IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 3, May 2025
Impact Factor: 7.67

Smart Weather Monitoring System using Node MCU

Rudra Pratap Tyagi, Lakshaya Sharma, Mr. Isharar Ahamad, Mr. Brijesh Kumar Mishra
Department of Computer Science And Engineering (IOT)
Raj Kumar Goel Institute of Technology, Ghaziabad, India

Abstract: *The demand for accurate and real-time environmental data is growing rapidly due to its vital importance in areas such as agriculture, disaster management, urban planning, and climate research. Traditional weather monitoring systems, while effective, are often expensive, bulky, and lack the capability for real-time data transmission and remote accessibility. With the rise of the Internet of Things (IoT), it has become possible to design more compact, affordable, and efficient systems for environmental monitoring.*

This paper presents the design and development of a Smart Weather Monitoring System based on the NodeMCU ESP8266 microcontroller, which features integrated Wi-Fi capabilities. The system employs a range of sensors including the DHT11 for temperature and humidity, the BMP180 for atmospheric pressure, and a rain sensor to detect precipitation levels. These sensors continuously collect environmental data which is then transmitted over Wi-Fi to a cloud-based platform (ThingSpeak), where it can be accessed in real time through a web interface..

Keywords: real-time environmental

I. INTRODUCTION

1.1 Background and Motivation

In the era of climate change and unpredictable weather patterns, the need for accurate, real-time weather monitoring has become more pressing than ever. Traditional weather monitoring stations, although reliable, are expensive and geographically limited. With the advent of the Internet of Things (IoT), low-cost microcontrollers like the NodeMCU provide an accessible way to deploy scalable, real-time weather monitoring systems. The integration of IoT into meteorological monitoring empowers communities to gather localized weather data and enhance environmental awareness and disaster preparedness.

1.2 Problem Statement

Conventional weather stations are not only costly but also sparse in coverage, especially in rural and underdeveloped areas. These systems often lack real-time communication capabilities and are challenging to maintain. Moreover, manual data collection is prone to errors and delays. Hence, there is a critical need for an automated, real-time, and low-cost weather monitoring system that can be easily deployed in diverse geographic locations.

1.3 Objectives

- To design and implement a low-cost, real-time weather monitoring system using NodeMCU.
- To interface multiple environmental sensors (temperature, humidity, pressure, and rainfall) with NodeMCU.
- To transmit sensor data to a cloud-based platform for real-time monitoring and analysis.
- To evaluate the accuracy, reliability, and performance of the system.

1.4 Paper Organization

The paper is structured as follows: Section 2 reviews existing literature and identifies the gaps in current systems. Section 3 details the proposed system design, including both hardware and software components. Section 4 describes the implementation process, including hardware integration and data communication. Section 5 discusses the experimental setup and presents the results. Section 6 provides a discussion of findings, limitations, and possible improvements. Finally, Section 7 concludes the paper and outlines future research directions.

II. LITERATURE REVIEW

2.1 Traditional Weather Monitoring Systems

Traditional weather stations rely on analog or digital sensors connected to centralized data loggers. These systems are often maintained by governmental or academic institutions and require significant investment in infrastructure and skilled manpower. Data is typically recorded manually or logged into standalone systems, limiting real- time accessibility and scalability.

2.2 IoT in Environmental Monitoring

IoT has revolutionized environmental monitoring by enabling real-time data acquisition, transmission, and analysis. Low-cost microcontrollers like Arduino and NodeMCU, coupled with wireless communication protocols such as Wi-Fi, Zigbee, and LoRa, provide a foundation for distributed sensing networks. Research by Sharma et al. (2021) highlights how IoT can be used to monitor not only weather but also air quality, water quality, and agricultural parameters.

2.3 Existing IoT-Based Weather Monitoring Systems

Numerous studies and prototypes have emerged demonstrating IoT-based weather monitoring systems. For example, Patel et al. (2019) implemented a system using Raspberry Pi and BME280 sensors, while Khan et al. (2020) utilized NodeMCU and DHT11 sensors with cloud integration using ThingSpeak. These systems provide accurate data with reduced latency, though challenges remain in terms of power management and long-term deployment.

2.4 Gaps and Challenges in Current Approaches

While many IoT-based systems are promising, several issues need addressing:

- Power efficiency: Most systems rely on continuous power sources, limiting their use in remote areas.
- Scalability: Existing solutions often lack modularity, making expansion difficult.
- Environmental resilience: Hardware must be rugged and weatherproof for field deployments.
- Data security: With cloud-based solutions, data integrity and access control are crucial.

The proposed system in this paper seeks to address these gaps by providing a modular, scalable, and energy-efficient architecture suitable for a wide range of applications.

III. SYSTEM DESIGN

3.1 Overview of System Architecture

The smart weather monitoring system consists of sensor nodes, a microcontroller (NodeMCU ESP8266), and a cloud-based server. The sensors collect real-time weather data and send it to the NodeMCU, which processes and transmits the data via Wi-Fi to an online IoT platform. The platform stores, visualizes, and allows remote access to data.

3.2 Hardware Requirements

3.2.1 NodeMCU ESP8266

NodeMCU is a low-cost microcontroller with built-in Wi-Fi capability based on the ESP8266 chip. Its features include:

Supports Lua and Arduino IDE

Compared to other microcontrollers like Arduino Uno or Raspberry Pi, NodeMCU offers superior wireless connectivity at a lower cost. Its smaller size and energy efficiency make it ideal for compact, long-term deployment in remote weather monitoring scenarios.

Implementation

3.2.1.1 Sensor Integration with NodeMCU

The implementation began with connecting the various sensors to the NodeMCU. The DHT11 sensor was connected to a digital pin (e.g., D4), the BMP180 was connected via the I2C protocol using SDA and SCL pins, and the YL-83 rain sensor was connected to an analog input. Pull-up resistors and proper voltage regulation (5V to 3.3V) were applied where necessary to ensure signal compatibility and stable operation.

3.2.1.2 Programming the Microcontroller

The Arduino IDE was used to write and upload the firmware to the NodeMCU. Libraries such as "DHT.h," "Adafruit_BMP085.h," and "ESP8266WiFi.h" were included to interface the sensors and manage Wi-Fi communication. The program logic includes initializing the sensors, reading values at regular intervals, and sending the data to the cloud.

3.2.1.3 Data Transmission to the Cloud

Data is transmitted to ThingSpeak using HTTP GET requests. The Wi-Fi credentials and ThingSpeak API key were embedded in the code. Once Wi-Fi is connected, sensor readings are formatted as parameters and sent to the ThingSpeak server. Each parameter corresponds to a field in a ThingSpeak channel, enabling organized storage and visualization.

3.2.1.4 Real-time Visualization

On the ThingSpeak dashboard, widgets such as line graphs, gauges, and tables were configured to visualize temperature, humidity, pressure, and rainfall data. This web- based interface allows users to monitor weather conditions remotely from any internet- connected device.

3.2.1.5 Power Supply and Deployment

The system is powered using a solar panel connected to a lithium-ion battery with a charging controller circuit. A voltage regulator ensures a steady 3.3V supply to the NodeMCU. The entire unit, including sensors, was enclosed in a weatherproof casing to allow outdoor deployment. Vents and protective filters were incorporated to maintain sensor accuracy.

3.2.1.6 Optional Features

Additional features such as an LCD display for local readout and buzzer alerts for extreme conditions (e.g., high temperature or heavy rainfall) were implemented in select deployments. These features enhance usability in areas with limited internet access.

3.2.1.7 Data Logging and Analysis

For long-term monitoring, data is periodically backed up from ThingSpeak to Google Sheets using an integration tool such as IFTTT or MATLAB Analysis scripts. This allows further data analysis, trend identification, and statistical forecasting.

3.2.2 DHT11 Temperature and Humidity Sensor

The DHT11 sensor measures temperature and humidity. It is easy to interface with microcontrollers and provides reasonably accurate readings suitable for basic weather monitoring.

3.2.4 Rain Sensor (YL-83)

The YL-83 rain sensor detects rainfall intensity by measuring conductivity on its surface. It helps determine precipitation levels and acts as a simple water presence detector.

IJARSCT

International Journal of Advanced Research in Science, Communication and Technology

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

ISSN: 2581-9429 Volume 5, Issue 3, May 2025 Impact Factor: 7.67

3.2.5 LCD Display (Optional)

An optional 16x2 LCD display can be added for on-site data visualization without requiring internet access.

3.3.1 Arduino IDE

The Arduino IDE is used to program the NodeMCU using C/C++. It supports various libraries for sensor interfacing and cloud communication.

3.3.2 Blynk/ThingSpeak/Google Firebase

These platforms are used for real-time data visualization and storage. ThingSpeak is commonly used for academic and prototype purposes.

3.3.3 Embedded C/C++

The system is programmed using C/C++, ensuring efficient resource management and control.

IV. SENSORS AND THEIR DETAILS

Sensor	Parameters Measured	Voltage	Range
DHT11/DHT22	Temperature, Humidity	3.3–5V	0–50°C / -40–80°C, 20–90% RH
BMP180/BMP280	Barometric Pressure, Temperature	1.8–3.6V	300–1100 hPa, -40–85°C
MQ-135	Air Quality (NH_3, NO_x, CO_2, etc.)	5V	Variable
Rain Sensor	Rain presence and intensity	3.3–5V	Analog Output
LDR	Light Intensity	3.3–5V	Analog Output

4.1 System Functionality and Performance

The smart weather monitoring system was deployed in a semi-urban environment for a continuous period of 30 days. The system consistently collected and transmitted data every 5 minutes. Observations indicated that the NodeMCU handled multiple sensor readings efficiently with negligible delay in data transmission. Average latency for data upload to ThingSpeak was under 2 seconds. The sensors performed within expected accuracy margins: ±2°C for DHT11, ±4% RH for humidity, and ±1 hPa for BMP180.

4.2 Data Accuracy and Validation

To validate the accuracy of the collected data, sensor readings were compared against a nearby government weather station's data. Temperature readings had a deviation of

±1.8°C, while humidity readings showed a 3-5% variance. The pressure sensor showed a minor deviation of ±1.2 hPa. Rain detection was consistent with observable rainfall events, although it lacked quantitative measurement capabilities.

4.3 Visualization and Remote Accessibility

The ThingSpeak platform provided effective data visualization tools, enabling real-time tracking of weather metrics. Graphs, time-series plots, and threshold alerts enhanced the usability for end-users. The system could be accessed via smartphone or PC, offering mobility and convenience.

4.4 Energy Efficiency and Reliability

The solar-powered setup ensured uninterrupted operation during the test period, even with partial cloud cover. The energy harvested was sufficient to sustain the system for

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 3, May 2025
Impact Factor: 7.67

more than 72 hours without direct sunlight, proving its resilience for remote and off- grid deployment.

4.5 Cost Analysis

The overall cost of the prototype system was under $30 USD, including all sensors and the microcontroller. This makes it significantly more affordable compared to conventional weather stations, which may cost several hundred dollars.

4.6 Scalability and Expandability

The modular architecture of the system allows easy integration of additional sensors (e.g., wind speed, UV index, air quality) as required. The software design is scalable, with support for additional communication protocols such as MQTT and LoRa for extended range and low-power applications.

4.7 Challenges Encountered

Sensor calibration was necessary to improve accuracy, especially for the DHT11. Environmental factors such as dust and condensation affected rain sensor performance.

Wi-Fi connectivity was occasionally disrupted due to weak signal strength, suggesting the need for a stronger antenna or alternative protocols.

4.8 User Feedback

A group of test users including students and local farmers provided feedback. They found the interface intuitive and the data insightful, especially for planning agricultural activities and learning about local climate trends.

Below is an example of average daily sensor readings during the test period:

- Parameter Minimum Maximum Average
- Temperature (°C) 18.5 35.2 26.8
- Humidity (%) 42 86 64
- Pressure (hPa) 991 1018 1005
- Air Quality (ppm) 180 450 320
- Light Intensity (%) 5 98 62

Graphs generated on ThingSpeak displayed clear trends and validated the reliability of data collection. The system proved to be reliable and cost-efficient. Issues such as waterproofing and sensor calibration were identified and can be mitigated through hardware design improvements.

4.9 COST AND POWER ANALYSIS

The overall cost of the prototype system was under ₹850 approx, including all sensors and the microcontroller. This makes it significantly more affordable compared to conventional weather stations, which may cost several thousands of rupees .

4.10 SECURITY CONSIDERATIONS

As data is transmitted over Wi-Fi, it is essential to use secure protocols such as HTTPS for cloud communication. Basic network security measures such as SSID hiding and MAC filtering were implemented.

V. CONCLUSION AND FUTURE WORK

5.1 Conclusion

The smart weather monitoring system designed using NodeMCU has proven to be a reliable, scalable, and cost-effective alternative to traditional weather stations. The successful integration of temperature, humidity, pressure, and rainfall sensors with real- time cloud-based data visualization demonstrates the feasibility and utility of IoT in environmental monitoring. The system's low cost and ease of deployment make it particularly well-suited for rural, remote, and underdeveloped areas where traditional infrastructure is lacking. With a modular design and solar-powered operation, the system can be adapted for a wide range of geographic and climatic conditions.

The performance evaluation confirmed that the system operates with acceptable levels of accuracy and reliability. The use of ThingSpeak provided a user-friendly platform for real-time data access, visualization, and long-term logging. Overall, the project showcases the transformative potential of IoT in building smarter, more responsive environmental monitoring systems.

5.2 Future Work

While the current system provides core weather monitoring capabilities, several enhancements can be made to increase functionality, robustness, and user experience:

- Sensor Upgrade: Replace DHT11 with higher-accuracy sensors such as DHT22 or SHT31 for improved temperature and humidity readings. Incorporate quantitative rain gauges and wind sensors for broader meteorological coverage.
- Data Analytics and Prediction: Integrate AI/ML algorithms for predictive analytics and early warning systems based on historical trends.
- Mobile Application: Develop a dedicated mobile app for easier access and push notifications for weather alerts.
- Extended Communication Range: Implement LoRa or NB-IoT to support data transmission in remote locations with limited Wi-Fi access.
- Robust Housing: Design more durable and waterproof enclosures for extended outdoor usage.
- Energy Optimization: Explore deep sleep modes and alternative energy sources to further enhance power efficiency.
- Security Enhancements: Implement encrypted communication protocols and authentication to ensure data integrity and privacy.

REFERENCES

[1] M. Abdulrahman and M. Latif, "IoT-Based Weather Monitoring System Using NodeMCU," J. Sensor Netw., vol. 5, no. 2, pp. 45–52, 2020.
[2] R. Sharma and P. Gupta, "Smart Weather Monitoring System Using IoT," Int. J. Adv. Res. Electron. Commun. Eng., vol. 8, no. 3, pp. 10– 15, 2019.
[3] Arduino, "DHT11 Technical Specifications," [Online]. Available: https://www.arduino.cc
[4] Bosch Sensortec, "BMP280 Datasheet," [Online]. Available: https://www.bosch-sensortec.com
[5] SparkFun, "MQ135 Gas Sensor Datasheet," [Online]. Available: https://www.sparkfun.com
[6] NodeMCU Team, "NodeMCU Documentation," [Online]. Available: https://nodemcu.readthedocs.io
[7] ThingSpeak, "ThingSpeak Documentation," [Online]. Available: https://thingspeak.com
[8] K. Patel and A. Mehta, "Design and Implementation of IoT Based Weather Station," IJERT, vol. 10, no. 4, pp. 50–55, 2021.
[9] Espressif Systems, "ESP8266 Technical Reference," 2020.
[10] Blynk Inc., "Blynk IoT Platform," [Online]. Available: https://blynk.io
[11] S. Kumar et al., "Real-time Environmental Monitoring Using IoT," in Proc. IEEE Int. Conf. IoT, pp. 87–91, 2020.
[12] Y. Zhang and C. Lin, "Air Quality Monitoring Using Low-Cost Sensors," Environ. Monit. J., vol. 13, no. 2, pp. 78–84, 2019.
[13] D. Singh and R. Verma, "Sensor Fusion Techniques for Weather Monitoring," Sensors Appl., vol. 7, no. 1, pp. 60–66, 2020.
[14] A. Rao and J. Thomas, "Embedded System Based Weather Forecasting," J. Embedded Syst., vol. 6, no. 3, pp. 30–35, 2021.

[15] T. Mehmood and M. Ashfaq, "Weather Monitoring with Cloud Connectivity," Int. J. Comput. Sci. Netw. Secur., vol. 22, no. 1, pp. 100–107, 2022.

[16] S. Islam and R. Haque, "Sustainable Energy Approaches for IoT Devices," Smart Energy Syst., vol. 4, no. 1, pp. 22–30, 2022.

[17] K. Pandey and A. Roy, "Mobile-Based Monitoring for IoT Systems," Mob. Comput. Rev., vol. 9, no. 2, pp. 40–47, 2021.

[18] M. Li and H. Cheng, "Low Power Wireless Systems in IoT Applications," J. Embedded Technol., vol. 5, no. 4, pp. 88–95, 2023.

[19] World Meteorological Organization, "Best Practices for Weather Data Collection," [Online]. Available: https://public.wmo.int

[20] J. Brown and M. Ali, "IoT for Smart Agriculture," Agric. Eng. Today, vol. 65, no. 2, pp. 20–27, 2021