Weather Monitoring System using ESP8266 NodeMCU

JEFFIN JAISON-33
S3 ER, Saintgits College of Engineering
jeffinj.er2428@saintgits.org

ASWIN SUDEEV-20
S3 ER, Saintgits College of Engineering
aswins.er2428@saintgits.org

ASHWIN GIREESH-19
S3 ER, Saintgits College of Engineering
ashwing.er2428@saintgits.org

Project Report on IoT based Weather Monitoring System

Submitted on: 17th October 2025

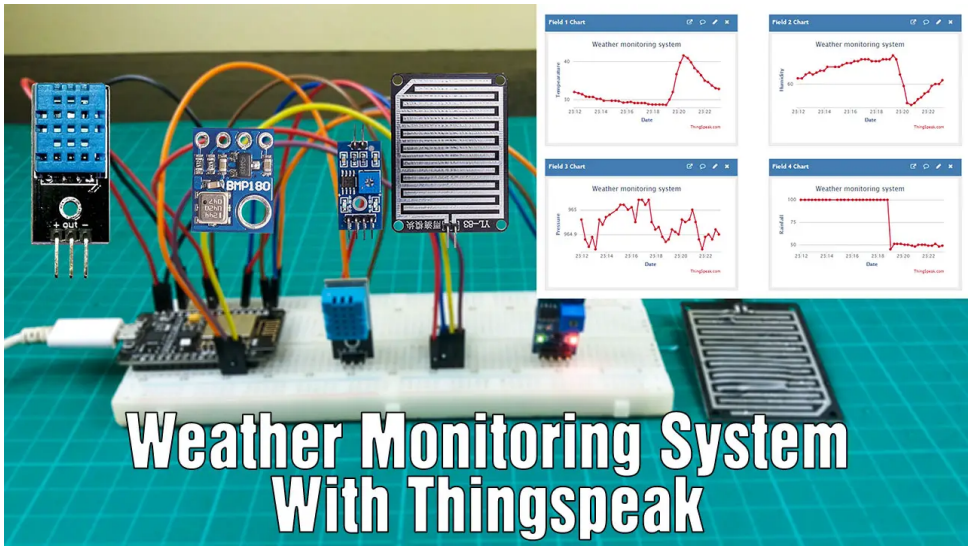

Figure 1: System Architecture of IoT-based Weather Monitoring System

Abstract

This project presents an IoT-based weather monitoring system using ESP8266 NodeMCU integrated with sensors such as DHT11 (temperature and humidity), rain sensor, and LDR (light-dependent resistor). The system continuously collects environmental data and uploads it to a cloud platform for real-time monitoring. The objective is to develop a low-cost, scalable, and easily deployable solution for remote weather observation and data analysis. Data can also be visualized on mobile or web dashboards using IoT services such as ThingSpeak or Blynk.

1 Introduction

Monitoring weather parameters such as temperature, humidity, rainfall, and light intensity is essential for agriculture, environmental analysis, and smart city applications. Traditional weather stations are often expensive and limited to specific regions. With advances in IoT technology, low-cost microcontrollers and sensors can now provide remote monitoring capabilities in real-time. This project aims to design and implement an IoT-based weather monitoring system using ESP8266 NodeMCU, which gathers sensor data and uploads it to the cloud for visualization and analysis.

2 Baseline or Initial Experiments

Initially, each sensor—DHT11, rain sensor and LDR—was individually tested with the NodeMCU to ensure accurate readings. The DHT11 was used to verify humidity and temperature accuracy, the rain sensor was tested for sensitivity in wet conditions, and the LDR for light intensity variation. Data were manually observed via the serial monitor before cloud integration was enabled to confirm sensor calibration and correct pin interfacing.

3 Objectives of the Project

- To design and implement an IoT-based weather monitoring system using ESP8266 NodeMCU.

- Measured display temperature, humidity, light intensity, and rainfall in real time.
- Transfer of collected data to an IoT platform such as ThingSpeak for cloud visualization.
- Provide a scalable and low-cost solution for the collection and analysis of weather data.

4 Working and Block Diagram

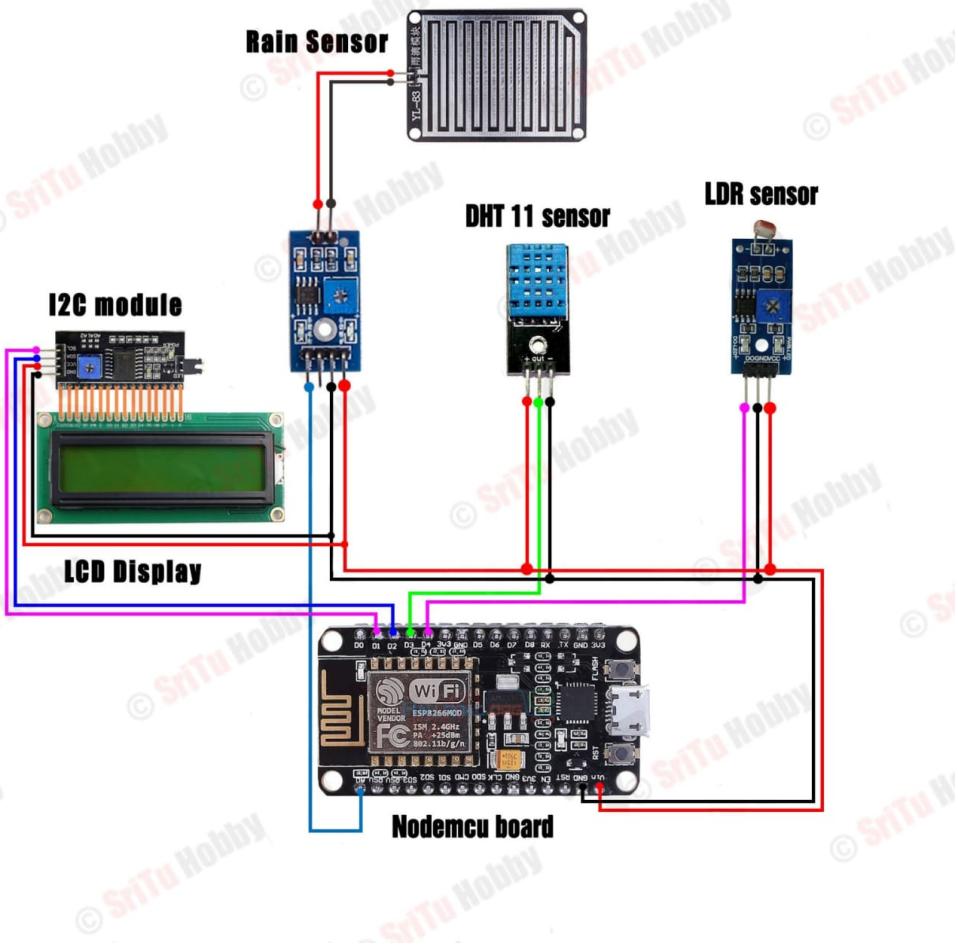

Figure 2: Block Diagram of IoT-based Weather Monitoring System

Working Principle:
The ESP8266 NodeMCU acts as the central controller, interfacing with all sensors:

- The **DHT11** measures temperature and humidity and sends digital data to the NodeMCU.
- The **rain sensor** detects the presence of water through its analog signal, determining the intensity of the rain.
- The **LDR** measures light intensity based on resistance variation with brightness.

The NodeMCU processes these data and transmits them to a cloud platform (e.g., ThingSpeak or Blynk) through Wi-Fi. Users can visualize and analyze real-time environmental conditions on mobile or web dashboards. The system ensures continuous monitoring with minimal human intervention.

5 Final Goals & Evaluation

- Develop an IoT-based weather station using NodeMCU, DHT11, rain sensor, and LDR.
- Accurately record and upload environmental parameters to the cloud.
- Evaluate system performance based on:
  - **Accuracy:** Compare sensor data with standard weather sources.
  - **Responsiveness:** Measure data update intervals and transmission delay.
  - **Stability:** Check continuous data logging over extended periods.
  - **Power Efficiency:** Analyze system performance under low power usage.

6 Applications and Advantages

The IoT-based weather monitoring system offers multiple practical applications and benefits:

- **Agriculture:** Helps farmers make informed decisions on irrigation, planting, and crop protection.
- **Environmental Monitoring:** Provides real-time data for air quality, rainfall, and humidity studies.
- **Smart Cities:** Supports urban planning and disaster management with accurate weather data.
- **Cost-effective:** Low-cost microcontrollers and sensors reduce installation and maintenance expenses.
- **Remote Monitoring:** Cloud integration allows data access from anywhere via web or mobile dashboards.
- **Scalability:** Additional sensors can be integrated to monitor more environmental parameters.
- **Flood Prevention:** Early detection of heavy rainfall can trigger alerts to prevent flood damage.
- **Urban Heat Monitoring:** Continuous temperature and humidity data help in monitoring urban heat islands.
- **Disaster Management:** Real-time data can assist authorities in preparing for storms, heavy rain, or extreme weather events.
- **Research and Education:** Provides live datasets for environmental science research and classroom projects.
- **Energy Management:** Data on sunlight (via LDR) can help optimize solar panel usage or smart lighting systems.
- **Climate Analysis:** Continuous long-term data collection supports climate trend analysis and modeling.

- **Community Awareness:** Local communities can receive weather updates and warnings, improving safety and preparedness.
- **Integration with Smart Devices:** Data can trigger smart irrigation systems, fans, or dehumidifiers automatically.
- **Historical Data Logging:** Enables analysis of past weather patterns for planning agricultural or urban projects.

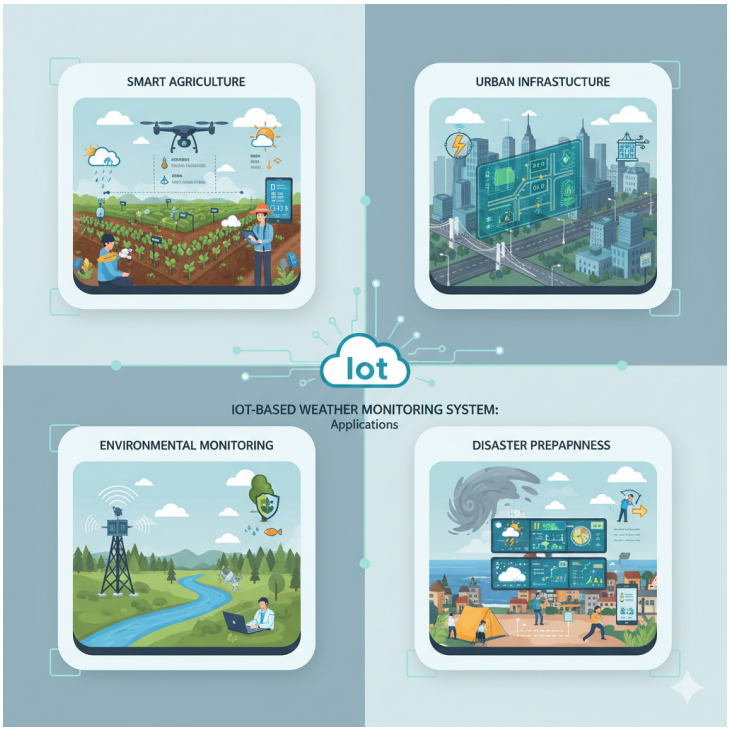

Figure 3: Visual Representation of Project Applications

7 Data & Technical Requirements

- **Microcontroller:** ESP8266 NodeMCU (Wi-Fi enabled, GPIO interface)
- **Sensors:** DHT11 (Temperature & Humidity), Rain Sensor, LDR
- **Power Supply:** 5V USB / regulated adapter
- **IoT Platform:** ThingSpeak or Blynk for real-time data visualization
- **Software Tools:** Arduino IDE for coding, cloud dashboard for monitoring

References

[Bengio & LeCun(2007)Bengio and LeCun] Yoshua Bengio and Yann LeCun. Scaling learning algorithms towards AI. In *Large Scale Kernel Machines*. MIT Press, 2007.

[Hinton et al.(2006)Hinton, Osindero, and Teh] Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18: 1527–1554, 2006.