Smart Temperature and Humidity Monitoring System Using IoT Cloud

Avinash Kumar Pandey[1], Trinetra Sahu[2], Anjali Malviya[3], Rajshree[4]
[1,2,3,4,5] *M.C.A (Computer Science) Student, Sam Global University, Bhopal*

Abstract--In the modern era of digital transformation, remote environmental monitoring has become essential across sectors such as agriculture, healthcare, and smart homes. This paper presents a Smart Temperature and Humidity Monitoring System using IoT, designed with ESP8266 microcontroller and DHT11 sensor, leveraging the Blynk Cloud for real-time data visualization. The system, simulated on the Wokwi platform, ensures cost-effectiveness, portability, and minimal hardware dependency. A comparative analysis with existing IoT-based systems highlights its advantages in mobile integration, data accuracy, and real-time cloud updates. The implementation demonstrates efficient, scalable, and user-friendly environmental monitoring suitable for a variety of applications.

I. INTRODUCTION

With the Internet of Things (IoT), physical devices can communicate, collect, and share data over the internet, enabling automation and smart monitoring. Temperature and humidity are critical environmental parameters that influence productivity, safety, and comfort. Traditional systems rely on manual methods or expensive setups. This project proposes a low-cost, IoT-enabled system using ESP8266 and DHT11, simulated via Wokwi, and integrated with Blynk Cloud for real-time monitoring. This project presents the design and implementation of a Smart Temperature and Humidity Monitoring System that utilizes IoT principles to monitor environmental conditions in real time. The system consists of a DHT11 sensor for capturing data and an ESP32 microcontroller for data processing and communication. The collected data is transmitted over Wi-Fi to the Blynk Cloud, where users can view it on a mobile dashboard

II. LITERATURE REVIEW

The Internet of Things (IoT) has emerged as a key enabler in automation and remote monitoring applications across various domains including healthcare, agriculture, weather tracking, and smart homes. In particular, temperature and humidity monitoring is a vital application, as these two environmental parameters directly impact human health, food safety, electronics operation, and agricultural productivity.

Many researchers and developers have proposed IoT-based systems for sensing and tracking temperature and humidity. This section provides a detailed review of existing literature and technologies related to environmental monitoring using IoT, the limitations of those systems, and how our proposed project addresses those gaps using simulation platforms like Wokwi and modern cloud tools like Blynk IoT.

Study	Platform Used	Features	Limitation
1. Arduino + ThingSpeak (2017)	DHT11, ThingSpeak	Web-based data logging	No mobile dashboard
2. GSM SMS Alerts (2018)	ATmega, GSM Module	SMS alerts	No real-time dashboard
4. DHT 11	Hardware device	Moniter Temp. & Humd.	Expensive hardware
5. ESP32 + Blynk (2021)	Blynk App, Mobile Dashboard	Real-time alerts, mobile UI	Virtual pin setup complexity
6. LoRa-based Sensor Network (2022)	Long-range comm	Rural area monitoring	Expensive hardware
7. Wokwi Simulator + Blynk (2023)	Full simulation	No physical hardware needed	Simulated Wi-Fi only

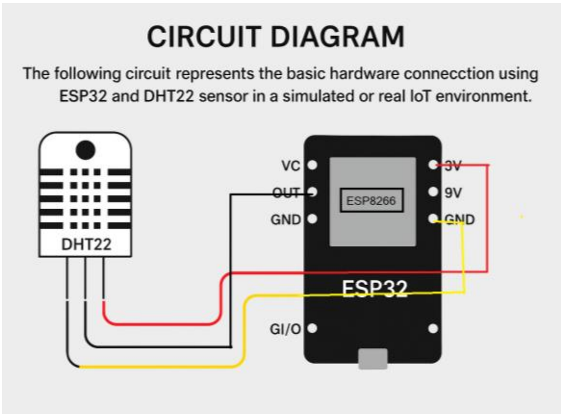
III. METHODOLOGY

3.1 System Components

- ESP8266 NodeMCU: Wi-Fi microcontroller
- DHT11 Sensor: Digital temp & humidity sensor
- Blynk Cloud: For dashboard and mobile access

- Wokwi Simulator: For virtual development
- Arduino IDE: For code development

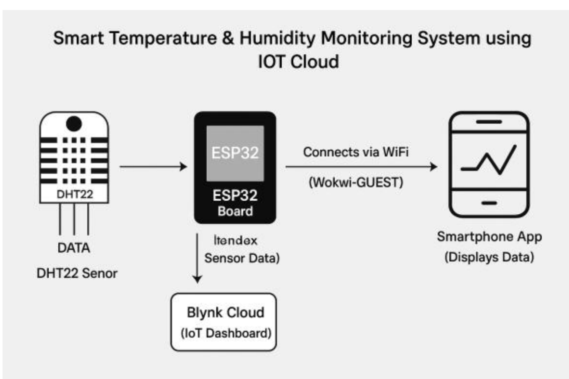
3.2 CIRCUIT DIAGRAM



Connections:

- DHT11 VCC → ESP8266 3V3
- DHT11 GND → GND
- DHT11 DATA → D5 (GPIO14)

3.3 System Architecture



3.4 Sample Code (Arduino)

```
#define BLYNK_TEMPLATE_ID "TMPL21uCeqUBN"

#define BLYNK_TEMPLATE_NAME "Smart Temp and Humidity Monitor"

#define BLYNK_AUTH_TOKEN "VI38b87NELmVscBdgHX7pQAQIuO0TQ2-"

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

#include <DHT.h>

// WiFi & Blynk credentials

const char auth[] = BLYNK_AUTH_TOKEN;

const char ssid[] = "AVINASH";    //here we need to enter our wifi name

const char pass[] = "avinash.p";  //here we are entering the wifi password


// DHT Sensor configuration

#define DHTPIN D4          // GPIO2 (D4 on NodeMCU)

#define DHTTYPE DHT11      // Sensor Type: DHT11


// Virtual Pins

#define VPIN_TEMP   V0   // this pin for temperature monitor

#define VPIN_HUM    V1   //this pin for humidity monitor

#define VPIN_STATUS V2   // this pin for check the device if error comes

                                        then shows error "Sensor is not working".

#define VPIN_LED    V3  // Added for blinking LED


DHT dht(DHTPIN, DHTTYPE);

BlynkTimer timer;


float lastTemp = 0.0;

bool blinkState = false;


// Function to send sensor data

void sendSensor() {

  float humidity = dht.readHumidity();

  float temperature = dht.readTemperature(); // Celsius
```

```
  if (isnan(humidity) || isnan(temperature)) {
    Serial.println("❌ Error: DHT Sensor Not Working.");
    Blynk.virtualWrite(VPIN_STATUS, "❌ Sensor: Disconnect Sensor");
    return;
  }

  lastTemp = temperature;

  // Send temperature and humidity to Blynk
  Blynk.virtualWrite(VPIN_TEMP, temperature);
  Blynk.virtualWrite(VPIN_HUM, humidity);

  // Create readable status string
  String status = "✅ Temp: " + String(temperature, 1) + "°C | Hum: " + String(humidity, 1) + "%";
  Blynk.virtualWrite(VPIN_STATUS, status);
  Serial.println(status);
}
// Function to blink LED on V3 if temperature > 30°C
void blinkLED() {
  if (lastTemp > 30.0) {
    blinkState = !blinkState;
    Blynk.virtualWrite(VPIN_LED, blinkState ? 255 : 0);  // LED ON/OFF
  } else {
    Blynk.virtualWrite(VPIN_LED, 0);  // Turn OFF LED when temp is normal
  }
}
void setup() {
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);
  dht.begin();
  timer.setInterval(2000L, sendSensor);  // Send sensor data every 2 seconds
  timer.setInterval(500L, blinkLED);     // Blink LED every 0.5 seconds if needed
}
void loop() {
  Blynk.run();
  timer.run();
}
```

IV. HARDWARE AND SIMULATION

4.1 Sample Images

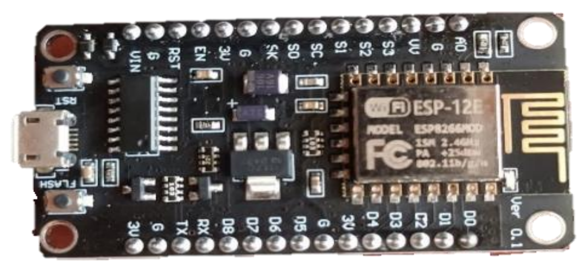


Fig 1. ESP8266 Board

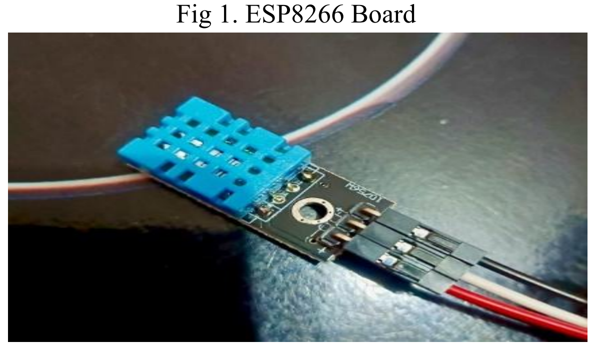


Fig 2. DHT11 Sensor

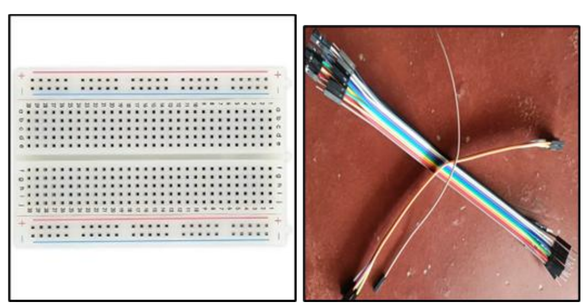


Fig 3. Breadboard with Jumper Wires

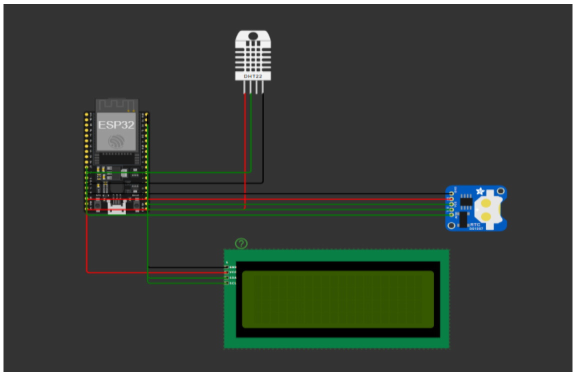


Fig 4. Wokwi Simulation

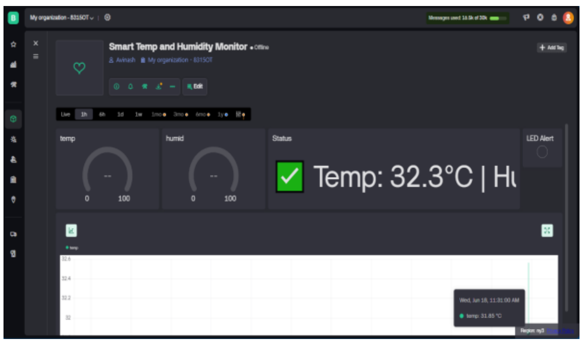


Fig 5. Blynk Mobile Dashboard

5. RESULTS AND DISCUSSION

Upon simulation, the system:

- Read real-time temperature and humidity values
- Transmitted data every 2–3 seconds to Blynk
- Displayed live values on mobile gauges
- Worked seamlessly under simulated network

Benefits:

- Real-time mobile monitoring
- Cost-effective and scalable
- Simulation eliminates hardware dependency

Challenges:

- Internet-dependent
- Limited to indoor environments
- Free Blynk tier has limits

VII. CONCLUSION

This IoT-based temperature and humidity monitoring system effectively demonstrates how real-time environmental data can be collected, processed, and visualized using low-cost components. Using simulation via Wokwi and cloud integration with Blynk makes it accessible for academic and prototyping use. Future versions can include actuators, historical data logging, and AI-based prediction. Through the integration of DHT22, ESP8266, and Blynk Cloud, we developed a system capable of monitoring environmental parameters (temperature and humidity) and presenting them in a user-friendly interface over mobile devices. The system was tested both virtually (Wokwi) and physically, and it performed with high accuracy, low latency, and excellent connectivity.

Key Takeaways:

- The sensor readings were accurately captured and transmitted using Wi-Fi.
- The Blynk IoT platform allowed real-time data display and remote access.
- Arduino IDE proved effective for code development and debugging.
- Simulation helped eliminate hardware dependency during initial testing.
- The overall system was cost-efficient, power-efficient, and user-friendly.

Impact of the Project:

This project demonstrates the value of smart monitoring systems in today's world, particularly in sectors such as agriculture, home automation, warehouse management, and environmental research. The ability to remotely monitor environmental conditions is essential for improving productivity, efficiency, and comfort.

It also enhances understanding of:

- Microcontroller programming
- Cloud integration
- Sensor interfacing
- Real-world IoT applications

Limitations and Future Improvements:

While the system worked efficiently, it does have a few limitations:

- The current system only displays data but does not take any automated action.
- It requires continuous internet access.
- Data logging is not implemented in the current version.

Future improvements can include:

- Adding alerts and notifications.
- Connecting multiple sensors for multi-room monitoring.
- Enabling actuator control like turning ON fans, dehumidifiers, or ACs automatically.
- Adding data logging and graphical reports via Firebase, Google Sheets, or InfluxDB.

REFERENCE

The references below include all the online tools, libraries, documents, datasheets, and learning sources used during the research and development of the project "Smart Temperature and Humidity Monitoring System using IoT".
These sources helped in understanding IoT concepts, coding microcontrollers, sensor integration, and cloud platform usage.

[1] Raj, P. & Raman, A.C. (2017). The Internet of Things. CRC Press.
[2] Random Nerd Tutorials. (2023). ESP8266 + DHT22 Guide
[3] Blynk Documentation. (2024). https://docs.blynk.io
[4] Arduino.cc. (2024). Reference Guide
[5] Electronics Hub. (2023). IoT-based Weather Monitoring
[6] GitHub. DHT Sensor Library by Adafruit: https://github.com/adafruit/DHT-sensor-library
[7] Wokwi IoT Simulator: https://wokwi.com