UNIVERSITÀ DEGLI STUDI DE L'AQUILA

Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica

Università degli Studi dell'Aquila

Smart Fan System

2023 / 2024

Course: Software Engineering for Internet of Things

Prof. Davide Di Ruscio

Project GitHub Repository: https://github.com/adam-bouafia/Smart-Fan-Project---SE4IOT

Progetto realizzato da:

Adam Bouafia Matr. ***293137*** Adam.bouafia@student.univaq.it

UNIVERSITÀ DEGLI STUDI DE L'AQUILA

UNIVERSITÀ DEGLI STUDI DE L'AQUILA

Project Specification

The Smart Fan IoT System is an innovative solution designed to enhance environmental control through the integration of home automation technologies. This system enables the precise regulation of temperature and humidity within a designated space, such as residential areas, offices, or specific zones requiring controlled climates like server rooms or storage facilities.

Environmental Control and Regulation

The core function of the Smart Fan IoT System lies in its ability to automatically adjust fan speed based on real-time environmental data, ensuring optimal conditions are maintained. Key parameters such as ambient temperature and air humidity are continuously monitored through high-precision sensors. This proactive approach to environmental management not only promotes comfort and well-being but also contributes to energy efficiency by adapting the fan operation to the current needs.

Measurement Sensor List:

- **Smart Thermometer:** Measures ambient temperature with high accuracy.
- **Air Humidity Sensor:** Tracks air moisture levels to inform fan speed adjustments.
- **Stepper Motor-Controlled Fan:** Acts as both sensor and actuator, adjusting speeds based on input from environmental data.

These components work in harmony to provide a responsive and adaptive system capable of maintaining desired environmental conditions, automatically responding to changes in temperature and humidity.

User Interaction and System Automation

The system is designed to offer both automated control and manual intervention, allowing users to customize settings based on their preferences. Through the integration with Node-RED and Telegram, users receive real-time updates and notifications on environmental conditions, enabling them to manually adjust fan speeds or set specific conditions for automatic adjustments.

Dashboard and Real-Time Data Visualization

A critical feature of the Smart Fan IoT System is its user-friendly dashboard, powered by Grafana, which displays real-time data on temperature, humidity, and fan speed. This dashboard not only allows users to monitor current environmental conditions but also provides insights into historical data trends, aiding in the further optimization of the system settings.

Cloud Deployment and System Scalability

With a focus on flexibility and scalability, the Smart Fan IoT System is containerized using Docker, facilitating easy deployment across various platforms, including cloud services. This ensures that the system can be scaled to accommodate additional sensors or expanded to cover larger areas without compromising performance. The cloud-based deployment model also addresses non-functional requirements such as system security, data privacy, and the scalability of server-side hardware, entrusting these aspects to the cloud service provider.

Conclusion

The Smart Fan IoT System represents a significant advancement in environmental control technology, offering a highly adaptive, efficient, and user-friendly solution for maintaining optimal conditions. By leveraging IoT technologies, this system not only automates fan speed regulation based on precise temperature and humidity measurements but also empowers users with real-time data visualization and interaction capabilities, setting a new standard for personal and commercial environmental management.

Functional Requirements

1 - Environmental Monitoring

Priority: High

The system continuously monitors and displays the following environmental parameters in real-time:

- **Air Humidity:** Measures the moisture level in the air to ensure optimal humidity.
- **Temperature:** Monitors ambient temperature to maintain comfortable conditions.

These parameters are crucial for adjusting the fan speed to maintain desired environmental conditions automatically.

2 - User Notifications

Priority**: High**

Alerts the user via Telegram notifications when temperature or humidity levels exceed predefined thresholds. This function ensures users can take immediate action or adjust settings if the system's automatic responses need further intervention.

3 - Manual Control

Priority: High

Through the system's interface (Node-RED dashboard and Telegram bot), users can manually control the fan speed. This feature allows users to customize the environmental conditions according to their preferences or specific requirements beyond the system's automatic adjustments.

4 - Data Visualization and Analysis

Priority: Medium

Offers users the ability to view real-time and historical data through a Grafana dashboard. This capability enables users to analyze trends in temperature and humidity over time, facilitating informed decisions on environmental control strategies.

5 - System Scalability and Integration

Priority: Medium

The system should be designed to easily integrate additional sensors or expand its capabilities without significant modifications. This includes scalability in terms of software (Node-RED flows, MQTT topics) and hardware (more complex sensor networks, additional fans), ensuring the system can grow with the user's needs.

Non Functional Requirements

High-Performance Database

- **Priority**: **High**
- **Description:** The Smart Fan IoT System generates continuous streams of temperature and humidity data. To efficiently handle, store, and process this vast amount of information, a high-performance time-series database is required. InfluxDB is selected for its excellent capabilities in handling time-series data, enabling fast data ingestion, efficient storage, and real-time queries and aggregation functions. This ensures the system can perform complex data analysis and visualization tasks with minimal latency, supporting effective environmental monitoring and control.

Usability

- **Priority: High**
- **Description:** The system interfaces, including the Node-RED dashboard for control and the Grafana dashboard for data visualization, must be user-friendly and intuitive. Users should be able to easily interact with the system, configure settings, and understand environmental data at a glance. This encompasses clear navigation, responsive design, and comprehensive documentation to ensure a seamless user experience.

Reusability

- **Priority: Medium**
- **Description:** Components and modules within the Smart Fan IoT System should be designed for reusability, facilitating the easy expansion of the system or adaptation to different environments and applications. This includes modular code in the Node-RED flows, easily configurable MQTT topics, and Docker containers that encapsulate system components for deployment in varied infrastructures without the need for substantial modifications.

Cross-Platform Compatibility

- **Priority: Medium**
- Description: The Smart Fan IoT System must be operable across different platforms and devices, ensuring users can access and control the system from various operating systems (such as Windows, macOS, Linux) and devices (including PCs, smartphones, tablets). This broad compatibility is achieved through web-based interfaces for the Node-RED and Grafana dashboards and ensures the Telegram bot notifications are accessible on all devices supporting the Telegram application.

UNIVERSITÀ DEGLI STUDI DE L'AQUILA

SYSTEM ARCHITECTURE

Component Diagram

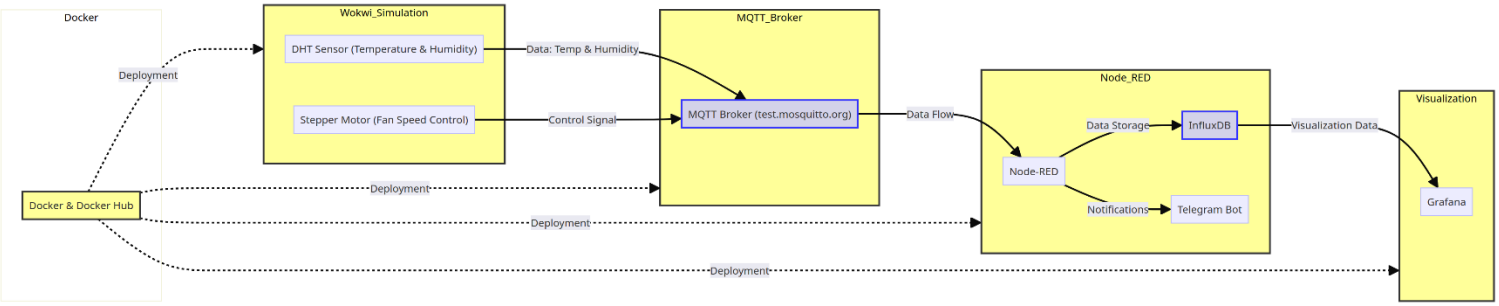

Figure 2: Smart Fan System - Component Diagram

- **Sensors**: Temperature and humidity sensors for data collection.
- **Fan Control Mechanism**: Stepper motor-controlled fan, with speed adjustments based on sensor inputs.
- **MQTT Broker:** test.mosquitto.org used for messaging between IoT devices and the Node-RED server.
- **Node-RED:** Implements logic for processing sensor data, controlling fan speed, and managing user notifications.
- **Telegram Bot:** Provides an interface for real-time alerts and system status updates.
- **InfluxDB:** Database for storing environmental data over time.
- **Grafana:** Dashboard for displaying real-time and historical data.
- **Docker:** Containers for each component to facilitate deployment and scaling.

Sensors and Actuators

Sensors

The Smart Fan IoT System employs two primary types of sensors to monitor environmental conditions accurately. These sensors are integral for collecting real-time data, which is then used to automate the control of the fan speed, ensuring optimal temperature and humidity levels.

Air Humidity Sensor

- **Function**: Measures the moisture content in the air.
- **Role:** Provides data to adjust the fan speed to maintain or achieve desired humidity levels, crucial for comfort and specific environmental requirements.

Temperature Sensor (Smart Thermometer)

- **Function**: Measures the ambient temperature.
- **Role:** Essential for monitoring the environment's temperature, allowing the system to increase or decrease the fan speed to keep the temperature within a comfortable and predetermined range.

Actuators

Stepper Motor-Controlled Fan

- **Function:** The fan's speed is regulated by a stepper motor, which can precisely adjust the fan's operation based on input from the system's control logic.
- **Role:** Acts as the primary actuator in the system, directly influencing the environment by adjusting airflow based on real-time sensor data. The fan speed is dynamically controlled to cool down or circulate air to balance the temperature and humidity, ensuring the environmental conditions meet the user's settings and preferences.

Integration and Control Logic

The sensors continuously transmit data to the system, where the control logic, implemented through Node-RED, processes this information. Depending on the current readings and predefined thresholds, Node-RED sends commands to the stepper motor to adjust the fan speed accordingly. This automated process is central to the Smart Fan IoT System's capability to maintain optimal environmental conditions.

MQTT Broker

Overview

The MQTT (Message Queuing Telemetry Transport) broker serves as the central nerve of the Smart Fan IoT System, facilitating seamless communication between the system's components. It ensures the efficient transmission of data collected by sensors to the processing unit (Node-RED), and subsequently, to the actuator (stepper motor-controlled fan), as well as forwarding notifications and alerts to the user through the Telegram bot.

Functionality

- **Data Transmission:** The MQTT broker handles messages containing temperature and humidity data from the sensors. This data is published under specific topics, which Node-RED subscribes to, allowing for real-time data processing and decision-making based on current environmental conditions.
- **System Commands:** Node-RED publishes commands to topics that the fan's control system is subscribed to. These commands dictate the fan's operational status—whether to increase, decrease, or maintain the current speed—based on the analysis of incoming sensor data.
- **User Notifications:** Alerts and updates are formulated in Node-RED based on the logic defined for temperature and humidity thresholds. These notifications are then published to a topic that the Telegram bot subscribes to, ensuring that users are promptly informed about significant environmental changes or system statuses.

Role in the Smart Fan IoT System

- **Central Communication Hub:** Acts as the intermediary for all message exchanges within the system, ensuring data integrity and timely delivery.
- **Scalability and Flexibility:** Supports the addition of more sensors or actuators by simply subscribing them to relevant MQTT topics, allowing the system to expand its monitoring and control capabilities without significant reconfiguration.

- **Reliability and QoS:** Ensures reliable message delivery even in fluctuating network conditions, with Quality of Service (QoS) levels that can be adjusted to prioritize certain messages over others, crucial for critical alerts or commands.

Implementation

- **Broker Selection:** The Smart Fan IoT System utilizes **test.mosquitto.org** as its MQTT broker, a widely recognized and reliable public broker that facilitates easy setup and robust performance for IoT projects.
- **Security Considerations:** While **test.mosquitto.org** is used for demonstration and development purposes, for production environments, it's recommended to deploy a private MQTT broker or select a secure, managed MQTT service that offers enhanced authentication and encryption features to protect sensitive data and ensure privacy.

Integration with Other Components

The MQTT broker's effectiveness lies in its seamless integration with the rest of the Smart Fan IoT System's components. By serving as the communication backbone, it enables the dynamic and intelligent control of the fan based on real-time environmental data, alongside providing a platform for user interaction through instant notifications and alerts.

Real Time display and interaction

Overview

The Smart Fan IoT System offers a real-time display and interaction interface through a Node-RED dashboard and Telegram bot. This setup enables users to monitor current environmental conditions—specifically temperature and humidity—and manually adjust fan speed settings as needed.

Node-RED Dashboard

- **Functionality:** Provides a live view of temperature and humidity readings, along with the current fan speed. Users can interact with this dashboard to manually override automatic fan speed adjustments, setting it to their preferred levels based on real-time data.
- **User Interaction:** The dashboard is designed for ease of use, allowing for quick adjustments and real-time feedback on system responses. It includes sliders for fan speed, toggle switches for manual or automatic control, and displays for current sensor readings.

Telegram Bot

- **Alerts and Notifications:** Sends instant notifications to users when temperature or humidity exceeds predefined thresholds. This feature ensures that users are always aware of significant changes in environmental conditions, even when not actively monitoring the Node-RED dashboard.
- **Remote Control:** Allows users to send commands via Telegram to adjust the fan speed remotely. This adds an additional layer of convenience, enabling environmental control even when away from the primary interface.

Integration for Seamless Control

- Both interfaces are integrated with the MQTT broker and the system's logic processing in Node-RED, ensuring that any adjustments or commands are immediately reflected in the system's operation. This real-time communication is vital for maintaining optimal environmental conditions.

Dashboard for Historic Data

Overview

- Grafana is utilized to create a comprehensive dashboard for historical data visualization, offering users insights into the environmental conditions over time. This tool is instrumental in identifying trends, patterns, and the effectiveness of the Smart Fan IoT System in maintaining desired conditions.

Features

- **Data Visualization:** Grafana provides various chart types (e.g., line graphs, bar charts) to represent historical data of temperature, humidity, and fan speeds. This allows for easy comparison and analysis of environmental conditions over different periods.
- **Customization and Filtering:** Users can customize the dashboard to focus on specific data ranges, compare data points, and filter information based on their analysis needs.
- **Insight and Optimization:** By reviewing historical data, users can make informed decisions on adjusting the system's settings or thresholds for better environmental control.

Benefits

- **Informed Decision Making:** Access to historical data empowers users to optimize the environmental conditions based on past performance and trends.
- **System Performance Evaluation:** Enables the evaluation of the Smart Fan IoT System's effectiveness and efficiency over time, guiding potential system improvements or adjustments.

Grafana - Real Time Dashboard

UNIVERSITÀ DEGLI STUDI DE L'AQUILA

How Functional Requirements have been addressed

Functional Requirement #1 - Monitoring environment

The NodeRed sensor flows and OpenHab UI allows the user to actively monitor the greenhouse status in real time. The sensors are all the required ones.

Implementation:

- **Sensors:** Utilize air humidity and temperature sensors to constantly monitor the environmental conditions. This real-time data is crucial for the system to make informed decisions.
- **MQTT and Node-RED Integration:** Sensor data is transmitted to Node-RED via the MQTT broker. Node-RED analyzes this data against predefined thresholds to determine the necessary adjustments.

Outcome: The system ensures continuous monitoring, enabling automated adjustments to fan speed in response to fluctuations in temperature and humidity, maintaining optimal environmental conditions.

Functional Requirement #2 - Automatic Fan Speed Adjustment

Implementation:

- **Control Logic in Node-RED:** Based on the current temperature and humidity levels, Node-RED calculates the required fan speed and sends a command to adjust the fan accordingly.
- **Stepper Motor-Controlled Fan:** Receives the speed adjustment commands from Node-RED and alters its speed to match the required settings, ensuring the environmental conditions remain within the desired ranges.

Outcome: This process allows the system to automatically control the fan speed, efficiently managing the environment without the need for manual intervention, ensuring comfort and energy efficiency.

Functional Requirement #3 - User Notifications and Manual Overrides

Implementation:

- **User Alerts:** When environmental conditions approach or exceed set thresholds, the system sends alerts to users via the Telegram bot, notifying them of potential adjustments or conditions that may require attention.
- **Manual Control:** Despite the system's automated capabilities, users can manually adjust fan speeds or change thresholds via the Node-RED dashboard or Telegram bot, offering flexibility and personalization.

Outcome: Users remain informed and in control, able to intervene manually if desired, while primarily benefiting from the system's automated environmental management.

Functional Requirement #4 - Data Visualization and Analysis

Implementation:

- **Grafana Dashboard for Historical Data:** Integrates with InfluxDB to visualize historical temperature, humidity, and fan speed data. This setup allows users to analyze trends and the effectiveness of the system's automatic adjustments over time.
- **Insights and Optimization:** The dashboard enables users to identify patterns, assess system performance, and fine-tune thresholds or settings for improved environmental control.

Outcome: Providing access to detailed environmental data over time empowers users to optimize conditions based on historical performance and trends, enhancing the overall efficacy of the Smart Fan IoT System.

TECHNOLOGIES