LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title
CS5068NI– Cloud Computing & IoT

Smart Dustbin

Assessment Type
50% Group Report
Semester
2025 Autumn

Group Members

London Met ID	Student Name
23047310	Rahul Chaudhary
23047306	Aman Adhikari
23047305	Nisha Roy
23047367	Rahul Thapa

Assignment Due Date: March 26, 2025, at 12:16 PM
Assignment Submission Date: May 15, 2025, at 01:00 PM
Submitted to: Mr. Sugat Man Shakya
Word Count: 2977

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

5% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

10 Not Cited or Quoted 3%
Matches with neither in-text citation nor quotation marks

6 Missing Quotations 2%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

1% Internet sources
0% Publications
4% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Acknowledgement

We thank **Mr. Sugat Man Shakya**, our lecturer and **Mr. Ayush Bhakta Pradhanang** our tutor, for their guidance, support, and valuable insights throughout this coursework. Their skill and dedication have played an essential part in developing our understanding of Cloud Computing and IOT and guiding us through the investigation and content of this study. We also want to thank from the bottom of our hearts for all the help and guidance they gave us. Their advice and support have been very helpful in guiding us through this project's challenges and ensuring it is finished successfully.

Abstract

This is a design and development project on an IoT technology-based Smart Dustbin to improve waste management solutions. The Smart Dustbin incorporates sensors such as ultrasonic to sense the waste level, another ultrasonic to sense motion, and DHT22 to sense environmental parameters such as temperature and humidity. It is made feasible due to the presence of an ESP32 microcontroller within the system for automation, as well as real-time data transfer, to facilitate better touchless operation and effective collection of trash. Through offering a smart and intelligent solution, this project will end issues like overflowing bins, poor collection efficiency, and cleanliness. This intelligent solution renders the environment cleaner and more environmentally friendly through the prevention of human verification and remote monitoring capabilities.

Table of Contents

Table of figures

Tables of table

1. Introduction

Internet of Things (IoT) is a term that means ordinary things are internet-enabled so that they are able to send, receive, and collect information. These "things" may comprise anything from home appliances and wearable devices to cars and industrial machines.

In daily life, IoT improves effectiveness and ease. For example, your mobile device enables you to modify heating and cooling controls together with security camera functions and lighting options through smart home platforms. The mobile tech tracks your heart rate together with your sleep patterns. Cars make use of IoT technology to execute safety alerts and vehicle inspection checks as well as GPS navigation capabilities. The implementation of the Internet of Things by cities includes monitoring air quality and implementing traffic control systems along with garbage collection supervision

The name of the project is **Smart Dustbin**. A smart dustbin with IoT-based project can optimize the disposal of waste by monitoring the level in the bins, automatically opening lids, and giving real-time visual of temperature and humidity. It also features like alerts and an LCD display to show real-time status. This helps minimize overflow and optimize collection routes. These bins are now commonplace in homes, offices, and other public areas, supporting the promotion of sustainable waste management and cleanliness. These smart bins improved hygiene through contactless interaction, reduced manual labor and increased efficiency.

1.1 Current scenarios

In developing countries, poor waste management is the source of pollution and public health issues. Garbage bins overflow and become densely packed due to rare collection and absence of real-time monitoring. Overfilled garbage bins release bad odors, attract rodents, and spread dengue, malaria, and cholera diseases. According to the World Bank (2022), the global world generates approximately **2.24 billion tons** of solid waste per annum, of which at least **33%** are not put to environmentally sound disposal. A smart dustbin is the smart solution to make the process of waste collection and manual labor more efficient. Waste collection is the cumbersome task here in Nepal. Hence, we decided to accept this challenge and develop smart dustbin management technology (addon, 2024).

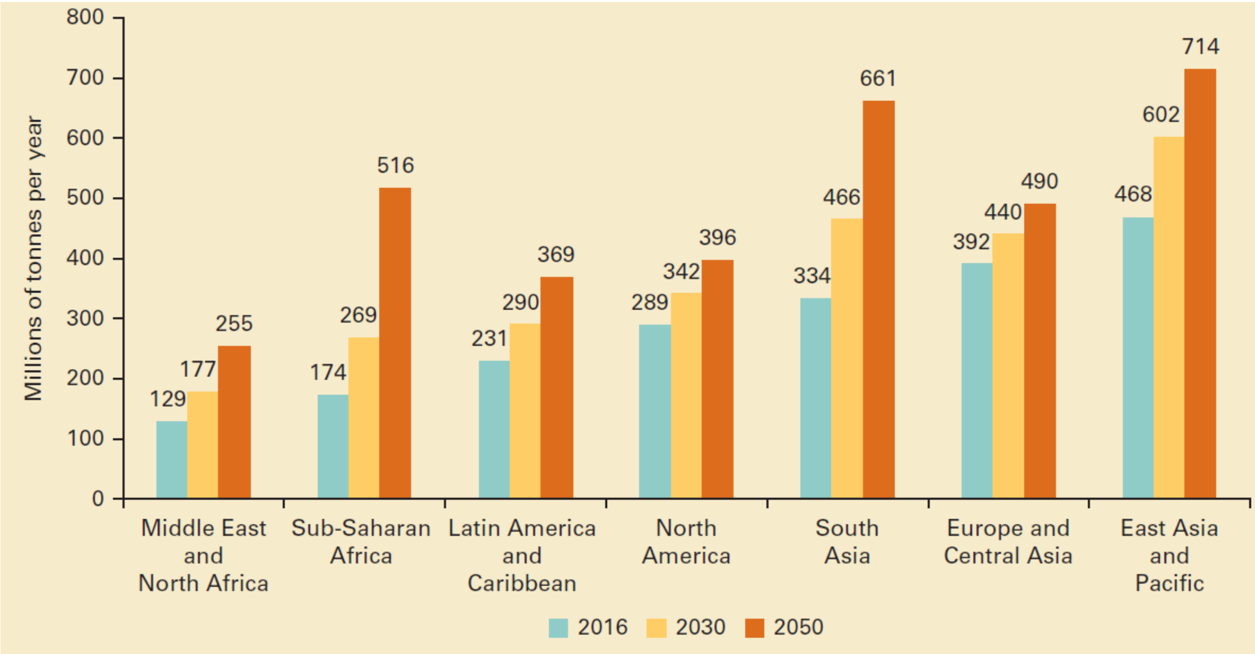


Figure 1: Wastes generated per year (World Bank, 2024)

1.2 Problem Statement and Project as a solution

The traditional way of collecting waste includes collecting unsegregated waste from the houses, which is the major problem of waste management. The existing waste collection system is reactive, inefficient, and lacks real-time data. Garbage bins are often emptied on fixed schedules without regard to actual bin usage, leading to either overflowing bins or unnecessary trips by collection trucks. The combination of full garbage containers and late waste pickup services and poor waste management practices leads to pollution and disease spread while creating public frustration.

To address this, our proposed solution is a **Smart Dustbin System** using IoT technology that aims to automate waste monitoring. The system integrates sensors such as ultrasonic sensors for waste level detection alongside another ultrasonic sensors for presence detection along with servo motors for automatic lid operation, a buzzer and LED indicator for alerts, and additionally uses DHT22 for monitoring temperature and humidity. The LCD display shows real time monitoring environment conditions inside the bin enable real time planning and decision making in waste management. Smart dustbin reduced labor required for monitoring bin status by hand or manually.

2. Aim and Objectives

The main goal of this IoT project is to create a smart dustbin system with automatic opening, detect the fill level of the dustbin, and monitor environmental condition in real-time.

Objectives

- The fill level of the bin will be measured by an ultrasonic sensor.
- The system should operate touchless functions through the combination of another ultrasonic motion detector and a servo motor.
- A DHT22 sensor will monitor both the temperature and humidity inside the device.
- The system activates the buzzer and LED alert as a signal that the bin has reached its maximum capacity.
- To monitor the environmental condition, and all data inside the bin.

3. Background

3.1 System Overview

The smart dustbin system operates using IoT technology to provide a hygienic automated waste system that also promotes operational efficiency. The technology contains sensors that work with a microcontroller (ESP32) over the Internet to minimize manual effort while improving cleanliness.

The system detects human presence, then opens its lid automatically, checks the bin's level of filling, monitoring temperature and humidity inside the bin, and alerts the user when the dustbin level is full. It displays this information on an LCD display.

The ultrasonic sensor detects human activity, which activates a servo motor to operate the lid without operator interaction. Another ultrasonic sensor checks the waste level. The devices will warn you with red lights and buzzer sounds when the bin goes over its capacity. A green light blinks when the bin is empty. A DHT22 sensor checks temperature and humidity located within the bin continuously checks environmental conditions for security and hygiene purposes. The small display screen presents all recorded information and data.

3.2 Design Diagrams

3.2.1 Block Diagram

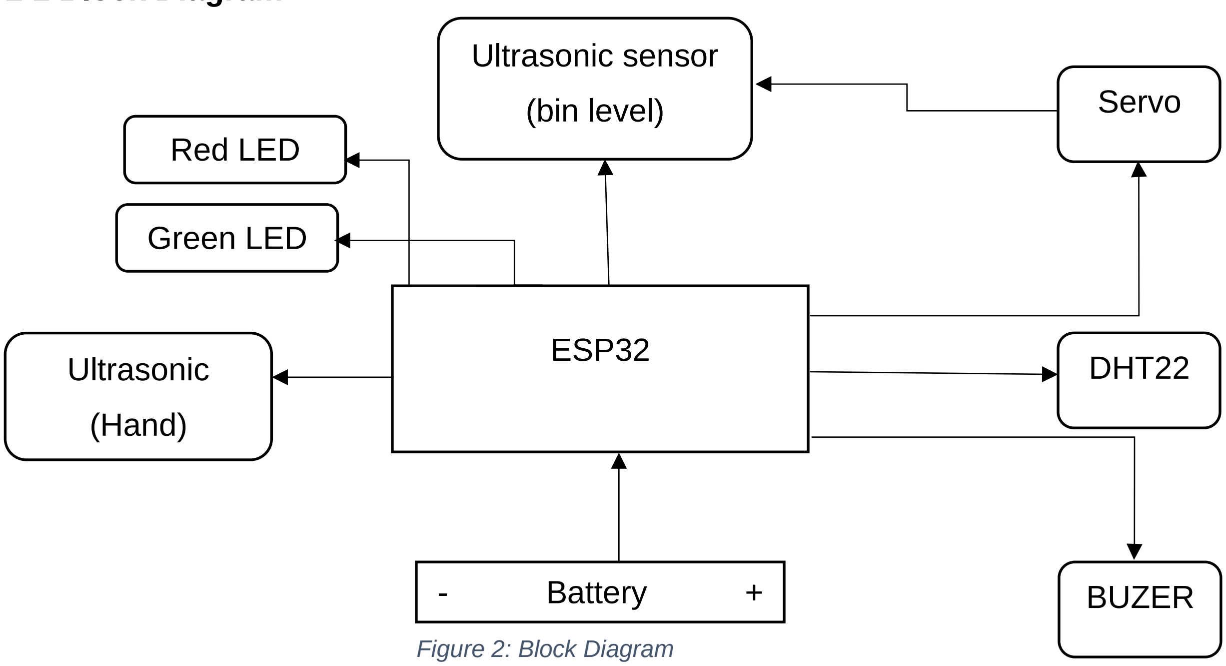


Figure 2: Block Diagram

3.2.2 System architecture

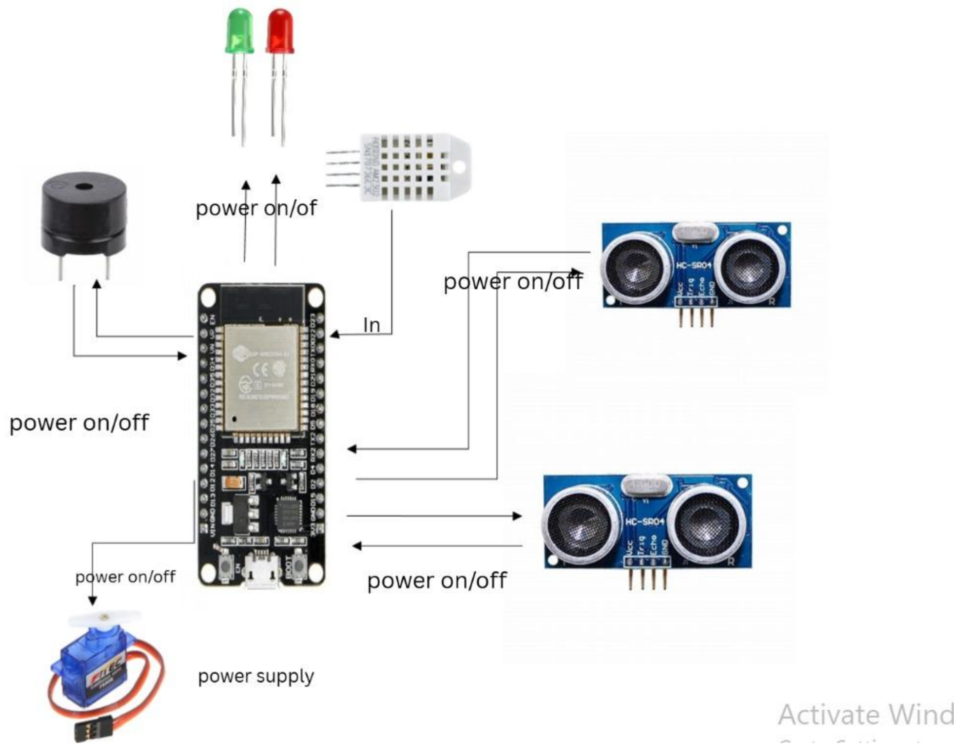


Figure 3: System Architecture diagram

3.2.3 Circuit Diagram

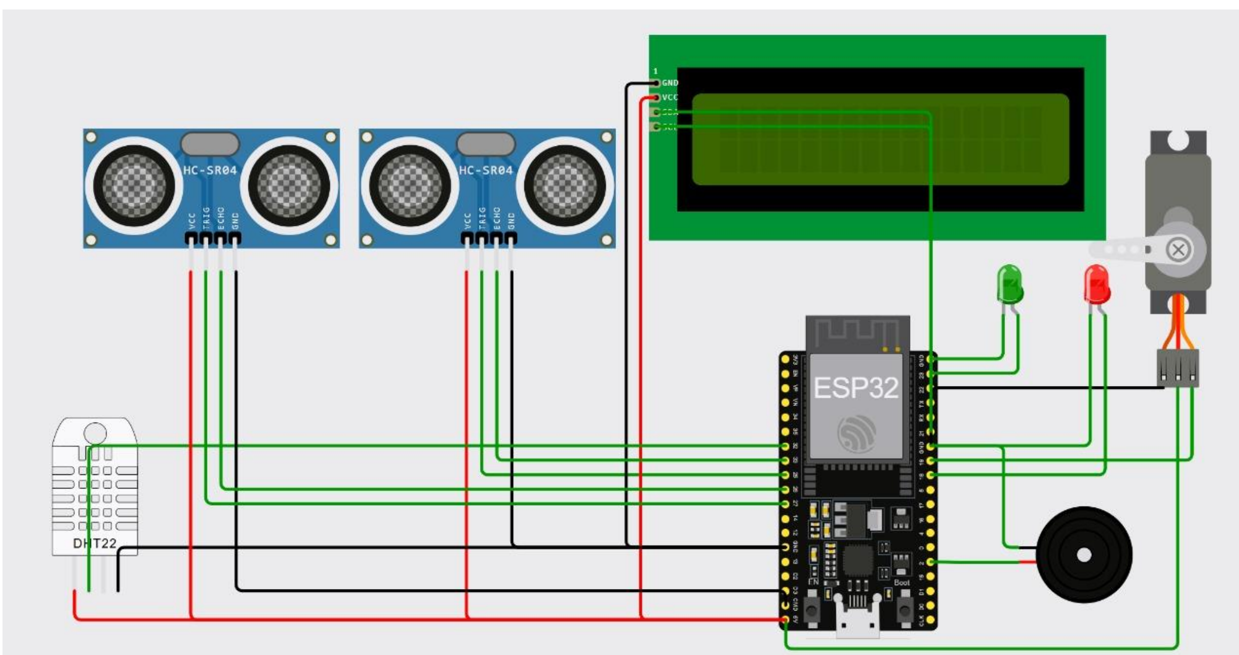


Figure 4: Circuit diagram

3.2.4 Schematic Diagram

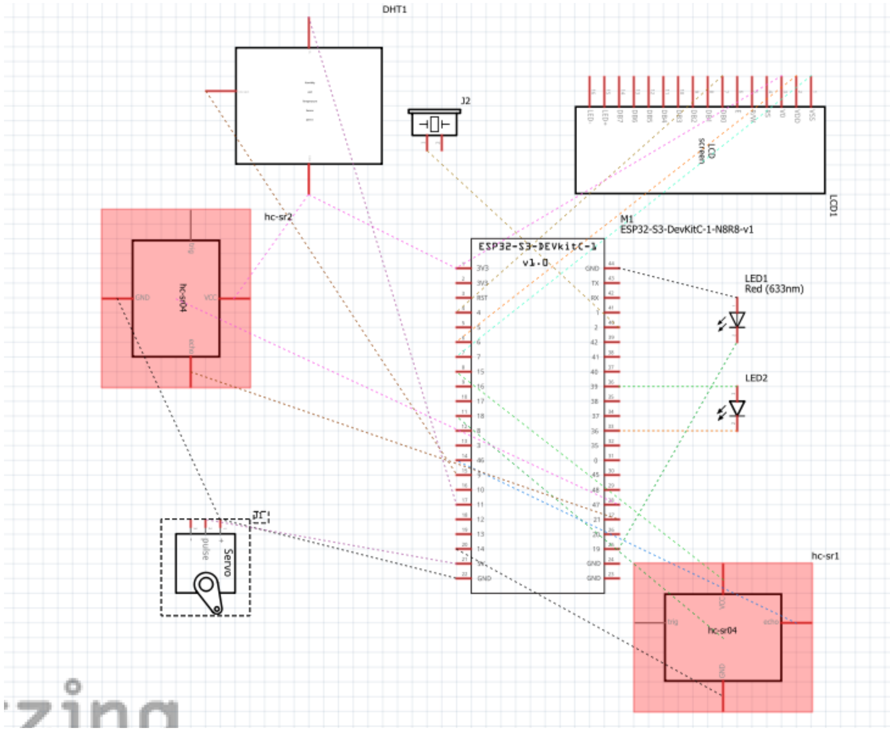


Figure 5: Schematic diagram

3.2.5 Flowchart

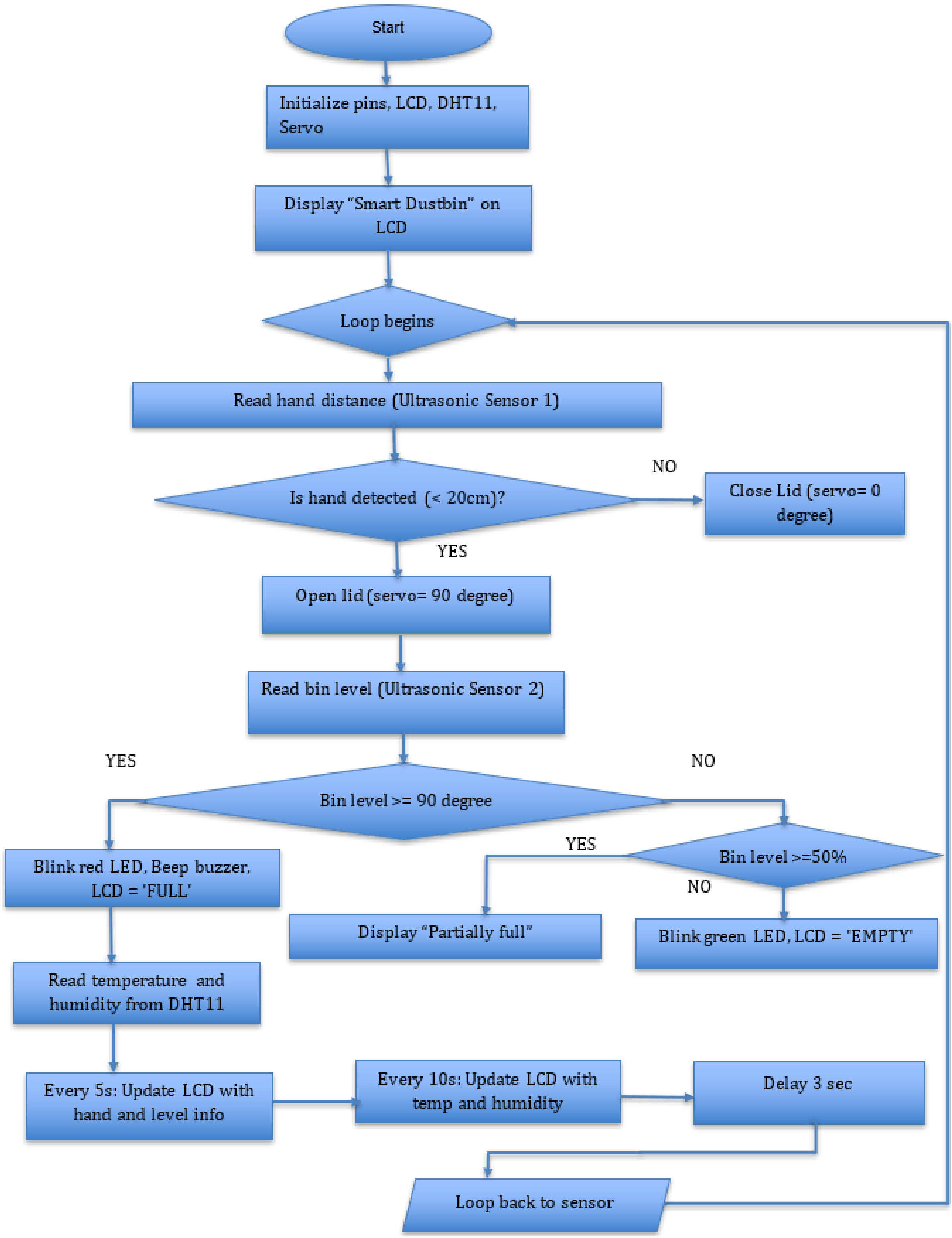


Figure 6: Flowchart

3.3 Requirement Analysis

3.2.1 Hardware Components

ESP32 **HC-SR04 Ultrasonic Sensor** **DHT22/DHT11** **Battery Holder with DC jack**	**SG90 or MG996R Servo Motor** **Jumper Wires (Male-to-Male, Male-to-Female)** **Buzzer / LED Indicator** **Lithium Ion Cells**

Table 1: Table of hardware component

The detailed information of the components are given below:

3.2.2 Software components

- **MS-word:**

  Description: A software architecture style where an application is divided into small, independent services that communicate with each other (Improve, 2021).

- **draw.io:**

  Description: A free, web-based tool used to create and share diagrams such as flowcharts, UML diagrams, and more with features like collaboration and cloud integration (Drawio, 2024).

- **Arduino IDE:**

  Description: Arduino software IDE is open source, making it easy to write code and upload code to the microcontroller board. The software can be used with any Arduino board (Arduino, 2024).

4. Development

4.1 Planning and Design

Our team began the project through analysing how regular waste containers generate problems for users. People face common issues in traditional waste disposal systems, such as overflowing bins, poor hygiene, manual monitoring, and inefficient collection scheduling. Crowded locations make excessive garbage an issue that produces health risks which results in unclean areas. The development of an IoT-operated smart dustbin represented our project's primary focus for management problem resolution.

The main objective of this stage involved the development of functional specifications alongside hardware needs for the system. We chose ESP32 microcontroller because it integrates built-in Wi-Fi features and supports multiple sensors. Our team finalized essential device configurations by choosing ultrasonic sensors to detect motion and check bin levels while adding a DHT22 sensor for temperature and humidity recordkeeping alongside a servo motor for automated lid control, a green LED blinks when the bin is empty and a buzzer alongside red LED for alerts besides an LCD display for status displays.

The system clarity required our team to create a block diagram along with a flowchart and thorough system architecture diagram showing component interactions. These basic actions were established the foundation for efficient system deployment.

4.2 Resource Collection

After finishing design and planning we began gathering all necessary hardware and software to construct the smart dustbin. Our team created a component list which derived from our system plan. The system uses an ESP32 controller as its main device because it provides Wi-Fi functionality and enables connections to multiple sensors while controlling output tools.

The following parts served for sensing and automation functions:

- **Two HC-SR04 ultrasonic sensors:** enable motion detection near the bin while simultaneously measuring its current level.

- **DHT22 sensors:** The interior temperature and humidity measurement within the dustbin utilizes DHT22 sensors.
- **SG90 servo motor:** It works as an automatic system to operate the lid.
- **Two LEDs light:** Red and green function to indicate that the container is full through the red LED but displays a blinking green LED for an empty bin.
- **Buzzer:** to give sound alerts.
- **L2C LCD:** The display serves as a real-time information display system.
- **Jumper wire and Breadboard:** The project needs Jumper wires and breadboard to establish all electrical connections.

In term of Software, we initialized the Arduino IDE development environment while adding required libraries including Servo, DHT, Wire, and LiquidCrystal_I2C.

4.3 System Development

Phase1: We started the development process by creating virtual simulation of the smart dustbin through Wokwi.com, based on Esp32 microcontroller. This helped us visualize the connections and test the logic of each component before physical implementation.

This is my virtual simulation related to the physical implementation, and table of wire connection.

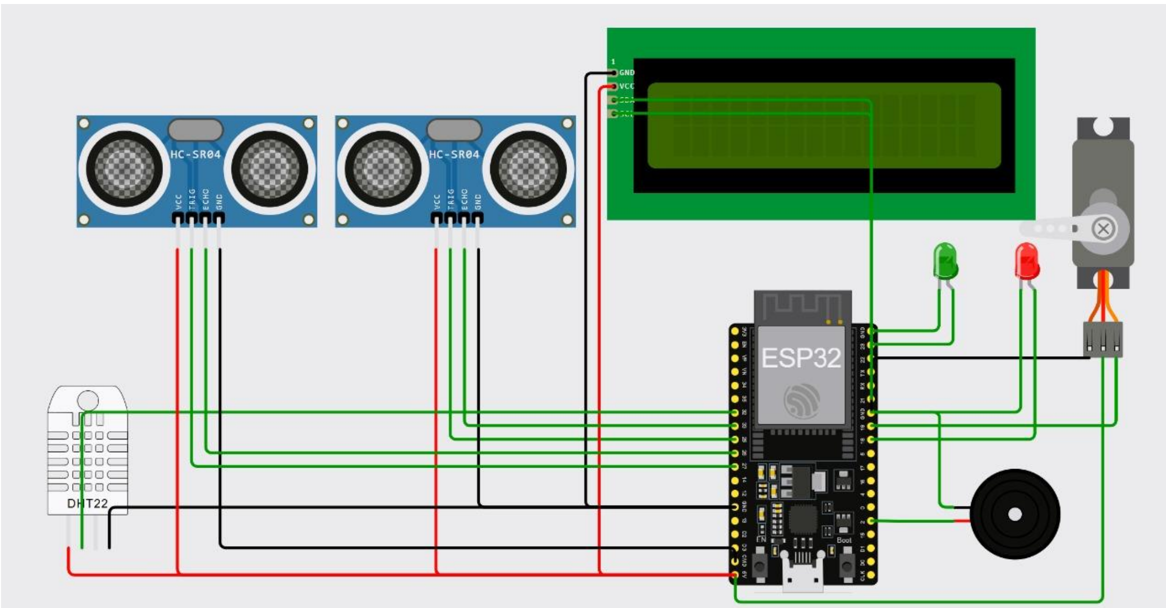


Figure 7: Circuit diagram

Smart Dustbin - ESP32 Pin Mapping Table

Component	**Function**	**GPIO Pin**	**Notes**
Ultrasonic Sensor (1)	Hand Detection Trig	GPIO 27	Detects the user's hand
Ultrasonic Sensor (1)	Hand Detection Echo	GPIO 26	
Ultrasonic Sensor (2)	Level Detection - Trig	GPIO 25	Measures trash level
Ultrasonic Sensor (2)	Level Detection Echo	GPIO 33	
Servo Motor	Lid Control (PWM)	GPIO 19	Opens/closes bin
Buzzer	Audio Alert	GPIO 18	Beeps when bin full
Red LED	Bin Full Indicator	GPIO 21	Blinks when bin full
Green LED	Bin Empty Indicator	GPIO 22	Blinks when bin empty
DHT11 Sensor	Temperature Humidity	GPIO 4	Single-wire data pin
I2C LCD (16x2)	SDA (Data Line)	GPIO 32	Custom SDA pin for I2C
I2C LCD (16x2)	SCL (Clock Line)	GPIO 14	Custom SCL pin for I2C
All Components	GND	GND	Common connection
All Components	Power	3.3V / 5V	Depending on sensor

Table 2: Table of wiring

Phase 2: Once all the components were assembled, we began implementing the system of the smart dustbin. To power the ESP32 and upload code via the Arduino IDE, we first mounted the microcontroller on a breadboard and connected it to a USB. The components were individually connected to their respective GPIO pins by the system. We followed very closely our wiring diagram to avoid accidental connection faults and short circuits. Wiring units are mounted downward slope.

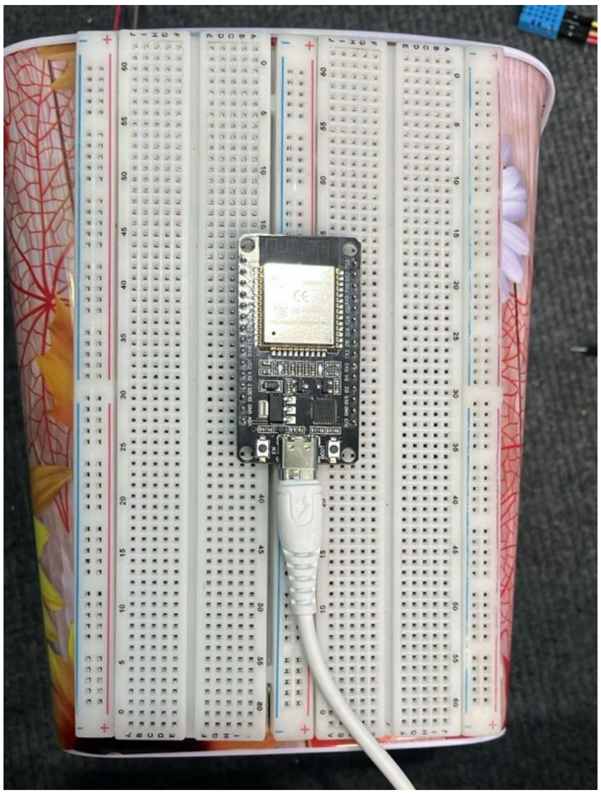


Figure 8: ESP32 without power supply from laptop

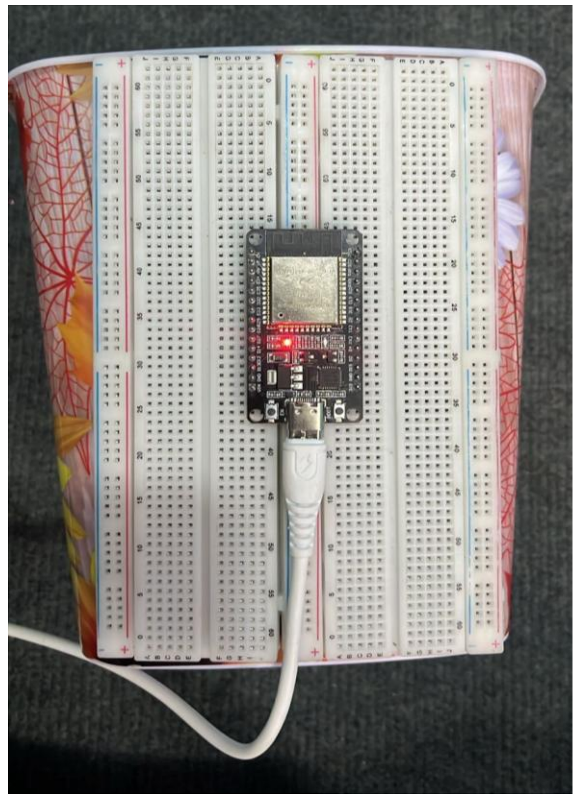


Figure 9: ESP32 with power supply from laptop

Phase 3: The first ultrasonic sensor placed on the face of the dustbin lid functions to identify nearby people within **20 cm**. It is connected to the ESP32 using **GPIO 27(Trig)** and **GPIO 26 (Echo)**, with the power from the **5V pin** and ground **GND**. The sensor tells the ESP32 controller about hand presence or a person approaching within 20 cm. The ESP32 sends a signal via **GPIO 19** to the servo motor, which automatically starts its rotation to a **90-degree** position and opens the lid. The lid stays open when the hand stands close to the dustbin container. When the hand exits outside the 20cm zone, the ESP32 uses the servo motor to rotate to **0 degrees** and close the lid.

Figure 10: Ultrasonic sensor placed outside of bin

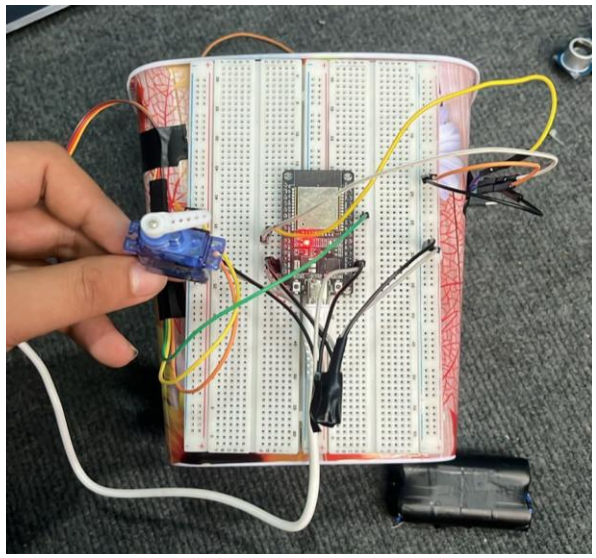


Figure 11: Motor server connected to ESP32

Phase 4: After that, second ultrasonic sensor is placed inside the dustbin to measure bin how the full it is. It is connected through **GPIO 25 (Trig)** and **GPIO 33 (Echo)** to the ESP32, with the power from the **5V pin** and ground **GND**. This device measured distance of dustbin level. Also, green LED is connected to **GPIO 22**, and red LED and buzzer is connected to the **GPIO 21** and **GPIO 18**, where green LED indicates the bin is empty by blink it and red LED and buzzer indicates alert sign, when the bin level crosses above **90%.**

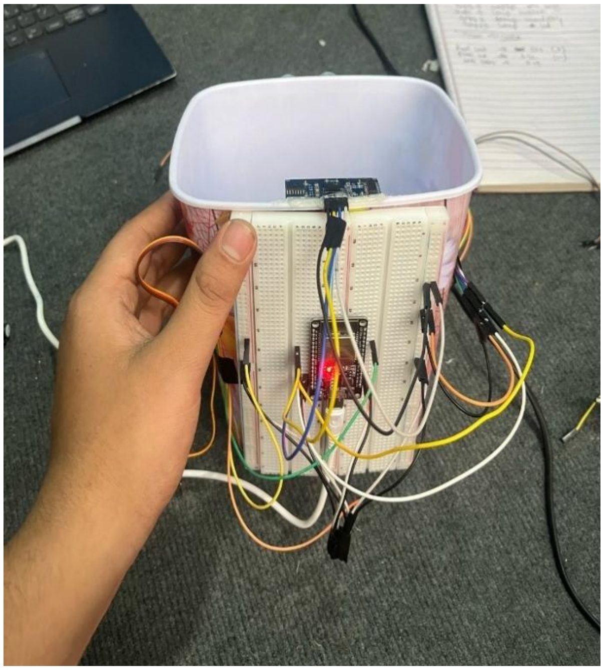


Figure 12:Second ultrasonic sensor connected to ESP32

Figure 13: Bin level from surface dustbin to second ultrasonic sensor

Phase 5: The DHT22 sensor is placed inside the dustbin to check real time temperature and humidity. It is connected to the ESP32 via **GPIO 4**, powered by the **3.3.V** pin, and ground through **GND**. This sensor continuously sends data to the ESP32, which process it and displays the result on the LCD display.

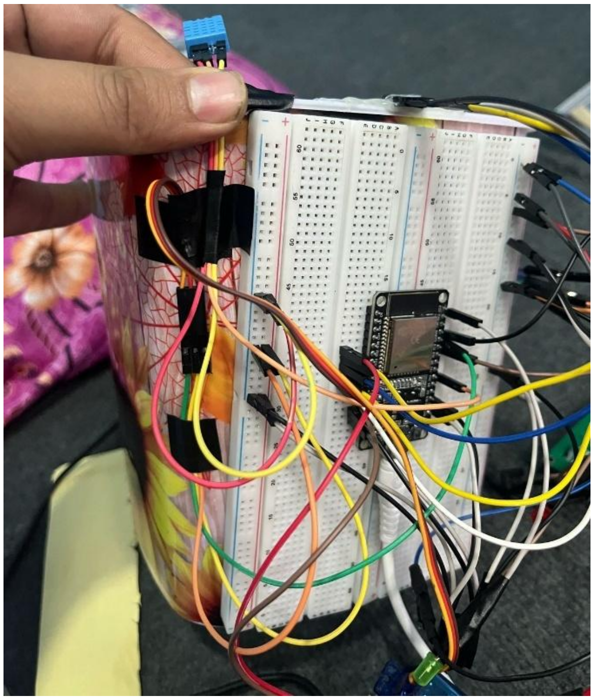


Figure 14: DHT22 sensor connected to ESP32

Figure 15: LCD shows temperature and humidity

Phase 6: At last, I2C LCD is connected to the Esp32 via **GPIO 32 (SDA)** and **GPIO 14 (SCL)**, along with **VCC (5V)** and **GND** for power. It is used to show as real time data such as **Startup message, dustbin fill level, hand detection, temperature and humidity** along with sensor's data.

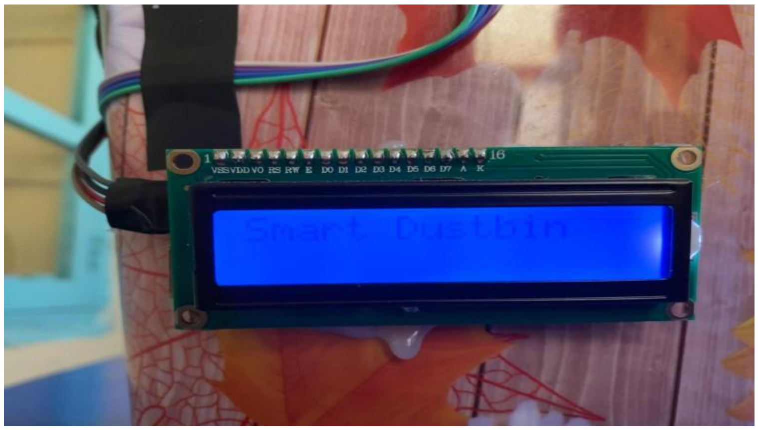


Figure 16: Showing startup message

Figure 17:Showing dustbin level status

Figure 18: Showing hand detection and bin level in percentag

5. Results

At the end of the project, our team successfully developed a complete Internet of Things-based on Smart Dustbin solution with the setup of the ESP32 microcontroller. Two ultrasonic sensors are included in the device, where the external sensor detects a hand nearby and the inside sensor measures the volume of garbage in the bin. The servo motor operates the bin lid automatically following a detected hand approaching within 20 cm and then closes the lid once the hand leaves. The system checks the bin fill status through its sensors, which activate red or green LEDs or sound a buzzer alert based on full or empty situations. The DHT22 sensor detects the temperature and humidity within the bin for improved cleaning. The real-time data monitoring system displays on the 16x2 I2C LCD screen. All system functionalities successfully operated during real-world testing when the system was developed according to guidelines.

6. Testing

6.1 Test 1: Hand Detection and Lid Opening.

Test	1
Objective	To check if the system detects a hand and opens the lid.
Input/Condition	Hand placed within 20 cm of Ultrasonic sensor.
Expected Result	Servo motor rotates to 90°and lid opens until hand detect.
Actual Result	Lid opened as expected
Conclusion	The test was successful.

Table 3: Table of Hand Detection and Lid Opening.

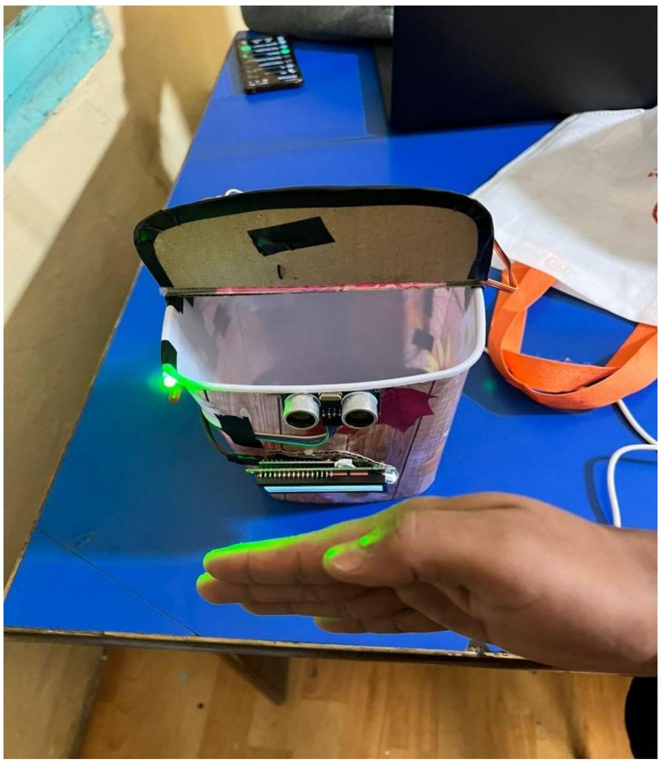


Figure 19: Showing lid open after hand detect within 20cm

6.2 Test 2: Bin Level Detection and Alerts

Test	2
Objective:	To verify bin status detection and correct audio alerts
Input/Condition	Bin at various levels: • Less than 50% • Between 50% - 100% • Greater than 90%
Expected Result	• Less than 50%: Green LED blinks, and LCD display shows Dustbin Empty. • Between 50% - 90%: Green LED blinks, and LCD display shows Dustbin Partial. • Greater than 90%: Red LED blinks, buzzer beep, and LCD display shows Dustbin Full.
Actual Result	All conditions tested and responded correctly.
Conclusion	The test was successful.

Table 4: Table of Bin Level Detection and Alerts

Figure 20: Showing bin status "Dustbin Empty" after bin level below 50%

Figure 21: Showing bin status "Dustbin Partial" after bin level above 50%

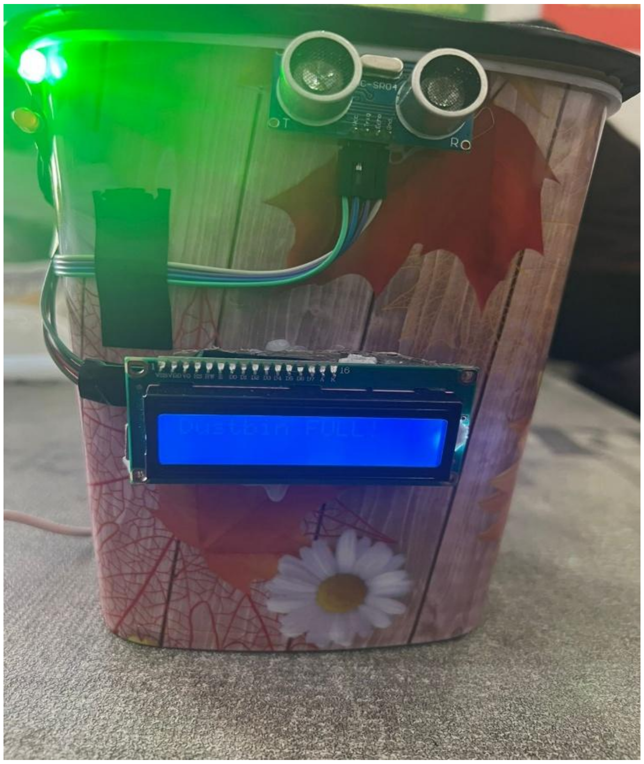


Figure 22: Showing bin status "Dustbin Full" after bin level above 90%

6.3 Test 3: Temperature and Humidity Reading

Test	3
Objective:	To verify DHT22 reads and displays environment data.
Input/Condition	Inside dustbin environment with stable temperature and humidity
Expected Result	LCD displays real-time temperature (°C) and humidity (%) correctly inside of the environment.
Actual Result	Displayed temperature and humidity data.
Conclusion	The test was successful.

Table 5: Table of Temperature and Humidity Reading

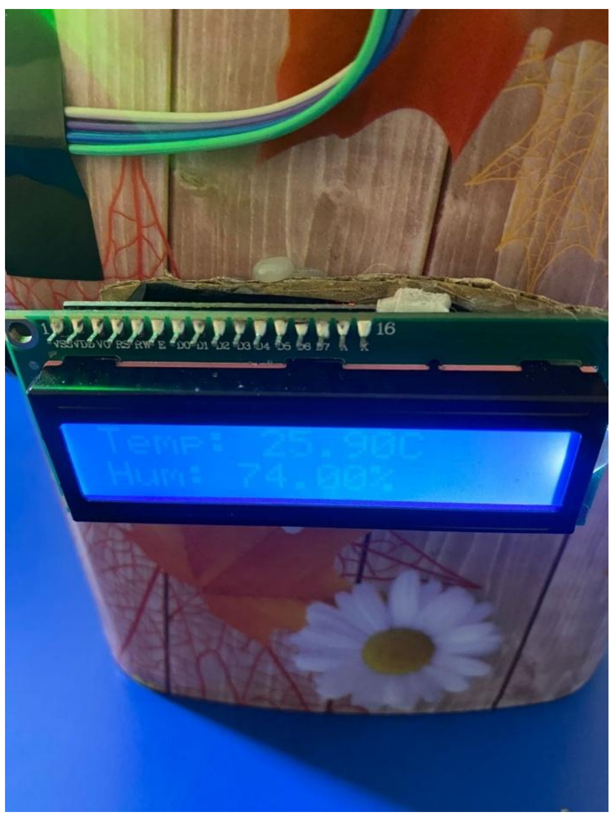


Table 6: LCD showing temperature and humidity reading

6.4 Test 4: LCD shows live bin in percentage level.

Test	4
Objective:	To confirm real-time bin level is shown.
Input/Condition	Trash placed and monitored on LCD.
Expected Result	Bin level in % displayed correctly on LCD.
Actual Result	The percentage of bin level shows on LCD display.
Conclusion	The test was successful.

Table 7: Table of LCD shows live bin in percentage level.

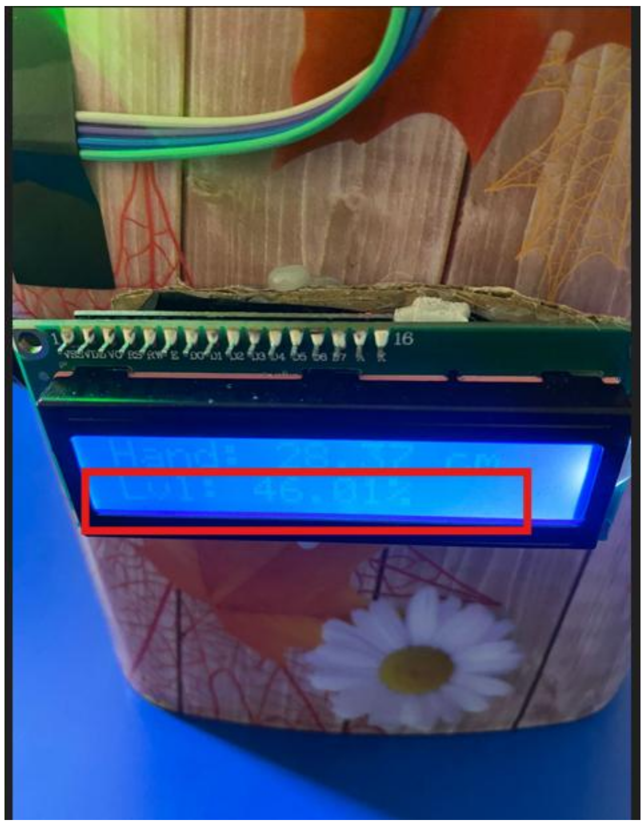


Table 8: LCD shows bin level in percentage

6.5 Test 5: DHT22 sensor is disconnected

Test	5
Objective:	To test system behavior when a DHT22 sensor fails.
Input/Condition	DHT22 wire removed during operation.
Expected Result	The message “Failed to read from DHT sensor!” was shown in the serial monitor, but the system kept running without crashing or freezing.
Actual Result	Error message shown in serial monitor; other system continues operate.
Conclusion	The test was unsuccessful.

Table 9: Table of DHT22 sensor is disconnected

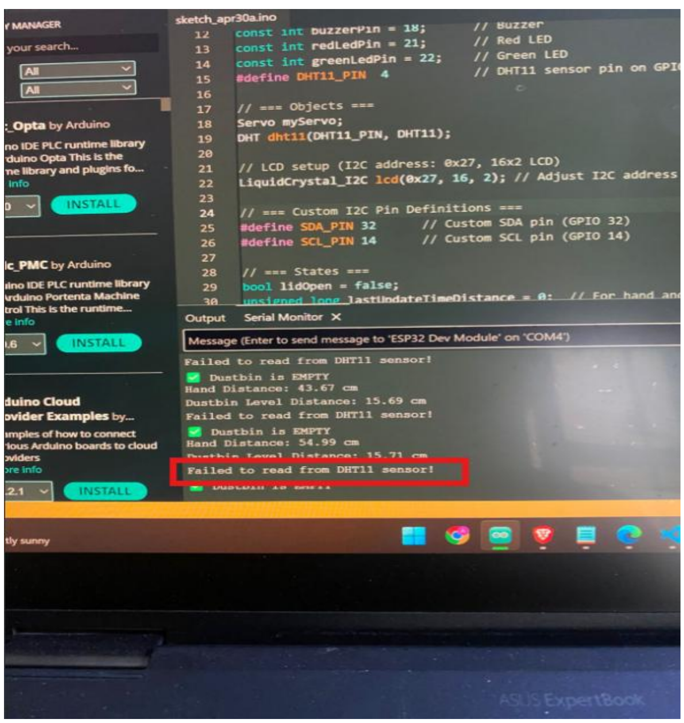


Figure 23: Screenshot of serial monitor show data fail to read from DHT22 sensor

7. Future Works

We are planning to add some new features to the Smart Dustbin in the future to make it smarter and more useful. Adding a gas monitor like the MQ-135 is one idea. These sensors can find harmful gases like smoke, carbon monoxide, ammonia, and methane. This will help keep the area around the bin safer and check for bad smells. It will also be able to get smart warnings. The trash can let people know when the gas level is too high by beeping, blinking LEDs, or showing a message on the LCD screen and closed the dustbin permanently.

The device will also be able to read sensors in real time and store the information. This data can be sent to a cloud platform or over Wi-Fi or Bluetooth. People who use cloud computing can get to their data from anywhere, see reports, and keep an eye on how things change over time. This makes it easier to keep track of air quality and better handle waste in general.

8. Conclusion

This project successfully achieved its goal of creating a Smart Dustbin system using IoT technology and the ESP32 microcontroller. The system was designed to improve hygiene and make waste management more efficient through automation and real-time monitoring.

Sensors like ultrasonic modules and DHT22, dustbin can become automated. When a hand is close by, the lid will open automatically. It can also measure the bin's full and determine the humidity and temperature within. The system uses an LCD display, buzzer, and LEDs to alert people based on the fill level. The study also demonstrated that there is a strong possibility that many components were work well together with appropriate wiring, logical development, and coding.

The system worked well in several situations where it was implemented. It can be used in public spaces, workplaces, schools, and households to promote cleanliness and reduce the amount of waste that is handled by hand. This project did not only give us an opportunity to practically apply our technical knowledge but also make us think of how technology can make hygiene better, reduce human efforts and make the most basic task smarter and more efficient. Overall, this project provided a realistic and brilliant response to a real-world problem.

References

addon, 2024. *addon.com.* [Online]
Available at: https://www.addon.com.np/smart-dust-bin-in-nepal.html
[Accessed 07 04 2025].

Agarwal, T., 2021. *ElProCus - Electronic Projects for Engineering Students.* [Online]
Available at: https://www.elprocus.com/buzzer-working-applications/
[Accessed 13 5 2025].

Arduino, 2024. *Arduino.cc.* [Online]
Available at: https://www.arduino.cc/en/software/
[Accessed 13 5 2025].

Drawio, 2024. *About draw.io.* [Online]
Available at: https://www.drawio.com/about
[Accessed 13 5 2025].

Electronicwings, 2024. *Electronicwings.com.* [Online]
Available at: https://www.electronicwings.com/
[Accessed 15 5 2025].

Improve, {. F., 2021. *GeeksforGeeks.* [Online]
Available at: https://www.geeksforgeeks.org/introduction-to-microsoft-word/
[Accessed 13 5 2025].

Nick, 2023. *DeepSea.* [Online]
Available at: https://www.deepseadev.com/en/blog/esp32-chip-explained-and-advantages/
[Accessed 13 5 2025].

Robocraze, 2022. *All about servo motor SG90.* [Online]
Available at: https://robocraze.com/blogs/post/all-about-servo-motor-sg90
[Accessed 07 04 2025].

Santos, S., 2024. *Random Nerd Tutorials.* [Online]
Available at: https://randomnerdtutorials.com/esp32-servo-motor-web-server-arduino-

ide/
[Accessed 15 5 2025].

sparkfun, 2024. *Ultrasonic distance sensor - 5V (HC-SR04).* [Online]
Available at: https://www.sparkfun.com/ultrasonic-distance-sensor-hc-sr04.html
[Accessed 07 04 2025].

waveshare, 2024. *DHT22 temperature-humidity sensor - waveshare wiki.* [Online]
Available at: https://www.waveshare.com/wiki/DHT22_Temperature-Humidity_Sensor
[Accessed 7 4 2025].

World Bank, 2024. *Worldbank.org.* [Online]
Available at: https://datatopics.worldbank.org/what-a-waste/trends_in_solid_waste_management.html?ref=goodoil.news
[Accessed 3 5 2025].

Appendix

- **ESP32** is a low-cost Wi-Fi microcontroller developed by Espressif Systems. It consists of a 32-bit processor (80–160 MHz), support for RTOS, and low power consumption, making it a good fit for IoT and battery-powered applications. The chip supports 802.11 b/g/n Wi-Fi and features useful peripherals like ADC, UART, SPI, and I2C. It can be programmed using Arduino IDE and MicroPython platforms. Priced low (around $3), the NodeMCU dev board is a favorite with 4MB flash, 11 GPIOs, and 10-bit ADC—a great option for beginners and wireless projects (Nick, 2023).

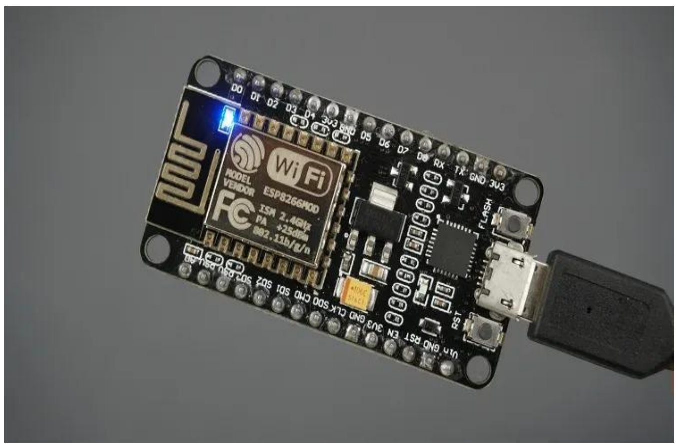


Figure 24: Esp32

- This is the **HC-SR04** ultrasonic distance sensor. This is a low-cost sensor that provides 2cm to 400cm of non-contact measuring ability with a ranging accuracy that can be as low as 3mm. Each HC-SR04 module has an ultrasonic transmitter, a receiver and a control circuit. You only have to deal with four pins on the HC-SR04: VCC (Power), Trig (Trigger), Echo (Receive), and GND (Ground) (sparkfun, 2024).

Figure 25: HC-SR04

- This is a calibrated digital temperature and humidity module on-board sensor **DHT22 (AM2302),** which has better precision and wider measuring range than DHT11. It can be used for ambient temperature and humidity detection, utilizing the standard single-wire interface (waveshare, 2024).

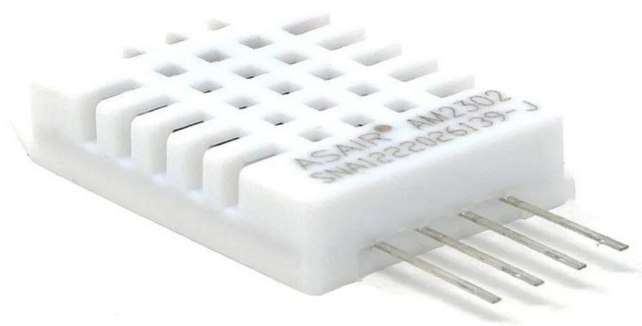


Figure 26: DHT22

- **Motor server SG90:** The motor is called SG90 because it rotates from 0 to 180 degrees in 90-degree intervals. The server motor offers more than just decrease the motor's rotational speed by lowering its RPM and increasing torque (Robocraze, 2022).

Figure 27: Motor server

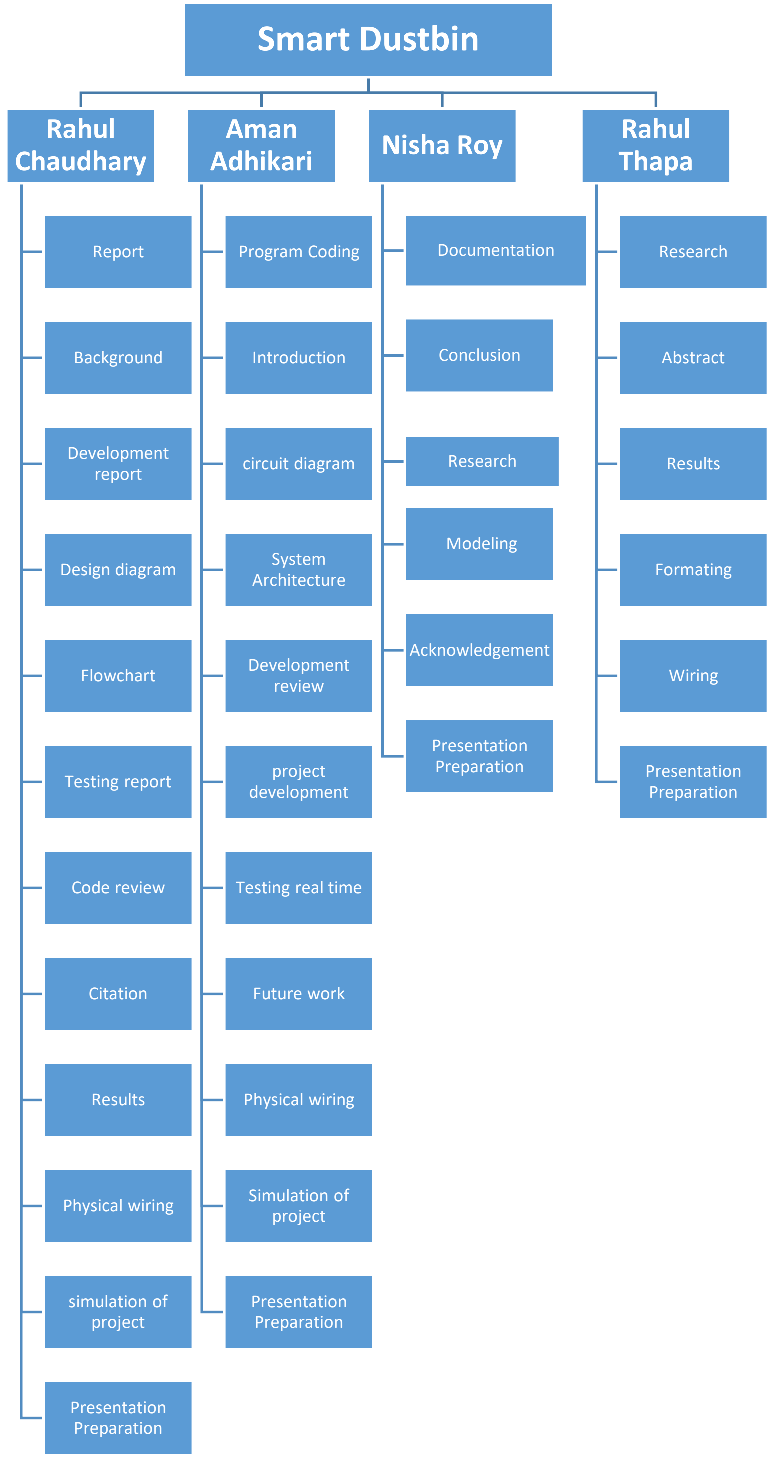
- **Buzzer:** A buzzer is an electric component. It is used for audio signalling device which provide an audible alert (Agarwal, 2021).

Figure 28: Buzzer

Individual Contribution plan

Student Name	Task	Contribution
Rahul Chaudhary	• **Proposal:** Abstract, Introduction, Aim and objectives, Current scenario, problem statement, project as solution, and citation. • **Main Report:** Introduction, background, Development, design diagram, Flowchart, testing, code review, citation, Results, wiring, and simulation of project • **Presentation:** Requirement analysis and system overview.	30%
Aman Adhikari	• **Proposal:** Acknowledgement, Requirement Analysis, expected outcomes and deliveries, system architecture of the project, and report formatting. • **Main Report:** Background, circuit diagram, Development, coding, testing development review, Future work, wiring, and simulation of project • **Presentation:** Development and testing of the project.	30%

Nisha Roy	• **Main Report:** Acknowledgment, Conclusion, Checking wiring between device and Documentation • **Presentation**	20%
Rahul Thapa	• **Main report:** formatting, Abstract, Results • **Presentation:** Future work and expectation outcome of the project	20%


Smart Dustbin
Rahul Chaudhary
Aman Adhikari
Nisha Roy
Rahul Thapa
Report
Background
Development report
Design diagram
Flowchart
Testing report
Code review
Citation
Results
Physical wiring
simulation of project
Presentation Preparation
Program Coding
Introduction
circuit diagram
System Architecture
Development review
project development
Testing real time
Future work
Physical wiring
Simulation of project
Presentation Preparation
Documentation
Conclusion
Research
Modeling
Acknowledgement
Presentation Preparation
Research
Abstract
Results
Formating
Wiring
Presentation Preparation

Source code of my Smart Dustbin

```
#include <ESP32Servo.h>

#include <DHT.h>

#include <Wire.h>

#include <LiquidCrystal_I2C.h>


// === Pins ===

const int trigPin = 27;        // Hand detection - Trigger

const int echoPin = 26;        // Hand detection - Echo

const int trigPinLevel = 25;   // Level detection - Trigger

const int echoPinLevel = 33;   // Level detection - Echo

const int servoPin = 19;       // Servo motor

const int buzzerPin = 18;      // Buzzer

const int redLedPin = 21;      // Red LED

const int greenLedPin = 22;    // Green LED

#define DHT11_PIN  4           // DHT11 sensor pin on GPIO4


// === Objects ===

Servo myServo;

DHT dht11(DHT11_PIN, DHT11);

LiquidCrystal_I2C lcd(0x27, 16, 2); // Adjust I2C address if needed


// === Custom I2C Pin Definitions ===
```

```
#define SDA_PIN 32      // Custom SDA pin (GPIO 32)
#define SCL_PIN 14      // Custom SCL pin (GPIO 14)

// === States ===
bool lidOpen = false;
unsigned long lastUpdateTimeDistance = 0;
unsigned long lastUpdateTimeTempHumidity = 0;
const unsigned long updateIntervalDistance = 5000;
const unsigned long updateIntervalTempHumidity = 10000;

// === Setup ===
void setup() {
  Serial.begin(115200);
  Wire.begin(SDA_PIN, SCL_PIN);

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(trigPinLevel, OUTPUT);
  pinMode(echoPinLevel, INPUT);
  pinMode(buzzerPin, OUTPUT);
  pinMode(redLedPin, OUTPUT);
  pinMode(greenLedPin, OUTPUT);
```

```
  myServo.attach(servoPin);
  myServo.write(0); // Lid starts closed

  dht11.begin();

  lcd.begin(16, 2);
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Smart Dustbin");
  delay(2000);
  lcd.clear();
}

// === Function to read distance ===
float getDistance(int trig, int echo) {
  digitalWrite(trig, LOW);
  delayMicroseconds(2);
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);

  long duration = pulseIn(echo, HIGH, 30000); // 30ms timeout
  return duration * 0.034 / 2;
```

```
}

// === Function to calculate bin level percentage ===
float calculateBinLevelPercentage(float distance, float maxDistance) {
  return ((maxDistance - distance) / maxDistance) * 100;
}

// === Main loop ===
void loop() {
  float handDistance = getDistance(trigPin, echoPin);
  float levelDistance = getDistance(trigPinLevel, echoPinLevel);
  float maxDistance = 25.0;
  float binLevelPercentage = calculateBinLevelPercentage(levelDistance, maxDistance);

  float humi  = dht11.readHumidity();
  float tempC = dht11.readTemperature();

  Serial.print("Hand Distance: ");
  Serial.print(handDistance);
  Serial.println(" cm");

  Serial.print("Dustbin Level Distance: ");
  Serial.print(levelDistance);
```

```
Serial.println(" cm");


if (isnan(tempC) || isnan(humi)) {
  Serial.println("Failed to read from DHT11 sensor!");
} else {
  Serial.print("Humidity: ");
  Serial.print(humi);
  Serial.print("%  |  ");
  Serial.print("Temperature: ");
  Serial.print(tempC);
  Serial.println("°C");
}


unsigned long currentMillis = millis();


// Open/close lid based on hand detection
if (handDistance > 0 && handDistance < 20 && !lidOpen) {
  Serial.println("Hand detected. Opening lid...");
  myServo.write(90);
  lidOpen = true;
} else if ((handDistance >= 20 || handDistance <= 0) && lidOpen) {
  Serial.println("Hand removed. Closing lid...");
  myServo.write(0);
```

```
    lidOpen = false;
    lcd.clear();  // Clear LCD when lid closes
  }

  // === Display info only when lid is open ===
  if (lidOpen) {
    if (currentMillis - lastUpdateTimeDistance >= updateIntervalDistance) {
      lastUpdateTimeDistance = currentMillis;
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("Hand: ");
      lcd.print(handDistance);
      lcd.print(" cm");
      lcd.setCursor(0, 1);
      lcd.print("Lvl: ");
      lcd.print(binLevelPercentage);
      lcd.print("%");
    }

    if (currentMillis - lastUpdateTimeTempHumidity >= updateIntervalTempHumidity) {
      lastUpdateTimeTempHumidity = currentMillis;
      lcd.clear();
      lcd.setCursor(0, 0);
```

```
    lcd.print("Temp: ");
    lcd.print(tempC);
    lcd.print("C");
    lcd.setCursor(0, 1);
    lcd.print("Hum: ");
    lcd.print(humi);
    lcd.print("%");
  }
}

// === Bin Level Alerts (Serial + LEDs/Buzzer) ===
if (binLevelPercentage >= 90) {
  Serial.println("⚠ Dustbin is FULL!");
  digitalWrite(greenLedPin, LOW);

  for (int i = 0; i < 3; i++) {
    digitalWrite(redLedPin, HIGH);
    delay(300);
    digitalWrite(redLedPin, LOW);
    delay(300);
  }
```

```
    for (int i = 0; i < 3; i++) {
      digitalWrite(buzzerPin, HIGH);
      delay(300);
      digitalWrite(buzzerPin, LOW);
      delay(300);
    }
  } else if (binLevelPercentage >= 50) {
    Serial.println("🔄 Dustbin is PARTIALLY FULL");
    digitalWrite(buzzerPin, LOW);
    digitalWrite(redLedPin, LOW);
    digitalWrite(greenLedPin, LOW);
  } else {
    Serial.println("✅ Dustbin is EMPTY");
    digitalWrite(buzzerPin, LOW);
    digitalWrite(redLedPin, LOW);
    for (int i = 0; i < 3; i++) {
      digitalWrite(greenLedPin, HIGH);
      delay(300);
      digitalWrite(greenLedPin, LOW);
      delay(300);
    }
  }
```

```
  delay(300); // Slight delay for stability
}
```

Source: (Electronicwings, 2024) and (Santos, 2024)

The End