NIRMA UNIVERSITY
INSTITUTE OF TECHNOLOGY
NAAC ACCREDITED 'A+' GRADE

Embedded Systems - Special Assignment
CANSAT Based Weather Monitoring Station

Work By :- 1) Sneh Shah(22BEC121)
2) Vidhi Ruparelia(22BEC143)

21 March , 2025

Overview

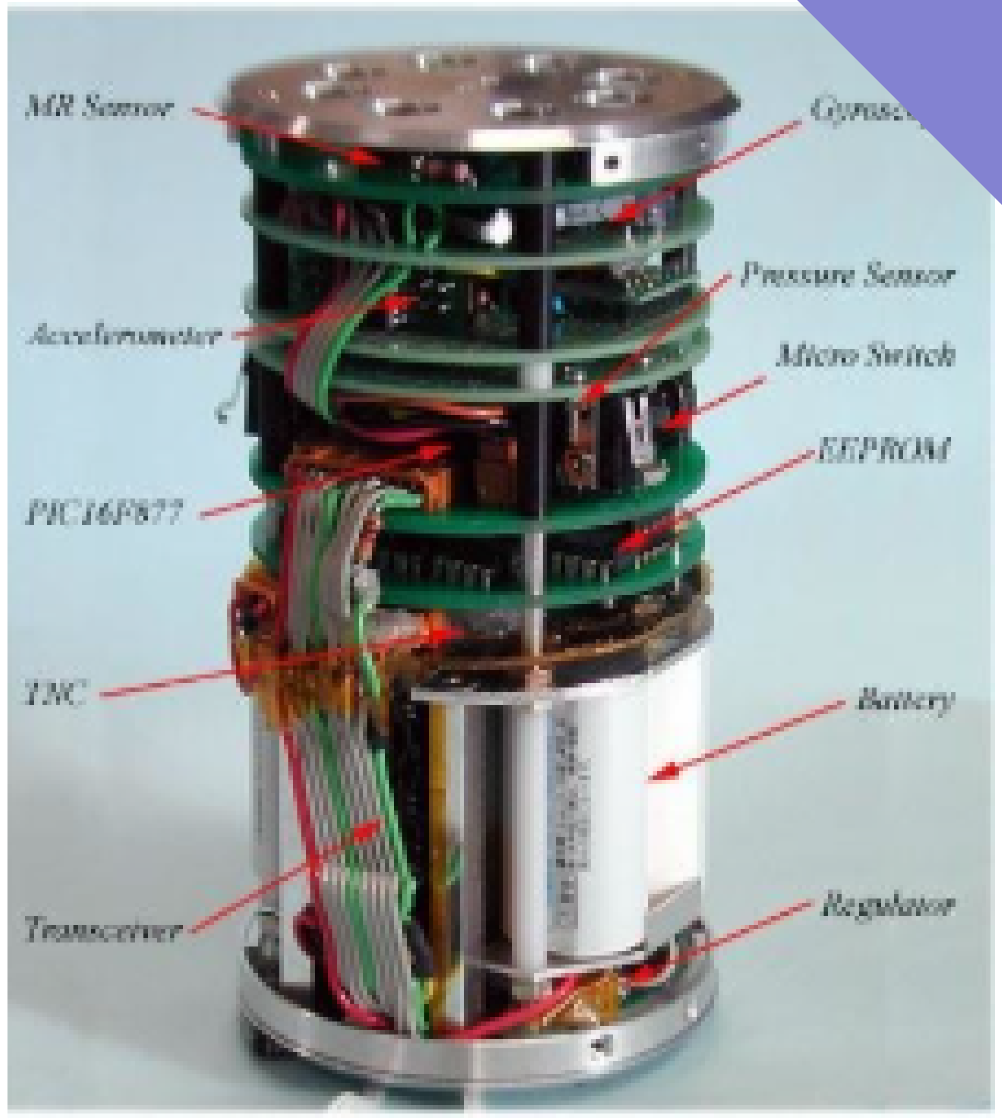
What is CANSAT?

1. What is a CANSAT?

- A miniature satellite in the shape of a soft drink can.
- Used for educational and research purposes in aerospace and embedded systems.
- Designed to simulate real satellite missions on a small scale.

2. Components of a CANSAT

- Microcontroller/Processor (Raspberry Pi, Arduino, ESP32) – Controls operations.
- Sensors (BMP280, IMU, GPS, temperature, pressure, humidity) – Collects environmental data.
- Power System (Battery, Solar Panel) – Provides energy for operation.
- Communication Module (LoRa, RF, Wi-Fi, MQTT) – Transmits data to a ground station.
- Parachute System – Ensures a safe descent after launch.



What is CANSAT?

3. How Does a CANSAT Work?

- Launch – Deployed from a drone, balloon, or rocket.
- Data Collection – Sensors measure atmospheric parameters.
- Telemetry Transmission – Real-time data is sent to the ground station.
- Descent & Recovery – Lands using a parachute while continuously transmitting data.
- Data Analysis – Information is processed for research and applications.

4. Importance of CANSAT in Real Life

- Aerospace & Satellite Training – Provides hands-on experience in space technology.
- Weather Monitoring – Collects real-time meteorological data.
- Environmental Research
- Disaster Management
- Agriculture – Assists in monitoring temperature, humidity, and soil conditions.
- Military & Surveillance – Can be used for aerial reconnaissance and intelligence gathering.

Literature Review

- **1. Introduction**
  - Essential for environmental studies, aerospace missions, and IoT automation.
  - Uses embedded sensors, real-time data acquisition, and wireless communication.
- 2. **Importance in CANSAT Missions**
  - Requires size, power efficiency, speed , and lightweight design.
  - Low-power sensors like BMP280 (pressure & temperature) and IMU (motion tracking) ensure precise data collection
- **Hassan Ali et al.: IoT-based system using BMP280 & ESP8266 for wireless transmission.**
- **Salai Thillai Thilagam et al.: LabVIEW-based weather monitoring with myDAQ for real-time visualization.**
- **Stoyanov et al.: Multi-sensor IoT system using IMU for environmental tracking.**
- **Stoyan Stoyanov et al.: DIY IoT weather station with ESP32, InfluxDB, and Grafana for cloud-based analytics.**
- **R. Math et al.: Low-cost weather station using Node-RED & MQTT for real-time data processing**.

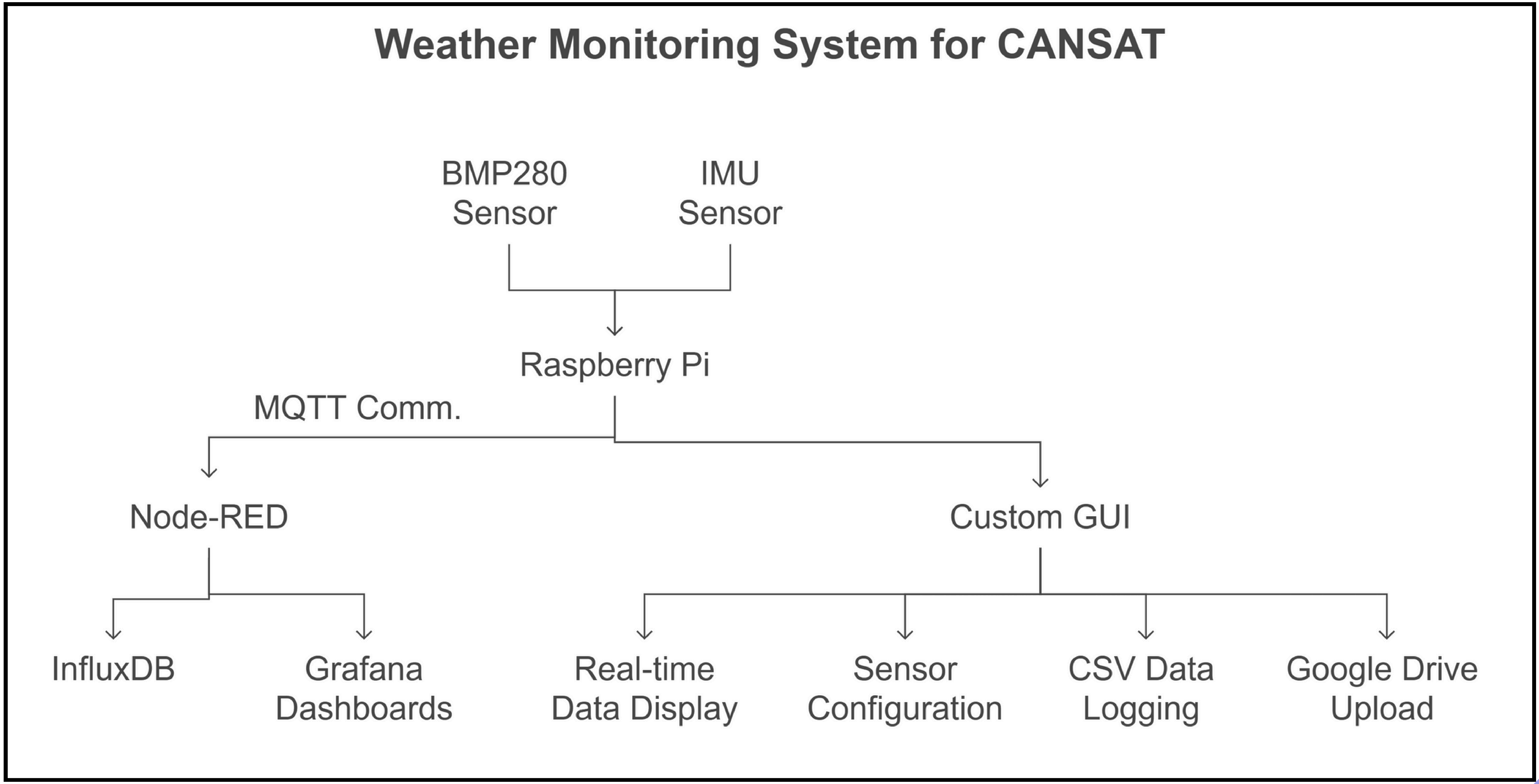
Limitations in the current technology

- **Limited Power Supply** – Small form factor restricts battery capacity, affecting mission duration.
- **Restricted Payload Capacity** – Space and weight constraints limit the number of sensors and advanced instruments.
- **Communication Range** – Low-power transmission systems (RF, LoRa, Wi-Fi) have limited range compared to full-scale satellites.
- **Environmental Vulnerability** – Harsh atmospheric conditions (temperature, wind, pressure) can affect sensor accuracy and system reliability.
- **Data Processing Constraints** – Limited onboard computing power restricts real-time analysis and AI-based applications.

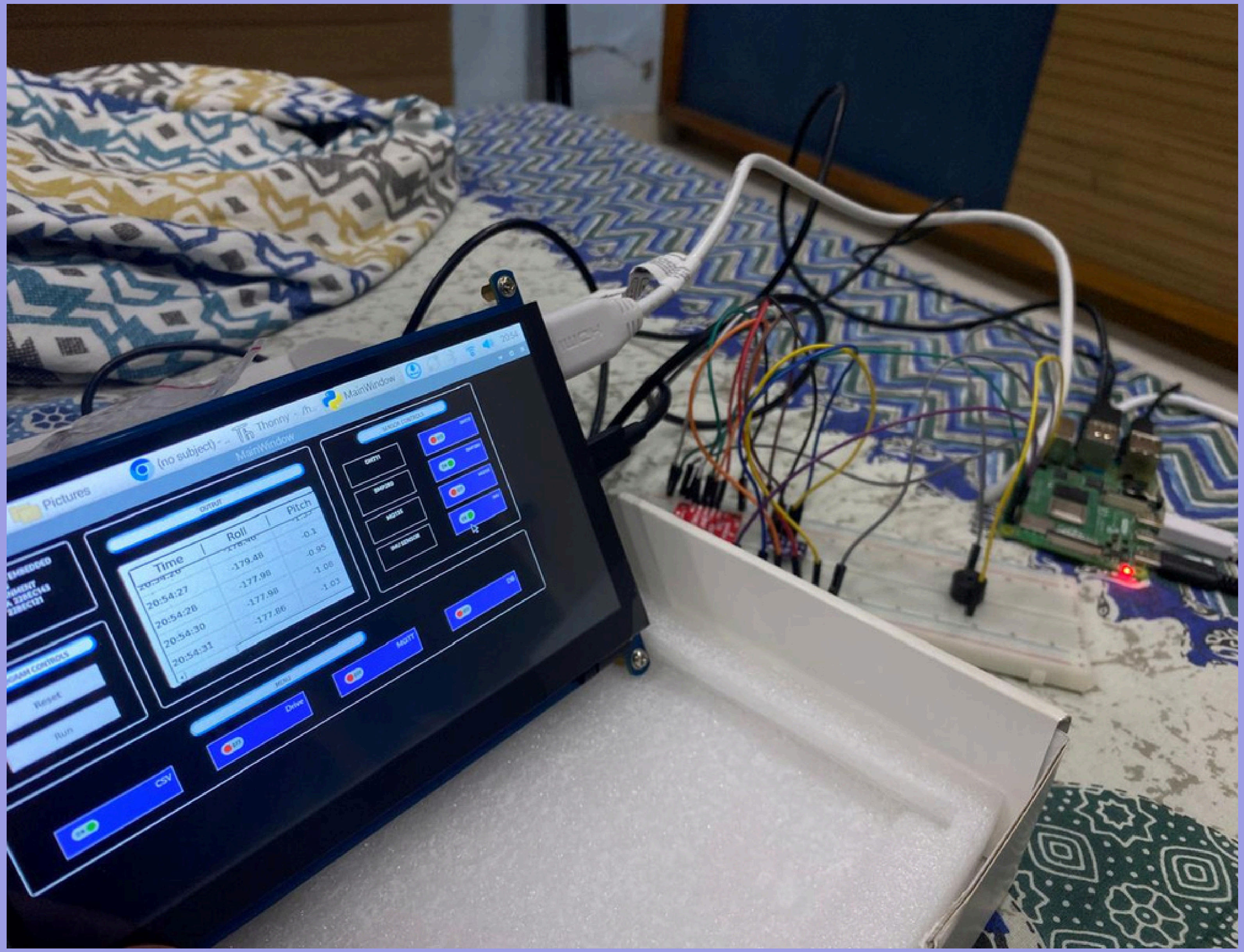
Proposed Methodology

- **Sensor Integration & Data Acquisition** – Interfacing BMP280, IMU, GPS, and RTC with Raspberry Pi or ESP32 for real-time atmospheric data collection.
- **Data Processing & Storage** – Utilizing Node-RED for data handling, InfluxDB for time-series storage, and MongoDB for metadata management.
- **Real-Time Visualization & Monitoring** – Implementing Grafana for live dashboard access on both desktop and mobile devices.
- **Automated Alerts & Notifications** – Configuring threshold-based alerts via SMS, Telegram, or Email for critical weather conditions.
- **Data Logging & Cloud Storage** – Periodic CSV file generation and automatic upload to Google Drive for remote access.
- **Wireless Communication & Telemetry** – Using MQTT and LoRa/Wi-Fi for real-time data transmission between CANSAT and the ground station.

Flowchart/Block Diagram



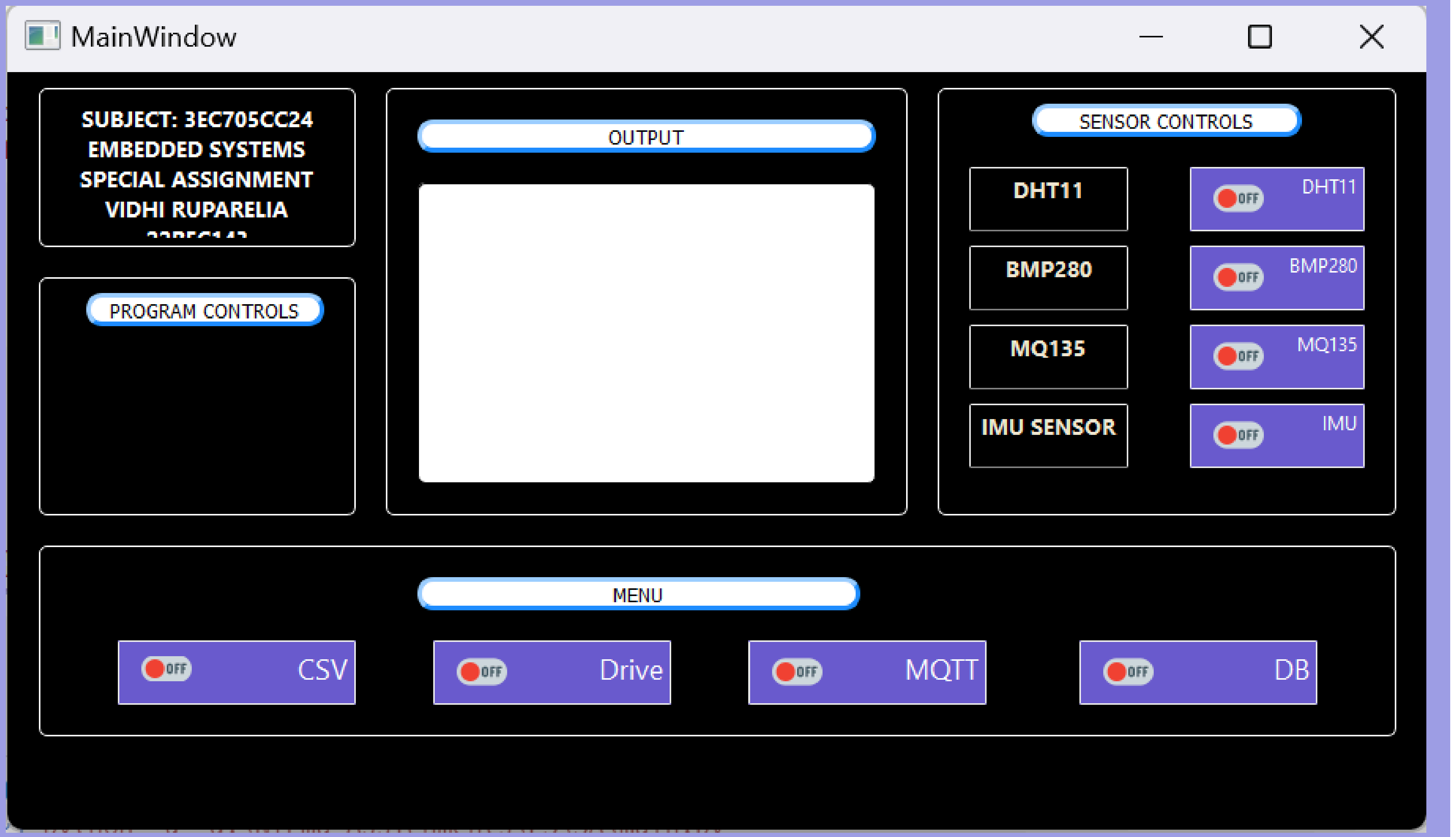
Implementation



1. **System Architecture Overview**

- **Microcontroller/Processor:** Raspberry Pi or ESP32 for sensor interfacing and data handling.
- **Sensors:** BMP280 (pressure & temperature), IMU (motion tracking), RTC (real-time clock), GPS (location tracking).
- **Communication:** LoRa, MQTT, or Wi-Fi for real-time data transmission.
- **Storage & Visualization:** MING stack (MongoDB, InfluxDB, Node-RED, Grafana) for data collection, processing, and cloud integration.
- **Mobile Dashboard:** Accessible via a smartphone for real-time monitoring and alerts.

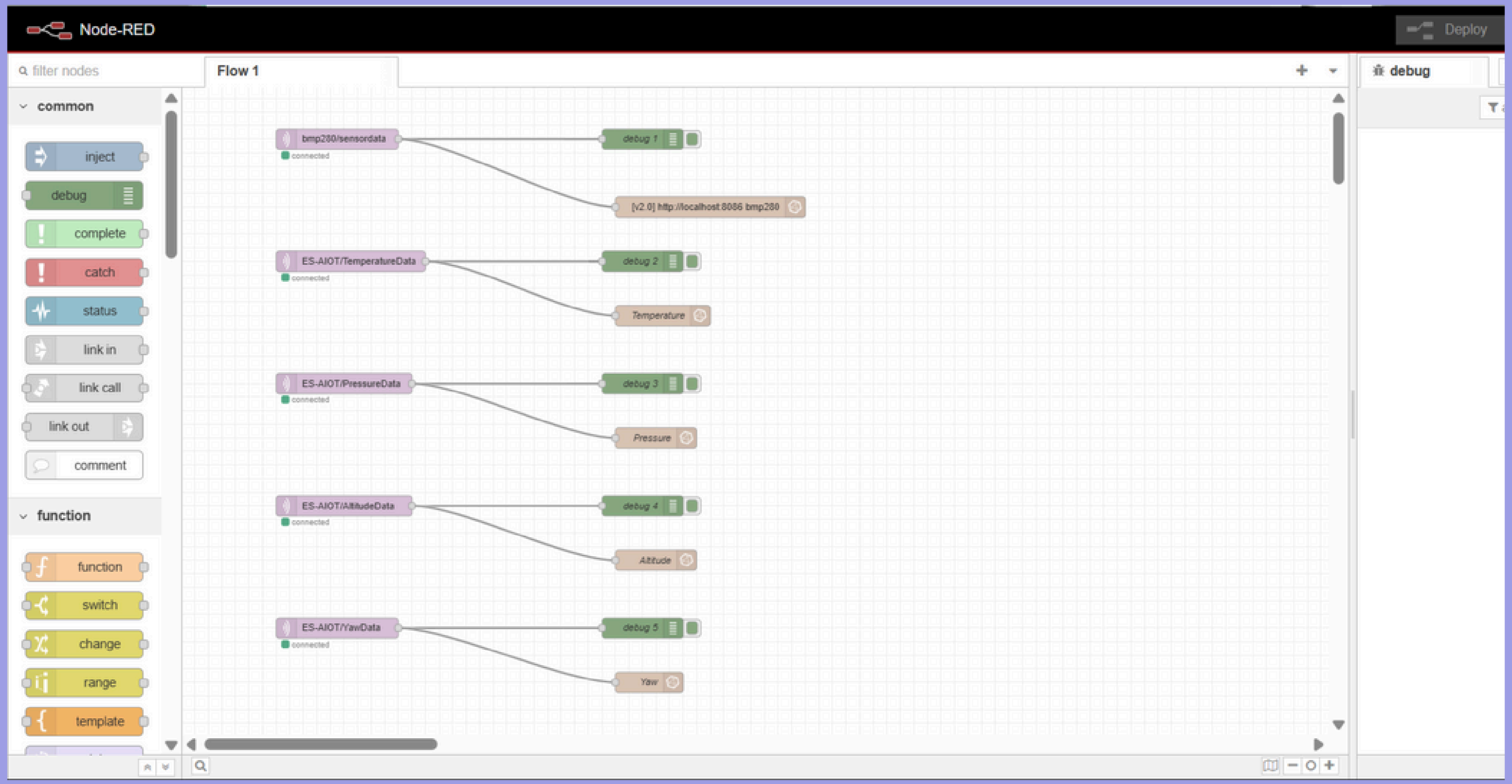
Implementation



2. Sensor Interfacing & Data Acquisition

- **BMP280** → Measures atmospheric pressure and temperature.
- **IMU (Accelerometer & Gyroscope)** → Tracks motion, orientation, and stability.
- **GPS Module(Future Plan)** → Captures real-time location (latitude, longitude, altitude).
- **RTC Module(Future Plan)** → Provides accurate timestamping for sensor data synchronization.
- **Embedded System Implementation:**
  - Raspberry Pi (or ESP32) reads sensor values at predefined intervals.
  - Filters and formats data for transmission and storage.

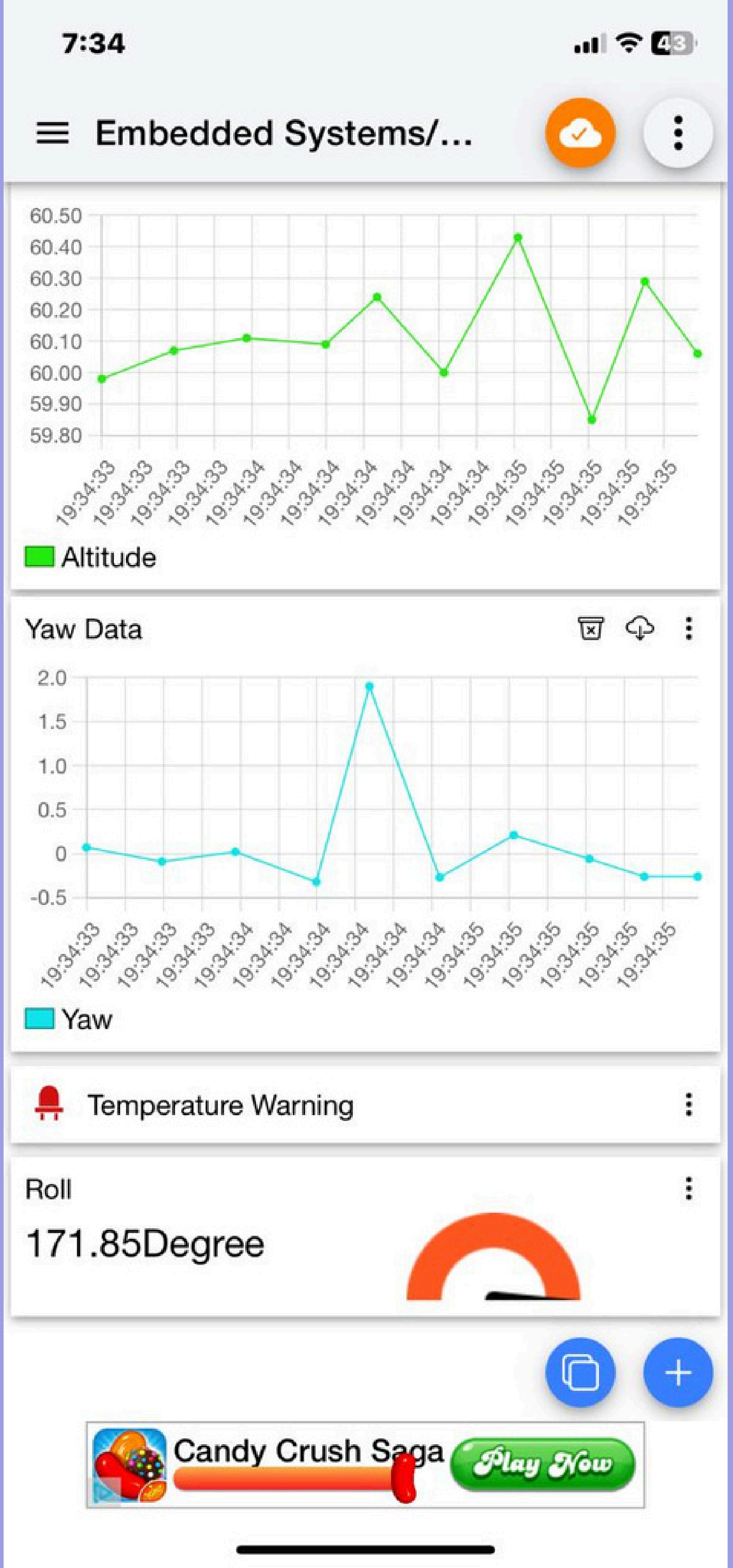
Implementation

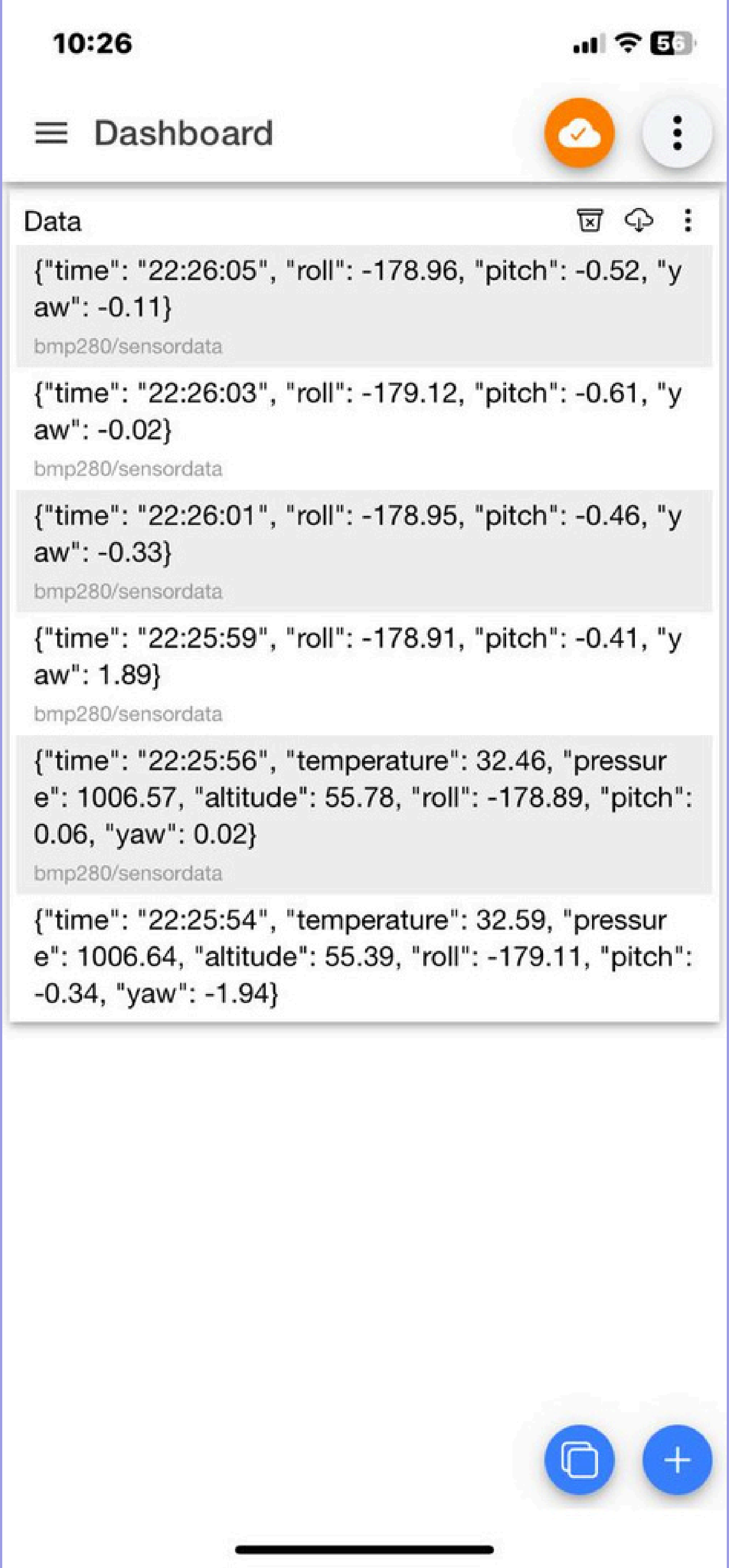


3. Data Processing & IoT Integration (MING Stack)

- **Node-RED:**
  - Manages sensor data flow.
  - Formats raw sensor values into readable metrics.
  - Triggers automated alerts based on threshold values.
- **InfluxDB:**
  - Efficiently stores time-series data from sensors.
  - Supports real-time querying for analysis.
- **MongoDB:**
  - Stores metadata, sensor configurations, and mission logs.
- **Grafana:**
  - Provides live visualization of telemetry data on desktop & mobile dashboards.
  - Displays real-time graphs, maps, and analytics.

Implementation





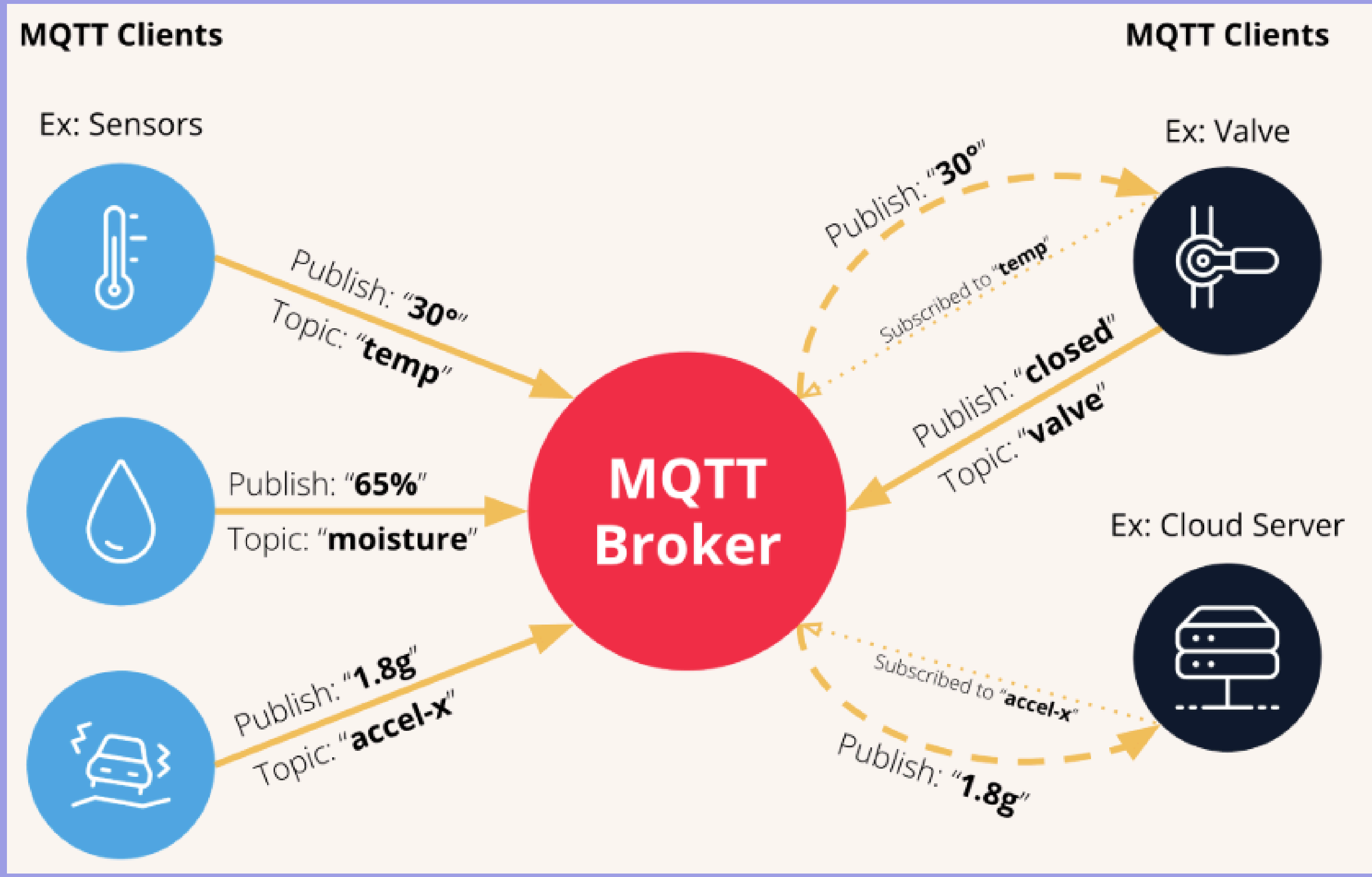
4. Mobile Dashboard & Alerts

- **Smartphone Access:**
  - Interactive UI with sensor trends, weather analytics, and CANSAT telemetry.
- **Automated Alerts via Telegram, SMS, or Email:**
  - Trigger Conditions:
    - Sudden temperature/pressure changes.
    - Unstable motion detected by IMU.
  - Alerts sent to user-defined contacts for immediate response.

5. Backend Processing & Data Logging

- **CSV File Generation:**
- **Automated Cloud Upload**:
  - CSV files are periodically uploaded to Google Drive via API for remote access.

Implementation



6. Communication & Telemetry

- **MQTT Protocol:**
  - Enables low-latency data exchange between CANSAT and ground station.
  - Facilitates real-time sensor monitoring and control.
- **LoRa / Wi-Fi Data Transmission:**
  - Ensures long-range connectivity for telemetry in remote environments.

7. System Deployment & Testing

- **Field Testing:**
  - Deploy in an open area (e.g., balloon, drone launch).
  - Validate real-time data transmission, storage, and dashboard performance.
- **Performance Evaluation:**
  - Optimize power efficiency, sensor accuracy, and alert system responsiveness.

Conclusion/Future Scope

- **Conclusion**
  - Successfully implemented a small-scale IoT-based CANSAT weather monitoring system.
  - Integrated real-time data acquisition, processing, and cloud storage using MING stack.
  - Enabled automated alerts, remote monitoring, and CSV-based logging.
  - Achieved low-power, cost-effective, and wireless telemetry for atmospheric studies.
- **Future Scope**
  - **Expanded Sensor Integration** – Adding gas, humidity, and radiation sensors.
  - **AI-Based Weather Prediction** – Implementing ML models for forecasting.
  - **Edge Computing** – Onboard processing using Jetson Nano or ESP32 AI modules.
  - **Power Optimization** – Using solar panels and low-power modes.
  - **Advanced Communication** – Exploring satellite/5G connectivity for telemetry.
  - **Multi-CANSAT Network** – Deploying multiple units for large-area monitoring.

References

- **MING Stack Integration in IoT Systems**
  - M. Pous. "MING: An Open-Source LAMP-like Stack for the Internet of Things." GitHub Repository. [Online]. Available: GitHub
- **AI-Based Weather Prediction Models**
  - J. Pathak, A. Subramanian, A. McGibbon, J. Zeng, S. E. Howland, N. Fox, M. J. Shelby, and M. P. Brenner, "FourCastNet: A Global Data-Driven High-Resolution Weather Model using Adaptive Fourier Neural Operators," arXiv preprint arXiv:2202.11214, 2022. [Online]. Available: Nature
- **Edge Computing with AI Modules**
  - A. Anandkumar, "Anima Anandkumar Highlights AI's Potential to Solve 'Hard Scientific Challenges'," Time Magazine, Feb. 2025. [Online]. Available: Time
- **Power Optimization in IoT Devices**
  - A. Mothukuri, "IoT Based Weather Monitoring System," SSRN Electronic Journal, 2021. [Online]. Available: SSRN
- **Advanced Communication Technologies for IoT**
  - S. Stoyanov, "DIY IoT-Based Automatic Weather Station," Instructables, 2023. [Online]. Available: Google DeepMind
- **Multi-CANSAT Networks for Environmental Monitoring**
  - M. Hasan, I. I. Rahman, M. H. E. Haider, and A. A. M. S. Sadman, "Design of CanSat for Environmental Monitoring and Object Detection," in Proceedings of the International Conference on Mechanical, Industrial and Energy Engineering, Khulna, Bangladesh, 2021. [Online]. Available: ResearchGate

Thank You