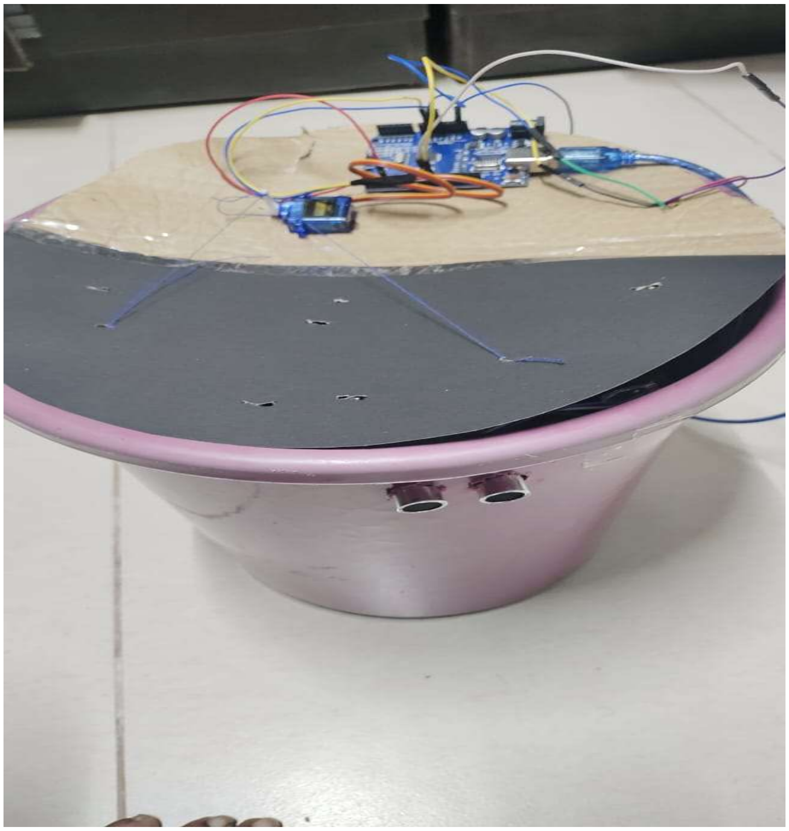
Project Title: Automated Dustbin System



Problem Statement:

Develop a simple Arduino-Uno based Automatic Dust Bin system which can open the lid when it senses movement nearby.

A smart dustbin offers several benefits and use cases, particularly in promoting hygiene, efficiency, and sustainability.

SCOPE OF SOLUTION:

Hygiene:

- **Touchless Operation**: Reduces the need for physical contact, minimizing the spread of germs and bacteria.
- **Automatic Lid Closure**: Helps contain odors and prevents pests from accessing the trash

Convenience:

- **Automatic Opening**: Makes it easier to dispose of waste, especially when your hands are full.
- **Fullness Detection**: Some smart bins can alert you when they are full and need to be emptied.

Efficiency:

- **Optimized Waste Management**: Sensors can help in tracking waste levels, allowing for more efficient collection and disposal.
- **Time-Saving**: Reduces the need for frequent manual checks of waste levels.

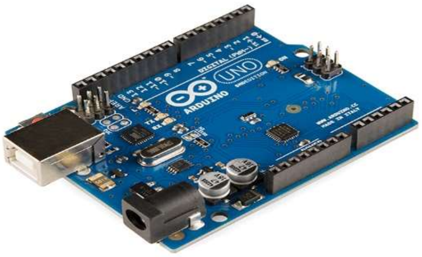
Environmental Impact:

- **Recycling Assistance**: Smart bins can be designed to help sort recyclable materials from non-recyclable ones.
- **Data Collection**: Provides data on waste generation patterns, helping in the development of better waste management strategies.

REQUIRED COMPONENT:

Hardware:

1. Arduino Uno



2.Ultrasonic Sensor

3.Servo Motor

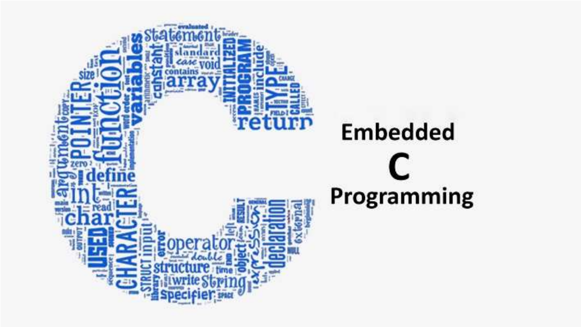
Software:

1.Arduino IDE 2.3.2

2.Tinkercad

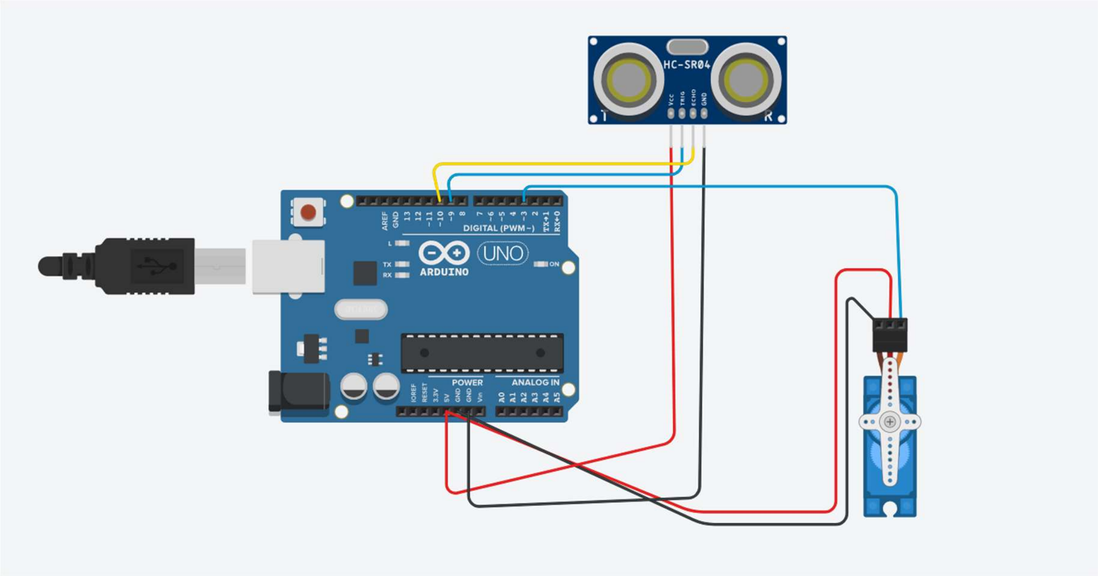
3.EasyEDA

4.Embedded C:



SIMULATED CIRCUIT:

Tinkercad:



Circuit Procedure:

- Connect the VCC and GND of the ultrasonic sensor to 5V and GND on the Arduino.
- Connect the Trig and Echo pins of the ultrasonic sensor to digital pins on the Arduino (e.g., 9 and 10).
- Connect the signal pin of the servo motor to a PWM-capable pin on the Arduino (e.g., pin 3).
- Connect the power and ground of the servo motor to 5V and GND on the Arduino.

Working:

1. **Arduino Board**
   - Acts as the brain of the system.
   - Processes inputs from the ultrasonic sensor and controls the servo motor based on these inputs.
2. **Ultrasonic Sensor (e.g., HC-SR04)**
   - **Purpose**: Measures the distance to an object using sound waves.
   - **Working Principle**:
     - The sensor emits an ultrasonic sound wave from the `Trig` (trigger) pin.
     - When the sound wave hits an object, it reflects back to the sensor and is received by the `Echo` pin.
     - The time taken for the sound wave to return is measured and used to calculate the distance to the object.
   - **Pin Connections**:
     - `Trig` Pin: Connects to a digital pin on the Arduino and sends a pulse to start the measurement.

        - `Echo` Pin: Connects to another digital pin on the Arduino and receives the reflected pulse.
    - **Distance Calculation**: The distance is calculated using the formula: Distance=(Duration2)×0.034 cm\text{Distance} = \left(\frac{\text{Duration}}{2}\right) \times 0.034 \text{ cm}Distance=(2Duration)×0.034 cm where 0.034 cm/µs is the speed of sound in air divided by two to account for the round-trip of the sound wave.
3. **Servo Motor**
    - **Purpose**: Mechanically opens and closes the lid of the dustbin.
    - **Working Principle**:
        - The servo motor is controlled by sending a PWM (Pulse Width Modulation) signal from the Arduino.
        - The position of the servo motor is determined by the width of the PWM pulse.
        - In this project, a 0-degree pulse closes the lid, while a 90-degree pulse opens it.
    - **Pin Connections**:
        - Signal Pin: Connects to a PWM-capable pin on the Arduino to receive control signals.
        - Power and Ground: Connect to the Arduino's 5V and GND pins to power the motor.
4. **Power Supply**
    - Provides electrical energy to the Arduino and the connected components.
    - Typically sourced from a USB connection or an external power adapter.

Component Interaction

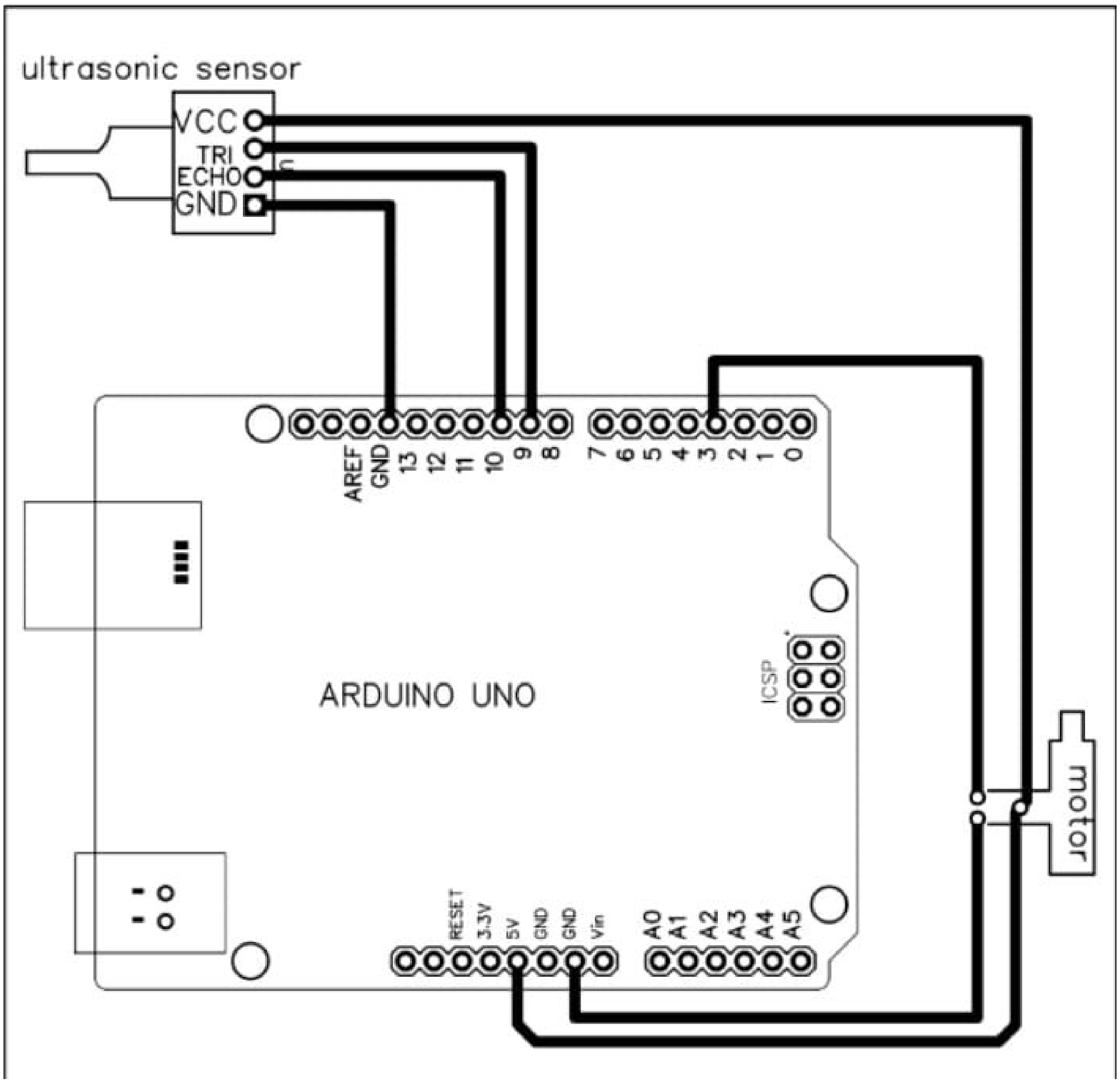
1. **Ultrasonic Sensor**:
    - The Arduino sends a 10-microsecond pulse to the `Trig` pin.
    - The sensor emits an ultrasonic wave.
    - If the wave hits an object, it bounces back to the `Echo` pin.
    - The Arduino measures the time it takes for the echo to return and calculates the distance.
2. **Servo Motor**:
    - The Arduino calculates the distance and compares it with the threshold.
    - If the distance is less than or equal to 30 cm, the Arduino sends a signal to rotate the servo motor to 90 degrees, opening the lid.
    - After a delay (5 seconds in your code), the Arduino sends another signal to rotate the servo back to 0 degrees, closing the lid.

Overall Workflow

1. **Detection**: The ultrasonic sensor continuously monitors the distance to nearby objects.
2. **Processing**: When an object is detected within 30 cm, the Arduino processes this information.
3. **Actuation**: The Arduino sends a signal to the servo motor to open the lid.
4. **Delay and Reset**: After a delay, the Arduino resets the servo to close the lid, and the process repeats.

This combination of components allows the smart dustbin to operate automatically and hygienically, opening only when an object (like a hand or waste) is detected nearby.

GERBER FILE:



We are not using any PCB for this Project. But the gerber file can be used to understand the connections better

CODE FOR SOLUTION:

```
#include <Servo.h>  //Servo library required for this

const int trigPin = 9;  //Set Pin 9 as Trig output
const int echoPin = 10; // Set Pin 10 as echo input Pin
const int servoPin = 3; //Set Pin 3 Servo Motor out Pin
const int distanceThreshold = 10; // defines constant integer with value 30 cm

Servo myServo; //creates Servo class named ‘myservo’

void setup() {
  pinMode(trigPin, OUTPUT);  // sets ‘trigPin’ as output
  pinMode(echoPin, INPUT);  //sets ‘echopin’ as input
  myServo.attach(servoPin); //attaches servo motor to ‘servopin’
  myServo.write(0); // lid closed
  Serial.begin(9600); //set baud rate at 9600 bits per second
}

void loop() {        //initiate loop
  long duration, distance;    //declares two variables
  digitalWrite(trigPin, LOW);  //sends low signal to trigPin
  delayMicroseconds(2);  //wait for 2 microseconds
  digitalWrite(trigPin, HIGH);  //sends high signal to trigPin
  delayMicroseconds(10);  //Wait 10 microseconds
  digitalWrite(trigPin, LOW);  // sends low signal stop the pulse


  duration = pulseIn(echoPin, HIGH);  //Measures time required for echo
                                        signal to return
 distance = (duration / 2) / 29.1;   //since sounds travel back and forth we divide
                                    by 2 and then by 29.1 which is the number
                                    of microseconds required for sound to travel
                                     one centimeter
  if (distance <= distanceThreshold) {  //checks if distance is less than equal to
                                         threshold distance
myServo.write(90);       // lid open
    delay(5000); // keep the lid open for 5 seconds
    myServo.write(0); // lid closed
  }


  delay(200); //delay 200 milliseconds
}
```