Multi-Sensor Weather Station and Environmental-Safety Monitoring System Based on IoT

by **Kosar Ali Janlou**

This project presents a monitoring and alert system based on the Internet of Things (IoT), which continuously monitors environmental quality and various conditions in real time and sends alerts to the user when necessary.

Problem Statement:

In today's world, real-time and precise monitoring of environmental and atmospheric conditions is essential for natural resource management, accident prevention, and safety enhancement. Climate changes and unexpected events such as fires, floods, and storms have highlighted the need for intelligent systems. However, many traditional weather stations:

- Are expensive and require advanced infrastructure.
- Provide data with delays or on a limited scale.
- Cannot be integrated with other environmental safety and monitoring systems.

In industrial, agricultural, or even remote areas, having a system capable of monitoring not only atmospheric data but also environmental conditions such as air quality, ambient light, and fire detection is crucial. Such a system can support more accurate decision-making, reduce risks, and enhance safety.

Project Necessity:

Our project leverages IoT technology and multi-purpose sensors to deliver a smart weather station that:

- Is cost-effective and deployable in various locations.
- Collects and analyzes environmental and atmospheric data in real time.
- Can transmit data to a central server and display it on a responsive and elegant user interface.
- Includes safety modules such as fire detection and air quality measurement, extending its applications beyond conventional stations.

This system is designed not only for weather forecasting and reporting but also for advanced applications such as agricultural planning, water resource management, and early warning of natural disasters.

Detailed Product Definition:

1. **Data Collection from Sensors:**
   The ESP32 microcontroller serves as the system's central processor. It collects data from multiple sensors, including MQ135 (air pollution detection), LDR (ambient light measurement), DHT11 (temperature and humidity measurement), HC-SR04 (distance measurement), and KY-26 (flame detection). The sensor data are formatted, for example as JSON, and transmitted to the Raspberry Pi.
2. **Data Processing and Analysis on Raspberry Pi:**
   The Raspberry Pi functions as a server. It receives data from the ESP32, processes them, and can display them live on a web interface. This allows the user to easily monitor environmental conditions and take necessary actions when required.

3. **Sensor Applications:**
   - MQ135 measures air quality and triggers alerts if high concentrations of harmful gases like CO2 or ammonia are detected.
   - LDR monitors ambient light conditions, enabling the system to update environmental light changes.
   - DHT11 controls temperature and humidity, suitable for sensitive areas such as server rooms or greenhouses.
   - HC-SR04 ultrasonic sensor measures water levels in tanks or distances.
   - KY-26 detects flames or fire, alerting the user in case of potential fire hazards.
4. **Alert and Notification System:**
   If any sensor detects abnormal or hazardous conditions, the system automatically issues an alert. For example, if the MQ135 detects excessive air pollution or the KY-26 identifies flames, the system can send notifications or display the status to the user via the web interface.
5. **Energy Efficiency:**
   The ESP32's "Deep Sleep" feature helps reduce energy consumption. This feature allows the system to enter sleep mode when processing is not required and activate only when necessary.

In this project, by integrating ESP32 and Raspberry Pi, data from various environmental sensors are collected, processed, and displayed in real time. The system can monitor environmental status and issue alerts when needed. Its applications include air quality monitoring, environmental condition assessment, water level measurement, and fire safety.

Product Functions:

The final product is an intelligent environmental monitoring system designed to oversee various environmental conditions, such as air quality, light, temperature, humidity, water level, and fire detection. The system integrates hardware components like Raspberry Pi, ESP32, sensors, and software tools for data management including Python, Flask, and ThingsBoard. Collected sensor data are transmitted to a cloud platform for real-time analysis and live visualization.

1. **Air Quality Monitoring:**
   - Utilizes the MQ135 sensor to measure harmful gases such as NH_3, CO_2, and benzene.
   - Alerts users if dangerous gas concentrations increase.
   - Stores data on ThingsBoard for graphical analysis and visualization.
2. **Ambient Light Measurement:**
   - Uses the LDR sensor to measure environmental light intensity.
   - Detects light changes such as darkness or excessive brightness and sends data to the server.
   - Displays light information on the web interface as live-updating charts.
3. **Temperature and Humidity Measurement:**
   - Employs the DHT11 sensor to measure temperature and humidity.

- Monitors environmental conditions and sends data to the ESP32.
- Can issue alerts for inappropriate temperature or humidity levels, e.g., in temperature-sensitive or humid-sensitive locations.

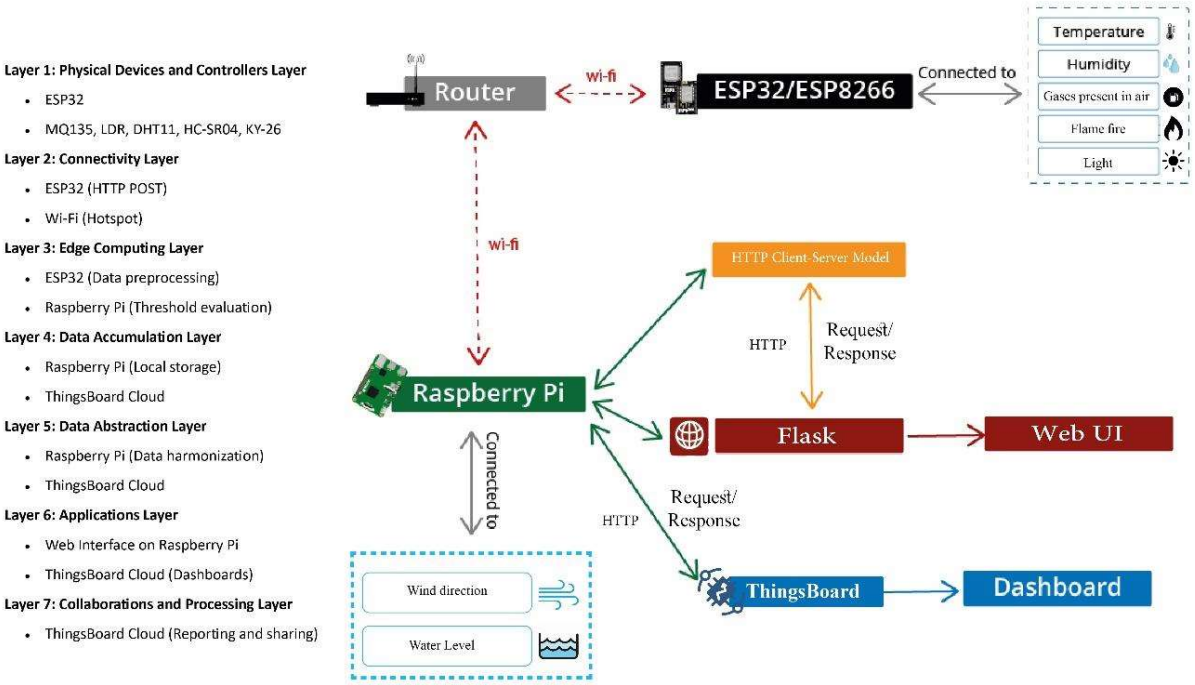
4. **Distance/Water Level Measurement:**
   - Uses the HC-SR04 ultrasonic sensor to measure distances or water levels.
   - Monitors water levels in tanks or aquariums and transmits data to the system.
   - Displays real-time water level changes on the web interface.
5. **Fire and Flame Detection:**
   - Employs the KY-26 IR Fire/Flame sensor to detect fire and flames.
   - Alerts users when fire or flames are detected.
   - Sends signals to the ESP32 for notification and crisis management.
6. **Data Control and Management via Web Interface:**
   - Sensor data are sent to the Raspberry Pi and then forwarded to ThingsBoard.
   - Displays data on a web interface in real time (charts and tables).
   - Enables remote monitoring of environmental conditions via the Internet.
7. **Live Reporting and Alerts:**
   - Sends alerts via email or SMS when abnormal environmental conditions are detected (e.g., increased harmful gases, temperature changes, or flames).
   - Allows users to view historical reports and analyze trends from collected data.
8. **Energy Efficiency:**
   - Utilizes the ESP32 Deep Sleep feature to reduce energy consumption when data processing is not required.
   - Enables automatic energy management and sensor activation via the cloud platform.

Product Features

1. **Real-Time Monitoring:**
   - Users can monitor air quality, temperature, humidity, light, water levels, and fire at any time from any location.
   - Data are continuously updated in real time and easily accessible through a web interface.
2. **Advanced Alarms and Alerts:**
   - Automatically triggers alerts in special conditions (e.g., elevated hazardous gases, extreme temperature changes, or fire detection).
   - Alerts can be sent via SMS, email, or in-system notifications.

3. **Data Analysis:**
   - Collected sensor data are stored and processed on the ThingsBoard platform.
   - Reports and analyses are automatically generated to monitor trends, such as air quality or temperature changes.
4. **Remote-Controlled Devices:**
   - Users can configure and manage alarms or sensor status through the web interface.
   - Supports adding more sensors or custom settings for specific environments, e.g., greenhouses or workshops.
5. **Historical Data Collection and Storage:**
   - Data are systematically stored on ThingsBoard and can be exported in various report formats.
   - Access to historical data allows comparison and trend analysis over time.

Block Diagram and Hardware-Software Layering:



Description of Used Hardware and Software:

1. Hardware:

ESP32 (Microcontroller)
A 32-bit microcontroller based on the Xtensa LX6 architecture, produced by Espressif Systems. This chip features dual cores that can operate simultaneously or independently.

- **Features:**
  - Wi-Fi 802.11 b/g/n (2.4 GHz) and BLE 4.2 support.

    - 34 GPIO pins supporting UART, I2S, I2C, SPI, and ADC/DAC protocols.
    - Deep Sleep capability for energy efficiency.
    - 520 KB internal SRAM and support for external memory.
- **Role in Project:**
    - Collects data from analog sensors (e.g., MQ135, LDR) and digital sensors (e.g., DHT11, KY-26).
    - Converts data into JSON format and sends it via HTTP POST to the Raspberry Pi.
    - Performs edge computing for preliminary data processing to reduce data volume.

Raspberry Pi (Mini Computer)
A versatile mini-computer with a quad-core ARM Cortex-A72 processor capable of running Linux-based operating systems such as Raspbian.

- **Features:**
    - 4 GB or 8 GB DDR4 RAM.
    - GPIO support and communication protocols like SPI, I2C, and UART.
    - USB 3.0 and Ethernet ports for faster connectivity.
- **Role in Project:**
    - Hosts a Flask server to receive and process data from the ESP32.
    - Executes Python scripts to read data from connected ultrasonic sensors.
    - Combines data and displays it via a web interface for live monitoring.
    - Uses GPIO for controlling sensors and additional equipment.

MQ135 (Air Quality Sensor)
A semiconductor gas sensor that detects air pollution using gas-sensitive materials.

- **Features:**
    - Detects harmful gases such as NH_3, benzene, and CO_2.
    - Provides an analog output proportional to the detected gas concentration.
    - Measurement range: 10 ppm to 1000 ppm.
- **Role in Project:**
    - Measures environmental air quality to detect pollution or harmful gases.
    - Sends analog signals to the ESP32 ADC for processing.
    - Issues alerts to users if harmful gas levels rise.

LDR (Light Dependent Resistor)
A resistance-based sensor whose resistance varies with light intensity, measured by the ADC.

- **Features:**
    - Sensitivity range: 1 KΩ to several MΩ depending on ambient light.
    - Sensitive to visible light.
    - Moderate response speed.
- **Role in Project:**
    - Monitors ambient light intensity and detects conditions such as darkness or excessive brightness.
    - Sends data to the ESP32 for logging and web interface visualization.
    - Assists in flame detection and specific environmental condition monitoring.

DHT11 (Temperature and Humidity Sensor)
A low-cost digital sensor that measures temperature and humidity using a resistive humidity sensor and a thermistor.

- **Features:**
    - Temperature range: 0–50 °C with ±2 °C accuracy.
    - Humidity range: 20–80% with ±5% accuracy.
    - Sampling rate: 1 measurement per second.
- **Role in Project:**
    - Measures environmental conditions for temperature and humidity monitoring.
    - Sends data to the ESP32 for logging, server transmission, and web display.
    - Supports alerting in sensitive environments.

Ultrasonic Sensor (HC-SR04)
Measures distance to an object by sending and receiving sound waves.

- **Features:**
    - Measurement range: 2 cm to 400 cm.
    - Accuracy: ±3 mm.
    - Operating frequency: 40 kHz.
- **Role in Project:**
    - Measures water levels in tanks and calculate depth.
    - Supports wind direction sensing by measuring distance from obstacles.
    - Provides accurate data to the Raspberry Pi server for monitoring and decision-making.

KY-26 IR Fire/Flame Detection Sensor
Detects fire by sensing infrared radiation emitted by flames.

- **Features:**
    - Wavelength sensitivity: 760–1100 nm (infrared).
    - Detection angle: approximately 60°.
    - Fast response (<1 second).
- **Role in Project:**
    - Detects flames to identify potential fire hazards.
    - Sends digital signals to the ESP32 for notification and alerting.
    - Integrates into the project's safety system.

2. Software:

ThingsBoard (Cloud Platform)
Used for IoT data management, graphical data visualization, storage, and alerting.

- **Role in Project:**
    - Stores sensor data.
    - Creates dashboards for graphical data representation.
    - Sends alerts if abnormal data changes occur.

Flask (Web Framework)
A Python framework for building and running a web server. In this project, Flask manages HTTP requests and sends data to the web interface.

- **Role in Project:**
    - Handles HTTP requests to receive data from ESP32.
    - Processes data and forwards it to ThingsBoard or displays it on the web interface.

Python (Programming Language)
Used for writing processing scripts and managing communication between hardware and software.

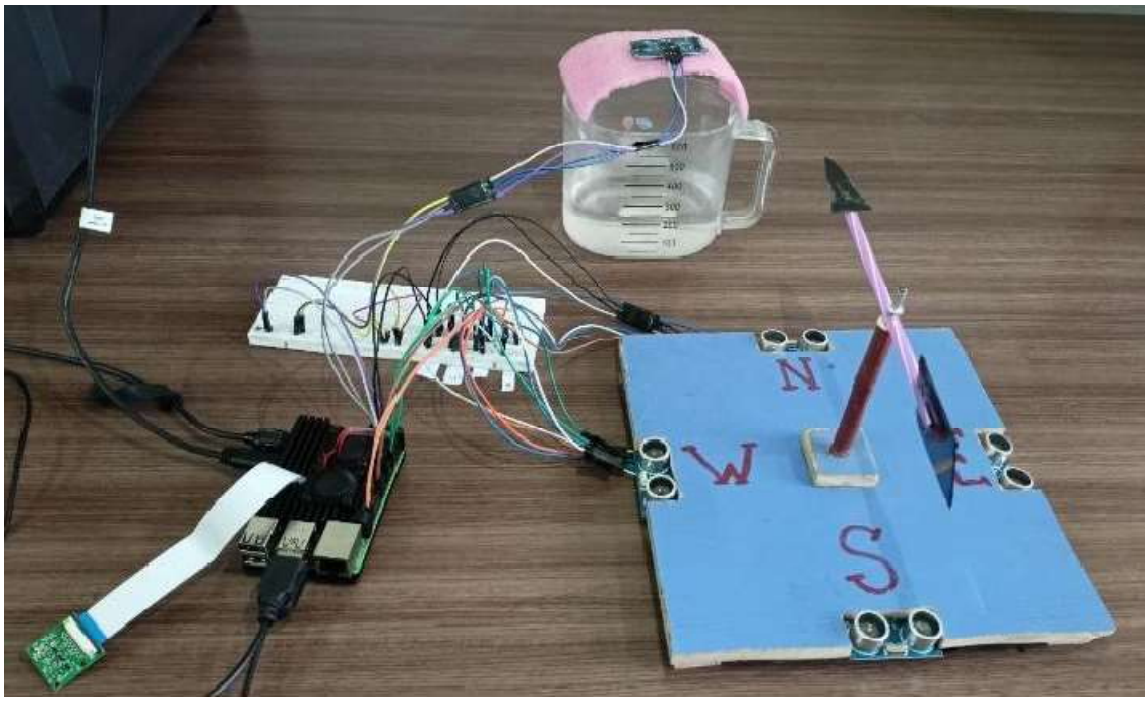
- **Role in Project:**
    - Processes sensor data.
    - Communicates with ThingsBoard and transmits data to the cloud platform.
    - Executes scripts necessary for web-based data visualization.

HTML/CSS/JavaScript (Frontend Technologies)
Used for designing and implementing the web user interface.

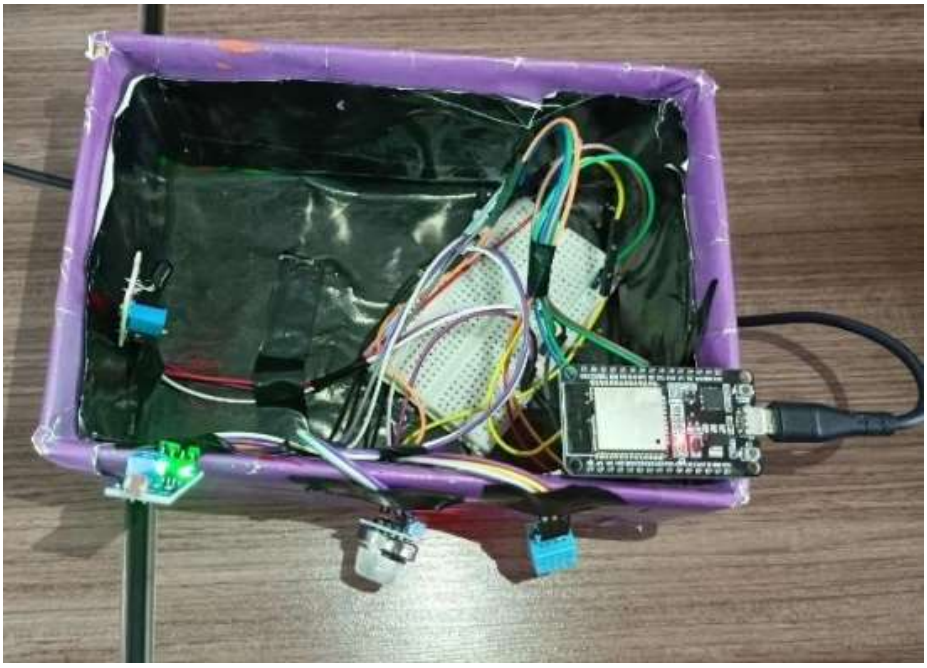
- **Role in Project:**
    - Designs the web page for displaying collected sensor data.
    - Visualizes data graphically through charts and tables.

Product Hardware Images:

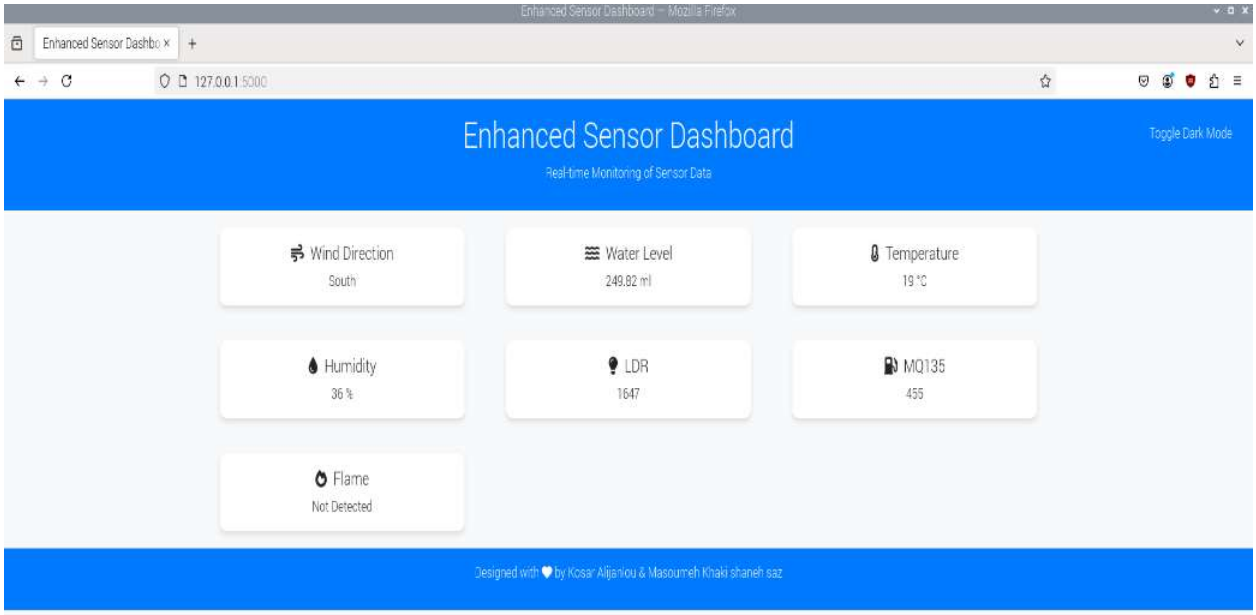
- **Raspberry Pi and Its Connections:**
  Shows the Raspberry Pi connected to ultrasonic sensors for measuring rainfall volume and wind direction:



- **ESP32 and Its Connections:**
  Displays the ESP32 connected to sensors for monitoring temperature, humidity, air pollution, light intensity, and flame/fire detection:



- **Web Interface Display:**
  Demonstrates the received sensor data visualized on a web page in real time:



- **ThingsBoard Visualization:**
  Shows the received data represented as charts on the ThingsBoard platform for graphical monitoring and analysis: