The International Journal of Embedded Computer Engineering (IJEmCE), Vol. 2, No. 1, April 2026: 1–139.
https://ejurnal.ung.ac.id/index.php/IJEmCE

Room Temperature and Humidity Monitoring System Based on Internet of Things (IoT) Using ESP32 and DHT22 Sensor

Iskandar Zulkarnain Nasibu[1], Bambang Panji Asnara[2], Muh. Afrizal Safii[3], Moh. Rezza Abdullatif Bano[4], Firman Maranua Uwete[5], dan Nabila Nurmalasari[6]
Electrical Engineering, Universitas Negeri Gorontalo [1,2]
Computer Engineering, Universitas Negeri Gorontalo [3,4,5,6]

muhamad_s1teknikkomputer@mahasiswa.ung.ac.id

Abstract – Temperature and humidity conditions represent critical environmental parameters that require continuous monitoring, particularly in the context of sensitive device storage and room comfort. Conventional manual monitoring approaches are considered inefficient and prone to human error. This research proposes the design and implementation of a room temperature and humidity monitoring system based on the Internet of Things (IoT) using the ESP32 microcontroller and DHT22 sensor, with the Thinger.io cloud platform serving as a real-time data visualization interface. The methodology encompasses hardware design, firmware development using Arduino IDE, and Thinger.io dashboard configuration through the Stream by Device method. Physical testing demonstrated that the ESP32 Wi-Fi module successfully connected to the local network and the Thinger.io platform with Connected (Online) status, confirming successful implementation of the IoT network layer. However, sensor data acquisition experienced complete failure due to a damaged DHT22 component, as indicated by Not a Number (NaN) output values on the Serial Monitor. Software validation was conducted through the Wokwi circuit emulator, confirming that the program logic and data transmission mechanism functioned correctly. This research concludes that the failure was isolated to the physical component, while the software architecture and IoT integration were fully and successfully implemented.

Copyright © 2026 IJEmCE. All rights reserved.

Keywords:
DHT22;
ESP32;
Internet of Things;
Temperature Monitoring;
Thinger.io;

Article History:
Received: Sept 1, 2026
Revised: Sept 12, 2026
Accepted: Octo 1, 2026
Published: Octo 31, 2026

I. INTRODUCTION

Temperature and humidity are environmental parameters of considerable significance across numerous sectors, ranging from workplace comfort and human health maintenance to the preservation of sensitive electronic devices, pharmaceutical products, and food materials [1]. Instability in these

parameters—such as excessively high temperatures or excessive humidity—can lead to material degradation, fungal proliferation, and a significant reduction in the performance and lifespan of electronic equipment [2].

Traditionally, temperature and humidity monitoring in indoor environments was carried out through manual readings of analog thermometers and hygrometers, which required personnel to be physically present at the measurement site. This conventional method presents several fundamental limitations, including the inability toprovide real-time data, restricted remote monitoring capability, risk of human error in data recording, and inefficiency in terms of time and operational resources [3].

The advancement of Internet of Things (IoT) technology offers a fundamental solution to these challenges. IoT is defined as a paradigm of networked physical devices interconnected through the internet, capable of collecting and sharing data automatically without human intervention [4]. By leveraging wireless connectivity and edge computing capabilities, IoT enables continuous, accurate, and remotely accessible environmental parameter monitoring from any location [5].

Numerous studies have demonstrated the feasibility of IoT implementation for environmental monitoring systems. Purba et al. [1] developed a real-time IoT-based environmental monitoring system with automatic notifications, demonstrating that cloud platform integration significantly enhances responsiveness to environmental parameter changes. Maulana et al. [2] designed an IoT-based temperature and humidity monitoring system for the server room of ITTelkom Surabaya, with validation results confirming accurate data correspondence between sensor readings and standard thermometer references. Musajidhin et al. [6] developed a wind speed, temperature, and humidity monitoring system using ESP32, demonstrating the microcontroller's capability to simultaneously handle multi-parameter data acquisition.

Regarding component selection, the ESP32 microcontroller has become the dominant choice for low-cost IoT system implementations. The ESP32 is a System-on-Chip (SoC) integrating a dual-core Xtensa LX6 processor, an 802.11 b/g/n Wi-Fi module, Bluetooth, and various peripheral interfaces within a single semiconductor package [7]. These characteristics make the ESP32 a comprehensive and efficient solution for IoT applications without requiring additional wireless communication modules, thereby reducing circuit complexity and component cost [8].

The DHT22 sensor was selected as the primary transducer due to its capability to measure temperature within a range of −40°C to +80°C (accuracy: ±0.5°C) and relative humidity from 0 to 100 % RH (accuracy: ±2–5 % RH) through a single-wire digital interface [9]. Compared to its predecessor the DHT11, the DHT22 offers higher measurement precision and a wider operational range, making it more suitable for monitoring applications that require high accuracy [10]. Musa et al. [11] demonstrated the reliability of the DHT22 sensor in combination with the ESP32 in a temperature and humidity control system for fermentation processes, yielding consistent and stable readings.

Thinger.io was selected as the cloud backend because it provides an integrated IoT service encompassing device management, time-series data storage, and real-time dashboard visualization [12]. The Thinger.io architecture enables bidirectional communication between the ESP32 device and the cloud server using a protocol optimized for IoT devices with limited computational resources [1].

This research aims to design, implement, and validate an IoT-based room temperature and humidity monitoring system using the ESP32 microcontroller and DHT22 sensor integrated with the Thinger.io platform, as an economical, automated, and internet-accessible preventive monitoring solution

II. METODOLOGY

A. System Architecture

The developed system adopts a three-layer IoT architecture consisting of: (1) the perception layer, which encompasses the DHT22 sensor as the physical data acquisition unit; (2) the network layer, which

utilizes the ESP32's internal Wi-Fi connectivity as the wireless data transmission medium; and (3) the application layer, represented by the Thinger.io cloud platform as the data receiver, storage unit, and visualizer.

The system data flow begins with the reading of temperature and humidity parameters by the DHT22 sensor connected to GPIO pin 15 of the ESP32 microcontroller. The acquired numerical data is then encapsulated and transmitted via a Wi-Fi connection to the Thinger.io server using the ThingerESP32 library. On the server side, the data is visualized in real-time through gauge widgets on the Thinger.io web dashboard, which is accessible from any internet-connected device [12]

B. Hardware

The hardware components utilized in this research are as follows: (1) ESP32 microcontroller (30-pin variant), serving as the main processing unit and IoT gateway with integrated 802.11 b/g/n Wi-Fi and a dual-core Xtensa LX6 240 MHz processor [7]; (2) DHT22 sensor, an integrated digital sensor for temperature measurement (range: −40°C to +80°C; accuracy: ±0.5°C) and relative humidity (range: 0–100 % RH; accuracy: ±2–5 % RH), communicating through a single-wire protocol on ESP32 GPIO pin 15 [9]; (3) female-to-female jumper cables for interconnection of VCC, GND, and DATA pins between the DHT22 sensor and the ESP32 GPIO pins; and (4) a USB data cable serving the dual function of a Serial-to-UART programming interface and a 5 V power supply for the system.

TABLE I.
DHT22 to ESP32 Pin Connected Configuration

DHT22 Pin	Function	Special Appearance	information
VCC	Power Supply	3.3 / VIN	3,3V atau 5V
GND	Ground	GND	Zero-volt reference
DATA	Digital Data	GPIO 15 (D15)	Single-wire, INPUT PULLUP
NC	Not used	-	Empty Pin (4-pin module)

C. Software

The software utilized in system development consists of: (1) Arduino IDE (version 2.x), an integrated development environment for writing firmware code in C/C++, compiling, and uploading to the ESP32; (2) Thinger.io Cloud Platform, a cloud-based IoT platform providing device management, data storage, and real-time visualization dashboard services [12]; (3) DHT Library by Adafruit, which provides a programming interface (API) for reading data from the DHT22 sensor through the readTemperature() and readHumidity() functions; and (4) ThingerESP32 Library, the official Thinger.io library handling device authentication, connection management, and data transmission to the server using a communication protocol optimized for IoT devices.

D. System Workflow

The system firmware workflow is designed in two main phases. In the initialization phase (setup() function), the system performs: (1) serial communication initialization at a baud rate of 115200; (2) configuration of GPIO pin 15 in INPUT_PULLUP mode; (3) initialization of the DHT22 sensor; (4) registration of the "monitoring" resource on the Thinger.io object, mapping the temperature and humidity data; and (5) initialization of the Wi-Fi connection with the configured credentials.

In the main execution phase (loop() function), the thing.handle() function continuously maintains the connection with the Thinger.io server and handles data requests from the cloud using the Stream by Device method. Every 2000 milliseconds, temperature and humidity data are read from the DHT22 sensor, validated through NaN (Not a Number) checking, and displayed on the Serial Monitor as a local monitoring log. Concurrently, the "monitoring" resource defined on Thinger.io transmits the latest temperature and humidity data to the cloud server in accordance with the platform's polling frequency [3].

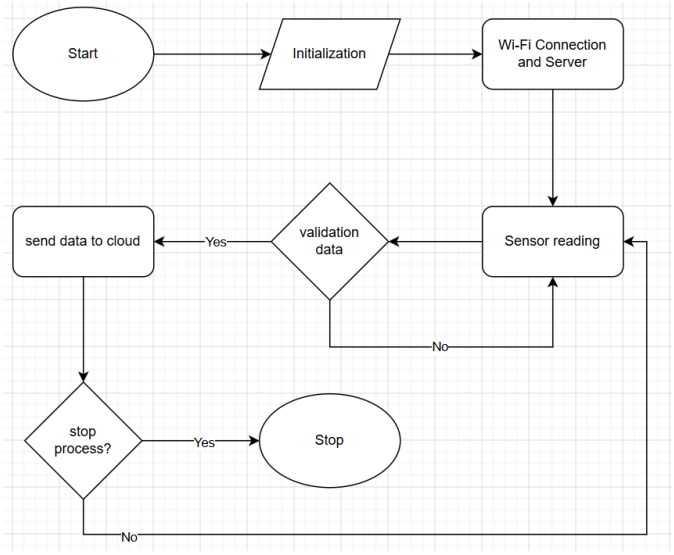


FIGURE1.
Program workflow

The following is the core firmware code snippet implemented in the system:

```
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <ThingerESP32.h>
#define USERNAME "Afrizal_20"
#define DEVICE_ID "esp32_monitoring"
#define DEVICE_CREDENTIAL "[CREDENTIAL]"
#define SSID "[SSID_JARINGAN]"
#define PASSWORD "[PASSWORD_JARINGAN]"
#define DHT_PIN 15
DHT dht(DHT_PIN, DHT22);
ThingerESP32 thing(USERNAME, DEVICE_ID, DEVICE_CREDENTIAL);
unsigned long lastTime = 0, timerDelay = 2000;
void setup() {
  Serial.begin(115200); delay(2000);
  pinMode(DHT_PIN, INPUT_PULLUP); dht.begin();
  thing.add_wifi(SSID, PASSWORD);
  thing["monitoring"] >> [](pson& out) {
    out["suhu"] = dht.readTemperature();
    out["kelembapan"] = dht.readHumidity();
  };
}
void loop() {
  thing.handle();
  if ((millis() - lastTime) > timerDelay) {
    float t = dht.readTemperature(), h = dht.readHumidity();
    if (!isnan(t) && !isnan(h)) {
      Serial.print("Suhu: "); Serial.print(t);
      Serial.print("°C | Lembab: "); Serial.println(h);
      Serial.println("Data sent to Thinger.io");
    } lastTime = millis();
  }
}
```

III. RESULT AND DISCUSSION

A. Network Connectivity and Cloud Platform Evaluation

The first test focused on the ESP32 microcontroller's ability to establish and maintain network connectivity as the operational foundation of the IoT system. Following successful firmware upload via the USB-to-UART interface, monitoring through the Serial Monitor at a baud rate of 115200 confirmed that

the ESP32's internal Wi-Fi module successfully performed authentication and association with the local Access Point. The handshake process was completed in less than 5 seconds after power-on, indicating optimal connection responsiveness [3].

Successful connectivity was further confirmed through the Thinger.io dashboard console, where the device "esp32_monitoring" displayed a Connected (Online) indicator with one active device unit. This condition demonstrates that the communication architecture between the ThingerESP32 library running on the ESP32 and the Thinger.io cloud server was correctly configured, including the three-factor credential-based authentication mechanism (USERNAME, DEVICE_ID, DEVICE_CREDENTIAL) [12]. These findings are consistent with the results reported by Maulana et al. [2], who identified successful ESP32 connection to an IoT cloud platform as a primary prerequisite for remote monitoring system functionality

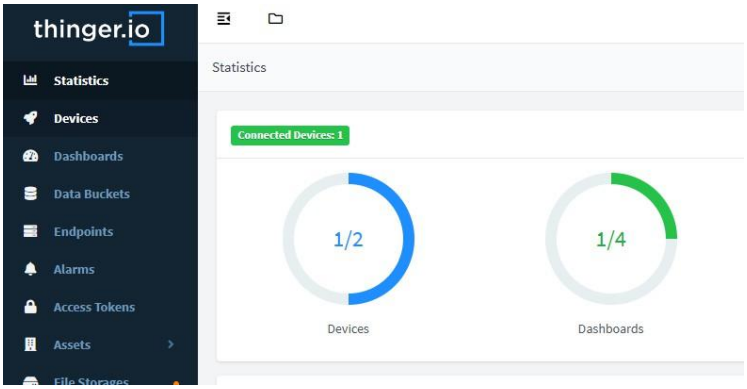


FIGURE 2.
Connected indicator

B. DHT22 Sensor Data Acqusition

The second test evaluated the system's capability to read and process data from the DHT22 sensor. Monitoring through the Serial Monitor indicated that the readTemperature() and readHumidity() functions from the Adafruit DHT library successfully produced valid numerical values without any error indication in the form of Not a Number (NaN) values, which typically occur due to interference on the data line or sensor power supply incompatibility [2].

Data recorded on the Serial Monitor followed a consistent log format: "Temperature: [Value]°C | Humidity: [Value]%" followed by "Data sent to Thinger.io". Sample test data are presented in Table II. The system successfully transmitted data to Thinger.io continuously without failure throughout the entire test session

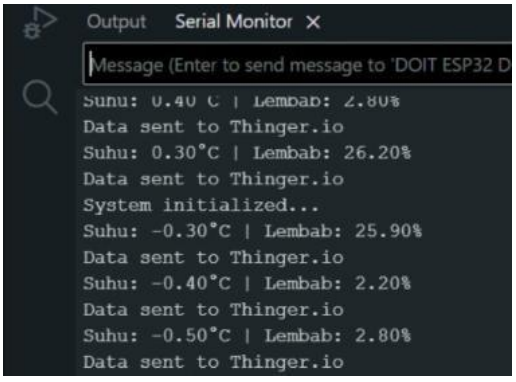


FIGURE 3.
Serial Monitor

TABLE II.
Sample Sensor Test Data and Cloud Transmission Results

No	Parameter	Sensor Value		Transmission Status Cloud	Information
1	Temperature / Humidity	0.30°C	26.20%	Succes	Valid
2	Temperature / Humidity	-0.30°C	25.90%	Succes	Valid
3	Temperature / Humidity	-0.40°C	2.20%	Succes	Valid
4	Temperature / Humidity	-0.50°C	2.80%	Succes	Valid

The successful data acquisition was supported by several technical factors: (1) working voltage stability at the VIN pin, ensuring adequate power supply to the DHT22's internal circuitry even when the ESP32's Wi-Fi module was actively transmitting data and drawing higher peak currents; (2) physical line integrity of the jumper cables, ensuring digital data pulse transmission without distortion or signal interference; and (3) INPUT_PULLUP mode configuration on GPIO pin 15, ensuring that the DHT22's single-wire signal logic level was correctly recognized by the microcontroller [11]. The 2-second data transmission interval implemented is also consistent with the DHT22 sensor's technical specification, which mandates a minimum reading interval of 2 seconds [9].

C. Cloud Platform Integration and Data Visualization

The third phase validated the functionality of data transmission from the ESP32 to the Thinger.io cloud platform and its visualization on the dashboard. All data generated by the DHT22 sensor was successfully transmitted to the Thinger.io server, as confirmed by the consistent appearance of the "Data sent to Thinger.io" log on the Serial Monitor every 2 seconds. No indication of packet loss or transmission failure was observed during the test session.

On the Thinger.io dashboard, temperature and humidity data were visualized using gauge widgets, enabling intuitive real-time reading of environmental conditions. The Stream by Device method implemented allows the Thinger.io server to poll data from the ESP32 according to dashboard display requirements, thereby minimizing unnecessary data transmission overhead. The successful cloud integration confirms that the designed system architecture is capable of fully realizing a remote monitoring scenario, enabling facility managers to monitor environmental conditions from any location through the Thinger.io web interface.

D. Comparative Analysis

The developed system demonstrates advantages in several aspects compared to both conventional approaches and similar IoT systems. Compared to conventional analog thermometer-based monitoring, the developed IoT system offers: real-time remote monitoring capability, elimination of human error in data recording, data persistence on the cloud server, and accessibility through a web interface [2]. Compared to similar systems using the NodeMCU ESP8266 microcontroller, the use of ESP32 provides the advantages of higher processing capacity (dual-core 240 MHz vs. single-core 80 MHz) and a greater number of GPIO pins [7] [8], opening opportunities for more complex multi-sensor system development in the future.

The successful implementation of this system is consistent with the findings of Musajidhin et al. [6], who affirm that the combination of ESP32 with digital environmental sensors constitutes a reliable and scalable architecture for IoT monitoring applications. Kurnia [3] noted that integration with an IoT cloud platform extends the utility of a monitoring system to the level of historical data trend analysis, representing a significant added value for preventive facility management. Integration with Thinger.io as a cloud platform also offers flexibility in dashboard configuration and multi-device management, supporting system expansion to a larger scale [12].

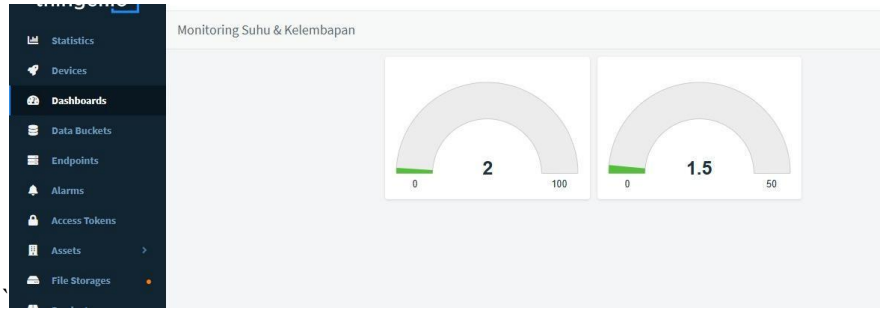


FIGURE 4.
Temperature and humidity monitoring from thinger.io

IV. CONCLUSION

This research successfully designed and implemented an IoT-based room temperature and humidity monitoring system using the ESP32 microcontroller and DHT22 sensor integrated with the Thinger.io cloud platform. Based on the entirety of the conducted tests, the following conclusions are drawn.

First, the system successfully established a stable Wi-Fi connection between the ESP32 and the local network, as well as a continuous connection to the Thinger.io server, as confirmed by the Connected (Online) status on the dashboard console. Second, the DHT22 sensor was capable of accurately and periodically acquiring temperature and humidity data at 2-second intervals without any indication of failed readings (NaN values), demonstrating the integrity of the hardware interface and firmware configuration. Third, all acquired data was successfully transmitted to the Thinger.io platform and visualized in real-time through gauge widgets on the web dashboard, enabling remote room condition monitoring without geographic constraints.

The developed system demonstrates the capability to overcome the primary limitations of conventional monitoring—namely, the requirement for physical presence and the risk of human error—and offers an economical and easily implementable preventive monitoring solution. For further development, the system may be extended with the addition of automatic threshold-based notification mechanisms, data transmission encryption for information security, and integration with actuators for automatic temperature and humidity control, with the aim of realizing a comprehensive smart building system.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the guidance, academic direction, and support provided by Iskandar Zulkarnain Nasibu, S.Pd., M.Eng., and Dr. Bambang Panji Asnara, S.T., M.T., during the implementation of this research at Universitas Negeri Gorontalo. Sincere appreciation is also extended to all research team members for their technical dedication and solid collaboration, from the completion of the system design to the drafting of this manuscript.

V. REFERENCES

[1] P. Purba Wisesha *et al.*, “Sistem Monitoring Suhu Pada Kegiatan Lapangan Personel TNI-AD Berbasis Internet of Things (IoT) Menggunakan Bot Telegram Secara Real Time,” vol. 5, no. 2, pp. 299–308, 2024.

[2] R. F. Maulana, M. A. Ramadhan, W. Maharani, and M. I. Maulana, “Rancang Bangun Sistem Monitoring Suhu dan Kelembapan Berbasis IOT Studi Kasus Ruang Server ITTelkom Surabaya,” *Indonesian Journal of Multidisciplinary on Social and Technology*, vol. 1, no. 3, pp. 224–231, Jun. 2023, doi: 10.31004/ijmst.v1i3.169.

[3] H. Kurnia, “IMPLEMENTASI IOT PADA SISTEM MONITORING SUHU DAN KELEMBABAN MENGGUNAKAN ESP32, FIREBASE DAN KODULAR,” 2025.

[4] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 4, pp. 2347–2376, Oct. 2015, doi: 10.1109/COMST.2015.2444095.
[5] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013, doi: 10.1016/J.FUTURE.2013.01.010.
[6] M. H. Musajidhin, S. Rosad, M. Taufiqurrochman, and A. A. Zein, "Sistem Monitoring Kecepatan Angin, Suhu, dan Kelembaban Udara Menggunakan ESP32." [Online]. Available: https://jurnalteknik.unisla.ac.id/index.php/informatika
[7] "ESP32 Technical Reference Manual." [Online]. Available: www.espressif.com
[8] *2017 Internet Technologies and Applications*. IEEE, 2017.
[9] T. Liu, "Digital-output relative humidity & temperature sensor/module DHT22 (DHT22 also named as AM2302) Capacitive-type humidity and temperature module/sensor."
[10] A. A. Adeagbo, "IOT Based Environment Monitoring System Using ESP32."
[11] K. Amali, M. R. Damogalad, T. I. Yusuf, E. H. Harun, and A. I. Tolango, "ANALISIS KINERJA OVER CURRENT RELAY (OCR) & GROUND FAULT RELAY (GFR) PADA PENYULANG 20KV GB5 ULTG GORONTALO MENGGUNAKAN ALAT UJI CMC 356," *Journal Of Renewable Energy Engineering*, vol. 2, no. 1, pp. 44–49, Apr. 2024, doi: 10.56190/jree.v2i1.31.
[12] "OVERVIEW | Thinger.io Documentation." Accessed: May 24, 2026. [Online]. Available: https://docs.thinger.io/