Embedded Systems Special Assignment: Weather Monitoring System for CANSAT using RPi

Vidhi Ruparelia*, Sneh Shah†,
*† Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad, India
Emails: *22bec143@nirmauni.ac.in, †22bec121@nirmauni.ac.in

***Abstract*—The Weather Monitoring System for CANSAT is a compact and efficient embedded system designed to collect and transmit atmospheric data during a satellite's descent. This system integrates various sensors to measure temperature, humidity, pressure, and altitude, providing real-time environmental insights. The collected data is processed using a microcontroller and transmitted wirelessly to a ground station for analysis. The system is housed in a CANSAT— a can-sized satellite— making it a cost-effective solution for space and atmospheric research. The project focuses on optimizing power consumption, ensuring sensor accuracy, and improving data transmission reliability. This work contributes to the development of miniaturized weather monitoring solutions for educational and research applications in aerospace and environmental sciences.**

***Index Terms*—MQTT, Weather Monitoring, IoT, Raspberry Pi, Sensor Networks, Node-RED, InfluxDB, Grafana, Real-time Data Processing, Environmental Sensing, Data Visualization, Time-Series Database, Wireless Communication, Remote Monitoring, Atmospheric Sensors, Edge Computing, Custom GUI, Cloud Integration, CSV Data Logging, Data Transmission.**

I. Literature Review

Weather monitoring systems are essential for applications such as environmental studies, aerospace missions, and IoT automation. These systems rely on embedded sensors, real-time data acquisition, and wireless communication to monitor meteorological parameters. In CANSAT missions, where size, power efficiency, and lightweight design are critical, low-power sensors like BMP280 and IMU enable precise atmospheric data collection.

Several studies highlight BMP280 for atmospheric pressure and temperature sensing. Hassan Ali et al. [1] developed an IoT-based system integrating BMP280 with ESP8266 for real-time data transmission, demonstrating feasibility for CANSAT payloads. Salai Thillai Thilagam et al. [2] explored BMP280 in LabVIEW-based weather monitoring, emphasizing real-time visualization and alerts. Stoyanov et al. [3] demonstrated IMU sensors' effectiveness in tracking environmental changes.

IoT technology has revolutionized weather monitoring through wireless transmission and cloud analytics. Stoyanov et al. [3] developed an IoT-based weather station using ESP32 and InfluxDB with Grafana, highlighting redundancy and real-time accessibility. A similar low-cost station by R. Math et al. [4] integrated Node-RED and MQTT, demonstrating efficient real-time analytics and cloud-based dashboards.

Embedded systems enable real-time processing, sensor integration, and low-power operation. Hassan Ali et al. [1] explored Raspberry Pi and MQTT for remote sensor control and data transmission. Thillai Thilagam et al. [2] demonstrated LabVIEW dashboards for real-time user interaction and alerts. Stoyanov et al. [3] highlighted InfluxDB and Grafana's role in GUI-based weather data control.

Our Embedded Systems Special Assignment: Weather Monitoring System for CANSAT integrates Raspberry Pi as the central processor, BMP280 for pressure/temperature sensing, and IMU for motion tracking. We use Node-RED for processing, InfluxDB for time-series storage, and Grafana for visualization. A custom GUI enables sensor monitoring, Google Drive uploads, MQTT connectivity, and CSV logging. The system ensures real-time CANSAT telemetry and data analysis, with future improvements focused on energy efficiency and AI-driven weather prediction.

II. Limitations or drawbacks of currently available technology

Limited Sensor Reliability and Environmental Impact

Sensors such as BMP280 and IMU are highly sensitive to temperature fluctuations, pressure variations, and environmental exposure, which can affect their performance over time. In CANSAT applications, where physical access for maintenance is impossible, ensuring long-term data reliability is a major challenge. Power Consumption Constraints

Battery-powered weather monitoring systems face severe energy limitations, especially in remote or space-constrained applications like CANSAT. Continuous data acquisition, processing, and transmission rapidly depletes power, requiring efficient power management strategies, such as solar integration or ultra-low-power microcontrollers, to extend operational life. Connectivity and Data Transmission Delays

MQTT and Wi-Fi-based communication are prone to latency, signal loss, or bandwidth limitations, particularly in low-network coverage areas or satellite-based applications. These delays can affect real-time telemetry and remote monitoring, making it essential to explore low-latency communication protocols and edge processing for faster decision-making.

III. Proposed methodology

The Weather Monitoring System for CANSAT is implemented using a Raspberry Pi (RPi) as the central processing unit, interfacing with sensors for real-time data acquisition. The BMP280 sensor is used for atmospheric pressure and temperature measurement, while an IMU (Inertial Measurement Unit) provides motion and orientation data. These sensors communicate with the RPi via I2C/SPI protocols, ensuring efficient data collection.

For real-time data processing and automation, Node-RED is deployed on the RPi, acting as the central hub for sensor data integration, automation, and MQTT-based network communication. The collected data is stored in InfluxDB, a time-series database optimized for handling continuous sensor data streams. The stored data is visualized in Grafana, which provides interactive dashboards for monitoring weather parameters.

A custom GUI has been developed for system control, featuring:

- Real-time sensor data display in the output panel.
- Control options for sensor configuration and calibration.
- Data logging with CSV file generation.
- Dataset uploading to Google Drive for cloud-based storage.
- MQTT network integration for telemetry and remote monitoring

IV. Project Flowchart

Figure 1 illustrates the overall architecture of the weather monitoring system, highlighting the integration of sensors, data processing, visualization, and user interaction.

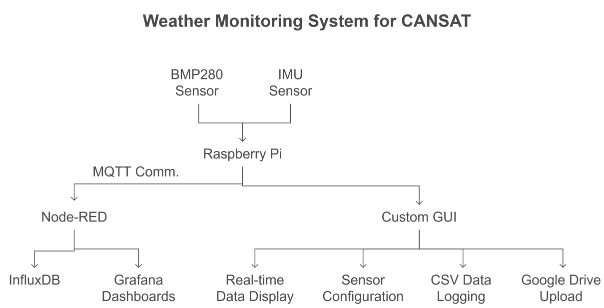

Fig. 1: Weather Monitoring System for CANSAT

a) Sensor Integration and Data Acquisition: The system includes BMP280 (temperature and pressure) and IMU sensors, which collect environmental data. These sensors are connected to a Raspberry Pi, which acts as the processing unit.

b) MQTT Communication for Data Transmission: The Raspberry Pi uses MQTT (Message Queuing Telemetry Transport) for wireless data communication. This ensures efficient data transfer between sensors and various processing and visualization modules.

c) Data Processing Using Node-RED and InfluxDB: Node-RED processes the sensor data for further analysis. The processed data is stored in InfluxDB, a time-series database designed for real-time sensor data storage.

d) Data Visualization with Grafana: The stored data is visualized using Grafana Dashboards, allowing users to analyze real-time trends.

e) Custom GUI for User Interaction: A custom-built GUI enables users to configure sensors, view real-time data, log data into CSV files, and upload data to Google Drive for remote access.

Figure 2 illustrates the sequential flow of operations in the weather monitoring system, detailing how data is collected, transmitted, processed, stored, visualized, and made available for user interaction.

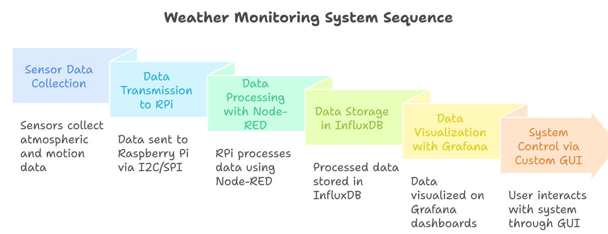

Fig. 2: Weather Monitoring System Sequence

f) Sensor Data Collection: Sensors collect atmospheric and motion data from the environment.

g) Data Transmission to Raspberry Pi: The collected data is transmitted to the Raspberry Pi via interfaces such as I2C or SPI.

h) Data Processing with Node-RED: The pre-processed sensor data is handled using Node-RED, ensuring smooth data flow and filtering.

i) Data Storage in InfluxDB: Processed data is stored in InfluxDB, a time-series database that efficiently manages continuous sensor readings.

j) Data Visualization Using Grafana: The stored data is graphically represented in Grafana dashboards, allowing real-time monitoring.

k) User Interaction via Custom GUI: Users can interact with the system through the custom GUI, which enables sensor configuration, real-time data display, and CSV data logging.

V. Implementation

The Weather Monitoring System for CANSAT was successfully implemented, integrating Raspberry Pi, BMP280, and IMU sensor for real-time environmental monitoring. The BMP280 effectively measured atmospheric pressure and temperature, while the IMU sensor provided motion and orientation data. These sensors communicated with the RPi via I2C protocols, ensuring reliable data acquisition.

For data processing and automation, Node-RED was deployed on the RPi, enabling real-time sensor integration and MQTT-based data transmission. The collected data was stored in InfluxDB, a time-series database optimized for sensor data streams. The platform we have used for visualization is Grafana dashboard, which easily helps visualize real-time weather parameters.

We have developed a custom GUI to enhance user interaction, displaying real-time sensor readings in an output panel as shown in the figure. The GUI also included functionalities such as sensor control, dataset uploading to Google Drive, CSV file generation, and MQTT connectivity. The system successfully transmitted sensor data over the network, enabling remote monitoring.

Figure 3 illustrates the project setup, showcasing the hardware components and connections.

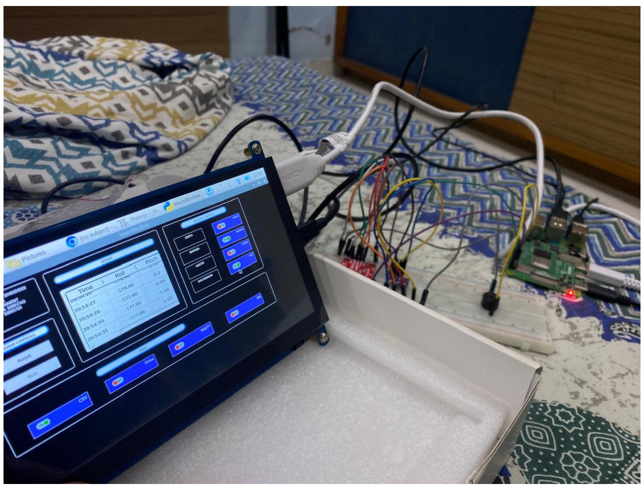

Fig. 3: Project Setup

Figure 4 displays the customized GUI, which enables users to interact with the system, monitor sensor data, and manage data logging.

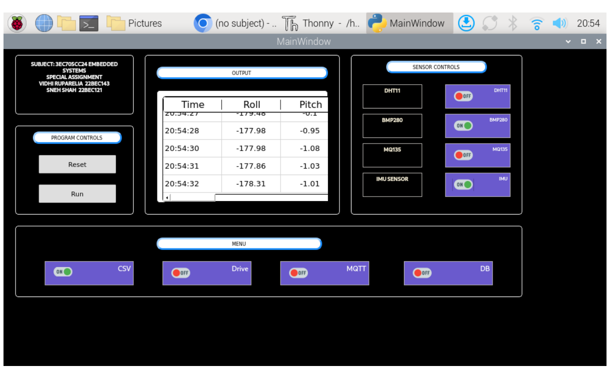

Fig. 4: Customized GUI

Figure 5 presents the Node-RED implementation, highlighting the data processing and workflow automation.

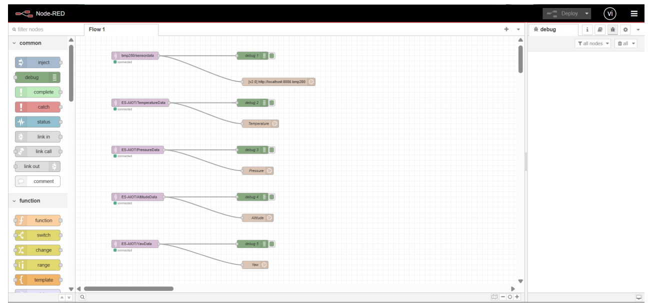

Fig. 5: NodeRed Implementation

Figure 6 shows the InfluxDB storage structure, which efficiently manages time-series sensor data.

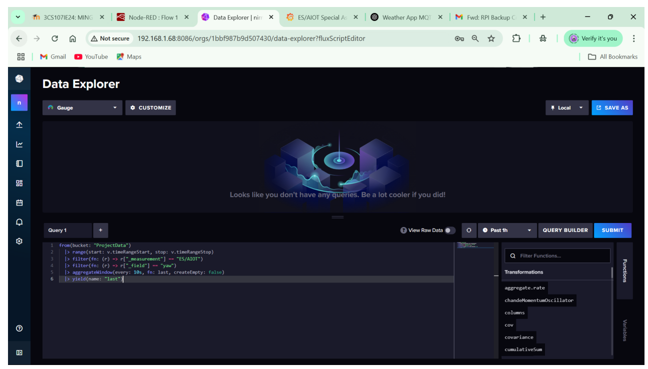

Fig. 6: InfluxDB

Figure 7 demonstrates the Grafana visualization, providing real-time monitoring with graphical dashboards.

Fig. 7: Grafana Visualization

Figure 8 illustrates the sensor readings and graphical visualization on a mobile device, demonstrating remote accessibility and real-time monitoring.

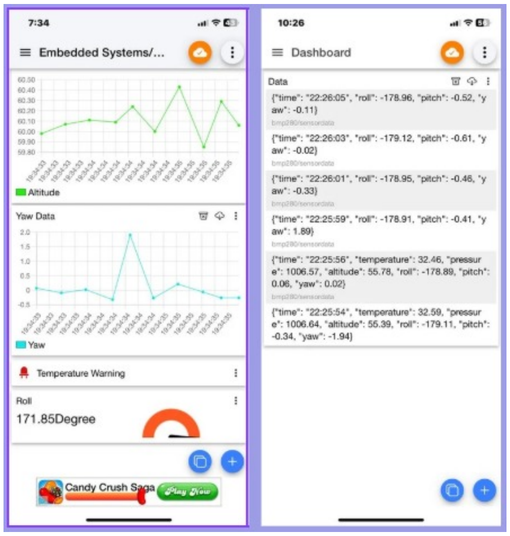

Fig. 8: Readings and Graphs on Phone

VI. Analysis of the Obtained Results

The Weather Monitoring System for CANSAT was successfully implemented using the MING pipeline (MQTT, InfluxDB, Node-RED, Grafana) in an IoT-based setup with Raspberry Pi as the central controller. All sensor readings from BMP280 and IMU were accurately displayed on the custom GUI, ensuring real-time monitoring and control.

Sensor data was efficiently stored in InfluxDB, enabling smooth data retrieval and analysis. The visualization in Grafana provided clear, real-time graphical representations of all sensor values, allowing trend identification and system evaluation.

Additional functionalities, including CSV file generation and automatic dataset uploads to Google Drive as an Excel sheet, were successfully integrated, enhancing data accessibility and storage options. The system performed reliably and efficiently, making it well-suited for CANSAT applications with potential future improvements in predictive analytics and power optimization.

VII. Conclusion and Future Scope

The Weather Monitoring System for CANSAT was successfully developed using Raspberry Pi, BMP280, and IMU sensors, with Node-RED, InfluxDB, and Grafana for real-time data processing and visualization. The system accurately captured sensor readings, displayed them on a custom GUI, and stored data in InfluxDB, enabling smooth analysis with Grafana dashboards. Additional features like CSV file generation and Google Drive uploads enhanced data accessibility. Future improvements include AI-based predictive analytics for weather forecasting, power optimization using low-power microcontrollers or solar energy, edge computing for faster on-device processing, expanded sensor integration for better environmental monitoring, and stronger data security for reliable cloud-based storage. These enhancements will make the system more efficient, intelligent, and adaptable for future aerospace and environmental applications.

References

[1] H. Ali, A. A. Farooque, F. Abbas, R. Yaqub, A. Abdalla, and P. Soora, "An iot based weather monitoring system for smart agriculture," in *2024 IEEE Conference on Technologies for Sustainability (SusTech)*, pp. 378–382, IEEE, 2024.

[2] T. S. Babu, B. S. Reddy, *et al.*, "Weather monitoring system application using labview," in *2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), 2018 2nd International Conference on*, pp. 52–55, IEEE, 2018.

[3] S. Stoyanov, Z. Kuzmanov, and T. Stoyanova, "Weather monitoring system using iot-based diy automatic weather station," in *2024 9th International Conference on Energy Efficiency and Agricultural Engineering (EE&AE)*, pp. 1–6, IEEE, 2024.

[4] R. K. M. Math and N. V. Dharwadkar, "Iot based low-cost weather station and monitoring system for precision agriculture in india," in *2018 2nd international conference on I-SMAC (IoT in social, mobile, analytics and cloud)(I-SMAC) I-SMAC (IoT in social, mobile, analytics and cloud)(I-SMAC), 2018 2nd international conference on*, pp. 81–86, IEEE, 2018.