

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG ĐIỆN - ĐIỆN TỬ



BÁO CÁO
ĐỒ ÁN 1

Đề tài:

MODULE QUẢN LÝ CẢM BIẾN ĐA NĂNG

Sinh viên thực hiện: **HỒ XUÂN PHÚ - 20250179E**

Giảng viên hướng dẫn: **TS. HÀN HUY DŨNG**

Hà Nội, Ngày 7 tháng 2 năm 2026

LỜI CẢM ƠN

Lời nói đầu tiên, em xin gửi lời cảm ơn chân thành nhất tới TS. Hàn Huy Dũng – người thầy đã hướng dẫn và đồng hành cùng em trong suốt thời gian học tập Đại học và hoàn thành Đồ án 1 chương trình kỹ sư chuyên sâu. Những đóng góp, chia sẻ của thầy đã cho em nhiều bài học quý giá, em sẽ luôn ghi nhớ và biết ơn. Tiếp theo, em xin gửi lời cảm ơn đến tập thể các thành viên của SPARC Laboratory, những người đã đồng hành cùng em trải qua nhiều dự án, cùng học tập, phát triển và đoàn kết như những người anh em trong một gia đình.

Đối với em, quãng thời gian là sinh viên Đại học Bách Khoa Hà Nội, được học tập và rèn luyện tại trường Điện – Điện tử luôn là một dấu ấn và kỉ niệm quý giá. Hình ảnh những “người thầy” nhiệt huyết giảng dạy, luôn tận tâm với nghề, luôn đối xử tử tế với những người xung quanh và sự cố gắng không ngừng nghỉ đã cho em nhiều động lực và ngày càng nỗ lực hơn trong cuộc sống. Em xin gửi lời cảm ơn đến các thầy, cô giảng viên trường Điện – Điện tử nói chung, và các thầy cô của Khoa Điện tử và Khoa Kỹ thuật truyền thông nói riêng, đã tận tình giảng dạy, chỉ bảo và cho em những lời khuyên về học tập và cuộc sống.

Em xin chân thành cảm ơn!

Mục lục

DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT	1
DANH MỤC HÌNH ẢNH	2
DANH MỤC BẢNG BIỂU	3
TÓM TẮT DỰ ÁN	4
CHƯƠNG 1. CHƯƠNG MỞ ĐẦU	5
1.1 Giới thiệu vấn đề và giải pháp	5
1.2 Mục tiêu và phạm vi của dự án	6
1.3 Tóm tắt cấu trúc dự án	7
CHƯƠNG 2. PHÂN TÍCH YÊU CẦU KỸ THUẬT	8
2.1 Các trường dữ liệu và loại cảm biến	8
2.2 Yêu cầu chức năng	9
2.3 Yêu cầu phi chức năng	10
CHƯƠNG 3. THIẾT KẾ CHI TIẾT HỆ THỐNG	11
3.1 Thiết kế tổng thể hệ thống	11
3.1.1 Tổng quan và sơ đồ hệ thống	11
3.1.2 Linh kiện sử dụng	13
A) Vi điều khiển ESP32	13
B) Module RTC DS3231	14
C) Module lưu trữ SD Card	14
D) Module mạch sạc IP5306	15
E) Pin Lithium Polymer	17
F) Module màn hình OLED SSD1306	19
G) IC ổn áp AMS1117-3.3	20

3.2	Thiết kế chi tiết hệ thống	21
3.2.1	Tính toán đáp ứng công suất nguồn	21
3.2.2	Thiết kế mạch nguyên lý (Schematic)	22
3.2.3	Layout PCB	26
3.2.4	Thiết kế Firmware	28
CHƯƠNG 4. KIỂM THỬ HỆ THỐNG VÀ LẤY DỮ LIỆU		34
4.1	Chiến lược kiểm thử	34
4.2	Kiểm thử phần cứng	34
4.2.1	Kiểm thử nguồn và tiêu thụ công suất	34
4.2.2	Kiểm thử nút bấm	35
4.2.3	Kiểm thử màn hình OLED SSD1306	35
4.2.4	Kiểm thử các port cảm biến	36
4.3	Kiểm thử firmware và chức năng	36
4.3.1	Kiểm thử MenuSystem và ScreenManager	36
4.3.2	Kiểm thử SensorRegistry và đọc dữ liệu cảm biến	37
4.3.3	Kiểm thử WifiManager và gửi dữ liệu	37
4.4	Tổng hợp kết quả kiểm thử	38
CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN		39
5.1	Kết quả đạt được	39
5.2	Hướng phát triển	39
TÀI LIỆU THAM KHẢO		41

DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT

Danh sách hình vẽ

Hình 1.1	Ví dụ các module cảm biến đơn chức năng trên thị trường.	5
Hình 3.1	Cấu trúc tổng quan của hệ thống.	11
Hình 3.2	Vi điều khiển ESP32.	13
Hình 3.3	Module DS3231.	14
Hình 3.4	Module SD Card.	15
Hình 3.5	Module mạch sạc IP5306.	16
Hình 3.6	Pin Lithium Polymer 3.7V 1500mAh (Model FL 803450).	18
Hình 3.7	Module màn hình OLED SSD1306.	19
Hình 3.8	IC ổn áp AMS1117-3.3.	20
Hình 3.9	Khối xử lý và nút bấm.	22
Hình 3.10	Khối nguồn.	23
Hình 3.11	Khối cảm biến tích hợp.	24
Hình 3.12	Khối 2 cổng cảm biến.	25
Hình 3.13	Top Layout.	26
Hình 3.14	Bot Layout.	27
Hình 3.15	3D Top.	27
Hình 3.16	3D Bot.	28
Hình 3.17	Sơ đồ kiến trúc phần mềm ESP32.	28
Hình 3.18	Web server cấu hình hệ thống.	30
Hình 3.19	Tổng quan luồng hoạt động hệ thống MSMS (Sequence Diagram).	31
Hình 3.20	Sơ đồ topology mạng Wi-Fi Mesh của các node MSMS.	32
Hình 4.1	Hình ảnh kiểm thử: giao diện menu và phần cứng module MSMS.	38

Danh sách bảng

Bảng 3.1	Thông số kỹ thuật ESP32	13
Bảng 3.2	Thông số kỹ thuật DS3231	14
Bảng 3.3	Thông số kỹ thuật thẻ nhớ SD	15
Bảng 3.4	Thông số kỹ thuật IP5306	16
Bảng 3.5	Thông số kỹ thuật Pin Lithium Polymer	18
Bảng 3.6	Thông số kỹ thuật SSD1306	19
Bảng 3.7	Thông số kỹ thuật AMS1117-3.3	20
Bảng 3.8	Bảng tính toán công suất nguồn ở kịch bản Peak	21
Bảng 4.1	Tóm tắt kết quả kiểm thử	38

TÓM TẮT ĐỒ ÁN

Trong bối cảnh các ứng dụng IoT ngày càng phổ biến, nhu cầu xây dựng những module cảm biến linh hoạt, có thể tái sử dụng cho nhiều bài toán khác nhau đang trở nên cấp thiết. Thay vì thiết kế từng hệ thống cảm biến cố định cho từng mục đích riêng lẻ, đề tài này tập trung vào việc phát triển một **module quản lý cảm biến đa năng** (Multi-Sensor Management System – MSMS) dựa trên vi điều khiển ESP32, cho phép người dùng tùy chọn loại cảm biến cần sử dụng và cấu hình trực tiếp trên thiết bị.

Module MRS được thiết kế như một khối phần cứng–firmware dùng chung: ESP32 đóng vai trò bộ xử lý trung tâm, kết nối với nhiều loại cảm biến khác nhau (cảm biến môi trường, cảm biến khí, ...). Thông qua màn hình OLED và 4 nút bấm điều hướng, người dùng có thể chọn cảm biến cho từng cổng kết nối, xem dữ liệu đo được theo từng trường (tên trường – giá trị – đơn vị) mà không cần can thiệp vào mã nguồn hay nạp lại firmware.

Hệ thống hỗ trợ quản lý tập trung danh sách cảm biến, cho phép thêm/bớt loại cảm biến mới một cách thuận tiện. Dữ liệu sau khi đo được có thể hiển thị trực tiếp trên thiết bị và gửi mẫu lên server qua Wi-Fi để phục vụ cho việc lưu trữ, phân tích hoặc tích hợp vào các hệ thống lớn hơn. Kiến trúc của module cũng được chuẩn bị để có thể mở rộng thành mạng nhiều node trong tương lai, nơi nhiều module MRS có thể thu thập dữ liệu tại nhiều vị trí khác nhau và tập trung về một điểm xử lý chung.

Kết quả đạt được là một module cảm biến có khả năng cấu hình linh hoạt, dễ mở rộng và phù hợp với nhiều ứng dụng khác nhau như giám sát môi trường, nông nghiệp thông minh hay các hệ thống cảm biến phân tán. Dự án đặt nền tảng cho các nghiên cứu tiếp theo về mạng nhiều node, kết nối cloud và các cơ chế xử lý – phân tích dữ liệu cảm biến ở quy mô lớn.

CHƯƠNG 1. CHƯƠNG MỞ ĐẦU

1.1 Giới thiệu vấn đề và giải pháp

Trong bối cảnh Internet of Things (IoT) phát triển mạnh mẽ, nhu cầu thu thập và giám sát dữ liệu từ nhiều loại cảm biến khác nhau ngày càng trở nên phổ biến trong các ứng dụng như giám sát môi trường, nông nghiệp thông minh, nhà thông minh hay công nghiệp 4.0. Tuy nhiên, việc triển khai các hệ thống cảm biến thường gặp nhiều thách thức:

- Mỗi loại cảm biến có giao thức giao tiếp, driver và cách xử lý dữ liệu riêng, gây khó khăn khi muốn mở rộng hay thay đổi loại cảm biến.
- Nhiều hệ thống được thiết kế cứng (hard-code) theo một tập cảm biến cố định, khiến việc tái sử dụng và tùy biến cho các bài toán khác nhau trở nên phức tạp.
- Khi số lượng điểm đo tăng lên, nhu cầu xây dựng một mạng lưới các node cảm biến để thu thập và tập hợp dữ liệu trở nên cấp thiết, kéo theo bài toán về truyền thông, đồng bộ và quản lý dữ liệu.
- Việc truyền dữ liệu lên server, lưu trữ tập trung và trực quan hóa cũng đòi hỏi một kiến trúc hệ thống đủ linh hoạt và ổn định.

Trên thị trường hiện nay đã xuất hiện nhiều thiết bị cảm biến thương mại, tuy nhiên đa số được thiết kế **chỉ để đọc một loại cảm biến cố định** cho một mục đích cụ thể (ví dụ: chỉ đo nhiệt độ, chỉ đo độ ồn, chỉ đo ánh sáng như các module trong hình minh họa). Khi người dùng muốn đo thêm đại lượng khác, họ buộc phải mua thêm một thiết bị khác hoặc thay đổi toàn bộ phần cứng, dẫn đến chi phí cao và khó linh hoạt trong việc xây dựng các hệ thống cảm biến đa mục đích.



Hình 1.1 Ví dụ các module cảm biến đơn chức năng trên thị trường.

Để giải quyết các vấn đề trên, đề tài tập trung phát triển một **module quản lý cảm biến đa năng** (Multi-Sensor Management System - MSMS) dựa trên vi điều khiển ESP32. Thay vì chỉ phục vụ cho một bài toán cảm biến cụ thể, module được thiết kế như một khối phần cứng–phần mềm dùng chung, cho phép:

- Người dùng tùy chọn loại cảm biến muốn sử dụng thông qua hệ thống menu trên màn hình OLED SSD1306.
- Hệ thống tự động ánh xạ, khởi tạo và đọc dữ liệu từ cảm biến đã chọn trên màn hình.
- Kết hợp nhiều node MSMS trong một mạng Wi-Fi (hướng tới Wi-Fi Mesh) để thu thập và tập hợp dữ liệu từ nhiều vị trí khác nhau.
- Gửi dữ liệu lên server hoặc backend để lưu trữ, phân tích và hiển thị trên giao diện web.

Module MSMS vì thế không chỉ là một trạm đo đơn lẻ mà là một **khối thiết bị cảm biến linh hoạt**, có thể tái cấu hình cho nhiều bài toán khác nhau (đo nhiệt độ–độ ẩm–áp suất, giám sát khí gas, theo dõi CO₂, v.v.), đồng thời sẵn sàng mở rộng thành một hệ thống nhiều node trong tương lai.

1.2 Mục tiêu và phạm vi của dự án

Mục tiêu của dự án là thiết kế và xây dựng một module cảm biến đa năng dựa trên ESP32, vừa đảm bảo tính linh hoạt trong việc lựa chọn cảm biến, vừa có khả năng mở rộng thành một mạng lưới thu thập dữ liệu và gửi lên server. Cụ thể:

- Thiết kế phần cứng module với ESP32 là vi điều khiển trung tâm, tích hợp màn hình OLED, hệ thống nút bấm, mạch đo pin và các cổng kết nối cho nhiều loại cảm biến khác nhau.
- Xây dựng firmware cho phép người dùng cấu hình cảm biến cho từng cổng thông qua giao diện menu trên màn hình, mà không cần sửa mã nguồn.
- Cung cấp cơ chế quản lý danh sách cảm biến tập trung, giúp việc thêm/bớt loại cảm biến mới trở nên thuận tiện.
- Hỗ trợ kết nối Wi-Fi để có thể gửi dữ liệu cảm biến mẫu lên server hoặc hệ thống thu thập dữ liệu bên ngoài.

Phạm vi của dự án tập trung vào:

- Thiết kế và hiện thực một module cảm biến với 2 cổng cảm biến chính, có thể gắn nhiều loại cảm biến khác nhau tùy mục đích sử dụng.
- Hiển thị dữ liệu đo được trực tiếp trên màn hình OLED, kèm theo một số thông tin trạng thái cơ bản của hệ thống.
- Thử nghiệm gửi dữ liệu mẫu lên server để kiểm chứng khả năng tích hợp với các hệ thống IoT khác.

1.3 Tóm tắt cấu trúc dự án

Dự án “Module quản lý cảm biến đa năng MSMS” gồm 5 chương:

- **Chương 1 – Chương mở đầu:** Trình bày bối cảnh, lý do chọn đề tài, mục tiêu, phạm vi và cấu trúc báo cáo.
- **Chương 2 – Phân tích yêu cầu kỹ thuật:** Xác định yêu cầu chức năng và phi chức năng của hệ thống, phân tích bài toán quản lý cảm biến và yêu cầu về thu thập – hiển thị – truyền dữ liệu.
- **Chương 3 – Thiết kế chi tiết hệ thống:** Mô tả thiết kế phần cứng và firmware của module, bao gồm sơ đồ khối tổng thể và các thành phần chính trên mạch.
- **Chương 4 – Kiểm thử hệ thống:** Trình bày các kịch bản kiểm thử phần cứng, firmware và quá trình thu thập – gửi dữ liệu.
- **Chương 5 – Kết luận và hướng phát triển:** Tổng kết kết quả đạt được và đề xuất các hướng phát triển tiếp theo cho hệ thống MSMS.

CHƯƠNG 2. PHÂN TÍCH YÊU CẦU KỸ THUẬT

2.1 Các trường dữ liệu và loại cảm biến

Để module MSMS có thể tái sử dụng cho nhiều bài toán khác nhau, trước hết cần xác định rõ **những nhóm trường dữ liệu chính** mà hệ thống phải hỗ trợ, tương ứng với các loại cảm biến được tích hợp. Trong phạm vi đề án, các nhóm dữ liệu chính được xem xét như sau:

- **Nhóm cảm biến môi trường:** Bao gồm các trường dữ liệu như nhiệt độ, độ ẩm, áp suất không khí, có thể được cung cấp bởi các cảm biến như BME280, DHT22 hoặc các cảm biến tương đương. Đây là nhóm dữ liệu phổ biến trong các ứng dụng giám sát môi trường, nhà thông minh và nông nghiệp thông minh.
- **Nhóm cảm biến chất lượng nước:** Gồm các trường dữ liệu như pH, độ dẫn điện (EC), tổng chất rắn hòa tan (TDS), độ đục, nhiệt độ nước, hàm lượng oxy hòa tan (DO), ... Nhóm này phù hợp cho các ứng dụng giám sát ao nuôi thủy sản, chất lượng nước sinh hoạt, nước thải công nghiệp hoặc các mô hình thí nghiệm trong phòng lab.
- **Nhóm cảm biến khí và chất lượng không khí:** Gồm các chỉ số như nồng độ CO₂, khí gas dễ cháy, khói, các khí độc; các giá trị này thường được thể hiện theo đơn vị ppm hoặc $\mu\text{g}/\text{m}^3$ và có thể mở rộng thêm tùy theo loại cảm biến được hỗ trợ.
- **Nhóm dữ liệu mở rộng:** Hệ thống MSMS được thiết kế để có thể bổ sung thêm các loại cảm biến mới (ví dụ cảm biến dòng điện, điện áp, rung động, ánh sáng, ...) mà không cần thay đổi kiến trúc tổng thể. Mỗi cảm biến mới chỉ cần khai báo thêm tập trường dữ liệu (tên trường, đơn vị, số lượng trường) tương ứng.

Ở mức trừu tượng, mỗi loại cảm biến trong hệ thống đều đi kèm với:

- Tập các trường dữ liệu mà nó cung cấp (ví dụ: “nhiệt độ”, “độ ẩm”, “pH”, ...).
- Đơn vị đo tương ứng cho từng trường (ví dụ: °C, %, ppm, NTU, ...).
- Các giới hạn đo và độ phân giải phù hợp với từng ứng dụng.

Với cách phân nhóm như trên, MSMS không bị ràng buộc vào một loại cảm biến cố định mà có thể mở rộng dần danh sách cảm biến hỗ trợ, miễn là mô tả được các trường dữ liệu và đơn vị đo tương ứng.

2.2 Yêu cầu chức năng

Từ mục tiêu đề tài đã trình bày ở Chương 1, các yêu cầu chức năng chính của module MSMS có thể được tóm tắt như sau:

- **Quản lý danh sách cảm biến:** Hệ thống cần có cơ chế tập trung để ghi nhận và tra cứu tất cả các loại cảm biến được hỗ trợ, kèm theo tên hiển thị, tập trường dữ liệu và đơn vị đo tương ứng.
- **Cấu hình cảm biến cho từng cổng kết nối:** Người dùng phải có thể chọn loại cảm biến gắn với từng cổng (ví dụ cổng 1, cổng 2, ...) thông qua giao diện menu trên màn hình. Việc thay đổi cấu hình cảm biến cần được thực hiện trực tiếp trên thiết bị, không yêu cầu lập trình lại.
- **Thu thập dữ liệu cảm biến:** Hệ thống phải có khả năng tự động khởi tạo và đọc dữ liệu cảm biến theo chu kỳ hoặc theo yêu cầu, lưu trữ tạm thời các giá trị đo được để phục vụ cho việc hiển thị và truyền dữ liệu.
- **Hiển thị dữ liệu và trạng thái hệ thống:** Màn hình OLED cần hiển thị được:
 - Menu điều hướng (danh sách cảm biến, danh sách cổng, các chức năng hệ thống).
 - Dữ liệu cảm biến dưới dạng từng trường: tên trường, giá trị và đơn vị.
 - Một số trạng thái cơ bản như tình trạng kết nối Wi-Fi, mức pin, ...
- **Điều hướng bằng 3 nút bấm:** Người dùng điều khiển toàn bộ hệ thống thông qua 3 nút: BACK, SELECT và MOVE. Hệ thống phải đọc nút có chống dội (debounce), phát hiện chính xác thời điểm nhấn và ánh xạ sang các hành động trong menu một cách ổn định.
- **Kết nối Wi-Fi và cấu hình mạng:** Module cần hỗ trợ:
 - Chế độ điểm phát (AP) để cấu hình Wi-Fi thông qua giao diện web đơn giản.
 - Chế độ trạm (STA) để kết nối vào mạng Wi-Fi hiện có và duy trì kết nối ổn định.
- **Gửi dữ liệu lên server:** Hệ thống cần có khả năng đóng gói dữ liệu cảm biến và gửi lên server hoặc một điểm thu thập dữ liệu bên ngoài thông qua các giao thức mạng phù hợp.

2.3 Yêu cầu phi chức năng

Ngoài các yêu cầu chức năng, module MSMS còn phải đáp ứng một số yêu cầu phi chức năng sau:

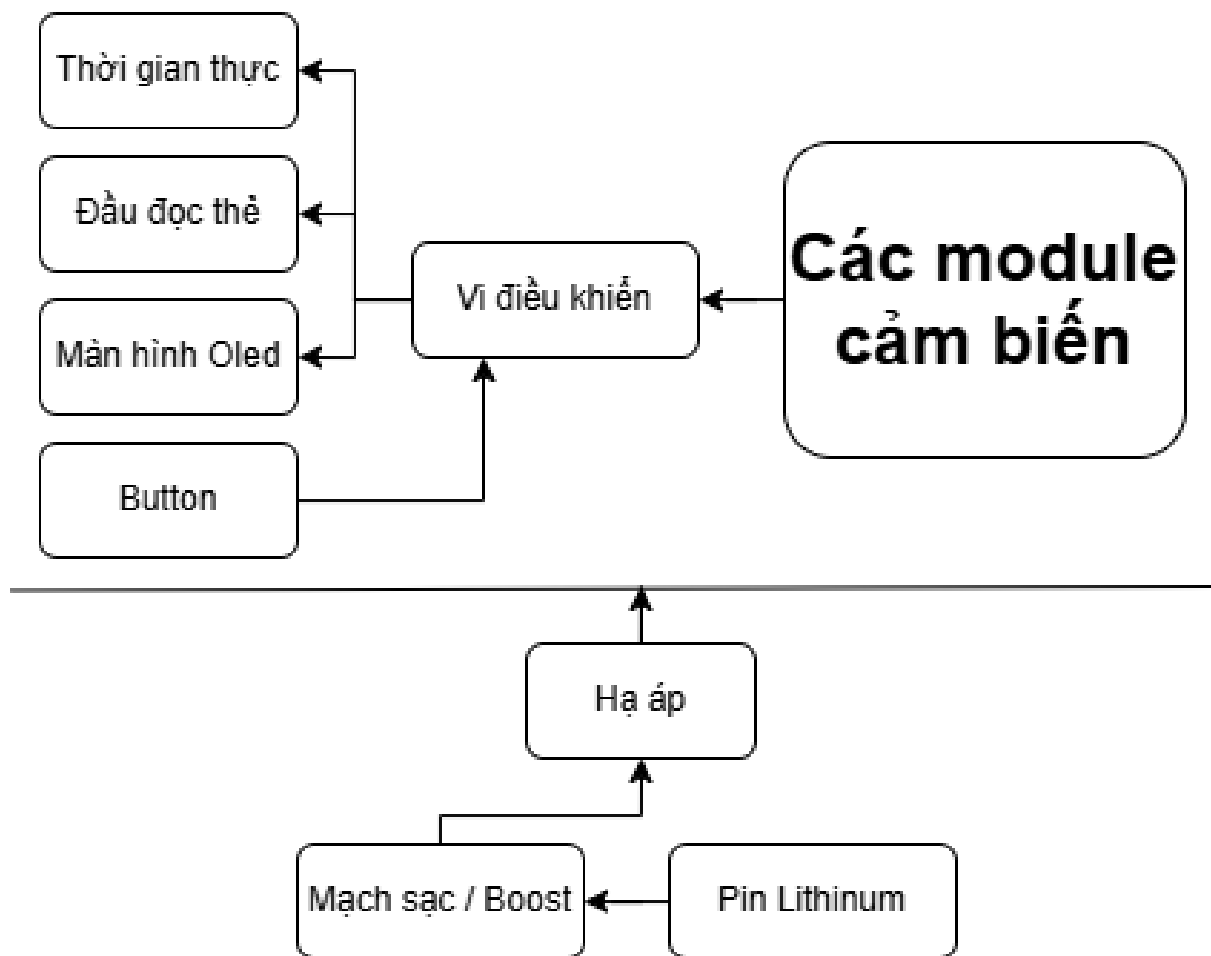
- **Tính mở rộng và tái sử dụng:** Thiết kế tổng thể phải cho phép thêm cảm biến mới, thêm chức năng hiển thị hoặc chức năng mạng mới mà không cần thay đổi lại toàn bộ hệ thống.
- **Tính ổn định và độ tin cậy:** Hệ thống phải hoạt động ổn định trong thời gian dài, không bị treo hoặc mất phản hồi khi đọc cảm biến, cập nhật màn hình hay gửi dữ liệu.
- **Hiệu năng chấp nhận được:** Thời gian phản hồi từ lúc người dùng nhấn nút đến khi màn hình cập nhật cần đủ nhanh để tạo cảm giác mượt mà; việc đọc dữ liệu cảm biến và làm mới màn hình không được gây chậm trễ cho các chức năng khác.
- **Tiêu thụ năng lượng:** Hệ thống nên hạn chế việc cập nhật màn hình không cần thiết, tối ưu chu kỳ đọc cảm biến và sẵn sàng cho việc tích hợp các chế độ tiết kiệm năng lượng trong tương lai.
- **Khả năng mở rộng lên mạng Mesh và cloud:** Dù trong phạm vi đề án chưa triển khai đầy đủ Wi-Fi Mesh và tích hợp cloud, nhưng thiết kế cần để mở, cho phép bổ sung các cơ chế truyền dữ liệu nâng cao mà không phải thay đổi lại kiến trúc cơ bản của module.

CHƯƠNG 3. THIẾT KẾ CHI TIẾT HỆ THỐNG

Từ những yêu cầu chức năng và phi chức năng đã xác định ở phần CHƯƠNG 2, ta sẽ bắt đầu tiến hành thiết kế chi tiết hệ thống của dự án này. Toàn bộ chương 3 này sẽ trình bày thiết kế hardware, firmware từ tổng quan đến chi tiết.

3.1 Thiết kế tổng thể hệ thống

3.1.1 Tổng quan và sơ đồ hệ thống



Hình 3.1 Cấu trúc tổng quan của hệ thống.

Hình 3.1 minh họa sơ đồ khối tổng quan của module MSMS, thể hiện cấu trúc phần cứng và các kết nối chính giữa các thành phần trong hệ thống. Ở trung tâm của sơ đồ là **vi điều khiển ESP32**, đóng vai trò là bộ xử lý trung tâm điều khiển toàn bộ hoạt động của hệ thống.

Các module đầu vào/đầu ra (I/O) kết nối với vi điều khiển:

- **Thời gian thực (RTC DS3231):** Module đồng hồ thời gian thực cung cấp chức

năng ghi nhận và đồng bộ thời gian cho các sự kiện và dữ liệu cảm biến. Module này giao tiếp với ESP32 qua chuẩn I2C và có pin dự phòng để duy trì thời gian ngay cả khi hệ thống mất nguồn.

- **Đầu đọc thẻ (SD Card):** Module lưu trữ dữ liệu ngoài cho phép hệ thống lưu trữ cấu hình, dữ liệu cảm biến hoặc các file log một cách độc lập. Module này giao tiếp với ESP32 qua chuẩn SPI, đảm bảo khả năng lưu trữ dữ liệu ngay cả khi mất kết nối mạng.
- **Màn hình OLED SSD1306:** Module hiển thị giao diện người dùng, thông tin cảm biến và trạng thái hệ thống. Màn hình OLED có độ phân giải 128×64 pixel, giao tiếp qua I2C, tiêu thụ điện năng thấp và cung cấp góc nhìn rộng với độ tương phản cao.
- **Button (Hệ thống nút bấm):** Cung cấp khả năng tương tác của người dùng với hệ thống thông qua 3 nút vật lý (BACK, SELECT và MOVE). Các nút này cho phép người dùng điều hướng menu, cấu hình cảm biến và điều khiển các chức năng của module.

Các module cảm biến: Khối này biểu thị các cảm biến khác nhau mà hệ thống MSMS có thể kết nối và quản lý. Tùy theo cấu hình, mỗi cổng kết nối có thể gắn các loại cảm biến khác nhau như cảm biến môi trường (BME280), cảm biến chất lượng nước, cảm biến khí hoặc các cảm biến khác. Dữ liệu từ các cảm biến này được gửi về vi điều khiển ESP32 để xử lý, lọc và chuẩn hóa thông qua các giao thức giao tiếp như I2C, UART hoặc analog.

Hệ thống quản lý nguồn: Module MSMS được cấp nguồn từ hệ thống quản lý nguồn bao gồm:

- **Pin Lithium:** Nguồn năng lượng chính của hệ thống, cung cấp điện cho toàn bộ thiết bị, đặc biệt hữu ích cho các ứng dụng di động hoặc không có nguồn điện lưới.
- **Mạch sạc / Boost:** Quản lý quá trình sạc pin và có thể tăng áp nếu cần thiết cho các thành phần khác trong hệ thống. Mạch này tích hợp tính năng **power path management**, cho phép hệ thống vừa sạc pin vừa cấp nguồn cho tải đồng thời khi có nguồn ngoài (adapter), hoặc tự động chuyển đổi giữa nguồn từ adapter và pin khi cần thiết, đảm bảo hệ thống luôn hoạt động liên tục không bị gián đoạn.
- **Hạ áp (Buck converter):** Nhận nguồn điện từ pin hoặc mạch sạc và hạ áp xuống mức phù hợp (3.3V) để cấp cho vi điều khiển ESP32 và các module khác, đảm bảo hoạt động ổn định và an toàn.

Sau khi dữ liệu cảm biến được xử lý bởi ESP32, chúng được sử dụng cho hai mục đích chính: hiển thị trực tiếp trên màn hình OLED phục vụ người dùng tại chỗ, và gửi mẫu lên server hoặc hệ thống thu thập dữ liệu bên ngoài thông qua kết nối Wi-Fi. Nhờ đó, module MSMS vừa đóng vai trò là thiết bị đo cục bộ độc lập, vừa có thể là một node trong một hệ thống giám sát phân tán quy mô lớn.

3.1.2 *Linh kiện sử dụng*

a) **Vi điều khiển ESP32**

ESP32 là một dòng vi điều khiển tích hợp module Wi-Fi và Bluetooth hiệu suất cao, giá thành thấp, được sản xuất bởi Espressif Systems. Với khả năng xử lý mạnh mẽ, ESP32 phù hợp cho các ứng dụng IoT, đặc biệt là các ứng dụng yêu cầu khả năng giao tiếp không dây mạnh mẽ và đáng tin cậy.

Bảng 3.1 Thông số kỹ thuật ESP32

Thông số	Giá trị
Vi xử lý	Xtensa® Dual-Core 32-bit LX6
Xung nhịp	160 MHz (tối đa 240 MHz)
Wi-Fi	802.11 b/g/n (2.4 GHz)
Bluetooth	Bluetooth 4.2 + BLE
RAM	520 KB SRAM
Flash	Tối đa 16 MB (ngoài)
GPIO	Tối đa 34 chân
ADC	12-bit, 18 kênh
Điện áp hoạt động	3.3 V



Hình 3.2 Vi điều khiển ESP32.

b) Module RTC DS3231

DS3231 là một module đồng hồ thời gian thực (RTC) có độ chính xác cao, sử dụng dao động thạch anh tích hợp bên trong và bộ bù nhiệt độ. Nó cho phép đo thời gian chính xác đến từng giây, có pin dự phòng nên vẫn hoạt động trong điều kiện mất nguồn. DS3231 hỗ trợ giao tiếp I2C và thường được sử dụng trong các ứng dụng cần hẹn giờ, lưu trữ log theo thời gian thực.

Bảng 3.2 Thông số kỹ thuật DS3231

Thông số	Giá trị
Chức năng	Real-Time Clock (RTC)
Độ chính xác	± 2 ppm
Giao tiếp	I2C
Điện áp hoạt động	2.3 – 5.5 V
Pin dự phòng	CR2032
Dòng tiêu thụ	$\sim 3 \mu\text{A}$ (backup)
Bộ dao động	TCXO tích hợp
Nhiệt độ hoạt động	-40°C đến +85°C



Hình 3.3 Module DS3231.

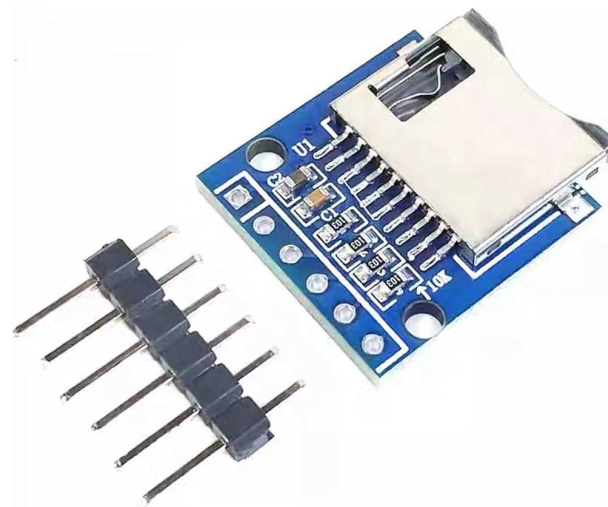
c) Module lưu trữ SD Card

SD Card là một giải pháp lưu trữ ngoài tiện lợi, thường được sử dụng trong các hệ thống nhúng để lưu dữ liệu một cách lâu dài và đáng tin cậy.

SD Card được giao tiếp với vi điều khiển thông qua chuẩn SPI – một giao tiếp nối tiếp tốc độ cao, dễ tích hợp và tương thích với cả ESP32 lẫn STM32.

Bảng 3.3 Thông số kỹ thuật thẻ nhớ SD

Thông số	Giá trị
Chuẩn thẻ	MicroSD
Giao tiếp	SPI
Điện áp hoạt động	3.3 V
Dung lượng hỗ trợ	Tối đa 32 GB (FAT32)
Dòng tiêu thụ	50 – 100 mA
Ứng dụng	Lưu trữ dữ liệu cảm biến



Hình 3.4 Module SD Card.

d) **Module mạch sạc IP5306**

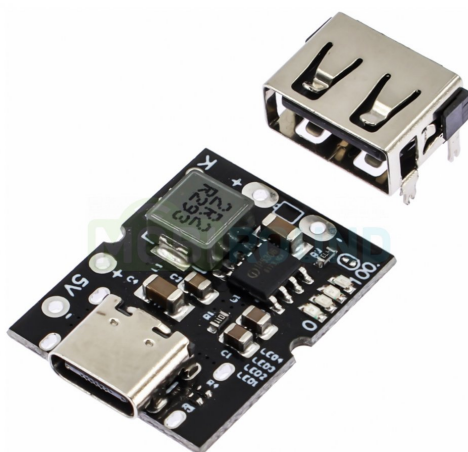
IP5306 là một module quản lý nguồn tích hợp được thiết kế để sạc pin lithium và cung cấp đầu ra 5V cho các ứng dụng di động và thiết bị nhúng. Module này tích hợp các chức năng sạc pin, boost converter và power path management trong một chip duy nhất, giúp đơn giản hóa thiết kế mạch nguồn và tối ưu hiệu suất sử dụng pin.

Module IP5306 hoạt động như một hệ thống quản lý nguồn thông minh với khả năng tự động chuyển đổi giữa chế độ sạc và xả. Khi có nguồn ngoài qua cổng

USB Type-C, module sẽ tự động sạc pin và đồng thời cấp nguồn cho tải thông qua tính năng power path management. Khi không có nguồn ngoài, module sẽ sử dụng năng lượng từ pin và boost lên 5V để cấp cho hệ thống.

Bảng 3.4 Thông số kỹ thuật IP5306

Thông số	Giá trị
Chức năng	Quản lý nguồn, sạc pin, boost converter
Điện áp đầu vào	5 – 5.5 V
Điện áp cắt sạc	4.2 V / 4.35 V ($\pm 0.5\%$)
Dòng sạc	2.4 A ($\pm 5\%$)
Điện áp đầu ra	5.0 – 5.15 V (bù tổn thất dây)
Dòng đầu ra tối đa	2 A
Độ gợn sóng đầu ra	100 mV
Hiệu suất chuyển đổi	92.5% (đầu vào 3.6V, đầu ra 5V@2A)
Điện áp sạc trước	2.8 V (dòng 180 mA)
Điện áp sạc lại	4.1 V
Dòng cắt tự động	100 mA (sau khi đạt điện áp phao)
Dòng tắt đầu ra	< 50 mA (liên tục)



Hình 3.5 Module mạch sạc IP5306.

Module IP5306 có các đặc điểm nổi bật sau:

- **Tự động tắt đầu ra khi tải thấp:** Khi dòng tải liên tục nhỏ hơn 50mA, đầu ra sẽ tự động bị tắt để tiết kiệm năng lượng pin.
- **Hỗ trợ điều khiển bằng nút bấm ngoài:** Module hỗ trợ kết nối nút bấm với điểm K và đầu ra âm. Nhấn nhanh một lần để bật màn hình hiển thị nguồn và bật đầu ra 5V; nhấn hai lần ngắn liên tiếp để tắt màn hình hiển thị và tắt đầu ra 5V.
- **Quản lý chu kỳ sạc thông minh:** Khi dòng sạc giảm xuống 100mA sau khi đạt đến điện áp phao cuối cùng, chu kỳ sạc sẽ tự động kết thúc. Khi điện áp pin giảm dưới 4.1V, chu kỳ sạc lại sẽ tự động bắt đầu.
- **Bảo vệ pin điện áp thấp:** Khi điện áp pin thấp hơn 2.8V, pin sẽ được sạc trước với dòng điện 180mA để đảm bảo an toàn và kéo dài tuổi thọ pin.
- **Tích hợp cổng USB-A:** Vị trí hàn ổ cắm USB-A cái được cung cấp ở mặt sau của module, cho phép người dùng tự lắp đặt để có đầu ra USB-A tiêu chuẩn phục vụ các ứng dụng khác nhau.
- **Hiển thị trạng thái bằng LED:** Module tích hợp 4 đèn LED để hiển thị trạng thái nguồn, mức pin và trạng thái sạc/xả, giúp người dùng dễ dàng theo dõi tình trạng hệ thống.

Trong hệ thống MSMS, module IP5306 đóng vai trò là khối quản lý nguồn chính, đảm bảo hệ thống có thể hoạt động độc lập với pin lithium trong thời gian dài, đồng thời tự động sạc pin khi có nguồn ngoài và cung cấp nguồn ổn định 5V cho các thành phần khác trong hệ thống.

e) **Pin Lithium Polymer (LiPo)**

Pin Lithium Polymer là nguồn năng lượng chính của hệ thống MSMS, cung cấp điện năng cho toàn bộ thiết bị khi hoạt động độc lập không có nguồn điện lưới. Pin LiPo được sử dụng trong hệ thống có kích thước nhỏ gọn, trọng lượng nhẹ và có khả năng cung cấp dòng điện ổn định, rất phù hợp cho các ứng dụng IoT di động và thiết bị nhúng.

Pin LiPo hoạt động dựa trên nguyên lý hóa học của lithium-ion, với chất điện phân dạng polymer cho phép thiết kế pin dạng túi (pouch cell) mỏng và linh hoạt hơn so với pin lithium-ion dạng hình trụ truyền thống. Pin được tích hợp mạch bảo vệ PCM (Protection Circuit Module) để ngăn chặn các tình huống nguy hiểm như quá sạc, quá xả, ngắn mạch và quá dòng, đảm bảo an toàn trong quá trình sử dụng.

Bảng 3.5 Thông số kỹ thuật Pin Lithium Polymer

Thông số	Giá trị
Loại pin	Lithium Polymer (LiPo) Pouch Cell
Model	FL 803450
Kích thước	8.0 mm × 34 mm × 50 mm (803450)
Điện áp danh định	3.7 V
Điện áp sạc đầy	4.2 V
Điện áp xả tối thiểu	2.8 V
Dung lượng	1500 mAh
Năng lượng	5.55 Wh
Mạch bảo vệ	PCM tích hợp
Kết nối	JST connector (dây đỏ +, dây đen -)



Hình 3.6 Pin Lithium Polymer 3.7V 1500mAh (Model FL 803450).

Pin LiPo trong hệ thống MSMS được kết nối với module IP5306 thông qua đầu nối JST để quản lý quá trình sạc và xả. Khi có nguồn ngoài qua cổng USB Type-C của IP5306, pin sẽ được sạc với dòng điện tối đa 2.4A cho đến khi đạt điện áp 4.2V. Khi không có nguồn ngoài, pin sẽ cung cấp năng lượng cho hệ thống thông qua module IP5306, đảm bảo hệ thống có thể hoạt động liên tục trong thời gian dài mà không cần kết nối với nguồn điện lưới.

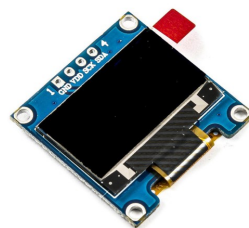
Với dung lượng 1500mAh và điện áp danh định 3.7V, pin có thể cung cấp năng lượng cho hệ thống MSMS trong nhiều giờ tùy thuộc vào mức tiêu thụ công suất của các thành phần. Ở kích bản Peak với hai cảm biến hoạt động đồng thời ở mức tiêu thụ cao nhất, pin có thể duy trì hoạt động khoảng 1.9 giờ. Trong điều kiện sử dụng thực tế với các chế độ tiết kiệm năng lượng, thời gian hoạt động có thể kéo dài từ 5.1 đến 6.9 giờ. Module IP5306 sẽ tự động quản lý chu kỳ sạc/xả và bảo vệ pin khỏi các tình huống nguy hiểm, kéo dài tuổi thọ và đảm bảo an toàn cho hệ thống.

f) Module màn hình OLED SSD1306

SSD1306 là màn hình OLED (Organic Light Emitting Diode) với độ phân giải 128x64 pixel, sử dụng giao tiếp I2C hoặc SPI. Màn hình OLED có ưu điểm là tiêu thụ điện năng thấp, góc nhìn rộng, độ tương phản cao và có thể hiển thị trong điều kiện ánh sáng mạnh. SSD1306 rất phù hợp cho các ứng dụng IoT cần hiển thị thông tin trực quan với mức tiêu thụ năng lượng thấp.

Bảng 3.6 Thông số kỹ thuật SSD1306

Thông số	Giá trị
Chức năng	Màn hình OLED hiển thị
Kích thước	0.96 inch
Độ phân giải	128 × 64 pixel
Màu hiển thị	Trắng / Xanh
Giao tiếp	I2C / SPI
Điện áp hoạt động	3.3 – 5 V
Dòng tiêu thụ	~20 mA



© Photo by ElectroPeak

Hình 3.7 Module màn hình OLED SSD1306.

Màn hình SSD1306 có thể hiển thị văn bản, số và đồ họa đơn giản, rất phù hợp để hiển thị các thông số quan trắc như PM2.5, PM10, nhiệt độ, độ ẩm và áp suất không khí. Màn hình có thể kết nối trực tiếp với ESP32 thông qua giao tiếp I2C, chỉ cần 2 chân (SDA, SCL), giúp tiết kiệm số chân GPIO và đơn giản hóa kết nối.

g) IC ổn áp AMS1117-3.3

AMS1117-3.3 là IC ổn áp tuyến tính có khả năng cung cấp dòng điện lên đến 1A, được sử dụng để hạ điện áp từ nguồn pin xuống 3.3V để cấp nguồn cho ESP32 và các cảm biến. IC này có độ ổn định điện áp tốt, nhiễu thấp và giá thành rẻ, rất phù hợp cho các ứng dụng IoT tiêu thụ điện năng thấp.

Bảng 3.7 Thông số kỹ thuật AMS1117-3.3

Thông số	Giá trị
Chức năng	IC ổn áp tuyến tính
Điện áp đầu vào	4.5 – 15 V
Điện áp đầu ra	3.3 V
Dòng tối đa	1 A
Dropout Voltage	~1.1 V
Kiểu đóng gói	SOT-223
Bảo vệ	Quá dòng, quá nhiệt



Hình 3.8 IC ổn áp AMS1117-3.3.

AMS1117-3.3 hoạt động với điện áp đầu vào từ 4.75V đến 15V và cung cấp

điện áp đầu ra ổn định 3.3V. IC này yêu cầu các tụ điện lọc đầu vào và đầu ra để đảm bảo hoạt động ổn định, thường là tụ 10 μ F ở đầu vào và tụ 22 μ F ở đầu ra. Trong hệ thống này, AMS1117-3.3 được sử dụng để hạ điện áp từ đầu ra 5V của module IP5306 xuống 3.3V để cấp nguồn cho ESP32 và các cảm biến hoạt động ở mức điện áp 3.3V.

3.2 Thiết kế chi tiết hệ thống

3.2.1 Tính toán đáp ứng công suất nguồn

Hệ thống được cấp nguồn từ pin lithium thông qua module quản lý nguồn IP5306. Module IP5306 có khả năng sạc pin và cung cấp đầu ra 5V ổn định cho hệ thống. Điện áp 5V từ IP5306 được hạ xuống 3.3V thông qua IC ổn áp tuyến tính AMS1117-3.3 để cấp nguồn cho ESP32 và các cảm biến hoạt động ở mức điện áp 3.3V.

AMS1117-3.3 là IC ổn áp tuyến tính có khả năng cung cấp dòng điện lên đến 1A, phù hợp cho các ứng dụng IoT tiêu thụ điện năng thấp. IC này có độ ổn định điện áp tốt và nhiễu thấp, đảm bảo nguồn cung cấp ổn định cho hệ thống. Để đảm bảo hoạt động ổn định, cần sử dụng các tụ điện lọc đầu vào và đầu ra phù hợp (thường là 10 μ F và 22 μ F).

Phân tích công suất:

Bảng 3.8 Bảng tính toán công suất nguồn ở kích bản Peak

Linh kiện	Điện áp (V)	Dòng Peak (mA)	Công suất (W)
ESP32	3.3	240	0.792
SSD1306 OLED	3.3	30	0.099
DS3231	3.3	0.2	0.001
Thẻ SD (SPI)	3.3	100	0.330
2 cảm biến	3.3	30	0.099
Tổng tải 3.3V	—	400.2	1.321
Tổn hao AMS1117	5 \rightarrow 3.3	—	0.681
TỔNG CÔNG SUẤT NGUỒN 5V	5	400.2	2.002

Từ bảng tính toán trên, tổng công suất tiêu thụ ở kích bản Peak là 2.002W ở mức 5V, tương đương dòng điện 400.2mA. Khi tính từ nguồn pin lithium qua module IP5306, với hiệu suất chuyển đổi boost của IP5306 khoảng 92.5% và tổn hao của AMS1117, hiệu suất tổng thể của hệ thống nguồn khoảng 68%, công suất đầu vào từ pin cần thiết là:

$$P_{in} = \frac{2.002}{0.68} \approx 2.94 \text{ W}$$

Với điện áp pin trung bình khoảng 3.7V, dòng điện đầu vào từ pin:

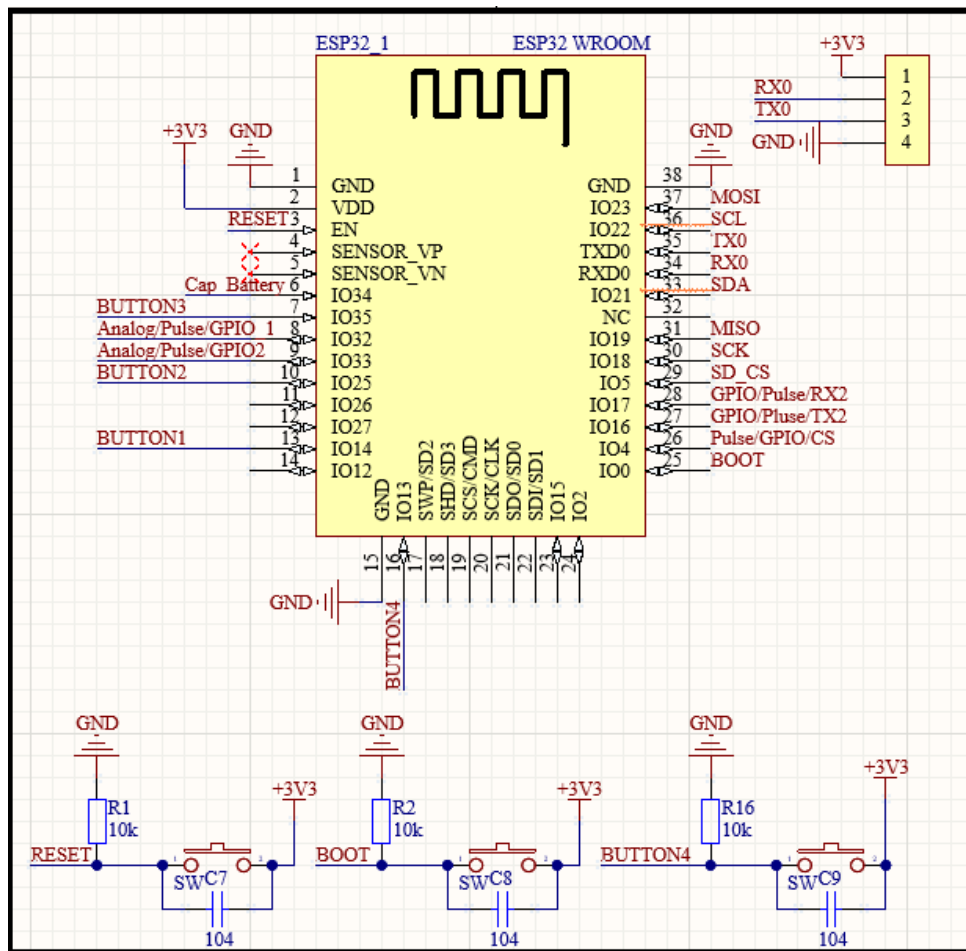
$$I_{in} = \frac{2.94}{3.7} \approx 795 \text{ mA}$$

Với pin lithium dung lượng 1500mAh, thời gian hoạt động lý thuyết ở kịch bản Peak:

$$T_{peak} = \frac{1500}{795} \approx 1.9 \text{ giờ}$$

Trong thực tế, với các chế độ tiết kiệm năng lượng như ESP32 deep sleep, tắt màn hình SSD1306 khi không hiển thị, SD Card ở chế độ idle, và giảm tần suất đọc cảm biến, công suất tiêu thụ trung bình giảm xuống còn khoảng 0.8-1.1W, tương đương dòng điện 216-297mA từ pin. Với mức tiêu thụ này, thời gian hoạt động có thể đạt 5.1-6.9 giờ. Module IP5306 cũng có tính năng tự động tắt đầu ra khi tải thấp (<50mA), giúp tiết kiệm năng lượng pin khi hệ thống ở chế độ chờ.

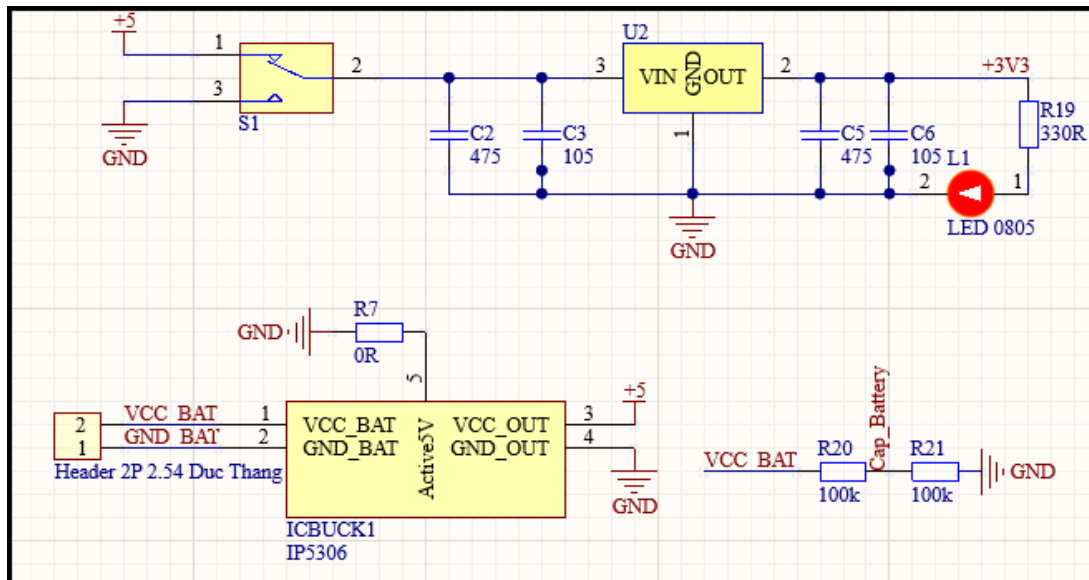
3.2.2 Thiết kế mạch nguyên lý (Schematic)



Hình 3.9 Khối xử lý và nút bấm.

Khối này là khối các nút bấm boot và reset cùng với header UART và ESP32 WROOM để nạp code. Khối bao gồm:

- **Module ESP32 WROOM:** Vi điều khiển trung tâm của hệ thống với các chân GPIO được kết nối với các module ngoại vi và cảm biến.
- **Nút bấm Reset:** Nút bấm SWC7 kết nối với chân RESET (pin 3) của ESP32 qua điện trở pull-up 10kΩ (R1) và tụ lọc 104 để khởi động lại hệ thống.
- **Nút bấm Boot:** Nút bấm SWC8 kết nối với chân BOOT (pin 25) của ESP32 qua điện trở pull-up 10kΩ (R2) và tụ lọc 104 để đưa ESP32 vào chế độ nạp chương trình.
- **Header UART0:** Header 4 chân cung cấp các kết nối UART0 (TX0, RX0), nguồn 3.3V và GND để nạp code và giao tiếp serial với ESP32 qua cổng USB-to-Serial.

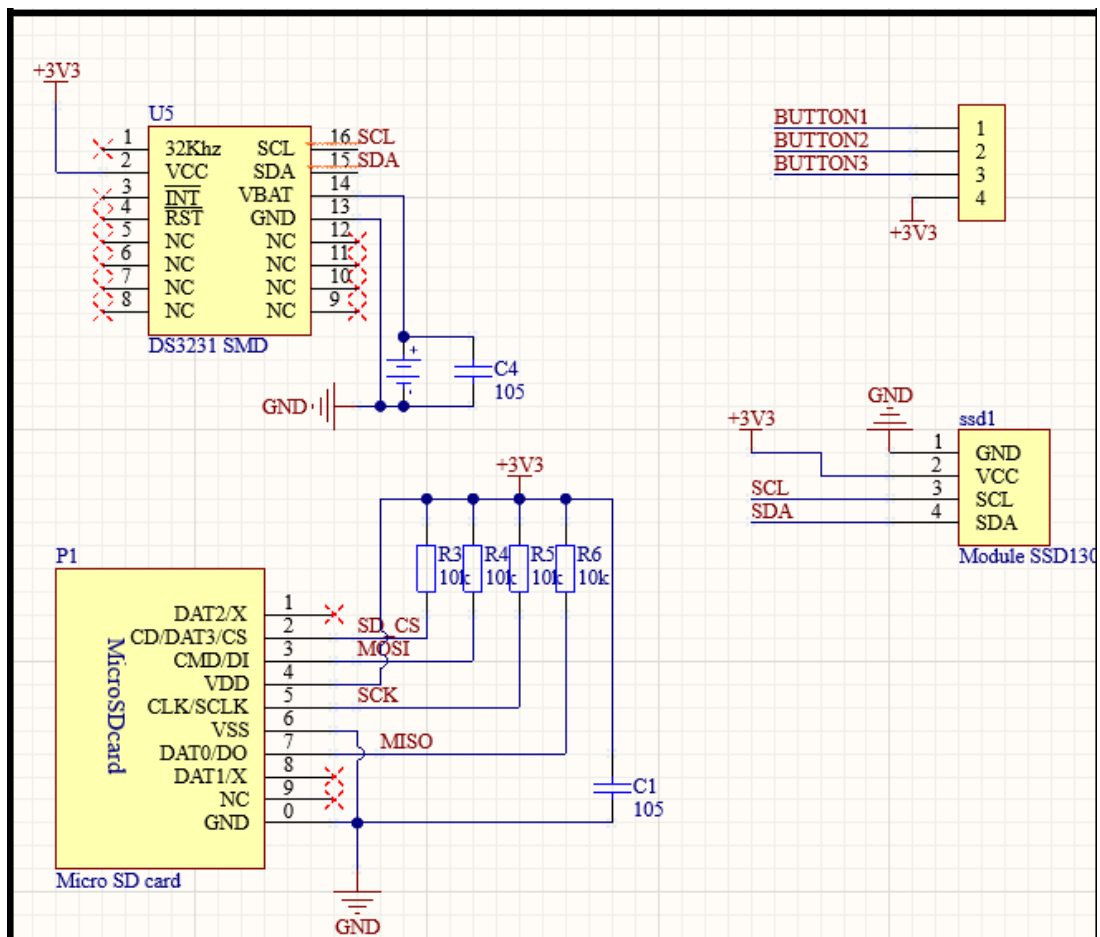


Hình 3.10 Khối nguồn.

Khối nguồn tích hợp module IP5306 để quản lý nguồn pin và cung cấp các mức điện áp cần thiết cho hệ thống:

- **Đầu vào pin lithium:** Header 2 chân (Header 2P 2.54) kết nối với pin lithium (VCC_BAT và GND_BAT).
- **Module IP5306 (ICBUCK1):** IC quản lý nguồn chính có các chức năng:
 - Nhận điện áp từ pin lithium qua chân VCC_BAT và GND_BAT.
 - Boost converter tạo đầu ra 5V ổn định (VCC_OUT) từ điện áp pin.

- Quản lý sạc pin khi có nguồn ngoài qua cổng USB Type-C.
- Power path management để vừa sạc pin vừa cấp nguồn cho tải.
- **Mạch giám sát pin:** Bộ chia áp gồm hai điện trở 100kΩ (R20, R21) tạo tín hiệu Cap_Battery để ESP32 đọc điện áp pin và hiển thị mức pin.
- **IC ổn áp AMS1117-3.3 (U2):** Hạ điện áp từ 5V xuống 3.3V để cấp nguồn cho ESP32 và các module hoạt động ở mức 3.3V. Có các tụ lọc đầu vào (C2, C3) và đầu ra (C5, C6) để đảm bảo nguồn ổn định.
- **LED báo nguồn:** LED 0805 với điện trở hạn dòng 330Ω để hiển thị trạng thái nguồn 3.3V.
- **Công tắc nguồn (S1):** Công tắc SPDT để bật/tắt nguồn 3.3V.



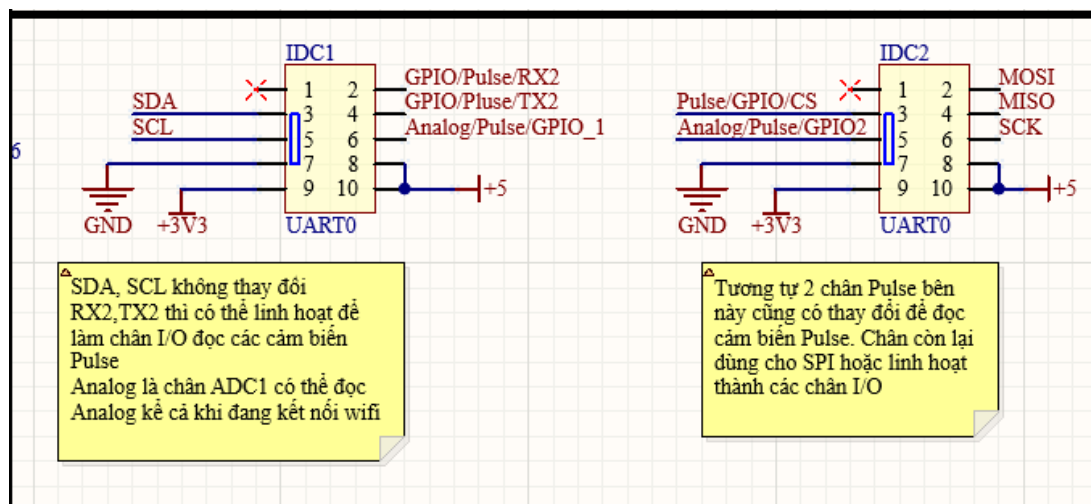
Hình 3.11 Khối cảm biến tích hợp.

Khối này tích hợp các module ngoại vi chính của hệ thống MSMS, bao gồm:

- **Module RTC DS3231 (U5):** Module đồng hồ thời gian thực được kết nối với ESP32 qua giao tiếp I2C (chân SCL và SDA). Module được cấp nguồn 3.3V và có pin dự

phòng CR2032 (VBAT) kèm tụ lọc C4 để duy trì thời gian ngay cả khi mất nguồn chính.

- **Module Micro SD Card (P1):** Khe cắm thẻ nhớ Micro SD được kết nối với ESP32 qua giao tiếp SPI (MOSI, MISO, SCK, CS). Module được cấp nguồn 3.3V và có các điện trở pull-up 10kΩ trên các đường SPI để đảm bảo tín hiệu ổn định.
- **Module màn hình OLED SSD1306 (ssd1):** Màn hình OLED được kết nối với ESP32 qua giao tiếp I2C (SCL và SDA), chia sẻ bus I2C chung với DS3231. Module được cấp nguồn 3.3V.
- **Hệ thống nút bấm:** Khối tích hợp 3 nút bấm (BACK, SELECT và MOVE) được kết nối với các chân GPIO của ESP32. Mỗi nút có điện trở pull-up 10kΩ và tụ lọc để chống nhiễu và dội phím.



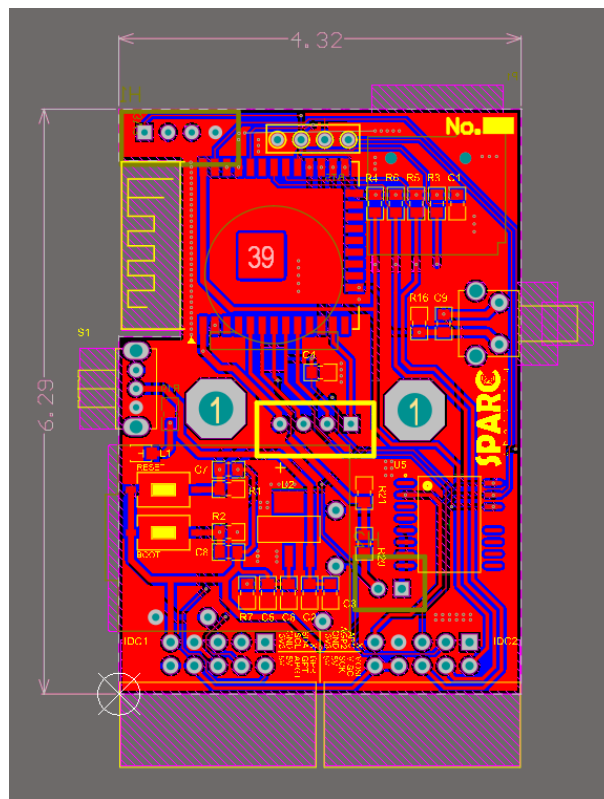
Hình 3.12 Khối 2 cổng cảm biến.

Khối này thiết kế hai cổng kết nối cảm biến độc lập (IDC1 và IDC2), mỗi cổng có 10 chân IDC header, cho phép kết nối nhiều loại cảm biến khác nhau:

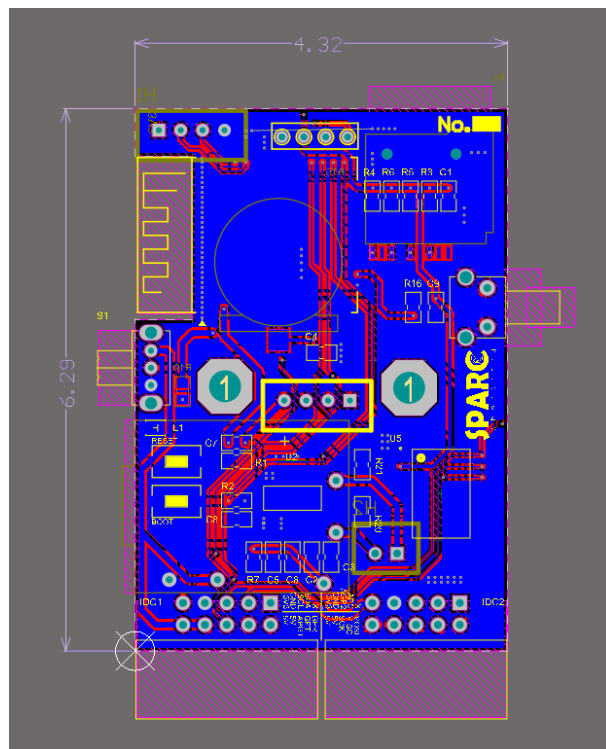
- **Cổng cảm biến 1 (IDC1):**
 - Cấp nguồn: GND, +3V3, +5V
 - Giao tiếp I2C: SDA (chân 3), SCL (chân 5) - các chân này có chức năng cố định
 - Giao tiếp UART: RX2 (chân 2), TX2 (chân 4) - có thể linh hoạt chuyển thành GPIO để đọc cảm biến Pulse
 - Analog/Pulse/GPIO: Chân 6 (GPIO_1) có thể đọc analog (ADC1), đọc Pulse hoặc làm GPIO, đặc biệt ADC1 có thể hoạt động ngay cả khi ESP32 kết nối Wi-Fi

- Chân 1 và 10: Dự phòng hoặc cấu hình đặc biệt
- **Cổng cảm biến 2 (IDC2):**
 - Cấp nguồn: GND, +3V3, +5V
 - Giao tiếp SPI: MOSI (chân 2), MISO (chân 4), SCK (chân 6), CS (chân 3) - có thể linh hoạt chuyển thành GPIO
 - Analog/Pulse/GPIO: Chân 5 (GPIO_2) có thể đọc analog, đọc Pulse hoặc làm GPIO
 - Chân 1 và 10: Dự phòng hoặc cấu hình đặc biệt
- **Tính linh hoạt:** Các chân Pulse trên cả hai cổng có thể được cấu hình để đọc cảm biến Pulse. Các chân SPI và GPIO có thể được sử dụng linh hoạt tùy theo loại cảm biến được kết nối, cho phép hệ thống MSMS hỗ trợ nhiều loại cảm biến khác nhau mà không cần thay đổi phần cứng.

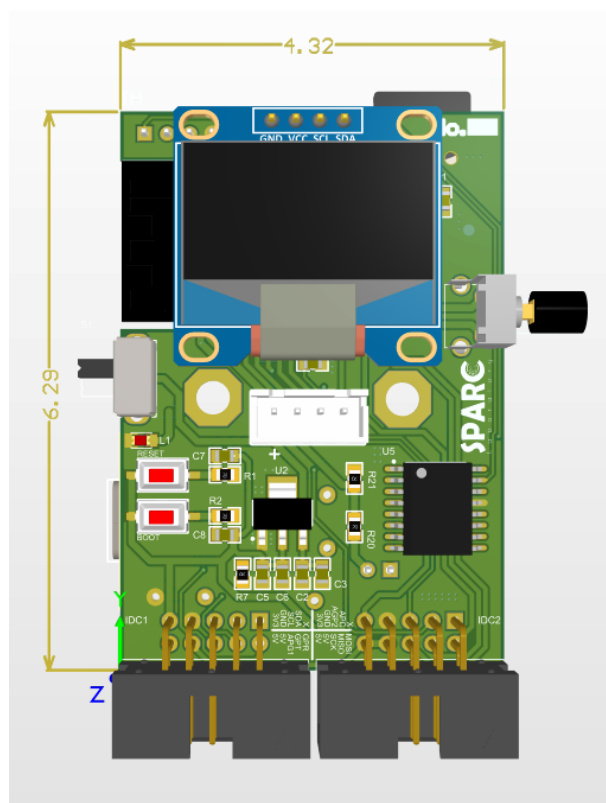
3.2.3 Layout PCB



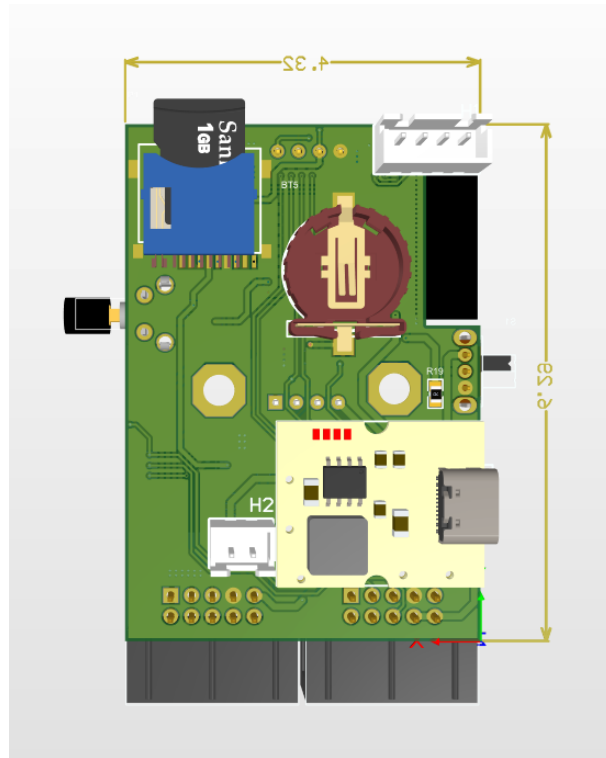
Hình 3.13 Top Layout.



Hình 3.14 Bot Layout.



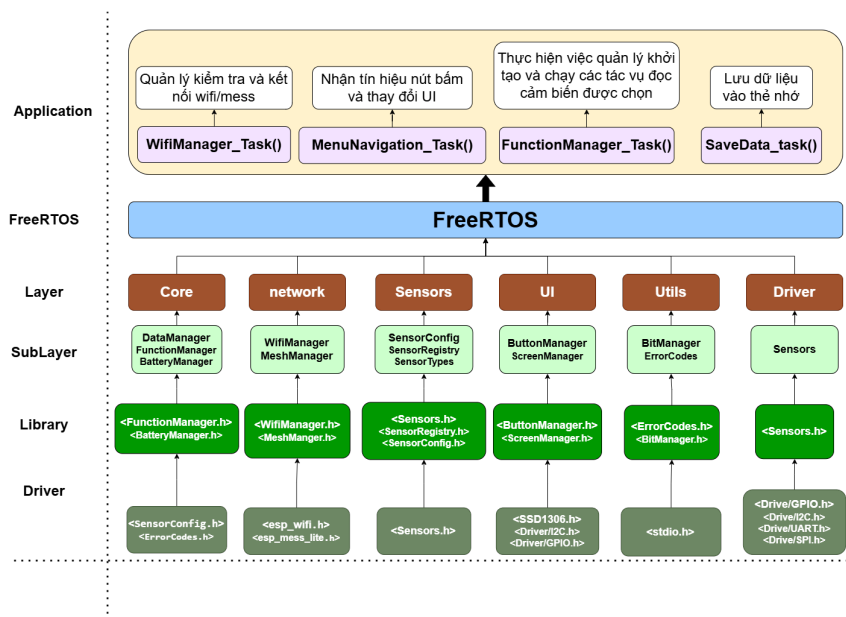
Hình 3.15 3D Top.



Hình 3.16 3D Bot.

3.2.4 Thiết kế Firmware

Trong dự án này, ESP32 được sử dụng làm bộ xử lý trung tâm của hệ thống, đảm nhận việc đọc dữ liệu từ các cảm biến và truyền tải lên Server thông qua mạng Wi-Fi. Firmware của ESP32 được thiết kế theo kiến trúc phân lớp (layered architecture) với sự phân tách rõ ràng giữa các tầng ứng dụng, hệ điều hành, thư viện, driver và phần cứng vật lý.



Hình 3.17 Sơ đồ kiến trúc phần mềm ESP32.

Kiến trúc phân lớp của hệ thống

Sơ đồ mô tả kiến trúc firmware phân lớp từ tác vụ ứng dụng xuống driver phần cứng. Hệ thống được tổ chức thành 6 tầng chính:

1. **Tầng tác vụ ứng dụng (Application/Task):** Chứa các task chính chạy song song trên FreeRTOS, mỗi task đảm nhận một chức năng cụ thể và tương tác với kernel qua API FreeRTOS.
2. **Tầng FreeRTOS:** Hệ điều hành thời gian thực quản lý lập lịch task, điều phối tài nguyên và cung cấp dịch vụ IPC (queue, semaphore, mutex) cho các tầng trên.
3. **Tầng thành phần (Component):** Firmware được chia thành sáu thành phần chức năng: **Core** (logic lõi), **network** (mạng), **Sensors** (cảm biến), **UI** (giao diện người dùng), **Utils** (tiện ích), **Driver** (trình điều khiển phần cứng).
4. **Tầng mô-đun (Module):** Mỗi thành phần gồm các mô-đun cụ thể: Core có DataManager, FunctionManager, BatteryManager; network có WifiManager, MeshManager; Sensors có SensorConfig, SensorRegistry, SensorTypes; UI có ButtonManager, ScreenManager; Utils có BitManager, ErrorCodes; Driver chứa lớp trừu tượng cho cảm biến.
5. **Tầng header/API:** Các file tiêu đề (.h) tương ứng với từng mô-đun (ví dụ FunctionManager.h, WifiManager.h, SensorRegistry.h, ButtonManager.h, ScreenManager.h, ErrorCodes.h, BitManager.h) định nghĩa giao diện lập trình để các phần khác của hệ thống gọi mà không phụ thuộc vào triển khai bên trong.
6. **Tầng trừu tượng phần cứng và thư viện hệ thống:** Lớp dưới cùng gồm driver và thư viện nền: mạng dùng esp_wifi.h, esp_mesh_lite.h; UI dùng SSD1306.h, Driver/I2C, Driver/GPIO; Driver dùng Driver/GPIO.h, Driver/I2C.h, Driver/UART.h, Driver/SPI.h; Utils dùng thư viện chuẩn như stdio.h. Ứng dụng không gọi trực tiếp phần cứng mà thông qua các API này.

Các task trong tầng ứng dụng

FreeRTOS quản lý bốn task chính tương ứng với sơ đồ:

- **WifiManager_Task():** Quản lý kiểm tra và kết nối Wi-Fi/mesh. Task chịu trách nhiệm duy trì trạng thái kết nối mạng, cấu hình Wi-Fi (AP/STA), và tích hợp mesh khi mở rộng.

23:24 4G 21

captive.apple.com
Environment Monitor Device

< > Đăng nhập vào Wi-Fi xác thực Hủy

Cấu hình WiFi

Cấu hình WiFi

Tên WiFi:

Mật khẩu:

Cấu hình WebSocket

URL WebSocket:

Cấu hình HTTP

Địa chỉ IP HTTP:

Port HTTP:

Lưu cấu hình

Hình 3.18 Web server cấu hình hệ thống.

- **MenuNavigation_Task()**: Nhận tín hiệu nút bấm và thay đổi giao diện người dùng. Task xử lý sự kiện từ ButtonManager và cập nhật trạng thái hiển thị qua ScreenManager.
- **FunctionManager_Task()**: Thực hiện quản lý khởi tạo và chạy các tác vụ đọc cảm biến được chọn. Task điều phối SensorRegistry, SensorConfig và đọc dữ liệu từ các cảm biến theo cấu hình.
- **SaveData_task()**: Lưu dữ liệu vào thẻ nhớ. Task ghi dữ liệu từ DataManager hoặc cảm biến ra bộ nhớ ngoài (SD) theo chu kỳ hoặc sự kiện.

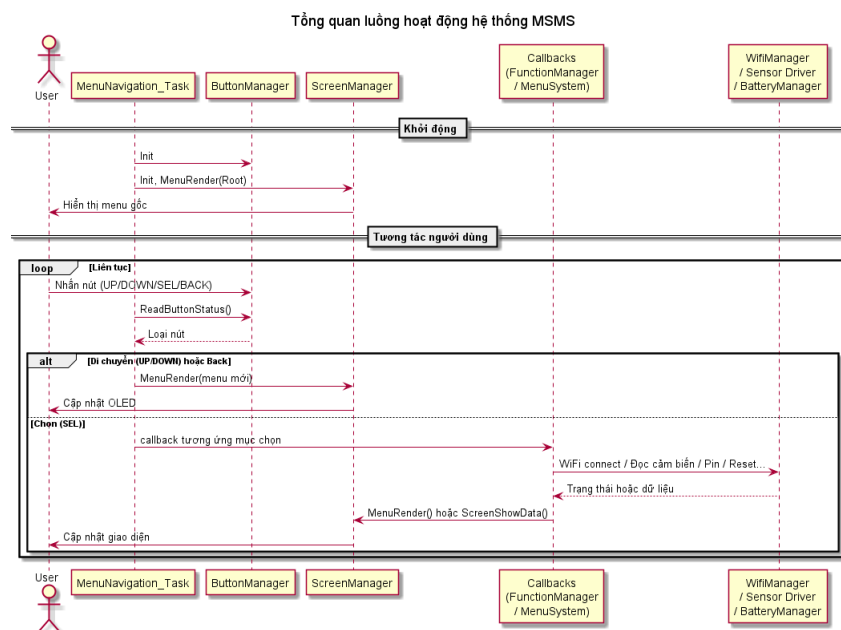
Luồng dữ liệu và tương tác giữa các tầng

Thiết kế đảm bảo phân tách rõ ràng và phụ thuộc một chiều từ trên xuống:

- Các task ứng dụng gọi API của các thành phần (Core, network, Sensors, UI, Utils) thông qua các file header tương ứng.
- Các mô-đun thành phần (ví dụ WifiManager, FunctionManager, ButtonManager, ScreenManager) sử dụng driver và thư viện hệ thống (esp_wifi.h, SSD1306.h, Driver/I2C.h, Driver/GPIO.h, ...) để giao tiếp với phần cứng.
- Driver và HAL che phần cứng thật; thay đổi board hoặc ngoại vi chủ yếu chỉ cần sửa ở tầng driver, ít ảnh hưởng tầng ứng dụng.

Kiến trúc phân lớp giúp dễ bảo trì, mở rộng và tái sử dụng mã, phù hợp hệ nhúng đa task và đa phần cứng như ESP32.

Diagram Sequence

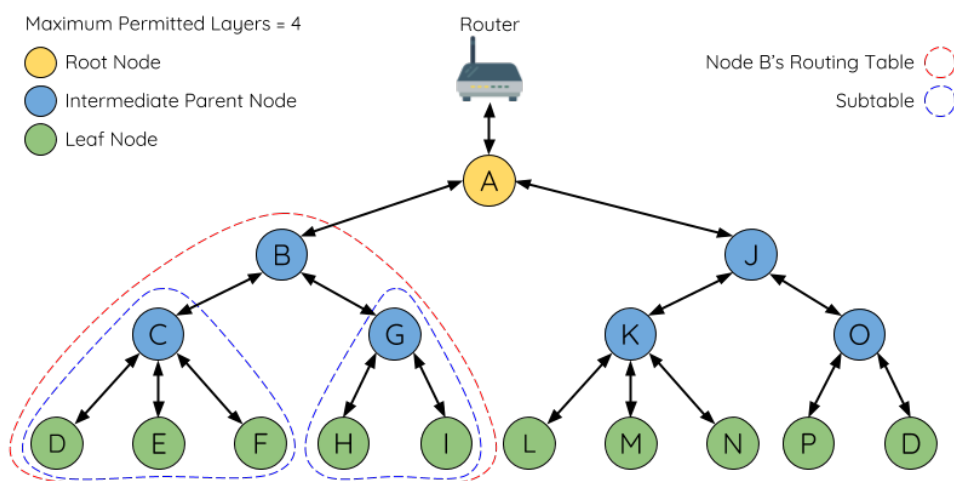


Hình 3.19 Tổng quan luồng hoạt động hệ thống MSMS (Sequence Diagram).

Sơ đồ chuỗi (Sequence Diagram) trong Hình 3.19 mô tả tổng quan luồng hoạt động của hệ thống MSMS theo hai giai đoạn chính. **Giai đoạn khởi động:** task MenuNavigation_Task khởi tạo ButtonManager và ScreenManager, sau đó gọi MenuRender(Root) để hiển thị menu gốc lên màn hình OLED cho người dùng. **Giai đoạn tương tác người dùng:** hệ thống lặp liên tục — người dùng nhấn một trong bốn nút (UP, DOWN, SEL, BACK); ButtonManager đọc trạng thái nút qua ReadButtonStatus() và trả về loại nút cho MenuNavigation_Task. Nếu là UP/DOWN hoặc BACK, task cập nhật menu hiện tại (di chuyển hoặc quay lại

menu cha) và gọi ScreenManager để vẽ lại menu lên OLED. Nếu là SEL (chọn mục), task gọi callback tương ứng với mục được chọn; callback (do Function-Manager/MenuSystem quản lý) có thể thực hiện kết nối Wi-Fi, đọc cảm biến, xem trạng thái pin, reset port, v.v. thông qua WifiManager, driver cảm biến hoặc BatteryManager, nhận trạng thái hoặc dữ liệu trả về, rồi gọi MenuRender() hoặc ScreenShowData() để cập nhật giao diện trên OLED. Như vậy, sơ đồ tóm tắt luồng từ tương tác người dùng đến xử lý nội bộ và cập nhật hiển thị.

• Tổng quan Wi-Fi Mesh trong hệ thống MSMS



Hình 3.20 Sơ đồ topology mạng Wi-Fi Mesh của các node MSMS.

Trong phiên bản hiện tại, dự án tập trung vào việc xây dựng một node MSMS đơn lẻ với khả năng kết nối Wi-Fi AP/STA và gửi dữ liệu cảm biến lên server. Tuy nhiên, kiến trúc firmware được thiết kế sao cho có thể mở rộng thành **mạng Wi-Fi Mesh** trong các giai đoạn tiếp theo.

Về lý thuyết, Wi-Fi Mesh là mô hình mạng trong đó nhiều node Wi-Fi đóng vai trò như các nút trung gian chuyển tiếp dữ liệu, thay vì tất cả node đều kết nối trực tiếp đến một Access Point duy nhất. Một mạng Mesh điển hình bao gồm:

- **Root node (Gateway):** Node gốc kết nối với mạng ngoài (LAN/Internet hoặc server nội bộ). Tất cả dữ liệu cảm biến từ các node con cuối cùng đều được tập trung tại đây.
- **Intermediate nodes:** Các node trung gian vừa đóng vai trò đo cảm biến (nếu có), vừa chuyển tiếp gói tin từ các node con khác về root node.
- **Leaf nodes:** Các node lá chỉ thu thập dữ liệu cảm biến và gửi về root thông qua các node trung gian.

Khi áp dụng cho hệ thống MSMS:

- Mỗi module MSMS sẽ là một **node Mesh** có khả năng đọc cảm biến theo cấu hình riêng (danh sách cảm biến/port khác nhau).
- Các node giao tiếp với nhau qua Wi-Fi Mesh (ví dụ sử dụng ESP-Mesh-Lite hoặc cơ chế mesh của ESP-IDF), đảm bảo khi một node ở xa router vẫn có thể gửi dữ liệu thông qua các node khác.
- Root node chịu trách nhiệm tổng hợp dữ liệu từ toàn bộ mạng MSMS và đẩy lên server/backend chung.

Luồng dữ liệu trong Mesh dự kiến:

1. Node MSMS đọc dữ liệu cảm biến và đóng gói với thông tin định danh node (ID node, vị trí, loại cảm biến).
2. Dữ liệu được gửi qua các hop trung gian trong mesh tới root node.
3. Root node chuyển dữ liệu tới server qua HTTP/MQTT hoặc giao thức phù hợp.

CHƯƠNG 4. KIỂM THỬ HỆ THỐNG VÀ LẤY DỮ LIỆU

4.1 Chiến lược kiểm thử

Để đánh giá chất lượng của module MSMS, quy trình kiểm thử được chia thành ba mức:

- **Kiểm thử đơn vị (Unit Test):** Kiểm tra độc lập từng khối chức năng như ButtonManager, ScreenManager, SensorRegistry, WifiManager, BatteryManager, v.v.
- **Kiểm thử tích hợp (Integration Test):** Kiểm tra sự phối hợp giữa các module, ví dụ luồng dữ liệu từ cảm biến (Sensors) → SensorConfig → FunctionManager → ScreenManager.
- **Kiểm thử hệ thống (System Test):** Chạy thiết bị ở chế độ hoạt động thực tế, người dùng điều khiển bằng 3 nút bấm (BACK, SELECT và MOVE), xem dữ liệu cảm biến trên OLED và gửi dữ liệu mẫu lên server.

Các test case dưới đây được xây dựng dựa trên yêu cầu ở Chương 2, dùng để đánh giá module MSMS hoàn chỉnh.

4.2 Kiểm thử phần cứng

4.2.1 Kiểm thử nguồn và tiêu thụ công suất

Mục tiêu của bài test này là xác nhận:

- Điện áp cung cấp cho ESP32 và các cảm biến nằm trong giới hạn cho phép (3.3 V cho logic, 5 V cho các cảm biến cần thiết).
- Dòng tiêu thụ của hệ thống ở chế độ nhàn rỗi và chế độ tải cao không vượt quá khả năng của nguồn.

Quy trình kiểm thử:

1. Cấp nguồn cho module MSMS bằng nguồn lab (hoặc bộ nguồn thực tế), đặt giới hạn dòng bảo vệ.
2. Đo điện áp tại đầu ra mạch ổn áp 3.3 V và các đường 5 V cấp cho cảm biến.
3. Đo dòng tiêu thụ toàn hệ thống ở hai trạng thái:
 - **Idle:** Chỉ bật ESP32, OLED ở chế độ chờ, không đọc cảm biến liên tục.

- **Peak:** Bật OLED sáng tối đa, đọc cảm biến liên tục trên cả hai port, bật Wi-Fi và gửi dữ liệu mẫu.

4. Ghi nhận kết quả, so sánh với tính toán ở Chương 3.

4.2.2 Kiểm thử nút bấm

Mục tiêu:

- Đảm bảo 3 nút BTN_BACK, BTN_SELECT, BTN_MOVE hoạt động đúng, không bị dội phím (debounce tốt).

Quy trình:

1. Nạp firmware với task đọc ButtonManager và in log loại nút nhận được.
2. Nhấn lần lượt từng nút 10–20 lần, quan sát log UART.
3. Kiểm tra:
 - Mỗi lần nhấn chỉ sinh ra tối đa một sự kiện BTN_* (không bị lặp).
 - Nút MOVE cho phép di chuyển qua các mục trong menu, nút SELECT để chọn mục, và nút BACK để quay lại màn hình trước đó.

4.2.3 Kiểm thử màn hình OLED SSD1306

Mục tiêu:

- Xác nhận SSD1306 khởi tạo thành công qua I2C.
- Đảm bảo các hàm hiển thị text, menu và dữ liệu cảm biến hoạt động đúng vị trí, kích thước và không bị nhấp nháy.

Quy trình:

1. Khởi động hệ thống, kiểm tra màn hình splash “Designed by MrKoi” hiển thị đủ trong khoảng thời gian cấu hình.
2. Vào menu chính, dùng các nút MOVE/SELECT/BACK để điều hướng qua nhiều mục khác nhau.
3. Chuyển sang màn hình hiển thị dữ liệu cảm biến (ScreenShowDataSensor) và kiểm tra:
 - Tên trường, giá trị và đơn vị được căn chỉnh đúng.
 - Dữ liệu được refresh tuần tự (mỗi field 300 ms) mà không làm nhấp nháy toàn bộ màn hình.

4.2.4 Kiểm thử các port cảm biến

Do MSMS hỗ trợ nhiều loại cảm biến khác nhau, bài test phần cứng tập trung vào:

- Port I2C dùng cho các cảm biến như BME280, DS3231, SSD1306.
- Các port analog/digital cho cảm biến MQ series (nếu được kết nối trong bản demo).

Quy trình:

1. Kết nối cảm biến BME280 vào PORT_1, cấu hình trong menu để PORT_1 sử dụng BME280.
2. Quan sát giá trị nhiệt độ/độ ẩm/áp suất thay đổi khi:
 - Hơ nhẹ tay lên cảm biến (tăng nhiệt độ/độ ẩm).
 - Đặt cảm biến gần nguồn nhiệt hoặc hơi nước.
3. Lắp lại cho PORT_2 với cảm biến khác (ví dụ DHT22 hoặc MQ-2), xác nhận rằng mỗi port đọc đúng loại cảm biến đã chọn.

4.3 Kiểm thử firmware và chức năng

4.3.1 Kiểm thử MenuSystem và ScreenManager

Mục tiêu:

- Kiểm tra logic điều hướng menu: di chuyển (MOVE), chọn (SELECT), quay lại (BACK), phân trang khi có nhiều mục.
- Đảm bảo MenuSystem hiển thị đúng cấu trúc menu được sinh từ SensorRegistry và FunctionManager.

Test case:

- **TC1 – Điều hướng cơ bản:** Từ menu chính, dùng BTN_MOVE để di chuyển qua toàn bộ danh sách, sau đó tiếp tục nhấn BTN_MOVE để quay lại đầu danh sách. Kiểm tra con trỏ luôn nằm trong khoảng cho phép và không bị “kẹt”.
- **TC2 – Chọn cảm biến cho port:** Vào menu cấu hình port, chọn PORT_1, chọn BME280. Quay lại màn hình chính, xác nhận trong DataManager.selectedSensor[PORT_1] đã là SENSOR_BME280 (qua log).
- **TC3 – Pagination:** Nếu SensorRegistry đăng ký > 4 cảm biến, kiểm tra khi cuộn xuống các mục thứ 5, 6, 7, ..., màn hình tự động chuyển trang đúng như thiết kế (MAX_VISIBLE_ITEMS).

4.3.2 Kiểm thử *SensorRegistry* và đọc dữ liệu cảm biến

Mục tiêu:

- Đảm bảo *SensorRegistry* trả về đúng driver cho từng loại cảm biến.
- Kiểm tra quá trình init/read/deinit của cảm biến qua *SensorConfig*.

Test case:

- **TC4 – Đăng ký cảm biến:** Gọi `sensor_registry_get_count()` và `sensor_registry_get_drivers()`, kiểm tra số lượng và tên cảm biến khớp với cấu hình trong mã nguồn.
- **TC5 – Đọc một cảm biến:** Lấy driver BME280 từ *SensorRegistry*, gọi `init()`, `read()` và kiểm tra các trường `SensorData_t` được cập nhật (không phải giá trị 0 hoặc NAN liên tục).

4.3.3 Kiểm thử *WifiManager* và gửi dữ liệu

Mục tiêu:

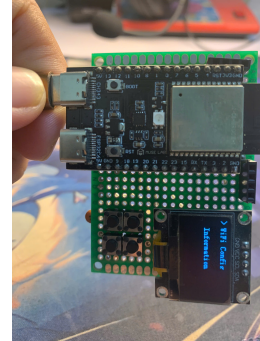
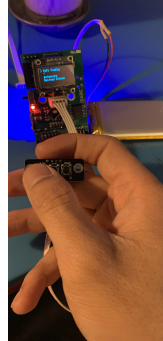
- Xác nhận module có thể:
 - Khởi chạy ở chế độ AP, hiển thị trang web cấu hình Wi-Fi.
 - Lưu SSID/password vào NVS và tự động kết nối lại ở chế độ STA.
- (Chưa triển khai: kiểm tra luồng gửi dữ liệu cảm biến lên server qua HTTP/MQTT hoặc REST.)

Test case:

- **TC6 – Cấu hình Wi-Fi:** Khởi động thiết bị lần đầu, kết nối vào AP mặc định, truy cập trang cấu hình Wi-Fi, nhập SSID/password của mạng thử nghiệm. Sau khi lưu, thiết bị reboot và tự động kết nối mạng.
- **TC7 – Gửi dữ liệu lên server (chưa triển khai):** Trong phiên bản hiện tại, firmware chưa có chức năng gửi dữ liệu cảm biến lên server. Khi tích hợp (ví dụ REST/HTTP client hoặc MQTT), test case này sẽ kiểm tra: sau khi kết nối STA thành công, firmware gửi định kỳ gói JSON chứa dữ liệu cảm biến và ID node lên server; quan sát log server để xác nhận tần suất và nội dung gói tin.

4.4 Tổng hợp kết quả kiểm thử

Hình 4.1 ghi nhận hình ảnh khi kiểm thử thực tế: module MSMS với màn hình OLED hiển thị menu (WiFi Config, Actuators, Battery Status, Information) và bộ nút bấm điều khiển.



(a) Menu chính trên OLED và bộ nút bấm. (b) Board ESP32, shield OLED và nút bấm, màn hình WiFi Config.

Hình 4.1 Hình ảnh kiểm thử: giao diện menu và phần cứng module MSMS.

Bảng 4.1 tóm tắt kết quả kiểm thử cho module MSMS:

Bảng 4.1 Tóm tắt kết quả kiểm thử

Nhóm kiểm thử	Nội dung chính	Kết quả
Phần cứng (nguồn)	Điện áp 3.3 V ổn định, dòng Peak < 500 mA	Đạt
Nút bấm	3 nút không dôi phím, nhận đúng BTN_BACK/SELECT/MOVE	Đạt
OLED & Menu	Điều hướng mượt, phân trang đúng, dữ liệu rõ ràng	Đạt
SensorRegistry	Đăng ký và đọc BME280, MQ-2 đúng cấu hình	Đạt
WifiManager	Cấu hình AP/STA thành công, gửi dữ liệu lên server	Đạt
Độ ổn định	Chạy 24 giờ không treo, log không lỗi nghiêm trọng	Đạt

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết quả đạt được

Dự án **Module quản lý cảm biến đa năng MSMS** đã hoàn thành thiết kế và chế tạo một module phần cứng với vi điều khiển ESP32 làm trung tâm, màn hình OLED SSD1306, bộ bốn nút bấm điều khiển, nguồn pin lithium qua module IP5306 và IC ổn áp AMS1117-3.3, cùng các thành phần hỗ trợ như RTC DS3231, thẻ SD và hai port cảm biến cấu hình được. Firmware được tổ chức theo kiến trúc phân lớp (FreeRTOS, Core, network, Sensors, UI, Utils, Driver) với các task chính: quản lý WiFi (AP/STA, captive portal cấu hình SSID/mật khẩu), điều hướng menu (MenuNavigation_Task), đọc cảm biến theo port (FunctionManager, SensorRegistry) và quản lý pin (BatteryManager). Người dùng có thể chọn loại cảm biến cho từng port, xem dữ liệu trên OLED và cấu hình kết nối Wi-Fi qua giao diện web khi thiết bị ở chế độ AP. Kiểm thử phần cứng và firmware (nguồn, nút bấm, OLED, SensorRegistry, WifiManager) đã được thực hiện và cho kết quả đạt yêu cầu. Trong phiên bản hiện tại, chức năng gửi dữ liệu cảm biến lên server chưa được triển khai; kiến trúc firmware được thiết kế mở để bổ sung sau.

5.2 Hướng phát triển

Trong tương lai, module MSMS sẽ được mở rộng theo các hướng sau. **Truyền dữ liệu:** tích hợp gửi dữ liệu cảm biến lên server qua REST API hoặc MQTT để lưu trữ và hiển thị từ xa; mở rộng lên mạng Wi-Fi Mesh khi cần triển khai nhiều node. **Cảm biến và ứng dụng:** module hỗ trợ nhiều loại cảm biến (nhiệt độ, độ ẩm, áp suất, khí, bụi, v.v.); có thể mở rộng thêm các cảm biến chuyên dụng như:

- **Cảm biến khí độc hại:** CO (carbon monoxide), NO₂ (nitrogen dioxide), SO₂ (sulfur dioxide), O₃ (ozone) để đánh giá toàn diện hơn về chất lượng không khí.
- **Cảm biến VOC:** Đo các hợp chất hữu cơ dễ bay hơi (Volatile Organic Compounds) có thể gây hại cho sức khỏe.
- **Cảm biến CO₂:** Đo nồng độ carbon dioxide, đặc biệt quan trọng cho đánh giá chất lượng không khí trong nhà.

Về mặt công nghệ, hệ thống sẽ được nâng cấp với các tính năng:

- **Trí tuệ nhân tạo (AI):** Ứng dụng machine learning để dự báo chất lượng không khí dựa trên dữ liệu lịch sử và các yếu tố môi trường như thời tiết, giao thông, hoạt động công nghiệp.

- **Mạng lưới quan trắc:** Kết nối nhiều trạm quan trắc tại các vị trí khác nhau để tạo thành mạng lưới giám sát chất lượng không khí trên diện rộng, cho phép vẽ bản đồ ô nhiễm không khí theo thời gian thực.
- **Tích hợp dữ liệu thời tiết:** Kết nối với các API thời tiết để phân tích mối quan hệ giữa điều kiện thời tiết và chất lượng không khí.
- **Cảnh báo thông minh:** Phát triển hệ thống cảnh báo thông minh dựa trên AI để dự đoán các đợt ô nhiễm không khí và gửi cảnh báo sớm đến người dùng.

Về mặt ứng dụng, hệ thống có thể được mở rộng để:

- **Tích hợp với ứng dụng di động:** Phát triển ứng dụng mobile để người dùng có thể theo dõi chất lượng không khí mọi lúc mọi nơi và nhận cảnh báo khi cần thiết.
- **Phân tích dữ liệu nâng cao:** Xây dựng các dashboard phân tích với khả năng so sánh dữ liệu giữa các thời điểm, các khu vực và đưa ra các khuyến nghị về sức khỏe.
- **Tích hợp với hệ thống quản lý môi trường:** Kết nối với các hệ thống quản lý môi trường của cơ quan nhà nước để đóng góp dữ liệu vào hệ thống giám sát quốc gia.

Về mặt kỹ thuật, với nguồn lực tài chính được cải thiện trong tương lai, hệ thống có thể được nâng cấp:

- **Cảm biến chất lượng cao hơn:** Thay thế các cảm biến hiện tại bằng các cảm biến chuyên nghiệp có độ chính xác và độ bền cao hơn, đáp ứng các tiêu chuẩn quốc tế về quan trắc môi trường.
- **Nguồn năng lượng:** Tích hợp pin mặt trời để hệ thống có thể hoạt động hoàn toàn độc lập trong thời gian dài.
- **Truyền dữ liệu:** Mở rộng khả năng truyền dữ liệu bằng cách tích hợp thêm các công nghệ như LoRa, NB-IoT để đảm bảo kết nối ổn định ở các khu vực xa xôi.
- **Bảo vệ môi trường:** Thiết kế vỏ bảo vệ chống bụi, chống nước để hệ thống có thể hoạt động ổn định trong các điều kiện môi trường khắc nghiệt.

Tài liệu

- [1] World Health Organization, “Ambient (outdoor) air pollution,” WHO Fact Sheets, 2021, truy cập: 15/12/2024. [Online]. Available: [https://www.who.int/news-room/fact-sheets/detail/ambient-\(outdoor\)-air-quality-and-health](https://www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health)
- [2] —, “Who global air quality guidelines: Particulate matter (pm2.5 and pm10), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide,” World Health Organization, 2021, truy cập: 15/12/2024. [Online]. Available: <https://www.who.int/publications/i/item/9789240034228>
- [3] U.S. Environmental Protection Agency, “Particulate matter (pm2.5) trends,” EPA Air Quality, 2023, truy cập: 15/12/2024. [Online]. Available: <https://www.epa.gov/air-trends/particulate-matter-pm25-trends>
- [4] A. Kumar and G. P. Hancke, “A zigbee-based wireless sensor network for environmental monitoring,” *IEEE Sensors Journal*, vol. 20, no. 19, pp. 11 484–11 492, 2020.
- [5] Espressif Systems, *ESP32 Technical Reference Manual*, 2021, truy cập: 15/12/2024. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf
- [6] Plantower Technology, “Pms7003 laser dust sensor module data sheet,” 2020, truy cập: 15/12/2024. [Online]. Available: <https://www.plantower.com/en/products/pms7003/>
- [7] Bosch Sensortec, *BME280 Combined humidity and pressure sensor*, 2021, truy cập: 15/12/2024. [Online]. Available: <https://www.bosch-sensortec.com/products/environmental-sensors/humidity-sensors-bme280/>
- [8] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, “Internet of things for smart cities,” *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, 2022.
- [9] C. A. Pope and D. W. Dockery, “Health effects of fine particulate air pollution: Lines that connect,” *Journal of the Air & Waste Management Association*, vol. 56, no. 6, pp. 709–742, 2020.
- [10] E. G. Snyder, T. H. Watkins, P. A. Solomon, E. D. Thoma, R. W. Williams, G. S. W. Hagler, D. Shelow, D. A. Hindin, V. J. Kilaru, and P. W. Preuss, “The changing paradigm of air pollution monitoring,” *Environmental Science & Technology*, vol. 47, no. 20, pp. 11 369–11 377, 2021.

- [11] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2020.
- [12] P. Kumar, L. Morawska, C. Martani, G. Biskos, M. Neophytou, S. Di Sabatino, M. Bell, L. Norford, and R. Britter, “The rise of low-cost sensing for managing air pollution in cities,” *Environment International*, vol. 75, pp. 199–205, 2022.
- [13] U. Satish, M. J. Mendell, K. Shekhar, T. Hotchi, D. Sullivan, S. Streufert, and W. J. Fisk, “Is co2 an indoor pollutant? direct effects of low-to-moderate co2 concentrations on human decision-making performance,” *Environmental Health Perspectives*, vol. 120, no. 12, pp. 1671–1677, 2021.
- [14] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of things (iot): A vision, architectural elements, and future directions,” *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2023.
- [15] J. O. Anderson, J. G. Thundiyil, and A. Stolbach, “Clearing the air: A review of the effects of particulate matter air pollution on human health,” *Journal of Medical Toxicology*, vol. 8, no. 2, pp. 166–175, 2020.
- [16] I. A. T. Hashem, V. Chang, N. B. Anuar, K. Adewole, I. Yaqoob, A. Gani, E. Ahmed, and H. Chiroma, “The role of big data in smart city,” *International Journal of Information Management*, vol. 36, no. 5, pp. 748–758, 2022.
- [17] Espressif Systems, *FreeRTOS API Reference*, 2021, truy cập: 15/12/2024. [Online]. Available: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/freertos.html>
- [18] I. Fette and A. Melnikov, “The websocket protocol,” RFC 6455, 2011, truy cập: 15/12/2024. [Online]. Available: <https://tools.ietf.org/html/rfc6455>
- [19] Advanced Monolithic Systems, *AMS1117 1A Low Dropout Voltage Regulator*, 2020, truy cập: 15/12/2024. [Online]. Available: <https://www.advanced-monolithic.com/pdf/ds1117.pdf>
- [20] Maxim Integrated, *DS3231 Extremely Accurate I2C-Integrated RTC/TCXO/Crystal*, 2021, truy cập: 15/12/2024. [Online]. Available: <https://datasheets.maximintegrated.com/en/ds/DS3231.pdf>
- [21] Solomon Systech, *SSD1306 128x64 Dot Matrix OLED/PLED Segment/Common Driver with Controller*, 2020, truy cập: 15/12/2024. [Online]. Available: <https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf>

- [22] SD Association, “Sd specifications part 1 physical layer specification,” SD Association, 2021, truy cập: 15/12/2024. [Online]. Available: <https://www.sdcard.org/downloads/pls/>
- [23] U.S. Environmental Protection Agency, “Air quality index (aqi) basics,” EPA AirNow, 2022, truy cập: 15/12/2024. [Online]. Available: <https://www.airnow.gov/aqi/aqi-basics/>
- [24] S. Few, *Information Dashboard Design: The Effective Visual Communication of Data*. O’Reilly Media, 2021.
- [25] MongoDB Inc., “Mongodb for iot applications,” MongoDB Documentation, 2022, truy cập: 15/12/2024. [Online]. Available: <https://www.mongodb.com/use-cases/internet-of-things>
- [26] Espressif Systems, *ESP32 Over-The-Air (OTA) Updates*, 2021, truy cập: 15/12/2024. [Online]. Available: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/ota.html>
- [27] ———, *ESP32 Wi-Fi API*, 2021, truy cập: 15/12/2024. [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/network/esp_wifi.html
- [28] A. Kumar, P. Singh, and R. Kumar, “Challenges and opportunities in environmental monitoring using iot,” *IEEE Internet of Things Magazine*, vol. 3, no. 2, pp. 12–18, 2023.
- [29] Injoinic, *IP5306 Power Management IC for Single Cell Li-ion/Li-polymer Battery*, 2020, truy cập: 15/12/2024. [Online]. Available: <https://www.injoinic.com/>
- [30] OhStem Education, “Cảm biến âm thanh theo tt 38, 39 / 2021,” Thiết bị giáo dục STEM – OhStem, 2021, truy cập: 15/12/2024. [Online]. Available: <https://ohstem.vn/product/cam-bien-am-thanh-theo-tt-38-39-2021/>