constructed with the supplied materials within 30 seconds to complete the course.

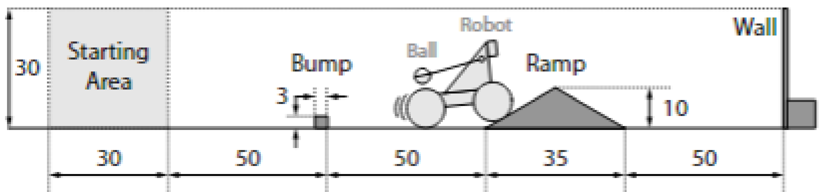


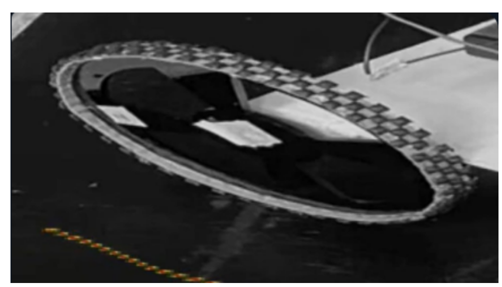
Figure 1.1 Layout of Obstacle Course

To achieve this goal, we adopt the method of prototyping and testing. We used Fusion360 and Tinker CAD software to prototype and test the circuits without the hardware. Through much trial and error, we managed to create a robot capable of completing the objectives.

2. Prototyping and Testing

During this stage, we faced a variety of challenges that hindered our progress in completing the robot. Thus, we had to make several changes to our robot throughout the semester, gaining valuable insights and learning points along the way.

2.1. Problems

Initial planning / Problems faced	**Diagrams**
Wheels The initial design was to round the wheel with rubber by using hot glue, which could increase the friction effectively. We made the width of the rubber roughly match the width of the wheel (Figure 2.1). However, it was difficult to fix the rubber on the wheel and the excess part of the rubber would touch the board or wiring. The rubber mat did not stick adhesively onto the wheels and came off easily. This resulted in the wheels consistently contacting the robot platform and wiring leading to consistent failed runs.	 *Figure 2.1 Initial Wheel Design*

Catapult

Our initial idea was to employ the catapult seen in movies such as Lord of the Rings, as we observed that they provided significant power to launch objects in a parabolic motion, given that the wall we needed to overcome was 30cm. Hence, we sought to design a catapult with a similar design (Figure 2.2).

As we progressed with the development of our robot, we sought to minimise weight and size to fit the strict dimensions requirements. Our catapult was too large, and our servo holding the rubber band had a high tendency to fail due to the intense vibrations experienced by the robot throughout the obstacle course. Furthermore, the high tension of the rubber band resulted in excessive warping of our robot frame causing it veer off course.

Figure 2.2 Initial Catapult Design

Ultrasonic Sensor

The initial design of the sample robot placed the ultrasonic sensor at the front of the robot elevated at an angle (Figure 2.3). We were unable to evaluate if the sensor would provide feedback as the robot could not overcome the first obstacle.

The ultrasonic sensor was placed on the chassis itself in the pursuit of weight reduction, however it resulted in false detections. The drone would detect the ramp as the wall, resulting in not meeting the requirement of part of the robot within 5cm of the wall and throwing the ball over it.

While our robot overcame all obstacles with the four-wheel methods we implemented, we were looking to fulfil the extra points of the ability to move back to its start position. While in our simulation, the robot was able to return to its start position, in practice the sheer number of wires resulted in poor connection to the motors, leading to inconsistent performance of the motors.

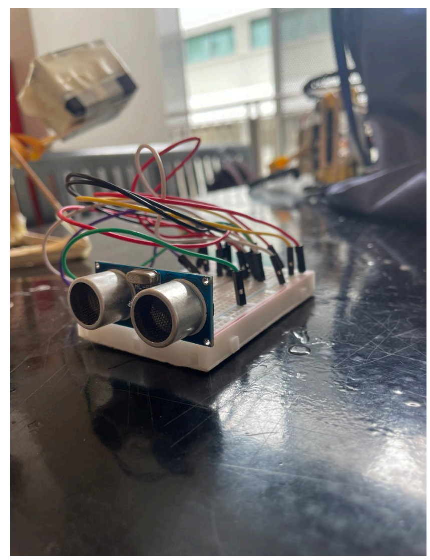


Figure 2.3 Initial Ultrasonic Sensor Position

Platform

The initial idea was to use a propylene board to support the breadboard and Arduino board, and carry four motors and a catapult (Figure 2.4). We made it a bit long to ensure that there was enough space to hold everything.

We optimised the previous design to include just three motors.

The propylene board was not rigid enough to hold the combined weight of the batteries and the breadboard causing the wheels to slant at an angle, thus leading to the robot veering off the obstacle course.

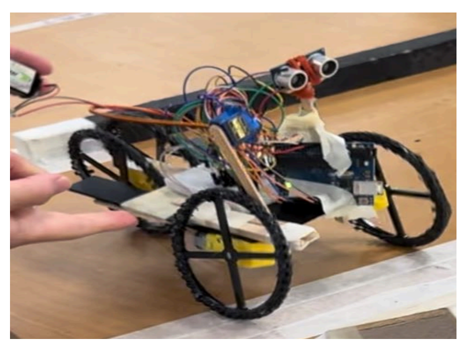
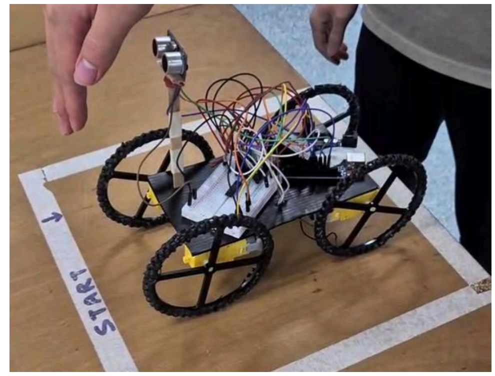


Figure 2.4 Initial Platform Design

Electronics / Motors The original two-wheeled robot design caused significant friction due to the rear section dragging against the ground, hindering mobility. To address this, we explored a four-wheel configuration to reduce frictional resistance, aiming to replace sliding friction with rolling friction and enhance the robot's overall movement efficiency (Figure 2.5). While our robot overcame all obstacles with the four-wheel methods we implemented, we were looking to fulfil the extra points of the ability to move back to its start position. While in our simulation, the robot was able to return to its start position, in practice the sheer number of wires resulted in poor connection to the motors, leading to inconsistent performance of the motors	 *Figure 2.5 Initial Motor Design*

3. Final Design and Solutions

Features / Solutions	Diagrams
Wheels The hollowed-out center of the wheels reduced the weight while the four spokes ensured the structural integrity of the wheels. The rubber mat surrounding the wheels attached with super glue increases the wheel's friction, ensuring the robot can overcome all obstacles. Ice-cream sticks are added as a support for the wheels to ensure that the wheels are straight and stable (Figure 3.1). By connecting the electrical terminal of the DC motor to each output of the LD26 electrical component, we are able to make the robot move forward and backward by reversing the direction of the current.	 *Figure 3.1 Final Wheel Design*

Catapult
Made of ice cream sticks which are folded into many segments and the shape is fixed into a bowl with tape.

For the vertical section of the catapult, two sticks are banned in series and inserted vertically into the platform, allowing for a seamless launch of the ball over the wall. This assembly is securely attached to the servo motor using a screwdriver, with a screw carefully driven through the stick to ensure a stable and reliable connection (Figure 3.2).

For the horizontal section of the catapult, three sticks are neatly overlapped and tightly wrapped together, creating a sturdy and stable tower. Atop this structure, the servo motor supporting the vertical section is securely hot-glued in place, ensuring it remains firmly attached and resistant to movement.

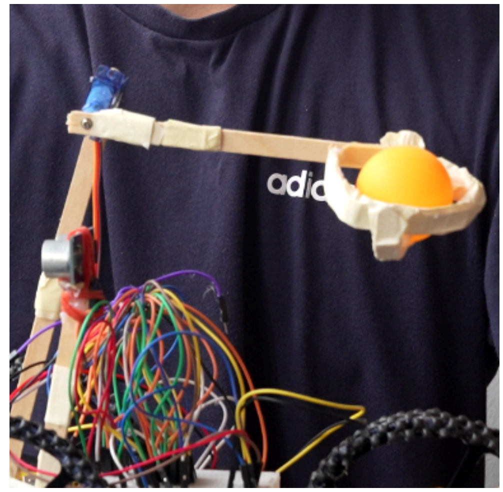


Figure 3.2 Final Catapult Design

Ultrasonic Sensor
The final design featured the Arduino sensor positioned atop a T-frame constructed from ice cream sticks, elevating the sensor for optimal performance (Figure 3.3). After several test runs, we employed rubber bands to fine-tune the height of the ultrasonic sensor, minimizing the likelihood of false detections and enhancing overall accuracy.

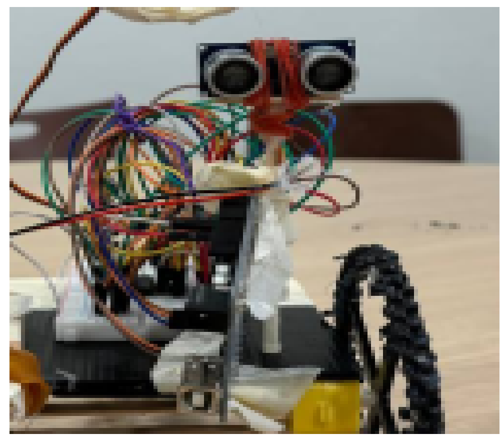


Figure 3.3 Final Ultrasonic Sensor Position

Arduino Code & Circuit
The logic of our code is as follows
- **Initial Movement**: Set the robot to move forward.
- **Obstacle Detection**: If the robot detects a wall **and** forward time >= 200 (to reduce false detections from ramps), stop the robot.
- **Catapult Launch**: Upon meeting the condition, trigger the servo to launch the catapult.
- **Reverse Movement**: After a short delay, reverse the electrical of the two signals to move the robot backwards.

The Innovation Point in our Circuit Design:
- **Simplified Motor Control**: By connecting two inputs of the H-bridge and wiring its outputs to the motor, we reduce control to just two Arduino pins.
- **Efficient Direction Control**: Setting one pin to HIGH and the other to LOW drives the motor forward, while reversing the states

Refer to the Appendix 4.4. Arduino Source Code for the full Arduino source code and Appendix 4.3. TinkerCAD diagram for our full Circuit Design

switches the direction. ● **Innovation Advantage**: This approach minimizes wiring complexity and optimizes pin usage, making it a streamlined solution for bidirectional motor control.	
Platform To address the slant of the wheels, we opted to hot glue two pieces of ice cream sticks to the front motors. This modification not only enhanced the structural integrity of the robot but also ensured proper alignment of the front wheels (Figure 3.4). This ensured the rigidity of the frame and the robot would travel in a straight line throughout the obstacle course.	 *Figure 3.4 Reinforcement of Platform*
Electronics / Motors The final design was a three-wheeled robot that utilises two motors in the front and a single motor driving two wheels at the back. The two wheels at the back guarantee the robot's stability while scaling the ramp. As we had reduced the number of motors, we decided to provide more power to each motor by directly supplying 9V to each motor. This increased their torque, allowing the robot to overcome the obstacles with ease. Additionally, this helped to further reduce the weight of the robot.	Refer to Appendix [4.3 TinkerCAD diagram]

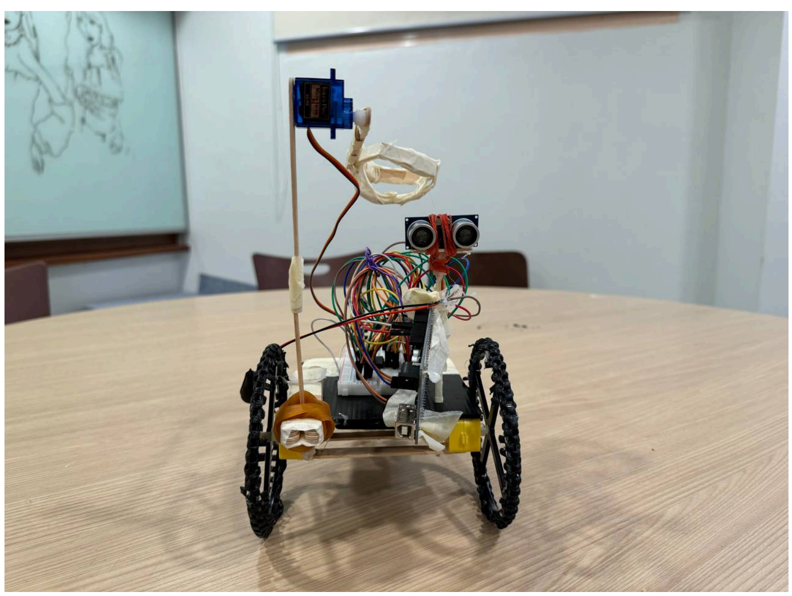
3.1. Reflections

This module provided valuable insights into integrating Arduino software with AutoCAD hardware design to prototype and build a robot from the ground up.

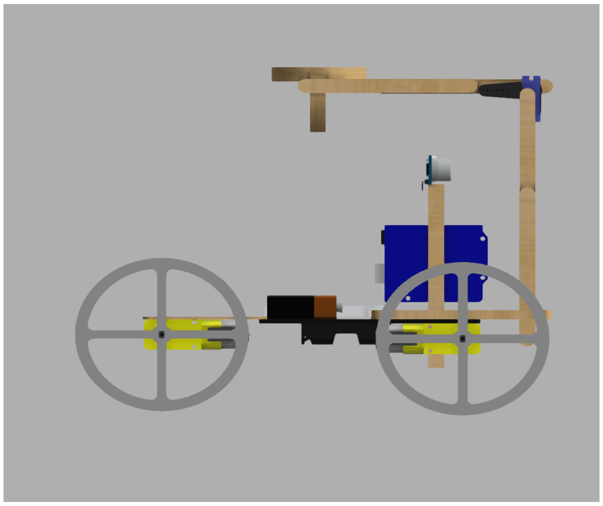
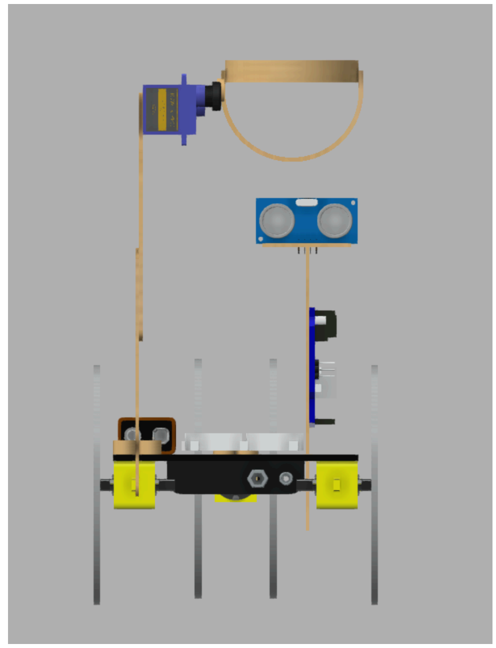
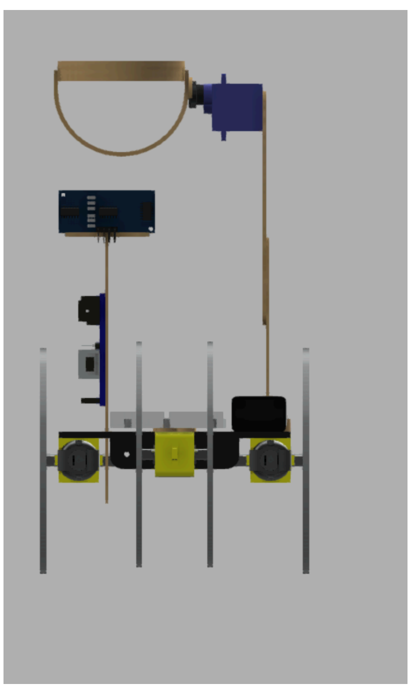
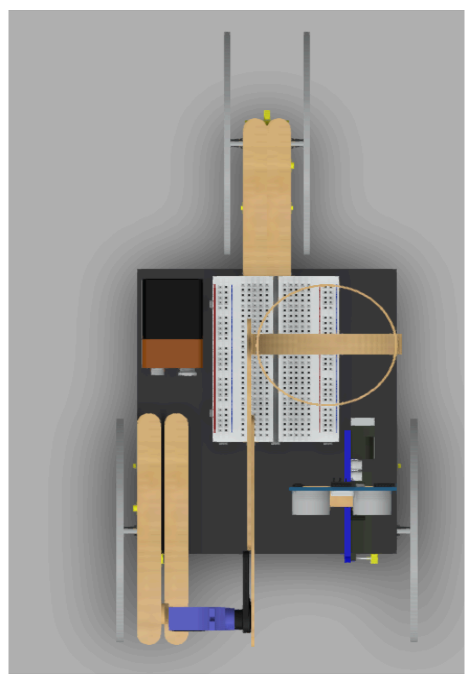
- **Practical Skills:** We gained foundational skills in CAD modelling with Fusion 360 and programming with Arduino, bridging software and hardware to develop functional prototypes.
- **The importance of prototyping:** Through iterative prototyping, we identified and addressed design challenges, using a continuous feedback loop to enhance our robot's functionality and ensure it met all objectives. This approach allowed for ongoing refinement toward an optimal design.
- **Collaboration:** This module emphasized effective teamwork, guiding us to enhance group efficiency through collaboration. When different ideas emerged, we evaluated each one's practicality—particularly given the constraints of time and resources in robotics. This experience taught us how to work cohesively to make informed decisions under these limitations that can maximize our project outcomes.

4. Appendix

4.1. Photograph of the final physical robot

Right View
 *Figure 5.1 Side View of Final Physical Robot*
Front View
 *Figure 5.2 Front View of Final Physical Robot*

4.2. CAD rendering of the final robot

Right View	Front View
 *Figure 5.3 Side CAD diagram of final robot*	 *Figure 5.4 Front CAD diagram of final robot*
Back View	**Top View**
 *Figure 5.5 Back CAD diagram of final robot*	 *Figure 5.6 Top CAD diagram of final robot*

4.3. TinkerCAD diagram

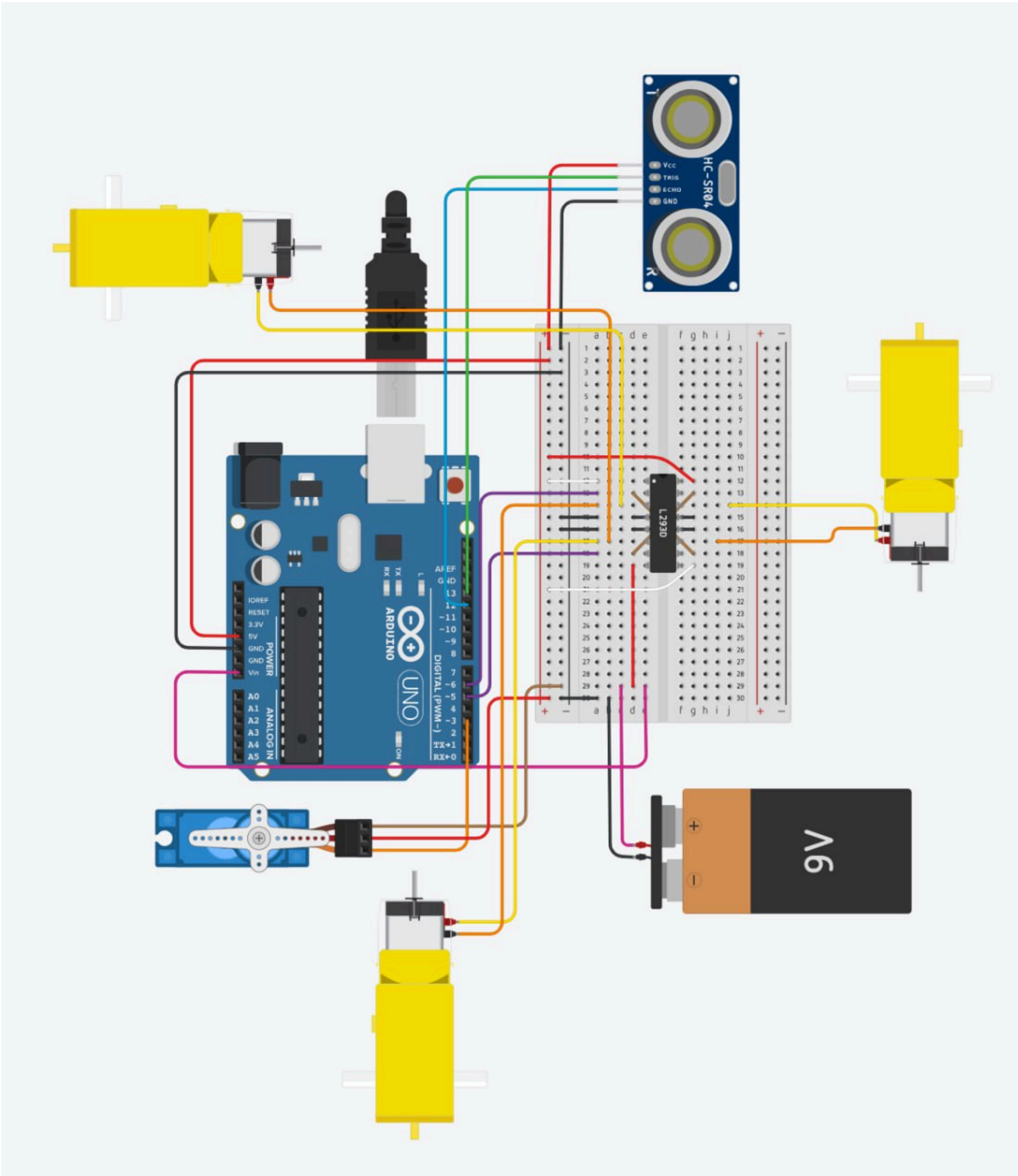


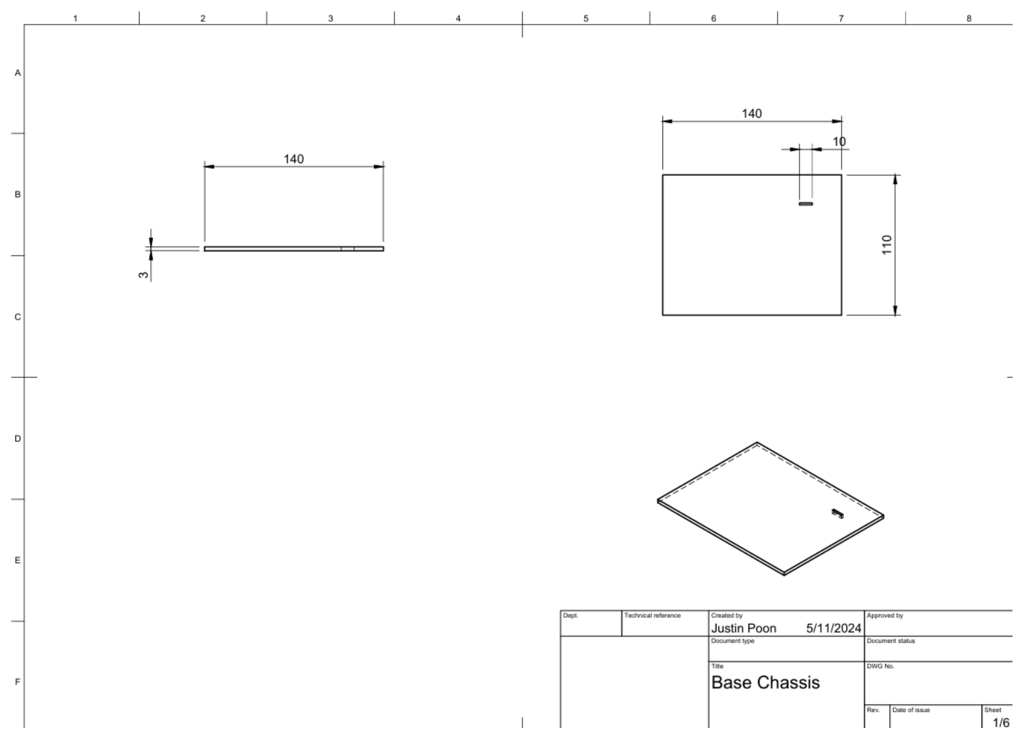
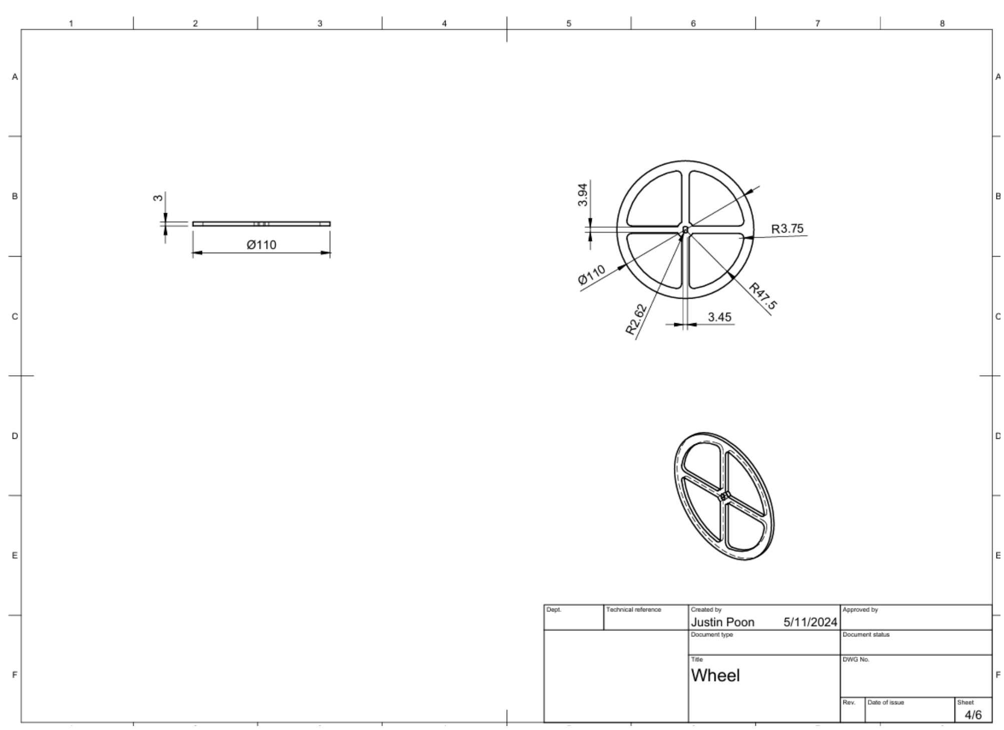
Figure 5.7 Final TinkerCad Design

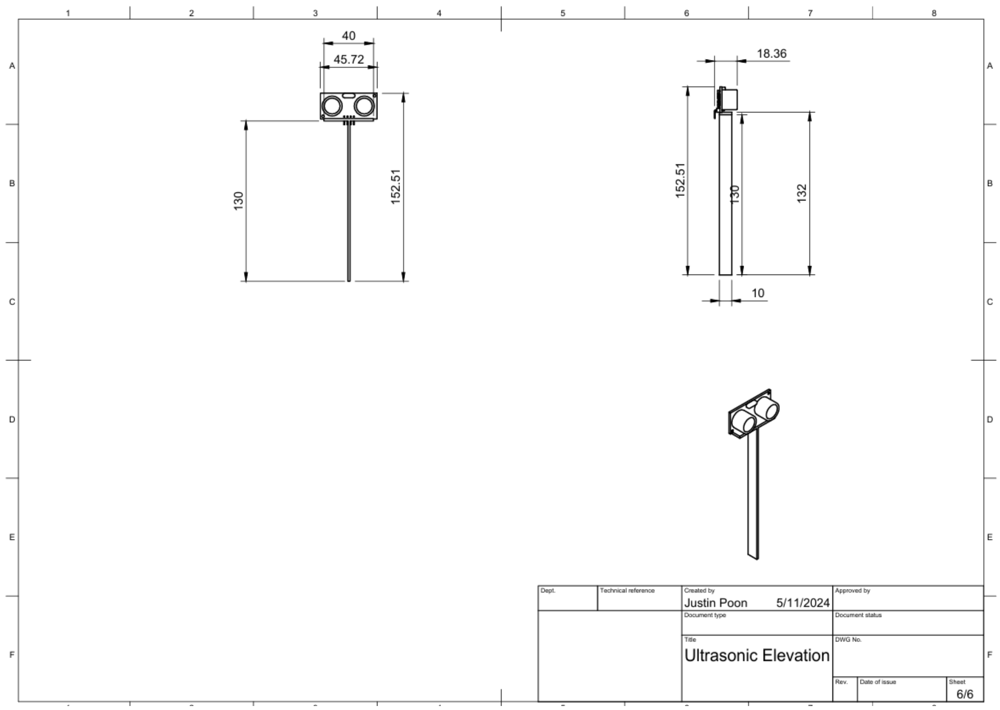
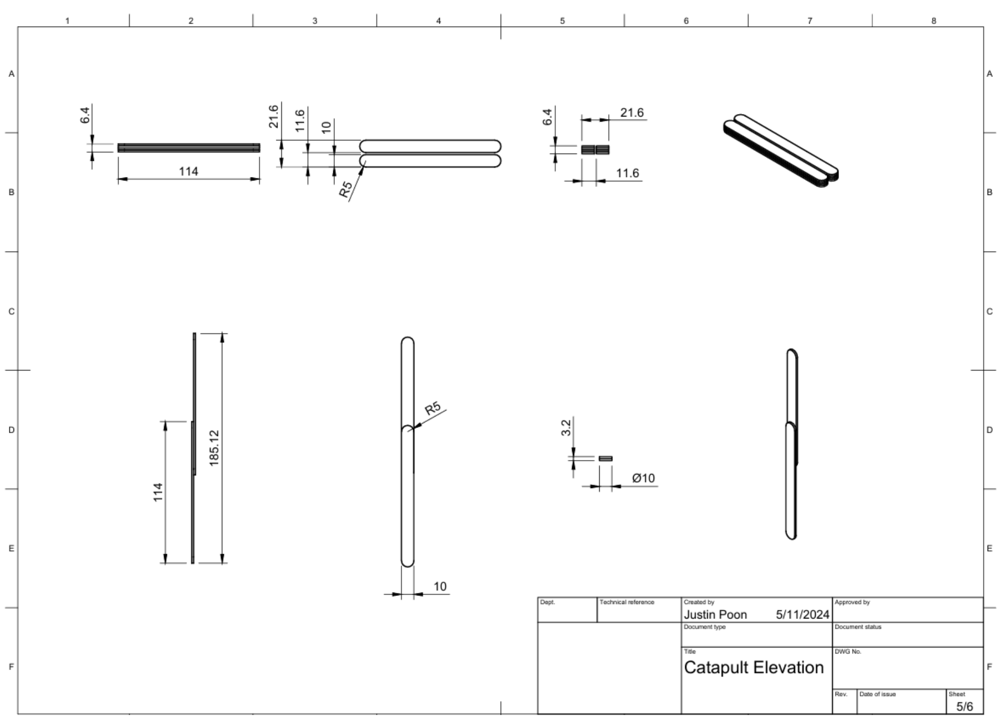
4.4. Arduino Source Code

```
#include <Servo.h>
int TRIG_PIN = 13;
int ECHO_PIN = 12;
int MOTOR_PIN1 = 5;
int MOTOR_PIN2 = 6;
int SERVO_PIN = 3;
float SPEED_OF_SOUND = 0.0345;
bool is_backward = false;
unsigned long forward_time;
Servo servo;

void setup() {
  // Set all the motor pins to be out
  pinMode(MOTOR_PIN1, OUTPUT);
  pinMode(MOTOR_PIN2, OUTPUT);
  digitalWrite(MOTOR_PIN1, LOW);
  digitalWrite(MOTOR_PIN2, LOW);
  // Initialize forward time
  forward_time = 0;
  // Set servo
  // setup the servo at 0 degrees because otherwise by default, it will
be set to 90 degrees.
  // use servo.attach(SERVO_PIN, num1, num2) to set the pulse width for
0 degree and 180 degree separately
  servo.write(0);
  servo.attach(SERVO_PIN);
  // Set ultrasonic sensor
  pinMode(TRIG_PIN, OUTPUT);
  digitalWrite(TRIG_PIN, LOW);
  pinMode(ECHO_PIN, INPUT);
  Serial.begin(9600);
  Serial.println(is_backward);
}

void loop() {
  // Ultrasonic Sensor
  digitalWrite(TRIG_PIN, HIGH);
```

```
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  int microsecs = pulseIn(ECHO_PIN, HIGH);
  float cms = microsecs * SPEED_OF_SOUND / 2;

  // Control Logic
  if (cms < 15 && !is_backward && forward_time >= 200)
  {
    digitalWrite(MOTOR_PIN1, LOW);
    digitalWrite(MOTOR_PIN2, LOW);
    // Wait for 2 seconds
    delay(2000);
    // Shoot the ball
    servo.write(90);
    delay(1000);
    // Move the catapult
    servo.write(0);
    delay(2000);
    is_backward = true;
  }
  else
  {
    // Move Forward
    digitalWrite(MOTOR_PIN1, HIGH);
    digitalWrite(MOTOR_PIN2, LOW);
    forward_time++;
  }
  if (is_backward)
  {
    // Move Backward
    digitalWrite(MOTOR_PIN1, LOW);
    digitalWrite(MOTOR_PIN2, HIGH);
  }
  delay(10);
}
```

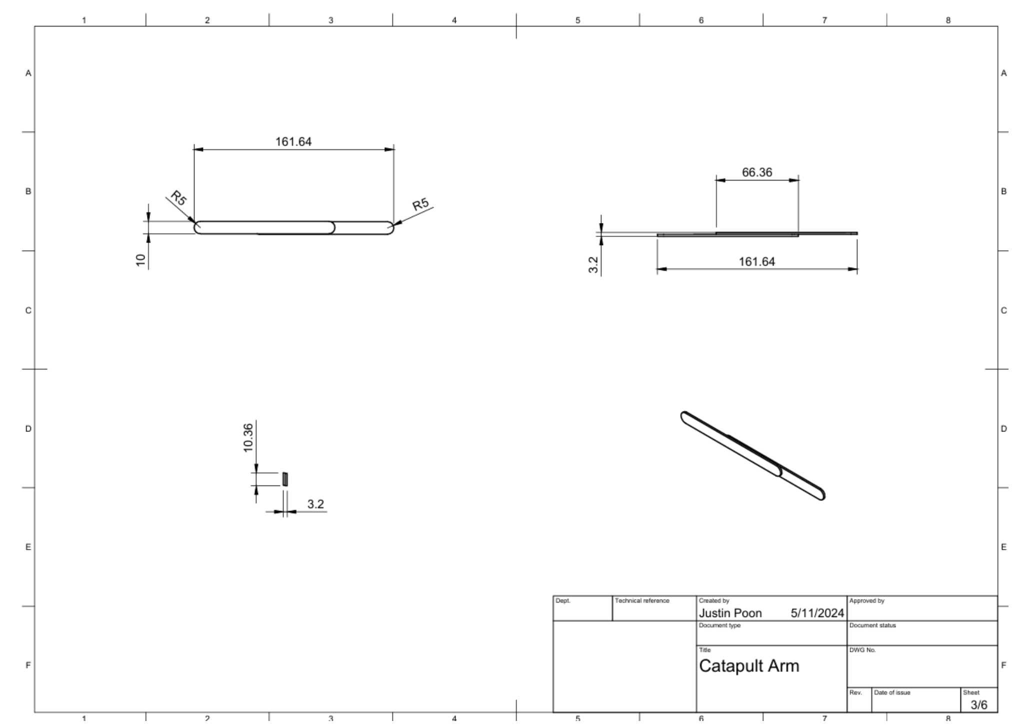
4.5. 2D CAD drawings for each component

Components	Measurements
Base Chassis: Material: Polypropylene Cutting Method: Penknife Dimensions: 140mm by 11mm	
Wheels: Material: Acrylic Cutting Method: Laser Cutter Dimensions: Diameter of 110mm	

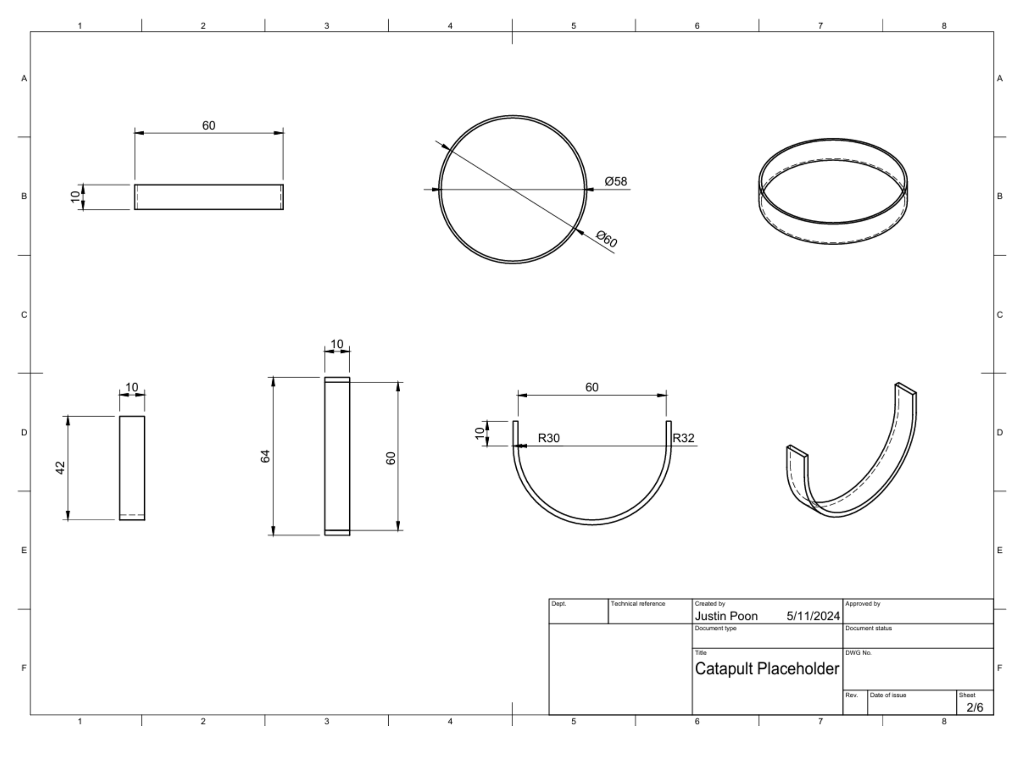
Ultrasonic Sensor Elevation: Material: Ice-Cream Stick Cutting Method: NIL (used tape to banned them together) Dimensions: 130mm	
Catapult Elevation: Material: Ice-Cream Stick Cutting Method: NIL (used tape to banned them together) Dimensions: 165 mm	

<table>
<tr>
<td>Catapult Arm:
Material: Ice-Cream Stick

Cutting Method: NIL (used tape to band them together)

Dimensions: 161 mm</td>
<td>
</td>
</tr>
<tr>
<td>Ball holder:
Material: Ice-Cream Stick

Cutting Method: Penknife (used tape to band them together)

Dimensions: Diameter of 60mm</td>
<td>
</td>
</tr>
</table>