IoT Environmental Monitoring System – User Guide

1. Project Overview

This IoT system monitors environmental parameters using three types of sensors:

- **Dust Sensor SKU : 10500** – Measures air particle concentration.
- **Sound Sensor SPH0645** – Measures ambient sound levels.
- **Temperature Sensor BMP280** – Measures environmental temperature.

Data Flow

1. The sensors are connected to an **ESP32 Receiver**.
2. The receiver sends data via **LoRa** to an **ESP32 Sender**.
3. The sender transmits the data over **Wi-Fi** to a custom **API**.
4. The API:
   - Stores data in a **database**.
   - Sends data to **Metabase** for dashboard visualization.
   - Sends air quality data to the **Sensor Community** platform.

2. Required Components

Hardware

- 1x ESP32 Receiver (connected to sensors)
- 1x ESP32 Sender (with Wi-Fi capability)
- Dust Sensor (SKU : 10500)
- Sound Sensor (SPH0645)
- Temperature Sensor (BMP280)
- 1 x Lora Module
- Breadboard or PCB + wires
- USB cable and 5V power supply

Software

- ESP-IDF + C language
- ESP32 firmware code (receiver and sender)
- API (Node.js)
- PostgreSQL
- Metabase (self-hosted)
- Sensor Community account + registered sensor

3. System Setup

3.1. Hardware Configuration

ESP32 Receiver:

- Connect the dust sensor to UART or a GPIO pin.
- Connect the sound sensor to an analog input.
- Connect the temperature sensor to a digital GPIO pin.
- Connect the LoRa module to SPI pins.

ESP32 Sender:

- Connect the LoRa module
- Configure Wi-Fi access credentials in the firmware

3.2. Flashing the ESP32

1. Install Arduino IDE and the ESP32 board support package.
2. Load the appropriate code (receiver or sender).
3. Adjust parameters (pins, Wi-Fi, API endpoint) in the source code.
4. Connect the ESP32 via USB and upload the code.

3.3. API Setup

Follow the readme steps located in the « api » folder.

1. Clone or install the API server code.
2. Configure environment variables
3. Start the API server

3.4. Database and Metabase

Follow the readme steps located in the « Metabase » folder.

- Set up a PostgreSQL database.
- Connect Metabase to this database using the interface.
- Create dashboards using provided queries or build custom visualizations:
    - Air quality trends
    - Temperature graphs
    - Noise level monitoring

3.5. Sensor Community Integration

- Ensure your API sends data in the format required by Sensor Community.
- Use HTTPS POST requests with authentication token.
- Verify that your station appears on the platform’s map and dashboard.

4. System Usage

1. Power on both ESP32 boards.
2. Open the serial monitor (optional) to confirm sensor readings and data flow.
3. Ensure LoRa modules are properly communicating.
4. Access Metabase dashboards to visualize the incoming data.
5. Use Sensor Community to share and validate air quality data globally.

6. Troubleshooting

Issue	**Cause**	**Solution**
No data in Metabase	API not running or DB offline	Restart the API and check DB connection

Issue	Cause	Solution
ESP32 not connecting to Wi-Fi	Incorrect SSID/password	Check credentials in the firmware
LoRa connection failure	Bad antenna or wiring issue	Verify connections and reflash both devices
No data in Sensor Community	Wrong token or format	Check logs, validate API call formatting
ESP32 not detected on USB	Missing drivers	Install drivers depending on the board

6. Maintenance and Updates

- Restart ESP32 devices periodically to avoid memory issues.
- Regularly check sensor values for calibration drift.
- Archive old data from the database if needed.
- Update firmware or API code to apply improvements or fix bugs.

7. Support and Further Development

For code updates, issue tracking, or collaboration, refer to the GitHub repository or contact the project development team directly : **https://github.com/clement-mendes/T-IOT-902**