**INTRODUCTION**

* **ABSTRACT**

This project is itended for construction of a **smart garbage management** device. The project is based on **NODE MCU** and for the waste detection we are using 2 types of sensors

* 1. **Ultra Sonic Sensor** : This sensor is used to identify the depth of the dustbin once it has fulfilled the required balance of the waste then it sends the signal to the **IR sensor.**
  2. **IR Sensor**: This sensor is used to rectify the signals which we are send or recive by the sensor.

• **About NODE MCU**

NodeMCU is an open-source [LUA](https://www.lua.org/start.html) based firmware developed for the ESP8266 wifi chip. By exploring functionality with the ESP8266 chip, NodeMCU firmware comes with the ESP8266 Development board/kit i.e. NodeMCU Development board.



* **NodeMCU ESP 8266 Technical Specifications and features:**

* **Wireless Module ESP8266:**With wireless connectivity for Internet of Things applications, the ESP8266 Wi-Fi module is the brains behind NodeMCU.

* **Unit of Microcontroller (MCU):**A 32-bit Tensilica L106 microprocessor is part of the ESP-12E/F module.

* **Speicher Flash:** The firmware and user programs are stored in the 4MB of flash memory that Node MCU normally ships with.

* **GPIO Pins**: General Purpose Input/Output Many GPIO pins on the NodeMCU can be utilized for a variety of purposes, including digital input/output, analog input, PWM (Pulse Width Modulation), I2C, SPI, and more.

* **ADC, or analog-to-digital converter**: It has an ADC that enables the board to read values from analog sensors.

* **USB to TTL Convertor**: The USB-TTL converter that Node MCU incorporates makes it simple to connect the board to a computer for programming and troubleshooting. USB connector included in, There is a on the Node MCU board.

* **Firmware for Node MCU**: For the ESP8266, the Node MCU firmware offers a straightforward and user-friendly programming interface.

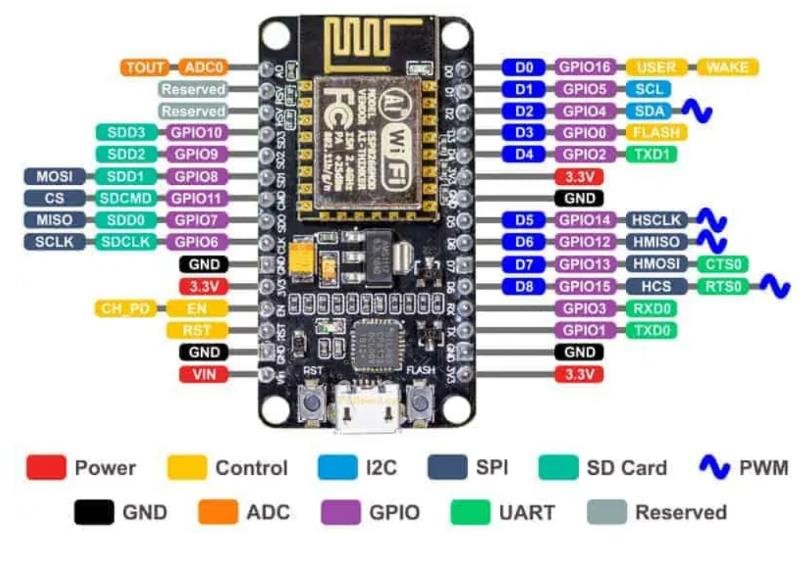
* **IDE Compatibility for Arduino**: The Arduino IDE may be used to program NodeMCU, making it available to a wide range of Arduino developers.

* **Open-Source Hardware**: The design files for Node MCU, an open-source hardware platform, are accessible to everyone.

* **Community Assistance**: With Node MCU, there is a sizable and vibrant developer and maker community that offers assistance, guides, and a plethora of resources.

* **Low Price:** Because Node MCU boards are typically inexpensive, hobbyists and developers on a tight budget often choose them.

• **NodeMCU ESP8266 Pinout and Functions Explained**



**Power**: The NodeMCU development board is a popular platform for IoT (Internet of Things) projects, and it is based on the ESP8266 WiFi module. Usually, the NodeMCU board has multiple pins for connections pertaining to power.

**Control**: The NodeMCU ESP8266 is a popular development board that uses the ESP8266 WiFi module. It is commonly used for IoT (Internet of Things) projects. To control a pin on the NodeMCU ESP8266, you typically use the Arduino IDE and write code in C++.

**12C**: The following are the default I2C pins on the NodeMCU ESP8266 boardI2C devices are frequently connected to the ESP8266 via these pins. Make sure the SDA and SCL pins on your I2C device are connected correctly if you are using it. For I2C communication, don't forget to include the Wire library in your Arduino sketch.

**SPI**: The ESP8266 microcontroller, including the NodeMCU development board, supports SPI (Serial Peripheral Interface) for communication with other devices. SPI is a synchronous serial communication protocol commonly used to communicate with sensors, displays, and other peripherals.

**SD Card**: An SD card module typically includes an SD card slot and a microcontroller that communicates with the ESP8266 through SPI (Serial Peripheral Interface) communication. This module allows you to read and write data to an SD card, providing a convenient way to add storage to your ESP8266 projects.

**PWM:** The ESP8266, including the NodeMCU development board, supports PWM (Pulse Width Modulation), which allows you to simulate analog output by varying the duty cycle of a digital signal. On the ESP8266, you can use the **analogWrite()** function to generate PWM signals on certain pins.

**GND:** The Ground (GND) pins on the ESP8266, including the NodeMCU development board, are the pins that provide the reference voltage for the board and complete electrical circuits. Each of the GPIO (General Purpose Input/Output) pins on the ESP8266 has a corresponding GND pin for ground connection.

**ADC**: The ESP8266, including the NodeMCU development board, has an Analogto-Digital Converter (ADC) that allows you to read analog signals. The ESP8266 ADC is capable of reading analog voltages in the range of 0 to 1.0V.

**GPIO**: The ESP8266, including the NodeMCU development board, has GeneralPurpose Input/Output (GPIO) pins that you can use for various purposes such as digital input or output, PWM (Pulse Width Modulation), and more. GPIO pins are versatile and can be configured for different functionalities in your ESP8266 projects.

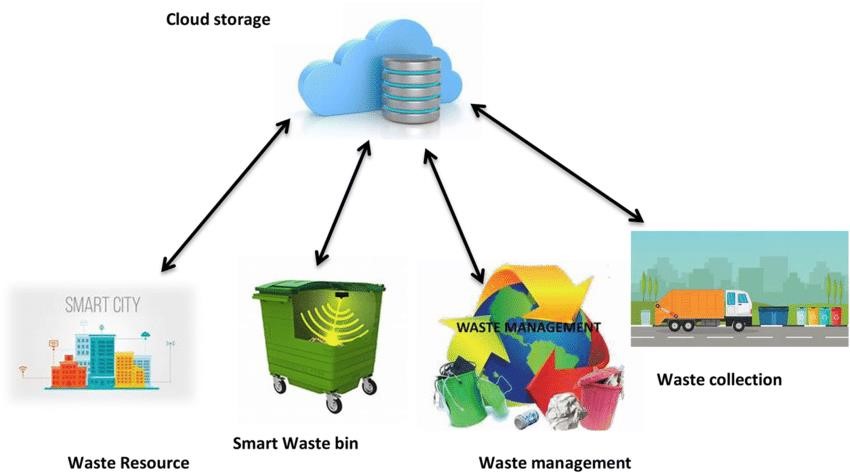
**UART**: The ESP8266, including the NodeMCU development board, has UART (Universal Asynchronous Receiver/Transmitter) functionality, allowing communication over serial ports. The ESP8266 typically has two hardware

UARTs: UART0 and UART1. UART0 is commonly used for programming and debugging, while UART1 is often available for general-purpose communication.

**Reserved**: the NodeMCU development board, typically refers to certain pins or features that have specific functions or limitations imposed by the hardware design or the firmware. These reserved pins or features may have special purposes or characteristics that differentiate them from general-purpose GPIO (GeneralPurpose Input/Output) pins.

* **Why this project**:

It would open a lot opportunities of using it. It will be used for cleaning the area sequentially. Beside this, as the dustbin is managed by the user depends on the live monitoring system. If the dustbin get filled we got an notification on the app.



# **Objective**

The objectives of a Smart Waste Management System project using IoT (Internet of Things) typically revolve around improving efficiency, reducing costs, and enhancing sustainability in waste management processes. Here are common objectives for such a project:

**1.Real-Time Monitoring:** Implementing sensors in waste bins to monitor the fill level in real-time. This helps optimize waste collection routes and schedules, reducing unnecessary pickups and associated costs.

**2.Optimized Collection Routes:** Analyzing the data collected from sensors to optimize waste collection routes. This reduces fuel consumption, minimizes traffic congestion, and lowers carbon emissions.

**3.Cost Reduction:** By optimizing collection routes and schedules, the project aims to reduce operational costs associated with fuel, labor, and vehicle maintenance. This can result in significant cost savings for waste management services.

**4.Prevent Overflow and Littering:** Timely notifications and alerts triggered by sensor data can help prevent overflowing waste bins, reducing the likelihood of littering. This contributes to a cleaner environment and improved public health.

**5.Resource Efficiency:** Efficient waste management leads to better resource allocation. By collecting waste only when necessary, resources such as fuel and manpower are used more efficiently.

**6.Data-Driven Decision Making:** Gathering and analyzing data from the IoT devices allows waste management authorities to make informed decisions based on trends, patterns, and historical data. This data-driven approach enhances overall decision-making processes.

**7.Remote Monitoring and Management:** Providing the capability to remotely monitor and manage waste bins and collection processes. This is particularly useful for large-scale waste management systems where physical inspection of every bin is not practical.

**8.Environmental Sustainability:** Contributing to environmental sustainability by reducing the carbon footprint associated with waste collection. Optimizing routes and schedules minimizes the impact of waste management activities on the environment.

**9.Public Awareness and Engagement:** Integrating the system with communication channels to keep the public informed about waste collection schedules, environmental impact, and ways to participate in waste reduction

**10.efforts**. This promotes public awareness and engagement in sustainable practices.

**11.Scalability:** Designing the system to be scalable, allowing it to accommodate the growing needs of urban areas and adapt to changes in waste generation patterns over time.

**12.integration with Smart City Initiatives:** Aligning the Smart Waste

Management System with broader Smart City initiatives to create a cohesive urban infrastructure that leverages technology for sustainable, efficient, and intelligent urban living.

By addressing these objectives, a Smart Waste Management System using IoT can contribute to more sustainable and efficient waste management practices in urban areas.

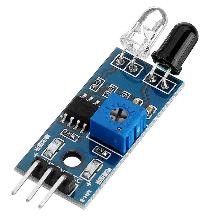
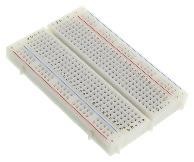


# **Components**

There are basic components of our project :

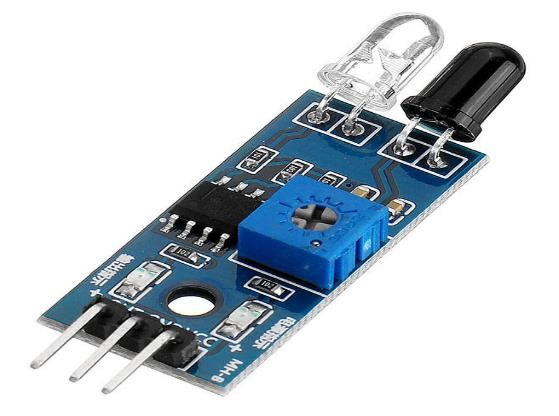
**Hardware Software**

1. Node MCU ESP8266 1. MIT app inventor
2. Breathboard 2. Arduino IDE
3. IR Sensor 3. Google FireBase
4. Jumper Wires
5. UltraSonic Sensor
6. Servo Motor
7. Dustbin



* **IR Sensor**

This model, Smart Waste Management System (SWM) effectively employs IR sensor to identify dry waste items, and capacitive soil moisture sensor along with IR sensor to differentiate between dry and wet wastes.



* **Servo Motor**

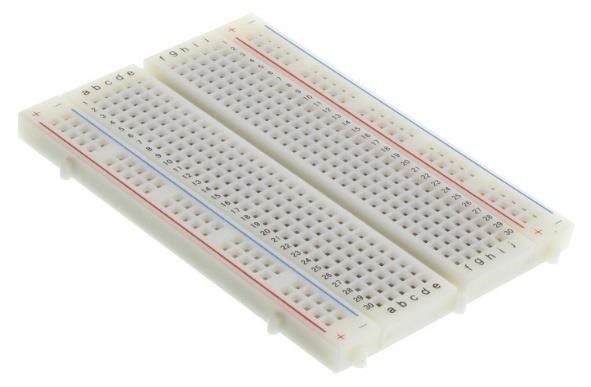
The servo motor is used to open and close the dustbin automatically, providing a more convenient and hygienic way of disposing of garbage. The system is connected to the internet using Node mcu and is controlled remotely using the Webserver Platform.



* **UltraSonic Sensor**

Ultrasonic sensor is a device that uses sound waves beyond the range of human hearing (ultrasonic waves) to measure distance, detect objects, or create images of surroundings. It typically consists of a transmitter that emits ultrasonic pulses and a receiver that detects the reflected waves, allowing the sensor to calculate distances based on the time taken for the waves to travel to an object and back. Ultrasonic sensors are commonly used in various applications, including proximity sensing, object detection, and navigation systems.

* **Bread Board:**

A breadboard is a prototyping tool used in electronics to build and test circuits without soldering. It allows for quick and temporary connections of electronic components, facilitating experimentation, testing, and the development of prototypes. Breadboards are widely used by hobbyists, students, and engineers for designing and troubleshooting circuits before finalizing them on a more permanent platform.

* **Connection of ESP-8266 and IR Sensor**

|  |  |
| --- | --- |
| **Pin on ESP 8266 Module** | **Pins on IR sensor** |
| D3 | OUT |
| G | GND |
| 3V | VCC |

* **Connection of ESP-8266 and Ultra sonic sensor**

|  |  |
| --- | --- |
| **Pin on ESP 8266 Module** | **Pin on Ultra Sonic Sensor** |
| VIN | VCC |
| G | GND |
| D1 | TRIG |
| D2 | ECHO |

**BASIC BLOCK DIAGRAM**

