**Mục Lục**

**Hệ thống kiến trúc**

[1 Giới thiệu 2](#_Toc190896175)

[1.1 Tổng quan về tài liệu 2](#_Toc190896176)

[1.2 Từ viết tắt và thuật ngữ. 2](#_Toc190896177)

[1.2.1 Từ viết tắt 2](#_Toc190896178)

[1.2.2 Thuật ngữ 2](#_Toc190896179)

[1.3 Tài liệu tham khảo 2](#_Toc190896180)

[1.3.1 Tài liệu tham khảo về dự án 2](#_Toc190896181)

[1.3.2 Tài liệu tham khảo về tiêu chuẩn quy định 2](#_Toc190896182)

[1.4 Quy ước 3](#_Toc190896183)

[2 Kiến trúc 4](#_Toc190896184)

[2.1 Tổng quan về kiến trúc 4](#_Toc190896185)

[2.2 Tổng quan về kiến trúc logic 5](#_Toc190896186)

[2.2.1 Mô tả thành phần phần mềm 5](#_Toc190896187)

[2.3 Tổng quan về kiến trúc vật lý 6](#_Toc190896190)

[2.3.1 Mô tả thành phần phần cứng 6](#_Toc190896191)

[2.4 Phần mềm SOUP 7](#_Toc190896194)

[3 Hành vi động của kiến trúc 9](#_Toc190896199)

[4 Lý do chính đáng cho kiến trúc 10](#_Toc190896204)

[5 Khả năng truy xuất yêu cầu 12](#_Toc190896210)

# Giới thiệu

## Tổng quan về tài liệu

Tài liệu này mô tả kiến trúc của hệ thống hiển thị nhiệt độ, độ ẩm phòng và dự báo mưa.

* Mô tả tổng quan về hệ thống, bao gồm mục đích, chức năng chính và thiết kế tổng thể.
* Các thành phần của hệ thống, như cảm biến, bộ xử lý,mô-đun truyền thông và giao diện người dùng.
* Luồng dữ liệu và sự tương tác giữa các thành phần, bao gồm thu thập dữ liệu cảm biến, xử lý, kích hoạt cảnh báo và cơ chế thông báo.
* Các công nghệ và giao thức được sử dụng để truyền dữ liệu, lưu trữ và hiển thị.
* Các yếu tố liên quan đến độ tin cậy, khả năng mở rộng và bảo mật của hệ thống.

## Từ viết tắt và thuật ngữ.

### Từ viết tắt

Add here abbreviations

COTS: Components Off the Shelf (software industry acronym)

OTSS: Off The Shelf Software (FDA acronym)

SOUP: Software Of Unknown Provenance (IEC 62304 acronym)

### Thuật ngữ

Add here words definitions

## **Tài liệu tham khảo**

### Tài liệu tham khảo về dự án

| # | Document Identifier | Document Title |
| --- | --- | --- |
| [R1] | Wireless Sensor Networks | Ian F. Akyildiz & Mehmet Can Vuran  Cung cấp kiến thức về mạng cảm biến không dây, nền tảng của các hệ thống giám sát môi trường. |
| [R2] | Smart Sensors and Systems | S. Mukhopadhyay, T. Islam  Tổng quan về cảm biến thông minh và cách tích hợp chúng vào hệ thống giám sát. |
| [R3] | Internet of Things: Principles and Paradigms | Rajkumar Buyya, Amir Vahid Dastjerdi  Giới thiệu tổng quan về IoT và các nguyên tắc thiết kế hệ thống giám sát thông minh. |

### Tài liệu tham khảo về tiêu chuẩn quy định

|  |  |  |
| --- | --- | --- |
| # | Document Identifier | Document Title |
| [STD1] | ISO 9241-210:2019 | Ergonomics of human-system interaction  Hướng dẫn về thiết kế hệ thống có tính đến yếu tố con người, hữu ích cho việc phát triển giao diện người dùng của hệ thống giám sát. |
| [STD2] | ISO 50001:2018 | Energy Management Systems  Tiêu chuẩn quản lý năng lượng, giúp tối ưu hóa hệ thống giám sát trong các môi trường công nghiệp. |
| [STD3] | ASHRAE Standard 55-2020 | Thermal Environmental Conditions for Human Occupancy  Tiêu chuẩn về nhiệt độ và độ ẩm tối ưu cho không gian làm việc và sinh hoạt. |
| [STD4] | FDA 21 CFR Part 11 | Electronic Records; Electronic Signatures  Tiêu chuẩn về lưu trữ dữ liệu điện tử, hữu ích nếu hệ thống yêu cầu lưu trữ dữ liệu môi trường để phân tích hoặc báo cáo |

## Quy ước

# Kiến trúc

## Tổng quan về kiến trúc

Quan điểm của người dùng:

**- Môi trường hoạt động:** Hệ thống được thiết kế để hoạt động trong các môi trường yêu cầu giám sát nhiệt độ và độ ẩm và lượng mưa liên tục trong đời sống.

- **Người dùng:**Cá nhân quan tâm đến môi trường sống của họ

**- Mục đích sử dụng:**

Theo dõi ghi nhận sử lý và hiển thị nhiệt độ, độ ẩm tỉ lệ xảy ra mưa theo thời gian thực đồng thời Lưu trữ dữ liệu để phân tích, dự đoán xu hướng môi trường

**- Các chức năng chính:**

Đo lường nhiệt độ và độ ẩm: Cảm biến đo các thông số này và truyền dữ liệu về hệ thống xử lý.

Hiển thị thông tin: Dữ liệu được hiển thị trên màn hình LCD hoặc giao diện web/mobile.

Lưu trữ và truy xuất dữ liệu: Hệ thống lưu trữ dữ liệu để người dùng có thể xem lịch sử và phân tích.

Kết nối từ xa: Hỗ trợ kết nối Wi-Fi/Bluetooth để người dùng có thể theo dõi qua ứng dụng di động hoặc trình duyệt web.

**- Các giao diện, đầu vào và đầu ra chính:**

**Đầu vào:**

- Cảm biến nhiệt độ, độ ẩm (, DHT22, SHT31…)

- Kết nối Wi-Fi/Bluetooth để nhận lệnh từ người dùng

**Đầu ra:**

Màn hình LCD/OLED hiển thị thông tin nhiệt độ, độ ẩm...

Gửi thông báo qua ứng dụng di động hoặc email

Nếu hệ thống được tích hợp trong một hệ thống lớn hơn (ví dụ: hệ thống giám sát môi trường thông minh trong bệnh viện hoặc nhà máy), tài liệu mô tả chi tiết về cách thức kết nối và tích hợp sẽ được tham khảo riêng

## Tổng quan về kiến trúc logic

Các thành phần phần cứng

* Cảm biến nhiệt độ và độ ẩm: DHT11, DHT22 hoặc SHT31 để đo lường nhiệt độ và độ ẩm.
* Vi điều khiển: ESP32 hoặc Arduino để xử lý dữ liệu và giao tiếp với các thiết bị khác.
* Màn hình hiển thị: OLED, LCD để hiển thị thông tin nhiệt độ và độ ẩm.
* Kết nối không dây: Mô-đun Wi-Fi/Bluetooth tích hợp trên vi điều khiển để giao tiếp từ xa.
* Nguồn điện: Adapter 5V/9V hoặc pin dự phòng.

Tương tác và mối quan giữa các thành phần

* Cảm biến gửi dữ liệu đến vi điều khiển thông qua giao tiếp I2C hoặc GPIO.
* Vi điều khiển xử lý dữ liệu và quyết định hiển thị thông tin.
* Dữ liệu hiển thị trên màn hình LCD/OLED để người dùng theo dõi.
* Vi điều khiển gửi dữ liệu lên đám mây hoặc ứng dụng di động thông qua Wi-Fi/Bluetooth để theo dõi từ xa.

Sơ đồ minh họa

* Sơ đồ thành phần: Mô tả các thành phần phần cứng và phần mềm.
* Sơ đồ triển khai: Mô tả cách thức triển khai hệ thống trong thực tế.
* Sơ đồ mạng: Mô tả cách các thiết bị giao tiếp với nhau qua mạng nội bộ hoặc Internet.
* Sơ đồ giao diện: Minh họa giao diện ứng dụng di động hoặc web hiển thị dữ liệu.

### Mô tả thành phần phần mềm

Phần mềm nhúng trên vi điều khiển

 Chức năng:

* Thu thập dữ liệu từ cảm biến nhiệt độ, độ ẩm.
* Xử lý sơ bộ dữ liệu (lọc nhiễu, hiệu chỉnh).
* Gửi dữ liệu đến máy chủ qua giao thức truyền thông (MQTT, HTTP).
* Hiển thị thông tin nhiệt độ và độ ẩm và tỉ lệ mưa

 Công nghệ sử dụng:

* Lập trình bằng C/C++ (Arduino IDE, ESP-IDF) hoặc Python.
* Chạy trên vi điều khiển ESP32, Raspberry Pi.

Hệ thống máy chủ và cơ sở dữ liệu

 Chức năng:

* Nhận dữ liệu từ vi điều khiển qua giao thức MQTT/HTTP.
* Lưu trữ dữ liệu vào cơ sở dữ liệu (SQL hoặc NoSQL).
* Phân tích dữ liệu, phát hiện bất thường và quyết định kích hoạt cảnh báo.

 Công nghệ sử dụng:

* Backend: Node.js, Python (Flask/Django).
* Cơ sở dữ liệu: Firebase, MongoDB, MySQL.
* Chạy trên máy chủ cục bộ hoặc cloud (AWS, Google Cloud).

Giao diện hiển thị và cảnh báo

 Chức năng:

* Hiển thị dữ liệu nhiệt độ, độ ẩm theo thời gian thực.
* Cung cấp giao diện điều khiển cho người dùng (web/mobile).
* Gửi cảnh báo qua email, SMS hoặc ứng dụng khi có bất thường.

 Công nghệ sử dụng:

* Web: React.js, Vue.js.
* Mobile: Flutter, React Native.
* API kết nối với backend qua RESTful API hoặc WebSocket.

## Tổng quan về kiến trúc vật lý

Hệ thống theo dõi nhiệt độ, độ ẩm và xác định tỉ lệ mưa gồm nhiều thành phần phần cứng, đảm bảo thu thập dữ liệu chính xác, truyền thông ổn định và hiển thị cảnh báo kịp thời.

### Mô tả thành phần phần cứng

Cảm biến nhiệt độ, đọ ẩm

 Chức năng: Đo nhiệt độ và độ ẩm từ môi trường.

 Thiết bị sử dụng:

* DHT22: Đo nhiệt độ (-40°C đến 80°C) và độ ẩm (0-100% RH).
* SHT31: Độ chính xác cao hơn, giao tiếp I2C.

 Giao tiếp: Kết nối với vi điều khiển qua giao thức I2C, 1-Wire hoặc Analog.

Vi điều khiển/ Bo mạch xử lý

* Chức năng: Thu thập dữ liệu từ cảm biến, xử lý và gửi lên máy chủ.
* Thiết bị sử dụng:

- ESP32: Vi điều khiển hỗ trợ Wi-Fi, Bluetooth, giao tiếp nhiều cảm biến.

* Giao tiếp: Kết nối cảm biến qua GPIO, I2C, UART, giao tiếp với máy chủ qua Wi-Fi, Ethernet, LoRa.

Hệ hiện thị

 Chức năng: hiển thị dữ liệu thời gian thực.

 Thiết bị sử dụng:

* LCD I2C: Hiển thị thông tin trực tiếp trên thiết bị.

 Giao tiếp: Kết nối với vi điều khiển qua GPIO, I2C, PWM

## Phần mềm SOUP

Trong hệ thống theo dõi nhiệt độ, độ ẩm và cảnh báo, có một số phần mềm SOUP (Software of Unknown Provenance) được sử dụng để hỗ trợ vận hành hệ thống. Dưới đây là danh sách và mô tả chi tiết từng thành phần SOUP:

### **Hệ điều hành nhúng**

* **Nhận dạng**: Raspbian OS (Linux) trên Raspberry Pi, FreeRTOS trên ESP32.
* **Mục đích**: Cung cấp nền tảng chạy cho phần mềm thu thập và xử lý dữ liệu từ cảm biến.
* **Nguồn gốc**:

**- Raspbian OS**: Được phát triển bởi Raspberry Pi Foundation.

**- FreeRTOS**: Phát triển bởi Amazon Web Services (AWS).

* **Duy trì & hỗ trợ**: Được cập nhật bởi cộng đồng phát triển.
* **Tài nguyên phần cứng sử dụng**:

- Raspberry Pi 4 (CPU 1.5GHz, RAM 4GB, bộ nhớ microSD).

- ESP32 (CPU dual-core, RAM 520KB).

* **Giao diện & luồng dữ liệu**: Hệ điều hành nhận dữ liệu từ vi điều khiển và gửi lên máy chủ thông qua Wi-Fi hoặc Ethernet.

### **Thư viện phần mềm nhúng & giao tiếp**

* **Nhận dạng**: Arduino Core, ESP-IDF (ESP32), Paho MQTT (giao tiếp MQTT).
* **Mục đích**:

- Hỗ trợ lập trình và điều khiển cảm biến nhiệt độ, độ ẩm.

- Quản lý kết nối mạng và giao tiếp giữa thiết bị và máy chủ.

* **Nguồn gốc**:

**- Arduino Core**: Mã nguồn mở do Arduino phát triển.

**- ESP-IDF**: Framework chính thức của Espressif Systems dành cho ESP32.

**- Paho MQTT**: Thư viện giao tiếp MQTT do Eclipse Foundation phát triển.

* **Duy trì & hỗ trợ**: Được cập nhật định kỳ bởi các nhà phát triển và cộng đồng.
* **Giao diện & luồng dữ liệu**:

- Arduino Core và ESP-IDF thu thập dữ liệu từ cảm biến.

- Paho MQTT truyền dữ liệu từ thiết bị đến máy chủ.

### **Cơ sở dữ liệu & nền tảng lưu trữ**

* **Nhận dạng**: Firebase Realtime Database, MongoDB.
* **Mục đích**: Lưu trữ dữ liệu nhiệt độ, độ ẩm và quản lý thông tin cảnh báo.
* **Nguồn gốc**:

**- Firebase**: Phát triển bởi Google.

**- MongoDB**: Hệ quản trị cơ sở dữ liệu NoSQL phổ biến.

* **Duy trì & hỗ trợ**: Do Google và MongoDB Inc. cung cấp dịch vụ và cập nhật bảo mật.
* **Tài nguyên phần cứng sử dụng**: Chạy trên nền tảng cloud của Google hoặc server riêng.
* **Giao diện & luồng dữ liệu**:

- Nhận dữ liệu từ vi điều khiển thông qua MQTT/HTTP.

- Lưu trữ và phân tích dữ liệu, kích hoạt cảnh báo khi vượt ngưỡng.

### **Cách tích hợp SOUP vào hệ thống**

* **Các thành phần SOUP được tích hợp vào phần mềm nhúng, máy chủ và giao diện web** để hỗ trợ thu thập, xử lý và hiển thị dữ liệu.
* **Các biện pháp kiểm soát rủi ro khi sử dụng SOUP**:

- Định kỳ kiểm tra và cập nhật phiên bản phần mềm để giảm thiểu lỗ hổng bảo mật.

- Kiểm thử hiệu suất và độ ổn định trước khi triển khai thực tế.

- Xác định phương án thay thế nếu có sự cố với phần mềm SOUP.

# Hành vi động của kiến trúc

Phần này mô tả cách hệ thống hoạt động theo trình tự các bước, bao gồm cách các thành phần phần cứng và phần mềm tương tác với nhau để theo dõi nhiệt độ nhằm đưa ra dự đoán tỉ lệ mưa

## **Quy trình làm việc**

### **Thu thập và xử lý dữ liệu cảm biến**

**Mô tả**: Quy trình này mô tả cách hệ thống thu thập dữ liệu từ cảm biến, xử lý và gửi dữ liệu đến máy chủ.

#### **Trình tự hoạt động:**

1. **Cảm biến nhiệt độ và độ ẩm (DHT22/SHT31)** đo giá trị môi trường theo chu kỳ (ví dụ: mỗi 5 giây).
2. **Vi điều khiển (ESP32/Raspberry Pi)** nhận dữ liệu từ cảm biến thông qua giao thức I2C hoặc GPIO.
3. **Vi điều khiển xử lý dữ liệu**:

- Kiểm tra tính hợp lệ của dữ liệu.

- Chuyển đổi đơn vị nếu cần thiết.

- Lọc nhiễu để giảm sai số.

1. **Vi điều khiển gửi dữ liệu lên máy chủ** bằng giao thức MQTT hoặc HTTP REST API.
2. **Máy chủ xử lý và lưu trữ dữ liệu** trong cơ sở dữ liệu Firebase/MongoDB.
3. **Người dùng có thể truy cập dữ liệu** qua ứng dụng web hoặc mobile.

#### **Trình tự hoạt động:**

1. **Máy chủ liên tục theo dõi dữ liệu nhiệt độ và độ ẩm** từ cơ sở dữ liệu.
2. Khi phát hiện **giá trị vượt ngưỡng** (ví dụ: nhiệt độ > 40°C hoặc độ ẩm < 20%):
   * Máy chủ gửi thông báo đến thiết bị cảnh báo.
   * Hiển thị thông báo trên giao diện web/app.
3. **Vi điều khiển nhận lệnh từ máy chủ**, kích hoạt thiết bị cảnh báo:
   * **Buzzer phát âm thanh cảnh báo**.
   * **LED RGB nhấp nháy đỏ** để báo hiệu tình trạng nguy hiểm.
   * **Màn hình LCD hiển thị thông báo** về nhiệt độ và độ ẩm hiện tại.
4. **Người dùng nhận được cảnh báo qua ứng dụng mobile/email/SMS**.
5. Khi điều kiện trở lại bình thường, hệ thống **tắt cảnh báo** và ghi nhận sự kiện vào lịch sử hoạt động.

# Lý do chính đáng cho kiến trúc

### ****4.1 Khả năng kiến trúc hệ thống****

Hệ thống theo dõi nhiệt độ và độ ẩm được thiết kế theo mô hình **IoT phân tán**, bao gồm các thiết bị cảm biến, vi điều khiển, máy chủ lưu trữ và giao diện người dùng.

#### **Lý do lựa chọn kiến trúc:**

* **Tính mở rộng**: Có thể dễ dàng thêm nhiều cảm biến hoặc thiết bị đầu cuối.
* **Hiệu suất cao**: Dữ liệu được xử lý nhanh chóng tại thiết bị và gửi lên máy chủ theo thời gian thực.
* **Khả năng bảo trì**: Các thành phần phần cứng và phần mềm có thể được nâng cấp hoặc thay thế mà không ảnh hưởng toàn bộ hệ thống.

### **4.2 Khả năng kiến trúc mạng**

Hệ thống sử dụng giao thức MQTT và HTTP REST API để truyền dữ liệu giữa các thành phần.

#### **Lý do lựa chọn kiến trúc mạng:**

* **MQTT**: Nhẹ, tiết kiệm băng thông, phù hợp với thiết bị IoT có tài nguyên hạn chế.
* **HTTP REST API**: Tích hợp dễ dàng với các nền tảng web và mobile.
* **Wi-Fi/Ethernet**: Đảm bảo kết nối ổn định, có thể mở rộng bằng cách thêm Gateway hoặc sử dụng mạng 4G/5G.

#### **Độ tin cậy của mạng**

* Dữ liệu được gửi kèm theo timestamp để xử lý tình huống mất gói tin.
* Cơ chế **retry và xác nhận** đảm bảo dữ liệu được truyền thành công.
* Máy chủ có cơ chế **sao lưu dữ liệu** để tránh mất thông tin quan trọng.

### **4.3 Đầu ra phân tích rủi ro**

Hệ thống được đánh giá rủi ro theo tiêu chuẩn IEC 62304 nhằm đảm bảo an toàn và ổn định.

#### **Các rủi ro chính và biện pháp giảm thiểu:**

1. **Sai số hoặc lỗi cảm biến** → Sử dụng bộ lọc nhiễu và kiểm tra nhiều lần trước khi ghi nhận dữ liệu.
2. **Mất kết nối mạng** → Tích hợp cơ chế lưu trữ cục bộ (buffer) và gửi lại khi mạng ổn định.
3. **Tấn công an ninh mạng** → Mã hóa dữ liệu và xác thực người dùng bằng JWT hoặc OAuth2.
4. **Hỏng phần cứng** → Cung cấp thiết bị dự phòng và cơ chế tự động phát hiện lỗi.

### **4.4 Đầu ra kỹ thuật yếu tố con người**

Hệ thống được thiết kế để tối ưu trải nghiệm người dùng, giúp dễ dàng theo dõi và phản ứng với cảnh báo.

#### **Các yếu tố được xem xét:**

* **Giao diện trực quan**: Hiển thị dữ liệu theo dạng biểu đồ, màu sắc dễ hiểu.
* **Cảnh báo đa kênh**: Gửi thông báo qua email, ứng dụng di động hoặc tin nhắn SMS.
* **Tính dễ sử dụng**: Hệ thống có dashboard đơn giản, không yêu cầu kỹ thuật cao để vận hành.
* **Hỗ trợ nhiều nền tảng**: Có thể truy cập từ trình duyệt web, điện thoại thông minh hoặc màn hình cảm ứng công nghiệp.

### **4.5 SOUP tích hợp**

Hệ thống sử dụng một số thành phần phần mềm SOUP để hỗ trợ vận hành.

#### **Lý do sử dụng SOUP:**

* **Tiết kiệm thời gian phát triển**: Tận dụng các thư viện mã nguồn mở và hệ điều hành nhúng.
* **Được cộng đồng hỗ trợ**: Cập nhật liên tục và dễ dàng tìm tài liệu hướng dẫn.
* **Khả năng mở rộng**: Tích hợp linh hoạt với các công nghệ khác.

#### **Kiểm soát rủi ro khi sử dụng SOUP:**

* **Xác thực nguồn gốc**: Chỉ sử dụng SOUP từ các nhà phát triển uy tín như Google, Amazon, Espressif.
* **Cập nhật định kỳ**: Theo dõi các bản vá bảo mật và cập nhật khi cần.
* **Kiểm thử trước khi triển khai**: Đánh giá mức độ tương thích với hệ thống.

# **Khả năng truy xuất yêu cầu**

|  |  |  |
| --- | --- | --- |
| Requirement | Component | Comment |
| REQ-001: Hệ thống phải đo và ghi nhận nhiệt độ, độ ẩm | COMPO-001: Cảm biến DHT22/SHT31 | Cảm biến thu thập dữ liệu nhiệt độ, độ ẩm. |
| REQ-002: Hệ thống phải truyền dữ liệu đo được lên máy chủ | COMPO-002: Vi điều khiển ESP32/Raspberry Pi | Vi điều khiển đọc dữ liệu từ cảm biến và gửi lên server. |
| REQ-003: Hệ thống phải hiển thị dữ liệu theo thời gian thực | COMPO-003: Giao diện web/mobile | Người dùng có thể theo dõi thông tin từ ứng dụng. |
| REQ-004: Hệ thống phải lưu trữ lịch sử đo đạc | COMPO-004: Cơ sở dữ liệu Firebase/MongoDB | Lưu trữ dữ liệu để xem lại và phân tích. |