iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** VI **Month of publication:** June 2026

DOI: https://doi.org/10.22214/ijraset.2026.83493

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VI June 2026- Available at www.ijraset.com

IoT-Based Environmental Monitoring and Automation System

Rosmi Koley, Roshni Roy, Rupkatha Goswami, Sayandip Dutta, Sudeshna Mondal, Kaushik Roy, Surajit Basak, Nabaneeta Banerjee
Department of Electronics and Communication Engineering, Guru Nanak Institute of Technology, Kolkata, India

Abstract—Lately the Internet of Things has completely changed how we monitor and automate things in our lives. This paper presents an environmental monitoring system designed using an ESP32 microcontroller tested on the Wokwi online simulator. The system uses a DHT22 sensor to measure temperature and humidity, an LDR sensor to sense light intensity and a potentiometer dial to simulate a rain sensor. All sensor data is continuously sent to the ESP32 microcontroller, which takes actions based on predefined conditions. If the temperature crosses 30°C, the relay module turns on the Green LED, representing a cooling fan. The ESP32 also switches on the Red LED when the environment becomes dark and activates a buzzer alarm when rain is detected through the potentiometer input. A 20x4 LCD screen displays all live sensor readings and device status in real time. The ESP32 also uses its built-in Wi-Fi to upload the collected data to the ThingSpeak cloud platform, allowing users to monitor and log data remotely. The simulation proved that the sensors, LEDs, buzzer and cloud communication worked properly. This IoT system acts as a working prototype suitable for smart homes, farming, greenhouses and warehouse monitoring applications.

Keywords— IoT; ESP32; environmental monitoring; ThingSpeak; Wokwi simulation

I. INTRODUCTION

The Internet of Things (IoT) has changed the way we monitor environments and automate tasks in homes and industries. IoT systems can collect data remotely and control devices automatically, making them useful for smart cities and connected homes. The ESP32 microcontroller is widely used for such systems because it has built-in Wi-Fi and faster processing compared to older controllers like the Arduino Uno.

In this project we developed an IoT based environmental monitoring and automation system using the ESP32 microcontroller. The system measures temperature, humidity and light intensity and also detects rain conditions through a simulated rain sensor. Based on sensor readings, the ESP32 automatically turns on a green LED when the temperature becomes high, switches on a red LED when the surroundings become dark and activates a buzzer alarm during rainyconditions. Sensors like the DHT22, LDR and potentiometer were used to make the system responsive to environmental changes. A 20×4 LCD screen displays real-time sensor readings and actuator status, allowing users to monitor the environment easily. The system also uploads sensor data to the internet through ThingSpeak cloud using the ESP32 Wi-Fi feature, enabling remote monitoring and data analysis. Through this project we learned how IoT devices communicate, process sensor data and automate real-world applications using ESP32.

II. SYSTEM ARCHITECTURE

In this project we used the ESP32 microcontroller to read sensor data and control different actuators automatically. The complete setup was designed and tested using the Wokwi online simulator. The ESP32 was selected because of its fast-processing capability and built-in Wi-Fi support, which allows direct internet connectivity without additional modules.

The ESP32 continuously collects data from three different sensors: a DHT22 sensor for temperature and humidity measurement, an LDR sensor for sensing ambient light intensity and a potentiometer that acts as a simulated rain sensor. Since the project was developed in a simulator, the potentiometer was used to mimic rainfall conditions by changing voltage values. The LDR sensor is connected to an analog pin and provides readings between 0 and 4095.The ESP32 processes all sensor data and controls the connected devices according to predefined conditions. When the temperature becomes high, the system turns on a green LED representing a cooling fan. When the LDR senses low light conditions, the ESP32 switches on a red LED. Similarly, when the potentiometer simulates rain, the buzzer gets activated. All sensor readings and device status are displayed on the LCD screen for local monitoring.

At the same time, the ESP32 sends temperature, humidity, light and rain data to the ThingSpeak cloud platform through the Wi-Fi, converting the local setup into a complete IoT based monitoring system.

III. WORKING MECHANISM

The ESP32 microcontroller works in a continuous closed loop monitoring system where it constantly receives signals from the sensors and processes them. Since the sensor outputs are analog in nature, the ESP32 converts them into digital values between 0 and 4095 for processing. The controller then compares these values with predefined threshold conditions. If the temperature crosses 30°C, the ESP32 turns on the green LED, which represents a cooling fan. When the simulated rain value falls below 500, the system detects moisture or rain and activates the buzzer alarm. Similarly, if the light intensity value rises above 2000, indicating darkness, the ESP32 switches on the red LED. In this way the microcontroller automatically controls the connected devices based on environmental conditions. The ESP32 also updates the 20x4 LCD screen with live temperature, humidity, light and device status information. At the same time, all sensor data is uploaded to the ThingSpeak cloud platform using Wi-Fi connectivity. This allows users to monitor environmental conditions remotely from any internet-connected device.

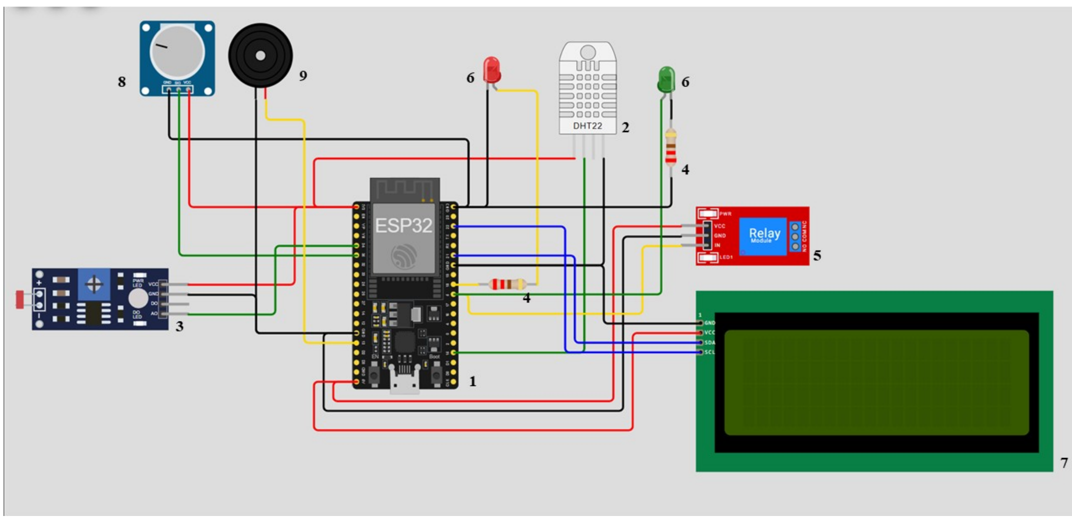


Fig. 1Circuit Diagram of the IoT System

IV. RESULTS AND DISCUSSION

1) System Automation and Response

The simulation confirmed that the ESP32 successfully managed the continuous feedback loop for all IoT sensors. For climate control, the relay kept the cooling fan OFF during normal temperature conditions such as 25.6°C. When the temperature crossed the threshold value of 30°C, especially under high humidity conditions, the system automatically activated the relay and turned the fan ON. The LCD screen also updated the status to “Fan: ON” in real time.

2) Light and Rain Detection

The ESP32 accurately handled ambient light monitoring using the LDR sensor. During bright conditions, such as 1591 lux, the red LED remained OFF. When the light intensity decreased and the sensor value crossed 2000, for example 2144 lux, the ESP32 correctly switched the LED ON to indicate darkness. The potentiometer also successfully simulated rain conditions. Values above the threshold of 500 kept the buzzer inactive, while values below 500, such as 380 or 0, immediately activated the buzzer and displayed a “RAIN!” warning on the LCD screen.

3) Data Logging and Display

The 20x4 I2C LCD Display worked perfectly and it continuously shows updated temperature, humidity, light intensity, weather status and actuator conditions without any noticeable delay. Along with the local monitoring, the ESP32 successfully used its built-in Wi-Fi module to send sensor data to the ThingSpeak cloud platform. This confirmed that the prototype can perform reliable remote environmental monitoring and data logging.

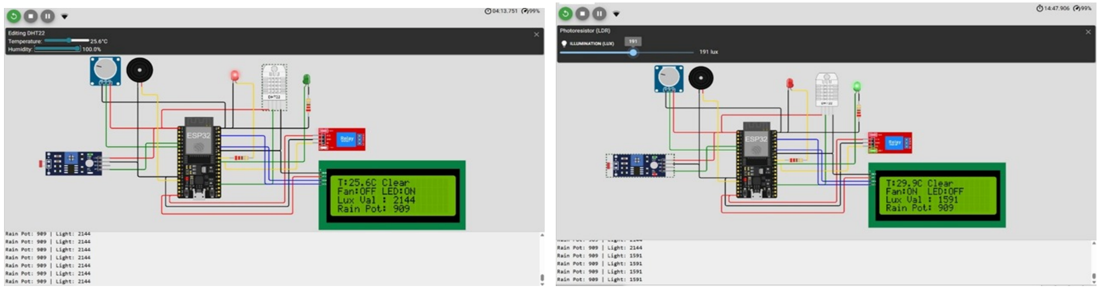


Fig. 2 Simulation Outputs of the System

V. APPLICATIONS

The Internet of Things-Based Environmental Monitoring and Automation System is really useful in situations where we need to monitor and control the environment. For example, in our homes this system can turn on the fans, lights and alarms when it is necessary based on what's happening in the environment. This helps us save energy and be safer. In places where they grow plants like greenhouses, the Internet of Things-Based Environmental Monitoring and Automation System checks the temperature how humid it is and how light there is so that the plants can grow well. The Internet of Things-Based Environmental Monitoring and Automation System can also be used in storage rooms to always check the conditions for things that are sensitive to temperature and it keeps all the records on the internet. In rooms, with a lot of computers the Internet of Things-Based Environmental Monitoring and Automation System helps keep the temperature safe by turning on the cooling systems and letting us check everything from away using the internet.

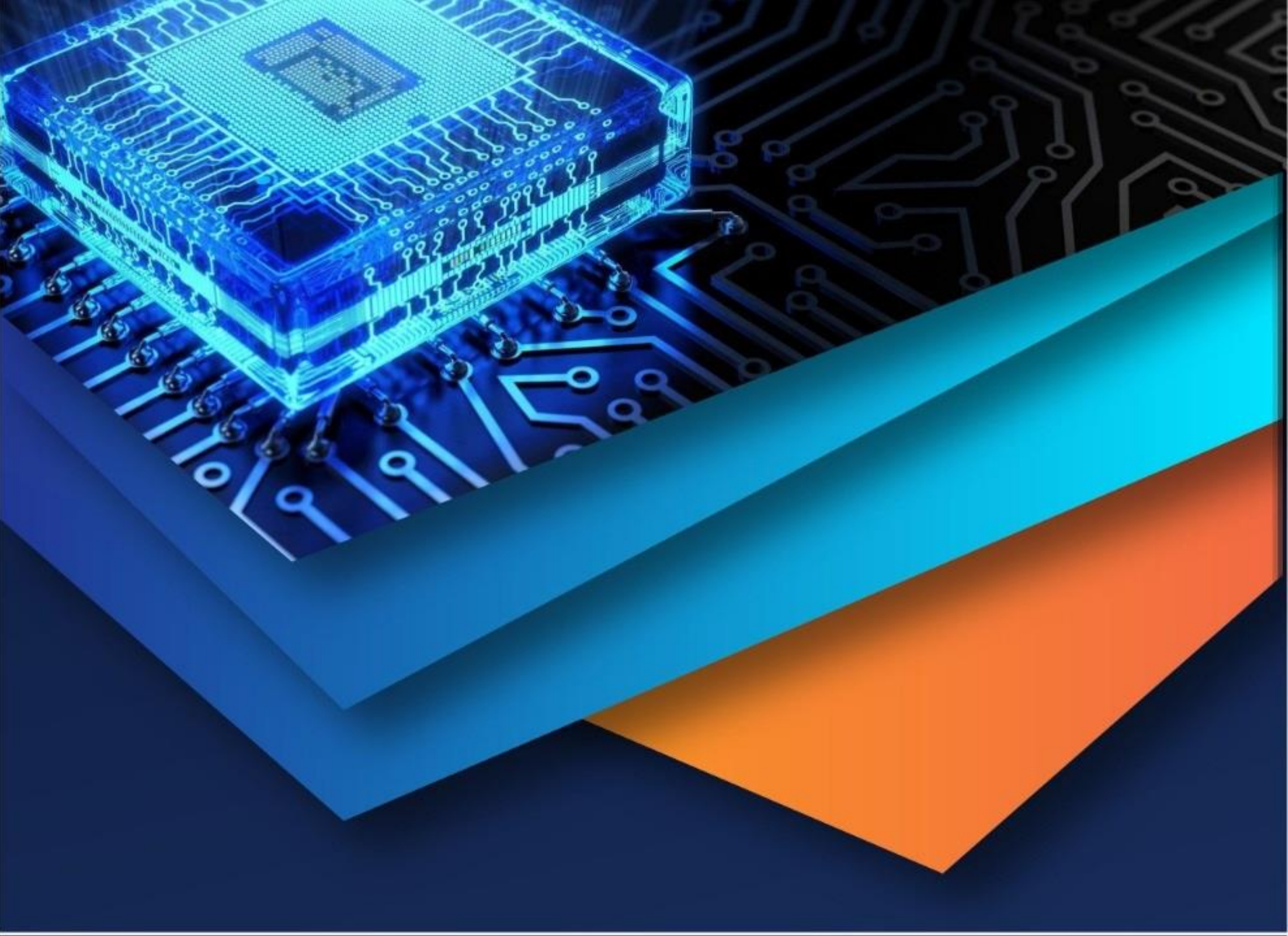
VI. CONCLUSIONS AND FUTURE SCOPE

This project showed that an Environmental Monitoring and Automation System using the ESP32 microcontroller works well. The system checked temperature, humidity, light intensity and rain. It also controlled LEDs, a relay and a buzzer automatically based on set limits. The ESP32 sent sensor data to the ThingSpeak cloud for monitoring from anywhere. A Wokwi simulation test proved that all parts, including sensors, actuators and cloud connection worked correctly. This makes the prototype reliable for use.

In the future we can make the system better by adding advanced sensors and actuators. This will improve automation. We can also add an app, for easier monitoring and control and send notifications in real-time. The project can be expanded to homes, industrial safety, greenhouse monitoring and big IoT automation projects. The ESP32 and IoT-Based Environmental Monitoring and Automation System can be used in areas. The system and ESP32 have a lot of potential for growth.

REFERENCES

[1] Espressif Systems, *ESP32 Technical Reference Manual*. Available: https://www.espressif.com/en/support/documents/technical-documents
[2] Wokwi, *Wokwi Online Arduino & ESP32 Simulator*. Available: https://wokwi.com/
[3] MathWorks, *ThingSpeak API Reference: Read Data from Channel*. Available: https://uk.mathworks.com/help/thingspeak/read-data-from-channel.html
[4] Arduino, *WiFi Library for ESP32*. Available: https://www.arduino.cc/reference/en/libraries/wifi/
[5] Adafruit Industries, *DHT Sensor Library*. Available: https://github.com/adafruit/DHT-sensor-library
[6] F. de Brabander, *LiquidCrystal I2C Library*. Available: https://github.com/johnrickman/LiquidCrystal_I2C
[7] M. S. Bakare and K. Abubaker, "IoT-based indoor environmental monitoring system using multi-parameter sensing and ESP32-WROOM integration," *Discover Electronics*, vol. 3, no. 6, 2026.
[8] N. Wivanius, W. I. Sihombing, and D. Kamsyah, "Design and Prototype Development of an IoT-Based Temperature and Humidity Monitoring System with Real-Time Data and Automated Alerts," *Jurnal Teknologi dan Riset Terapan*, vol. 7, no. 2, 2026.



10.22214/IJRASET

INDEX COPERNICUS 45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY