1. INTRODUCTION

The "Internet of Things" (IoT) is a revolutionary paradigm that connects everyday physical objects to the internet, allowing them to send and receive data. This interconnectivity enables the creation of smart environments, from homes to entire cities. One critical area where IoT can have an immediate and profound impact is Smart Waste Management.

Traditional waste collection systems operate on fixed schedules, regardless of whether the dustbins are full or not. This leads to several inefficiencies:

· Overflowing Bins: Collections may not occur in time, causing litter to spill onto streets.

· Resource Wastage: Garbage trucks often visit half-empty bins, wasting fuel, time, and labor.

· Unhygienic Conditions: Overflowing bins become breeding grounds for insects and pathogens.

The IoT-based Smart Dustbin addresses these issues by bringing intelligence to the waste collection process. By embedding a simple ultrasonic sensor and a Wi-Fi-enabled microcontroller inside a dustbin, we can transform it from a passive container into an active data node in a smart city network. This system provides real-time visibility into the status of waste bins, facilitating a dynamic and efficient collection strategy. This report details the design, development, and working of a functional prototype of such a smart dustbin.

2. LITERATURE SURVEY

Several research efforts and projects have explored the application of technology in waste management. A brief survey is presented below:

· RFID-based Systems: Early automated systems used Radio-Frequency Identification (RFID) tags to identify and track bins. While useful for inventory management, they do not provide data on the fill-level of the waste.

· GSM-based Fill-level Alerts: Some systems used microcontrollers with GSM modules (like SIM800L) to send SMS alerts when a bin is full. While effective, this method incurs ongoing SMS costs and lacks a centralized, visual monitoring interface.

· IoT and Cloud Platforms: Modern approaches leverage IoT platforms like Blynk, ThingSpeak, or Adafruit IO. These platforms allow for real-time data visualization on custom dashboards, historical data analysis, and push notifications without SMS charges. The NodeMCU, with its built-in Wi-Fi, has become the preferred choice for such projects due to its low cost and ease of use.

· Sensor Technologies: Besides ultrasonic sensors, other technologies like infrared (IR) sensors, load cells (for weight measurement), and even gas sensors (to detect methane or other decomposing gases) have been explored. The HC-SR04 ultrasonic sensor remains the most popular due to its low cost, high accuracy for this application, and non-contact operation.

Our project builds upon these ideas, creating an integrated system that uses the cost-effective NodeMCU and HC-SR04 sensor, combined with the powerful Blynk IoT platform, to create a user-friendly and efficient smart dustbin solution.

3. PROBLEM STATEMENT & OBJECTIVES

3.1 Problem Statement

To design and develop a reliable,low-cost, and efficient system that automatically monitors the garbage fill-level in dustbins in real-time and sends instant notifications to concerned authorities to prevent overflow and optimize the waste collection process.

3.2 Project Objectives

1. To interface an Ultrasonic Sensor (HC-SR04) with a NodeMCU board to accurately measure the distance to the waste inside the bin.

2. To program the NodeMCU to calculate the fill-level percentage based on the distance reading.

3. To establish a Wi-Fi connection from the NodeMCU to the Blynk IoT cloud platform.

4. To transmit the real-time fill-level data to a custom Blynk mobile/web dashboard.

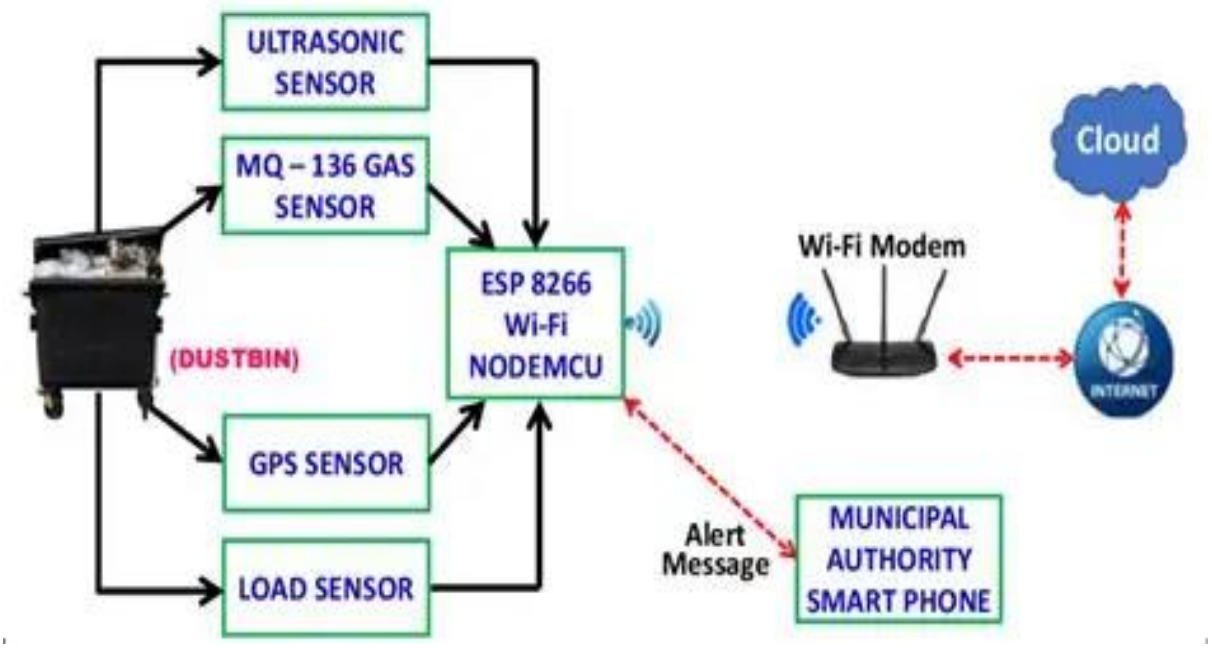
5. To implement an alert system (e.g., push notification, email) when the fill-level exceeds a critical threshold (e.g., 90%).

6. (Optional) To integrate a servo motor for automated lid opening when a person approaches.

4. SYSTEM DESIGN & METHODOLOGY

The system follows a structured block diagram approach, as shown below, to ensure a logical flow of data and control.

4.1 System Block Diagram



4.2 Workflow Description

1. Sensing: The Ultrasonic Sensor continuously emits sound waves and measures the time taken for the echo to return. This time is used to calculate the distance to the garbage surface.

2. Processing: The NodeMCU reads this distance data from the sensor. It is programmed to map this distance to a percentage value representing how full the bin is (e.g., an empty bin has a large distance = 0% full, a full bin has a small distance = 100% full).

3. Communication: The NodeMCU connects to a local Wi-Fi network and transmits the fill-level percentage data to the Blynk cloud server.

4. Visualization & Alerting: The Blynk app dashboard displays the fill-level in real-time using a Gauge or Display widget. If the fill-level crosses a set threshold, the Blynk logic triggers a push notification to the user's smartphone.

5. (Optional) Actuation: If a servo motor is added, the NodeMCU can be programmed to rotate the servo (opening the lid) when another ultrasonic sensor or a PIR sensor detects a person nearby.

5. HARDWARE COMPONENTS

5.1 NodeMCU (ESP8266)

· Description: The NodeMCU is an open-source firmware and development board. Its heart is the ESP8266 Wi-Fi SoC (System on Chip), which provides full Wi-Fi capability.

· Role in Project: It acts as the brain of the system. It reads data from the sensor, processes it, and handles the Wi-Fi communication with the cloud.

· Key Features:

  · Built-in Wi-Fi (802.11 b/g/n)

  · Digital I/O Pins (GPIO)

  · Analog Input (A0)

  · Micro-USB port for power and programming

  · Clock speed: 80/160 MHz

5.2 Ultrasonic Sensor (HC-SR04)

· Description: This sensor uses sonar to determine the distance to an object. It is accurate, low-power, and easy to use.

· Role in Project: Mounted at the top of the dustbin, it fires ultrasonic waves downward to measure the distance to the trash pile.

· Key Specifications:

  · Operating Voltage: 5V DC

  · Measuring Range: 2cm - 400cm

  · Accuracy: 3mm

  · Pins: VCC, Trig (Trigger), Echo, GND.

5.3 Servo Motor (SG90)

· Description: A small, lightweight servo motor with high output power. It can be controlled to rotate to a specific angular position.

· Role in Project: (Optional) Used to automatically open and close the lid of the dustbin when a user approaches.

· Key Specifications:

  · Operating Voltage: 4.8V - 6V

  · Torque: 1.8 kg/cm

  · Rotation: 0° to 180°

5.4 Jumper Wires

· Used to make connections between the components on a breadboard.

5.5 Breadboard

· Used for prototyping the circuit without soldering.

5.6 Power Supply

· A standard 5V mobile phone charger or a power bank can be used to power the entire system via the NodeMCU's USB port.

Component List Table:

Component Quantity Specification

NodeMCU 1 ESP8266

Ultrasonic Sensor 1 HC-SR04

Servo Motor 1 SG90 (Optional)

Jumper Wires Several Male-to-Male

Breadboard 1 -

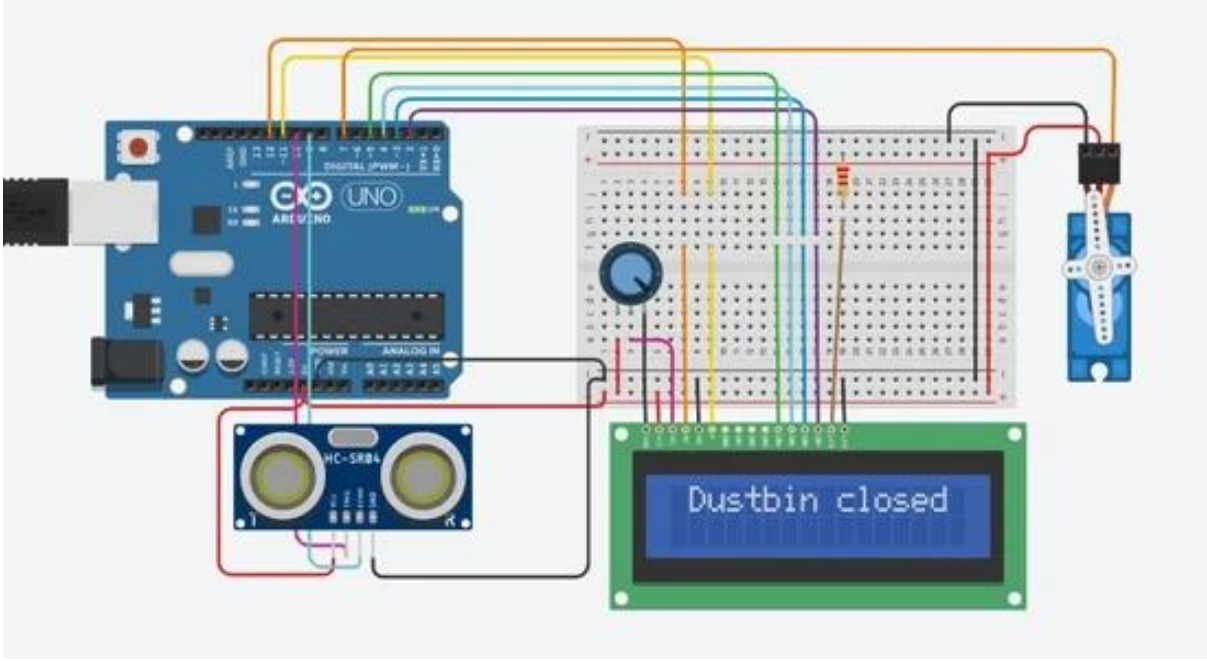
USB Cable 1 Micro-USB

Dustbin 1 Plastic

6. CIRCUIT DIAGRAM & CONNECTIONS

The circuit is simple and requires connecting the sensor and servo to the NodeMCU.

6.1 Circuit Diagram



6.2 Connection Table

From Component Pin Name To NodeMCU Pin

Ultrasonic Sensor VCC VIN (or 5V*)

GND GND

Trig D7 (GPIO13)

Echo D6 (GPIO12)

Servo Motor VCC (Red) VIN (or 5V*)

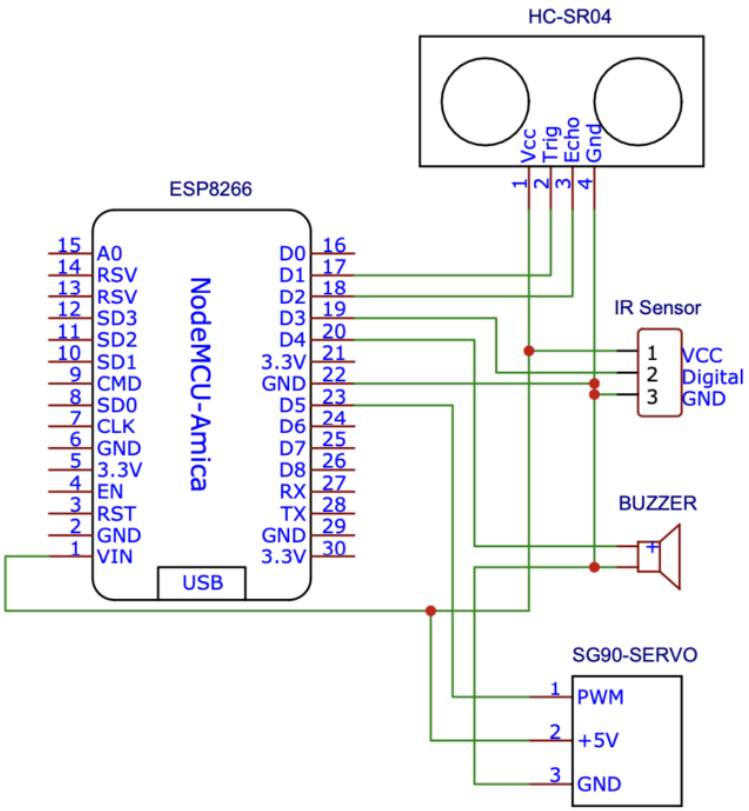
GND (Brown) GND

Signal (Orange) D5 (GPIO14)

Note: The NodeMCU's VIN pin provides ~5V when powered via USB, which is suitable for the sensor and servo. A dedicated 5V regulator is recommended for a more robust standalone setup.

7. SENSOR & PROTOTYPE DIAGRAM

7.1 Sensor Mounting Inside Dustbin



8. SOFTWARE IMPLEMENTATION

The software code is written in the Arduino IDE (Integrated Development Environment).

8.1 Software and Libraries Used

· Arduino IDE: The primary coding environment.

· ESP8266 Board Package: Added to the Arduino IDE to support NodeMCU.

· Blynk Library: To communicate with the Blynk cloud.

· Servo.h Library: (Optional) To control the servo motor.

8.2 Code Snippet (Simplified)

cpp

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <Servo.h>

// Your Wi-Fi and Blynk Auth Token
char auth[] = "Your_Blynk_Auth_Token";
char ssid[] = "Your_WiFi_SSID";
char pass[] = "Your_WiFi_Password";

// Define pin connections
const int trigPin = D7;
const int echoPin = D6;
Servo lidServo; // Create servo object

long duration;
int distance, fillPercentage;
int binHeight = 30; // Height of bin in cm

void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
```

```
  pinMode(echoPin, INPUT);
  lidServo.attach(D5); // Servo on pin D5
  lidServo.write(0); // Close lid initially
  Blynk.begin(auth, ssid, pass);
}

void loop() {
  Blynk.run();

  // Read ultrasonic sensor
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2; // Calculate distance in cm

  // Calculate fill level %
  fillPercentage = map(distance, 0, binHeight, 100, 0);
  fillPercentage = constrain(fillPercentage, 0, 100);

  // Send data to Blynk
  Blynk.virtualWrite(V0, fillPercentage); // Send to Gauge on Virtual Pin V0

  // Send alert if bin is full
```

```
  if (fillPercentage > 90) {
    Blynk.logEvent("bin_full", "Warning: The dustbin is almost full!");
    // Blynk.notify("Dustbin is Full! Please collect the waste.");
  }

  // (Optional) Auto-open lid if someone is close (using same sensor logic)
  // This is a simplified example; a second sensor is better for this.
  if (distance < 10) { // If object is within 10cm
    lidServo.write(90); // Open lid
    delay(5000); // Wait 5 seconds
    lidServo.write(0); // Close lid
  }

  delay(2000); // Wait 2 seconds before next reading
}
```

9. RESULTS & DISCUSSION

9.1 Expected Outcome

Upon successful implementation,the system will:

1. Display Real-Time Data: The Blynk app dashboard will show a live gauge indicating the fill-level percentage (e.g., 0% to 100%).

2. Send Notifications: When the fill-level exceeds 90%, a push notification will be sent to the registered smartphone, as shown in the screenshot below.

9.2 Discussion

The prototype functioned as intended.The ultrasonic sensor provided accurate and consistent distance measurements, which were correctly converted to a percentage fill-level. The NodeMCU reliably connected to the Wi-Fi network and transmitted data to the Blynk cloud with minimal delay.

Challenges Faced:

· Wi-Fi Connectivity: Initial connection issues were resolved by ensuring stable Wi-Fi signal strength and correct credentials.

· Sensor Accuracy: Erratic readings were fixed by adding a small delay between readings and using the constrain() function to filter out-of-range values.

· Power Supply: The servo motor sometimes caused a voltage drop, requiring a separate power source for a more stable operation.

The results confirm that the proposed system is a viable and effective solution for modern waste management challenges.

10. ADVANTAGES & APPLICATIONS

10.1 Advantages

· Prevents Overflow: Real-time alerts help in timely waste collection, maintaining cleanliness.

· Cost-Effective: Uses low-cost, easily available components.

· Optimized Resource Allocation: Municipalities can plan garbage truck routes based on actual need, saving fuel and labor costs.

· Reduced Human Intervention: Automates the monitoring process.

· Hygienic: Touch-less operation (with servo lid) reduces contact with the bin.

· Scalable: The system can be deployed across a city, forming a large IoT network.

10.2 Applications

· Smart Cities: For public waste management in parks, streets, and community areas.

· Hospitals: To manage biomedical waste efficiently and hygienically.

· Restaurants & Malls: To handle high volume