

REPORT
ON
INTERNSHIP-I
CARRIED OUT ON
IoT- BASED REAL TIME WEATHER MONITORING VIA
BLINK PLATFORM

Submitted to

NMAM INSTITUTE OF TECHNOLOGY, NITTE

(Off-Campus Centre, Nitte Deemed to be University, Nitte - 574 110, Karnataka,
India)

In partial fulfilment of the requirements for the award of the Degree of Bachelor of
Technology in

Electronics and Communication Engineering

By

ASHLESH Y SALIYAN

(NNM24EC028)

UNDER THE GUIDANCE OF

DR. SATEESH RAO

(ASSISTANT PROFESSOR Gd-III)

ACKNOWLEDGEMENT:

I would like to sincerely thank my project guide, Dr. Sateesh Rao, for their support and guidance during the completion of this project.

I'm also thankful to the faculty of Department of Electronics and Communication Engineering for providing the resources and environment needed to carry out this work.

Lastly, I appreciate the encouragement and help from my fellow team members Ashwin, Ashley, Ashmith, Ayisha, Ayush and Lokita throughout this project.

WEATHER MONITORING SYSTEM

TABLE OF CONTENTS:

Title	Page No
Cover page	1
Acknowledgement	2
Table of contents	3
Abstract	4
Introduction	5
Project Overview	6
Conclusion	13
References	14

WEATHER MONITORING SYSTEM

ABSTRACT:

This project focuses on creating a simple Internet of Things (IoT)-based environmental monitoring system utilizing an ESP32 microcontroller. Its main objective is to monitor and display real-time data for temperature, humidity, and air quality. For this, the DHT11 sensor is employed to gather temperature and humidity information, while the MQ135 sensor detects gases and assesses overall air quality.

The ESP32 collects data from these sensors and leverages its built-in Wi-Fi module to transmit the readings to the Blynk IoT platform. The data is then presented in real-time on the Blynk mobile app, enabling convenient remote access and monitoring.

This system highlights an affordable and effective approach to tracking fundamental environmental parameters. It is ideal for educational applications and small-scale setups requiring real-time insights into climate and air quality.

WEATHER MONITORING SYSTEM

INTRODUCTION:

With the increasing focus on environmental consciousness, the demand for real-time monitoring systems has grown significantly. IoT-based solutions provide a smart and effective way to collect, transmit, and visualize environmental data remotely.

This project showcases a fundamental IoT-enabled system that employs the ESP32 microcontroller to gather temperature and humidity data using the DHT11 sensor, along with air quality data through the MQ135 sensor. Using its built-in Wi-Fi capabilities, the ESP32 sends this data to the Blynk platform, enabling real-time monitoring through a smartphone application.

One key benefit of integrating Blynk is the ability to access and monitor data from anywhere in the world, provided there's an internet connection. This feature allows users to conveniently track and analyze environmental conditions remotely.

The primary goals of this project include:

- Capturing temperature, humidity, and air quality measurements via sensors.
- Transmitting collected data wirelessly with the ESP32 module.
- Displaying real-time data on the Blynk mobile application/platform.
- Demonstrating a simple, yet effective, environmental monitoring system tailored for educational purposes.

PROJECT OVERVIEW:

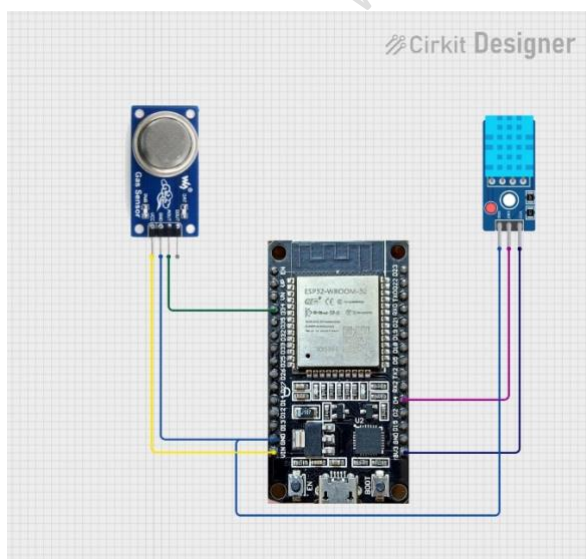
COMPONENTS USED:

- **ESP32 Microcontroller:** Handles the processing and communication between the sensors and the Blynk platform via Wi-Fi.
- **DHT11 Sensor:** Measures temperature and humidity levels in the environment.
- **MQ135 Gas Sensor:** Detects the presence of harmful gases and monitors air quality.
- **Blynk Platform:** Used for displaying real-time data on a mobile application.
- **Jumper Wires and Breadboard:** Used for connecting the components in the circuit.

WORKING:

1. The **DHT11** sensor measures the temperature and humidity in the surrounding environment.
2. The **MQ135** sensor detects the concentration of various gases (such as ammonia, benzene, CO2, etc.) to monitor air quality.
3. The **ESP32** collects data from both sensors, processes it, and sends it over Wi-Fi to the **Blynk platform**.
4. The **Blynk app** displays this data in real-time, allowing the user to monitor environmental conditions remotely from anywhere with an internet connection.

CIRCUIT DIAGRAM:



You can view and interact with the full simulation on: <https://bit.ly/42RHjUM>

PROGRAMMING CODE:

```
#define BLYNK_TEMPLATE_ID "TMPL3hjMiGISE"
#define BLYNK_TEMPLATE_NAME "Air Quality Monitoring System Copy"
#define BLYNK_AUTH_TOKEN "WBPzKvjUQxw5Xg1L97ub-bPjSMERKJql"

// Enable serial debug prints
#define BLYNK_PRINT Serial

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <DHT.h>

// Sensor pins
#define DHT11_PIN 4 // ESP32 pin GPIO4 connected to DHT11 sensor
#define MQ135_PIN 34 // ESP32 pin connected to MQ135 A0 pin

// Initialize DHT sensor
DHT dht11(DHT11_PIN, DHT11);

// Wi-Fi credentials
char ssid[] = "ASHJ4";
char pass[] = "samgalj4";

// Smoke detection threshold (adjust based on calibration)
const int smokeThreshold = 300;

// Variables for MQ135 sensor smoothing
const int numReadings = 10;
int mq135Readings[numReadings];
int mq135Index = 0;
int total = 0;
int average = 0;
```

```

void setup() {
  // Initialize serial communication
  Serial.begin(115200);

  // Initialize DHT sensor
  dht11.begin();

  // Connect to Blynk
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);

  // Initialize MQ135 readings array
  for (int i = 0; i < numReadings; i++) {
    mq135Readings[i] = 0;
  }

  // Print connection status
  Serial.println("System initialized");
}

void loop() {
  // Run Blynk operations
  Blynk.run();

  // Read and send DHT11 data
  readAndSendDHTData();

  // Read and send MQ135 data
  readAndSendMQ135Data();

  // Short delay between readings
  delay(1000);
}

```

```

void readAndSendDHTData() {
    // Read humidity and temperature
    float humi = dht11.readHumidity();
    float tempC = dht11.readTemperature();
    float tempF = dht11.readTemperature(true);

    // Check if readings are valid
    if (isnan(tempC) || isnan(tempF) || isnan(humi)) {
        Serial.println("Failed to read from DHT11 sensor!");
        return;
    }

    // Send data to Blynk
    Blynk.virtualWrite(V0, tempC); // V0 - Temperature in Celsius
    Blynk.virtualWrite(V1, tempF); // V1 - Temperature in Fahrenheit
    Blynk.virtualWrite(V2, humi); // V2 - Humidity in percentage

    // Print to serial monitor
    Serial.print("Temperature: ");
    Serial.print(tempC);
    Serial.print("°C, Humidity: ");
    Serial.print(humi);
    Serial.println("%");
}

void readAndSendMQ135Data() {
    // Read raw sensor value
    int sensorValue = analogRead(MQ135_PIN);

    // Update moving average
    total -= mq135Readings[mq135Index];
    mq135Readings[mq135Index] = sensorValue;
}

```

```
total += mq135Readings[mq135Index];
mq135Index = (mq135Index + 1) % numReadings;

// Calculate average
average = total / numReadings;

// Convert to PPM (adjust calibration as needed)
float ppm = map(average, 0, 1023, 0, 1000);

// Send to Blynk
Blynk.virtualWrite(V3, ppm); // V3 - Gas Concentration in PPM

// Print to serial monitor
Serial.print("Air Quality: ");
Serial.print(ppm);
Serial.println(" PPM");

// Check threshold (optional alert)
if (ppm > smokeThreshold) {
  Blynk.logEvent("smoke_alert", "High smoke concentration detected!");
}
}
```

OUTPUT:

The ESP32 successfully collected real-time temperature, humidity, and air quality data from the DHT11 and MQ135 sensors. The data was transmitted to the Blynk app via Wi-Fi, allowing remote monitoring of environmental conditions.

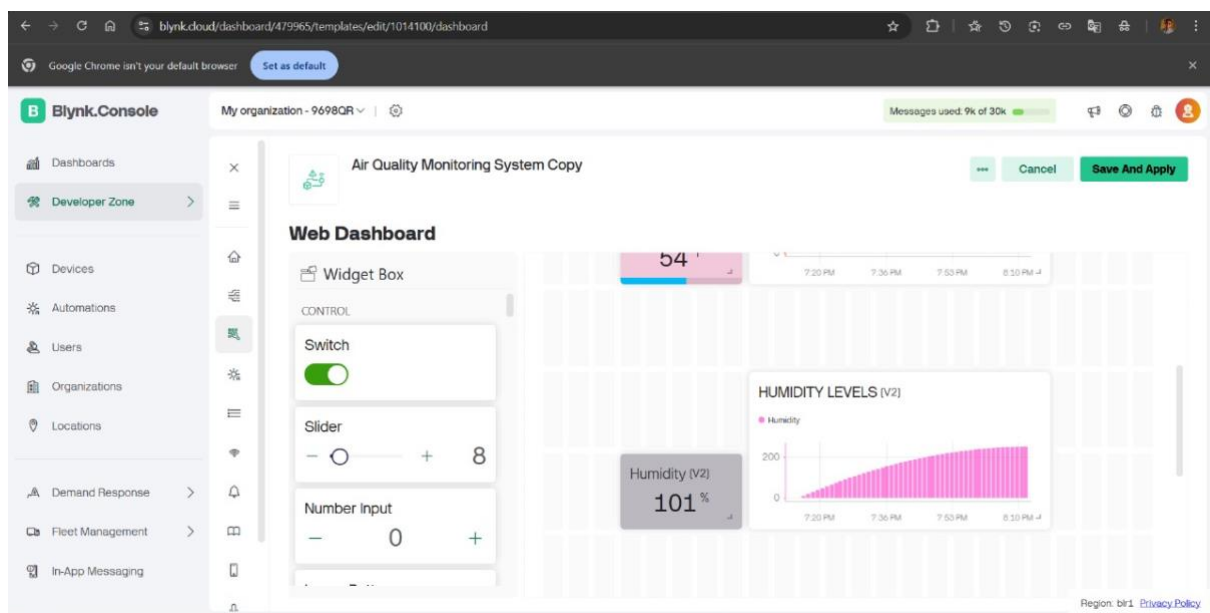


Figure 1.1: Web Dashboard of the Blynk platform, showing real-time humidity data in percentage, collected using DHT11 sensor via ESP32 microcontroller.

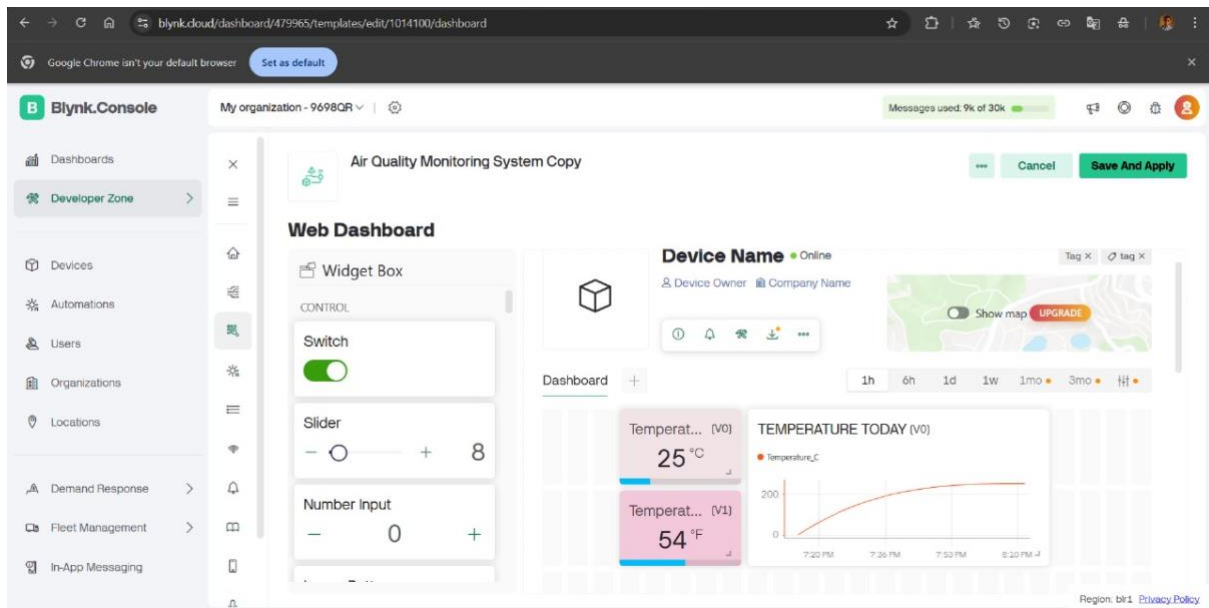


Figure 1.2: Web Dashboard of the Blynk platform, showing real-time temperature data in Celsius and Fahrenheit, collected using DHT11 sensor via ESP32 microcontroller.

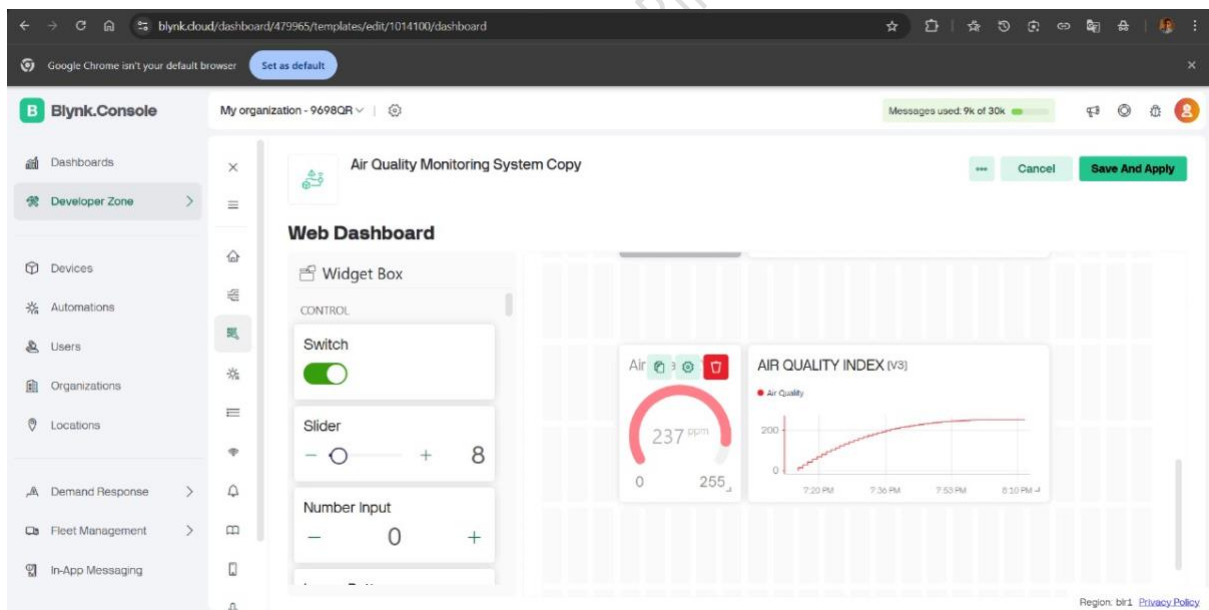


Figure 1.3: Web Dashboard of the Blynk platform, showing real-time air quality data in ppm, collected using MQ135 sensor via ESP32 microcontroller.

CONCLUSION:

The project effectively showcased real-time monitoring of environmental parameters, including temperature, humidity, and air quality, through the integration of DHT11 and MQ135 sensors with an ESP32 microcontroller. The collected data was seamlessly processed and transmitted via Wi-Fi to the Blynk application, enabling users to access environmental conditions remotely, anytime and anywhere. This reliable, cost-effective, and easy-to-implement system serves as a robust model for basic IoT-based monitoring solutions. Furthermore, the project sets a solid groundwork for developing more sophisticated systems incorporating additional sensors and features in the future.

APPLICATIONS:

1. Continuous monitoring of air quality and temperature in residential, commercial, and industrial settings.
2. Seamless integration with smart home ecosystems to optimize indoor air management.
3. Real-time data collection for environmental studies and educational advancements.
4. Development of alert systems for health and safety in regions with compromised air quality.
5. Smart farming applications, such as greenhouse climate control, ensuring ideal conditions for growth.
6. Foundational model for scalable IoT projects incorporating advanced sensor networks.

REFERENCES:

Khan, S. (2020, May 8). **How to use the MQ135 gas sensor with Arduino** [Video]. YouTube. <https://youtu.be/gyEXT2BwaRs>

Maker Pro. (2024, January 15). *IoT based air quality monitoring system with ESP32 board using MQ135 sensor and an LCD display* [Video]. YouTube. <https://www.youtube.com/watch?v=zz9KBPVDMhE>

Random Nerd Tutorials. (2024, February 10). *ESP32 DHT11/DHT22 temperature and humidity sensor with Arduino IDE* Random Nerd Tutorials. <https://randomnerdtutorials.com/esp32-dht11-dht22-temperature-humidity-sensor-arduino-ide/>

IoTWebPlanet. (2024, September 5). *ESP32 with an MQ-135 gas sensor*. IoTWebPlanet. <https://www.iotwebplanet.com/esp32-with-an-mq-135-gas-sensor/>

WEATHER MONITORING SYSTEM