

Name: Rohit Khatri

Submission Date: 14-11-2025

IoT Lab Report: Smart Room Environment Monitoring System Using Blynk

1. System Explanation

The Smart Room Environment Monitoring System is an IoT-based project that monitors the temperature and humidity of a room in real time. The system uses a DHT11 sensor to collect environmental data and an ESP8266 Wi-Fi module to send this data to the Blynk IoT platform. Through Blynk's mobile and web dashboards, users can view the readings and receive alerts if the temperature or humidity exceed safe thresholds. This system can be deployed in homes, hospitals, server rooms, or greenhouses to ensure environmental stability and safety.

2. Resources Used

Hardware Components:

- ESP8266MOD (Generic ESP8266 Module): Acts as the main microcontroller and Wi-Fi interface.
- DHT11 Sensor: Measures temperature and humidity.
- Jumper Wires and Breadboard: For connecting the components.

Software Components:

- Arduino IDE: Used for programming the ESP8266.
- Blynk IoT Platform (Web + Mobile App): Used for real-time visualization and notifications.
- Libraries: ESP8266WiFi.h, BlynkSimpleEsp8266.h, DHT.h, BlynkTimer.h

Output Devices:

- Blynk Web and Mobile Dashboards: Display real-time sensor data.
 - Push Notification: Sends alerts when values exceed predefined limits.
-

3. System Architecture

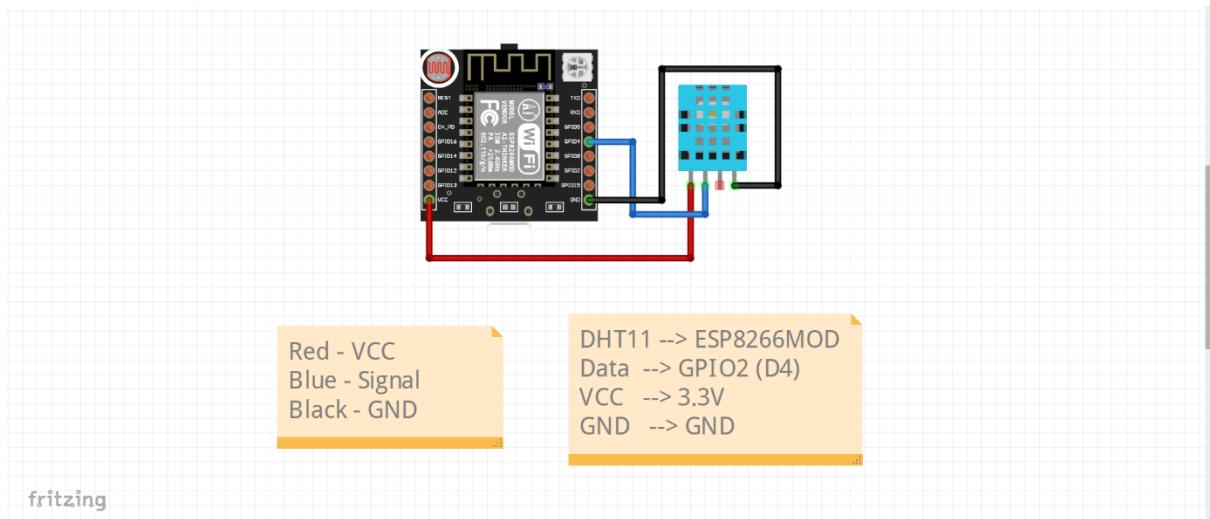
Component Flow:



Key Functions:

- The DHT11 sensor captures data.
- ESP8266 reads this data and sends it over Wi-Fi.
- Blynk receives the data via virtual pins (V0 for temperature, V1 for humidity).
- Blynk also manages alerts using events like temp_alert and humidity_alert.

Circuit Diagram: (Fritzing)



4. Protocols Used

- **GPIO (General Purpose Input Output)**: Used to read data from DHT11.
- **UART (Universal Asynchronous Receiver Transmitter)**: Used internally by ESP8266 to communicate with the serial monitor.
- **Wi-Fi (802.11)**: ESP8266 connects to the internet via Wi-Fi.
- **HTTP (via Blynk Library)**: Communicates with Blynk's cloud server.
- **Blynk Protocol**: Abstraction layer that simplifies cloud interaction using `Blynk.virtualWrite()` and `Blynk.logEvent()`.

5. Programming Code

```
// Replaced actual Blynk Template ID for security reasons
#define BLYNK_TEMPLATE_ID "YourTemplateID"
// Template name retained as it does not expose any credentials
#define BLYNK_TEMPLATE_NAME "Smart Room Monitoring"
// Replaced actual Blynk Auth Token for security reasons
#define BLYNK_AUTH_TOKEN "YourAuthToken"

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>

// Replaced actual token for security reasons
char auth[] = "YourAuthToken";
// Replaced actual WiFi SSID for security reasons
char ssid[] = "YourWiFiSSID";
// Replaced actual WiFi password for security reasons
char pass[] = "YourWiFiPassword";
```

```
#define DHTPIN 2 // Use GPIO number, not D4
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

BlynkTimer timer;

void sendSensor() {
    float h = dht.readHumidity();
    float t = dht.readTemperature();

    if (isnan(h) || isnan(t)) {
        Serial.println("Failed to read from DHT sensor!");
        return;
    }

    Blynk.virtualWrite(V0, t); // Send temp to app
    Blynk.virtualWrite(V1, h); // Send humidity to app

    if (t > 33.0) {
        Blynk.logEvent("temp_alert", "Room temperature is above 33°C!");
    }

    if (h > 70.0) {
        Blynk.logEvent("humidity_alert", "Humidity level is above 70%!");
    }
}

void setup() {
    Serial.begin(9600);
    Blynk.begin(auth, ssid, pass);
    dht.begin();

    timer.setInterval(2000L, sendSensor); // every 2 sec
}

void loop() {
    Blynk.run();
    timer.run();
}
```

6. Snapshots

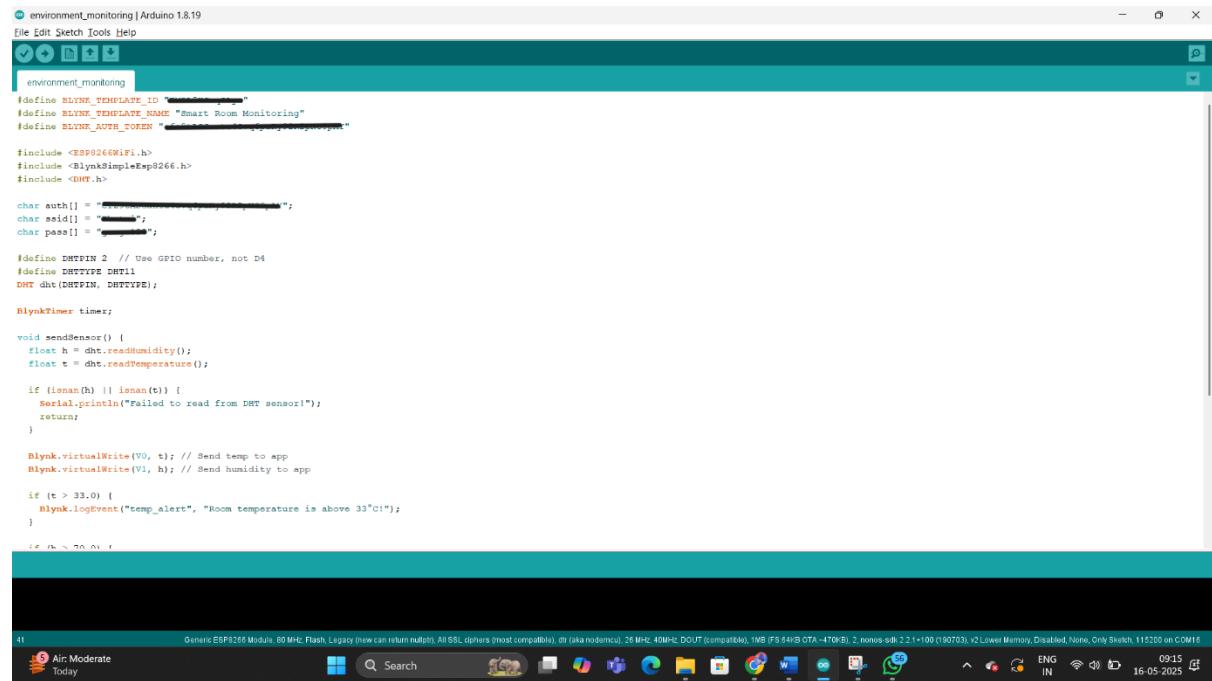
Blynk Web Console:



Blynk Mobile App screen shot showing live temperature and humidity:



Arduino IDE:



The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** environment_monitoring | Arduino 1.8.19
- File Menu:** File Edit Sketch Tools Help
- Sketch Name:** environment_monitoring
- Code Content:**

```
#define BLYNK_TEMPLATE_ID "████████"
#define BLYNK_TEMPLATE_NAME "Smart Room Monitoring"
#define BLYNK_AUTH_TOKEN "████████████████████████████████████████"

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>

char auth[] = "████████";
char ssid[] = "████";
char pass[] = "████";

#define DHTPIN 2 // Use GPIO number, not D4
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

BlynkTimer timer;

void sendSensor() {
    float h = dht.readHumidity();
    float t = dht.readTemperature();

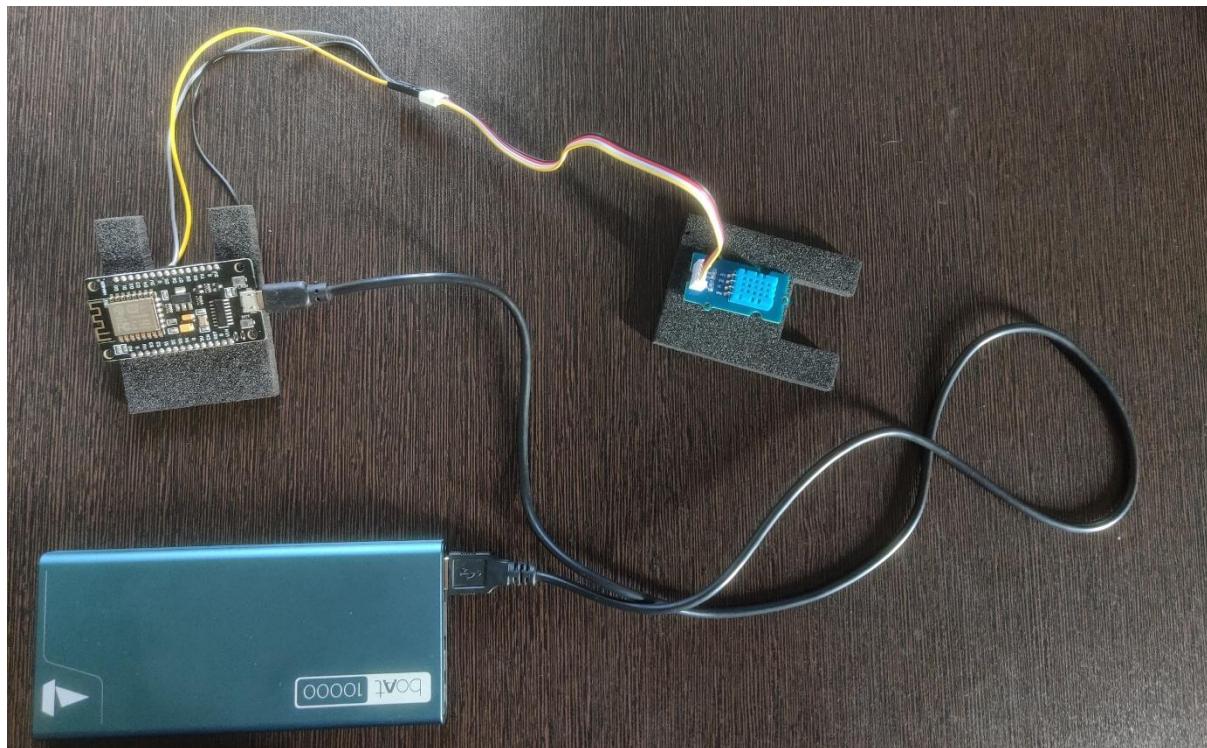
    if (isnan(h) || isnan(t)) {
        serial.println("Failed to read from DHT sensor!");
        return;
    }

    Blynk.virtualWrite(V0, t); // Send temp to app
    Blynk.virtualWrite(V1, h); // Send humidity to app

    if (t > 33.0) {
        Blynk.logEvent("temp_alert", "Room temperature is above 33°C!");
    }
}

if (dh > 70.0) f
```
- Bottom Status Bar:** Generic ESP8266 Module, 80 MHz, Flash, Legacy (new can return nullptr), All SSL ciphers (most compatible), dt (aka nodemcu), 26 MHz, 40MHz DOUT (compatible), 1MB (FS 64kB OTA - 470kB), 2, nomos-sdk 2.2.1+100 (190703), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM16, Air: Moderate Today, ENG IN, 09:15, 16-05-2025

Real photo of connections:



7. Issues & Privacy/Security Discussion

Issues Faced:

- ESP8266 compile errors due to using symbolic pin names (e.g., D4 instead of GPIO number).
- Unstable sensor readings in uncalibrated environments.

Privacy/Security Considerations:

- The system uses Wi-Fi to send data to the cloud, so using a secure password-protected network is essential.
 - Blynk authentication token should be kept private.
 - Optionally, HTTPS can be used in place of HTTP if supported.
-

Conclusion: The Smart Room Monitoring System offers a simple yet powerful solution for real-time environmental monitoring. Using Blynk simplifies the cloud integration process and makes IoT development accessible and efficient.