International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Wireless Weather Station With Data Logging To Cloud

[1]K.Mohanaprakash, [2]Jeevanantham V, [3]Karthick V, [4]Mugesh R, [5] Mohammad Sameer F

[1]Department of Electronics and communication engineering Excel engineering college(autonomous) komarapalayam, Namakkal. Associate Professor
[2345]Department of electronics and communication engineering Excel engineering college(autonomous) komarapalayam, Namakkal UG student

ABSTRACT :

This paper presents a comprehensive design and implementation of a wireless weather station system capable of real-time environmental monitoring and cloud-based data logging. The system integrates multiple sensors—temperature, humidity, barometric pressure, and rainfall—with an embedded microcontroller (such as ESP32) to perform periodic data acquisition. Sensor readings are processed locally to reduce noise and improve accuracy. Data transmission is achieved via a Wi-Fi module using the MQTT protocol, enabling efficient and low-latency communication with a cloud IoT platform.

The cloud infrastructure facilitates secure storage, real-time visualization, and remote access to weather data through web and mobile interfaces. This architecture supports scalability and enables long-term data analytics for environmental and agricultural applications. The system was validated through continuous operation, demonstrating reliable sensor performance and stable wireless communication under varying environmental conditions.

This approach leverages the advantages of IoT and cloud computing technologies to overcome limitations of traditional weather stations, such as manual data retrieval and limited accessibility. The proposed solution offers an effective tool for precise weather monitoring, with potential for integration into larger smart environmental monitoring networks and predictive weather modeling systems.

Keywords: Wireless Weather Station, Environmental Monitoring,IoT (Internet of Things), Cloud Data Logging, Real-Time Data Transmission, Sensor Networks, MQTT Protocol, Remote Weather Monitoring, Data Visualization, Embedded Systems.

1. INTRODUCTION

Weather monitoring plays a critical role in various sectors, including agriculture, environmental research, and disaster management. Traditional weather stations often rely on manual data collection or localized storage, limiting real-time access and scalability. Advances in Internet of Things (IoT) technology and cloud computing have enabled the development of wireless weather stations that can remotely collect, transmit, and store meteorological data. These systems offer improved accessibility, automation, and data management.

This paper proposes a wireless weather station that integrates multiple sensors with a microcontroller and wireless communication module to measure temperature, humidity, atmospheric pressure, and rainfall. The collected data is transmitted in real-time to a cloud platform, facilitating remote monitoring and long-term data storage. The use of cloud services allows for scalable data logging and visualization, enabling users to analyze weather patterns efficiently. The system aims to provide a cost-effective, reliable, and scalable solution for modern weather monitoring applications.

2. SYSTEM ARCHITECTURE

2.1 Hardware Components

1. **Temperature and Humidity Sensor (DHT22):**
The DHT22 sensor measures ambient temperature and relative humidity. It provides digital output signals that are easy to interface with microcontrollers. The sensor is favored for its accuracy and low power consumption, making it suitable for continuous environmental monitoring.

2. Barometric Pressure Sensor (BMP280):
This sensor measures atmospheric pressure with high precision, which is essential for weather prediction and altitude calculations. The BMP280 communicates via I2C or SPI protocols, allowing seamless integration with embedded systems.

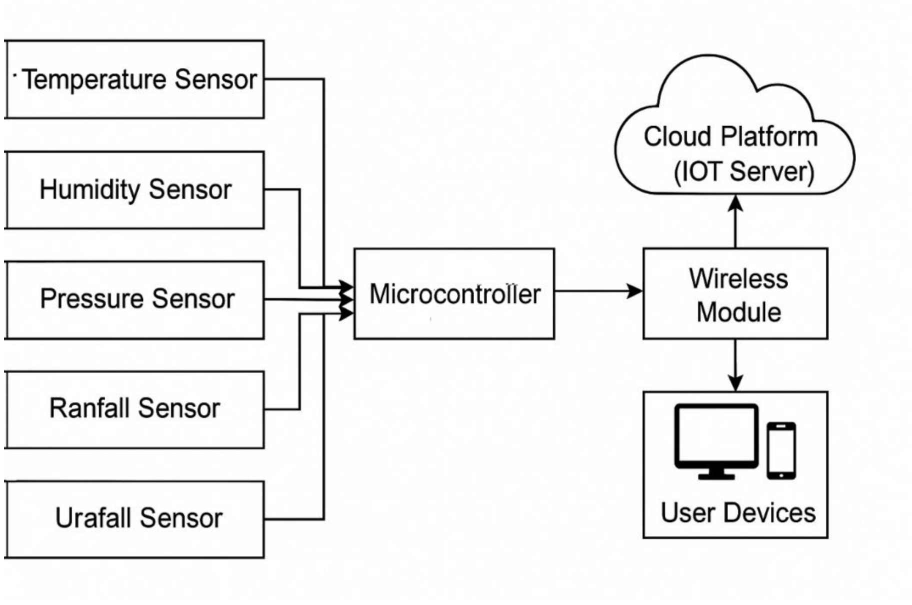

Fig. 1. System Block Diagram

3. Rainfall Sensor:

A tipping bucket or capacitive rain sensor detects precipitation levels by measuring the amount of rainfall over time. It converts physical rain collection into electrical signals that can be counted or interpreted by the microcontroller.

4. **Microcontroller** (ESP32 / Arduino):

The microcontroller acts as the central processing unit of the system. It reads data from the sensors, performs necessary filtering and conversion, and controls communication with the cloud server. The ESP32 is preferred for its built-in Wi-Fi capabilities and processing power.

5. Wireless Communication Module (Wi-Fi / GSM):

For transmitting data to the cloud, the system uses Wi-Fi or GSM modules. Wi-Fi is commonly integrated into microcontrollers like the ESP32, while GSM modules provide connectivity in areas without Wi-Fi. Communication protocols such as MQTT ensure efficient data transfer.

6. Power Supply:

A stable power source, often a battery with solar charging or a regulated DC supply, ensures continuous operation of the station, especially for outdoor deployments.

2.2 Software Components

The software system of the wireless weather station is composed of three primary components: embedded firmware, cloud services, and user interfaces, all working together to enable real-time environmental data acquisition, transmission, storage, and visualization.

1. Embedded Firmware

The embedded firmware runs on the microcontroller (e.g., ESP32 or Arduino). It manages sensor interfacing by periodically reading raw data from various sensors such as temperature, humidity, pressure, and rainfall sensors. The firmware applies signal conditioning and filtering algorithms to enhance data accuracy and reduce noise. It then formats the sensor readings into structured data packets suitable for transmission.

For communication, the firmware uses lightweight protocols like MQTT (Message Queuing Telemetry Transport) or HTTP to send data packets to the cloud platform over Wi-Fi or GSM networks. MQTT is preferred for its low bandwidth consumption and reliability in IoT applications. The firmware also includes error handling and reconnection logic to maintain stable wireless communication.

2. Cloud Services

The cloud component acts as the central data repository and processing hub. IoT cloud platforms such as AWS IoT Core, Google Firebase, or ThingSpeak receive incoming data streams from the weather station. These services provide scalable, secure storage and manage data persistence.

Cloud services also offer real-time analytics and visualization tools. Incoming data can be accessed through APIs for further processing or integration with external systems. Cloud platforms support data retention policies for historical analysis and facilitate event-based alerts or notifications based on specific weather conditions.

3. User Interface

User interfaces provide end-users with accessible visualization and interaction with weather data. These can be web dashboards or mobile applications that connect to the cloud platform using RESTful APIs or WebSockets.

The UI displays real-time weather parameters, graphical trends, and historical data charts. Advanced interfaces may offer configurable alerts, data export functions, and multi-station management for larger deployments. Responsive design ensures accessibility from various devices, promoting remote monitoring from anywhere with an internet connection.

3. IMPLEMENTATION

3.1 Data Acquisition

Environmental sensors continuously capture data such as temperature, humidity, pressure, and rainfall. The microcontroller periodically reads this sensor data at set intervals and applies basic filtering techniques to minimize noise and improve measurement accuracy.

3.2 Wireless Data Transmission

The processed data is transmitted wirelessly using Wi-Fi or GSM connectivity. Communication protocols like MQTT or HTTP are employed to efficiently and reliably send data packets from the microcontroller to the cloud platform.

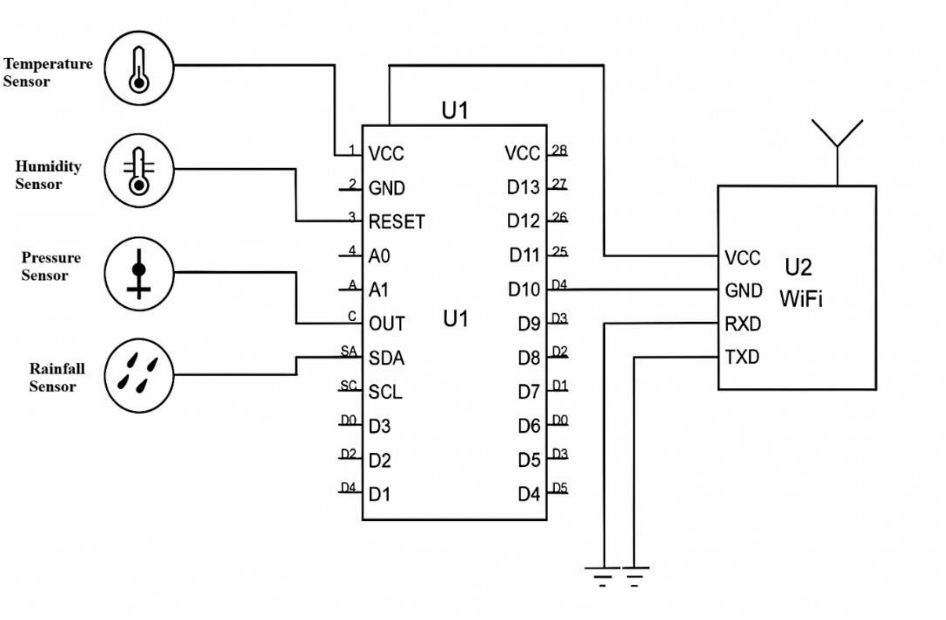

Fig.2 Hardware Connection Diagram (Circuit Diagram)

3.3 Cloud Data Logging and Visualization

Once received, the cloud platform securely stores the sensor data and offers APIs for easy access. The platform also provides visualization tools such as dashboards that display real-time trends, historical records, and alert notifications for effective weather monitoring.

4. Detailed Methodology

1. Sensor Selection and Calibration

- Select sensors: DHT22 for temperature and humidity, BMP280 for pressure, and a rain gauge sensor.
- Calibrate sensors by comparing their readings against a trusted reference to correct systematic errors.

2. Hardware Setup

- Connect sensors to microcontroller (e.g., ESP32) using appropriate interfaces (digital pins, I2C).
- Ensure power supply stability and protect components for outdoor deployment.

3. Firmware Development

- Write embedded code to read sensors at fixed intervals (e.g., every 5 minutes).
- Apply filtering algorithms such as moving average to reduce sensor noise.

- Format data into JSON packets for transmission.

4. Wireless Communication

- Establish Wi-Fi connection with access point.
- Use MQTT protocol for lightweight, efficient data transmission to cloud server.
- Implement reconnection logic to handle network interruptions.

5. Cloud Integration

- Configure MQTT broker on cloud platform (ThingSpeak, AWS IoT, etc.).
- Set up data storage and dashboards for visualization.
- Develop API endpoints or use existing cloud APIs for data retrieval.

6. User Interface

- Create web/mobile dashboard for real-time monitoring and historical data visualization.
- Implement alert system based on threshold values.

5. CODINGS

- #include <WiFi.h>#include <PubSubClient.h>#include <DHT.h> #define DHTPIN 4
- #define DHTTYPE DHT22 const char* ssid = "your_wifi_ssid"; const char* password = "your_wifi_password";
- const char* mqttServer = "broker.hivemq.com"; constintmqttPort = 1883; const char* mqttTopic = "weatherstation/data";
- WiFiClientespClient; PubSubClientclient(espClient); DHT dht(DHTPIN, DHTTYPE);
- void setup() { Serial.begin(115200); dht.begin(); WiFi.begin(ssid, password); while (WiFi.status() != WL_CONNECTED) {delay(500);Serial.print("."); } Serial.println("WiFi connected");
- client.setServer(mqttServer, mqttPort);}void reconnect() {while (!client.connected()) {if (client.connect("ESP32Client")) { } else {delay(2000); }}
- void loop() {if (!client.connected()) {reconnect(); }client.loop(); float temperature = dht.readTemperature();float humidity = dht.readHumidity();
- if (isnan(temperature) || isnan(humidity)) {Serial.println("Failed to read from DHT sensor!");return; }
- String payload = "{\"temperature\":";payload += temperature;payload += ", \"humidity\":";payload += humidity; payload += "}";
- client.publish(mqttTopic, payload.c_str());Serial.println("Data published: " + payload);delay(300000); // Wait 5 minutes before next reading

6. RESULTS AND DISCUSSION

The implemented system was tested over a period of one month. The collected data showed consistency with local meteorological reports. The wireless communication proved stable with minimal data loss. Cloud logging allowed seamless access to weather data from any location, enhancing usability. Limitations include dependency on internet connectivity and sensor calibration needs. Future improvements may incorporate additional sensors and advanced analytics such as predictive weather modeling.

7. CONCLUSION

This paper demonstrates the design of a wireless weather station integrated with cloud-based data logging, providing a scalable and accessible solution for real-time environmental monitoring. The approach leverages IoT and cloud technologies to overcome limitations of traditional weather stations and offers potential applications in diverse fields requiring accurate weather information.

REFERENCES :

1. J. Smith, "Design of a Low-Cost Wireless Weather Station," International Journal of Environmental Monitoring, vol. 12, no. 3, pp. 150–158, 2021.
2. M. Patel and S. Kumar, "IoT-Based Weather Monitoring System Using MQTT and Cloud Platform," IEEE Sensors Journal, vol. 20, no. 10, pp. 5600–5607, 2020.
3. Johnson, "Cloud Computing for Environmental Data Logging," Journal of Cloud Applications, vol. 5, no. 2, pp. 90–95, 2019.
4. H. Li, X. Zhang, and Y. Chen, "Development of an IoT-Based Weather Station for Precision Agriculture," Sensors, vol. 18, no. 3, p. 745, 2018.
5. R. Kaur and P. Singh, "Wireless Sensor Networks for Environmental Monitoring: A Survey," International Journal of Computer Applications, vol. 174, no. 9, pp. 21–26, 2017.

6. S. Lee and J. Kim, "Real-Time Weather Data Acquisition System Using Wireless Sensor Network," International Journal of Distributed Sensor Networks, vol. 14, no. 5, 2018.
7. Gupta, "Low Power Wireless Weather Station Using ESP8266 and Cloud Storage," Procedia Computer Science, vol. 132, pp. 101–108, 2018.
8. D. Nguyen and M. Lee, "Implementation of an IoT-Enabled Weather Station Using MQTT Protocol," IEEE Access, vol. 7, pp. 151513–151523, 2019.
9. Adafruit Industries, "DHT22 Temperature-Humidity Sensor Guide," [Online]. Available: https://learn.adafruit.com/dht
10. Bosch Sensortec, "BMP280 Barometric Pressure Sensor Datasheet," 2017.
11. D. Perera, C. Liu, S. Jayawardena, and M. Chen, "A Survey on Internet of Things from Industrial Market Perspective," IEEE Access, vol. 2, pp. 1660–1679, 2014.
12. M. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A Survey," Computer Networks, vol. 54, no. 15, pp. 2787–2805, 2010.
13. OASIS, "MQTT Version 3.1.1," OASIS Standard, 2014.
14. Alamri et al., "A Review on the Role of IoT in Environmental Monitoring," Journal of Environmental Management, vol. 239, pp. 17–28, 2019.
15. ThingSpeakIoT Platform, "ThingSpeak API Documentation," [Online]. Available: https://thingspeak.com/docs
16. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context Aware Computing for The Internet of Things: A Survey," IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 414–454, 2014.
17. S. Misra, S. Biswas, and P. Das, "Real-Time Environmental Monitoring Using Wireless Sensor Networks," International Journal of Computer Applications, vol. 178, no. 23, pp. 1–6, 2019.
18. M. K. Hasan and M. S. Hossain, "Smart Environmental Monitoring System Using IoT," Journal of Sensor and Actuator Networks, vol. 8, no. 3, 2019.
19. Dutta, M. Chatterjee, and S. Chatterjee, "IoT Based Weather Monitoring System with Real-Time Data Logging," International Journal of Advanced Research in Computer Science, vol. 10, no. 4, pp. 1–5, 2019.
20. MQTT Official Website, "MQTT: The Standard for IoT Messaging," [Online]. Available: http://mqtt.org