IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 2, February 2025

DOI: 10.17148/IJARCCE.2025.14217

IOT- Based Smart Garbage Monitoring and Segregation System

Trupti Thakre [1], Dhanshree Kalbande[2], Ayush Kamble[3], Gunika Gaikwad[4], Pratik Thakre[5]

Professor, Artificial Intelligence and Data, Priyadarshini College of Engineering, Nagpur, India[1]

Student, Artificial Intelligence and Data, Priyadarshini College of Engineering, Nagpur, India[2]

Student, Artificial Intelligence and Data, Priyadarshini College of Engineering, Nagpur, India[3]

Student, Artificial Intelligence and Data, Priyadarshini College of Engineering, Nagpur, India[4]

Student, Artificial Intelligence and Data, Priyadarshini College of Engineering, Nagpur, India[5]

Abstract: This project proposes an IoT-based smart garbage monitoring and segregation system that utilizes sensors, robotic arms and real-time notification system to optimizes waste management. The system monitors garbage bin fill levels, temperature , humidity and gas emissions Real-time data is transmitted to the Municipal corporation via Wi-Fi/GSM , ensuring timely emptying and maintenance. This system aims to reduce manual labor and provide data-driven insight for efficient waste management.

Keywords: IoT, Smart Garbage Monitoring, Waste Segregation, Sensors, Robotic Arm, Real-time Notification, Wi-Fi, GSM, Municipal Corporation, Data-driven Insights, Waste Management Optimization.

I. INTRODUCTION

Introduction

Waste management has become a critical challenge in urban areas due to the rapid increase in population, industrialization, and urbanization. Inefficient waste collection methods lead to overflowing garbage bins, causing environmental pollution, spreading diseases, and degrading the quality of life. Traditional waste management relies on periodic collection schedules, which often result in uncollected waste piling up or unnecessary emptying of partially filled bins. These inefficiencies contribute to higher operational costs, increased fuel consumption, and excessive human labor.

To address these challenges, an IoT-based Garbage Monitoring System has been developed, which automates the monitoring and reporting of waste levels in garbage bins. The system integrates advanced technologies such as ultrasonic sensors, Arduino microcontrollers, Wi-Fi/GSM modules, and cloud computing to optimize waste collection and disposal. By continuously monitoring garbage levels and transmitting real-time data to municipal authorities, the system ensures timely waste collection, prevents overflowing bins, and reduces the need for manual inspections. This smart waste management system plays a vital role in achieving cleaner cities, efficient resource utilization, and improved environmental sustainability.

The proposed system utilizes ultrasonic sensors mounted on garbage bins to measure the fill level. The collected data is then processed by an Arduino UNO microcontroller, which determines whether the bin has reached its threshold capacity.If the bin is nearly full, the system sends an SMS notification to the municipal corporation through a Wi-Fi/GSM IoT module. Additionally, the system provides real-time updates on a centralized dashboard, allowing authorities to track bin status and optimize collection routes. The system is powered by a 12V transformer, ensuring continuous operation even in areas with unstable power supply.

By integrating geolocation tracking, the system provides the exact longitude and latitude of each bin, enabling waste collection teams to locate and empty the bins efficiently. This smart approach eliminates unnecessary fuel consumption and reduces the carbon footprint associated with traditional waste collection methods. Furthermore, embedded systems allow for real-time monitoring of environmental factors such as temperature, humidity, and gas emissions, which are crucial for detecting potential hazards in waste disposal sites.

The IoT-based Garbage Monitoring System is a step toward smart city development, where technology is leveraged to improve urban infrastructure and services. This system ensures that waste collection is carried out systematically,

DOI: 10.17148/IJARCCE.2025.14217

reducing health risks, enhancing public hygiene, and minimizing the burden on waste management authorities. The integration of cloud-based data analytics further enables municipal corporations to analyze waste generation patterns and optimize waste management strategies based on real-time insights.

In addition to municipal applications, this system can be implemented in commercial complexes, educational institutions, hospitals, and industrial zones, where large amounts of waste are generated daily. By providing automated, data-driven solutions, the system enhances operational efficiency, minimizes human intervention, and promotes sustainable waste disposal practices.

The adoption of IoT-based waste management solutions is crucial for tackling modern urban challenges. By implementing real-time waste monitoring, authorities can move toward zero-waste cities, improving environmental sustainability while reducing the economic burden associated with inefficient waste collection. This system not only optimizes existing waste disposal mechanisms but also paves the way for future advancements, such as AI-driven waste segregation and automated robotic waste disposal systems, making urban waste management more efficient and environmentally friendly.

II. LITRATURE REVIEW

Literature Review

2.1 IoT and Cloud-Based Solutions

The Internet of Things (IoT) has revolutionized waste management by enabling real-time monitoring, cloud storage, and data-driven decision-making. Patel et al. (2022) developed a cloud-based garbage monitoring system that integrates: Wi-Fi-enabled sensors, transmitting data to a central server.

AI-based predictive analysis, forecasting waste generation trends. Automated alerts, notifying authorities of critical waste levels. Their study found that IoT-based solutions reduce waste collection costs by 40% by improving efficiency and eliminating redundant waste pickups.

2.2 Smart Waste Segregation Using AI and Robotics

AI and robotics have been **explored** for automated waste segregation. Rao et al. (2023) developed a robotic arm equipped with image processing and machine learning to identify and separate different types of waste (biodegradable, non-biodegradable, and recyclable). Their system achieved 85% accuracy in waste segregation, reducing landfill waste and improving recycling rates. Another study by Das et al. (2023) focused on AI-powered garbage bins that use cameras and sensors to classify waste into categories, automatically directing it to the appropriate disposal section. Their findings suggest that AI integration can enhance waste segregation efficiency by 30-40% compared to manual sorting.

2.3 Comparative Analysis of Waste Management Techniques

Studies confirm that sensor-based monitoring combined with AI-driven segregation offers the most efficient and sustainable approach to waste management.

2.5 IoT-Enabled Monitoring Using Ultrasonic Sensors

Recent research has focused on sensor-based garbage monitoring systems to automate waste tracking. Gupta et al. (2021) proposed a smart garbage monitoring system that uses ultrasonic sensors to detect waste levels and transmit data to municipal authorities via Wi-Fi/GSM modules. Their findings indicate that:Real-time monitoring reduces the chances of overflowing bins. Wireless communication allows authorities to optimize waste collection schedules. Location tracking helps in efficient route planning for waste trucks. Another study by Patel et al. (2022) demonstrated that sensor-based waste monitoring reduces waste collection frequency by 40%, leading to significant cost savings for urban municipalities.

DOI: 10.17148/IJARCCE.2025.14217

III. OBJECTIVES

Introduction
Waste management is a critical issue in urban areas, and traditional methods often lead to inefficiencies such as overflowing garbage bins, increased pollution, and high operational costs. The proposed IoT-based smart garbage monitoring and segregation system integrates sensors, artificial intelligence, and automation to improve waste collection, segregation, and disposal processes. This system will enhance real-time monitoring, optimize collection routes, and promote environmental sustainability.

3.1 Real-Time Monitoring of Garbage Levels :Real-time monitoring of garbage levels enhances waste management efficiency by utilizing ultrasonic sensors to detect waste levels in bins. The system transmits real-time data to municipal authorities via Wi-Fi or GSM, ensuring timely intervention. Automatic alerts are generated when bins reach a critical level, preventing overflow and maintaining cleanliness. By reducing the need for manual inspections, the system optimizes operational efficiency, saving time and resources while ensuring a more responsive and effective waste collection process.

3.2 Automated Waste Segregation Using AI and Robotics: Automated waste segregation using AI and robotics enhances waste management by utilizing an AI-powered classification system to identify biodegradable, non-biodegradable, and recyclable waste. By integrating image processing and machine learning, the system ensures accurate sorting, minimizing errors and contamination. This automation significantly reduces manual labor costs while improving efficiency in waste segregation. Additionally, it enhances recycling processes, reducing the volume of waste sent to landfills and promoting a more sustainable and eco-friendly wa**ste management system.**

3.3 Optimization of Waste Collection Routes: Optimization of waste collection routes enhances efficiency by enabling GPS tracking of bins for real-time locati**on** updates, allowing waste management authorities to monitor bin statuses remotely. By utilizing cloud-based analytics, the system optimizes waste collection schedules, ensuring that trucks operate on the most efficient routes. This reduces fuel consumption, minimizes transportation costs, and decreases overall operational expenses. Additionally, the system prioritizes bins that are nearly full, preventing overflow and unnecessary trips, ultimately making waste collection more cost-effective and environmentally sustainable.

3.4 Environmental Sustainability and Public Health: Cost reduction and resource optimization in waste management focus on improving efficiency while minimizing expenses. By eliminating unnecessary waste collection trips through real-time monitoring, the system significantly reduces operational costs. Optimizing garbage truck routes helps minimize fuel consumption, lowering transportation expenses and environmental impact. Additionally, efficient waste segregation extends landfill lifespan by diverting recyclable and biodegradable waste from disposal sites. Automating waste management processes further enhances efficiency, reducing manual labor and operational delays, ultimately leading to a more cost-effective and sustainable waste management system.

3.5 Integration with Smart City Infrastructure: Optimization of waste collection routes improves efficiency by using GPS tracking for real-time bin location updates and cloud-based analytics to optimize collection schedules. This ensures that waste trucks prioritize bins that are nearly full, reducing unnecessary trips and preventing overflow. By streamlining collection processes, the system minimizes fuel consumption, lowers transportation costs, and enhances overall operational efficiency, making waste management more cost-effective and environmentally sustainable.

3.6 Cost Reduction and Resource Optimization:
Cost reduction and resource optimization in waste management focus on eliminating unnecessary waste collection trips, thereby reducing operational costs and improving efficiency. By optimizing garbage truck routes, the system minimizes fuel usage, lowering transportation expenses and environmental impact. Effective waste segregation extends the lifespan of landfills by diverting recyclable and biodegradable materials from disposal sites. Additionally, automating waste management processes enhances efficiency, reduces manual labor, and streamlines operations, ultimately leading to a more cost-effective and sustainable waste management system.

IV. ARCHITECTURE AND MODULES.

Architecture
Waste management is a significant challenge in urban areas due to inefficient garbage collection, improper segregation, and environmental pollution. This project proposes an IoT-enabled smart garbage monitoring and segregation system that automates waste segregation, detects harmful gases, and provides real-time data to municipal authorities. The system

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ≒ Peer-reviewed & Refereed journal ≒ Vol. 14, Issue 2, February 2025

DOI: 10.17148/IJARCCE.2025.14217

integrates sensors, microcontrollers, actuators, and a cloud-based alert mechanism to optimize waste collection and reduce manual effort. The IoT-based Smart Garbage Monitoring and Segregation System is designed to automate waste management using sensor networks, microcontrollers, and cloud-based communication. This system efficiently detects garbage levels, segregates waste, and notifies municipal authorities in real time. The system architecture consists of hardware and software components that work together to monitor, analyze, and optimize waste disposal operations.

4.1 System Architecture

The perception layer comprises sensors and actuators responsible for data collection and physical operations. The NodeMCU (ESP8266/ESP32) serves as the core processing unit, reading sensor data and communicating with the cloud. Various sensors are integrated, including an Ultrasonic Sensor to measure garbage fill levels, a Gas Sensor (MQ-135) to detect harmful gases like Methane (CH4), Carbon Monoxide (CO), and Ammonia (NH3), a Moisture Sensor to differentiate between dry and wet waste for segregation, and a Temperature & Humidity Sensor (DHT11) to monitor environmental conditions inside the bin. Servo motors are used to operate robotic arms for waste segregation. Additionally, a power supply unit with capacitors and a power regulation circuit ensures stable power for seamless system operation.

4.2 Modules:

The proposed system consists of multiple functional modules working together to enable efficient garbage monitoring and segregation. The Sensing and Data Collection Module is responsible for continuously monitoring garbage levels, gas emissions, moisture content, and temperature using various sensors. The Processing and Decision-Making Module processes this sensor data using the NodeMCU (ESP8266/ESP32), determining whether the garbage bin is full, if gas levels exceed safety limits, and whether the waste is dry or wet. This information helps in automating waste segregation and alerts municipal authorities about the bin's status.

The Waste Segregation and Disposal Module uses servo motors to operate robotic arms that separate dry and wet waste into designated compartments. Once the bin reaches its full capacity, the Communication and Notification Module transmits alerts via Wi-Fi or GSM to municipal authorities for timely waste collection. Additionally, the User Interface and Dashboard Module provides real-time monitoring of bin status, sensor readings, and historical data for efficient waste management planning. The integration of these modules ensures a smart, automated, and data-driven approach to urban waste management, minimizing manual intervention and improving overall efficiency.

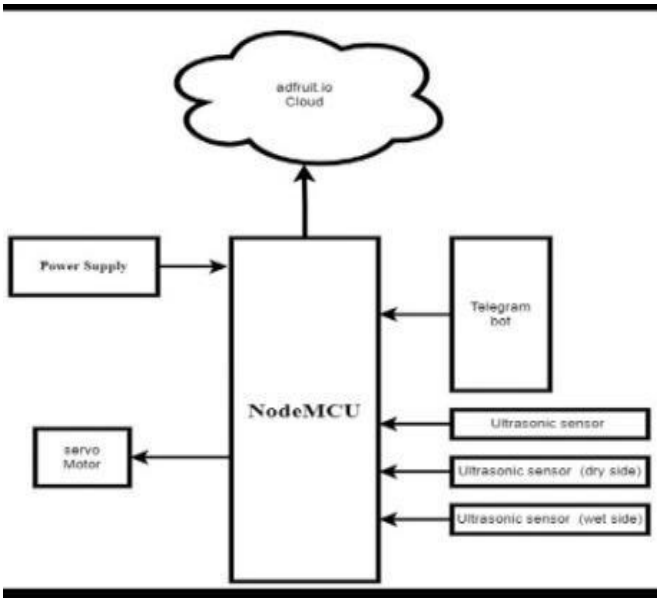
The proposed IoT-based Smart Garbage Monitoring and Segregation System is composed of several functional modules, each performing a specific role in automating waste management. The Sensing and Data Collection Module continuously gathers real-time data from multiple sensors, including the Ultrasonic Sensor for garbage fill levels, the Gas Sensor (MQ-135) for detecting harmful gases like Methane (CH4), Carbon Monoxide (CO), and Ammonia (NH3), the Moisture Sensor for distinguishing between dry and wet waste, and the Temperature & Humidity Sensor (DHT11) for monitoring environmental conditions inside the bin.

This data is then sent to the Processing and Decision-Making Module, where the NodeMCU (ESP8266/ESP32) microcontroller processes the inputs and determines whether the bin is full, if gas concentrations exceed safe limits, or if segregation is required. Based on these decisions, appropriate actions are taken, such as segregating waste or notifying authorities.

The Waste Segregation and Disposal Module utilizes servo motors to operate robotic arms that automatically sort dry and wet waste into designated compartments. When the bin reaches full capacity, the Communication and Notification Module sends alerts through Wi-Fi or GSM to municipal authorities, ensuring timely waste collection and reducing the risk of overflowing garbage bins. The Cloud Storage and Analytics Module stores collected data, allowing for predictive analysis of waste patterns and optimization of garbage collection schedules.

Finally, the User Interface and Dashboard Module provides a web or mobile-based dashboard for municipal authorities, displaying real-time bin status, sensor readings, and historical data for strategic waste management planning. By integrating these modules, the system ensures automated, efficient, and data-driven waste collection, minimizing manual intervention, optimizing collection schedules, and promoting a cleaner urban environment.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 14, Issue 2, February 2025
DOI: 10.17148/IJARCCE.2025.14217



V. ALGORITHM

Step 1: System Initialization

1. Start the system and initialize all components (NodeMCU, sensors, servo motors, communication module).
2. Connect to the Wi-Fi or GSM network for data transmission.
3. Set initial values for sensor thresholds (e.g., bin full level, gas emission limits).

Step 2: Data Collection from Sensors

4. Continuously read data from the Ultrasonic Sensor to check the garbage fill level.
5. Read data from the Gas Sensor (MQ-135) to detect harmful gases (CH_4, CO, NH_3).
6. Read data from the Moisture Sensor to determine if the waste is wet or dry.
7. Read temperature and humidity levels using the DHT11 Sensor.

Step 3: Data Processing and Decision Making

8. If garbage bin is full (fill level ≥ threshold), proceed to Step 6.
9. If gas concentration exceeds safe limits, trigger an emergency alert to the authorities.
10. If waste is wet, activate the servo motor to segregate it into the wet waste bin.

VI. HARDWARE COMPONENT

NodeMCU
NodeMCU is an open-source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC, and hardware which is based on the ESP-12 module. It is widely used for IoT applications and is programmable with the Arduino IDE.

Key Features:
Wi-Fi Connectivity: Built-in ESP8266 module allows for wireless communication. GPIO Pins: General-purpose input/output pins for interfacing with other devices. Programming: Can be programmed using Lua scripting language or the Arduino IDE.

2. Capacitors
Capacitors store and release electrical energy. They are used in various applications such as power supply filtering and signal coupling.
4700µF, 25V: Large capacitor for filtering power supply. 220µF, 25V: Medium capacitor for filtering and decoupling.

DOI: 10.17148/IJARCCE.2025.14217

100µF, 16V: Smaller capacitor for smoothing small voltage fluctuations. 1000µF, 25V: Another large capacitor for power supply stabilization.

3. Ultrasonic Sensor (HC-SR04)

These sensors measure distance by sending out ultrasonic waves and measuring the time it takes for the waves to bounce back.

Key Features:
Range: Typically 2 cm to 400 cm. Accuracy: ±3 mm.
Usage: Used in obstacle detection, distance measurement, and level sensing.

4. 7805 Voltage Regulator IC

The 7805 is a voltage regulator IC that outputs a constant 5V DC from a higher voltage input (7V-35V). Key Features:
Output Voltage: 5V. Input Voltage: 7V to 35V.
Current Rating: Typically up to 1.5A.

5. PCB (Printed Circuit Board)

A PCB is used to physically support and electrically connect electronic components. The components are soldered onto the board, creating a permanent and robust assembly.

6. Wires (1m)

Wires are used to connect different components in a circuit. They come in different colors and gauges to help in organizing and managing connections.

7. Servo Motor

A servo motor is a rotary actuator that allows for precise control of angular position. It is commonly used in robotics, CNC machinery, and remote control systems.

Key Features:
Control: Operated via PWM (Pulse Width Modulation) signals. Rotation: Typically can rotate 0° to 180°.
Torque: Depending on the model, can handle different loads.

8. Moisture Sensor

A moisture sensor measures the moisture level in soil. It is used in agriculture and gardening to automate watering systems.
Key Features:
Probes: Two or more probes inserted into the soil to measure conductivity. Output: Analog or digital signal indicating moisture level.
Usage: Connected to microcontrollers like NodeMCU to automate watering based on soil moisture. How It All Comes Together

These components can be combined to create various projects. For example, in a smart irrigation system: The NodeMCU processes input from the moisture sensor and sends data over Wi-Fi.
The ultrasonic sensors measure water levels in tanks. The 7805 voltage regulator ensures a stable 5V supply. Capacitors smooth out the power supply.
Servo motor controls the valve for watering. Wires connect everything together on the PCB..

Description

Hs pin of the IC positive unregulated voltage is given in the regulation.
In this pin where the ground is given. This pin is neutral for equally the input and output.
The output of the regulated 5V is taken out at this pin of the IC regulator.

The NodeMCU is a popular open-source IoT platform that is designed to facilitate the development of connected devices and smart applications. It’s based on the ESP8266 Wi-Fi module from Espressif Systems, which provides powerful wireless networking capabilities. Herse a detailed overview of its features, specifications, and applications:
Specifications

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 14, Issue 2, February 2025
DOI: 10.17148/IJARCCE.2025.14217

Microcontroller: ESP8266 (Ten silica 32-bit RISC CPU Xtensa LX106) Operating Voltage: 3.3V
Input Voltage: 5V via USB, 7-12V via Vin pin
Digital I/O Pins: 16 GPIO pins (General Purpose Input/Output) Analog Input Pins: 1 ADC (Analog to Digital Converter) pin
Clock Speed: 80 MHz (up to 160 MHz) Flash Memory: 4 MB
SRAM: 64 KB
Wi-Fi Standards: 802.11 b/g/n
USB-TTL: On-board USB to TTL converter for easy programming Key Features
Wi-Fi Connectivity: Built-in ESP8266 module allows for seamless wireless communication, making it ideal for IoT applications.

Compact and Integrated Design: Combines microcontroller and Wi-Fi capabilities on a single chip, reducing the complexity and size of the development board.
Easy Programming: Can be programmed using the Arduino IDE or Lua scripting language, providing flexibility for developers.
Extensive GPIO: Multiple GPIO pins for interfacing with sensors, actuators, and other peripherals. Power Management: Supports deep sleep mode to conserve power in battery-operated applications. Pinout

Heres a brief description of some important pins:
3.3V: Power supply pin (provides 3.3V). GND: Ground pin.
Vin: Voltage input pin (7-12V).
D0-D8, SD3, RX, TX: Digital I/O pins. A0: Analog input pin.
RST: Reset pin.
EN: Enable pin (used to enable or disable the module). Getting Started with NodeMCU
Install the Arduino IDE: Download and install the Arduino IDE from the official website. Add ESP8266 Board Manager:
Go to File > Preferences and add the following URL in the Additional Boards Manager URLs: http://arduino.esp8266.com/stable/package_esp8266com_index.json
Go to Tools > Board > Boards Manager, search for ESP8266, and install it. Select NodeMCU Board:
Go to Tools > Board and select NodeMCU 1.0 (ESP-12E Module).
Connect NodeMCU to Computer: Use a Micro-USB cable to connect the NodeMCU to your computer. Upload Code:

Write your code in the Arduino IDE.

Select the correct COM port under Tools > Port.

Click on the Upload button to upload the code to the NodeMCU. Sample Code

Heres a simple example to blink an LED connected to a GPIO pin: Cpp

```
Void setup() {
pinMode(D1, OUTPUT);  // Initialize D1 (GPIO5) as an output pin
}

Void loop() {
digitalWrite(D1, HIGH);  // Turn the LED on delay(1000);        // Wait for 1 second digitalWrite(D1, LOW);  // Turn the LED off delay(1000);    // Wait for 1 second
}
```

Applications
NodeMCU is widely used in various applications, including:
Home Automation: Controlling lights, appliances, and security systems via Wi-Fi.
Environmental Monitoring: Measuring and transmitting data such as temperature, humidity, and air quality. Smart Agriculture: Automating irrigation and monitoring soil moisture.

Robotics: Controlling robots and drones with wireless communication. Wearable Technology: Developing smart wearables that connect to the internet.

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 2, February 2025

DOI: 10.17148/IJARCCE.2025.14217

NodeMCU provides a powerful and flexible platform for developing IoT projects. Its integration of Wi-Fi capabilities with a microcontroller makes it ideal for a wide range of applications. By leveraging its features and ease of use, developers can create innovative solutions for connected devices.

NodeMCU is an IoT Module based on the ESP8266 wifi chip Module. NodeMCU uses the Lua Scripting language and is an open-source Internet resource (IoT) platform. This module has CH340g USB to TTL Converter IC on board.

The ESP8266 NodeMCU CP2102 board has ESP8266 which is a highly integrated chip designed for the needs of the new IoT-connected world. Provides a complete solution and contains a Wi-Fi network, allowing it to host an application or download all Wi-Fi communication activities from another app processor.

The ESP8266 has the ability to process and store capabilities allowing it to be integrated with sensors and other device-specific devices through its GPIOs with a few earlier improvements and fewer loading during operation. Its high level of chip integration allows for minimal external rotation, and the entire solution, including the front module, is designed to accommodate a small PCB area.

ESP8266 NodeMCU development board – a real plug-and-play solution for less expensive projects using WiFi. The module comes up first with the NodeMCU firmware so they are ready to go – just install your USB driver.
The ESP-12 Lau NodeMCU WIFI Development Board Internet Of Things board contains a complete ESP8266 Wi-Fi module with all broken GPIO, a complete USB-serial interface, and power in a single packet of the breadboard.
The board is pre-loaded with NodeMCU – Lau-based firmware for ESP8266 that allows easy control with pure scripting language – Lau – so you're ready to go in just a few minutes.

ESP-12 Lau NodeMCU WIFI Dev Board Internet Of Things with ESP8266 is a very easy-to-use Wi-Fi microcontroller + in one platform to build projects with Wi-Fi and IoT-Internet of Things.

The board is based on the ESP8266 Wi-Fi Module chip with the ESP-12 SMD footprint. The Wi-Fi upgrade board has already installed on its board all the ESP8266 (ESP-12E) code for organizing and uploading the code. It has built-in USB serial chip upload codes, a 3.3V voltage regulator, and a logic level converter circuit so you can quickly download codes and connect your circuits.

HomeElectronicsESP8266 Node MCU board specifications and pins descriptions. Frequently asked questions related to WI-FI board esp8266 node MCU

ESP8266 Node MCU board specifications and pins descriptions. Frequently asked questions related to WI-FI board esp8266 node MC

NodeMCU is an IoT Module based on the ESP8266 Wi-Fi chip Module. NodeMCU uses the Lau Scripting language and is an open-source Internet resource (IoT) platform. This module has CH340g USB to TTL Converter IC on board.

The ESP8266 NodeMCU CP2102 board has ESP8266 which is a highly integrated chip designed for the needs of the new IoT-connected world. Provides a complete solution and contains a Wi-Fi network, allowing it to host an application or download all Wi-Fi communication activities from another app processor.

The ESP8266 has the ability to process and store capabilities allowing it to be integrated with sensors and other device-specific devices through its GPIOs with a few earlier improvements and fewer loading during operation. Its high level of chip integration allows for minimal external rotation, and the entire solution, including the front module, is designed to accommodate a small PCB area.

ESP8266 NodeMCU development board – a real plug-and-play solution for less expensive projects using Wi-Fi. The module comes up first with the NodeMCU firmware so they are ready to go – just install your USB driver.

The ESP-12 Lau NodeMCU WIFI Development Board Internet Of Things board contains a complete ESP8266 Wi-Fi module with all broken GPIO, a complete USB-serial interface, and power in a single packet of the breadboard.

The board is pre-loaded with NodeMCU – Lua-based firmware for ESP8266 that allows easy control with pure scripting language – Lua – so you're ready to go in just a few minutes.

DOI: 10.17148/IJARCCE.2025.14217

ESP-12 Lua NodeMCU WIFI Dev Board Internet Of Things with ESP8266 is a very easy-to-use Wi-Fi microcontroller + in one platform to build projects with Wi-Fi and IoT-Internet of Things.

The board is based on the ESP8266 Wi-Fi Module chip with the ESP-12 SMD footprint. The Wi-Fi upgrade board has already installed on its board all the ESP8266 (ESP-12E) code for organizing and uploading the code. It has built-in USB serial chip upload codes, a 3.3V voltage regulator, and a logic level converter circuit so you can quickly download codes and connect your circuits.

If you are familiar with Arduino, using NodeMCU is the next step if you are looking for a compatible alternative, with Wi-Fi. NodeMCU is based on ESP8266-12E Wi-Fi System-On-Chip. Loaded with open source, Lua-based firmware. Getting started with NodeMCU is very easy.

Ideal for IoT Applications, and other wireless connectivity applications. This chip has a lot in common with Arduino – both of these microcontroller-armed prototyping boards can be configured using the Arduino IDE. ESP8266 is more recently released than Arduino and has stronger meanings. It has a 32-bit RISC processor clocked at 80MHz, as well as a RAM supply and supports up to 16 MB of external flash storage.

The device is especially useful for IoT applications, due to its small footprint and built-in Wi-Fi support. ESP8266 Integrates 802.11b / g / n HT40 Wi-Fi transceiver, so it can only connect to a Wi-Fi network and interact with the Internet. It can also set up its own network, allowing other devices to connect directly to it. There is a board controller that ensures very clean power on the MCU itself, as well as a push-button reset and USB connection to make it easier to connect to your computer.

So we start with its power requirements. The operating voltage of ESP8266 is 3 to 3.3v. The NodeMCU board has produced an LDO voltage regulator that maintains a power level of 3.3v. In case it provides 600mA reliably.

Energy requirements:
Operating voltage: 2.5 to 3.3 v.
On-board power control 3.3v 600mA It currently operates 800mA

Switching Board and LED Indicator:
The ESP8266 comes with 2 switches one is reset and the other is a flash button, a reset button to use of course reset NodeMCU and a flash button is used to download and use while updating firmware.

The board is also built into the LED indicator connected to the D0 pin and can be adjustable.

NodeMCU installs the CP2102 USB-to-UART Bridge Controller, which helps to convert the USB signal to serial and allows your computer to configure and connect to the ESP8266 chip. It also has a connection speed of 4.5Mbps.

I / O pins
The ESP6266 NodeMCU board comes with 17 GPIO, these anchors assign all kinds of structural functions like- 10 bit ADC channel.

PWM output. UART display.
SPI, I2C, I2S display: to connect all kinds of sensors and devices. I2S: sound installation for your project.
Due to the pin multiplexing feature of ESP8266 (multiple parameters multiplied by one GPIO pin). Which means that one GPIO pin can work as a PWM / UART / SPI.

Esp8266 is a microchip in a QFN package with both TCP / IP suite capabilities and a microcontroller. Esp8266 brings a highly integrated Wi-Fi solution that meets the Internet industry's needs for things like low cost, efficient power consumption, reliable performance, and integrated architecture. Manufactured by Espressif Systems in Shanghai, China.

The full power of the Wi-Fi network can serve as a slave to the microcontroller manager or as a standalone application. When we say a host microcontroller host it means it can be used as a Wi-Fi adapter on any microcontroller using SPI or UART interface. While used as a stand-alone device it can enable microcontroller and Wi-Fi networking. Esp8266 is based around the Tensilica L106 Diamond series which is a 32-bit processor and has on-chip SRAM. It also includes power modules, RF balun, RF receiver and transmitter, analog and transmitter-receiver, digital baseband, amplifier, filters, and other small components.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁒ Peer-reviewed & Refereed journal ⁒ Vol. 14, Issue 2, February 2025
DOI: 10.17148/IJARCCE.2025.14217

Details of ESP8266 NodeMCU Board

The Esp8266 specification is divided into three parts: Hardware, Software, and Wi-Fi.
By Hardware specification, its package size is QFN 32pins with a size of 5mm x 5mm. Performance ratings range from 2.5V to 3.6V. The chip uses 80mA current on average.

Its CPU is a Tensilica L106 32bit processor with on-chip SRAM.
The border interface contains UART, SDIO, SPI, I2C, I2S, IR remote control, GIPO's, ADC, PWM, LED Light, and buttons.

Its firmware can be updated using OTA and UART. It uses IPv4, TCP, UDP, and HTTP as network protocols. The user can configure to use pre-configured AT commands, Cloud Server, and run the mobile app.

Wi-Fi frequency ranges from 2.4G to 2.5G. It uses the Wi-Fi protocol IEEE 802.11 b / g / n. The power of Esp6266 Wi-Fi is guaranteed by the Wi-Fi Alliance.

ESP8266 NodeMCU Release:

1) Power Pins
There are four power pins namely- a VIN pin and three 3.3V pins.
The VIN pin can be used to directly supply ESP8266 and its components if you have a controlled 5V voltage source.
The 3.3V pins are the output of the voltage board controller. These pins can be used to supply power to external parts.

2) GND- Ground
It is the ground pin of the ESP8266 NodeMCU development board.

3) I2C Pins
These are used to integrate all types of I2C sensors and parameters in the project. Both I2C Master and I2C Slave are supported.
The performance of the I2C optical connector can be systematically detected, and the clock frequency is 100 kHz at maximum speed. It should be noted that the frequency of the I2C clock should be greater than the frequency of the slowest clock of the slave device.

4) GPIO Pins
The ESP8266 NodeMCU has 17 GPIO anchors that can be assigned to various functions such as I2C, I2S, UART, PWM, IR Remote Control, LED Light, and Button respectively. Each GPIO digitally-enabled can be adjusted to internal drag or drop or set to high intensity. When set as input, it can also be set to Edge-trigger or level-trigger to produce CPU interference.

5) ADC channel
NodeMCU is embedded 10-bit with SAR ADC accuracy. These two functions can be performed using the ADC viz. VDD3P3 pin power supply and TOUT pin power input. However, they cannot be used simultaneously.

6) UART
The ESP8266 NodeMCU has 2 UART domains, namely UART0 and UART1, which offer different connections (RS232 and RS485), and can communicate up to 4.5 Mbps. UART0 pins (TXD0, RXD0, RST0 & CTS0) can be used for communication. Supports fluid control. However, the UART1 (TXD1 pin) includes a data transfer signal only, so it is used to print the log.

7) SPI Pins
ESP8266 incorporates two SPIs (SPI and HSPI) into slave and master modes.
These SPIs also support the following SPI features:
4-time modes for SPI format transfer
Up to 80 MHz with split clocks of 80 MHz Up to 64-Byte FIFO

8) SDIO Pins
ESP8266 incorporates the Secure Digital Input / Output Interface (SDIO) which is used to connect directly to SD cards.
4-bit 25 MHz SDIO v1.1 and 4-bit 50 MHz SDIO v2.0 are supported.

ISSN (O) 2278-1021, ISSN (P) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 2, February 2025
DOI: 10.17148/IJARCCE.2025.14217

9) PWM Pins
The board has 4 Pulse Width Modulation (PWM) channels.
PWM output can be programmed and used to drive digital engines and LEDs.
The frequency range of PWM ranges from 1000 μs to 10000 μs, eg between 100 Hz and 1 kHz.

10) Control Pins
These anchors are used to control ESP8266.
These anchors include Chip Enable pin (EN), Reset pin (RST), and WAKE pin.

11) EN Pin(Enable)
The ESP8266 chip is enabled when the EN pin is pulled INSIDE. When pulled LOW the chip works at low power.

12) RST Pin(Reset)
The RST pin is used to reset the ESP8266 chip.

13) Wake Pin
The use of Wake pin is used to wake up the chip from a deep sleep.

Ultrasonic sensor
An ultrasonic sensor is an electronic device that measures the distance to an object by emitting ultrasonic sound waves and then detecting the reflected waves. Heres a detailed overview:

Working Principle
Emission: The sensor emits ultrasonic sound waves (typically at a frequency of 40 kHz) using a transmitter. Reflection: These sound waves travel through the air until they hit an object and bounce back as echoes.
Detection: The sensors receiver detects the reflected waves.
Calculation: The sensor measures the time taken for the sound waves to travel to the object and back. Using the speed of sound (approximately 343 meters per second), it calculates the distance to the object.

Key Components
Transducer: Converts electrical signals into ultrasonic sound waves and vice versa. Microcontroller: Processes the signals and performs calculations.
Signal Processing Circuit: Amplifies and filters the received signals. Specifications
Range: Typically between 2 cm to 4.5 meters. Accuracy: ±3 mm.
Response Time: Around 50 milliseconds to 200 milliseconds. Beam Angle: Approximately 15°.
Operating Voltage: Usually 5V to 30V.
Output: Digital (e.g., pulse width modulation) or analog signal indicating distance. Applications

Ultrasonic sensors are used in various applications, including:
Obstacle Detection: In robotics and automotive systems to detect obstacles and avoid collisions. Distance Measurement: In industrial automation for precise distance measurements.
Level Sensing: In liquid level measurement in tanks and containers. Medical Devices: In medical imaging and diagnostics.

Example: HC-SR04 Ultrasonic Sensor
One of the most commonly used ultrasonic sensors is the HC-SR04. Its widely used in DIY projects and educational purposes.
Pin Configuration:
Trig: Trigger pin to start the measurement. Echo: Echo pin to receive the signal.
Vcc: Power supply (5V). GND: Ground.

Figure-2 HC-SR04 ultrasonic sensor
Servo motors are widely used in robotics, CNC machinery, and other applications where precise position control is necessary.
Components of a Servo Motor
Motor: Typically a DC motor that provides the necessary rotational movement.
Control Circuit: An electronic circuit that processes control signals and adjusts the motor's position. Position Sensor: Often a potentiometer, this sensor provides feedback on the motor's current position. Gear Assembly: Reduces the speed and increases the torque of the motor.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⪤ Peer-reviewed & Refereed journal ⪤ Vol. 14, Issue 2, February 2025
DOI: 10.17148/IJARCCE.2025.14217

Working Principle
Servo motors operate using a closed-loop feedback mechanism to control their position. Heres a step-by-step overview: Control Signal: A pulse-width modulation (PWM) signal is sent to the control circuit, indicating the desired position. Error Detection: The position sensor continuously monitors the motor’s position and sends this data to the control circuit. Correction: The control circuit compares the desired position with the current position and adjusts the motor’s movement accordingly to minimize the error.

Achieve Position: The motor rotates to the required position, and the control loop keeps it there. Types of Servo Motors Positional Rotation Servo: Can rotate up to 180 degrees. It is the most common type used in robotics and RC (radio-controlled) applications.

Continuous Rotation Servo: Can rotate continuously in either direction. It is used where full 360-degree rotation is needed.

Linear Servo: Converts rotational motion into linear motion, suitable for applications requiring linear positioning. Specifications

Torque: The force exerted by the motor, typically measured in kg.cm or oz.in. Speed: The speed at which the motor can rotate, measured in seconds per 60 degrees. Voltage: The operating voltage range of the motor, usually 4.8V to 6V.

Weight: The weight of the servo motor, important in applications where weight is a consideration. Applications

Robotics: Servo motors are used to control the movement of robotic arms, legs, and other parts.

Remote Control Vehicles: Used in RC cars, boats, and planes to control steering, throttle, and other functions. Automation: Used in CNC machines, 3D printers, and other automated equipment for precise control.

Aerospace: Used in flight control systems for precise adjustments of control surfaces. Industrial: Employed in conveyor belts, packaging machines, and other industrial machinery. Programming a Servo Motor with Arduino

Heres a simple example of how to control a servo motor using an Arduino:

Connections:
Servo Motor: Connect the control wire (usually yellow or white) to a PWM-capable pin on the Arduino (e.g., pin 9). Power: Connect the power wire (usually red) to the 5V pin on the Arduino.

Ground: Connect the ground wire (usually black or brown) to the GND pin on the Arduino.

Figure – 3 servo motor

A moisture sensor is a device used to measure the moisture content in soil or other materials. Heres a detailed overview: Working Principle

Moisture sensors typically measure the volumetric water content in soil by using properties such as electrical resistance, dielectric constant, or interaction with neutrons. The sensor outputs a signal that corresponds to the moisture level, which can be read by a microcontroller or other device1.

Components
Probes: Two or more probes that are inserted into the soil to measure moisture. These probes act as variable resistors whose resistance changes with soil moisture2.
Electronic Module: Converts the resistance measurement into a readable output, usually an analog voltage. Potentiometer: Adjusts the threshold value for moisture detection.

LEDs: Indicate the status of the sensor and whether the moisture level exceeds the threshold. Specifications
Voltage: Typically operates at 5V. Current: Consumes less than 20mA.
Output: Analog (voltage proportional to moisture level) and digital (indicating if moisture exceeds a set threshold). Temperature Range: Works best between 10°C and 30°C.
Applications

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁒ Peer-reviewed & Refereed journal ⁒ Vol. 14, Issue 2, February 2025

DOI: 10.17148/IJARCCE.2025.14217

Moisture sensors are widely used in:
Agriculture: To automate irrigation systems and ensure optimal watering of crops. Gardening: To prevent over- or under-watering of plants.
Environmental Monitoring: To measure soil moisture in scientific experiments and studies. Smart Irrigation Systems: To control watering based on real-time soil moisture data.
Example: FC-28 Soil Moisture Sensor
The FC-28 is a common soil moisture sensor used in DIY projects. It has four pins:
VCC: Power supply (5V).
A0: Analog output (voltage proportional to moisture level). D0: Digital output (indicating if moisture exceeds threshold).
GND: Ground.

Figure-4 moisture sensor Capacitors

4700uF, 25V Capacitor: This is an electrolytic capacitor with a capacitance of 4700 microfarads (uF) and a voltage rating of 25 volts. It's commonly used for filtering, smoothing out voltage fluctuations, and storing charge in circuits1.

220uF, 25V Capacitor: Another electrolytic capacitor, this one has a capacitance of 220 microfarads and a voltage rating of 25 volts. It's used in similar applications as the 4700uF capacitor but with a lower capacitance.

100uF, 16V Capacitor: This capacitor has a capacitance of 100 microfarads and a voltage rating of 16 volts. It's typically used for smaller filtering applications.

1000uF, 25V Capacitor: With a capacitance of 1000 microfarads and a voltage rating of 25 volts, this capacitor is used for larger filtering and energy storage applications.

Voltage Regulator IC (7805)
7805 Voltage Regulator IC: This is a popular voltage regulator IC that outputs a stable 5V DC voltage from a higher input voltage. It's widely used in various electronic circuits to provide a consistent power supply.

PCD (Printed Circuit Board)
PCD 1: This refers to a single-layer printed circuit board (PCB) with components mounted on it. It's a basic type of PCB used for simple circuits.

Wires (1m)
1m Wires: These are wires measuring 1 meter in length, typically used for connecting components on a PCB or in a circuit

Methodology Component Setup:
NodeMCU: Acts as the central controller.

Capacitors: Used for power smoothing and stabilization. Ultrasonic Sensors: Measure water levels in a tank.
7805 Voltage Regulator IC: Ensures stable 5V power supply to the components. Servo Motor: Controls the water valve for irrigation.

Moisture Sensor: Monitors soil moisture levels. PCD and Wires: For connecting all components. Sensor Readings:
The moisture sensor provides analog readings representing soil moisture levels. Ultrasonic sensors measure water levels in the storage tank.

Data Processing:
The NodeMCU processes sensor data to determine whether the soil needs watering. Actuation:
Based on soil moisture levels, the NodeMCU sends signals to the servo motor to open or close the water valve.

Communication:
The NodeMCU can send data to a cloud service for remote monitoring and control. Working Principle

Power Regulation:
The 7805 voltage regulator ensures a stable 5V power supply. Capacitors smooth out voltage fluctuations.

ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 2, February 2025

DOI: 10.17148/IJARCCE.2025.14217

Moisture Sensing:
The moisture sensor detects the soil's moisture level and sends an analog signal to the NodeMCU. Ultrasonic Sensing: Ultrasonic sensors measure the water level in the storage tank and send the data to the NodeMCU. Data Processing and

Decision Making:
The NodeMCU processes the sensor data.
If the soil moisture level is below a predefined threshold, the NodeMCU activates the servo motor.

Actuation:
The servo motor opens the valve to start irrigation.

Once the soil moisture reaches the desired level, the NodeMCU stops the irrigation.

Ultrasonic Sensor: You'll need an ultrasonic sensor (like the HC-SR04) to measure distances. This sensor emits ultrasonic waves and measures the time it takes for the waves to bounce back, calculating the distance to an object1.
Servo Motor: A servo motor allows for precise control of angular or linear position, speed, and torque. You can use it to move objects or control mechanisms based on the input from the ultrasonic sensor2.

NodeMCU: This is an open-source firmware and development kit based on the ESP8266 Wi-Fi SoC. It's great for IoT projects and can be programmed using Lua or Arduino IDE4.

Ad fruit: Ad fruit provides various components and development boards that can be integrated with your project. They offer a wide range of sensors, actuators, and other electronic components.

Cloud Integration: You can use cloud services like AWS, Google Cloud, or Azure to store and process data collected by your sensors. This allows for remote monitoring and control.

Telegram Bot: You can create a Telegram bot to receive notifications or control commands from your project. This can be done using the Bot API provided by Telegram.

Power Supply: Ensure you have a suitable power supply to power all your components. The NodeMCU and other sensors typically run on 5V, so a stable 5V power supply is essential.

Figure-4 connected diagram #include <Servo.h>
#include <ESP8266WiFi.h> #include <WiFiClientSecure.h> #include <UniversalTelegramBot.h> #include <Adafruit_MQTT.h>
#include <Adafruit_MQTT_Client.h>

Servo myservo;
// WiFi Credentials
Const char* ssid = "project";
Const char* password = "12345678";

// Telegram Credentials
#define BOTtoken "7952100255:AAGBUV_0l4oKcpciIrjJ9V20B7jH5lBKUrI" #define CHAT_ID "6746079346"//6746079346

// Adafruit IO Credentials
#define AIO_SERVER "io.adafruit.com"
#define AIO_SERVERPORT 1883 // Use 8883 for SSL #define AIO_USERNAME "gunika_123"
#define AIO_KEY "aio_pflE71y7xkHZOMUGiclWWDP4awCf"

// WiFi Client for Telegram
X509List cert(TELEGRAM_CERTIFICATE_ROOT);
WiFiClientSecure.h client; UniversalTelegramBot.h bot(BOTtoken, client);

WiFiClient aioClient;
Adafruit_MQTT_Client.h mqtt(&isocline, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 2, February 2025
DOI: 10.17148/IJARCCE.2025.14217

```
// Adafruit IO Feeds
Adafruit_MQTT_Publish distance1Feed = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/distance1");
Adafruit_MQTT_Publish distance2Feed = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/distance2");
Unsigned long lastPublishTime = 0;
Const unsigned long publishInterval = 5000; // 5 seconds interval for publishing Void MQTT_connect();
Const int trigPin = D6; Const int echoPin = D5; Const int trigPin1 = D3; Const int echoPin1= D4; Const int trigPin2=
D7; Const int echoPin2= D8; #define led D0
#define led1 D1 Int cnt=0;
Int cnt1=0;
Int value = A0;

Float duration, distance,duration1, distance1,duration2, distance2;

Void setup() {  pinMode(trigPin, OUTPUT); pinMode(echoPin, INPUT); pinMode(trigPin1, OUTPUT);
pinMode(echoPin1, INPUT); pinMode(trigPin2, OUTPUT); pinMode(echoPin2, INPUT); pinMode (value, INPUT);
pinMode(led, OUTPUT); pinMode(led1, OUTPUT);

Serial.begin(115200);

//delay(1000); WiFi.mode(WIFI_STA);
WiFi.begin(ssid, password);
While (WiFi.status() != WL_CONNECTED) { Serial.print(".");
Delay(500);
}
Serial.println("\nWiFi connected"); Serial.print("IP address: "); Serial.println(WiFi.localIP());

// Telegram setup
configTime(0, 0, "pool.ntp.org");          // Get UTC time via NTP client.setTrustAnchors(&cert); // Add root certificate
for api.telegram.org Bot.sendMessage(CHAT_ID, "System started", "");

delay(1000); myservo.attach(D2);

myservo.write(95); delay(1000);

}

Void loop()
{
MQTT_connect();

Value = analogRead(A0); If (value < 900)
{
Serial.println(value); Myservo.write(170); Delay(1000); Myservo.write(82); Delay(1000);

}

digitalWrite(trigPin, LOW); delayMicroseconds(2); digitalWrite(trigPin, HIGH); delayMicroseconds(10);
digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH); distance = (duration*.0343)/2; Serial.print("Distance: "); Serial.println(distance);
Delay(100);
If(distance<=15 && value > 950)
{
Delay(500); Myservo.write(8); Delay(1000); Myservo.write(82); Delay(1000);
}
ensor1(); Sensor2();
If (Millis() – lastPublishTime > publishInterval) {
If (distance1Feed.publish(distance1)&&distance2Feed.publish(distance2)) { Serial.println("Published distance1 to
```

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⸎ Peer-reviewed & Refereed journal ⸎ Vol. 14, Issue 2, February 2025
DOI: 10.17148/IJARCCE.2025.14217

```
Adafruit IO");

lastPublishTime = Millis(); // Update the last publish time
}
}
}
////////////////////////////////////////////////////////// Void sensor1(void)
{
digitalWrite(trigPin1, LOW); delayMicroseconds(2); digitalWrite(trigPin1, HIGH); delayMicroseconds(10);
digitalWrite(trigPin1, LOW);

duration1 = pulseIn(echoPin1, HIGH); distance1 = (duration1*.0343)/2; Serial.print("Distance1: ");
Serial.println(distance1);
Delay(100); If(distance1<=10)
{
digitalWrite(led,LOW); delay(1000);
++cnt; Sendsms_wet();
}
Else
{
digitalWrite(led ,HIGH); cnt=0;
}
}
//////////////////////////////////////////////////////////////////////
Void sensor2(void)
{
digitalWrite(trigPin2, LOW); delayMicroseconds(2); digitalWrite(trigPin2, HIGH); delayMicroseconds(10);
digitalWrite(trigPin2, LOW);
duration2 = pulseIn(echoPin2, HIGH); distance2 = (duration2*.0343)/2; Serial.print("Distance2: ");
Serial.println(distance2);
Delay(100); If(distance2<=10)
{
digitalWrite(led1,LOW); delay(300);
++cnt1; Sendsms_dry();
}
Else
{
digitalWrite(led1,HIGH); cnt1=0;
}
}
///////////////////////////////////////////////////////////////////////////////////////

Void sendsms_wet(void)
{
If(cnt1==1)
{
Bot.sendMessage(CHAT_ID, "Wet Dustbin Full", "");
}
}
/////////////////////////////////////////////////////////// Void sendsms_dry(void)
{
If(cnt==1)
{
Bot.sendMessage(CHAT_ID, "Dry Dustbin Full", "");
}
}

Void MQTT_connect() { Int8_t ret;
```

ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 2, February 2025

DOI: 10.17148/IJARCCE.2025.14217

```
If (mqtt.connect()) { Return;
}

Serial.print("Connecting to Adafruit IO... "); While ((ret = mqtt.connect()) != 0) {
Serial.println(mqtt.connectErrorString(ret)); Mqtt.disconnect();
Delay(5000); // wait 5 seconds and retry
}
Serial.println("Adafruit IO Connected!");
}
```

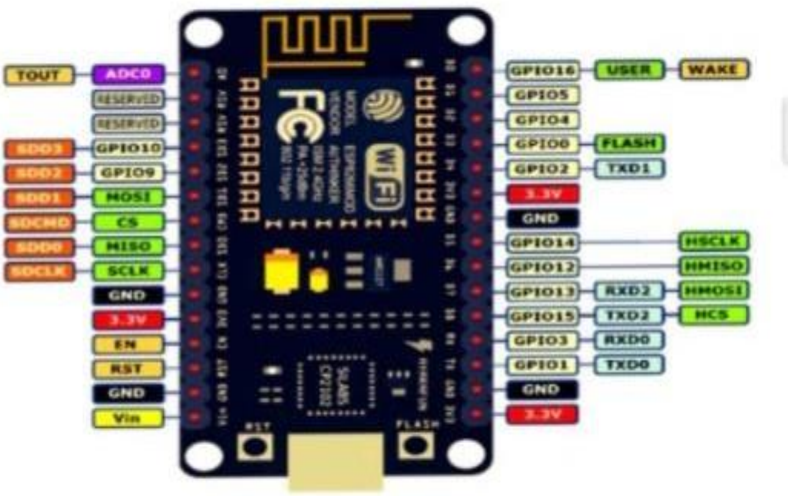


Figure-1 NODE MCU

VII. FUTURE SCOPE

Advanced Segregation: Incorporate machine learning algorithms to improve waste segregation accuracy.

Air Quality Monitoring: Add air quality sensors to monitor the environment around garbage bins.

Location Tracking: Use GPS modules to track the location of garbage bins and optimize collection routes.

User Interface: Develop a user-friendly app for real-time monitoring and control of the system.

Scalability: Expand the system to cover larger areas and integrate with municipal waste management systems.

REFERENCES

[1]. IoT -Based Smart Garbage Monitoring System S.S.lyenger et al.(2020)-This paper proposes an IoT-based system for monitoring garbage levels and detecting waste type.

[2]. Smart waste mangement using IOT and machine learning by A.Kumar et al.(2019)-This paper explores the application of machine learning algorithms in waste management.

[3]. IOT -based Garbage Monitoring System using Sensory Rajesshri k , Tangied, Dr. Mrs.A.U.Jawadekar- This paper provides better databases for garbage collection and quantity collection at every place.

[4]. IOT based garbage monitoring using arduino by Maksim Anwar , Pratik Sarkar , Rajenshwar Dutta , Md. Sadik Mohammed Mollick (2018)

[5]. Smart Waste management" by S.K.Ghosh(2019)-This paper provides overviews of smart waste management an

