



AUTOMATED DUSTBIN SYSTEM

**PRESENTED BY:
GROUP 11
NEHA KURIAN
SWEDHA S
PRASHOB S NAIR
SOORAJ SAJEEV
SHAHANA SHERIN**

Problem statement

Traditional dustbins require physical contact to open the lid, which can be unhygienic and inconvenient, especially in environments like public spaces, homes, hospitals, and offices. Manual interaction increases the risk of germ transfer and reduces overall user experience. Furthermore, many existing solutions lack affordability and simplicity, making them inaccessible for widespread adoption.

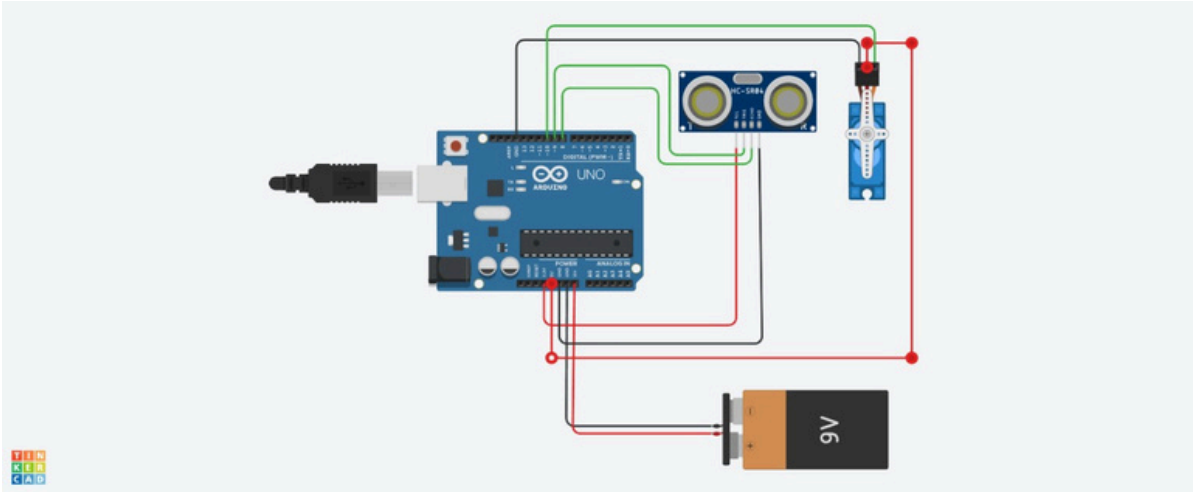
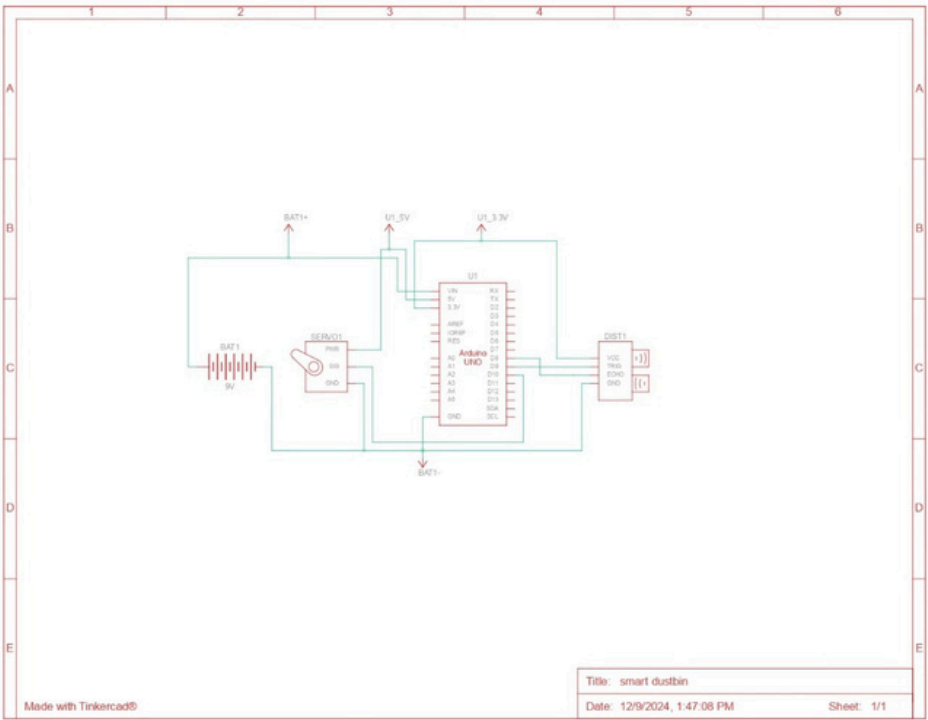
Scope of solution

The proposed solution is an Arduino-based automated dustbin that uses a proximity sensor to detect the movement of nearby objects and automatically opens its lid using a servo motor. This system aims to improve hygiene, reduce physical contact, and enhance convenience.

Required components

- Arduino IDE 2.3.3
- Software: Fritzing
- Hardware: ultrasonic sensor, servo motor

Simulated circuit



Video of the demo

[https://drive.google.com/drive/folders/
1Qod1W_0KNx30ku-
ZQRZ14glx0nukikHB?usp=sharing](https://drive.google.com/drive/folders/1Qod1W_0KNx30ku-ZQRZ14glx0nukikHB?usp=sharing)

**The demo video of automated dustbin is uploaded in the
above drive folder**

Gerber file

https://drive.google.com/drive/folders/1M4o9JsJoEwtJ8Yv_OO1PKrkuz20bMObG?usp=sharing

The gerber files are uploaded in the above drive folder

Code for the solution

```
#include <Servo.h>
Servo servoMain; // Define our Servo

// Ultrasonic sensor pins
const int trigpin = 9;
const int echopin = 8;

// Variables for distance calculation
float duration;
int distance;

void setup() {
  servoMain.attach(10); // Attach the servo to pin 10
  pinMode(trigpin, OUTPUT);
  pinMode(echopin, INPUT);
}

void loop() {
  // Trigger the ultrasonic sensor
  digitalWrite(trigpin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigpin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin, LOW);

  // Measure the duration of the echo pulse
  duration = pulseIn(echopin, HIGH);

  // Convert the duration to distance in cm
  distance = duration / 58.82;

  // Check if the distance is valid and below the threshold
  if (distance > 0 && distance < 30) {
    servoMain.write(180); // Rotate servo to 180 degrees (fully open)
    delay(3000); // Keep the servo in position for 3 seconds
  } else {
    servoMain.write(0); // Rotate servo back to 0 degrees (fully closed)
    delay(50); // Short delay to prevent rapid triggering
  }
}
```