EG1311
DESIGN AND MAKE
FINAL REPORT

GROUP B15, TEAM 2
YUANCHENG ELIJAH TAN
SURESH RUBESH
LIM QIN JIE
KHAIRUL HAZIQ B MOHAMAD HISHAM
PAUL JOASH TAN TSANG KIM
LAETISHA JANE VANJOO

INDEX OF CONTENTS

1 – PREAMBLE

For this project, our team was tasked with creating a robot that was able to traverse a pre-defined obstacle course, and this report articulates all such efforts and measures undertaken in the design and engineering process, including lessons that we had learnt during the process of developing this robot. This report encapsulates the whole process of development and elaborates on all the various areas of engineering considerations that were involved, inclusive of but not limited to – Electrical, Mechanical, Software, and Design. In this report, we shall attempt to document our process, from the template project at the very beginning, to the final run of the robot.

2 – INTRODUCTION

We had utilized multiple design prototypes for each component and a variation of combinations of each component was used in the making of a general rough prototype. Thus, the assembly of each prototype of the robot had many sub-prototypes pertaining to variations in these individual components. Generally, there were 4 main stages to the making of the robot. Stage 0 was the simple assembly of the template project given to us. Stage 1 was the design of the main vehicular body of the robot, and Stage 2 was the incorporation of the ball-deployment mechanism into the main body of the robot. Stage 3 was the finalization and optimization of the robot to prepare for the graded run and submission.

3 – STAGE 0: TEMPLATE PROJECT

PROTOTYPE 0

Given the template project, we had to understand and decipher the Arduino code, and the circuitry first before assembling the robot, which itself was a straightforward process. The template code stopped the robot 30 centimeters away from the wall and thus, we had to modify the code to stop the robot 5 centimeters away as required by the project specification. The robot could move but was veering off the path. We considered the different aspects of an ideal robot, like traction, obstacle height, structure, weight distribution, and started work around those aspects.

4 – STAGE 1: MAIN VEHICLE

PROTOTYPE 1.1 – 2 Wheel Cardboard Design

We attempted to build from the template project and its design, without implementing too many changes off the bat. We decided to stick to the 2 wheel design, however, there were multiple issues with this design. The cardboard wheels had poor traction and structural integrity as when the robot attempted to climb the slope, it was stuck at the base and the wheels were bending slightly, so we attempted to replace the wheels with layered cardboard wheels, and attempted to add rubber bands around the wheels to provide for more traction. However, the rubber bands could not be fully secured to the wheels as the elastic force of the rubber bands had caused the wheels to deform. (*Figure 6.1)*

PROTOTYPE 1.2 – 4 Wheel Cardboard Design

Our attempt at keeping it simple with 2 wheels had resulted in more problems and we decided to attempt a 4 wheel design. We attempted to fix the traction issue with more rotational power alone by adding 2 more motors, but the traction issue remained. Furthermore, hand-cut cardboard wheels each had slightly different diameters. The robot was veering off track and was unable to traverse straight. Code optimization was considered, but the perceived complications of that measure were greater than that of physically redesigning our robot.

PROTOTYPE 1.3 – 4 Wheel Solid-Acrylic Design
We decided to use 4 laser-cut solid acrylic wheels of 10cm diameter, to ensure uniform dimensions across all wheels. However, the traction problem caused by the wheels being too slippery had remained. We attempted to fix the traction issue with hot glue on the edges of the wheels but to our surprise, the glue did not adhere to the wheels as the material was too smooth. We circumvented this problem by using sandpaper to roughen up the circumference of the acrylic wheel, which allowed the glue to stick onto the wheels. Furthermore, given the larger 10cm radius of the acrylic wheels compared to the cardboard wheels, we had to reprogram the Arduino to stop at a distance further than that of the previous prototypes, so that the acrylic wheels do not make contact with the obstacle course wall, which was one of the grading criteria.

PROTOTYPE 1.4 – 4 Wheel Spoked-Acrylic Design
Learning from our previous prototypes, we distilled the important learning points of what had worked, and worked off of that. We kept the 4-wheel design, however, we changed the wheels to a hollowed out design with 4 spoke design. We added groves to increase the contact surface area with the obstacles, and sanded the wheels in the same way as before so that the hot glue would be able to adhere to it. Due to limited availability of the laser cutter, we were unable to fully print 4 wheels. Our prototype worked despite only having 2 of the newly designed wheels, when combined with 2 of the previous solid wheels. However, the 4 spoked wheels ***(Figure 6.3)*** were too fragile and one of the spokes of a wheel cracked upon attempting to insert it into the axel of a geared motor. Thus, we decided to use a new set of wheels using all the lessons learnt to use for our final wheel design, since we were short of 2 wheels anyway.

PROTOTYPE 1.5– 4 Wheel Spoked-Acrylic-Reinforced Design
Additionally, our initial design for the body was to incorporate an acrylic frame to provide structure and stability to the robot. However, upon testing, we found that a simple carboard body provided sufficient structure for the robot, despite carrying the weight of 4 motors. As the cardboard body was much lighter than the acrylic body, we decided to use the cardboard body.We also printed 4 of 8 spoked reinforced wheels (Figure 5.4) which we retained as our final wheel design. The robot was finally able to move without any problems and it completed the obstacle course successfully. Stage 1 was now complete.

5 – STAGE 2: MAIN VEHICLE + ARM

PROTOTYPE 2.1 – Spring-Loaded Catapult
How it worked: The servo 'blade' would be resting on the long arm of the catapult. There is elastic potential energy from the arm being stretched to the base. When the ultrasonic sensor detects that the robot is near the wall, the servo blade would turn 90 degrees. The arm is released from its tensioned state, converting elastic potential energy into kinetic energy. For our preliminary testing, we used our fingers to push the arm toward the base and release it. Based on this test, there was enough tension for the ping pong ball to go over the wall. Thus, we assumed that using the servo motor instead of manually using our fingers would work in a similar way.

Difficulties: We had issues trying to secure the servo blade to the arm of the catapult. The servo blade could not latch on to the catapult (arm slipped away from the blade) and there was not enough tension, causing the ping-pong ball to be unable to propel far enough. The walls of the basket containing the ping pong ball were too short which caused the ping pong ball to fall off even before reaching the end point.

Lesson learnt: There were limitations to the parts we were given in the project that we did not consider initially. Our prototype might have worked if the servo blade was longer and/or sturdier. However, we had to remember to work within the constraints of the project. Thus, we scrapped the idea and went for a different approach.

This prototype was developed concurrently with the main body of the robot. However, given the lack of success with this prototype, we did not reach the stage where this was to be incorporated into the main body, and we had changed the concept entirely.

PROTOTYPE 2.2 – Servo-Powered Mechanical arm
From prototype 2.1, we took upon the learning factors and redesigned the system to bring the ping pong over instead of relying on tension. The basket is initially resting on the robot, with the ping pong ball inside. When the ultrasonic sensor detects that the robot is near the wall, the servo blade would turn 135 degrees. This will rotate the arm and bring the basket containing the ping pong ball over the wall. The ping pong ball will roll over the wall when the basket is over the wall and facing downwards. Structure considerations: We needed a way to secure the delivery system to the body of the robot. Since the base was made of cardboard, we could easily cut some slits on it where we could insert the walls of the delivery system. We used masking tape and hot glue to further secure the system. The overall structure of the delivery system was flimsy due to a hollow base.

We resolved it by gluing rectangular pieces of corrugated board in between the two pillars to ensure the base is rigid. We also increased the height of the basket to ensure the ping pong ball is secured throughout the journey. Servo considerations: As the servo would be fixed in place, we had to ensure the servo was set to the same position every time the loop restarted. We solved this issue by testing specific angles for the servo with the Arduino code, specifying that angle in the void_setup() section of the code. Our servo had 180º of motion, and the arm was moving downwards in a clockwise direction instead of the required anti-clockwise direction. Thus, before fixing the servo to the vertical structure, we flipped the servo placement. To ensure that the ball rolls off at the optimal angle and distance without any component touching the wall. Taking note of the issues we faced in our previous prototype, we managed to improvise and come up with a much more efficient and reliable system. With the reliability that this new catapult design provided, we decided to go with it as our final prototype.

6 – STAGE 3: FINALIZATION

PROTOTYPE 3.1 – Penultimate Prototype
We affixed the arm to the front of the robot, right before the ultra-sound sensor, to offset the weight of the bread board and the arduino at the rear. The robot was able to complete the course, and the mechanical arm design was able to deploy the ball sufficiently, **both** individually. For prototype 2.2, the members working on the mechanical arm were working on a separate Arduino so that prototyping for stage 1 and 2 could be done simultaneously and concurrently. The servo was actuated by the Arduino code itself, without requiring external input such as that which is required by the final submission, the Ultra-sound sensor. The challenge was to incorporate the Arduino code of the servo for the mechanical arm into the existing code for the main vehicle.

The problem was that the robot had no mechanism to deploy the ball independently of the stopping of the robot, which itself was done so by signals read from the Ultra sound sensor.

We had to reprogram the Arduino such that upon receiving signal from the Ultra-sound sensor, both the wheels would stop and the arm would be deployed. It was possible to add a servo.write code together with the digitalWrite code within the conditional *if* statement, however, this presented 2 problems. Firstly, we were using the ultrasound sensor as a stopping mechanism, to stop the robot before we placed it on the given that we had no electrical means to disrupt power to the robot without removing the battery. With the addition of the arm, this would mean that the ball would be deployed immediately upon the start of the course. Secondly, the arm needs to return to its original position after each test run.

We fixed these issue by adding a nested conditional if statement (**Figure 4, Line 41**) within the existing if statement providing a buffer period. Using the millis() command, we stored a variable that reflected the duration for which the Arduino has been active. Both conditions, i.e. ultra-sound distance < 5 cms and buffer time > 10 seconds need to be fulfilled for the arm to be deployed.

Design issues – Upon testing the robot, we realised that while the robot successfully completes the run most of the time, the last few trials had been near misses. Firstly, we had realized that the increased weight of the robot had made the main cardboard structure bend slightly toward the middle, causing the wheels to flare sideways slightly. Secondly, due to wear, the motors was starting to come unstick from the body, contributing to the flaring of the cardboard body as well. Thirdly some wire connections between the motors and the breadboard had also come loose, causing one of the motors to occasionally lose connectivity from the Arduino via the breadboard. Thus we decided to re-assemble the robot.

PROTOTYPE 3.2 – Final Prototype

Based on the lessons learned thus far, we decided to reassemble the robot one last time. The breadboard was affixed to the body with the adhesive built in, the Arduino Uno was taped to the body, the motors were glued to the body securely and with as much precision as possible, and with the distance between each pair of motor decreased, the Arm of the robot was glued to the body more securely, the ball container was also glued to the arm, the wheels were affixed now closer together, and all wires were securely attached to the breadboard and tested with every connection. The final trouble we had was that, by the design of the materials provided, the battery could not be secured directly to the robot, and it had to be able to be taken out with ease. In one of the trials runs, the battery being too far in the front of the body had caused the weight to be distributed too close too far in front of the robot, and it had toppled over after crossing the second obstacle. We fixed this by simply moving the battery behind in our trial runs.

OPTIMISATION

After seeing other groups' runs, we realized that most others had used acrylic bodies which was much heavier than our cardboard body and thus we could contend for the lightest robot. We decided to make some last-minute adjustments, changing our ball basket from cardboard to masking tape, and removing pieces of the bread board. With this we managed to cut down 15 grams, bringing our robot down to 308 grams. ***(Figure 6.2)***

7 – LESSONS LEARNED

SOFTWARE

Problem 1: The servo motor behaved unexpectedly. The swinging arm is supposed to turn 180 degrees from the base of the robot towards the top. However, during our testing, one of the

problems of the servo motor was that it either did not swing at a sufficient angle or it swung beyond the intended angle. This was despite following the online Arduino documentation for the servo motor, using the "servo.write()" method.
Fix 1: By trial and error, we systematically input different angles into the Arduino code that was responsible for turning the motor until eventually, the arm swung at the correct angle.
Lesson 1: We learnt how to troubleshoot technical problems despite the lack of clear documentation available for some of our components. We overcame this problem by experimenting with different configurations of the code on our own and incrementally getting closer to (and eventually) solving a problem.

MECHANICAL

Problem 1: Initially, the wheels could not grip the surfaces strongly enough and the robot had a little difficulty smoothly manoeuvring the obstacle course.
Fix 1: Taking the design of the wheels in off road vehicles, we added grooves and sanded the wheels to improve the traction and reduced the wheel's weight. This resulted in enough grip to smoothly go over the obstacle course.
Lesson 1: Make use of existing features and things around us to find solutions to our problems.

DESIGN

Problem 1: Initially, certain components of the robot were heavy and the structure of the pillar securing our catapult system was flimsy.
Fix 1: We remove excess unwanted parts and reinforce the pillar of the catapult system by adding square pieces of cardboard to act as support to keep the structure rigid.
Lesson 1: Avoid overcomplicating our design and go with the simplest method first before implementing further features to avoid unnecessary complications

GENERAL

Problem 1: There was uncertainty about the division of labour in our project. Not all of us had proficiency in the skills required by each part of the project. This made it challenging to distribute the workload efficiently among our teammates.
Fix 1: We discussed each of our individual skillsets and split the tasks among us based on who had the most experience and proficiency and therefore, confidence in completing the task.
Lesson 1: We must consider each person's strengths and abilities before assigning them a task to ensure that each teammate is working to their full potential. Additionally, we also learnt that working together in a team can produce better results than working alone as everyone has a unique skillset that can bring something to the table.

8 – Conclusion

The lessons we had learned along the way were of a multifaceted nature. On top of the directly affecting aspects of making the robot, such as Mechanical, Electrical, Software, Designing, we had also learned other crucial things like the importance of work delegation, the efficiency of simultaneous and concurrent development of prototypes to fully utilize all available manpower, the importance of accuracy and precision in design, and most importantly, the virtue of accepting, understanding, and learning from our mistakes with a positive mindset and approach such that we are able to improve on them on our path to eventual success.

APPENDIX

Figures

1. Photograph of final physical robot
2. CAD rendering model for final robot
3. TinkerCAD circuitry diagram
4. Arduino source code
5. Fully dimensioned 2D CAD model for each component
6. *Additional* – Design & development reference figures

1 – Photograph of final physical robot

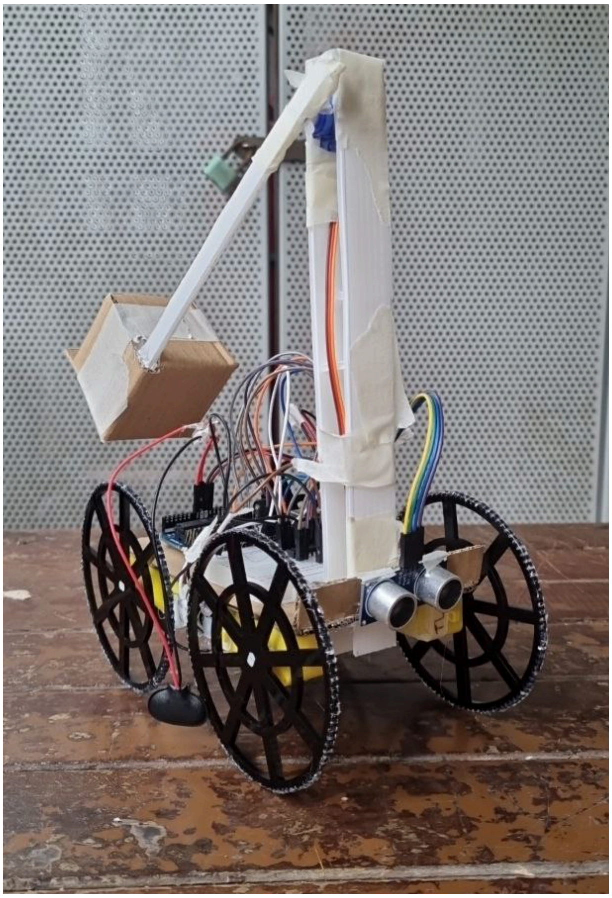

Figure 1.1 – Diagonal view of Robot

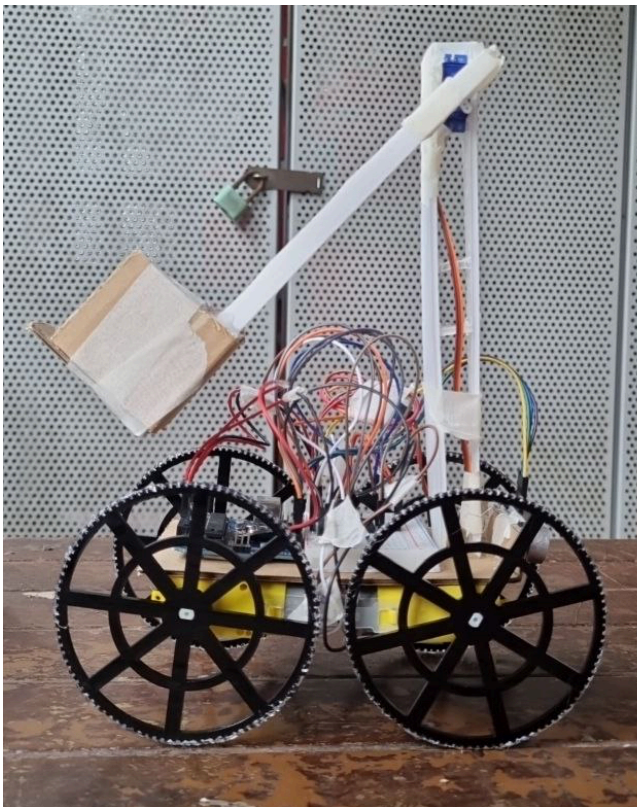

Figure 1.2 – Side view of Robot

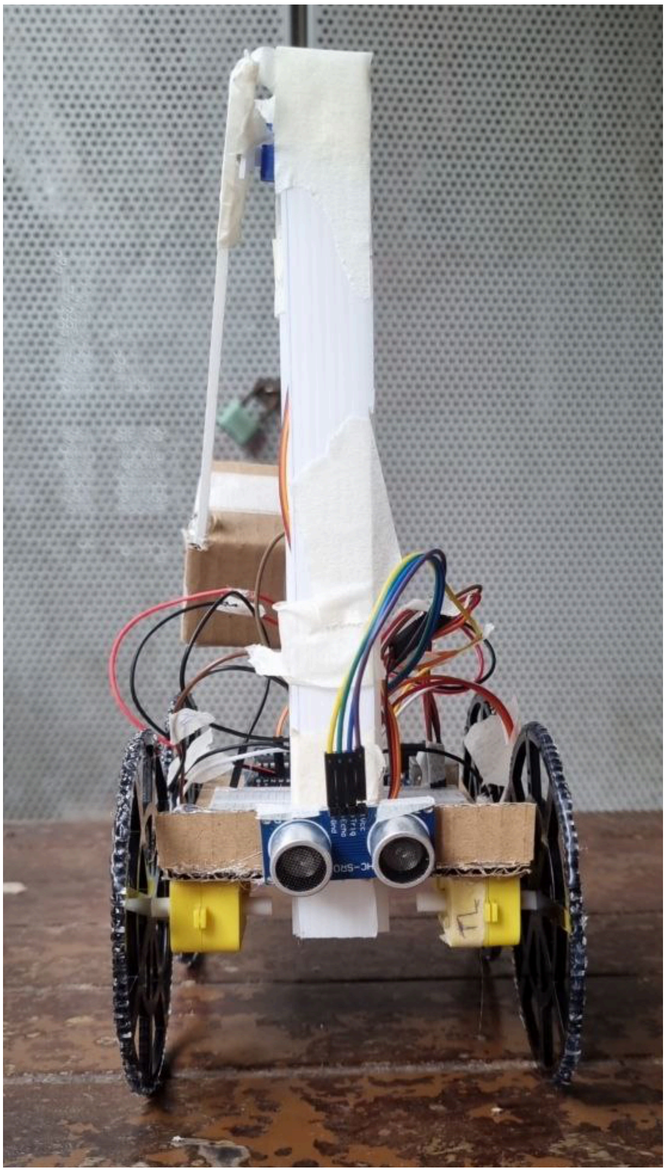

Figure 1.3 – Front view of Robot

2 – CAD rendering model for final robot

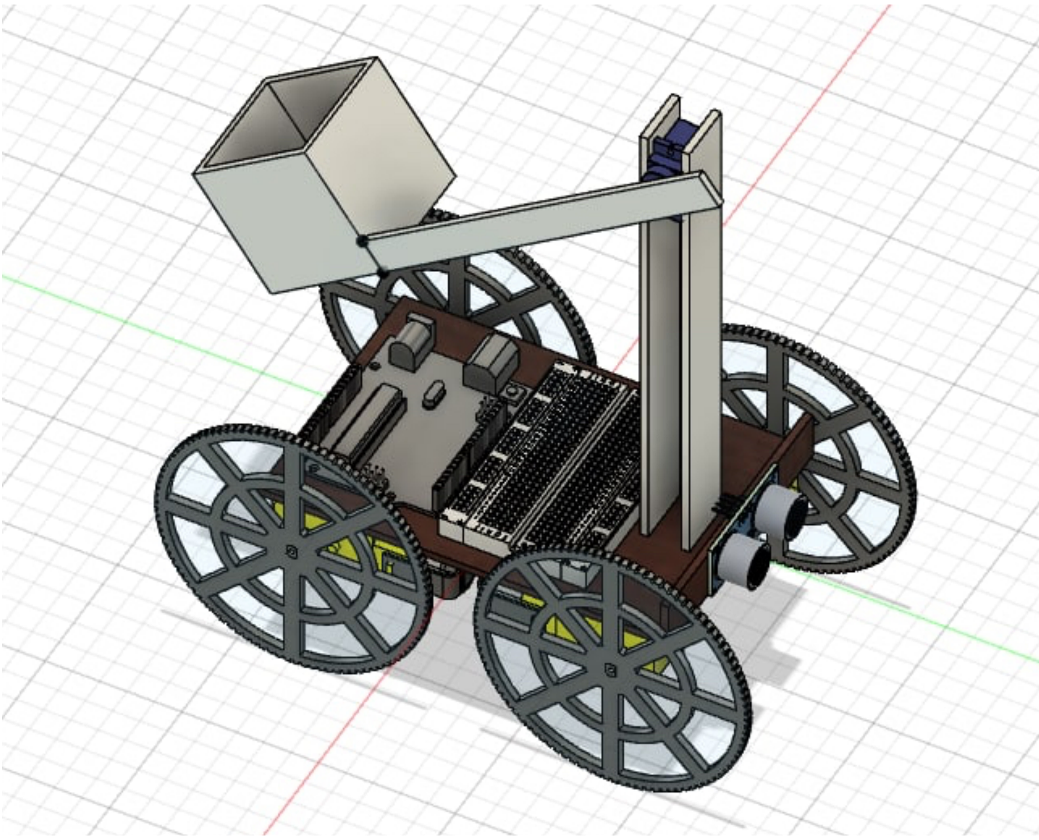

Figure 2.1 – Angular view of robot

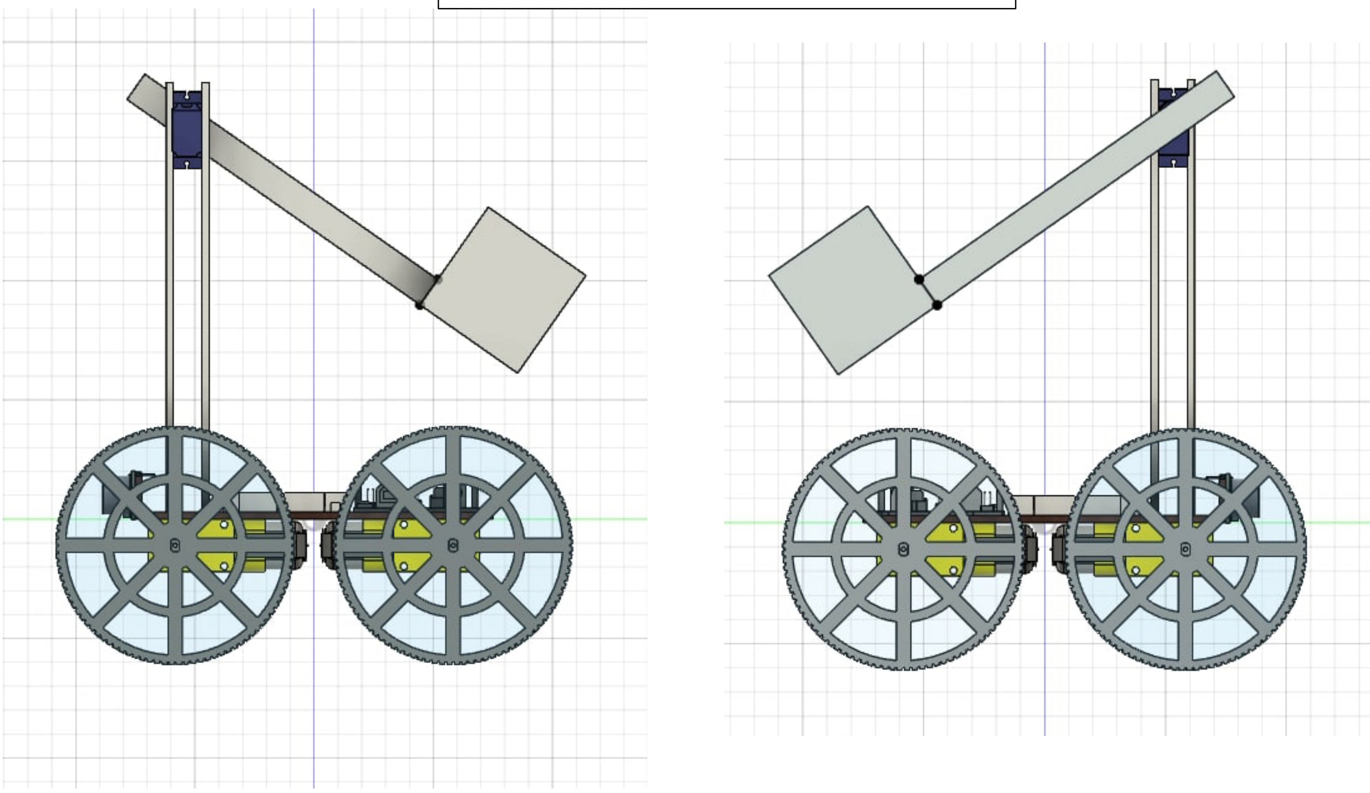

Figure 2.2 – Side view of Robot

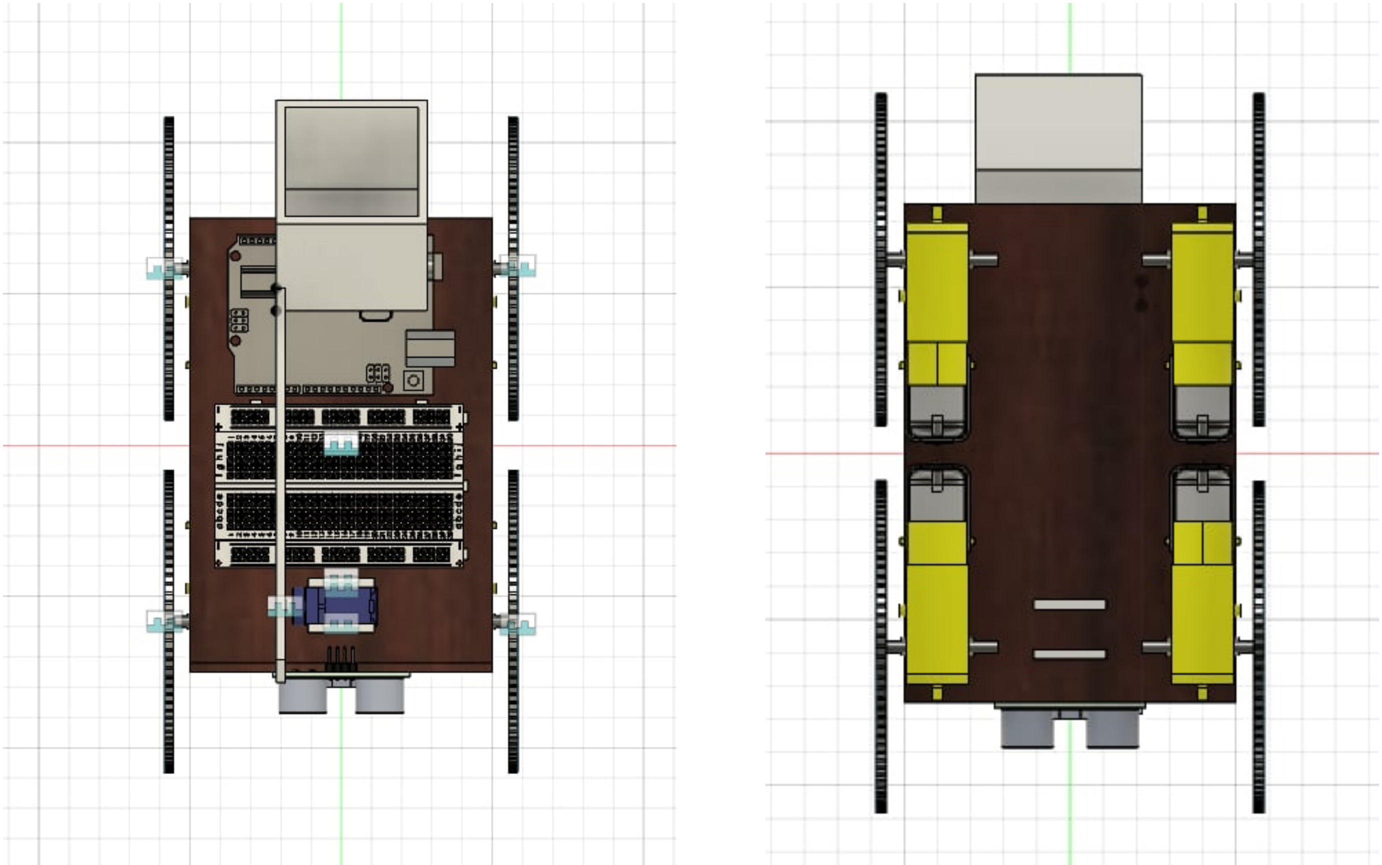

Figure 2.3 – Top and bottom view of Robot

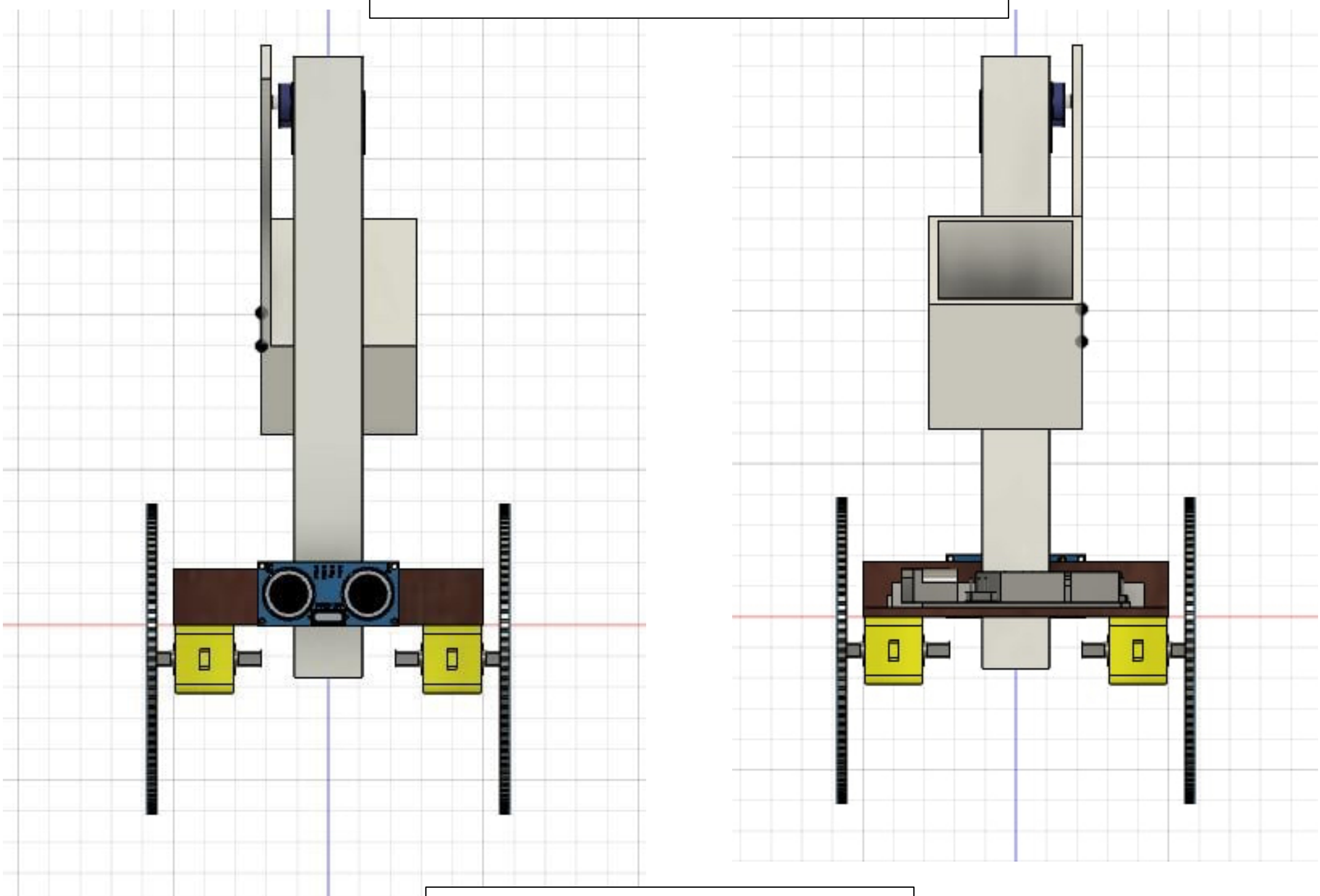

Figure 2.4 – Front and rear view

3 – TinkerCAD circuitry diagram

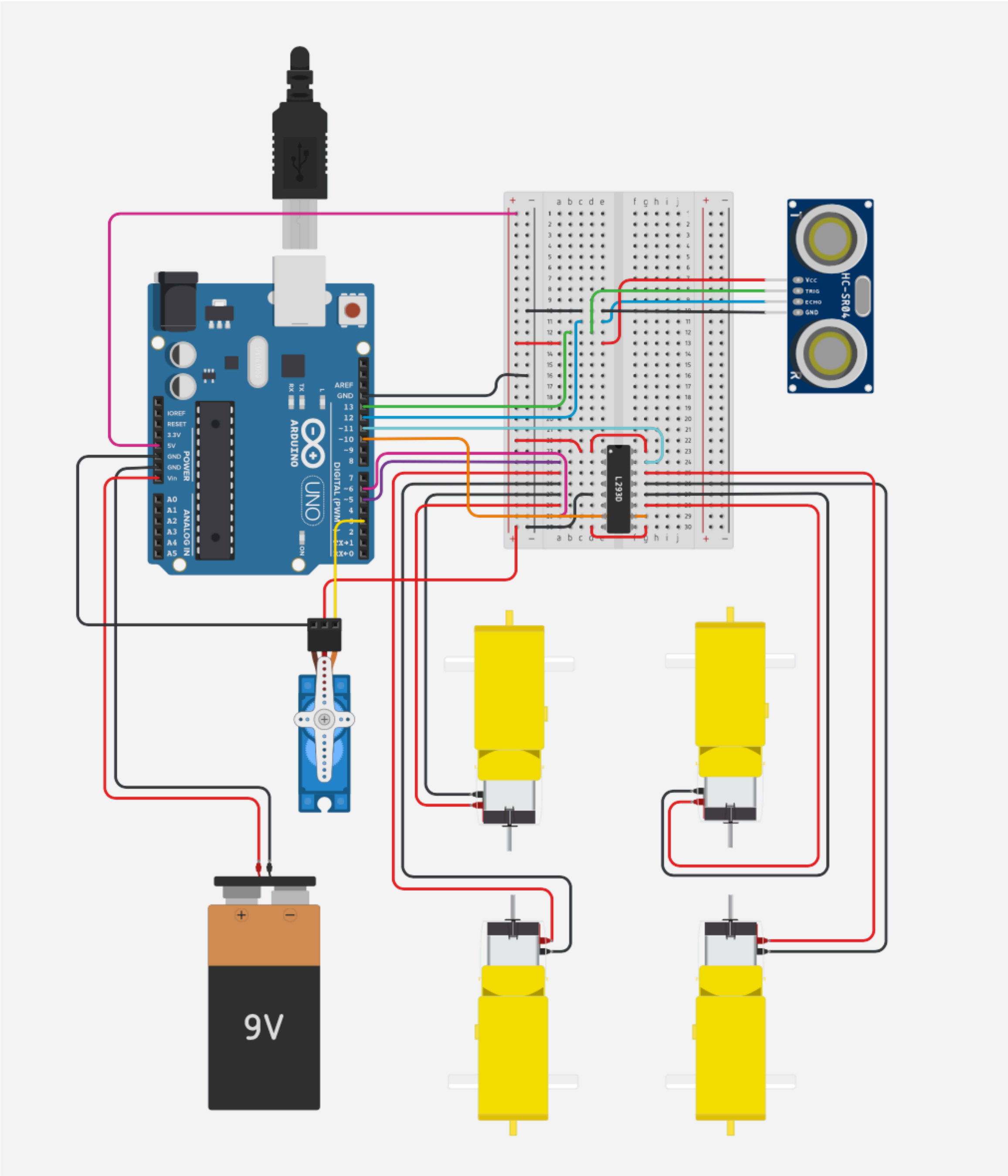

Figure 3.1 – TinkerCAD Circuitry Diagram

4 – Arduino Source Code

```
#include <Servo.h>
Servo servo;
int TRIG_PIN = 13;
int ECHO_PIN = 12;
int MOTOR_PIN1 = 6;
int MOTOR_PIN2 = 5;
int MOTOR_PIN3 = 10;
int MOTOR_PIN4 = 11;
float SPEED_OF_SOUND = 0.0345;
unsigned long Buffer_Time;

void setup() {
  pinMode (MOTOR_PIN1, OUTPUT);
  pinMode (MOTOR_PIN2, OUTPUT);
  pinMode (MOTOR_PIN3, OUTPUT);
  pinMode (MOTOR_PIN4, OUTPUT);
  pinMode (TRIG_PIN, OUTPUT);
  digitalWrite(TRIG_PIN, LOW);
  pinMode (ECHO_PIN, INPUT);
  Serial.begin (9600);
  servo.attach(3);
  servo.write(180);
  delay(100);
}

void loop() {
  Buffer_Time = (millis()/1000);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds (10);
  digitalWrite(TRIG_PIN, LOW);
  int microsecs = pulseIn (ECHO_PIN, HIGH);
  float cms = microsecs*SPEED_OF_SOUND/2;
  Serial.println(Buffer_Time);

  if (cms < 7) {
    digitalWrite (MOTOR_PIN1, LOW);
    digitalWrite (MOTOR_PIN2, LOW);
    digitalWrite (MOTOR_PIN3, LOW);
    digitalWrite (MOTOR_PIN4, LOW);

    if (Buffer_Time > 10){
    delay(500);
    servo.write(20);
    }

  } else {
    servo.write(180);
    digitalWrite (MOTOR_PIN1, HIGH);
    digitalWrite (MOTOR_PIN2, HIGH);
    digitalWrite (MOTOR_PIN3, HIGH);
    digitalWrite (MOTOR_PIN4, HIGH);
  }
    delay(10);
}
```

Figure 4.1 – Arduino Source Code

5 – Fully Dimensioned 2D CAD Model for each component

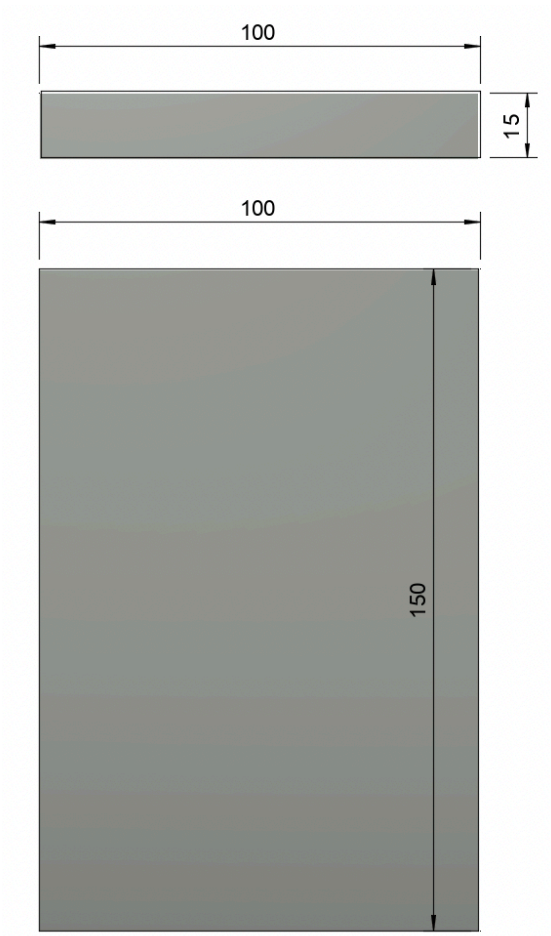

Figure 5.1 – Main Chassis (Cardboard)

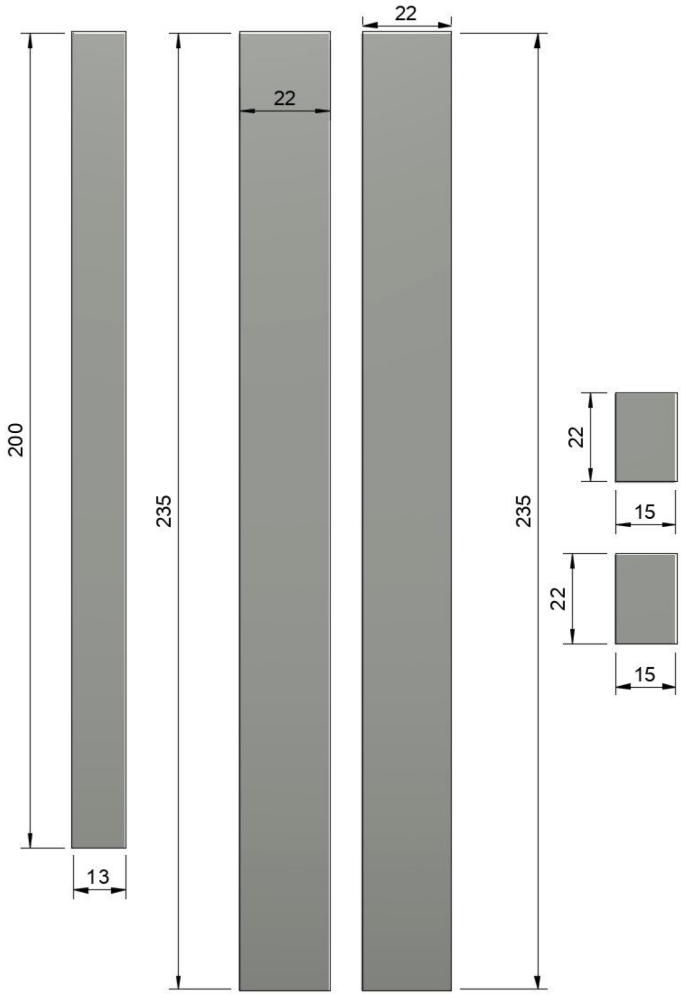

Figure 5.2 – Deployment Arm (Corrugated Board)

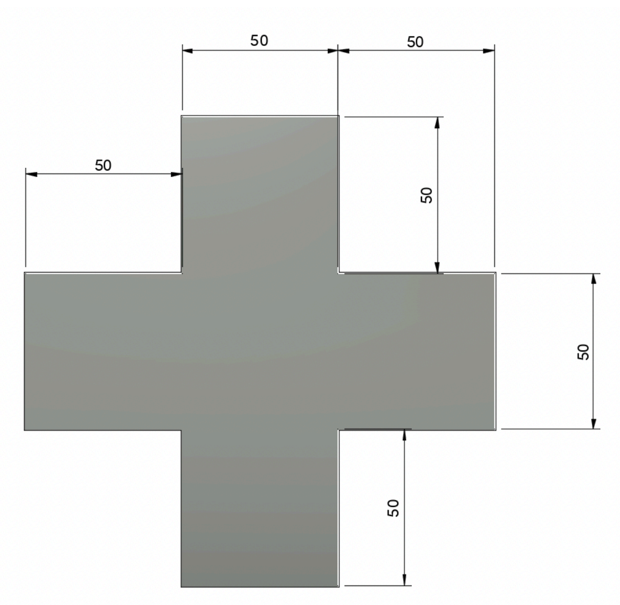

Figure 5.3 – Ball Basket (Cardboard)

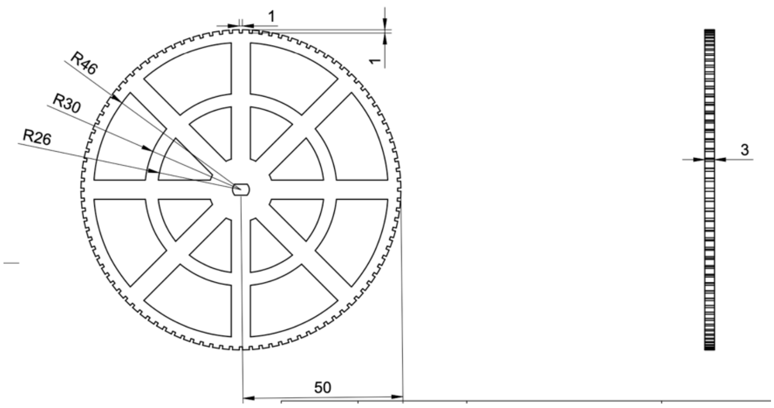

Figure 5.4 – 4x 8-Spoked Wheel (Acrylic)

6 – Additional Reference Diagrams

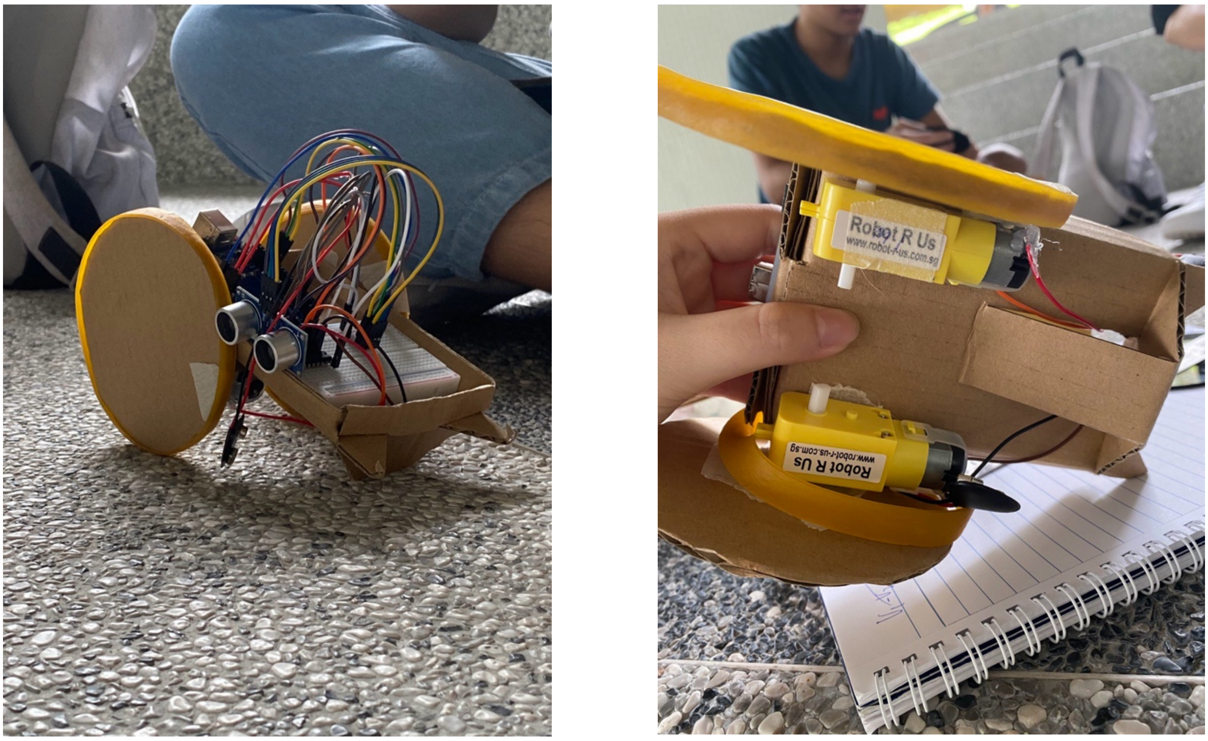

Figure 6.1 – Prototype 1.2

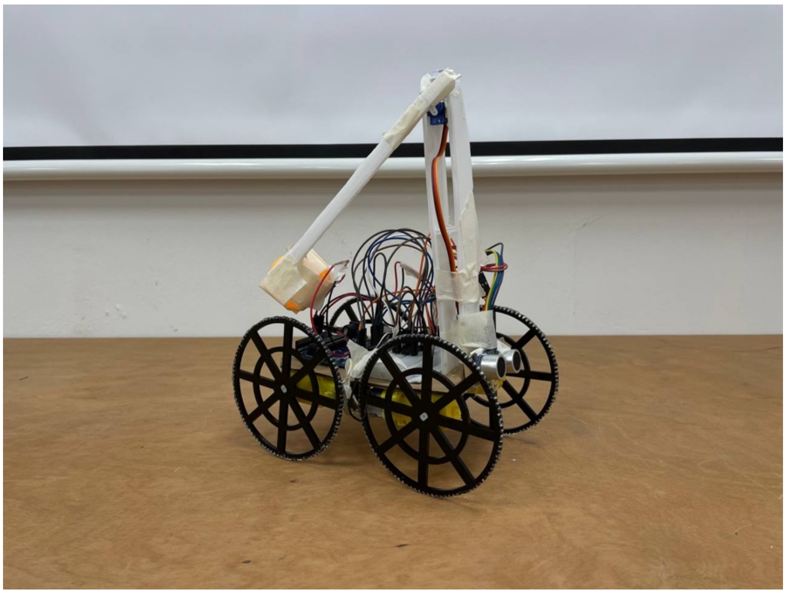

Figure 6.2 – Optimisation

Figure 6.3 – 4-Spoke Wheel