**Mục Lục**

[1 Giới thiệu](#_Toc190896175)

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

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

[1.2.1 Từ viết tắt](#_Toc190896178)

[1.2.2 Thuật ngữ](#_Toc190896179)

[1.3 Tài liệu tham khảo](#_Toc190896180)

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

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

[1.4 Quy ước](#_Toc190896183)

[2 Kiến trúc](#_Toc190896184)

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

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

[2.2.1 Mô tả thành phần phần cứng](#_Toc190896191)

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

[2.3.1 Mô tả thành phần phần mềm](#_Toc190896187)

[2.4 Phần mềm SOUP](#_Toc190896194)

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

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

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

# Giới thiệu

## Tổng quan về tài liệu

Tài liệu này mô tả kiến trúc của hệ thống hệ thống 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

| **Viết tắt** | **Diễn giải** |
| --- | --- |
| SRS | Software Requirements Specification (Tài liệu Đặc tả Yêu cầu Phần mềm) |
| UI | User Interface (Giao diện Người dùng) |
| DB | Database (Cơ sở dữ liệu) |
| API | Application Programming Interface (Giao diện lập trình ứng dụng) |
| SOUP | Software of Unknown Provenance (Phần mềm không rõ nguồn gốc) |

### Thuật ngữ

| **Thuật ngữ** | **Định nghĩa** |
| --- | --- |
| Phân tích hồi quy | Phương pháp thống kê dùng để dự đoán giá trị đầu ra dựa trên dữ liệu đầu vào. |
| MicroPython | Phiên bản thu gọn của Python dành cho các vi điều khiển. |
| UART | Giao tiếp nối tiếp không đồng bộ dùng để truyền dữ liệu giữa vi điều khiển và thiết bị khác. |
| Giao diện người dùng (UI) | Thành phần phần mềm cho phép người dùng tương tác với hệ thống. |

## **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

1. Quy ước về sơ đồ

- Sơ đồ kiến trúc logic: được biểu diễn bằng các khối (block) tương ứng với các thành phần phần mềm chính và các mối liên hệ giữa chúng bằng các mũi tên biểu thị dòng dữ liệu hoặc luồng điều khiển.

- Sơ đồ kiến trúc vật lý: sử dụng ký hiệu biểu thị phần cứng (như vi điều khiển, cảm biến, thiết bị đầu cuối) và mối liên kết vật lý (UART, GPIO, I2C, v.v.).

- Sơ đồ quy trình: tuân thủ chuẩn UML (Unified Modeling Language), sử dụng sơ đồ hoạt động (activity diagram) hoặc sơ đồ trình tự (sequence diagram) khi cần thiết để minh họa hành vi động.

1. Quy ước thuật ngữ

- COTS (Commercial Off-The-Shelf), OTSS (Off-The-Shelf Software), và SOUP (Software Of Unknown Provenance) đều đề cập đến cùng một khái niệm: phần mềm do bên thứ ba cung cấp, không được phát triển theo quy trình phần mềm kiểm soát và không có đầy đủ thông tin xác minh tuân thủ tiêu chuẩn hoặc quy định.

- MCU: Vi điều khiển (Microcontroller Unit), đơn vị xử lý chính trong hệ thống nhúng.

- UART: Giao tiếp nối tiếp không đồng bộ giữa các thiết bị phần cứng.

1. Quy ước định dạng văn bản

- Các từ in đậm dùng để nhấn mạnh các thành phần hoặc thuật ngữ quan trọng.

- Các mã lệnh hoặc tên thành phần phần mềm/hardware được đặt trong dấu monospace để dễ nhận diện.

# Kiến trúc

## Tổng quan về kiến trúc

Hệ thống dự báo mưa là một giải pháp tích hợp phần cứng và phần mềm được thiết kế để giám sát môi trường và dự đoán thời tiết cục bộ dựa trên dữ liệu cảm biến thực tế. Hệ thống hướng tới việc cung cấp một nền tảng nhẹ, chi phí thấp nhưng hiệu quả, có thể được triển khai linh hoạt tại các khu vực nông thôn, vùng xa hoặc trong các môi trường nghiên cứu học thuật.

* Môi trường hoạt động:

Hệ thống hoạt động trong các môi trường ngoài trời hoặc bán ngoài trời, như:

- Nông trại hoặc vườn nhà (home garden, greenhouse)

- Trạm giám sát khí tượng cục bộ

- Môi trường học tập, nghiên cứu (phòng thí nghiệm)

- Khu vực dân cư có mạng Internet hạn chế

* Người dùng:

- Nông dân, hộ gia đình, cần dự đoán thời tiết để phục vụ sản xuất hoặc sinh hoạt

- Nhà nghiên cứu, học sinh, sinh viên, sử dụng dữ liệu để nghiên cứu khí hậu, mô hình dự báo

- Quản trị viên hệ thống, giám sát và bảo trì thiết bị

* Mục đích sử dụng:

Hệ thống thu thập dữ liệu thời tiết từ môi trường như:

- Nhiệt độ không khí

- Độ ẩm không khí

- Độ ẩm đất

- Tình trạng mưa (qua cảm biến hoặc suy luận mô hình)

Dữ liệu sau đó được xử lý và phân tích để dự đoán khả năng mưa trong tương lai gần, giúp người dùng đưa ra các quyết định chính xác hơn trong trồng trọt hoặc sinh hoạt.

Các chức năng chính:

- Thu thập dữ liệu từ cảm biến (Bme280 nhiệt độ độ ẩm áp suất)

- Truyền dữ liệu về bộ xử lý trung tâm

- Phân tích dữ liệu bằng mô hình hồi quy

- Hiển thị kết quả dự báo qua màn hình LCD, website hoặc thông báo tin nhắn (nếu có kết nối)

- Lưu trữ dữ liệu lịch sử để phục vụ phân tích xu hướng

Giao diện, đầu vào và đầu ra chính:

- Đầu vào vật lý: Dữ liệu từ cảm biến Bme280 nhiệt độ độ ẩm áp suất

- Đầu vào phần mềm: Dữ liệu đầu vào dạng, có thể được hiệu chỉnh và tiền xử lý

- Đầu ra vật lý: Hiển thị trạng thái dự báo lên LCD; có thể mở rộng sang loa, đèn cảnh báo hoặc gửi dữ liệu đến máy chủ

- Giao diện phần mềm: Giao diện người dùng trên web

## Tổng quan về kiến trúc vật lý

Hệ thống dự báo lương mưa bao gồm một tập hợp các thành phần phần cứng đơn giản nhưng tích hợp chặt chẽ, được triển khai tại hiện trường để thu thập và xử lý dữ liệu môi trường theo thời gian thực. Các thành phần chính bao gồm:

***2.2.1 Thành phần phần cứng 1 – Vi điều khiển trung tâm (ESP32)***

Vai trò: Là bộ xử lý chính của hệ thống, điều khiển quá trình thu thập, xử lý dữ liệu và hiển thị hoặc truyền kết quả.

Kết nối:

- UART đến module cảm biến

- GPIO đến cảm biến và màn hình hiển thị

- WiFi (nếu có) để gửi dữ liệu lên đám mây

- Nguồn điện: 5V DC hoặc pin lithium

***2.2.2 Thành phần phần cứng 2 – Cảm biến môi trường***

- Bme280: nhiệt độ, độ ẩm và áp suất không khí

với độ chính xác cao, phù hợp cho các ứng dụng về thời tiết, IoT, thiết bị đeo tay, và nhà thông minh.

***2.2.3 Thành phần phần cứng 3 – Thiết bị đầu ra và truyền thông***

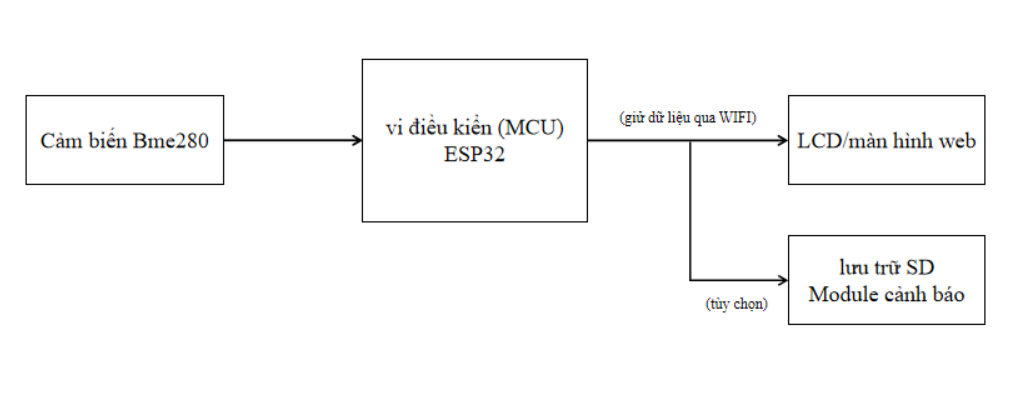
- LCD I2C 16x2: hiển thị dữ liệu thời tiết, kết quả dự đoán

- Module WiFi/Bluetooth: truyền dữ liệu đến server hoặc thiết bị di động

- Thẻ nhớ SD hoặc EEPROM (tuỳ chọn): lưu trữ dữ liệu cục bộ

- Nguồn dự phòng (pin lithium hoặc nguồn mặt trời): đảm bảo hoạt động ổn định ở vùng xa

Sơ đồ tổng quan kiến trúc vật lý:



## Tổng quan về kiến trúc logic

Hệ thống được triển khai theo kiến trúc phần mềm phân lớp, trong đó các thành phần phần mềm cấp cao được phân tách rõ ràng theo chức năng: thu thập dữ liệu cảm biến, xử lý dữ liệu & dự đoán, và hiển thị/thông báo kết quả.

Phần mềm chạy trên hệ điều hành **MicroPython** hoặc **FreeRTOS (trên ESP32)** hoặc một môi trường hệ điều hành nhẹ tương thích với IoT.

### Cấu trúc tổng thể gồm 3 thành phần chính:

**- Sensor Interface Module**

**- Processing & Forecasting Module**

**- Display & Communication Module**

## **2.3.1 **Sensor Interface Module (Mô-đun giao diện cảm biến)****

**Mục đích**: Thu thập dữ liệu từ cảm biến môi trường (Bme280 đo nhiệt độ, độ ẩm và áp suất không khi) theo chu kỳ định sẵn.

**Giao diện với thành phần khác**:

- Xuất dữ liệu thô (nhiệt độ, độ ẩm, áp xuất, trạng thái mưa)

**Giao diện mạng**: Không có

**Tài nguyên phần cứng sử dụng**:

- RAM: trung bình 10KB; đỉnh 20KB (tuỳ số lượng cảm biến)

- CPU: sử dụng trung bình 5%; đỉnh 10% trong chu kỳ quét

- GPIO: kết nối đến chân digital/analog

- Bộ đệm dữ liệu: ≤5KB

## **2.3.2 **Processing & Forecasting Module (Mô-đun xử lý và dự báo)****

**Mục đích**: Tiền xử lý dữ liệu, thực hiện thuật toán hồi quy/tổng hợp dữ liệu, và tạo ra kết quả dự báo (ví dụ: trời mưa hoặc không).

**Giao diện với thành phần khác**:

- Nhận dữ liệu từ Sensor Interface Module

- Xuất kết quả dự đoán cho Display Module

**Giao diện mạng**: (tuỳ chọn) Kết nối API để gửi dữ liệu lên đám mây hoặc mobile app

**Tài nguyên phần cứng sử dụng**:

- RAM: trung bình 20KB; đỉnh 40KB

- CPU: trung bình 10–15%; đỉnh 30% khi thực hiện tính toán hồi quy

- Bộ đệm dữ liệu: ≤10KB

- Không sử dụng thiết bị ngoại vi

## 2.3.3 ****Display & Communication Module(Mô-đun hiển thị và truyền thông)****

**Mục đích**: Hiển thị dữ liệu môi trường & kết quả dự báo lên LCD hoặc gửi dữ liệu qua WiFi/Bluetooth đến thiết bị khác.

**Giao diện với thành phần khác**:

- Nhận đầu vào từ Processing Module

- Tuỳ chọn gửi dữ liệu ra ngoài (server/app)

**Giao diện mạng:**

- Giao thức HTTP/MQTT để gửi dữ liệu (nếu có)

**Tài nguyên phần cứng sử dụng**:

- RAM: trung bình 15KB; đỉnh 25KB

- CPU: trung bình 5–10%; đỉnh 15% khi ghi màn hình hoặc gửi dữ liệu

- GPIO: I2C hoặc SPI đến màn hình LCD

- Module truyền thông: WiFi (ESP32) hoặc UART Bluetooth

## Phần mềm SOUP

Trong hệ thống theo dõi nhiệt độ, độ ẩm và cảnh báo, có một số phần mềm SOUP Hệ thống sử dụng một số thành phần phần mềm bên ngoài (SOUP – Software of Unknown Provenance). Các thành phần này có thể là thư viện mã nguồn mở, phần mềm của nhà cung cấp, hoặc mô-đun nhúng sẵn.

### **MicroPython**

**- Nhận dạng và phiên bản**: MicroPython v1.21.0

**- Mục đích**: Cung cấp môi trường runtime cho toàn bộ mã nguồn Python chạy trên ESP32.

**- Nguồn gốc**: Dự án mã nguồn mở do Damien George và cộng đồng phát triển.

**- Bên thứ ba bảo trì**: Có – cộng đồng duy trì trên GitHub.

**- Tệp thực thi**: Có, firmware nhúng.

**- Tài nguyên phần cứng sử dụng**:

RAM: khoảng 32–64 KB tùy chương trình

Flash: khoảng 256–512 KB

**- Cách nhiệt**: Không hoàn toàn cách ly, vì nó là môi trường thực thi chính.

Tuy nhiên, phần chức năng dự báo được phân lớp logic riêng biệt để cách ly với lỗi runtime.

**- Giao diện và luồng dữ liệu**: Mã Python giao tiếp trực tiếp với firmware MicroPython thông qua các API có sẵn.

**- Phần mềm sử dụng các hàm**:

machine, network, time, ujson, uos, ure, v.v.

**- Cách tích hợp**:

Flash firmware MicroPython vào ESP32, sau đó tải script Python lên.

**- Tài nguyên phần cứng/phần mềm cần thiết**:

Bộ vi điều khiển hỗ trợ MicroPython (ESP32)

Driver UART/USB để nạp chương trình

1. Thư viện Scikit-learn (trong giai đoạn huấn luyện mô hình)

- Nhận dạng và phiên bản: Scikit-learn v1.4.2

- Mục đích: Huấn luyện mô hình hồi quy để dự báo thời tiết (giai đoạn phát triển, không chạy trên thiết bị nhúng).

- Nguồn gốc: Thư viện mã nguồn mở

- Bên thứ ba bảo trì: Có – cộng đồng Python.

- Tệp thực thi: Không – chỉ dùng trong giai đoạn phát triển/offline.

- Cách nhiệt: Không chạy trong thiết bị thật, nên không cần cách ly.

- Giao diện và luồng dữ liệu:

Nhận dữ liệu huấn luyện (CSV)

Trả về mô hình đã huấn luyện (được chuyển thành dạng nhẹ hoặc rule-based nhúng lên ESP32)

- Tích hợp:

Chỉ sử dụng trong Python notebook hoặc IDE để huấn luyện

Tài nguyên yêu cầu: Máy tính phát triển (RAM ≥ 4GB, Python ≥ 3.9)

# Hành vi động của kiến trúc

Kiến trúc của hệ thống được thiết kế để đảm bảo hoạt động ổn định và đáp ứng các yêu cầu chức năng quan trọng, bao gồm: thu thập dữ liệu môi trường từ cảm biến, xử lý và phân tích dữ liệu, dự báo thời tiết, hiển thị dữ liệu và gửi cảnh báo.

**3.1 Quy trình làm việc / Trình tự 1: Thu thập dữ liệu môi trường**

Mô tả trình tự:

- Bộ định thời nội bộ (Timer) trong vi điều khiển kích hoạt sự kiện theo chu kỳ (ví dụ mỗi 5 phút).

- Vi điều khiển gửi tín hiệu kích hoạt cảm biến Bme280.

- Cảm biến phản hồi lại các giá trị môi trường (nhiệt độ, độ ẩm,áp xuất)- Vi điều khiển đọc dữ liệu từ các cảm biến thông qua GPIO/UART.

- Dữ liệu được xử lý thô và chuẩn hóa.

- Dữ liệu được ghi tạm vào bộ nhớ (buffer nội bộ hoặc lưu tạm vào RAM).

- Vi điều khiển có thể:

- Gửi dữ liệu qua UART/Wi-Fi tới hệ thống chủ (nếu có).

**3.2 Quy trình làm việc / Trình tự 2: Dự báo thời tiết và gửi cảnh báo**

Mô tả trình tự:

Sau khi thu thập đủ dữ liệu vi điều khiển sử dụng mô hình đơn giản được triển khai bằng hồi quy tuyến tính để phân tích.

Dựa trên ngưỡng giá trị và thuật toán, hệ thống quyết định:

- Dự báo mưa

Nếu điều kiện nguy hiểm được phát hiện:

- Gửi cảnh báo bằng cách hiển thị lên web

- Kích hoạt tín hiệu cảnh báo (buzzer hoặc LED)

- (Tùy chọn) Gửi tin nhắn qua Wi-Fi

**4 Lý do chính đáng cho kiến trúc**

**4.1 Khả năng kiến trúc hệ thống**

**Kiến trúc phần cứng và phần mềm của hệ thống được thiết kế để đảm bảo hiệu quả hoạt động, độ tin cậy và khả năng mở rộng trong môi trường ứng dụng thực tế, đặc biệt là tại các khu vực nông nghiệp, hộ gia đình hoặc các trạm quan trắc môi trường nhỏ. Cụ thể:**

**Hiệu suất:**

**- Thời gian phản hồi: Hệ thống phản hồi nhanh (<1 giây) sau mỗi chu kỳ đọc dữ liệu cảm biến.**

**- Lưu trữ dữ liệu: Dữ liệu được lưu trữ cục bộ dưới dạng nhật ký, có thể truyền qua UART/Wi-Fi để lưu trữ từ xa.**

**- Tính di động: Thiết bị có kích thước nhỏ gọn, tiêu thụ điện năng thấp, có thể hoạt động với pin sạc hoặc năng lượng mặt trời, thích hợp cho môi trường không có lưới điện ổn định.**

**An toàn cho người dùng:**

### **- Thiết bị hoạt động ở điện áp thấp (3.3V hoặc 5V), đảm bảo an toàn điện.**

### **- Bao vỏ cách điện, chống nước, hạn chế rủi ro tiếp xúc trực tiếp với mạch điện.**

### **Bảo vệ chống sử dụng sai mục đích:**

### **- Giao diện người dùng đơn giản, chỉ có chức năng theo dõi và cảnh báo, hạn chế nguy cơ cấu hình sai hoặc thao tác nhầm.**

### **- Hệ thống có xác thực đơn giản nếu truyền dữ liệu không dây (có thể thêm lớp bảo mật như mã hóa dữ liệu UART/Wi-Fi).**

### **Bảo trì:**

### **- Bảo trì nóng: Cho phép cập nhật firmware qua UART/Wi-Fi mà không cần tháo rời phần cứng.**

### **- Bảo trì lạnh: Dễ tháo lắp các cảm biến để thay thế, làm sạch hoặc hiệu chuẩn.**

### **Khả năng thích ứng và linh hoạt:**

### **-Hệ thống thiết kế mô-đun: có thể thêm hoặc thay cảm biến dễ dàng.**

### **- Có thể tùy chỉnh mô hình phân tích/dự báo thông qua cập nhật phần mềm.**

### **Khả năng mở rộng và tính khả dụng:**

### **- Hỗ trợ kết nối mạng UART hoặc Wi-Fi để tích hợp vào các hệ thống IoT lớn hơn.**

### **- Có thể mở rộng thành nhiều nút cảm biến chạy song song trong một mạng thu thập dữ liệu.**

### **Sao lưu và khôi phục:**

### **- Dữ liệu được sao lưu tạm thời trong EEPROM hoặc bộ nhớ flash.**

### **- Cho phép khôi phục tự động sau khi mất điện hoặc sự cố.**

### **Bảo mật phần cứng/phần mềm:**

### **- Tự khởi động lại khi phát hiện treo hệ thống.**

### **- Bảo vệ phần mềm qua chữ ký hoặc checksum để tránh bị thay đổi firmware trái phép.**

### **- Có thể triển khai thuật toán phát hiện lỗi cảm biến hoặc lỗi truyền thông.**

### **Quản lý và giám sát:**

### **- Cung cấp nhật ký hoạt động (log) theo thời gian thực.**

### **- Hỗ trợ giao thức UART để quản lý từ xa.**

### **Quốc tế hóa:**

### **- Các thông báo cảnh báo trên màn hình có thể tùy biến ngôn ngữ (Anh, Việt…).**

### **- Dữ liệu đầu ra dùng định dạng chuẩn (JSON hoặc CSV) để dễ tích hợp vào hệ thống quốc tế.**

### ****4.2 Khả năng kiến trúc mạng****

Hệ thống hỗ trợ kết nối mạng thông qua giao tiếp UART (nối tiếp) hoặc Wi-Fi để truyền dữ liệu cảm biến đến máy chủ hoặc hệ thống lưu trữ từ xa. Kiến trúc mạng được thiết kế để đảm bảo tính ổn định, an toàn và khôi phục trong trường hợp lỗi mạng.

Băng thông:

- Dữ liệu truyền tải định kỳ, dung lượng thấp (vài trăm byte mỗi lần).

- Tổng băng thông yêu cầu dưới 10KB/s, đủ nhỏ để hoạt động ổn định trên các kết nối Wi-Fi hoặc UART với tốc độ từ 9600–115200 bps.

- Không cần truyền tải video hoặc dữ liệu thời gian thực lớn, nên không yêu cầu băng thông cao.

Lỗi mạng:

- Hệ thống sử dụng cơ chế kiểm tra và xác nhận (ACK) khi gửi dữ liệu để phát hiện lỗi.

- Trong trường hợp mất kết nối, dữ liệu sẽ được lưu trữ tạm trong bộ nhớ cục bộ và tự động truyền lại khi kết nối khôi phục.

Mất dữ liệu:

- Dữ liệu quan trọng (nhiệt độ, độ ẩm,áp xuất, trạng thái mưa) được lưu đệm trong flash trước khi truyền.

- Hệ thống hỗ trợ cơ chế retry với giới hạn số lần gửi lại và nhật ký lỗi.

- Nếu vượt quá số lần gửi lại, hệ thống ghi nhận lỗi và tiếp tục chu kỳ mới mà không làm treo hệ thống.

Dữ liệu không nhất quán:

- Dữ liệu từ các cảm biến được đánh dấu thời gian (timestamp) tại thời điểm thu thập.

- Đảm bảo dữ liệu truyền đi phản ánh đúng giá trị đo tại thời điểm tương ứng, tránh hiện tượng trễ gây nhầm lẫn.

- Mô hình phân tích có thể kiểm tra sự bất thường hoặc mâu thuẫn dữ liệu để lọc bỏ.

### ****4.3 Đầu ra phân tích rủi ro****

Phân tích rủi ro được thực hiện theo quy trình quản lý rủi ro dựa trên tiêu chuẩn ISO 14971 để xác định và kiểm soát các mối nguy có thể ảnh hưởng đến độ chính xác, an toàn và khả năng hoạt động liên tục của hệ thống .

Dưới đây là các đầu ra chính từ quá trình phân tích rủi ro và các biện pháp giảm thiểu đã được thực hiện trong kiến trúc hệ thống:

Rủi ro 1: Dữ liệu cảm biến sai lệch dẫn đến dự đoán sai

Nguy cơ: Độ chính xác kém từ cảm biến Bme280 do hư hỏng, sai số hoặc lão hóa.

Biện pháp giảm thiểu kiến trúc:

- Tích hợp thuật toán lọc dữ liệu (trung bình trượt, kiểm tra giới hạn hợp lý).

- Thiết kế phần mềm có khả năng so sánh chéo dữ liệu từ nhiều mẫu liên tiếp để phát hiện bất thường.

- Ghi chú trạng thái cảm biến (ví dụ: mất tín hiệu, giá trị "NaN") trong dữ liệu đầu ra.

Rủi ro 2: Mất kết nối mạng gây mất dữ liệu

Nguy cơ: Không thể truyền dữ liệu thời gian thực đến máy chủ → mất dữ liệu quan trọng.

Biện pháp giảm thiểu kiến trúc:

- Thiết kế bộ nhớ đệm tạm thời (SD Card hoặc flash nội).

- Cơ chế retry và xác nhận (ACK) khi gửi dữ liệu.

- Cho phép cấu hình mức ưu tiên cho dữ liệu thời gian thực hoặc dữ liệu lưu trữ.

Rủi ro 3: Mất nguồn gây hỏng hệ thống hoặc mất dữ liệu đang xử lý

Nguy cơ: Tắt nguồn đột ngột có thể làm hỏng firmware hoặc gây mất dữ liệu đo chưa được truyền đi.

Biện pháp giảm thiểu kiến trúc:

- Sử dụng tụ điện siêu nhỏ (supercapacitor) hoặc pin phụ trợ để cấp điện duy trì trong vài giây sau khi mất nguồn để lưu dữ liệu tạm thời.

- Tích hợp cơ chế ghi an toàn (safe write) vào flash để đảm bảo không ghi sai lệch dữ liệu.

Rủi ro 4: Truy cập trái phép hoặc giả mạo dữ liệu

Nguy cơ: Người dùng không hợp lệ truy cập hệ thống hoặc thay đổi dữ liệu đầu ra.

Biện pháp giảm thiểu kiến trúc:

- Mã hóa đơn giản khi truyền dữ liệu (nếu sử dụng Wi-Fi).

- Xác thực thiết bị dựa trên ID cố định hoặc mã hóa khóa ban đầu.

- Hạn chế giao tiếp hai chiều, chỉ cho phép thiết bị gửi dữ liệu một chiều (write-only).

Rủi ro 5: Dự đoán sai dẫn đến hành động sai lầm

Nguy cơ: Mô hình học máy/thuật toán hồi quy dự đoán sai → người dùng đưa ra quyết định không chính xác (ví dụ: tưới khi trời sắp mưa).

Biện pháp giảm thiểu kiến trúc:

- Mô hình được huấn luyện từ dữ liệu thực tế với kiểm định độ chính xác.

- Cung cấp chỉ số độ tin cậy (confidence score) cho mỗi dự đoán.

- Cho phép người dùng kiểm tra lịch sử dữ liệu đầu vào cùng với dự đoán để xác minh.

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

Phân tích yếu tố con người (Human Factors Engineering - HFE) đã được tiến hành để xác định các tình huống mà người dùng cuối có thể gây ra lỗi do hiểu nhầm, thao tác sai hoặc thiết kế giao diện chưa phù hợp. Dưới đây là các rủi ro đã xác định và cách kiến trúc được điều chỉnh để giảm thiểu chúng:

Rủi ro 1: Người dùng không hiểu rõ thông tin dự báo thời tiết

Vấn đề: Người dùng (nông dân, học sinh...) có thể hiểu sai thông tin đầu ra như: xác suất mưa, nhiệt độ cao nhất/thấp nhất, hoặc chỉ số độ tin cậy.

Giải pháp kiến trúc:

- Thiết kế giao diện người dùng trực quan với biểu tượng dễ hiểu

- Sử dụng thang màu trực quan và biểu đồ đơn giản.

- Hiển thị thêm thông tin giải thích ngắn gọn kèm các kết quả phức tạp.

Rủi ro 2: Người dùng thao tác sai trên thiết bị (bấm nhầm, cắm sai nguồn, reset nhầm)

Vấn đề: Thiết bị nhỏ, cầm tay có thể bị thao tác sai, gây khởi động lại hoặc mất dữ liệu.

Giải pháp kiến trúc:

- Giới hạn số lượng nút vật lý xuống mức tối thiểu (ví dụ: chỉ 1 nút bấm).

- Tích hợp chống rung nút nhấn (debounce) trong phần mềm.

- Giao diện yêu cầu xác nhận hai bước trước hành động quan trọng (xóa dữ liệu, cập nhật...).

Rủi ro 3: Người dùng cắm sai cảm biến

Vấn đề: Người dùng có thể cắm nhầm các cổng kết nối của cảm biến vào vi điều khiển.

Giải pháp kiến trúc:

- Sử dụng cổng có thiết kế khóa vật lý (keyed connectors) hoặc đánh dấu màu rõ ràng.

- Thiết kế phần mềm kiểm tra loại cảm biến và cảnh báo nếu dữ liệu bất thường (ví dụ: nhiệt độ -100°C).

Rủi ro 4: Người dùng không bảo trì thiết bị đúng cách

Vấn đề: Người dùng quên kiểm tra pin, vệ sinh cảm biến → sai lệch dữ liệu.

Giải pháp kiến trúc:

- Thêm cảnh báo pin yếu và nhắc nhở kiểm tra định kỳ qua màn hình hoặc app.

- Tích hợp log thời gian vận hành cảm biến để theo dõi tuổi thọ.

Rủi ro 5: Người dùng không biết cách phản hồi khi thiết bị bị lỗi

Vấn đề: Khi thiết bị treo hoặc báo lỗi, người dùng không biết phải làm gì → gây gián đoạn sử dụng.

Giải pháp kiến trúc:

- Thêm chế độ khôi phục mặc định (reset safe mode) đơn giản bằng tổ hợp nút.

- Giao diện cung cấp mã lỗi dễ hiểu (hoặc biểu tượng kèm hướng dẫn khắc phục).

### **4.5 SOUP tích hợp**

Trong kiến trúc phần mềm của hệ thống, một số thành phần phần mềm của bên thứ ba (SOUP – Software of Unknown Provenance) đã được tích hợp để phục vụ cho việc thu thập, xử lý và hiển thị dữ liệu môi trường. Để đảm bảo tính an toàn, độ tin cậy và dễ bảo trì, quá trình tích hợp SOUP được thực hiện theo kiến trúc cô lập và kiểm soát, bao gồm:

| SOUP | Phương pháp tích hợp | Cấu trúc kiến trúc |
| --- | --- | --- |
| MicroPython | Cài đặt trực tiếp trên bo mạch ESP32 | Là môi trường chạy chính cho logic điều khiển cảm biến và xử lý ban đầu |
| matplotlib | Được bọc bởi lớp PlotWrapper trong phần mềm phân tích | Wrapper cách ly tính năng trực quan hóa với các luồng dữ liệu nội bộ |
| TensorFlow Lite | Tích hợp như một module riêng biệt qua socket giao tiếp | Giao tiếp với chương trình chính qua IPC socket trên nền tảng Linux |
| SQLite3 | Sử dụng trực tiếp qua API tiêu chuẩn | Truy cập dữ liệu thông qua lớp DatabaseManager để cách ly lỗi cơ sở dữ liệu |

* Kiến trúc tích hợp cách ly SOUP

Đối với các thư viện có khả năng không ổn định hoặc phức tạp như TensorFlow Lite, việc tích hợp được thực hiện theo mô hình client-server:

- Quy trình chính (main logic) gửi dữ liệu đến một quy trình TensorFlow riêng thông qua giao tiếp Unix socket.

- Dữ liệu đầu ra được kiểm tra tính hợp lệ trước khi đưa vào hệ thống.

- Điều này giúp cô lập lỗi hoặc crash từ TensorFlow không ảnh hưởng đến toàn bộ hệ thống.

Đối với matplotlib, để tránh tình trạng lỗi hiển thị ảnh hưởng đến luồng xử lý chính:

-Một lớp PlotWrapper được tạo, chỉ gọi matplotlib như một dịch vụ xuất biểu đồ tạm thời.

-Các lỗi từ matplotlib được ghi nhận log nhưng không làm gián đoạn quá trình xử lý dữ liệu.

* Kiểm soát tài nguyên và rủi ro

- Các SOUP đều được giới hạn về tài nguyên sử dụng (RAM, CPU) thông qua cấu hình hệ thống hoặc điều kiện runtime:

- Các lỗi đã biết từ SOUP (nếu có) được ghi nhận và theo dõi trong hệ thống quản lý lỗi nội bộ.

* Mục tiêu của kiến trúc tích hợp SOUP

- Tính cách ly: Các thành phần SOUP không ảnh hưởng đến sự ổn định chung của hệ thống nếu có lỗi.

- Dễ kiểm tra: Giao diện rõ ràng giúp thực hiện các test tự động cho từng module SOUP.

- Dễ bảo trì/nâng cấp: Có thể cập nhật SOUP mà không ảnh hưởng đến phần còn lại của hệ thống.

# **Khả năng truy xuất yêu cầu**

| Yêu cầu | Thành phần | Bình luận |
| --- | --- | --- |
| YÊU CẦU-001: Hệ thống phải thu thập dữ liệu môi trường từ các cảm biến (nhiệt độ, độ ẩm, độ ẩm đất, cảm ứng mưa) | HW-001: ESP32, SW-001: SensorManager | ESP32 nhận dữ liệu từ Bme280 quản lý chu trình thu thập và kiểm tra dữ liệu. |
| YÊU CẦU-002: Hệ thống phải lưu trữ dữ liệu thu thập được vào bộ nhớ cục bộ | SW-002: DatabaseManager, SOUP: SQLite3 | Dữ liệu được lưu dưới dạng bản ghi thông qua API SQLite. |
| YÊU CẦU-003: Hệ thống phải dự đoán khả năng mưa dựa trên dữ liệu hiện tại | SW-003: PredictionEngine, SOUP: TensorFlow Lite | PredictionEngine gọi TensorFlow Lite để xử lý và đưa ra dự đoán. |
| YÊU CẦU-004: Hệ thống phải hiển thị biểu đồ nhiệt độ và độ ẩm hàng ngày | SW-004: UI Module, SOUP: matplotlib, PlotWrapper | Giao diện người dùng hiển thị biểu đồ dựa trên dữ liệu được lấy từ SQLite và dựng bằng matplotlib. |
| YÊU CẦU-005: Hệ thống phải cho phép truy vấn lịch sử dữ liệu theo ngày | SW-002: DatabaseManager, SW-004: UI Module | Giao diện truy vấn được cung cấp qua UI, xử lý bởi DatabaseManager. |
| YÊU CẦU-006: Hệ thống phải gửi cảnh báo nếu có nguy cơ mưa cao | SW-005: AlertManager, SW-003: PredictionEngine | Cảnh báo sinh ra nếu xác suất mưa vượt ngưỡng cho phép. |
| YÊU CẦU-007: Hệ thống phải hoạt động độc lập không cần internet | HW-001: ESP32, SW-001 đến SW-005 | Tất cả xử lý và lưu trữ đều cục bộ, không yêu cầu mạng. |
| YÊU CẦU-008: Dữ liệu phải được sao lưu mỗi ngày vào bộ nhớ ngoài (nếu gắn USB/SD) | SW-006: BackupManager | Kiểm tra thiết bị lưu trữ và sao lưu cơ sở dữ liệu định kỳ. |