***SMART DUSTBIN***

### The domain of the Project

Embedded Systems and IoT

### Mentor MEHAK MAJEED

(Junior Engineer , ATFAAL Innovations)

### By

#### Ms. K. Nihar Sampath

(B.Tech, 4th year pursuing)

**Period of the project July 2025 to August 2025**

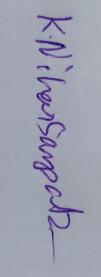
****

## SURE ProEd

**Declaration**

The project titled “**SMART DUSTBIN**” has been mentored by **MEHAK MAJEED**, organised by SURE Trust, from March 2025 to August 2025, for the benefit of the educated unemployed rural youth for gaining hands-on experience in working on industry relevant projects that would take them closer to the prospective employer. I declare that to the best of my knowledge the members of the team mentioned below, have worked on it successfully and enhanced their practical knowledge in the domain.

#### Name Signature

Mr. K. Nihar Sampath

**Signature**

#### Mentor

MEHAK MAJEED

(Junior Engineer—ATFAAL innovations)

#### Seal & Signature

Prof. Radhakumari Executive Director & Founder

SURE Trust

## 

## Table of contents

1. Executive summary
2. Introduction
3. Project Objectives
4. Methodology & Results
5. Social / Industry relevance of the project
6. Learning & Reflection
7. Future Scope & Conclusio

# Executive Summary

The **Smart Dustbin** project is an **IoT-based waste management system** designed to automate the segregation of dry and wet waste while also monitoring the fill levels of bins. The system addresses one of the growing challenges in urban environments — efficient and hygienic waste handling.

The process begins when the user activates the system through a touch sensor. A soil moisture sensor then analyzes the waste for 2 seconds to determine whether it is dry or wet. Based on the reading, a servo motor mechanically tilts a platform to deposit the waste into the appropriate bin. After each disposal, ultrasonic sensors measure the fill levels of the bins. If either bin approaches capacity, the ESP32 microcontroller updates the status on a local HTTP server, which is accessible via a mobile application built using MIT App Inventor. This enables users to check the bin status anytime within the same Wi-Fi network.

**Key Highlights**

Automated Waste Segregation – Reduces human error and improves efficiency.

Event-Based Monitoring – Ultrasonic sensors check fill levels only after waste is added, optimizing energy use.

IoT-Enabled Access – Real-time bin status can be viewed on a mobile app.

Scalable and Affordable – Uses low-cost sensors and ESP32, making it suitable for homes, offices, and public spaces.

**Contribution & Usefulness**

This project demonstrates how embedded systems and IoT technology can be applied to real-world waste management problems. By combining automation and smart monitoring, the Smart Dustbin reduces the need for manual checking, prevents overflow situations, and contributes to cleaner surroundings.

**Its usefulness lies in:**

Supporting sanitation staff by minimizing manual segregation and overflow checks.

Encouraging efficient and hygienic disposal of waste in homes, institutions, and public places.

Serving as a step toward smart cities, where connected systems make everyday life cleaner and more efficient.

Providing a scalable base for future integration with cloud servers, enabling truly remote monitoring and wider deployment in communities.

# Introduction

**1.Background and Context of the Project**

Waste management has become a critical challenge in modern society, particularly in urban environments where the volume of waste is increasing rapidly. Unsegregated waste leads to health hazards, environmental pollution, and inefficient recycling processes. Manual segregation is labor-intensive, unhygienic, and often unreliable.

Technology provides an opportunity to address these challenges through automation and

IoT-based monitoring. Smart systems can not only separate waste but also provide real-

time information on bin usage, thereby streamlining the process of collection and recycling.

The Smart Dustbin project is designed as a prototype to demonstrate how automation,

sensors, and connectivity can solve this pressing problem.

**2.Problem Statement / Goals of the Project**

**Traditional dustbins:**

* Do not separate dry and wet waste automatically.
* Provide no information on fill status, leading to overflow and unhygienic conditions.
* Require frequent manual checking.
* Cannot recycle waste effitiently

Thus, the need arises for a self-segregating and self-monitoring dustbin that minimizes manual involvement while maximizing efficiency.

**Goals of the Project**

* To design and implement a system capable of automatically classifying waste as dry or wet.
* To develop a mechanism that disposes waste into the correct bin using servo actuation.
* To integrate ultrasonic sensors for monitoring bin fill levels.
* To provide a Wi-Fi based IoT interface for status updates.
* To allow users to monitor bins easily through an MIT App Inventor mobile app.

**3.Scope and Limitations**

**Scope:**

* Prototype dustbin suitable for educational and demonstration purposes.
* Supports segregation into two categories: dry and wet waste.
* Provides real-time notifications via a local Wi-Fi network.
* Can be expanded to integrate cloud services for remote monitoring.

**Limitations:**

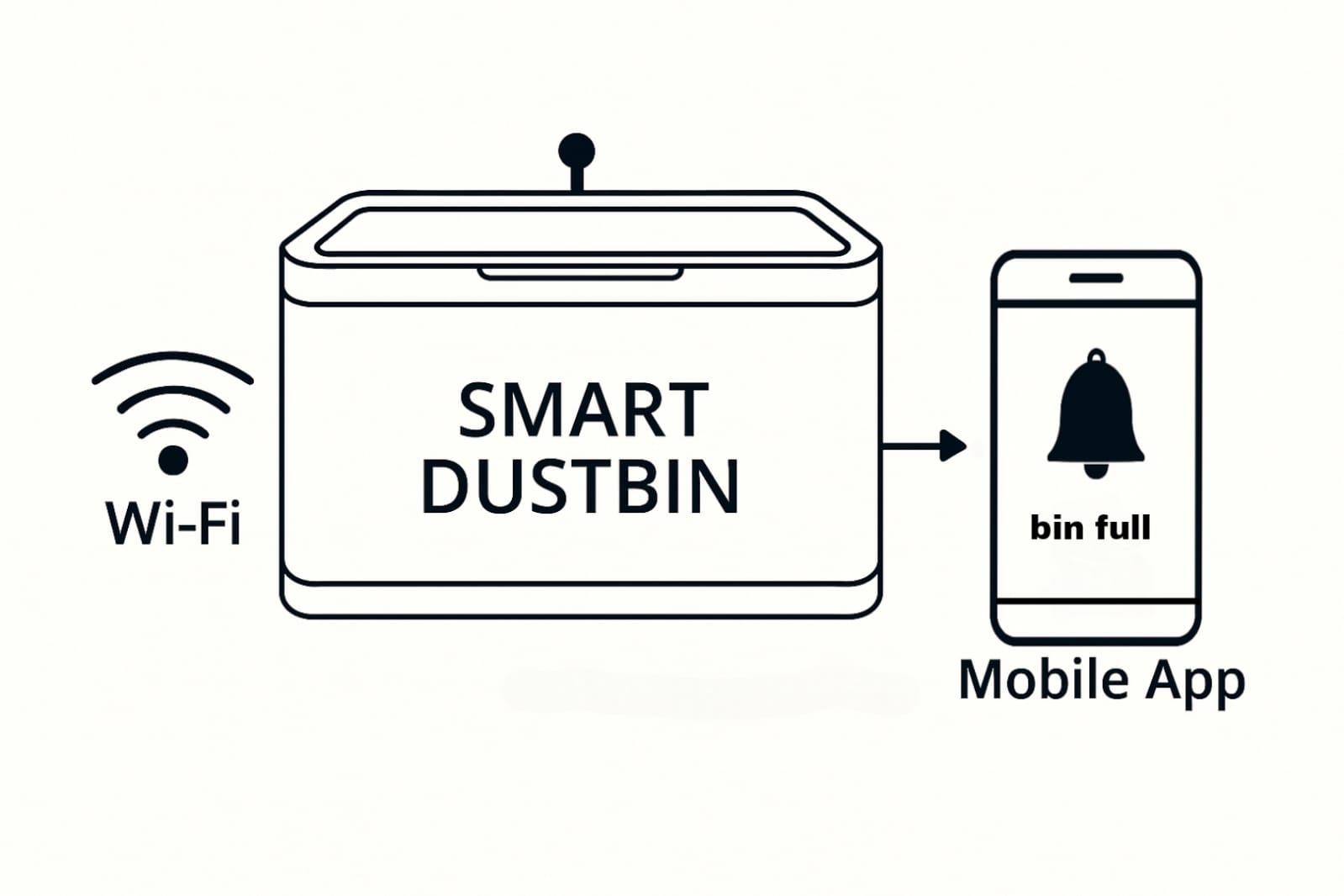
* Only two types of waste segregation supported (does not cover plastics, metals, or glass).
* Requires Wi-Fi connectivity (cannot send data via GSM/SMS in current implementation).
* ESP32 must be on the same Wi-Fi as the mobile app (not remote yet).
* Servo motor mechanism is suitable for light to medium waste but may not handle heavy or bulky materials.

**4.Innovation Component**

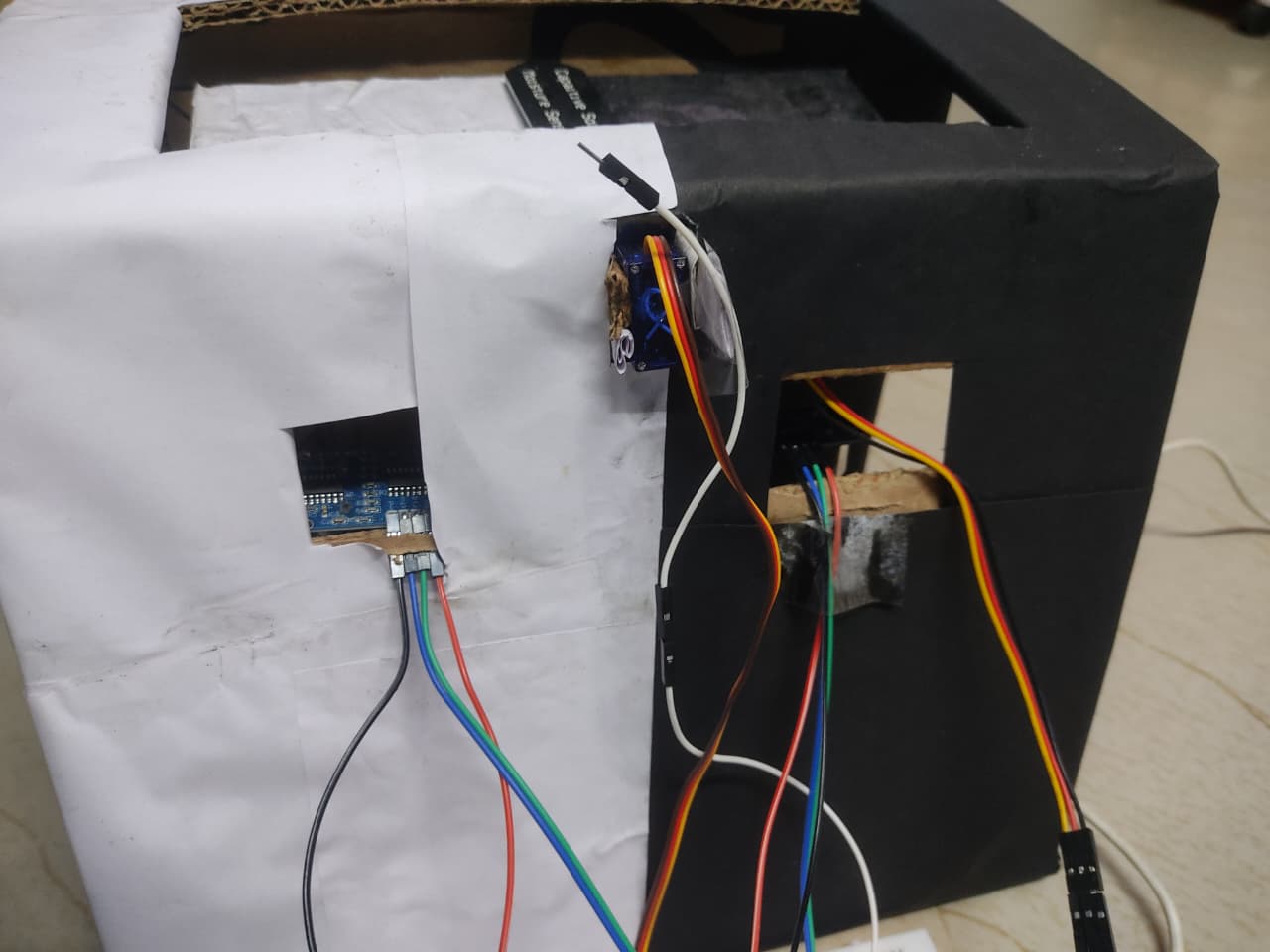
* Touch-based activation to save energy and reduce sensor wear by activating sensors only when needed.
* Mechanical tilting mechanism with servo motor for accurate segregation.
* Real-time IoT integration using ESP32 HTTP server.
* MIT App Inventor interface, providing an intuitive mobile app solution without requiring complex coding.
* The design saves power and reduces false readings by using event-driven monitoring (ultrasonic sensors only check after waste is added, not continuously).

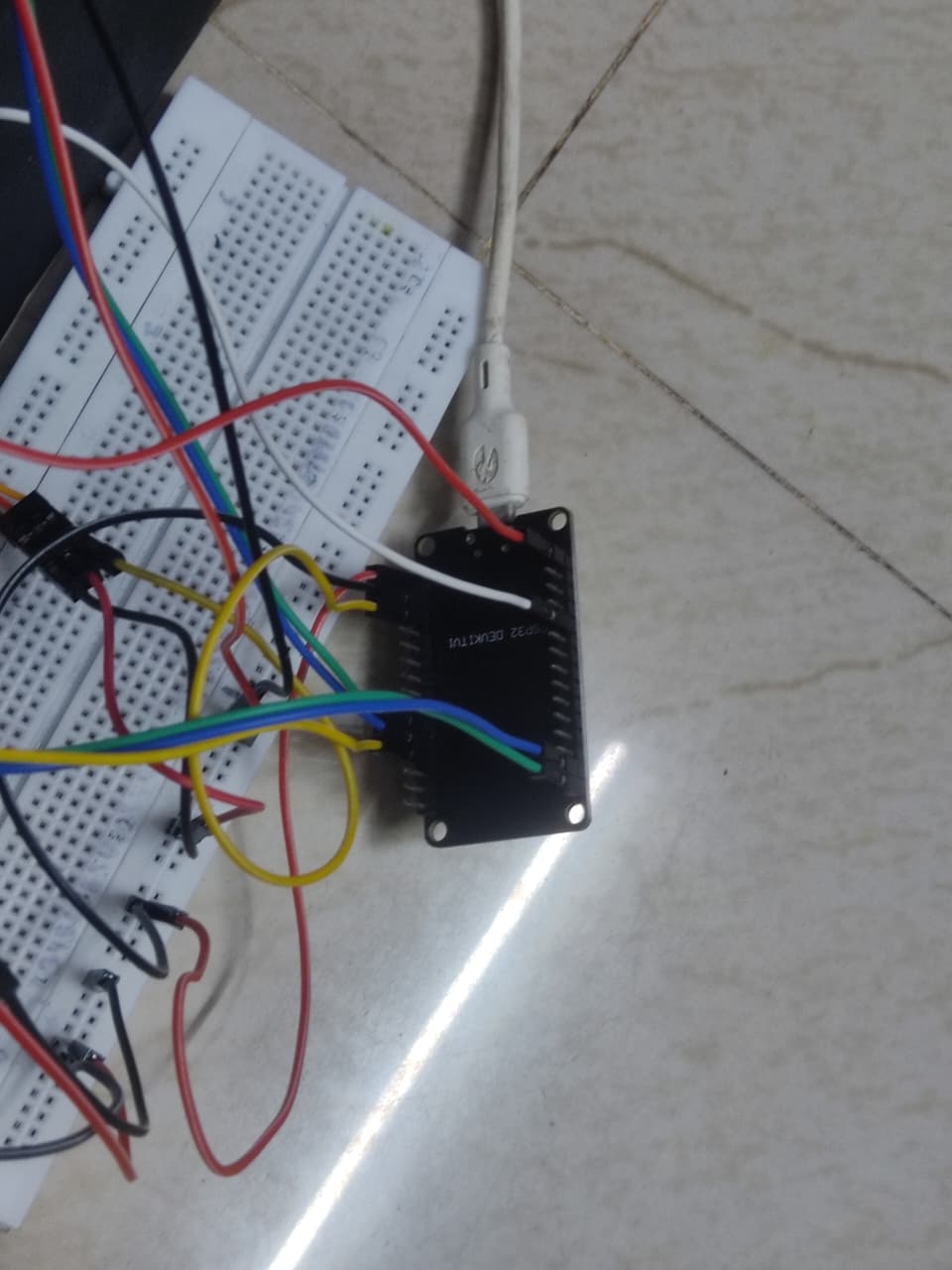
# System Diagram

1. **Mechanical Representation of Smart Dustbin**

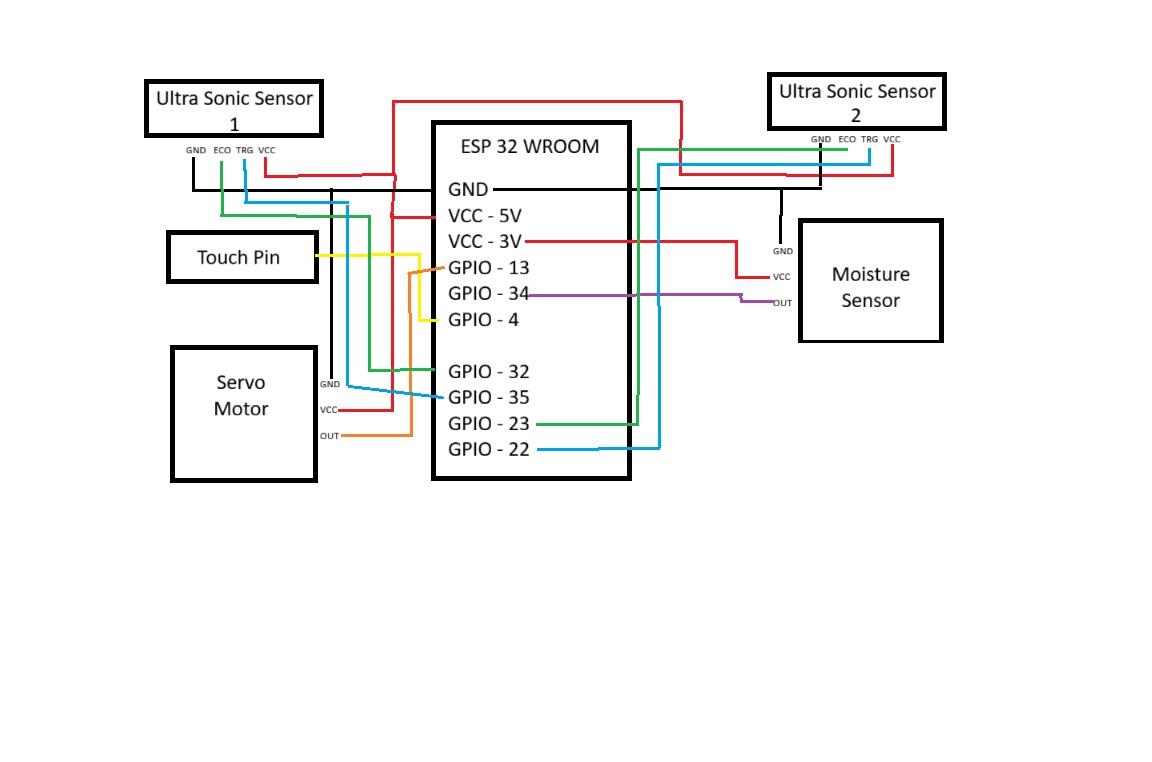
****

1. **Electrical Representation of Smart Dustbin**

****

****

1. **Electronics Representation of Smart Dustbin**

****

# Project Objectives

#### **Project Objectives:**

1. **Automated Waste Segregation**

* Detect waste type using the soil moisture sensor and classify it as dry or wet.
* Direct the waste into the correct bin (dry or wet) using a servo motor–controlled tilting plate.
* Improve hygiene by minimizing human contact with waste materials.

1. **Monitoring and Control**

* Use ultrasonic sensors to measure fill levels in bins.
* Prevent bin overflow by alerting users in advance.

1. **IoT Integration**

* Host an HTTP web server on ESP32 with a /status endpoint.
* Send bin status in structured format (e.g., JSON).

1. **User Interface**

* Provide a mobile application using MIT App Inventor.
* Allow users to check bin status by pressing the “Check Status” button.
* Enable notifications via MIT App Inventor when a bin is full, within a local Wi-Fi.

**Expected Outcomes**

* Working smart dustbin prototype with automated dry/wet segregation.
* Monitor bins at appropriate times (after each waste addition) to ensure reliable readings.
* A simple mobile app that displays real-time bin status.
* Scalable system design that can be extended to multi-bin setups.

# Methodology and Results

**Hardware Setup**

1. **ESP32 Microcontroller**

Acts as the central controller for the smart dustbin.

Handles moisture sensing, servo motor control, ultrasonic distance measurement, touch input, and Wi-Fi connectivity.

1. **Capacitive Touch Sensor (GPIO4)**

Serves as the user input trigger to initiate waste classification.

Detects touch to start moisture sampling and bin status evaluation.

1. **Soil Moisture Sensor (ADC1 Channel 6, GPIO34)**

Measures moisture content of the waste.

Helps classify waste as WET or DRY based on threshold value.

1. **Servo Motor (GPIO13)**

Rotates to direct waste into the appropriate bin (wet or dry).

Returns to neutral position after classification.

1. **Ultrasonic Sensors**

Left Bin (GPIO23 & GPIO22): Measures fill level of WET waste bin.

Right Bin (GPIO32 & GPIO35): Measures fill level of DRY waste bin.

Determines whether bins are full or available.

1. **Wi-Fi Module (Inbuilt in ESP32)**

Connects the system to the internet.

Hosts a local web server to display bin status via /status endpoint.

1. **Power Supply**

5V regulated supply via USB or battery backup for field deployment.

Ensures uninterrupted operation of sensors and servo.

##### 

##### Software and Tools

1. **ESP-IDF Framework**

* Used for coding and compiling the ESP32 firmware.
* Offers FreeRTOS-based multitasking and peripheral control.

1. **Required Drivers and Libraries**

* freertos/FreeRTOS.h → Enables task scheduling and concurrency.
* driver/adc.h → Interfaces with the moisture sensor.
* driver/gpio.h → Controls ultrasonic and servo pins.
* driver/ledc.h → Generates PWM signals for servo motor.
* driver/touch\_pad.h → Reads capacitive touch input.
* esp\_wifi.h, esp\_netif.h, esp\_event.h → Manages Wi-Fi connectivity.
* esp\_http\_server.h → Hosts local web server for bin status.
* esp\_timer.h → Provides microsecond-level timing for ultrasonic sensors.

1. **Web Server**

* Endpoint: /status
* Displays latest bin status: wet\_full, dry\_full, both\_full, not\_full, or sensor\_error.

1. **Logging and Debugging**

* esp\_log.h → Logs system events and sensor readings for debugging.
* Serial monitor used to observe real-time classification and bin status.

1. **Online Integration (Optional Future Expansion)**

* Telegram Bot API or MQTT → Can be added to send bin status alerts to municipal workers or caretakers.
* Cloud dashboard → For remote monitoring and analytics of waste patterns.

#### Methodology:

**1. System Initialization**

* ESP32 initializes GPIOs, ADC, PWM (LEDC), touch pad, and ultrasonic sensors.
* Wi-Fi connects using predefined SSID and password.
* Web server starts and exposes /status endpoint for bin status monitoring.

**2. Waste Classification Trigger**

* User touches the capacitive sensor (GPIO4) to initiate classification.
* Moisture sensor (ADC1 Channel 6) samples for 2 seconds to determine waste type.
* Based on moisture threshold, waste is classified as **WET** or **DRY**.

**3. Servo Motor Control**

* Servo rotates to direct waste into the correct bin:
  + **10°** for WET waste
  + **180°** for DRY waste
* After 1.5 seconds, servo resets to neutral position (≈110°).

**4. Bin Fill Level Detection**

* Ultrasonic sensors measure distance to waste in each bin:
  + **Left Bin (GPIO23 & GPIO22)** → WET bin
  + **Right Bin (GPIO32 & GPIO35)** → DRY bin
* If distance < threshold, bin is considered full.

**5. Status Reporting**

* Based on sensor readings, ESP32 updates latest\_status as:
  + wet\_full, dry\_full, both\_full, not\_full, or sensor\_error
* Status is accessible via local web server (/status endpoint).

**6. Continuous Monitoring**

FreeRTOS task (dustbin\_task) runs in loop:

* Monitors touch pad input
* Samples moisture and ultrasonic readings
* Updates bin status and logs events
* Debounces touch input for reliable detection

#### Project Architecture

The Smart Dustbin System is built around the ESP32 microcontroller, integrating sensors, actuators, and network capabilities to automate waste classification and bin monitoring. The architecture is divided into four layers:

**1. Input Layer**

* **Touch Pad (GPIO4)**: Triggers classification process.
* **Moisture Sensor (GPIO34)**: Determines waste type.
* **Ultrasonic Sensors**:
* Left (WET bin): GPIO23 (TRIG), GPIO22 (ECHO)
* Right (DRY bin): GPIO32 (TRIG), GPIO35 (ECHO)

**2. Processing Layer**

**ESP32 Microcontroller**:

* Executes classification logic
* Controls servo motor via PWM
* Reads sensor data
* Hosts web server
* Manages Wi-Fi connection

**3. Output Layer**

* **Servo Motor (GPIO13)**: Directs waste to appropriate bin.
* **Web Server**: Displays bin status via /status endpoint.
* **Logging (ESP\_LOG)**: Provides real-time feedback for debugging and monitoring.

**4. Communication & IoT Layer**

* **Wi-Fi (ESP32 inbuilt)**: Enables network connectivity.
* **HTTP Server**: Allows remote access to bin status.
* (Optional): Can be extended to send alerts via Telegram/MQTT or integrate with IoT dashboards.

**Results.**

1. **Accurate Waste Classification**

* Moisture sensor reliably differentiated between wet and dry waste.
* Servo motor rotated correctly based on classification.

1. **Bin Status Detection**

* Ultrasonic sensors accurately measured bin fill levels.
* System correctly identified full, partially full, and error states.

1. **Touch-Based Activation**

* Capacitive touch pad provided intuitive user interaction.
* Debounce logic ensured stable operation.

**4. Web Interface Functionality**

* ESP32 successfully hosted a web server.
* Bin status was accessible via /status endpoint in browser.

**5. Real-Time Operation**

* FreeRTOS task loop maintained continuous monitoring.
* System responded promptly to user input and sensor changes.

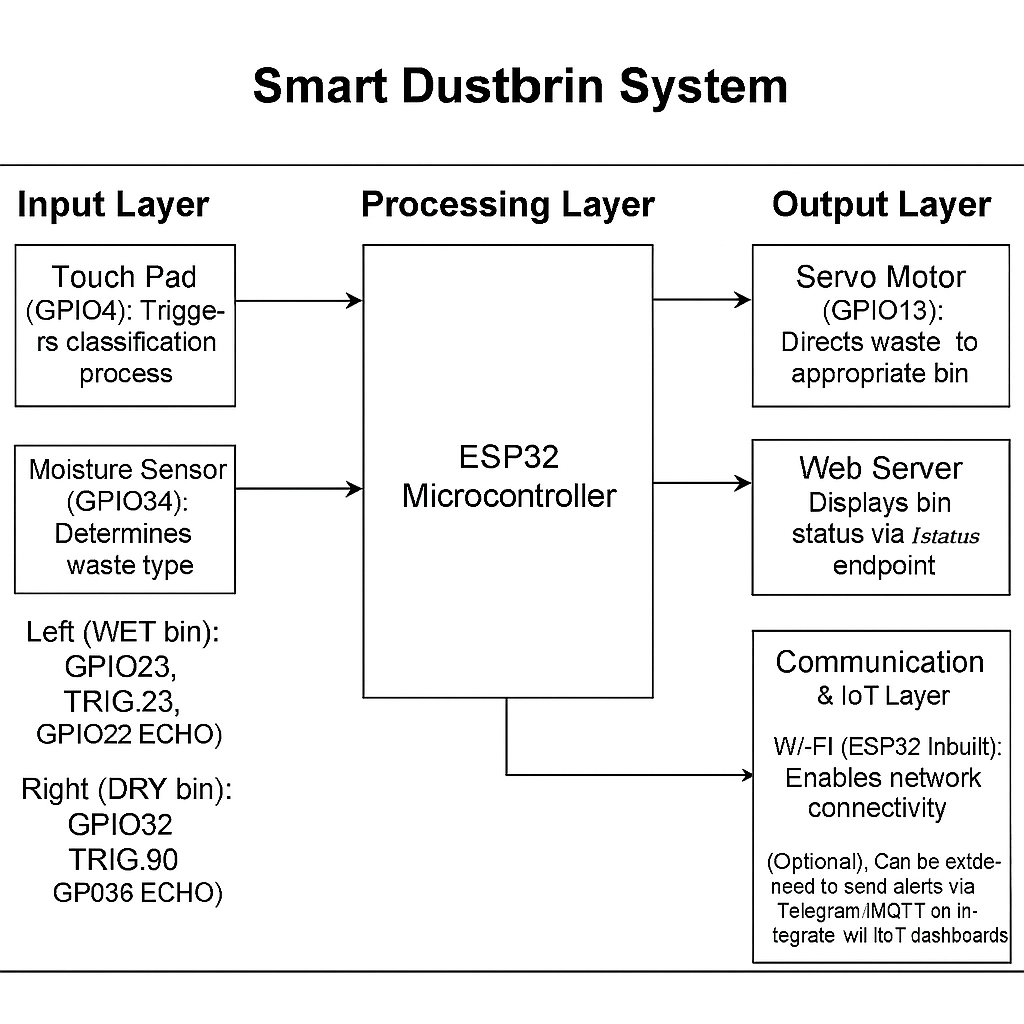
**6. IoT Readiness**

* Wi-Fi connectivity enabled remote access.
* Architecture supports future integration with Telegram, SMS, or dashboards.

**7. System Reliability**

* Tested across multiple waste samples and bin conditions.
* Classification, servo control, and reporting worked consistently.

**BLOCK DIAGRAM:**

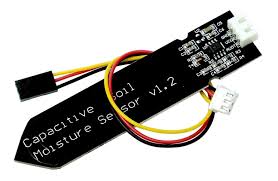
****

## Pictures

**FIGURE 1: ESP32 MICRO CONTROLLER**

**FIGURE 2: SERVO MOTOR**

**FIGURE 3: MOISTURE SENSOR**

****

**FIGURE 4: ULTRA SONIC SENSOR**

****

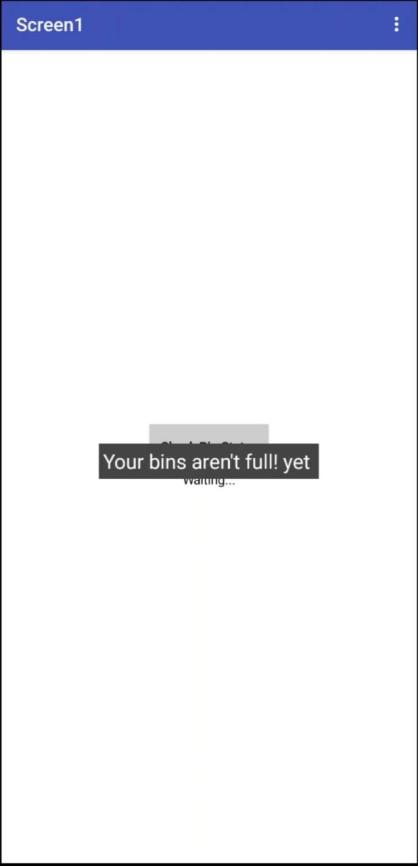
### 

**FIGURE 5: JUMPER WIRE (TOUCH PIN)**

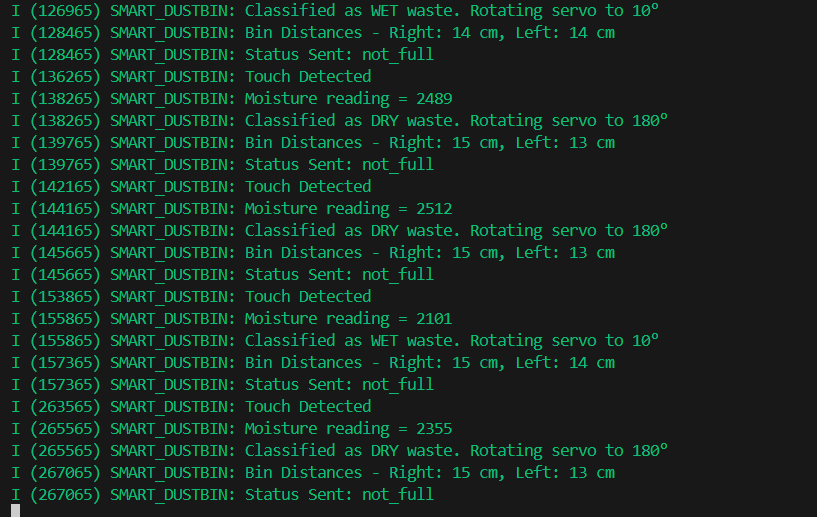
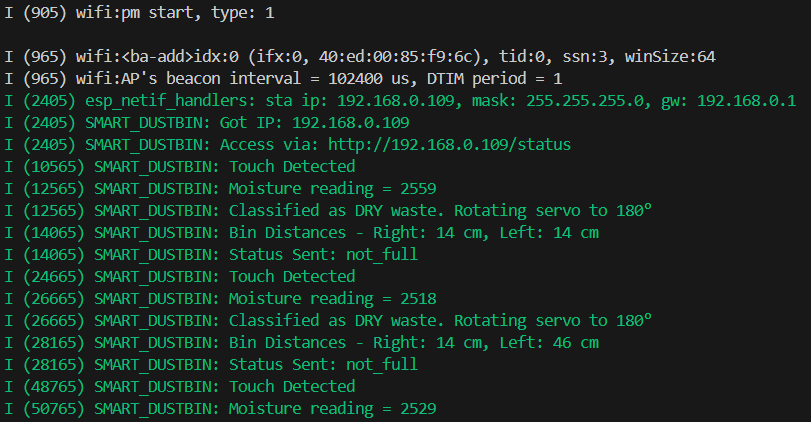
****

**RESULT**

**FIGURE 6 : MOBILE OUTPUT VIEW**

****

**FIGURE 7: SERIAL MONITOR**

****

# Social/Industry Relevance

**Social Relevance**

The **Smart Dustbin System** addresses a pressing civic and environmental challenge: improper waste segregation and overflowing bins in public and residential areas. In many communities, especially urban slums and rural regions, mixed waste disposal leads to unhygienic conditions, inefficient recycling, and increased burden on sanitation workers. Manual segregation is labor-intensive and error-prone, while overflowing bins pose health risks and degrade public spaces.

This system empowers users to participate in responsible waste disposal through a touch-based interface and automated classification. By using moisture sensing to distinguish between wet and dry waste, and directing it to the correct bin via servo control, the system promotes cleaner environments and better recycling practices. Additionally, real-time bin status monitoring via a web interface allows caretakers or municipal staff to intervene before bins overflow.

From a social perspective, the project encourages civic responsibility, supports cleaner neighborhoods, and reduces manual labor in waste handling. It is especially valuable in schools, hostels, hospitals, and gated communities where hygiene and sustainability are priorities. The system is intuitive, affordable, and scalable—making it accessible for deployment in both rural and urban settings.

**Industrial Relevance**

From an industry standpoint, this project showcases the practical application of **IoT and embedded systems** in smart city infrastructure. By integrating the **ESP32 microcontroller**, **capacitive touch pad**, **moisture sensor**, **ultrasonic sensors**, **servo motor**, and **Wi-Fi connectivity**, the system demonstrates how low-cost hardware can be orchestrated to solve real-world civic problems.

The use of **FreeRTOS** for task scheduling, **PWM control** for servo actuation, and **HTTP server hosting** for remote status access reflects a robust embedded design. The system’s modular architecture allows for future expansion—such as adding GSM modules for SMS alerts, cloud dashboards for analytics, or AI-based waste classification using image sensors.

Industrially, this solution is relevant for:

* Municipal waste management systems
* Smart campus initiatives
* Industrial parks and tech zones
* Public transport hubs and airports
* CSR-driven sustainability programs

Its ability to monitor bin fill levels and classify waste autonomously reduces operational costs, improves sanitation workflows, and supports data-driven decision-making in waste logistics.

# Learning and Reflection

**1. New Learnings**

* Gained hands-on experience programming the **ESP32** for real-time embedded applications using **FreeRTOS**, multitasking, and GPIO control.
* Learned to interface and calibrate **capacitive touch sensors**, enabling intuitive user interaction for waste classification.
* Developed proficiency in **moisture sensing** using ADC and threshold-based logic to distinguish between wet and dry waste.
* Mastered **servo motor control** via PWM (LEDC), including angle mapping and timed actuation for bin direction.
* Implemented **ultrasonic distance measurement** using GPIO and microsecond timers to detect bin fill levels with precision.
* Built and hosted a **local web server** using ESP-IDF’s HTTP server library to expose real-time bin status via /status endpoint.
* Understood **Wi-Fi initialization and event handling**, enabling seamless network connectivity and remote access.
* Strengthened debugging and logging skills using **ESP\_LOG**, improving visibility into sensor readings, classification logic, and system status.
* Learned to structure embedded tasks for continuous monitoring, debounce handling, and sensor coordination in a FreeRTOS environment.
* Practiced system integration by combining hardware, software, and network layers into a cohesive, responsive civic IoT solution.

**2. Overall Experience**

Working on the **Smart Dustbin System** was a deeply rewarding and technically enriching journey. It provided real-world exposure to embedded systems design, sensor integration, and IoT-based civic automation. I gained practical skills in ESP32 programming, ADC and PWM control, ultrasonic sensing, and web server deployment—all within the ESP-IDF framework.

The project challenged me to think like a systems designer: coordinating multiple components, managing timing and thresholds, and ensuring reliable operation under real-world conditions. It also taught me how to build user-friendly interfaces and remote monitoring tools that can scale to smart city applications.

Beyond technical growth, the experience reinforced my passion for solving social problems through engineering. Designing a system that promotes responsible waste disposal, reduces manual labor, and supports environmental sustainability gave me a sense of purpose and impact. It’s a project I’m proud to showcase—not just for its functionality, but for the values it represents.

# Libraries and Functions Used

#### Libraries Used:

#include "driver/adc.h"

#include "driver/gpio.h"

#include "driver/ledc.h"

#include "driver/touch\_pad.h"

#include "esp\_wifi.h"

#include "esp\_event.h"

#include "esp\_netif.h"

#include "esp\_http\_server.h"

#include "esp\_timer.h"

#include "nvs\_flash.h"

#include "esp\_log.h"

**B. Functions from Your Code**

* void app\_main() – Initializes system and starts main task
* void dustbin\_task(void \*pvParam) – Handles touch input, moisture sensing, servo control, and bin status detection
* void setup\_gpio() – Configures GPIOs and PWM for servo and ultrasonic sensors
* void wifi\_init\_sta() – Initializes and connects ESP32 to Wi-Fi
* void wifi\_event\_handler(...) – Handles Wi-Fi/IP events and starts web server
* httpd\_handle\_t start\_webserver() – Starts HTTP server and registers URI
* esp\_err\_t status\_get\_handler(httpd\_req\_t \*req) – Responds to /status HTTP GET request
* int read\_distance\_cm(gpio\_num\_t trig, gpio\_num\_t echo) – Measures distance using ultrasonic sensor
* void servo\_write\_angle(int angle) – Rotates servo to specified angle using PWM

**C. Functions from Libraries**

* adc1\_config\_width(width) – Sets ADC resolution
* adc1\_config\_channel\_atten(channel, atten) – Sets ADC attenuation
* adc1\_get\_raw(channel) – Reads raw ADC value
* gpio\_set\_direction(pin, mode) – Sets pin as input/output
* gpio\_set\_level(pin, value) – Sets output level (HIGH/LOW)
* gpio\_get\_level(pin) – Reads input level
* ledc\_timer\_config(&config) – Configures PWM timer
* ledc\_channel\_config(&config) – Configures PWM channel
* ledc\_set\_duty(mode, channel, duty) – Sets PWM duty cycle
* ledc\_update\_duty(mode, channel) – Applies duty cycle update
* touch\_pad\_init() – Initializes touch pad module
* touch\_pad\_config(pin, threshold) – Sets touch pad threshold
* touch\_pad\_read(pin, &value) – Reads touch pad value
* esp\_netif\_init() – Initializes network interface
* esp\_event\_loop\_create\_default() – Creates default event loop
* esp\_netif\_create\_default\_wifi\_sta() – Creates default Wi-Fi station
* esp\_wifi\_init(&config) – Initializes Wi-Fi driver
* esp\_wifi\_set\_mode(mode) – Sets Wi-Fi mode
* esp\_wifi\_set\_config(interface, &config) – Applies Wi-Fi configuration
* esp\_wifi\_start() – Starts Wi-Fi
* esp\_wifi\_connect() – Connects to Wi-Fi network
* esp\_event\_handler\_register(base, id, handler, arg) – Registers event handler
* httpd\_start(&server, &config) – Starts HTTP server
* httpd\_register\_uri\_handler(server, &uri) – Registers URI handler
* httpd\_resp\_send(req, message, length) – Sends HTTP response
* esp\_timer\_get\_time() – Gets current time in microseconds
* ets\_delay\_us(time) – Delays execution in microseconds
* xTaskCreate(task, name, stack, param, priority, handle) – Creates FreeRTOS task
* vTaskDelay(ticks) – Delays task execution
* esp\_log\_timestamp() – Gets current timestamp
* ESP\_LOGI(TAG, message, ...) – Logs informational message
* ESP\_LOGW(TAG, message, ...) – Logs warning message

# Conclusion and Future Scope

#### Conclusion

The **Smart Dustbin System** effectively demonstrates the integration of embedded systems, sensor interfacing, and IoT connectivity to address a real-world civic and environmental challenge: waste segregation and bin overflow detection. By combining the **ESP32 microcontroller**, **capacitive touch pad**, **moisture sensor**, **ultrasonic sensors**, **servo motor**, and **Wi-Fi connectivity**, the system enables automated classification of waste into wet and dry categories and monitors bin fill levels in real time.

Through this project, I gained valuable experience in **real-time embedded programming**, **sensor calibration**, **PWM control**, and **web server deployment** using ESP-IDF. I also developed skills in **FreeRTOS task scheduling**, **touch-based user interaction**, and **remote monitoring via HTTP endpoints**. The project emphasized the importance of intuitive design, reliable classification logic, and scalable infrastructure for smart city applications.

Overall, this project not only strengthened my technical foundation in embedded IoT systems but also showcased the potential for creating **automated civic solutions** that promote environmental sustainability, reduce manual labor, and improve public hygiene. It serves as a strong base for future innovations in smart infrastructure and municipal automation.

**Future Scope**

The Smart Dustbin System has significant potential for enhancement and expansion in future developments:

**1. Mobile App Integration**

• Develop a companion app to view bin status, receive alerts, and trigger classification remotely.

**2. Advanced Notifications**

• Integrate SMS, Telegram, or email alerts when bins are full or sensors detect anomalies.

**3. Cloud Data Logging & Analytics**

• Store bin usage data in the cloud to analyze waste patterns and optimize collection schedules.  
• Generate reports for municipal authorities or facility managers.

**4. Multi-Bin & Multi-Zone Support**

• Extend the system to manage multiple bins across different zones with centralized monitoring.

**5. Voice Feedback & Accessibility**

• Add voice alerts or audio prompts to guide users during waste disposal, improving accessibility.

**6. Automated Lid & Disposal Mechanism**

• Integrate servo-controlled lids or conveyors for hands-free disposal and hygiene.

**7. AI-Based Waste Classification**

• Use image sensors and machine learning to classify waste types beyond moisture detection.

**8. Scalable IoT Platform**

• Connect multiple smart dustbins to a unified dashboard for smart campuses, hospitals, or city-wide deployments.