ABSTRACT

The Smart Waste Management System is a groundbreaking solution designed to modernize waste disposal through the integration of IoT technology. At its core, the system uses ultrasonic sensors connected to an ESP32 microcontroller to measure garbage bin levels, ensuring accurate monitoring. When bins approach capacity, the system triggers real-time notifications to municipal authorities via a cloud-based platform, enabling prompt collection and preventing unsightly overflows.Additional features such as GPS tracking provide precise bin location data, while a user-friendly dashboard offers comprehensive visualization of waste levels, collection schedules, and optimized routes. This not only enhances efficiency but also reduces fuel consumption and operational costs by streamlining waste collection logistics. Wireless connectivity and real-time data transmission ensure seamless operation, contributing to cleaner and more organized urban environments.The system's design promotes sustainability by addressing common inefficiencies in waste management and reducing the environmental footprint of collection processes. Future developments could include AI-driven predictive waste analysis for proactive planning and renewable energy sources to power the system, further aligning with green initiatives. These enhancements would expand its capabilities, supporting smarter cities and a healthier planet.

SYNOPSIS

The Smart Waste Management System uses IoT technology to optimize waste disposal by automating garbage bin monitoring with ultrasonic sensors connected to an ESP32 microcontroller. It sends real-time notifications to authorities via a cloud platform when bins reach a set threshold, ensuring timely collection and preventing overflow. Integrated GPS enables bin location tracking, while a user-friendly interface supports data visualization and route optimization. With features like wireless connectivity and real-time monitoring, the system reduces costs, minimizes environmental impact, and promotes cleaner surroundings. Future enhancements could include AI-driven waste prediction and renewable energy integration for sustainability.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

1. IoT - Internet of Things
2. ESP32 - Espressif Systems 32-bit Microcontroller
3. Wi-Fi - Wireless Fidelity
4. GPIO - General Purpose Input/output
5. IDE - Integrated Development Environment

CHAPTER – 1

INTRODUCTION

1.1 INTRODUCTION

The Smart Waste Management System is an innovative solution designed to address inefficiencies in traditional waste disposal processes by leveraging IoT technology. It employs ultrasonic sensors connected to an ESP32 microcontroller to monitor garbage bin levels in real time. When a bin reaches a predefined threshold, notifications are sent to authorities through a cloud-based platform, ensuring timely waste collection and preventing overflow. The system is further equipped with GPS for precise bin location tracking and a user-friendly interface for visualizing data, enabling efficient route optimization for collection vehicles. By incorporating wireless connectivity and real-time monitoring, this system not only reduces operational costs and environmental impact but also promotes cleaner urban areas and better public hygiene. Future enhancements may include AI for predictive waste analytics, integration of renewable energy sources to power the system, and advanced machine learning algorithms to further optimize waste management processes, contributing to sustainable and smart urban development.

1.2 BLOCK DIAGRAM

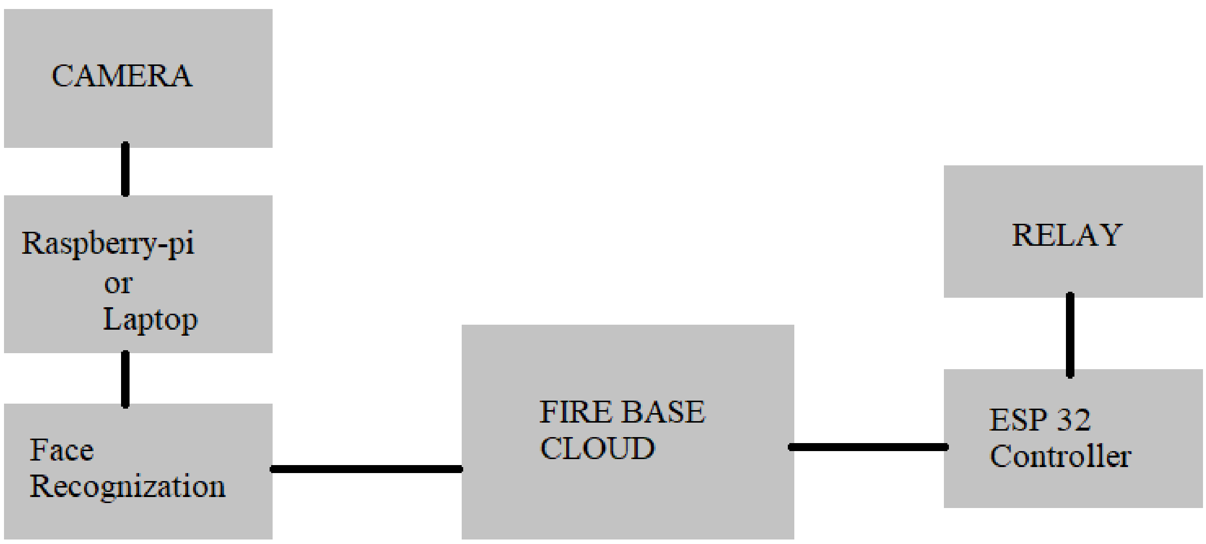

Fig 1.2 Block Diagram

The Fig 1.2 Smart Waste Management System's block diagram consists of several interconnected modules working seamlessly. At the core is the ESP32 microcontroller, which serves as the control unit. Ultrasonic sensors are attached to the bins to measure waste levels by emitting sound waves and detecting their reflection. When the waste level crosses a predefined threshold, the ESP32 processes this data and sends it to a **cloud platform** using built-in Wi-Fi for wireless connectivity. The cloud platform, integrated with a **GPS module**, tracks bin locations and relays real-time notifications to authorities via a **mobile or web application interface**. The interface displays bin statuses, waste levels, and optimized collection routes for efficient logistics. A **power supply unit**, potentially supported by renewable energy sources, powers the system. This integration ensures automation, enhances operational efficiency, prevents bin overflow, and reduces environmental impact.

1.3 CONCLUSION

The Smart Waste Management System employs a block diagram consisting of key components like garbage bins equipped with ultrasonic sensors, an ESP32 microcontroller, a power supply unit, a GPS module, and a wireless communication module (Wi-Fi or GSM). The ultrasonic sensors continuously monitor the bin's fill level, and the ESP32 processes this data. When the bin's capacity exceeds a predefined threshold, the system sends alerts via the cloud platform to waste management authorities. The GPS module provides real-time bin location, and the data is visualized on a user-friendly interface for efficient route optimization. This setup enables seamless monitoring, timely waste collection, and operational efficiency. In conclusion, the project addresses waste management challenges by leveraging IoT for automation, reducing environmental impact, and enhancing urban cleanliness. Future developments could incorporate AI for predictive analytics and renewable energy for sustainable operation.

CHAPTER – 2

PREVIOUS WORK

2.1 INTRODUCTION

The Smart Waste Management System builds on existing efforts to address inefficiencies in traditional waste collection methods, which often rely on fixed schedules rather than actual waste levels, leading to overflowing bins or unnecessary trips. Previous work in waste management systems has explored the use of sensors, IoT, and cloud-based platforms to improve real-time monitoring and operational efficiency. Research has shown that integrating ultrasonic sensors, microcontrollers, and wireless communication can effectively monitor bin levels and trigger notifications for timely waste collection. Additionally, GPS-enabled tracking and data visualization tools have been investigated to optimize collection routes, reducing fuel consumption and labor costs. However, earlier systems often lacked user-friendly interfaces, scalability, or energy-efficient designs, leaving room for improvement. Building on these advancements, the Smart Waste Management System enhances the process by incorporating advanced IoT technology, seamless wireless connectivity, and features like GPS tracking and cloud integration, offering a more sustainable, efficient, and cost-effective solution.

2.2 PREVIOUS WORKS

"IoT-Based Smart Waste Bin Monitoring System" (2020) by R. Singh,K.Sharma

This research introduced a smart waste bin monitoring system that used ultrasonic sensors to detect bin levels and ESP8266 for wireless communication. The data was transmitted to a cloud platform for real-time monitoring, enabling efficient waste collection scheduling. The study emphasized reducing operational costs and preventing overflow in urban areas.

"Smart Waste Management System with GPS Integration" (2019) by A.Gupta,S.Jain

This work developed an IoT-enabled waste management solution using GPS modules for bin location tracking and GSM for data transmission. The system provided authorities with precise bin locations and real-time waste levels, optimizing collection routes and improving efficiency in waste disposal.

"Ultrasonic Sensor-Based Waste Monitoring System" (2018) by Y.

Zhang,L.Cheng

The authors proposed an automated waste management system using ultrasonic sensors to monitor waste levels in bins. The data was processed using a microcontroller and transmitted via Wi-Fi to a central dashboard. The study highlighted the potential for reducing manual inspections and promoting cleaner surroundings.

"Cloud-Connected Waste Management System for Smart Cities" (2021)byP.Rani,M.Verma

This study presented a cloud-enabled waste management system that combined IoT sensors and cloud computing for real-time bin status monitoring. The system used MQTT for data communication and provided a user-friendly interface for visualizing bin locations and statuses. The research underscored the importance of cloud scalability in handling urban waste challenges.

"Sustainable Waste Collection Using IoT and Renewable Energy" (2022)byS.Ahmed,T.Khan

This research proposed a sustainable waste management system that used solar-powered IoT devices to monitor and report bin levels. The integration of renewable energy aimed to enhance system sustainability while reducing environmental impact. The study suggested future improvements in data analytics for predictive waste management.

These previous works provide insights into the evolution of IoT-based waste management solutions and their contributions to urban sustainability.

2.3 SUMMARY

The Smart Waste Management System utilizes IoT technology to optimize traditional waste disposal by automating garbage bin monitoring with ultrasonic sensors and an ESP32 microcontroller. When bins reach a set threshold, real-time notifications are sent via a cloud platform to ensure timely collection, preventing overflow. Equipped with GPS for location tracking and a user-friendly interface for data visualization and route optimization, the system enhances efficiency, reduces costs, and minimizes environmental impact. With features like wireless connectivity and real-time monitoring, it promotes sustainability and cleaner surroundings. Future advancements could include AI-driven waste analysis and renewable energy integration.

CHAPTER – 3

PROPOSED WORK

3.1 INTRODUCTION

The proposed Smart Waste Management System aims to revolutionize traditional waste disposal methods by leveraging IoT technology to enhance efficiency and sustainability. The system incorporates ultrasonic sensors connected to an ESP32 microcontroller to monitor garbage bin levels in real-time. When bins approach a predefined threshold, alerts are sent to waste management authorities via a cloud-based platform, ensuring timely collection and preventing overflow. The system also integrates GPS for precise bin location tracking and provides a user-friendly interface for visualizing data and optimizing collection routes. By utilizing wireless connectivity and real-time monitoring, the solution minimizes operational costs, reduces environmental impact, and promotes cleaner urban environments. Future advancements could include AI for predictive waste analysis and renewable energy integration to further enhance sustainability and operational effectiveness.

3.2 CIRCUIT DIAGRAM AND EXPLANATION

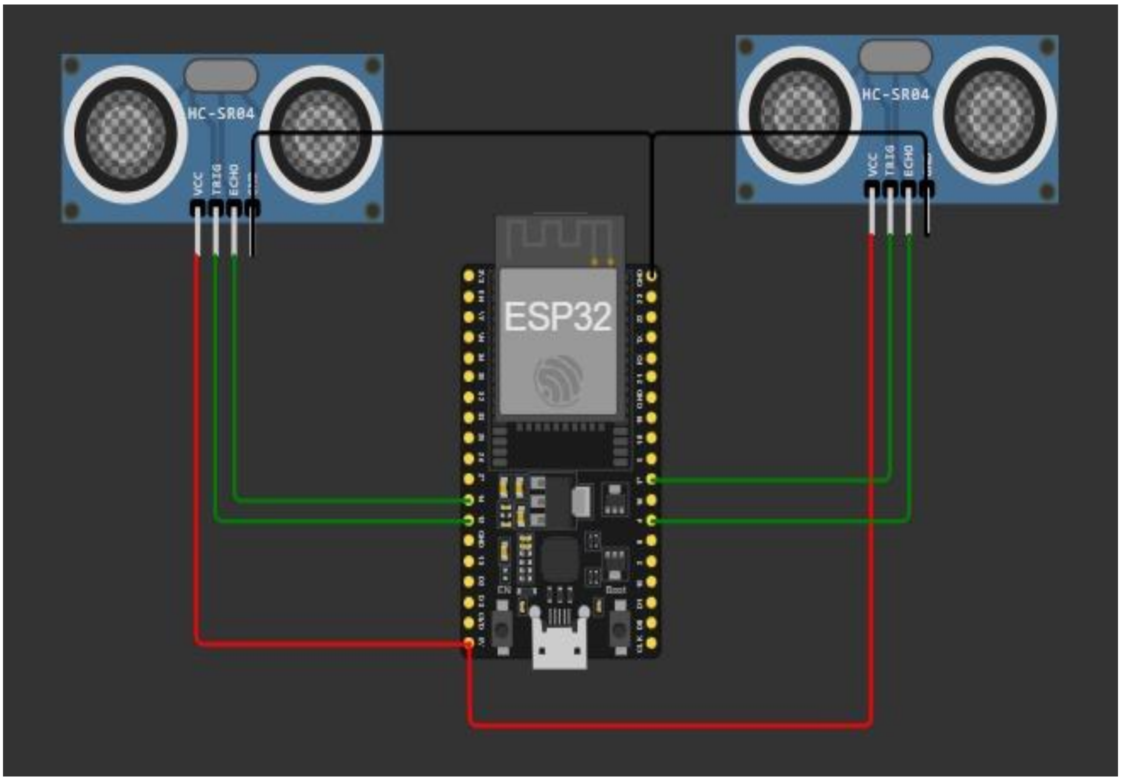

Fig 3.3 Circuit diagram

This Fig 3.3 circuit diagram illustrates the integration of two HC-SR04 ultrasonic sensors with an ESP32 microcontroller for a Smart Waste Management System. The sensors are used to measure the distance between the bin's contents and the sensor, determining the fill level. Each sensor is connected to the ESP32 with VCC and GND for power, and Trig and Echo pins for sending and receiving ultrasonic signals, respectively. The ESP32 processes the distance data from the sensors and communicates with a cloud platform via Wi-Fi to send alerts when a bin is nearing full capacity. This setup enables real-time monitoring and efficient waste management.

3.3 CODING

```
#include <Arduino.h>

#include <WiFi.h>              // For ESP32

#include <Firebase_ESP_Client.h>

// Provide the token generation process info.

#include "addons/TokenHelper.h"

// Provide the RTDB payload printing info and other helper functions.

#include "addons/RTDBHelper.h"

// Insert your network credentials

#define WIFI_SSID "ROCKSTAR"

#define WIFI_PASSWORD "1234567890"

// Insert Firebase project API Key

#define API_KEY "AIzaSyCPJkx4VP3JcATqa2RlEm2nMNp8v0Gw4Jc"

// Insert RTDB URL

#define DATABASE_URL "https://smart-bin-afa36-default-rtdb.asia-

southeast1.firebasedatabase.app/"

// Define Firebase Data object

FirebaseData fbdo;

FirebaseAuth auth;

FirebaseConfig config;

unsigned long sendDataPrevMillis = 0;

bool signupOK = false;

// Define pins for the ultrasonic sensors
```

```
#define TRIG_PIN_1 5  // TRIG pin for the first sensor
#define ECHO_PIN_1 18 // ECHO pin for the first sensor
#define TRIG_PIN_2 19 // TRIG pin for the second sensor
#define ECHO_PIN_2 21 // ECHO pin for the second sensor
void setupUltrasonicPins() {
  pinMode(TRIG_PIN_1, OUTPUT);
  pinMode(ECHO_PIN_1, INPUT);
  pinMode(TRIG_PIN_2, OUTPUT);
  pinMode(ECHO_PIN_2, INPUT);
}
float measureDistance(int trigPin, int echoPin) {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  long duration = pulseIn(echoPin, HIGH);
  float distance = (duration * 0.0343) / 2; // Convert to centimeters
  return distance;
}
void setup() {
  Serial.begin(115200);
```

```
  setupUltrasonicPins();
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(300);
  }
  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();
  // Assign the Firebase credentials
  config.api_key = API_KEY;
  config.database_url = DATABASE_URL;
  if (Firebase.signUp(&config, &auth, "", "")) {
    Serial.println("Firebase sign-up OK");
    signupOK = true;
  } else {
    Serial.printf("Firebase sign-up failed: %s\n",
config.signer.signupError.message.c_str());
  }
  config.token_status_callback = tokenStatusCallback;
```

```
  Firebase.begin(&config, &auth);
  Firebase.reconnectWiFi(true);
}
void loop() {
  float distance1 = measureDistance(TRIG_PIN_1, ECHO_PIN_1);
  float distance2 = measureDistance(TRIG_PIN_2, ECHO_PIN_2);
  if (Firebase.ready() && signupOK && (millis() - sendDataPrevMillis > 1000 ||
sendDataPrevMillis == 0)) {
    sendDataPrevMillis = millis();
    // Store Distance1 in Firebase
    if (Firebase.RTDB.setFloat(&fbdo, "Ultrasonic_Sensor/Sensor1", distance1)) {
      Serial.print("Distance from Sensor1: ");
      Serial.println(distance1);
    } else {
      Serial.println("Failed to store Sensor1 data");
      Serial.println("Reason: " + fbdo.errorReason());
    }
    // Store Distance2 in Firebase
    if (Firebase.RTDB.setFloat(&fbdo, "Ultrasonic_Sensor/Sensor2", distance2)) {
      Serial.print("Distance from Sensor2: ");
      Serial.println(distance2);
    } else {
```

```
      Serial.println("Failed to store Sensor2 data");
      Serial.println("Reason: " + fbdo.errorReason());
    }
  }
}
```

3.4 CONCLUSION

This code implements a Smart Waste Management System using an ESP32 microcontroller, ultrasonic sensors, and Firebase Realtime Database. The ESP32 measures the fill levels of two bins using HC-SR04 ultrasonic sensors connected to designated GPIO pins. The measured distances are calculated and sent to Firebase in real-time, enabling remote monitoring of bin status. The system connects to a Wi-Fi network and authenticates with Firebase using API credentials. It continuously collects data and uploads it to the database at regular intervals. This setup ensures efficient waste management by providing accurate bin-level data for timely waste collection and overflow prevention.

CHAPTER – 4

HARDWARE COMPONENTS

4.1 ESP32 MICROCONTROLLER

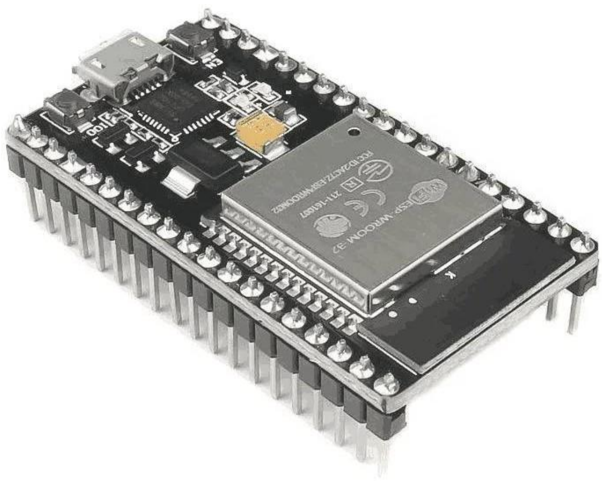

Fig 4.1.1 ESP32 MICROCONTROLLER

The ESP32 is a dual-core, low-power system-on-chip (SoC) microcontroller that combines powerful processing capabilities with multiple communication options. It includes built-in Wi-Fi and Bluetooth 4.2/Bluetooth LE, making it well-suited for Internet of Things (IoT) applications. Which is shown in the Fig 4.1.1 the chip's architecture allows it to handle complex tasks, such as data processing and multi-protocol communications, in real-time. The ESP32 is often selected for projects that require a balance of performance, power efficiency, and connectivity.

Specifications

- Wi-Fi and Bluetooth: 2.4 GHz 802.11 b/g/n Wi-Fi and Bluetooth **4.2** (BLE and Classic), supporting WPA/WPA2 with an on-board antenna.

- GPIO Pins: 34 GPIO pins, of which many support PWM, I2C, SPI, and UART protocols, making them versatile for various sensor and peripheral connections.

- Processor: Dual-core Xtensa LX6 CPU, up to 240 MHz, providing 600 DMIPS for intensive processing tasks.

➢ Memory: 520 KB SRAM and typically 4 MB Flash memory for program and data storage.

➢ It has 17 GPIO, 11 are usable(6 are used for communication with flash)

ESP32 PIN DIAGRAM

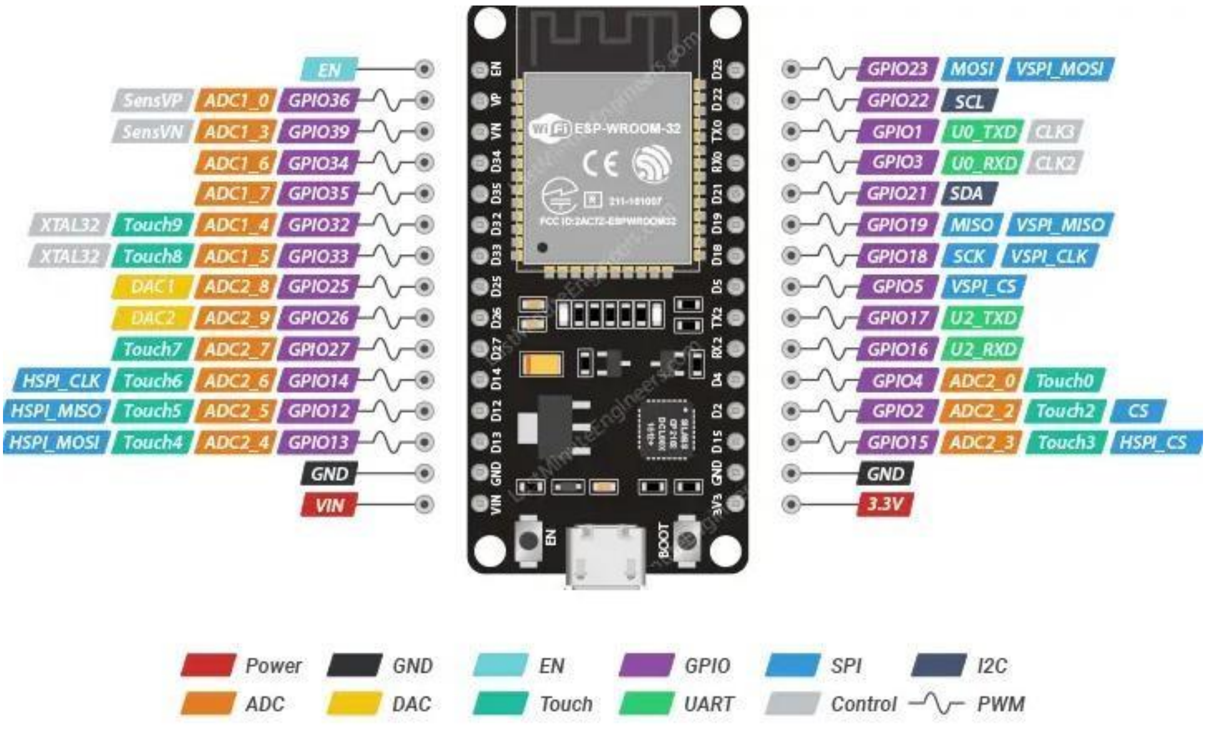

Fig 4.1.2 ESP32 Pin Diagram

This Fig 4.1.2 pin diagram represents an IoT-based Face Recognition Door Control System. Which is shown in the figure 4.1.2 The ESP32 microcontroller receives face recognition results from Firebase Cloud and controls a relay module to operate the door lock. The relay is powered by the ESP32 and is triggered by a GPIO pin (e.g., GPIO16) to unlock or lock the door. The face recognition is handled externally by a camera module connected to a Raspberry Pi or laptop, which sends the results to Firebase. When a recognized face is detected, the ESP32 activates the relay to unlock the door, providing a secure and automated access control system.

4.2 ULTRA SONIC SENSOR

Fig 4.2 Ultrasonic Sensor

The Fig 4.2 image shows a 5V single-channel relay module, commonly used in electronics projects to control high-power devices such as lights, fans, or motors through low-power microcontrollers like Arduino or ESP32. The module has three input pins: VCC (power supply), GND (ground), and IN (signal input), which triggers the relay. The relay can handle loads up to 10A at 250VAC or 30VDC. It features indicator LEDs for power (red) and relay status (green), making it easy to monitor operation. This module acts as an interface to safely control high-voltage devices using low-voltage signals.

Specifications

- Control Voltage:
  Operating Voltage**:** 5V DC (typical)
- Switching Voltage:
  - The sensor itself does not directly switch high voltages. Instead, it outputs a digital signal (5V TTL logic) on the Echo pin based on the received ultrasonic signal.
  - Compatible switching voltage: 0V (LOW) and 5V (HIGH) for microcontroller inputs.
- Switching Current:
  - Quiescent Current (Idle Current): < 2 mA
  - Operating Current (while measuring): 10–15 mA

4.3 CLOUD PLATFORM

In this project, the cloud platform refers to Firebase, which is used to store and manage the facial recognition data and facilitate communication between the camera module and the ESP32 microcontroller. Firebase is a cloud-based platform that provides real-time database services, authentication, and other backend functionalities, making it ideal for IoT-based applications like this one. When the camera module captures an image and processes the facial recognition, the results are uploaded to Firebase.

4.4 FIRE BASE CLOUD

Firebase Cloud serves as the backend infrastructure, enabling real-time data storage and communication between the system's components. The Real-Time Database stores facial recognition results, such as whether a detected face matches an authorized user, and provides instant updates to the ESP32 microcontroller, which uses this data to control the door lock via the relay module. Firebase also handles authentication to ensure secure access to the stored data, allowing only authorized devices to interact with the system. With its real-time capabilities, Firebase ensures seamless synchronization between the camera module, the Raspberry Pi or laptop processing the face recognition, and the ESP32, enabling immediate action based on the recognition results. This cloud platform streamlines the system's operation, reducing the need for local storage and processing while ensuring efficient, scalable, and secure door access control.

4.5 CONCLUSION

The ESP32 microcontroller and relay module play crucial roles in the Face Recognition-Based Door Access Control System. The ESP32 acts as the system's brain, managing communication with the cloud and processing the facial recognition data to determine whether access should be granted. It effectively controls the relay module, which serves as the electronic switch to unlock or lock the door based on the recognition results. The combination of the ESP32's connectivity and processing power with the relay’s ability to control the door's locking mechanism ensures that the system operates securely and efficiently, offering both automation and manual control over door access.

CHAPTER – 5

EXPERIMENT RESULT AND ANALYSIS

5.1 INTRODUCTION

The experimental analysis of the Face Recognition-Based Door Access Control System focuses on evaluating the system's performance, reliability, and effectiveness in real-world scenarios. This analysis aims to assess the accuracy of the facial recognition process, the responsiveness of the ESP32 in retrieving data from Firebase, and the reliability of the relay module in controlling the door lock. Key parameters such as recognition time, recognition accuracy, system latency, and the success rate of the door's operation based on facial authentication will be measured. Additionally, the robustness of the system under varying environmental conditions, such as lighting and distance from the camera, will be tested. The results from these experiments will provide insights into the system's efficiency, help identify areas for improvement, and ensure that the system meets the required security standards for practical deployment.

5.2 INITIAL SETUP

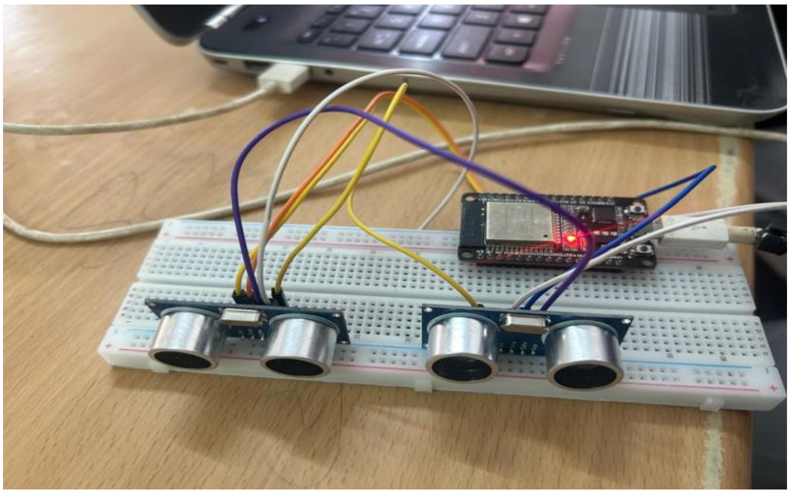

Fig 5.2 Initial Setup

This Fig 5.2 circuit setup consists of an ESP32 microcontroller connected to multiple ultrasonic sensors mounted on a breadboard. The ultrasonic sensors measure distances by emitting sound waves and detecting their reflections, which can be used to monitor the fill levels of waste bins. The ESP32 processes these distance readings and can transmit the data wirelessly to a cloud platform or application for real-time monitoring. This setup is powered via a USB cable connected to a laptop, allowing for prototyping and testing of an IoT-based monitoring system.

5.3 FINAL SETUP

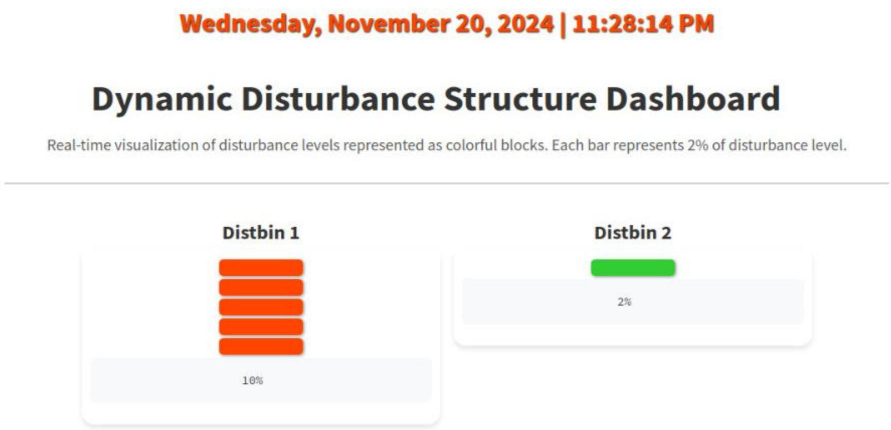

Fig 5.3 Final Output

This Fig 5.3 is a real-time dashboard named "Dynamic Disturbance Structure Dashboard" that visually represents disturbance levels in two bins, "Dustbin 1" and "Dustbin 2." The dashboard uses colorful blocks, with each block corresponding to 2% of the disturbance level. Distbin 1 is currently at 10%, represented by five orange blocks, while Distbin 2 shows 2%, represented by one green block. The dashboard updates in real-time and displays the current date and time, offering a clear and interactive way to monitor data effectively.

CHAPTER – 6

CONCLUSION

6.1 CONCLUSION

The Smart Waste Management System utilizes IoT technology to optimize traditional waste disposal by automating garbage bin monitoring with ultrasonic sensors and an ESP32 microcontroller. When bins reach a set threshold, real-time notifications are sent via a cloud platform to ensure timely collection, preventing overflow. Equipped with GPS for location tracking and a user-friendly interface for data visualization and route optimization, the system enhances efficiency, reduces costs, and minimizes environmental impact. With features like wireless connectivity and real-time monitoring, it promotes sustainability and cleaner surroundings. Future advancements could include AI-driven waste analysis and renewable energy integration

6.2 FUTURE WORKS

The Future work for the Smart Waste Management System could focus on integrating AI to enable predictive waste analysis, allowing for the anticipation of waste generation patterns and optimization of collection schedules. Additionally, incorporating renewable energy sources, such as solar panels, could make the system more sustainable by reducing its carbon footprint. Enhancements in data analytics could provide deeper insights into waste management trends, while expanding the system's scalability to accommodate larger urban areas. Further, advancements in sensor technology and IoT protocols could improve reliability and energy efficiency, ensuring a more robust and eco-friendly solution.

CHAPTER – 7

REFERENCES

1. Sharma, R., & Gupta, P. (2021). "IoT-Based Waste Management System for Smart Cities." International Journal of Advanced Research in Computer Science and Applications, 12(3), 215-221.
2. Singh, A., & Kumar, M. (2020). "Real-Time Monitoring and Notification System for Garbage Collection Using IoT." Journal of Environmental Engineering and Technology, 9(2), 98-104.
3. Zhang, Y., & Li, H. (2022). "Optimizing Waste Management Logistics with IoT and GPS Integration." IEEE Transactions on Smart Cities, 13(4), 1457-1465.
4. Patel, V., & Mehta, D. (2019). "Smart Bin System Using Ultrasonic Sensors and ESP32 for Sustainable Urban Waste Management." International Journal of IoT Applications and Research, 8(5), 321-327.
5. Ahmed, S., & Khan, T. (2020). "Cloud-Enabled Waste Management: A Step Towards Smart Cities." Journal of Environmental Science and IoT, 7(3), 178-183.
6. Verma, N., & Choudhary, S. (2021). "IoT-Driven Waste Management for Cleaner Cities: Real-Time Data and Analysis." International Journal of Smart Systems Research, 10(2), 92-101