

Flow Diagram and System Overview for Weather Station Project

Ravishan BBN

January 23, 2025

1 Introduction

This document provides a comprehensive overview of the Weather Station project, including a detailed system flow diagram and descriptions of how the various components interact. The project aims to monitor environmental parameters using sensors connected to a Raspberry Pi and to visualize the data in real-time through dashboards and cloud integration.

2 System Components

The Weather Station system is composed of the following key components:

- **Raspberry Pi:** Serves as the central processing hub, collecting and transmitting sensor data.
- **Sensors:**
 - **BMP280:** Measures temperature and pressure.
 - **MQ7:** Detects CO levels.
 - **MQ135:** Detects hazardous gas levels.
- **Node-RED:** A low-code programming tool for wiring together hardware devices, APIs, and online services.
- **ThingSpeak:** A cloud-based platform for visualizing data through charts and triggering actions based on thresholds.
- **MQTT Broker:** Facilitates communication between Raspberry Pi and Node-RED, ensuring reliable data transmission.
- **Push Notification Service (e.g., Pushbullet):** Sends real-time alerts to users when certain thresholds are breached.

3 Description of the Flow Diagram

The following steps outline the data flow in the Weather Station system:

3.1 1. Data Collection

- Sensors (BMP280, MQ7, and MQ135) collect environmental data such as temperature, pressure, CO levels, and hazardous gas levels.
- The Raspberry Pi, using Python scripts, reads and processes data from the connected sensors.

3.2 2. Data Transmission

- The processed sensor data is transmitted to an MQTT broker hosted on the Raspberry Pi.
- Node-RED subscribes to the MQTT topics to receive the data in real-time.

3.3 3. Data Processing

- Node-RED processes the incoming data, applying filters and converting it into a format suitable for visualization and cloud integration.
- Data is forwarded to ThingSpeak for cloud-based visualization and storage.

3.4 4. Data Visualization and Alerts

- ThingSpeak visualizes the data through real-time dashboards and charts.
- Threshold-based triggers are configured in ThingSpeak React to send alerts (e.g., push notifications) to mobile devices when predefined conditions are met.

3.5 5. Mobile Integration

- Pushbullet or similar services are used to deliver notifications to mobile devices.
- Alerts include critical updates such as "High CO Level Detected" or "Hazardous Gas Level Exceeded."

4 Creating the Flow Diagram

To design the flow diagram, you can use tools like Draw.io, Figma, or Microsoft Visio. Follow these steps:

1. **Define Components:** List all components and identify their roles in the system.
2. **Design Layout:** Arrange components logically to represent data flow clearly.
3. **Connect Components:** Use directional arrows to illustrate the flow of data between components.
4. **Label Components:** Add descriptive labels to ensure clarity.
5. **Export Diagram:** Save the diagram as a high-quality image (e.g., PNG) and include it in documentation.

5 Integration and Alerts

5.1 ThingSpeak React Configuration

- Use the React feature in ThingSpeak to monitor incoming data for predefined conditions (e.g., hazardous gas level > 100).
- Configure React to trigger a ThingHTTP request to send notifications via services like Pushbullet.

5.2 Pushbullet Notification Setup

1. Register a Pushbullet account and generate an API key.
2. Use ThingHTTP to send POST requests to the Pushbullet API when React conditions are met.
3. Example of a notification payload:

```
{
  "type": "note",
  "title": "High Threshold Alert!",
  "body": "The sensor value has exceeded the threshold."
}
```

6 Best Practices for Implementation

- Regularly calibrate sensors to ensure accurate readings.
- Use secure credentials for MQTT, ThingSpeak, and Pushbullet to protect data and notifications.
- Monitor system logs on Raspberry Pi and Node-RED to identify and resolve issues promptly.
- Test the entire workflow with sample data before deploying the system in a live environment.

7 Conclusion

The flow diagram and system overview illustrate how the Weather Station project integrates sensors, Raspberry Pi, Node-RED, ThingSpeak, and push notification services into a cohesive monitoring solution. Understanding these interactions is crucial for maintaining and expanding the system.