Progress Report on IoT Embedded System Project

Name:	**Abdul Rahman**
Student Id:	**243EA245NH**

Project Title:

IoT Embedded System for Smart Waste Management

Objective:

The objective is to develop an IoT-based system prototype to address real-world challenges aligned with the United Nations' Sustainable Development Goals (SDGs). This project emphasizes remote monitoring and sustainable IoT design, contributing to better waste management through smart sensing and control.

Project Overview:

- ## Description:

  This project involves designing a smart waste management IoT system that integrates the **ESP32-WROOM module,** various sensors, and actuators for real-time data collection, monitoring, and control. It leverages IoT platforms for remote interaction and promotes sustainability through efficient waste monitoring.

Hardware Requirements:

- **Sensors:**

  - **MQ-135 Gas Sensor** (air quality monitoring).
  - **HC-SR04 Ultrasonic Sensor** (waste bin level monitoring).

- **Actuators**:

  - **Red LED** (full waste bin indicator).
  - **Green LED** (normal status indicator).
  - **Buzzer (critical** alert for high waste/gas levels).
  - **Servo Motor SG90 (**Automate waste bin lid opening or closing based on waste levels detected by the ultrasonic sensor.)

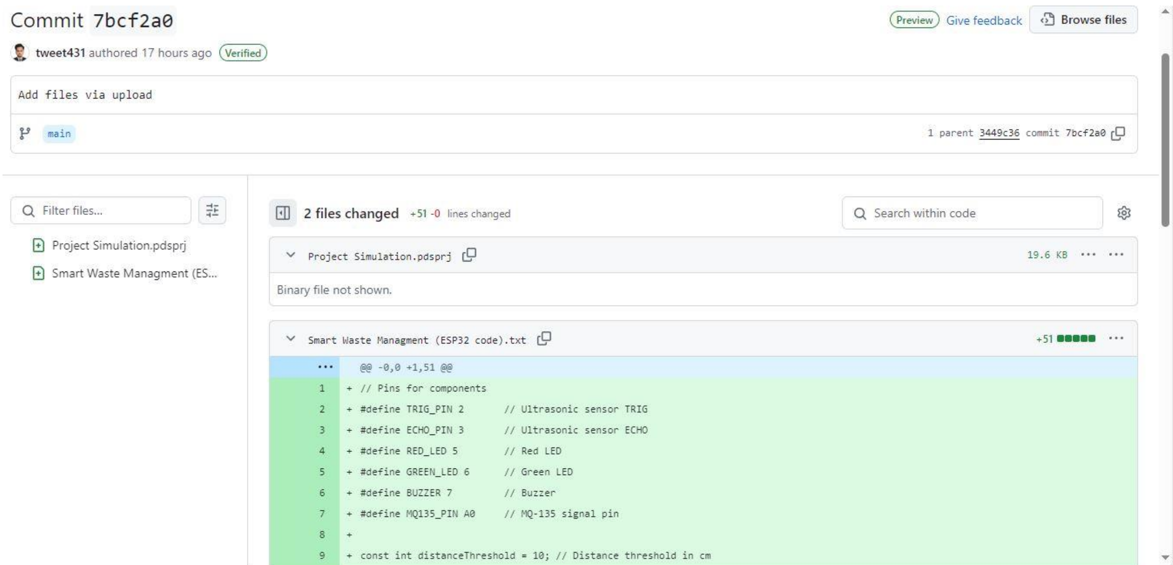
Software Requirements:

- **Simulation Tool:** Proteus for hardware simulation
- **IoT Platform:** [Specify platform, e.g., Blynk, MQTT, or Firebase] for remote monitoring and control
- **Development Tool:** Arduino IDE for **ESP32 programming**

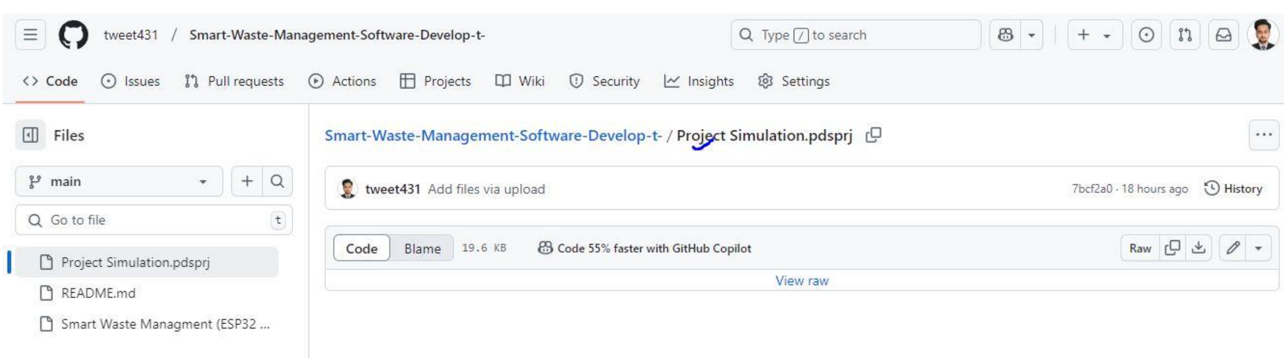
GitHub Management:

- **Repository link:**[https://github.com/tweet431/Smart-Waste-Management-Software-Develop-t-.git]
- **The repository includes:**

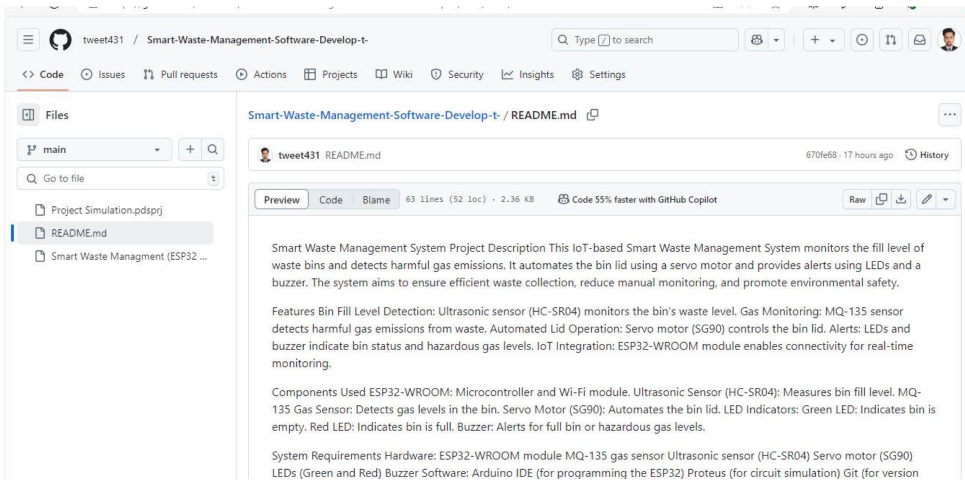
- ## ESP32 code:



- ## Proteus simulation files:



- **Documentation** (README file with project description, SDG alignment, and instructions)



Project Design and Development Process:

Hardware Design in Proteus:

- **Component Selection**:

  - **ESP32-WROOM** module for microcontroller functionality.
  - **MQ-135 sensor** connected to ADC pin for analog air quality readings.
  - **HC-SR04 sensor** connected to GPIO pins for distance measurements.
  - **LEDs connected** via resistors to GPIO16 and GPIO17 for signaling.
  - **Buzzer connected** to GPIO21 for critical alerts.

- **Circuit Design Steps in Proteus**:

  - Place ESP32, MQ-135, HC-SR04, LEDs, and buzzer in the workspace.
  - **Connect:**
    - **MQ-135**: VCC (3.3V), GND, and signal to ADC pin.
    - **HC-SR04**: TRIG to GPIO2, ECHO to GPIO4, VCC to 3.3V, and GND.
    - **Red LED (D1)**: Anode to GPIO16 through 220-ohm resistor, cathode to GND.
    - **Green LED (D2)**: Anode to GPIO17 through 220-ohm resistor, cathode to GND.
    - **Buzzer**: Signal pin to GPIO21, VCC to 5V, GND to GND.

➢ **Simulation Output**:

- Waste levels displayed accurately by the ultrasonic sensor.
- Gas quality indicated by the MQ-135 sensor values.
- LEDs and buzzer triggered based on defined thresholds.

ESP32 Code Development:

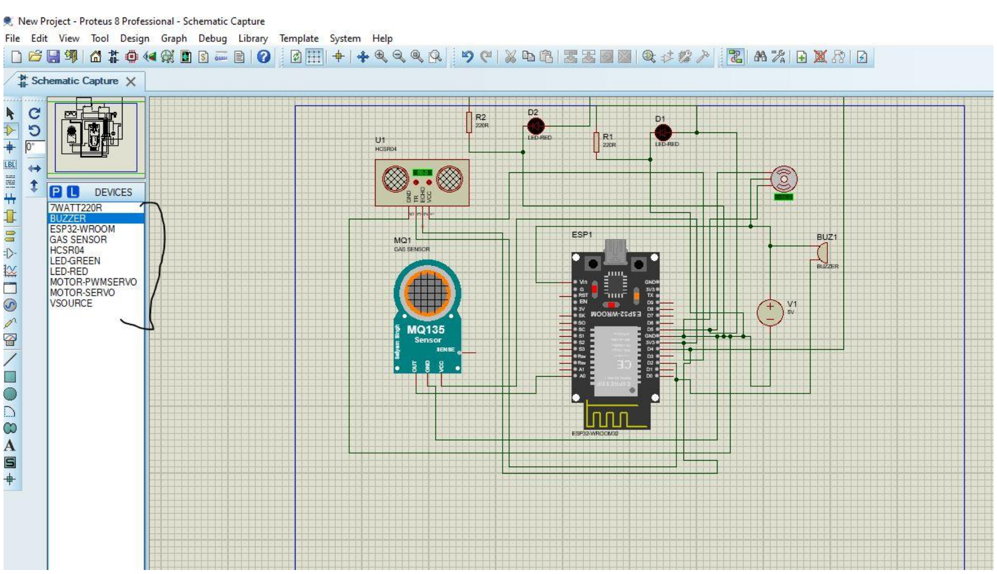
➢ **Code Features**:

- Reading real-time sensor data (gas and waste levels).
- Triggering actuators based on threshold conditions:
- Red LED for full bin or poor air quality.
- Buzzer for critical conditions.
- Sending data to IoT platform for monitoring and control.

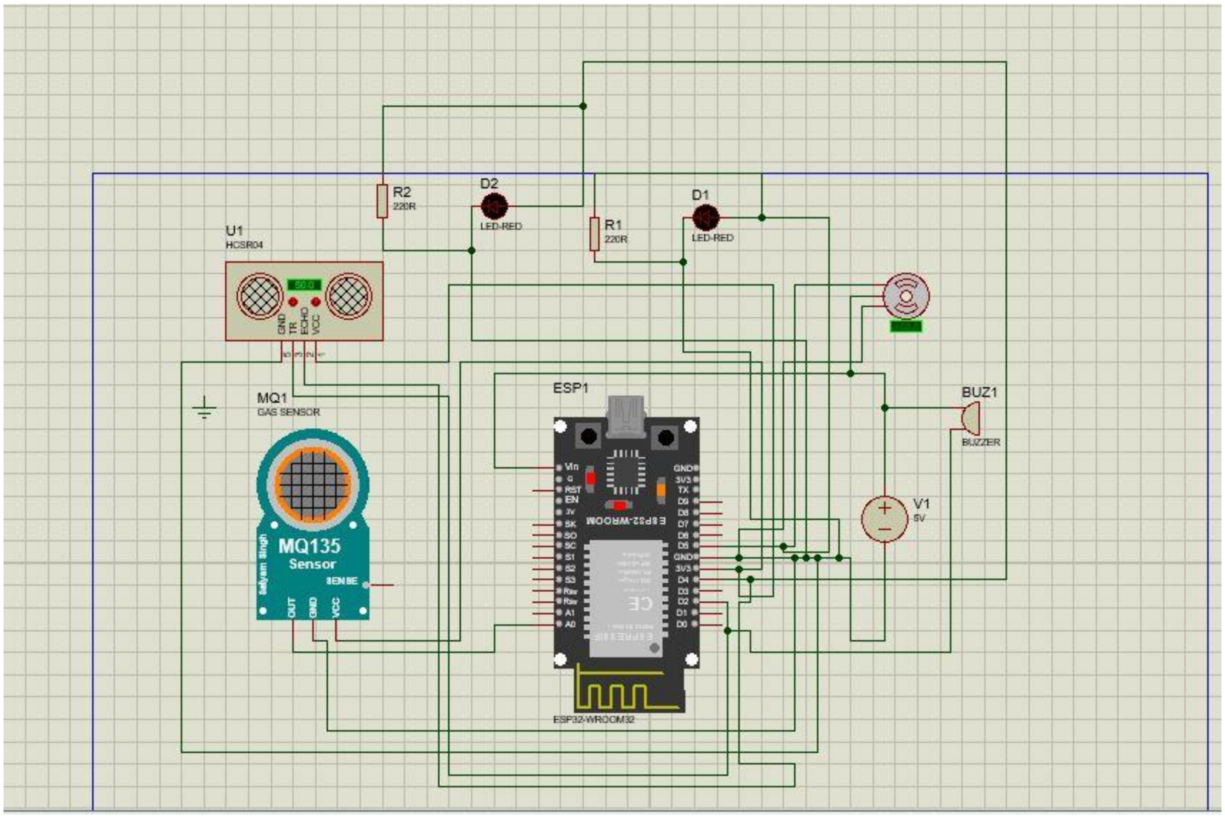
➢ **Implementation Steps**:

- Used the Arduino IDE for **ESP32-WROOM** programming.
- Defined sensor thresholds for triggering the actuators.
- Configured Wi-Fi for IoT platform communication.

Components Require in Proteus:



Proteus design:



IoT Platform Integration:

1. **Platform Setup**:

   - Configured IoT platform (e.g., Blynk/Firebase) for data logging and remote monitoring.
   - ESP32 sends sensor readings via Wi-Fi.

2. **Features**:

   - Real-time data visualization (gas and waste levels).
   - Remote control of actuators via IoT dashboard.

Results and Observations:

- **Proteus Simulation**:

- Sensors and actuators interact as expected.
- LEDs and buzzer respond to defined conditions

- **IoT Monitoring**:

- Real-time sensor data is displayed on the IoT platform.
- Users can remotely monitor and control the system.

➢ **System Behavior**:

- Red LED and buzzer activate for critical waste/gas levels.
- Green LED indicates normal status.

Final Simulation Result:

Challenges and Solutions:

➢ **Hardware Integration**:

- Power supply fluctuations resolved by using a stable 5V source.

➢ **Sensor Calibration**:

- MQ-135 calibrated for accurate air quality readings based on datasheet guidelines.

➢ **IoT Connectivity**:

- Wi-Fi configuration issues resolved by optimizing ESP32 settings.

Hardware Management and Breadboard Design:

To ensure proper organization and functionality, the hardware components were assembled and tested on a breadboard before final implementation. This approach allowed for:

- Efficient placement of components such as the ESP32, LEDs, resistors, ultrasonic sensor, MQ-135 sensor, and servo motor.

- Testing and verifying circuit connections, ensuring proper wiring of sensors and actuators.
- Flexibility in modifying the circuit for debugging and optimization before moving to a permanent setup.
- Clear visualization of connections, including VCC, GND, and GPIO pins, to validate the design.

Future Improvements:

- Integrate additional sensors (e.g., temperature, humidity).
- Use machine learning for advanced waste prediction and analytics.
- Miniaturize the prototype for real-world implementation.

Conclusion:

The Smart Waste Management System showcases the potential of IoT to address sustainability challenges, particularly waste management. By leveraging sensors, actuators, and IoT platforms, the project aligns with the SDG Goal 11, promoting cleaner and more sustainable cities.