Table of Contents

Table of Contents 2
Introduction 3
Iteration 1 – Template Model Evaluation 3
Iteration 2 – Structural & Circuit Enhancement 4
Iteration 3 – Final Augmentations & Uncertainty 5
Iteration 4 – Optimizations 6
Conclusion 6
Appendix 7
Photograph of Physical Robot (Final) 7
CAD Rendering (Final) 8
TinkerCAD Circuitry Diagram (Final) 8
Arduino Source Code (Final) 9
Structural Components (2D CAD Drawing) 10

Introduction

The objective of this project is to create a self-powered, autonomous robot capable of navigating through an obstacle course and launching a ping pong ball over a wall (*Fig 1*).

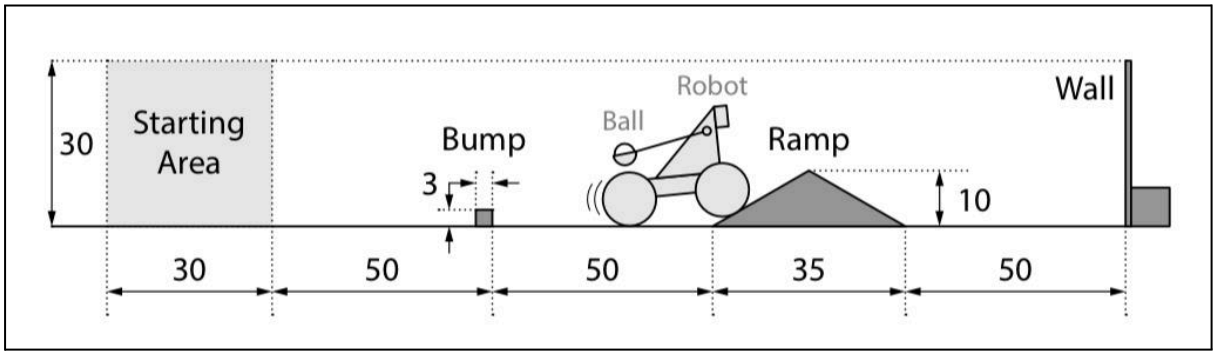
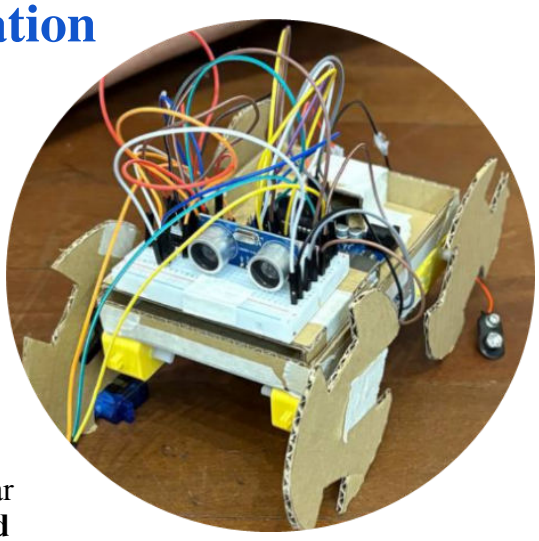


Fig 1: Overview of Obstacle Course (units in cm)

We were constrained in two ways: one in **time**, where we had a project deadline of about 10 weeks while juggling with other commitments, and the other in **materials**, where we had to build the robot using only the materials provided to us.

Given these constraints, our team adopted an **iterative approach**, beginning with a template model and progressively refining it through cycles of design, prototyping, and testing. This repetitive process of ideation and experimentation ultimately led to the successful creation of our robot. This report explains our methodology in approaching the problem statement, highlighting the challenges we encountered along the way, as well as our overall takeaways.

Iteration 1 – Template Model Evaluation

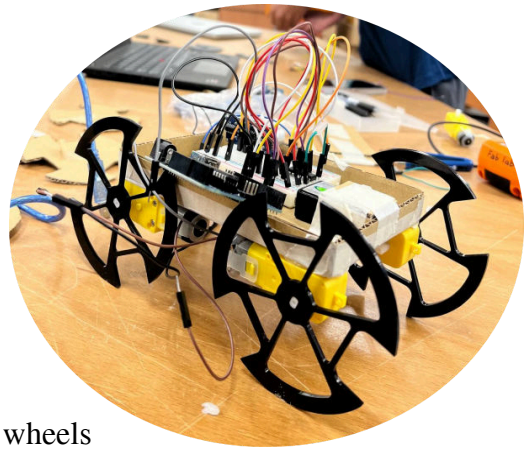


Starting with the provided template model, the robot performed reliably on flat surfaces, moving forward and stopping before the wall as expected. However, this configuration was insufficient for completing the full course. Notably, the robot consistently failed to cross both the bump and ramp. Its body also appeared unstable during motion, especially when moving off and stopping.

Upon analyzing and understanding the cause of these issues, we introduced several design augmentations. For one, we added an **additional pair of motors and wheels**, forming a standard car model to provide better stability and traction. Next, we **increased the diameter of the wheels** and clipped its circumference to create a **hook-like pattern**. This provided additional grip and allowed the wheels to "catch" onto the edges of the bump.

Testing the revised design on the actual course showed significant improvement. The robot successfully crossed the bump while maintaining a stable orientation and showed minimal angle deviation. These outcomes validated our analysis and confirmed that the modifications effectively addressed some initial shortcomings.

Iteration 2 – Structural & Circuit Enhancement



With a team of five, we began parallelizing the workload to work more efficiently. Two members focused on redesigning the robot's chassis to accommodate the servo motor and the 1.5V battery holders, while the remaining three concentrated on improving the circuit and Arduino code.

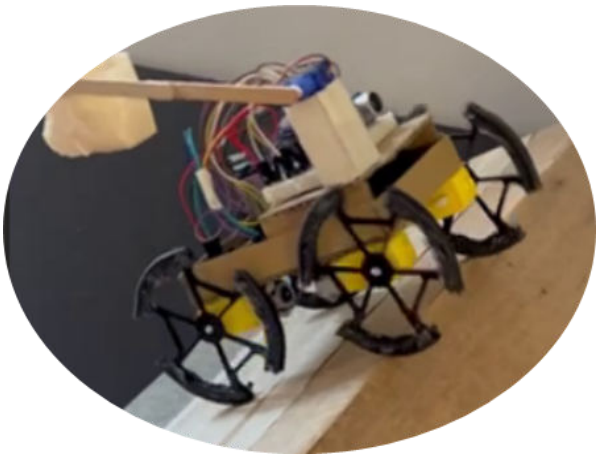
During this phase, we observed a critical mechanical issue: the cardboard wheels from the previous design began to **deform under the robot's weight**. This loss of structural integrity significantly reduced performance, as the wheels could no longer transmit force reliably. To address this, we decided to replace the wheels with a stronger material. That's when we had the idea to use laser-cutting to build **acrylic wheels**. While we could have simply cut the exact same pattern as the previous one, we foresaw a possible problem with its weight: Four completely solid acrylic pieces would substantially make our robot heavier, which might consequently cause it to be the heaviest among our lab section. Anticipating these effects, we refined the wheel design by **removing unnecessary material** while maintaining sufficient rigidity and stability.

In parallel, we made progress on improving the robot's circuitry and motion capabilities. The template model supported only forward motor rotation, which limited the robot's maneuverability. Through testing, we confirmed that **reversing the current direction** allowed the motors to spin backward. Applying concepts from Lecture L3C on motor driver switching, we incorporated the motor driver's H-bridge functionality to **control voltage polarity** using Arduino signal pins. This enabled reliable bidirectional control, allowing the robot to **reverse** after reaching the wall.

Upon testing these changes on both flat ground and the actual course, we observed notable improvements. The **acrylic wheels** provided significantly better load support, and the **refined hook-pattern edges** allowed the robot to clear the bump and partially ascend the ramp with greater consistency. The **updated circuitry** also performed as intended, with the robot stopping before the wall and successfully reversing afterward.

Despite these improvements, several issues remained unresolved. The robot was still unable to fully climb the ramp without external assistance, and the servo motor responsible for launching the ping-pong ball had not yet been mounted. These challenges were planned to be addressed in the next iteration of the design cycle.

Iteration 3 – Final Augmentations & Uncertainty



With only three weeks remaining before the graded run, we were on track toward completing the robot. To address the wheel slippage observed on the ramp, we implemented a simple but effective fix: wrapping the wheel circumference with the provided **anti-slip mat**. This substantially increased friction and allowed the wheels to maintain traction during the uphill climb.

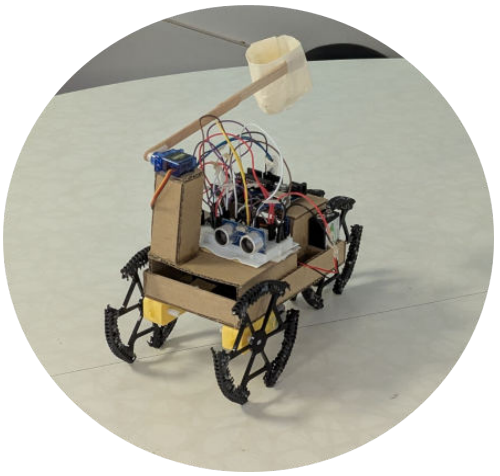
Long-term testing revealed another structural issue. The cardboard chassis, which had supported the robot for several weeks, began **sagging and weakening**. Although constructing a more durable base (similar to our redesign of the wheels) would have been ideal, we were limited by both material availability and time. We therefore rebuilt the chassis using **cardboard** again. Our reasoning was that the previous structure was able to last for the past six weeks, so it was sturdy enough to last through the graded run. Instead of pursuing an optimal long-term solution, we opted for the most feasible approach given our constraints.

This iteration also presented the most challenging part of the project – managing time pressure and addressing unforeseen abnormalities. We believed that our plan would allow us to smoothly address any problems of the robot, but failed to account for the uncertainty and unforeseen problems arising throughout the augmentations. After assembling the robot, we encountered **erratic behaviors** during testing. The robot moved back and forth repeatedly after reaching the wall, contradicting the logic of our code. Additionally, the servo motor **intermittently twitched** during initialization or after movement, causing the ping-pong ball to fall from its basket prematurely. Unlike previous issues, whose causes and fixes were straightforward, these symptoms were inconsistent and difficult to diagnose. We attempted to isolate the root cause using a voltmeter to measure potential differences across components, but the readings offered no conclusive explanation. The measurements suggested that the circuit should have been functioning correctly, which only deepened the uncertainty.

Through online research and guidance from lab teaching assistants, we narrowed the most probable cause to unstable current or voltage fluctuations within the circuit. However, the exact source of the instability remained unclear. With limited time left, we chose to adopt a trial-&-error approach, which includes **replacing several components** (breadboard, servo motor, and Arduino Uno), **rewiring the circuits**, and **refactoring parts of the code**. Although we were ultimately unable to pinpoint the precise cause, these corrective steps reduced the abnormal behavior to a manageable level, allowing the robot to operate as intended for the graded run.

This experience highlighted an important lesson: real-world hardware implementations can deviate significantly from theoretical reasoning and simulations (e.g. TinkerCAD). Uncertainty is inherent in physical systems, and future projects should incorporate both design strategies and schedule buffers to handle unexpected problems that cannot be analytically predicted.

Iteration 4 – Optimizations



With only a few days remaining before the graded run, this iteration focused on incremental optimizations to refine the existing design. The first improvement involved reducing unnecessary electrical resistance by relocating several intermediate wires on the breadboard, allowing **shorter connections** and slightly **increasing the available current** to the components. Additionally, **one of the rear motors was removed**, and the remaining motor was repositioned to reduce overall weight while still maintaining the stability afforded by a four-wheel configuration.

These adjustments enabled the robot to operate more consistently, offering improved efficiency and reduced mass without compromising structural stability or core functionality.

Conclusion

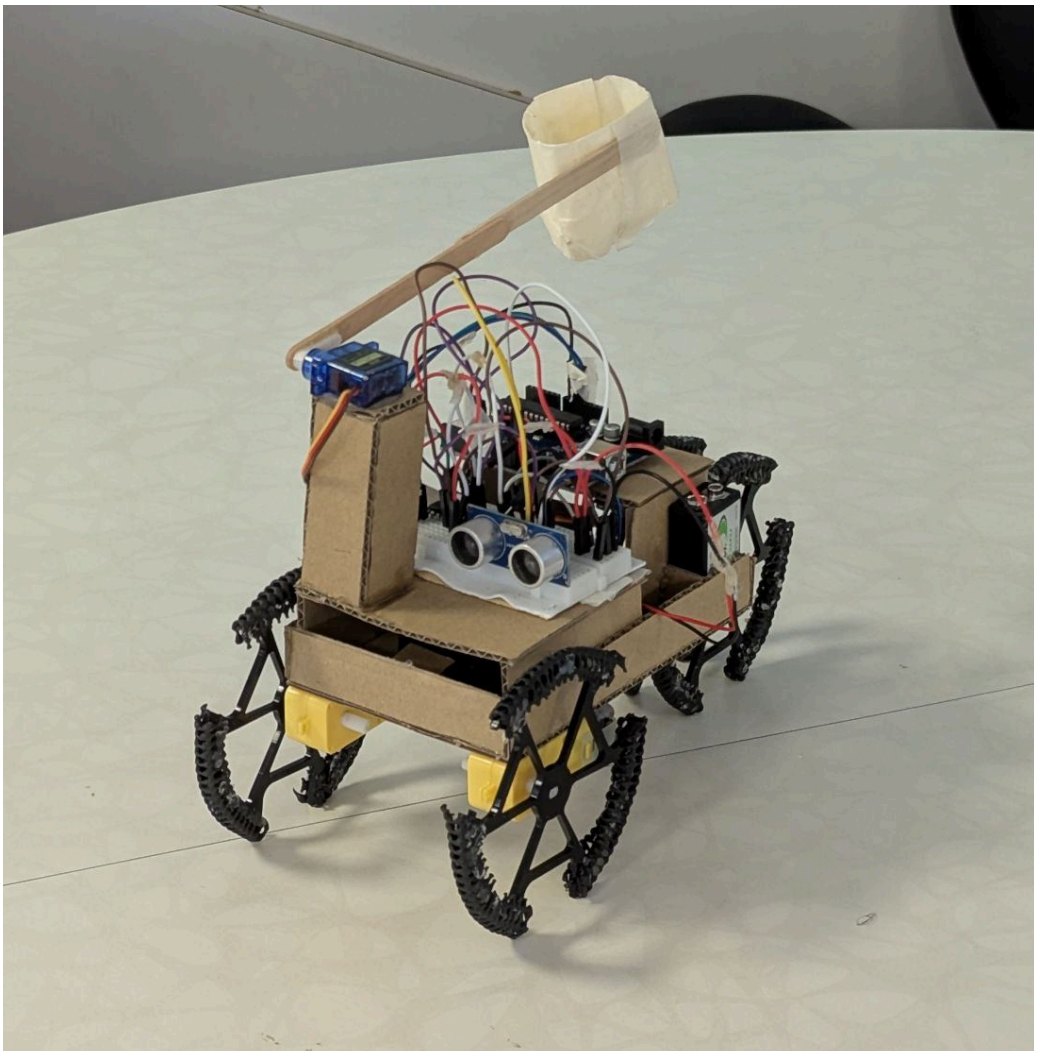
Through this project, we gained valuable insights into the strong interdependence between the mechanical, electrical, and programming components of a robotics system. Throughout the development process, we encountered multiple challenges, including unexpected component failures during test runs. Nevertheless, through persistence and systematic problem-solving, we were able to address these issues and refine our design. Our efforts paid off when our robot was able to clear the obstacle and end with the second lightest design within the first attempt.

The project also strengthened our teamwork and collaboration skills. Regular brainstorming sessions enabled us to leverage each member's strengths while supporting one another in areas requiring improvement. With team members specializing in mechanical, electrical, and computer engineering, we were able to distribute tasks efficiently and focus on our respective domains of expertise.

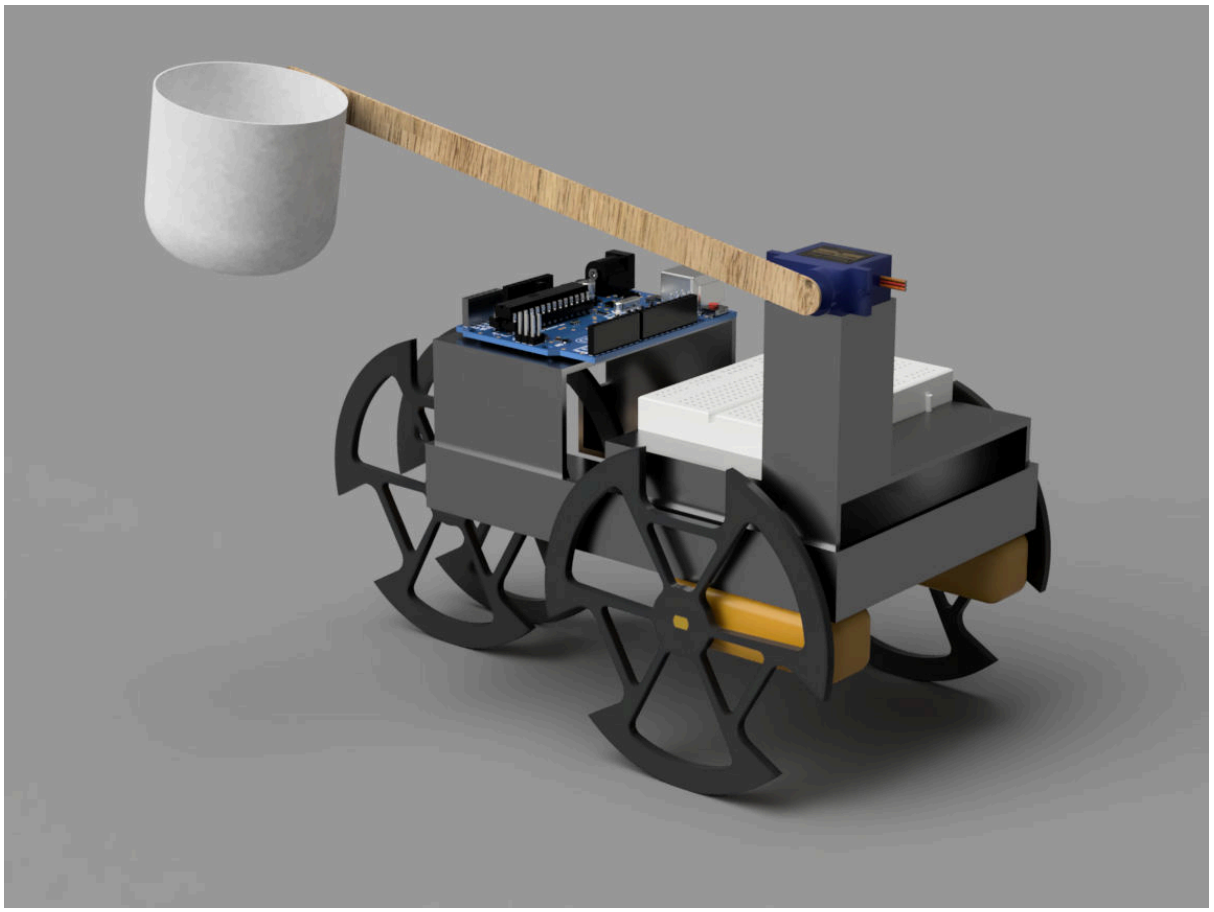
Overall, this project provided us with valuable engineering and project-management experience, reinforcing skills that will be essential in our future endeavours as undergraduate engineers.

Appendix

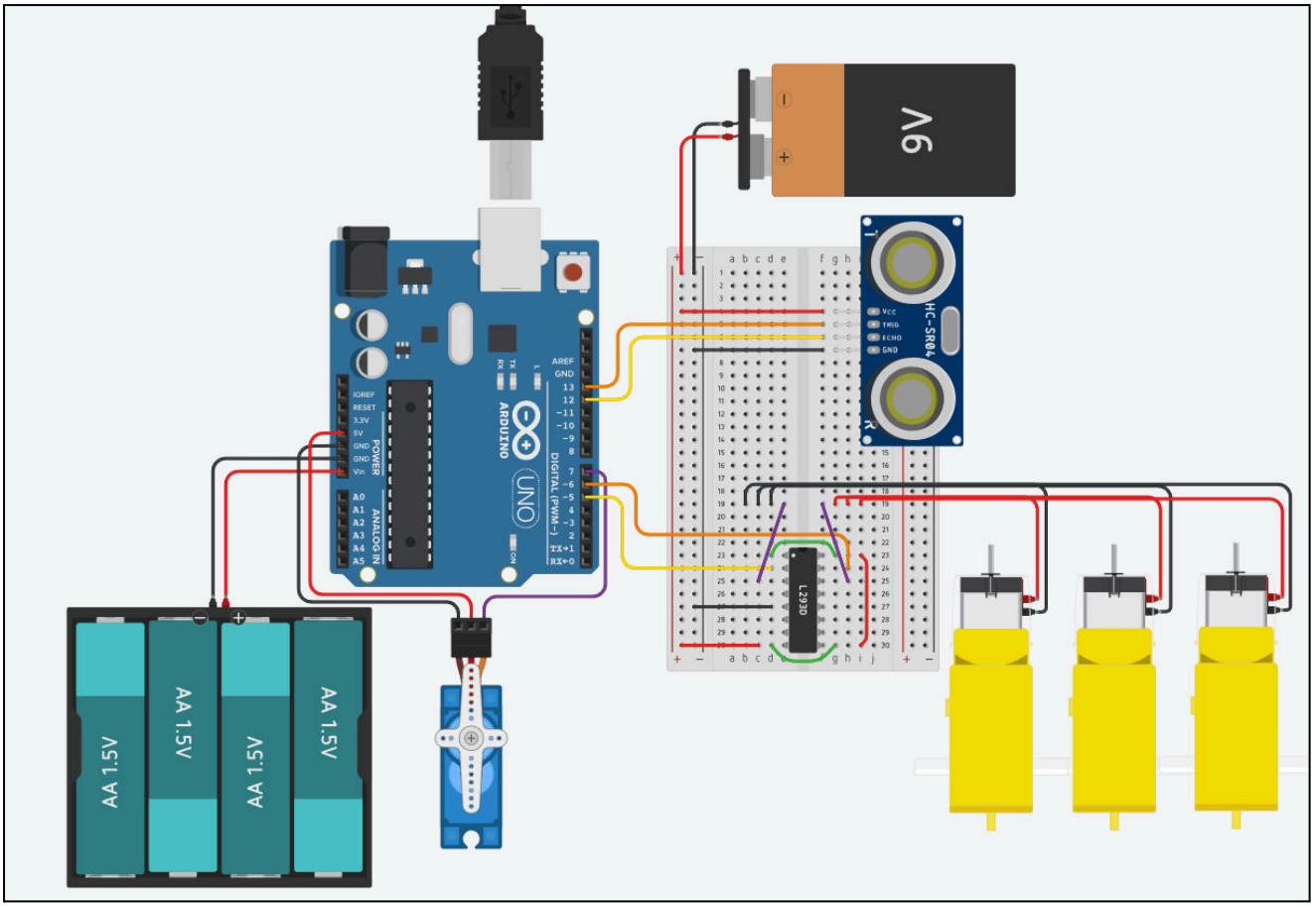
Photograph of Physical Robot (Final)



CAD Rendering (Final)



TinkerCAD Circuitry Diagram (Final)



Arduino Source Code (Final)

```
#include <Servo.h>
Servo servo;

int MOTOR_PIN1 = 5; // connect to +ve motor
int MOTOR_PIN2 = 6; // connect to -ve motor
int SERVO_PIN = 7;
int ECHO_PIN = 12;
int TRIG_PIN = 13;

float SPEED_OF_SOUND = 0.0345;
float STOP_DISTANCE = 15.0;

bool isReturning;

void setup() {
  delay(2000);
  pinMode(MOTOR_PIN1, OUTPUT);
  pinMode(MOTOR_PIN2, OUTPUT);
  pinMode(TRIG_PIN,   OUTPUT);
  pinMode(ECHO_PIN,   INPUT);

  digitalWrite(TRIG_PIN, LOW);

  servo.attach(SERVO_PIN, 500, 2400);
  delay(500);
  servo.write(150);
  delay(3000);

  isReturning = false;
}

void loop() {
  if (isReturning) {
    digitalWrite(MOTOR_PIN1, LOW);
    digitalWrite(MOTOR_PIN2, HIGH);
    delay(10); // 10 milliseconds
    return;
  }

  digitalWrite(MOTOR_PIN1, HIGH);
  digitalWrite(MOTOR_PIN2, LOW);

  // send sensor pulse
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(12);
  digitalWrite(TRIG_PIN, LOW);

  // check pulse response
  int microsecs = pulseIn(ECHO_PIN, HIGH, 8000); // 8 milliseconds
  float distance_from_wall = microsecs * SPEED_OF_SOUND / 2;

  // far from wall
  if (distance_from_wall < 1 || distance_from_wall >= STOP_DISTANCE) {
    delay(20);
    return;
  }
```

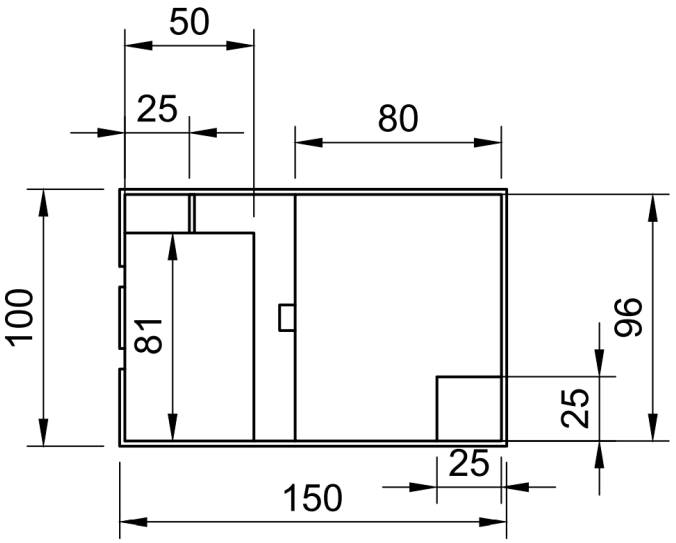
```
  // close to wall
  else {
    digitalWrite(MOTOR_PIN1, LOW);
    digitalWrite(MOTOR_PIN2, LOW);

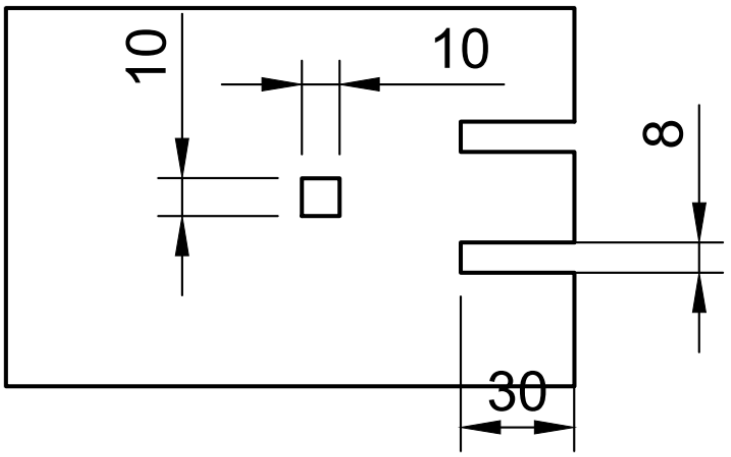
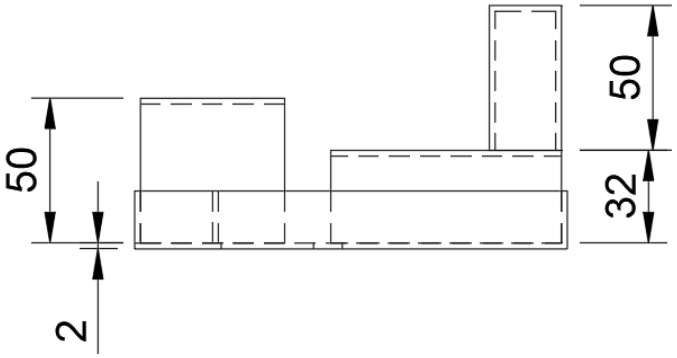
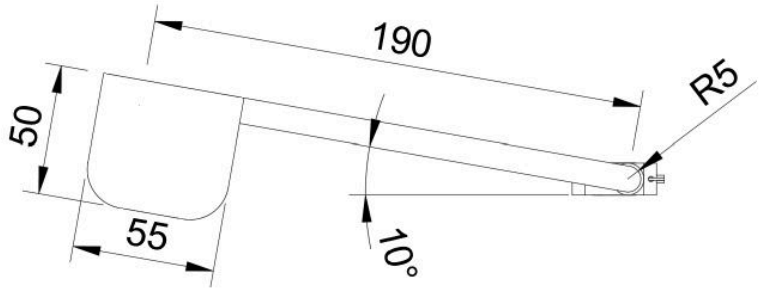
    delay(1000);
    catapult();

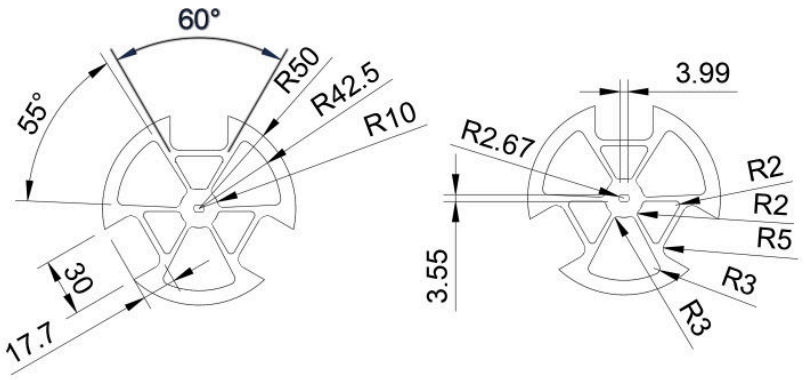
    isReturning = true;
    delay(500);
  }
}

void catapult() {
  servo.write(70);
  delay(1500);
  servo.write(150);
  delay(1200);
}
```

Structural Components (2D CAD Drawing)

Components	Dimensions
Chassis **Material:** Cardboard(2mm thick)	**Top View:** 

	Bottom View:  **Side View:** 
Catapult **Material:** Tape & popsicle sticks	**Side View:** 

Wheels **Material:** Acrylic & Anti-slip Rubber mat	

--- END OF REPORT ---