Dayananda Sagar University B.Tech VI Semester
PROJECT – I
March 2024

Department of Electronics and Communication Engineering

Course Title: Embedded Systems Design

Project Report
on

“E-Smart Dustbin”

Submitted as a Semester End Assessment component

As part of 6th Sem/ 3rd year program of
BACHELOR OF TECHNOLOGY

Group members:-
MANOJ TH: ENG21EC0058
SUHAS D NAIK: ENG21EC0120
MALLIKARJUN TD: ENG21EC0055
LINGARAJ: ENG21EC0052

Contents:

Introduction:

The Smart Dustbin project introduces an innovative solution to the challenge of waste management, focusing specifically on the segregation of dry and wet waste. By harnessing advanced technology and automation, this project aims to streamline waste disposal processes and promote sustainability in urban environments.

This report provides a detailed overview of the design, development, and deployment of the Smart Dustbin system. Through a rigorous design cycle, we have created a hardware-based solution that incorporates sensors, actuators, and software algorithms to achieve precise waste sorting capabilities. Our system not only facilitates efficient segregation of dry and wet waste but also offers real-time monitoring and notifications.

Throughout this report, we will explore the various components of the Smart Dustbin system, including its design rationale, cost considerations, hardware configuration, software architecture, and practical outcomes. Additionally, we will discuss the broader implications of this technology for waste management practices, emphasizing its potential to contribute to a more sustainable future.

The Design Cycle:

1. **Planning and Requirements Analysis:**
   - Identified the needs and expectations for the Smart Dustbin, focusing on functionalities like waste segregation and real-time monitoring.

2. **Conceptual Design:**
   - Developed an initial concept outlining the system's architecture and components, including sensors and microcontroller platforms.

3. **Detailed Design:**
   - Specified hardware components and designed sensor interfaces, as well as mapped out the software structure.

4. **Hardware Implementation:**
   - Procured and assembled the necessary hardware components, such as sensors and actuators, according to design specifications.

5. **Software Development:**
   - Created firmware to control the Smart Dustbin's operation, including interfacing with sensors and controlling output devices.

6. **Integration and Testing:**
   - Conducted testing to ensure all system components functioned together smoothly, verifying sensor readings and actuator responses.

7. **Refinement and Optimization:**
   - Iteratively improved the system's robustness, efficiency, and user experience through adjustments to sensor calibration and software algorithms.

8. **Validation and Deployment:**
   - Confirmed the system's performance against requirements through validation testing before deploying it for real-world use in waste management practices.

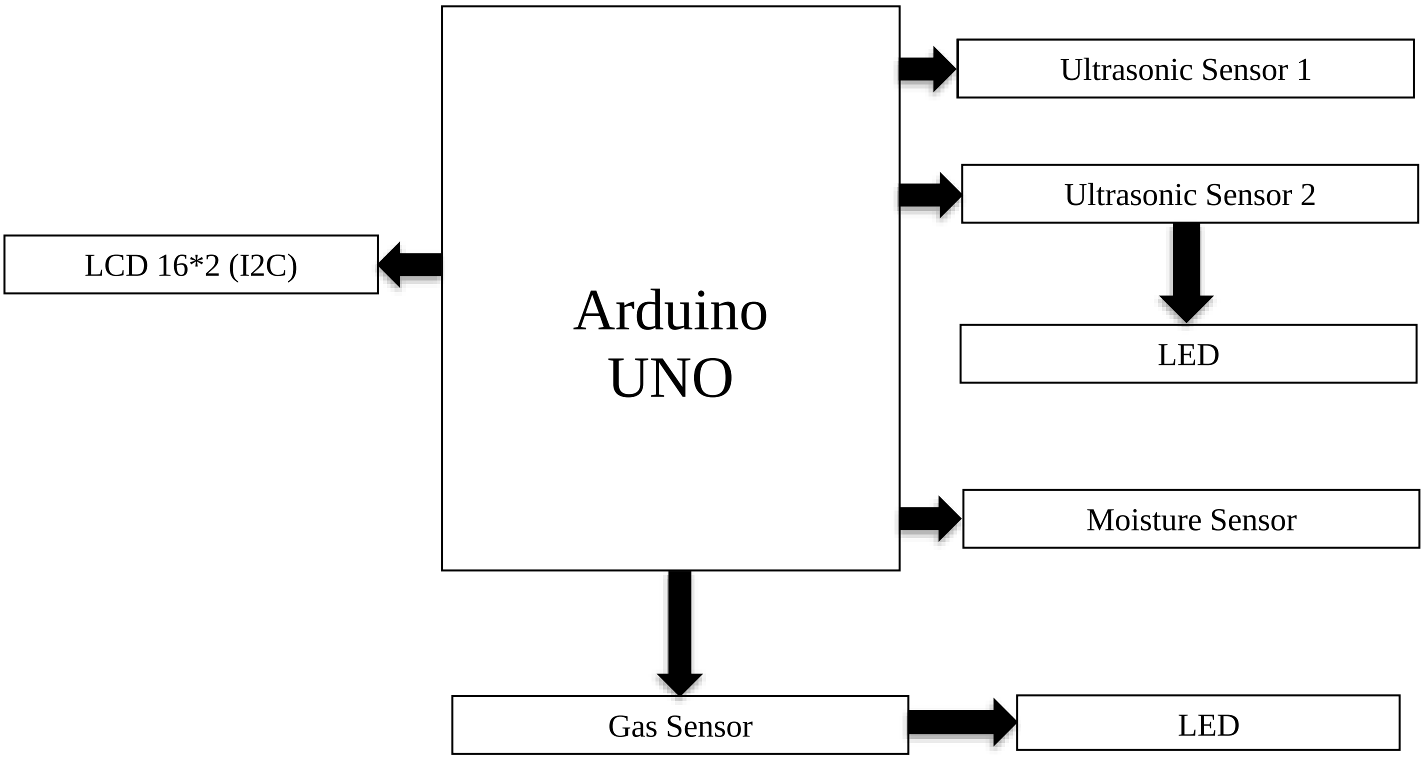
The components and cost calculation of our system:

1. **Arduino Uno**: Rs 250
    - The Arduino Uno is a popular microcontroller board based on the ATmega328P chip. It provides a simple and accessible platform for building various electronics projects, including the Smart Dustbin system.

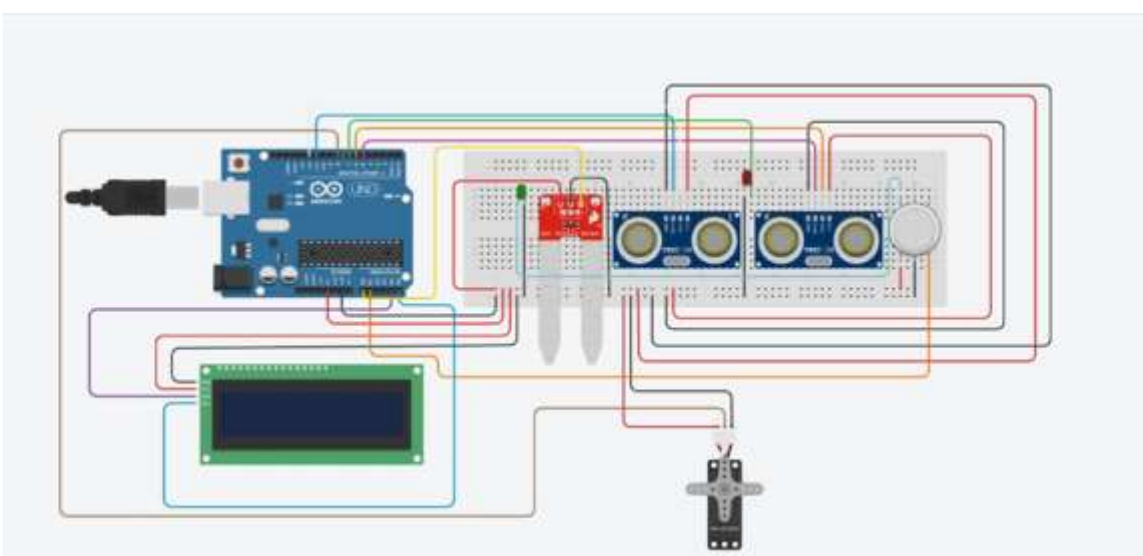
2. **Breadboard**: Rs 60
    - A breadboard is a reusable solderless prototyping board used to build and test electronic circuits. It allows for easy connection and experimentation with electronic components.
3. **LCD with I2C**: Rs 120
    - The LCD (Liquid Crystal Display) with I2C module is a 16x2 character display that can be easily interfaced with Arduino using the I2C communication protocol. It provides a convenient way to display information in the Smart Dustbin system.
4. **Ultrasonic Sensors (2x)**: Rs 90
    - Ultrasonic sensors are used for non-contact distance measurement by emitting ultrasonic waves and measuring the time taken for the waves to bounce back. In the Smart Dustbin system, these sensors are used to detect the presence and distance of objects, enabling waste level monitoring.
5. **Moisture Sensor**: Rs 45
    - A moisture sensor is used to detect the moisture level in the soil. In the Smart Dustbin system, it helps determine whether the waste is dry or wet, facilitating the segregation process.
6. **Gas Sensor**: Rs 100
    - The gas sensor detects the presence of gases in the environment. In the Smart Dustbin system, it can be used for detecting harmful gases emitted from waste, providing additional environmental monitoring capabilities.
7. **Servo Motor**: Rs 60
    - A servo motor is a rotary actuator that allows for precise control of angular position. In the Smart Dustbin system, it is used to open and close the waste compartment lids for segregation purposes.
8. **Jumper Wires and Connecting Cables**: Rs 30
    - Jumper wires and connecting cables are used to establish connections between various components in the Smart Dustbin system, facilitating the wiring and assembly process.
9. **Battery of 5V**: Rs 50 (Estimated)
    - A 5V battery is used to power the Smart Dustbin system, providing portability and flexibility in deployment.
10. **LEDs (2x)**: Rs 10
    - Light-emitting diodes (LEDs) are used as indicators in the Smart Dustbin system, providing visual feedback on system status or alerts.

Total Cost: Rs 815

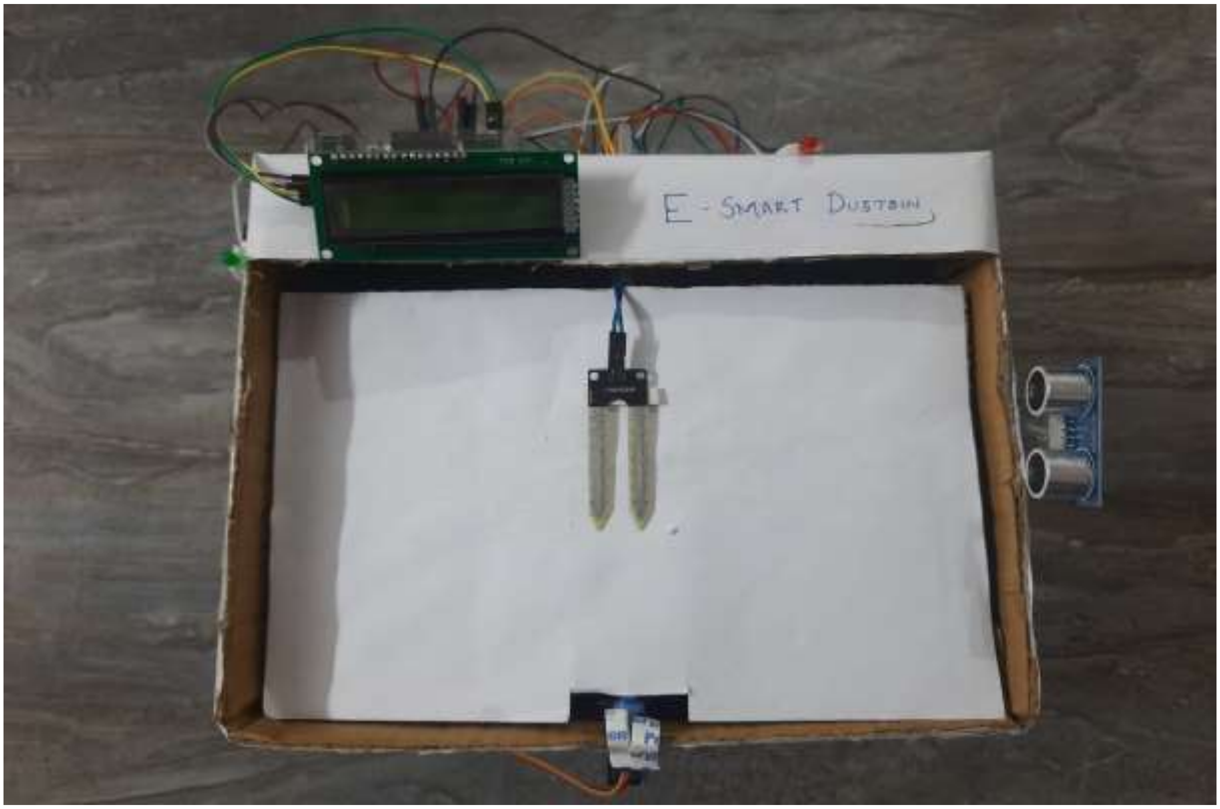
Block diagram

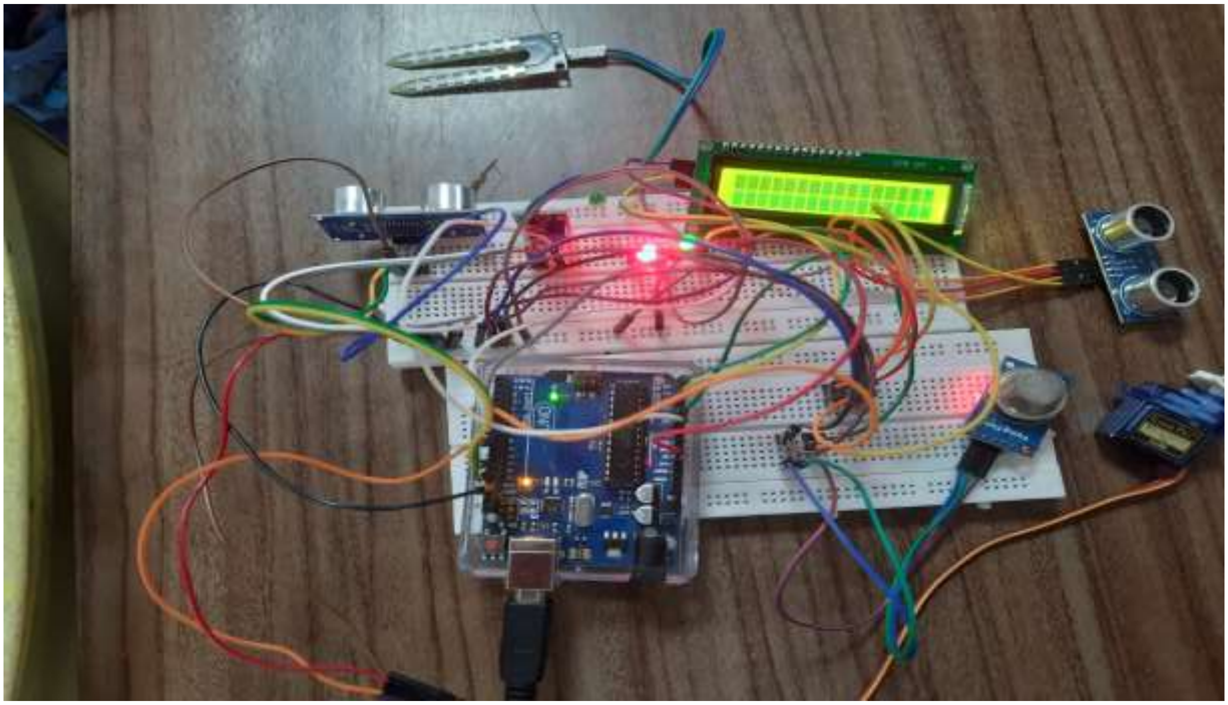


Circuit Diagram



Picture of the Entire Assemble of Hardware:





The software and the code

Here's some information about the software code used for the Smart Dustbin system :

1. **Arduino IDE:**
    - Open-source software for programming Arduino microcontroller boards.

2. **Code Overview:**
    - Written in Arduino language (based on C/C++), controlling hardware components and processing sensor data.

3. **Libraries Used:**
    - Wire.h: Supports I2C communication.
    - LiquidCrystal_I2C.h: Extends LiquidCrystal library for I2C-enabled LCD displays.
    - Servo.h: Controls servo motors for lid movement.

4. **Setup Function:**
    - Initializes system components, sets up serial communication, and configures sensors/actuators.

5. **Loop Function:**
    - Continuously reads sensor data, controls servo motor for lid movement, monitors dustbin fullness, and updates LCD display.

6. **Sensor Data Processing:**
    - Reads ultrasonic sensors for waste distance and moisture sensor for waste moisture level.

7. **Actuator Control:**
    - Controls servo motor for lid movement based on waste type and triggers buzzer for full dustbin.

8. **Serial Communication:**
    - Used for debugging, printing sensor readings and system messages to Serial Monitor.

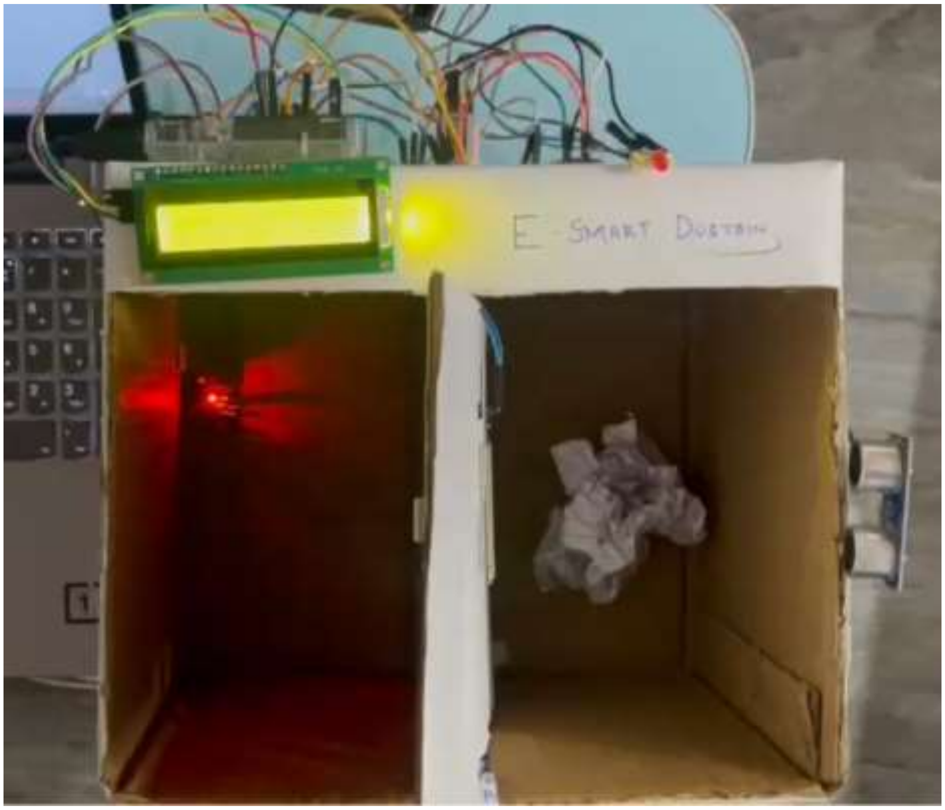
Code:

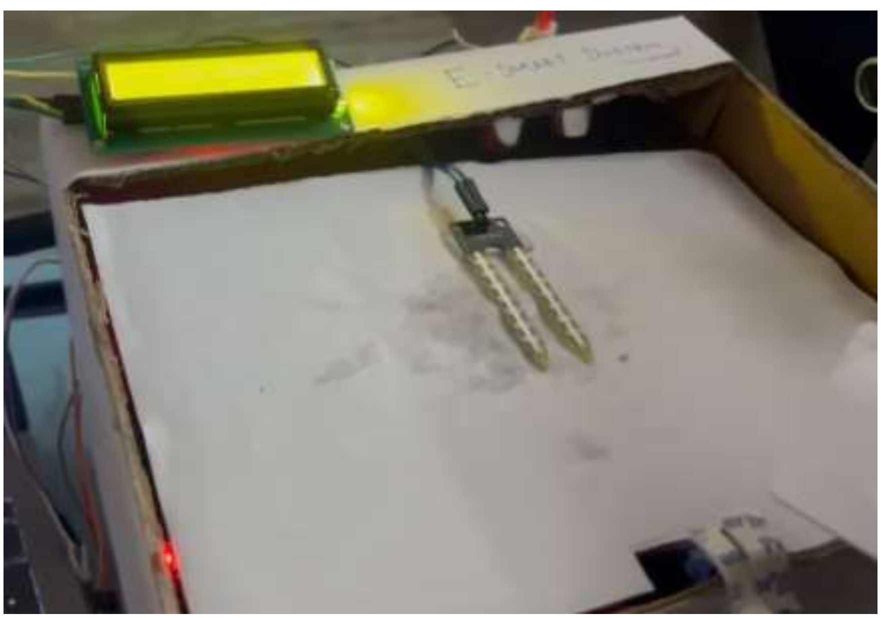
```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Servo.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);
Servo servoMotor;

const int trigPin = 12;
const int echoPin = 11;
const int trigPin2 = 5;
const int echoPin2 = 6;
long duration;
int distance = 0;
int soil = 0;
int fsoil = 0;
int potPin = A0;

const int gasPin = A1; // Analog pin for gas sensor
const int buzzerPin = 7; // Digital pin for buzzer
int gasThreshold = 200; // Adjust this threshold based on sensor
calibration
int distanceThreshold = 20; // Adjust this threshold for dustbin fullness

long ultrasonicRead(int trigPin, int echoPin) {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  return pulseIn(echoPin, HIGH) / 58.2;
}

void setup() {
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Initializing...");
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(trigPin2, OUTPUT);
```

```
  pinMode(echoPin2, INPUT);
  servoMotor.attach(8);
  pinMode(buzzerPin, OUTPUT);
  delay(2000); // Delay to stabilize LCD
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Smart Dustbin");
  lcd.setCursor(0, 1);
  lcd.print("Initializing...");
  delay(2000); // Delay for initialization
}

void loop() {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Dry Wet Waste");
  lcd.setCursor(0, 1);
  lcd.print("Segregator");

  // Ultrasonic Sensor 1 and Moisture Sensor
  distance = ultrasonicRead(trigPin, echoPin);
  Serial.println(distance);

  if (distance < 15 && distance > 1) {
    soil = readMoisture(potPin);
    Serial.println(soil);
    if (soil > 500) {
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("Garbage Detected!");
      lcd.setCursor(6, 1);
      lcd.print("DRY");
      servoMotor.write(180);
      delay(1000);
    }
    else {
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("Garbage Detected!");
      lcd.setCursor(6, 1);
      lcd.print("WET");
      servoMotor.write(0);
      delay(1000);
    }
    servoMotor.write(96);
```

```
  }

  // Ultrasonic Sensor 2 for Dustbin Fullness
  distance = getUltrasonicDistance();
  Serial.println(distance);

  if (distance < distanceThreshold) {
    digitalWrite(buzzerPin, HIGH); // Turn on the buzzer
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Dustbin is Full");
  } else {
    digitalWrite(buzzerPin, LOW); // Turn off the buzzer
  }

  delay(1000);
}

int getUltrasonicDistance() {
  digitalWrite(trigPin2, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin2, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin2, LOW);
  long duration = pulseIn(echoPin2, HIGH);
  int distance = duration * 0.034 / 2;
  return distance;
}

int readMoisture(int pin) {
  int moistureValue = analogRead(pin);
  return moistureValue;
}
void rotateServo(int angle) {
  servoMotor.write(angle);
  delay(1000); // Wait for servo to reach position
  servoMotor.write(90); // Return servo to original position
  delay(1000); // Wait for servo to return to original position
}
```

Picture of results.





If an OS has to be used a. Name an OS that can be used b. list the jobs that the OS will do.

a. Ferrets (Real-Time Operating System) is a suitable choice for the Smart Dustbin project due to its lightweight nature and real-time capabilities.

b. Ferrets will handle task scheduling, resource management, and interrupt handling to ensure timely execution of critical tasks such as sensor data processing and actuator control. It will manage communication between system components and external devices, implement error handling mechanisms for fault tolerance, and optimize power consumption for energy efficiency. Additionally, FreeRTOS will provide real-time performance, security features, and system monitoring capabilities for robust and reliable operation of the Smart Dustbin system

It’s Application:

1. **Smart Homes:**
    - Implementing the Smart Dustbin project in households to automate waste segregation and optimize waste management practices.
2. **Educational Institutions:**
    - Deploying the Smart Dustbin system in schools and colleges to educate students about waste segregation and environmental conservation.
3. **Commercial Buildings:**
    - Installing Smart Dustbins in office complexes and shopping malls to improve waste management efficiency and promote sustainable practices.
4. **Public Parks:**
    - Integrating Smart Dustbins in public parks and recreational areas to maintain cleanliness and enhance environmental monitoring efforts.
5. **Smart Cities:**
    - Incorporating the Smart Dustbin project into smart city initiatives to optimize waste collection routes, reduce operational costs, and promote urban sustainability.