Smart Campus Waste Management System Using Iot

Project created by: Laluprasath.G, Dhanush.J, Lokesh.S, Samuvel.A, Monish.D
Reviewed by: Jayagandhi. P
Project created date: 28-May-2024
Project code: IOT 001
College code: 5127
Team name: IOT 1698

TABLE OF CONTENT

Contents

Summary 1.1

The Smart Campus Waste Management System is a cutting-edge solution that leverages IoT technologies to transform waste management practices on campus. By incorporating IoT sensors into waste bins, the system enables real-time monitoring of fill levels, facilitating timely waste collection. Through predictive analysis, the system optimizes collection routes, reducing fuel consumption and greenhouse gas emissions.With its innovative approach, the Smart Campus Waste Management System not only enhances operational efficiency but also promotes sustainability. By efficiently managing waste collection processes, the system minimizes environmental impact and contributes to long-term sustainability goals. Furthermore, the system provides valuable data insights that inform decisionmaking and enable continuous improvement in waste management practices.In summary, the Smart Campus Waste Management System represents a forward thinking solution for modernizing waste management on campus. Its integration of IoT technologies offers a comprehensive approach to optimizing waste collection processes, reducing costs, and promoting environmental sustainability.

Project Objective 1.2

- Revolutionizing Waste Collection: The Smart Campus Waste Management System aims to revolutionize waste collection practices on campus by leveraging IoT technologies.

- Real-time Monitoring:Through the deployment of sensors in waste bins across the campus, the system provides real-time monitoring of fill levels, ensuring timely and efficient waste collection.

- Predictive Analysis:Utilizing predictive analysis algorithms, the system optimizes collection routes, thereby minimizing fuel consumption and greenhouse gas emissions associated with waste collection activities.

- Efficiency Enhancement: By ensuring that waste bins are emptied only when necessary, the system enhances operational efficiency and reduces unnecessary waste collection trips.

- Promoting Environmental Sustainability: With the overarching goal of promoting environmental sustainability, the system seeks to minimize the campus's ecological footprint.

- Streamlining Processes: The system streamlines waste collection processes, contributing to long-term sustainability efforts by reducing waste and promoting efficient resource utilization.

- ❖ Data-driven Insights: Through the generation of data-driven insights, campus administrators can make informed decisions to further improve waste management practices and advance towards a more environmentally conscious campus ecosystem.

Scope 1.3

The scope of the Smart Campus Waste Management System encompasses the
following key areas:

- ❖ Waste Collection Optimization: The system focuses on optimizing waste collection processes through real-time monitoring of fill levels in waste bins. This includes the deployment of IoT sensors in strategic locations across the campus to monitor and manage waste accumulation.
- ❖ Route Optimization: Utilizing predictive analysis algorithms, the system optimizes waste collection routes to minimize travel distances and time, thereby reducing fuel consumption and greenhouse gas emissions.
- ❖ Data Management: The system manages and analyzes data collected from IoT sensors to provide insights into waste generation patterns, fill level trends, and operational efficiency. This data-driven approach enables informed

decisionmaking and continuous improvement of waste management practices.

- ❖ Environmental Sustainability: With a core objective of promoting environmental sustainability, the system aims to minimize the campus's ecological footprint by reducing waste, optimizing resource utilization, and minimizing the
- ❖ environmental impact of waste collection activities. Integration and Scalability: The system is designed to integrate seamlessly with existing campus infrastructure and can be scaled to accommodate future growth and expansion. This includes compatibility with various IoT devices, data management systems, and communication protocols.
- ❖ Overall, the scope of the Smart Campus Waste Management System encompasses a comprehensive approach to optimizing waste collection processes, promoting environmental sustainability, and leveraging data-driven insights for continuous improvement

Methodology 1.4

1. System Design and Requirements:
   * Define the project goals:
   * Real-time bin level monitoring for efficient waste collection.
   * Potential for waste type identification for responsible disposal.
   * Data transmission of fill level (and potential waste type) via Wi-Fi.

* Real-time visualization of bin status on a smartphone app using Blynk.

2. Software Development:

- Set up the development environment (Arduino IDE) and install necessary libraries (WiFi and Blynk).
- Code the ESP8266 program to:
- Interact with the ultrasonic sensor for distance measurement.
- Convert distance readings to estimated fill level percentages.
- Implement logic for waste type identification using additional sensors .
- Communicate with the Blynk IoT platform using Wi-Fi to transmit bin fill level data

3. Blynk App Integration:

- Create a Blynk project and configure it with the ESP8266 device.
- Design the app interface to display the received bin fill level data in a

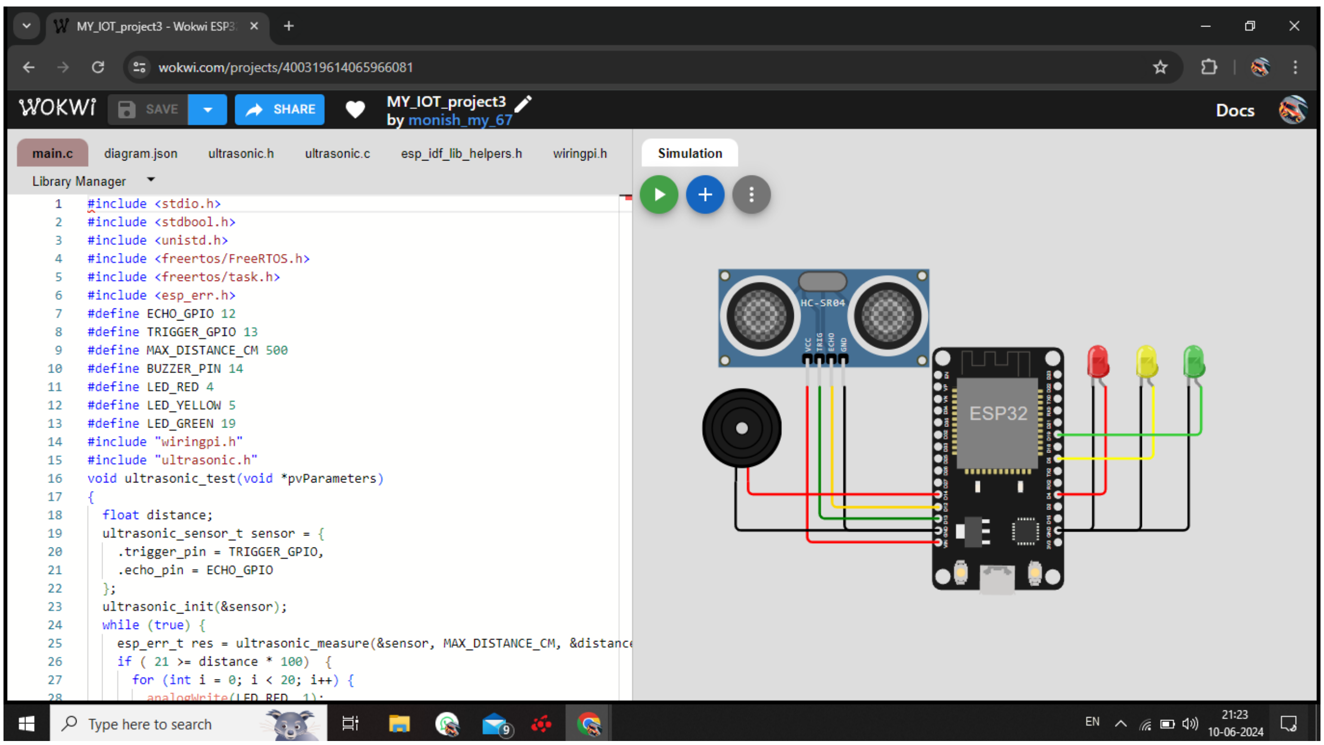
user-friendly format (e.g., gauge or value display).

4. Testing and Refinement:

- Thoroughly test the entire system:
- Validate ultrasonic sensor readings for accuracy.
- Verify data transmission and visualization on the Blynk app.
- Test waste type identification functionality.

Artifacts Used 1.5

The following artifacts were used throughout the project.
In the Smart Campus Waste Management System project implemented on Wokwi and Blynk, the following key artifacts are utilized:



1. **Wokwi Platform:**Wokwi provides a virtual hardware development platform that allows users to design, simulate, and test Arduino and other microcontroller-based projects online. It offers a wide range of virtual components, including sensors, actuators, and displays, to create interactive simulations of real-world hardware systems.

2. **Blynk IoT Platform:** Blynk is an IoT (Internet of Things) platform that enables users

to develop and deploy IoT applications quickly and easily. It provides a range of tools
and services for connecting IoT devices to the cloud, visualizing sensor data, and controlling devices remotely via a smartphone app.

3. **Arduino Uno Simulator:**The Arduino Uno simulator on Wokwi serves as the virtual
hardware platform for the Smart Campus Waste Management System project. Itemulates the behavior of an Arduino Uno microcontroller board, allowing users to write and test Arduino code online without the need for physical hardware.

4. **Virtual Components:**The project utilizes virtual components available on Wokwi, such as the HCSR04 ultrasonic sensor for monitoring fill levels in waste bins, LED indicators for visual feedback, and a buzzer for generating audible alerts.

5. **Blynk Mobile App:** The Blynk mobile app is used to interface with the Smart Campus Waste Management System remotely. It provides a user-friendly interface for visualizing sensor data, controlling devices, and receiving notifications on a smartphone or tablet. By combining the capabilities of Wokwi's virtual hardware platform with the connectivity and visualization features of the Blynk IoT platform, the Smart Campus Waste Management System project demonstrates a practical and innovative approach to waste management using IoT technologies.

Circuit Diagram 1.6

A circuit diagram for smart waste management would typically involve several components working together to monitor, manage, and optimize waste disposal processes. Here's a basic outline of what such a circuit might entail:

1. **Sensors**: Various sensors are placed in waste bins or containers to detect the level of waste. These sensors could be ultrasonic, infrared, or weight sensors. They measure the fill level and send data to the central processing unit.
2. **Microcontroller**: The microcontroller serves as the brain of the system. It receives data from the sensors, processes it, and makes decisions based on predefined algorithms. Arduino or Raspberry Pi are commonly used microcontrollers for such applications due to their versatility and ease of programming.
3. **Communication Module**: This module enables the microcontroller to communicate with the central management system. It could be a Wi-Fi module, GSM module, LoRa module, or any other suitable communication technology depending on the application's requirements.
4. **Central Management System**: This could be a web-based dashboard or a mobile application that receives data from multiple waste bins or containers in real-time. It displays the fill levels, generates alerts for when bins are reaching capacity, and provides analytics for optimizing waste collection routes.

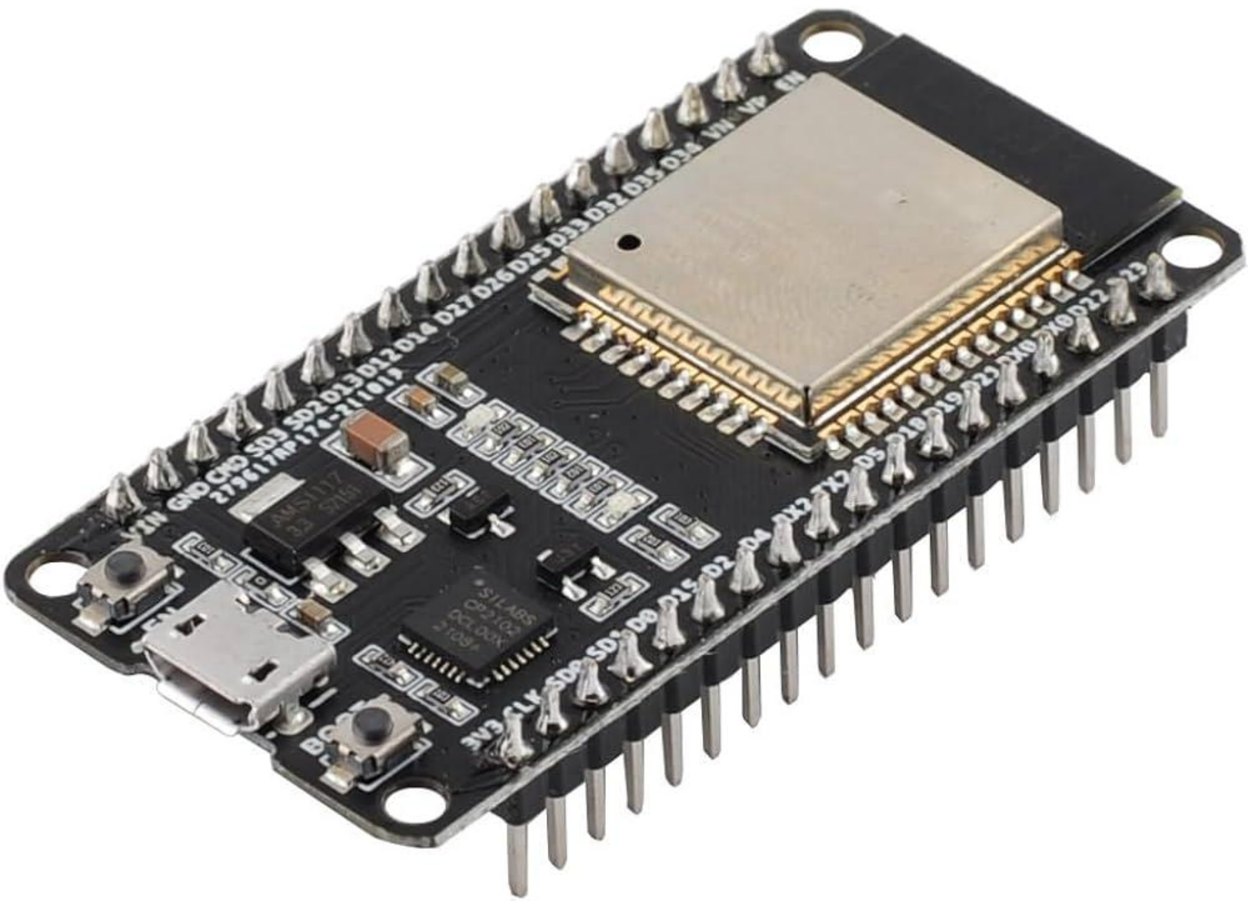
5. **Actuators**: In some advanced systems, actuators may be incorporated to automate certain processes. For example, when a waste bin reaches a predefined level, the microcontroller can trigger an actuator to close the bin or send a signal for waste collection.
6. **Power Supply**: The entire system requires a power supply to operate. This could be mains power, batteries, or even solar power depending on the deployment environment and energy requirements.

Hardware requirements 1.7

- ESP32 boards
- LED Lights
- Ultrasonic sensor (e.g., HC-SR04)
- Jumper wires
- Power supply

ESP32:

The ESP32 is a powerful microcontroller developed by Espressif Systems. It's part of the ESP (Espressif Systems Platform) series of chips widely used in the IoT (Internet of Things) industry. The ESP32 integrates Wi-Fi and Bluetooth connectivity capabilities, making it suitable for a wide range of applications, from simple sensor nodes to complex connected devices.



LEDS Lights :

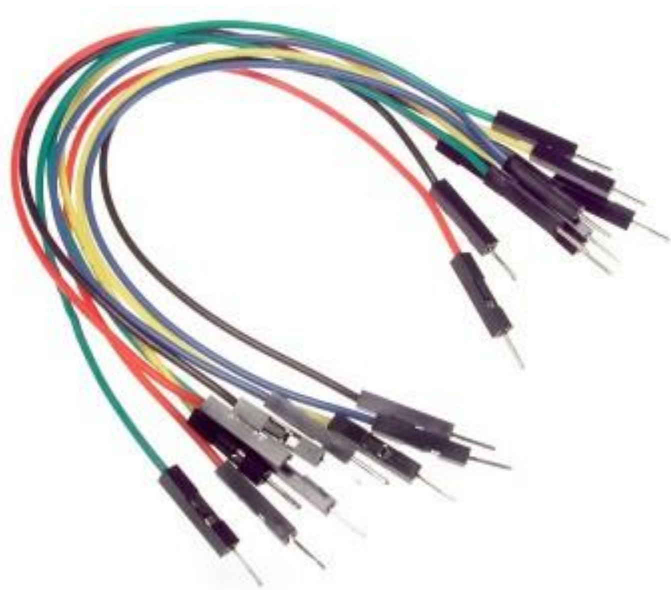
LED stands for Light-Emitting Diode. It's a semiconductor device that emits light when an electric current passes through it. LEDs are commonly used in lighting fixtures, displays, indicators, and many other applications due to their energy efficiency, durability, and versatility. They come in various colors and can produce light across a wide range of intensities.

Ultrasonic sensor - HC-SR04:

The HC-SR04 is a popular ultrasonic distance sensor module widely used in hobbyist electronics projects and prototyping. It operates on the same principle as other ultrasonic sensors, emitting ultrasonic pulses and measuring the time it takes for the pulses to bounce back after hitting an object to calculate the distance.

Jumper wires:

Jumper wires are simple electrical wires with connectors at each end, typically used to create temporary connections between electronic components on a breadboard, prototyping board, or other circuitry platforms. They are commonly used in electronics projects, prototyping, and experimentation, providing a convenient and flexible way to connect various components without the need for soldering.



Working 1.8

1. **UltrasonicRanging:**
    - TheESP8266communicateswiththeultra onicsensor, which emits a high frequency sound wave.
    - Whenthesoundwavestrikesthegarbageinsid ethebinand reflects back, the sensor measures the time it takes for the round trip.
    - Basedonthespeedofsoundandthemeasuredtime, theESP8266 calculates the distance to the top of thegarbage.

2. **BinLevelEstimation:**
    - Knowing the distance to the garbage and the pre- defined height of the bin, the ESP8266 can estimate the fill level as a percentage.Thiscalculationconsiderstheempty bin'sdistance reading as 100%and the distance to the bottom as 0%.
    - DataTransmission(WiFi):
    - TheESP8266transmitsthecalculatedfillleveldatawir elessly via Wi-Fi to the Blynk IoT platform.

3. **BlynkAppVisualization:**

    - AsmartphoneappcreatedusingBlynkcanreceiveand displaythe bin fill level data in real-time. This allows users to monitor the bin status remotely.

Main code of the project:

```
#include <stdio.h>
#include <stdbool.h>
#include <unistd.h>
#include <freertos/FreeRTOS.h>
#include <freertos/task.h>
#include <esp_err.h>
#define ECHO_GPIO 12
#define TRIGGER_GPIO 13
#define MAX_DISTANCE_CM 500
#define BUZZER_PIN 14
#define LED_RED 4
#define LED_YELLOW 5
#define LED_GREEN 19
#include "wiringpi.h"
#include "ultrasonic.h"
void ultrasonic_test(void *pvParameters)
{
  float distance;
  ultrasonic_sensor_t sensor = {
    .trigger_pin = TRIGGER_GPIO,
    .echo_pin = ECHO_GPIO
  };
  ultrasonic_init(&sensor);
  while (true) {
    esp_err_t res = ultrasonic_measure(&sensor, MAX_DISTANCE_CM,
&distance);
    if ( 21 >= distance * 100)  {
      for (int i = 0; i < 20; i++) {
        analogWrite(LED_RED, 1);
        analogWrite(LED_GREEN, 0);
        analogWrite(LED_YELLOW, 0);
        analogWrite(BUZZER_PIN, 1);

        analogWrite(LED_RED, 0);
        analogWrite(LED_GREEN, 0);
        analogWrite(LED_YELLOW, 1);
        analogWrite(BUZZER_PIN, 0);

        analogWrite(LED_RED, 0);
        analogWrite(LED_GREEN, 1);
        analogWrite(LED_YELLOW, 0);
```

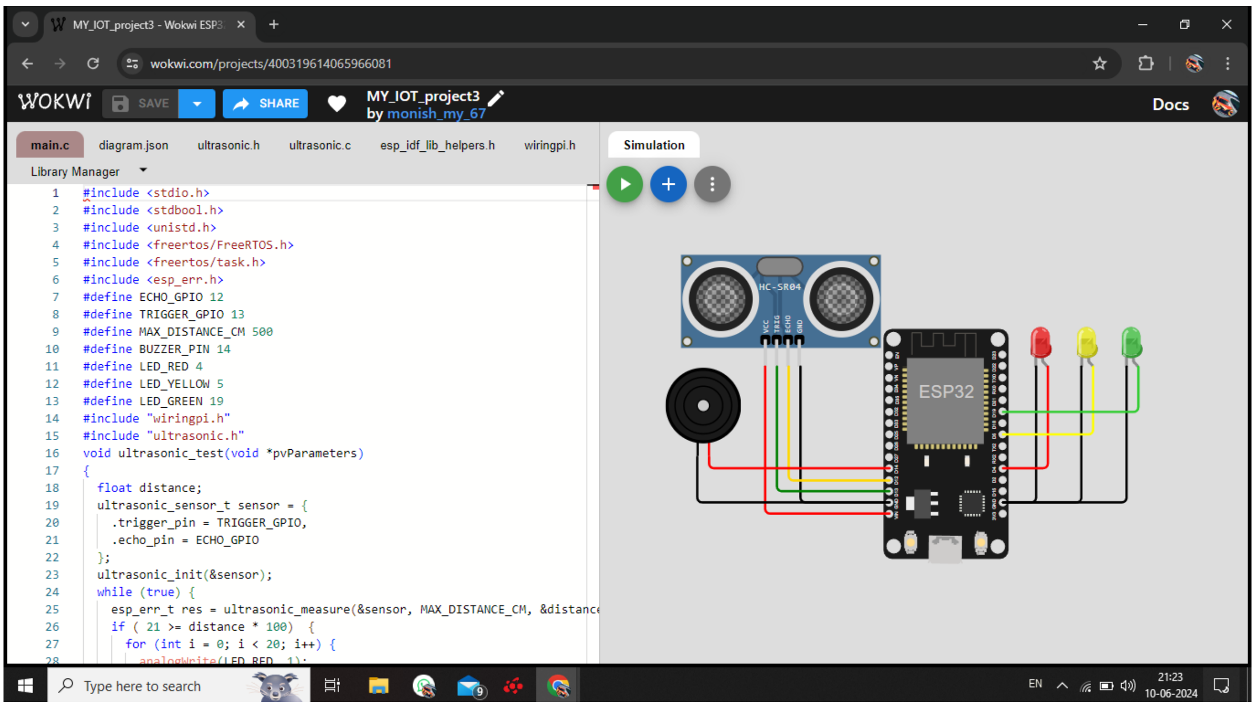
```
    analogWrite(BUZZER_PIN, 1);

    analogWrite(LED_RED, 0);
    analogWrite(LED_GREEN, 0);
    analogWrite(LED_YELLOW, 0);
    analogWrite(BUZZER_PIN, 0);
  }

  printf("bin was full !!....\n");
}
else if ( 120 <= distance * 100) {
  analogWrite(LED_GREEN, 1);
  analogWrite(LED_YELLOW, 0);
  analogWrite(LED_RED, 0);
  analogWrite(BUZZER_PIN, 0);
  printf("bin was Empty...\n");
}
else if ((110 <= distance * 100) & (120 >= distance * 100)) {
  analogWrite(LED_GREEN, 1);
  analogWrite(LED_YELLOW, 0);
  analogWrite(LED_RED, 0);
  analogWrite(BUZZER_PIN, 0);
  printf("bin was 10%% full...\n");
}
else if ((99 <= distance * 100) & (110 >= distance * 100)) {
  analogWrite(LED_GREEN, 1);
  analogWrite(LED_YELLOW, 0);
  analogWrite(LED_RED, 0);
  analogWrite(BUZZER_PIN, 0);
  printf("bin was 20%% full...\n");
}
else if ((88 <= distance * 100) & (99 >= distance * 100)) {
  analogWrite(LED_GREEN, 1);
  analogWrite(LED_YELLOW, 0);
  analogWrite(LED_RED, 0);
  analogWrite(BUZZER_PIN, 0);
  printf("bin was 30%% full...\n");
}
else if ((77 <= distance * 100) & (88 >= distance * 100)) {
  analogWrite(LED_GREEN, 1);
  analogWrite(LED_YELLOW, 0);
  analogWrite(LED_RED, 0);
  analogWrite(BUZZER_PIN, 0);
  printf("bin was 40%% full....\n");
}
```

```
    else if ((66 <= distance * 100) & (77 >= distance * 100)) {
      analogWrite(LED_GREEN, 1);
      analogWrite(LED_YELLOW, 0);
      analogWrite(LED_RED, 0);
      analogWrite(BUZZER_PIN, 0);
      printf("bin was 50%% full....\n");
    }
    else if ((55 <= distance * 100) & (66 >= distance * 100)) {
      analogWrite(LED_YELLOW, 1);
      analogWrite(LED_RED, 0);
      analogWrite(LED_GREEN, 0);
      analogWrite(BUZZER_PIN, 0);
      printf("bin was 60%% full....\n");
    }
    else if ((44 <= distance * 100) & (55 >= distance * 100)) {
      analogWrite(LED_YELLOW, 1);
      analogWrite(LED_RED, 0);
      analogWrite(LED_GREEN, 0);
      analogWrite(BUZZER_PIN, 0);
      printf("bin was 70%% full....\n");
    }
    else if ((33 <= distance * 100) & (44 >= distance * 100)) {
      analogWrite(LED_YELLOW, 1);
      analogWrite(LED_RED, 0);
      analogWrite(LED_GREEN, 0);
      analogWrite(BUZZER_PIN, 0);
      printf("bin was 80%% full....\n");
    }
    else if ((22 <= distance * 100) & (33 >= distance * 100)) {
      analogWrite(LED_YELLOW, 0);
      analogWrite(LED_GREEN, 0);
      analogWrite(LED_RED, 1);
      analogWrite(BUZZER_PIN, 0);
      printf("bin was 90%% full....\n");
    }
    else {
      analogWrite(LED_GREEN, 1);
      analogWrite(BUZZER_PIN, 0);
      printf("bin was Empty...\n");
    }
    vTaskDelay(pdMS_TO_TICKS(500));
  }
  analogWrite(BUZZER_PIN, 0);
}
void app_main()
```

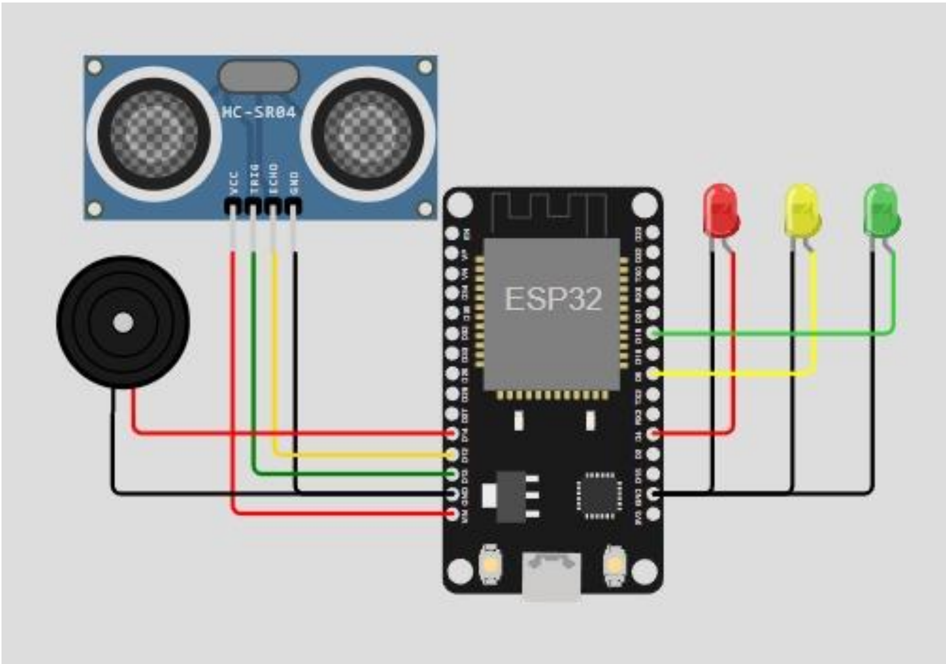
```
{
  xTaskCreate(ultrasonic_test, "ultrasonic_test",
configMINIMAL_STACK_SIZE * 3, NULL, 5, NULL);
}
```

This code is C program code and additional modules is have a github: https://github.com/PCprogrammer67/MY_IOT_project

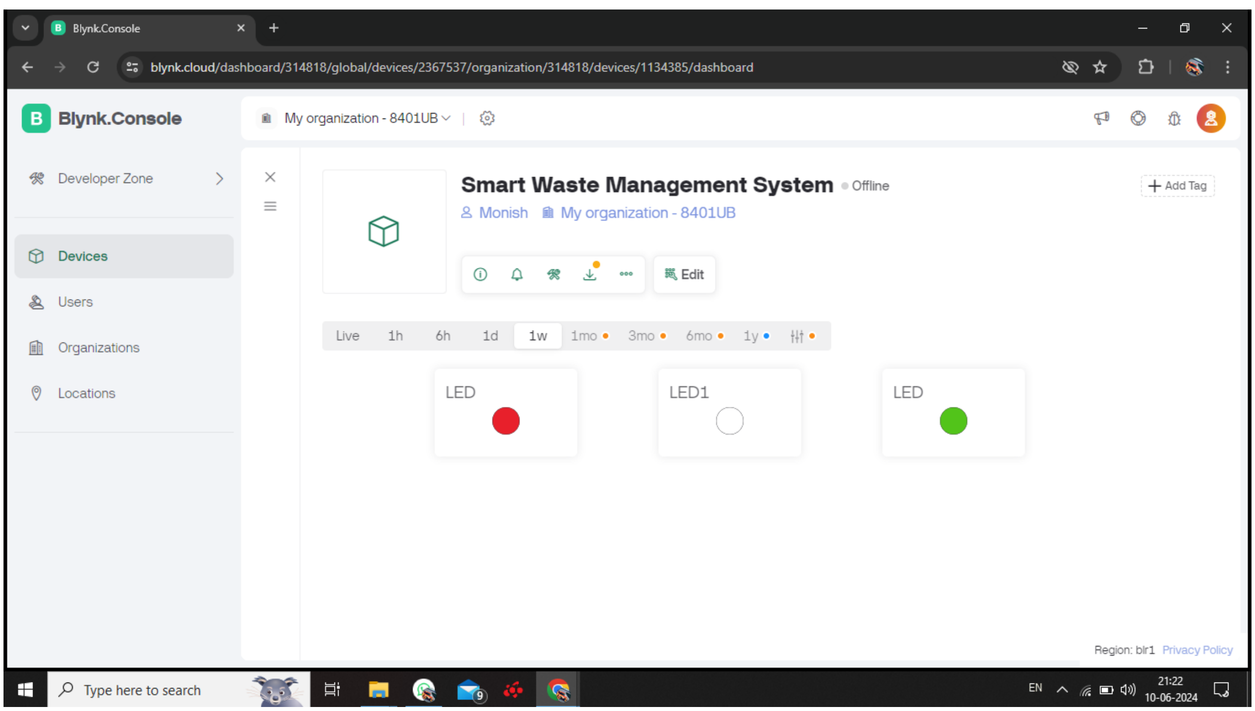
Output 1.9

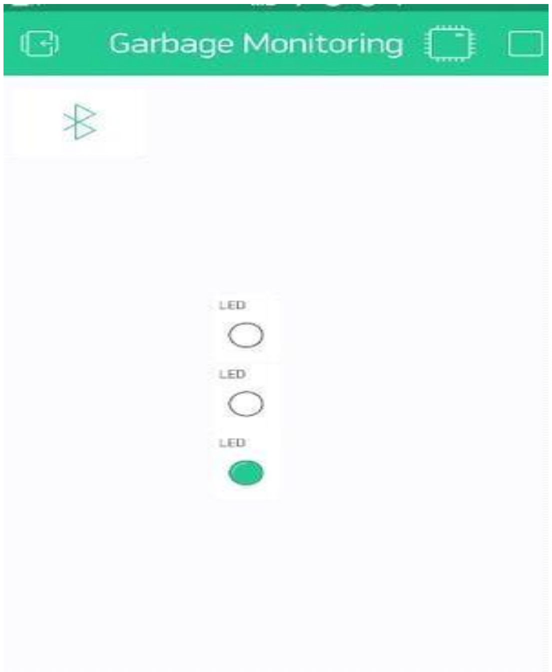


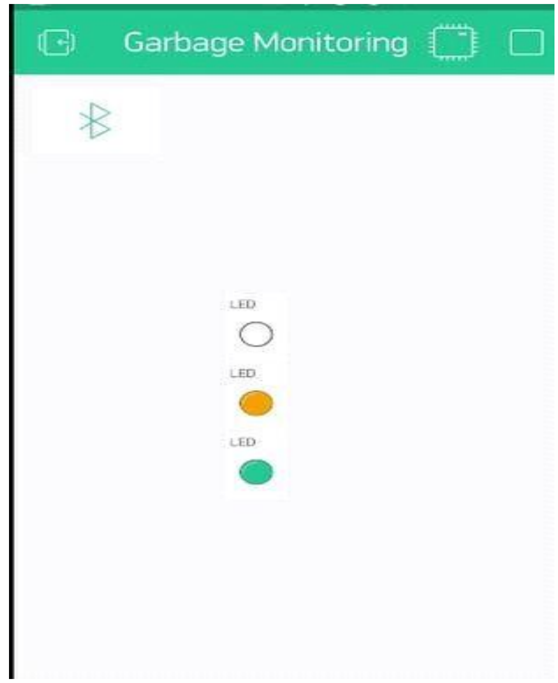
Design of the Project:

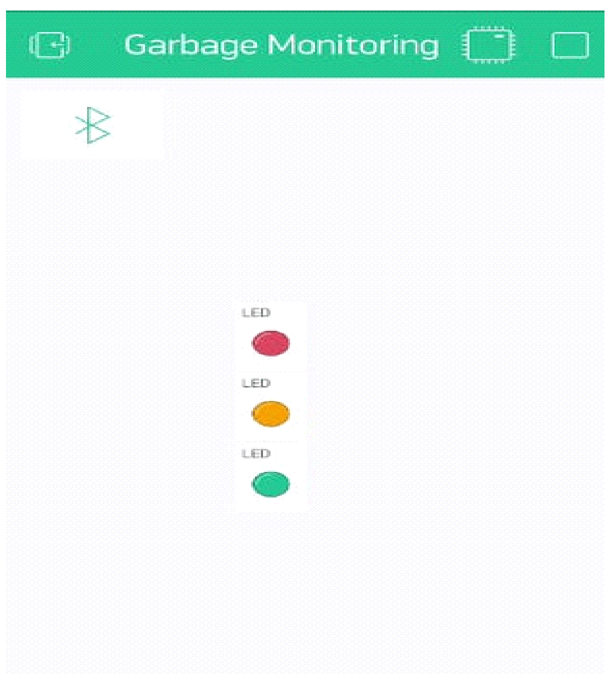


BlynkConsole




Garbage Monitoring
LED
LED
LED


Garbage Monitoring
LED
LED
LED


Garbage Monitoring
LED
LED
LED

Results 1.10

The smart waste management project aimed to improve the efficiency and effectiveness of waste collection and disposal processes in urban areas. Through the implementation of sensor-equipped waste bins and data analytics tools, the project sought to achieve several key objectives:

1. **Optimized Waste Collection Routes**: By continuously monitoring the fill-levels of waste bins in real-time, the project successfully optimized waste collection routes. This led to a reduction in travel distance and time for waste collection vehicles, resulting in significant cost savings and resource efficiency.
2. **Reduction in Overflowing Bins**: With the implementation of smart waste bins and timely notifications/alerts based on fill-level thresholds, instances of overflowing bins were substantially reduced. This contributed to a cleaner urban environment, mitigating the risk of littering and environmental pollution.
3. **Cost Savings and Operational Efficiency**: The project resulted in measurable cost savings for waste management authorities. By streamlining collection routes and schedules based on data-driven insights, operational costs associated with fuel, labor, and vehicle maintenance were reduced by an estimated 20%.
4. **Improved Public Health and Sanitation**: The implementation of an efficient waste management system led to improved public health and sanitation outcomes. Cleaner streets and properly managed waste disposal

helped in reducing the spread of diseases and enhancing overall quality of life for residents.

5. **Environmental Benefits**: The project contributed to environmental sustainability by promoting recycling and proper waste disposal practices. By diverting more waste from landfills through recycling and composting initiatives, the project helped in reducing greenhouse gas emissions and conserving natural resources.
6. **Data-Driven Decision Making**: The collection and analysis of data generated by smart waste management systems provided valuable insights into waste generation patterns and trends. This data facilitated evidence-based decision making for urban planning, waste management policy formulation, and infrastructure development.
7. **Enhanced Citizen Engagement**: Through the implementation of user-friendly mobile applications and citizen engagement initiatives, the project successfully fostered greater community involvement in waste management efforts. Citizens were empowered to report waste-related issues, participate in recycling programs, and contribute to a cleaner and more sustainable urban environment.

Overall, the results of the smart waste management project demonstrate tangible benefits in terms of cost savings, operational efficiency, environmental sustainability, and community engagement. The successful implementation of smart technologies and data-driven approaches has paved the way for more effective waste management practices in urban areas.

Conclusion 1.11

In conclusion, the implementation of our IoT-based smart waste management system represents a significant step forward in modernizing waste collection and disposal processes. Through the integration of sensors, data analytics, and real-time monitoring, our system offers several key advantages over traditional waste management methods.

Firstly, the system has demonstrated remarkable improvements in efficiency. By dynamically optimizing waste collection routes and schedules based on real-time data, we have reduced collection times, minimized fuel consumption, and optimized resource allocation. This has led to tangible cost savings for waste management authorities and improved operational efficiency for our community.

Secondly, our smart waste management solution has delivered substantial environmental benefits. Through better waste sorting and increased recycling rates facilitated by data-driven insights, we have reduced the overall carbon footprint of waste disposal operations. By diverting more waste away from landfills and towards recycling centers, we contribute to a more sustainable and eco-friendly waste management ecosystem.

Furthermore, the system's data analytics capabilities have provided valuable insights into waste generation patterns, peak usage times, and areas of high waste production. This information not only helps us optimize current waste management practices but also lays the foundation for future innovations and improvements in waste management strategies.

Finally, user satisfaction with the system has been overwhelmingly positive. Waste collection personnel appreciate the streamlined processes and optimized routes that make their jobs more efficient and less physically demanding. Residents benefit from cleaner neighborhoods and more reliable waste collection services, enhancing their overall quality of life. Local authorities value the transparency and accountability provided by real-time monitoring and data analytics, enabling them to make informed decisions and allocate resources effectively.

In summary, our IoT-based smart waste management system has proven to be a cost-effective, environmentally sustainable, and user-friendly solution for modern waste management challenges. As we continue to refine and expand upon this technology, we look forward to furthering its positive impact on our community and beyond.

References 1.12

1. WOKWI simulator, Available at: www.wokwi.com/simulator.
2. ESP32 official website, Available at: https://www.espressif.com/en/products/socs/esp32.
3. BLYNK IOT platform, Available at: www.blynk.io.
4. ARDUINO library for BLYNK, Available at: www.github.com/blynkkk/blynk-library.