TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

VÀ TRUYỀN THÔNG VIỆT – HÀN

**KHOA KỸ THUẬT MÁY TÍNH & ĐIỆN TỬ**



**BÁO CÁO**

**ĐỒ ÁN CHUYÊN NGÀNH III**

***Đề Tài*:** **XÂY DỰNG HỆ THỐNG IOT GIÁM SÁT VÀ ĐIỀU KHIỂN THIẾT BỊ THEO NHIỆT ĐỘ MÔI TRƯỜNG**

**Sinh viên thực hiện     : Thái Văn Hòa**

**Lớp : 21IR**

**Niên khoá : 2021 - 2026**

**Giảng viên hướng dẫn : TS.Dương Hữu Ái**

***Đà Nẵng, ngày 10 tháng 05 năm 2025***

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

VÀ TRUYỀN THÔNG VIỆT – HÀN

**KHOA KỸ THUẬT MÁY TÍNH & ĐIỆN TỬ**



**BÁO CÁO**

**ĐỒ ÁN CHUYÊN NGÀNH III**

***Đề Tài*:** **XÂY DỰNG HỆ THỐNG IOT GIÁM SÁT VÀ ĐIỀU KHIỂN THIẾT BỊ THEO NHIỆT ĐỘ MÔI TRƯỜNG**

**Sinh viên thực hiện     : Thái Văn Hòa**

**Lớp : 21IR**

**Niên khoá : 2021 - 2026**

**Giảng viên hướng dẫn : TS.Dương Hữu Ái**

***Đà Nẵng, ngày 10 tháng 05 năm 2025***

# NHẬN XÉT

**(Của Giảng viên hướng dẫn)**

....................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

Đà Nẵng,….. tháng … năm 2025

Giảng viên hướng dẫn

# LỜI CẢM ƠN

Đầu tiên, em xin gửi lời cảm ơn đến Trường Đại học Công Nghệ Thông Tin Và Truyền Thông Việt – Hàn đã luôn lắng nghe và luôn luôn tạo điều kiện học tập tốt nhất cho em và cũng như toàn thể sinh viên trong trường.

Tiếp đến em xin bày tỏ lòng biết ơn sâu sắc nhất tới giáo viên hướng dẫn TS. Dương Hữu Aí đã tận tình giúp đỡ, tìm ra nhiều lỗi sai của em trong quá trình tìm hiểu và nghiên cứu. Thầy còn hướng dẫn em rất nhiều để hoàn thành xong đồ án chuyên ngành .

Vì thời gian có hạn, trình độ hiểu biết của bản thân còn nhiều hạn chế. Cho nên trong đồ án không tránh khỏi những thiếu sót, em mong nhận được sự đóng góp ý kiến của các thầy cô giáo để đồ án của em được hoàn thiện hơn.

Em xin chân thành cảm ơn !

Đà Nẵng, tháng 05 năm 2025

*Sinh viên thực hiện*

*Thái Văn Hòa*

**MỤC LỤC**

[NHẬN XÉT iii](#_Toc198694733)

[LỜI CẢM ƠN iv](#_Toc198694734)

[MỤC LỤC v](#_Toc198694735)

[DANH MỤC HÌNH ẢNH vii](#_Toc198694736)

[DANH MỤC TỪ VIẾT TẮT viii](#_Toc198694737)

[MỞ ĐẦU ix](#_Toc198694738)

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 1](#_Toc198694739)

[1.1 Thực trạng hiện nay 1](#_Toc198694740)

[1.2 Mục tiêu chọn đề tài 1](#_Toc198694741)

[1.3 Lý do chọn đề tài 1](#_Toc198694742)

[1.4 Chức năng 1](#_Toc198694743)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 2](#_Toc198694744)

[2.1 WIFI 2](#_Toc198694745)

[2.1.1 Giới thiệu 2](#_Toc198694746)

[2.1.2 Công nghệ truyền nhận dữ liệu 2](#_Toc198694747)

[2.1.3 Thành phần của mạng WIFI 5](#_Toc198694748)

[2.1.4 Cấu trúc liên kết 6](#_Toc198694749)

[2.1.6 Cách thức hoạt động 7](#_Toc198694750)

[2.2 Tổng quan về Internet of thing 7](#_Toc198694751)

[2.2.1 Giới thiệu 7](#_Toc198694752)

[2.2.3 Ứng dụng 8](#_Toc198694753)

[2.3 Giao thức MQTT 9](#_Toc198694754)

[2.3.1 Giới thiệu 9](#_Toc198694755)

[2.3.3 Các thành phần của MQTT 9](#_Toc198694756)

[2.3.4 MQTT QoS 10](#_Toc198694757)

[2.3.5 Broker trong IoT 10](#_Toc198694758)

[2.3.6 Ưu điểm 11](#_Toc198694759)

[2.3.7 Bảo mật 11](#_Toc198694760)

[2.4 HTTP 12](#_Toc198694761)

[2.4.1 Giới thiệu 12](#_Toc198694762)

[2.4.2 Các thành phần chính của HTTP 12](#_Toc198694763)

[2.5 Webserver 14](#_Toc198694764)

[2.5.1 Giới thiệu 14](#_Toc198694765)

[2.5.2 Giao tiếp với Webserver 14](#_Toc198694766)

[2.6 FreeRTOS 15](#_Toc198694767)

[2.6.1 Tổng quan về FreeRTOS 15](#_Toc198694768)

[2.6.2 Kiến trúc FreeRTOS 15](#_Toc198694769)

[2.6.3 Lợi ích của FreeRTOS trong phát triển ứng dụng ESP32 16](#_Toc198694770)

[2.7 Node-RED 17](#_Toc198694771)

[2.7.1 Giới thiệu 17](#_Toc198694772)

[2.7.2 Node-RED trong IoT 18](#_Toc198694773)

[2.7.3 Tính năng 19](#_Toc198694774)

[2.8 MySQL 20](#_Toc198694775)

[2.8.1 Giới thiệu 20](#_Toc198694776)

[2.8.2 Các tính năng của MySQL 20](#_Toc198694777)

[2.9 Raspberry Pi 4 21](#_Toc198694778)

[2.9.1 Giới thiệu 21](#_Toc198694779)

[2.9.2 Các loại Raspberry Pi 4 22](#_Toc198694780)

[2.9.3 Tính năng 23](#_Toc198694781)

[2.10 ESP32 24](#_Toc198694782)

[2.10.1 Giới thiệu 24](#_Toc198694783)

[2.11 BMP280+AHT20 25](#_Toc198694784)

[2.11.1 Giới thiệu 25](#_Toc198694785)

[2.11.2 Cấu tạo 25](#_Toc198694786)

[2.11.3 Nguyên lý hoạt động 26](#_Toc198694787)

[2.12 OLED 26](#_Toc198694788)

[2.12.1 Giới thiệu 26](#_Toc198694789)

[2.12.3 Nguyên lý hoạt động 27](#_Toc198694790)

[2.13 Relay 2 chanel 27](#_Toc198694791)

[2.13.1 giới thiệu 27](#_Toc198694792)

[2.13.3 Nguyên Lý hoạt động 28](#_Toc198694793)

[CHƯƠNG 3: THIẾT KẾ HỆ THỐNG 29](#_Toc198694794)

[3.1 Tổng quan về hệ thống 29](#_Toc198694795)

[3.2 Sơ đồ khối chức năng 30](#_Toc198694796)

[3.3 Lưu đồ thuật toán 31](#_Toc198694797)

[3.4 Flow node red broker 40](#_Toc198694798)

[3.5 Phần mềm và ngôn ngữ lập trình 41](#_Toc198694799)

[CHƯƠNG 4 : THIẾT KẾ & ĐI DÂY 42](#_Toc198694800)

[4.1 Sản phẩm hoàn thiện 42](#_Toc198694801)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 44](#_Toc198694802)

[TÀI LIỆU THAM KHẢO 45](#_Toc198694803)

# DANH MỤC HÌNH ẢNH

[Hình 2. 1 Biểu tượng WIFI 2](#_Toc198694618)

[Hình 2. 2 Cấu trúc liên kết ngang hàng 6](#_Toc198694619)

[Hình 2. 3 Cấu trúc liên kết dựa trên AP 6](#_Toc198694620)

[Hình 2. 4 Cách thức hoạt động của mạng WIFI 7](#_Toc198694621)

[Hình 2. 5 Internet of Things 8](#_Toc198694622)

[Hình 2. 6 Giao thức MQTT 9](#_Toc198694623)

[Hình 2. 7 Hoạt động của client 10](#_Toc198694624)

[Hình 2. 8 Thành phần của một MQTT trong hệ thống IoT 11](#_Toc198694625)

[Hình 2. 9 Giao thức HTTP 12](#_Toc198694626)

[Hình 2. 10 Cấu trúc HTTP Request 13](#_Toc198694627)

[Hình 2. 11 Phương thức Request 13](#_Toc198694628)

[Hình 2. 12 Giao tiếp với Webserver 14](#_Toc198694629)

[Hình 2. 13 FreeRTOS 15](#_Toc198694630)

[Hình 2. 14 Kiến trúc FreeRTOS 16](#_Toc198694631)

[Hình 2. 15 Node trong Node – RED cơ bản 18](#_Toc198694632)

[Hình 2. 16 Node - RED và IoT 19](#_Toc198694633)

[Hình 2. 17 Giao diện Node - RED 19](#_Toc198694634)

[Hình 2.18Cơ sở dữ liệu MySQL 20](#_Toc198694635)

[Hình 2. 19 Raspberry Pi 4 22](#_Toc198694636)

[Hình 2. 20 Module ESP32-WROOM-32D 24](#_Toc198694637)

[Hình 2. 21 Module cảm biến BMP280+AHT20 25](#_Toc198694638)

[Hình 2. 22 Màn hình OLED 27](#_Toc198694639)

[Hình 2. 23 Relay 2 Chanel 28](#_Toc198694640)

[Hình 2. 24 Nguyên lý hoạt động của relay 28](#_Toc198694641)

[Hình 3. 1 Sơ đồ khối tổng quát 30](#_Toc198694642)

[Hình 3. 2 Sơ đồ khối Node 31](#_Toc198694643)

[Hình 3. 3Lưu đồ chương trình chính 31](#_Toc198694644)

[Hình 3. 4 Lưu đồ chương trình StartAPMode 32](#_Toc198694645)

[Hình 3. 5 Lưu đồ chương trình TaskControl 33](#_Toc198694646)

[Hình 3. 6 Lưu đồ chương trình TaskDisplay 34](#_Toc198694647)

[Hình 3. 7 Lưu đồ chương trình TaskMQTT 35](#_Toc198694648)

[Hình 3. 8 Lưu đồ chương trình TaskMQTTSubcribe 36](#_Toc198694649)

[Hình 3. 9 Lưu đồ chương trình TaskPrintData 37](#_Toc198694650)

[Hình 3. 10 Lưu đồ chương trình TaskReadSensor 38](#_Toc198694651)

[Hình 3. 11 Lưu đồ chương trình TaskResetTimeout 39](#_Toc198694652)

[Hình 3. 12 Flow View Dashboard Node - RED 40](#_Toc198694653)

[Hình 3. 13 Flow Control Dashboard Node - RED 40](#_Toc198694654)

[Hình 3. 14 Flow Save database MySQL 41](#_Toc198694655)

[Hình 3. 15 Arduino IDE 41](#_Toc198694656)

[Hình 4. 1 đi dây hệ thống 42](#_Toc198694657)

[Hình 4. 2Webserver Node-red 42](#_Toc198694658)

[Hình 4. 3 biểu đồ node-red 43](#_Toc198694659)

[Hình 4. 4 Web server 43](#_Toc198694660)

[Hình 4. 5 biểu dồ nhiệt độ độ ẩm 43](#_Toc198694661)

# DANH MỤC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **STT** | Từ viết tắt | Từ đầy đủ |
| 1 | IoT | Internet of Things |
| 2 | WiFi | Wireless Fidelity |
| 3 | IEEE | Institute of Electrical and Electronic Engineers |
| 4 | LAN | Local Area Network |
| 5 | MIMO | Multiple Input Multiple Output |

MỞ ĐẦU

**Giới thiệu đề tài**

Hiện nay, chất lượng không khí ngày càng được chú trọng, đặc biệt trong các khu vực nhạy cảm như phòng bệnh viện, nơi yêu cầu một môi trường sạch sẽ và an toàn để đảm bảo sức khỏe cho người dùng . Tình trạng ô nhiễm không khí không chỉ tồn tại ở không gian công cộng mà còn len lỏi vào trong các không gian kín như phòng bệnh, gây ảnh hưởng tiêu cực đến quá trình hồi phục của người dùng và làm tăng nguy cơ lây nhiễm chéo.

Chúng em đã chọn đề tài “HỆ THỐNG IOT GIÁM SÁT NHIỆT DỘ ĐỘ ẨM ÁP XUẤT MÔI TRƯỜNG ” để nghiên cứu và phát triển một mô hình giám sát thông minh, sử dụng các công nghệ IoT tiên tiến, với mục tiêu tối ưu hóa chất lượng không khí trong phòng bệnh. Đề tài này nhằm góp phần giải quyết vấn đề ô nhiễm không khí trong không gian kín, cải thiện điều kiện y tế và nâng cao chất lượng cuộc sống cho người dùng .

**Lý do lựa chọn đề tài**

Ứng dụng công nghệ IoT: Với sự phát triển của Internet of Things, việc tích hợp các cảm biến và hệ thống tự động vào giám sát không khí mang lại hiệu quả cao hơn. Hệ thống có thể theo dõi, phân tích và đưa ra các giải pháp kịp thời, giúp tiết kiệm thời gian và công sức so với các phương pháp giám sát truyền thống.

Bảo vệ sức khỏe và môi trường: Hệ thống không chỉ hỗ trợ bảo vệ sức khỏe cho người dùng mà còn giúp giảm thiểu tác động tiêu cực lên môi trường bằng cách quản lý hiệu quả các thiết bị điều hòa không khí, máy lọc không khí.

Tăng cường kiến thức và kỹ năng: Đề tài này mang tính ứng dụng cao và yêu cầu sự kết hợp kiến thức về cảm biến, điện tử, lập trình, và phân tích dữ liệu. Đây là cơ hội để chúng em phát triển kỹ năng trong lĩnh vực IoT, từ đó mở rộng khả năng ứng dụng công nghệ vào các lĩnh vực khác.

Thực tiễn và tiềm năng phát triển: Các bệnh viện và cơ sở y tế ngày càng chú trọng cải thiện môi trường không khí, tạo ra tiềm năng phát triển cho các hệ thống giám sát thông minh. Đề tài này không chỉ giải quyết vấn đề hiện tại mà còn có thể mở rộng để áp dụng trong các khu vực khác như trường học, văn phòng, hoặc nhà ở.

**Mục tiêu nghiên cứu**

Phát triển hệ thống giám sát chất lượng không khí thông minh: Nghiên cứu và phát triển hệ thống IoT giám sát chất lượng không khí trong phòng bệnh, tích hợp các công nghệ cảm biến tiên tiến và kết nối mạng để cung cấp dữ liệu thời gian thực.

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

### 1.1 Thực trạng hiện nay

Chất lượng không khí trong các cơ sở y tế, đặc biệt là phòng bệnh, đóng vai trò quan trọng trong việc bảo vệ sức khỏe người dùng và hỗ trợ quá trình hồi phục. Tuy nhiên, không gian kín như phòng bệnh thường gặp phải tình trạng ô nhiễm không khí, bao gồm bụi mịn (PM2.5, PM10), khí độc (CO, CO₂), và vi khuẩn, dẫn đến nguy cơ lây nhiễm và ảnh hưởng xấu đến sức khỏe người bệnh. Mặc dù có các biện pháp phòng ngừa như máy lọc không khí và điều hòa, nhưng các phương pháp giám sát chất lượng không khí hiện nay vẫn chưa đáp ứng được yêu cầu về độ chính xác, kịp thời và tự động hóa. Việc phát triển các hệ thống giám sát thông minh, sử dụng công nghệ IoT, là giải pháp khả thi để cải thiện vấn đề này.

### 1.2 Mục tiêu chọn đề tài

Mục tiêu của đề tài là xây dựng một , sử dụng công nghệ IoT để thu thập và phân tích dữ liệu môi trường. Hệ thống này sẽ cung cấp thông tin về các yếu tố như, nhiệt độ và độ ẩm, từ đó đưa ra các giải pháp tự động điều chỉnh nhằm đảm bảo không khí trong phòng bệnh luôn sạch và an toàn cho người dùng . Ngoài ra, hệ thống cũng hỗ trợ giám sát và điều khiển từ xa, giúp tiết kiệm thời gian và tăng cường hiệu quả quản lý..

### 1.3 Lý do chọn đề tài

Việc lựa chọn đề tài "HỆ THỐNG IOT GIÁM SÁT NHIỆT DỘ ĐỘ ẨM ÁP XUẤT MÔI TRƯỜNG " xuất phát từ nhu cầu cấp thiết trong việc cải thiện chất lượng không khí trong bệnh viện, nơi yêu cầu một môi trường trong lành và an toàn để phục vụ người dùng . Chúng em chọn công nghệ IoT vì khả năng theo dõi và quản lý môi trường thời gian thực, mang lại giải pháp hiệu quả hơn so với các phương pháp truyền thống. Hệ thống này không chỉ giúp bảo vệ sức khỏe người dùng mà còn giảm thiểu tác động tiêu cực đến môi trường, tối ưu hóa việc sử dụng các thiết bị như máy lọc không khí và điều hòa, đồng thời nâng cao chất lượng dịch vụ y tế.

### 1.4 Chức năng

Hệ thống giám sát chất lượng không khí trong phòng bệnh sẽ bao gồm các chức năng chính sau:

* Thu thập và giám sát dữ liệu môi trường: Cảm biến đo các thông số như nhiệt độ, độ ẩm.
* Phân tích dữ liệu: Dữ liệu thu thập được sẽ được phân tích để đưa ra cảnh báo kịp thời nếu các thông số vượt ngưỡng an toàn.
* Tự động hóa điều khiển: Hệ thống có thể tự động điều chỉnh các thiết bị như máy lọc không khí, điều hòa để duy trì môi trường tối ưu.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1 WIFI

2.1.1 Giới thiệu

Wi-Fi (viết tắt của Wireless Fidelity) là một công nghệ không dây cho phép các thiết bị kết nối internet và giao tiếp với nhau thông qua sóng radio. Wi-Fi hoạt động dựa trên chuẩn IEEE 802.11 và thường được sử dụng trong các mạng LAN (Local Area Network) để cung cấp kết nối internet không dây cho các thiết bị như máy tính, điện thoại, máy tính bảng, và các thiết bị IoT.

Wi-Fi sử dụng tần số sóng vô tuyến (2.4 GHz và 5 GHz) để truyền tải dữ liệu giữa các thiết bị và điểm truy cập (access point). Các điểm truy cập này kết nối với một mạng có dây, chẳng hạn như mạng internet, và phát sóng tín hiệu Wi-Fi cho các thiết bị trong khu vực phủ sóng.

Wi-Fi được sử dụng rộng rãi trong các không gian công cộng như quán cà phê, sân bay, và các văn phòng, cũng như trong các hộ gia đình để cung cấp kết nối internet cho nhiều thiết bị cùng lúc mà không cần dây cáp.



Hình 2. 1 Biểu tượng WIFI

2.1.2 Công nghệ truyền nhận dữ liệu

IEEE 802.11b (1999):

* Tần số: 2.4GHz.
* Tốc độ: 11Mbps (lý thuyết), thực tế khoảng 4-6Mbps.
* Ưu điểm: Tương đối rẻ, phù hợp cho các ứng dụng cơ bản.
* Nhược điểm: Phạm vi tín hiệu bị hạn chế, dễ bị nhiễu từ các thiết bị như Bluetooth, điện thoại di động, và các mạng không dây khác (ví dụ: lò vi sóng).
* Khả năng mở rộng: Không cao.

IEEE 802.11a (2001):

* Tần số: 5.0GHz.
* Tốc độ: 54Mbps (lý thuyết), thực tế khoảng 15-20Mbps.
* Ưu điểm: Có thể hỗ trợ băng thông lớn hơn và ít bị nhiễu hơn vì hoạt động ở dải tần 5.0GHz.
* Nhược điểm: Phạm vi tín hiệu ngắn hơn và khả năng xuyên qua các vật cản kém hơn so với 802.11b (do tần số cao hơn).
* Khả năng mở rộng: Tốt hơn 802.11b.
* Không tương thích với 802.11b: Vì khác nhau về tần số và phương pháp truyền tín hiệu.

IEEE 802.11g (2003):

* Tần số: 2.4GHz.
* Tốc độ: 54Mbps (lý thuyết), tương thích với chuẩn 802.11b và có hiệu suất truyền cao hơn.
* Ưu điểm: Tương thích ngược với 802.11b, cung cấp tốc độ truyền cao hơn (tương đương với 802.11a) và không dễ bị nhiễu như 802.11b.
* Nhược điểm: Bị nhiễu từ các thiết bị khác ở tần số 2.4GHz, chẳng hạn như Bluetooth.
* Khả năng mở rộng: Tốt hơn 802.11b, nhưng vẫn bị giới hạn bởi tần số 2.4GHz.

IEEE 802.11n (2009)

* Tần số: 2.4GHz và 5GHz (dual-band).
* Tốc độ: Lý thuyết đạt tối đa 600Mbps (thực tế khoảng 150-300Mbps tùy vào số lượng anten).
* Ưu điểm:
* MIMO (Multiple Input Multiple Output): Hỗ trợ nhiều anten, giúp cải thiện tốc độ và phạm vi tín hiệu.
* Tăng cường hiệu suất: Cải thiện tốc độ và độ ổn định so với các chuẩn trước.
* Dual-band: Có thể hoạt động trên cả tần số 2.4GHz và 5GHz, giúp giảm tắc nghẽn trên băng tần 2.4GHz.
* Nhược điểm: Dù tốc độ lý thuyết cao, nhưng phạm vi tín hiệu vẫn bị hạn chế bởi các yếu tố môi trường.
* Khả năng mở rộng: Tốt hơn các chuẩn trước (802.11b/g).

IEEE 802.11ac (2013)

* Tần số: 5.0GHz.
* Tốc độ: Lý thuyết đạt tối đa 1.3Gbps (thực tế khoảng 500-900Mbps, tùy vào thiết bị và môi trường).
* Ưu điểm:
* Tốc độ cao: Cải thiện tốc độ đáng kể so với 802.11n nhờ vào việc sử dụng Wide Channels (80-160MHz) và QAM (Quadrature Amplitude Modulation).
* MU-MIMO (Multi-User MIMO): Cho phép truyền dữ liệu đến nhiều thiết bị cùng lúc mà không làm giảm hiệu suất.
* Tương thích ngược: Hỗ trợ các thiết bị 802.11n và 802.11a.
* Nhược điểm: Dưới 5GHz, tín hiệu có thể dễ bị suy giảm trong môi trường có vật cản dày đặc.
* Khả năng mở rộng: Tốt, nhưng vẫn giới hạn trong phạm vi của tần số 5.0GHz.

IEEE 802.11ax (Wi-Fi 6) (2019)

* Tần số: 2.4GHz và 5GHz (với Wi-Fi 6E hỗ trợ thêm tần số 6GHz).
* Tốc độ: Lý thuyết đạt tối đa 9.6Gbps (thực tế khoảng 1.5-2Gbps).
* Ưu điểm:
* OFDMA (Orthogonal Frequency-Division Multiple Access): Cho phép chia sẻ băng thông hiệu quả hơn giữa nhiều thiết bị, giảm độ trễ và cải thiện hiệu suất mạng khi có nhiều thiết bị kết nối.
* MU-MIMO nâng cao: Hỗ trợ không chỉ trong tải lên mà còn trong tải xuống, giúp nhiều thiết bị có thể truyền nhận đồng thời mà không làm giảm hiệu suất.
* Tăng hiệu suất trong môi trường đông đúc: Wi-Fi 6 tối ưu hóa hiệu suất trong các môi trường có nhiều thiết bị như nhà ở, văn phòng, sân bay.
* Băng tần 6GHz (Wi-Fi 6E): Mở rộng thêm băng tần, giúp giảm tắc nghẽn và tăng khả năng truyền tải dữ liệu.
* Tối ưu hóa hiệu suất và tiết kiệm năng lượng: Sử dụng tính năng Target Wake Time (TWT) giúp các thiết bị tiết kiệm năng lượng khi không sử dụng, rất hữu ích cho các thiết bị IoT.
* Nhược điểm: Cần thiết bị tương thích với Wi-Fi 6 để tận dụng các tính năng mới. Mặc dù hiệu suất cao, nhưng cũng có thể bị giảm tốc trong môi trường nhiễu sóng hoặc khi sử dụng các thiết bị cũ không hỗ trợ Wi-Fi 6.
* Khả năng mở rộng: Rất tốt, phù hợp cho mạng đông đúc, nhiều thiết bị, và môi trường sử dụng các công nghệ mới.

2.1.3 Thành phần của mạng WIFI

Access Point (AP): AP là bộ thu phát không dây LAN (Local Area Network), hoặc là trạm cơ sở có thể kết nối đồng thời một hoặc nhiều thiết bị không dây với Internet.

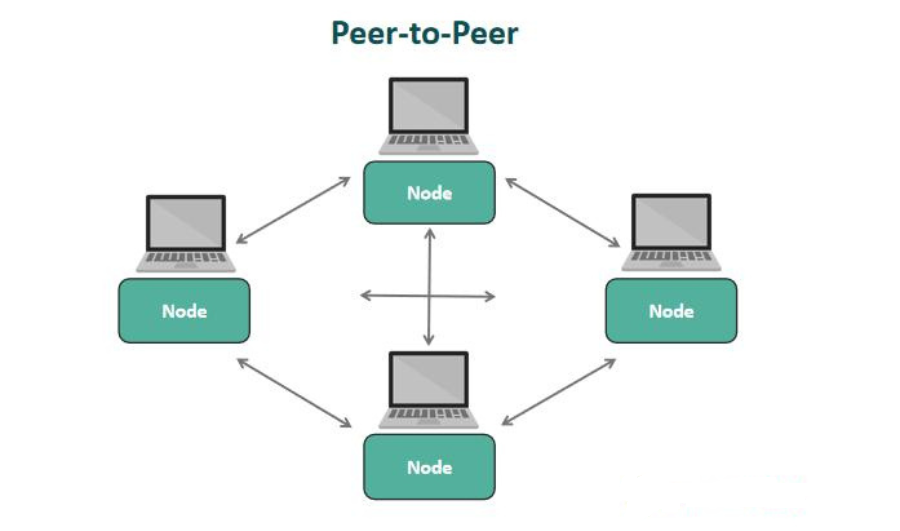
Wifi Card: Cho phép chấp nhận tín hiệu không dây và thông tin chuyển tiếp.

Safeguards: tường lửa và phần mềm chống virus giúp giữ an toàn thông tin cho người dùng.

2.1.4 Cấu trúc liên kết

Peer to peer

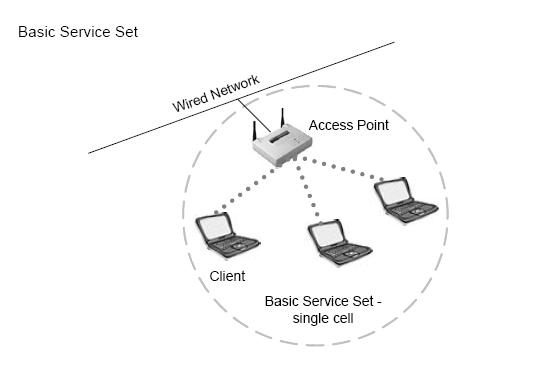
* Là cấu trúc liên kết ngang hàng, không bắt buộc AP, các thiết bị bên trong có thể giao tiếp trực tiếp với nhau.
* Phù hợp để thiết lập mạng không dây một cách nhanh chóng và dễ dàng.



Hình 2. 2 Cấu trúc liên kết ngang hàng

Infrastructure Mode

* Là cấu trúc liên kết dựa trên AP, liên lạc với nhau qua địa điểm truy cập (Access Point). Bất kỳ thông tin truy cập nào đều phải thông qua AP.
* Nếu một trạm di động như máy tính hoặc điện thoại muốn giao tiếp với một trạm di động khác đầu tiên cần phải gửi thông tin đến AP, sau đó AP sẽ gửi ngược lại trạm di động đó.



Hình 2. 3 Cấu trúc liên kết dựa trên AP

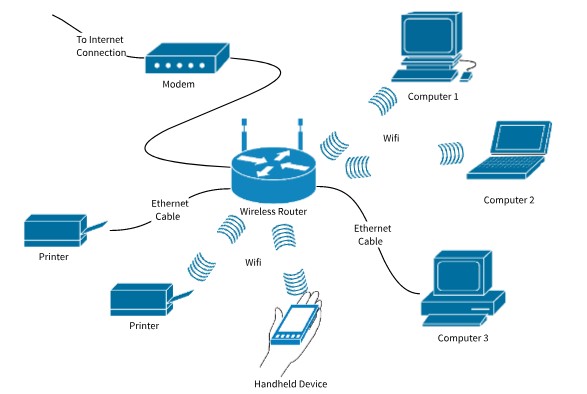
2.1.6 Cách thức hoạt động

Một Wifi Hotspot được tạo ra bằng cách cài đặt điểm truy cập vào kết nối Internet.

Một điểm truy cập hoạt động như một trạm cơ sở.

Khi thiết bị hỗ trợ Wifi bắt gặp điểm phát sóng, thiết bị có thể kết nối không dây với mạng đó.

Một điểm truy cập duy nhất có thể hổ trợ tối đa đến 30 người dùng, nhiều điểm truy cập có thể được kết nối với nhau thông qua cáp Ethernet để tạo ra một mạng lớn.



Hình 2. 4 Cách thức hoạt động của mạng WIFI

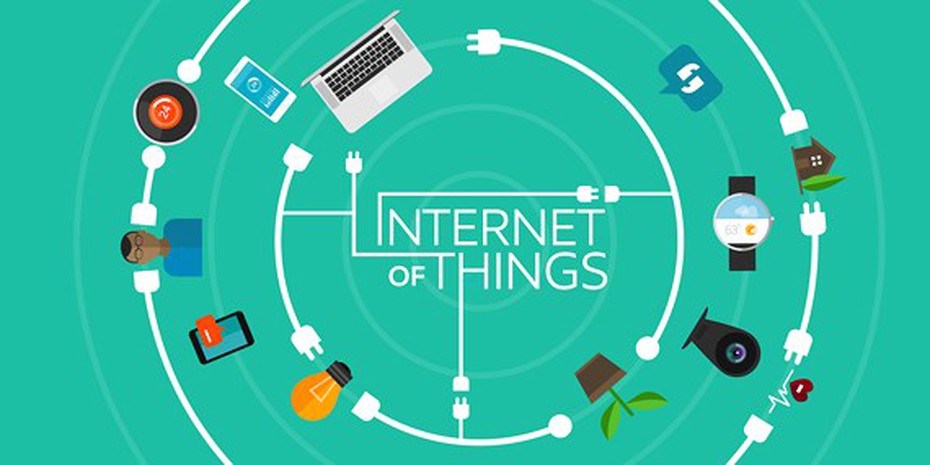
2.2 Tổng quan về Internet of thing

2.2.1 Giới thiệu

Internet of Things (IoT) là một khái niệm mô tả một mạng lưới các thiết bị vật lý được kết nối với Internet, cho phép chúng thu thập, chia sẻ và xử lý dữ liệu một cách tự động. Các thiết bị này có thể bao gồm cảm biến, thiết bị gia dụng, phương tiện giao thông, hệ thống công nghiệp, và nhiều đối tượng khác được tích hợp phần cứng, phần mềm, và kết nối mạng để thực hiện các nhiệm vụ thông minh.

IoT mang đến khả năng liên kết giữa thế giới thực và thế giới số, giúp con người không chỉ giám sát mà còn điều khiển các thiết bị từ xa thông qua Internet. Các thiết bị IoT hoạt động dựa trên sự kết hợp của cảm biến, bộ vi điều khiển, và các công nghệ kết nối như Wi-Fi, Bluetooth, Zigbee, LoRaWAN, hay mạng di động (3G, 4G, 5G).

Các thành phần chính trong một hệ thống IotT: Với một hệ thống IoT chúng sẽ bao gồm 4 thành phần chính đó là thiết bị (Things), trạm kết nối (Gateways), hạ tầng mạng (Network and Cloud) và bộ phân tích và xử lý dữ liệu (Services-creation and Solution Layers).



Hình 2. 5 Internet of Things

2.2.3 Ứng dụng

* Nhà thông minh (Smart Home): Điều khiển tự động thiết bị gia dụng và quản lý năng lượng hiệu quả.
* Y tế (Healthcare): Giám sát sức khỏe từ xa và thiết bị đeo thông minh.
* Công nghiệp (Industrial IoT): Tự động hóa sản xuất và bảo trì dự đoán.
* Giao thông (Transportation): Quản lý phương tiện, tối ưu hóa logistics và hỗ trợ xe tự hành.
* Nông nghiệp thông minh (Smart Agriculture): Giám sát cây trồng, chăn nuôi và tối ưu hóa tưới tiêu.
* Thành phố thông minh (Smart City): Quản lý giao thông, năng lượng và môi trường.

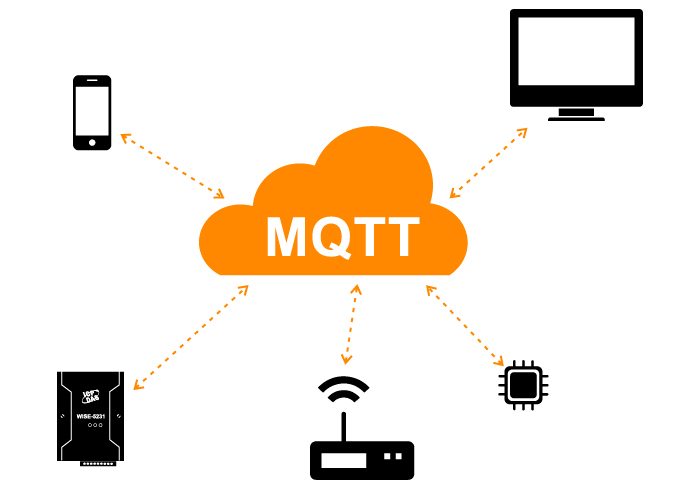
2.3 Giao thức MQTT

2.3.1 Giới thiệu

MQTT (Message Queuing Telemetry Transport) là giao thức truyền thông điệp (message) theo mô hình giao tiếp Publish/Subscribe, phù hợp cho việc vận chuyển dữ liệu từ xa.

Là một giao thức rất nhẹ do đó được sử dụng để giao tiếp các thiết bị (M2M Machine to Machine), WSN (Wireless Sensor Networks) và phổ biến nhất trong các dự án IoT.

Giao thức này được thiết kế để trao đổi dữ liệu giữa máy chủ (server) và khách hàng (client). Ngoài ra với kích thước nhỏ gọn, đơn giản, mức sử dụng năng lượng thấp, các gói dữ liệu được tối ưu hóa và dễ dàng thực hiện đã khiến nó trở nên lý tưởng hơn.



Hình 2. 6 Giao thức MQTT

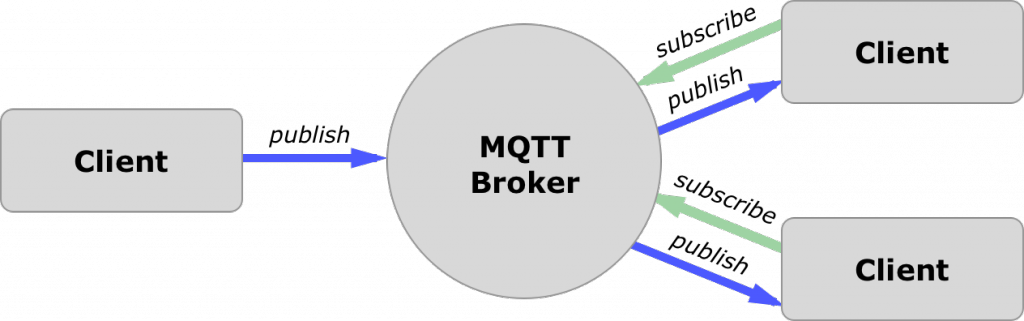
2.3.3 Các thành phần của MQTT

Client (Publisher, Subscriber)

Client (khách hàng) đăng ký một chủ đề để gửi và nhận message:

* Khi một client muốn gửi dữ liệu cho Broker: đây là hoạt động Publisher.
* Khi một client muốn nhận dữ liệu từ Broker: đây là hoạt động Subscriber.

Vì vậy Publisher và Subscriber đóng vai trò đặc biệt của client.



Hình 2. 7 Hoạt động của client

Server (Broker)

Server trong MQTT gọi là một Broker được xem là trung tâm, là điểm giao của các kết nối đến từ client. Nhiệm vụ chính của Broker là nhận đăng ký từ các client về các chủ đề (topic), nhận tin nhắn (message), sắp xếp các message theo hàng đợi rồi chuyển chúng đến một địa chỉ dựa trên việc đăng ký của client. Nhiệm vụ phụ của Broker là có thể đảm nhận thêm một vài tính năng liên quan đến truyền thông như: bảo mật message, lưu trữ message.

2.3.4 MQTT QoS

QoS (Quality of Service): MQTT cung cấp các mức đảm bảo sự chắc chắn trong việc gửi và nhận dữ liệu giữa các client và Broker. MQTT hổ trợ 3 mức:

QoSO: Đảm bảo mức thấp nhất, dữ liệu được gửi đi đúng một lần và sẽ không được kiểm tra đã đến các Broker hay chưa.

2.3.5 Broker trong IoT

Một trong những thành phần của hệ thống IoT là Platform (điện toán đám mây), nó chịu trách nhiệm kết nối các thiết bị với nhau, cho phép người dùng kiểm soát và giám sát các thiết bị của mình. Và Broker cũng chính là Platform, nó nằm trong hệ thống IoT.

Có hai cách tạo ra một Broker:

* Tự tạo Broker MQTT trên máy tính, raspberry, server, ...
* Sử dụng các dịch vụ MQTT Broker có sẵn như CloudMQTT.

A diagram of a server

Description automatically generated

Hình 2. 8 Thành phần của một MQTT trong hệ thống IoT

2.3.6 Ưu điểm

Đây là một giao thức nhẹ. Do đó, dễ dàng thực hiện trong phần mềm và nhanh chóng trong việc truyền nhận dữ liệu, ít bị ảnh hưởng bởi tốc độ mạng.

Giao thức dựa trên kỹ thuật tin nhắn, vì vậy tốc độ khá nhanh.

Gói dữ liệu truyền được tối ưu hóa.

Sử dụng nguồn điện năng thấp, tiếp kiệm năng lượng cho thiết bị được kết nối.

Thời gian thực, đây là điều đặc biệt quan trọng trong các dự án IoT.

2.3.7 Bảo mật

MQTT được thiết kế một cách nhẹ và linh hoạt nhất có thể. Do đó nó chỉ có một lớp bảo mật ở tầng ứng dụng: bảo mật xác thực (xác thực các client được quyền truy cập đến Broker). Tuy nhiên, MQTT vẫn có thể được cài đặt kết hợp với các giải pháp bảo mật đa tầng khác như kết hợp với VNP ở tầng mạng học hoặc SSLL/TLS ở tầng transport.

MQTT được thiết kế nhằm phục vụ truyền thông machine-to-machine nhưng trên thực tế chứng minh nó lại hoạt động một cách linh hoạt hơn mong đợi. Nó hoàn toàn có thể áp dụng cho các kịch bản truyền thông khác nhau: machine to machine, app to app. Chỉ cần có một Broker phù hợp và MQTT Client được cài đặt đúng cách, các thiết bị xây dựng trên nhiều nền tảng khác nhau có thể giao tiếp được với nhau một cách dễ dàng.

2.4 HTTP

2.4.1 Giới thiệu

HTTP (HyperText Transfer Protocol - Giao thức truyền tải siêu văn bản) là một trong các giao thức chuẩn về mạng Internet, được dùng để liên hệ thông tin giữa Máy cung cấp dịch vụ (Web server) và Máy sử dụng dịch vụ (Web client), là giao thức Client/Server dùng cho World Wide Web - WWW. HTTP là một giao thức ứng dụng của bộ giao thức TCP/IP (các giao thức nên tảng cho Internet).

HTTP hoạt động dựa trên mô hình Client - Server. Trong mô hình này, các máy tính của người dùng sẽ đóng vai trò làm máy khách (Client). Sau một thao tác nào đó của người dùng, các máy khách sẽ gửi yêu cầu đến máy chủ (Server) và chờ đợi câu trả lời từ những máy chủ này.

A computer screen shot of a computer screen

Description automatically generated

Hình 2. 9 Giao thức HTTP

2.4.2 Các thành phần chính của HTTP

a) HTTP – Requests

HTTP Request Method: Là phương thức để chỉ ra hành động mong muốn được thực hiện trên tài nguyên đã xác định.

Câu trúc của một HTTP Request:

* Một Request line = Giao thức HTTP (HTTP method) + URL Request (request target) + Phiên bản HTTP (HTTP version). Giao thức HTTP định nghĩa một tập các giao thức GET, POST, HEAD, PUT... Client có thể sử dụng một trong các phương thức đó để gửi request lên server.
* Có thể có hoặc không các trường header
* Một dòng trống để đánh dấu sự kết thúc của các trường Header.
* Request Header Fields: Các trường header cho phép client truyền thông tin bổ sung về yêu cầu, và về chính client, đến server. Một số trường: Accept-Charset, Accept-Encoding, Accept-Language, Authorization, Expect, From, Host, ..

Khi request đến server, server thực hiện một trong 3 hành động sau:

* Server phân tích request nhận được, maps yêu câu với tập tin trong tập tài liệu của server, và trả lại tập tin yêu cầu cho client.
* Server phân tích request nhận được, maps yêu câu vào một chương trình trên server, thực thi chương trình và trả lại kết quả của chương trình đó.
* Request từ client không thể đáp ứng, server trà lại thông báo lỗi

A diagram of a file

Description automatically generated

Hình 2. 10 Cấu trúc HTTP Request

Giao thức HTTP định nghĩa một tập các phương thức request, client có thể sử dụng một trong các phương thức này để tạo request tới HTTP server, dưới đây liệt kê một số phương thức phổ biến.

A screenshot of a computer

Description automatically generated

Hình 2. 11 Phương thức Request

2.5 Webserver

2.5.1 Giới thiệu

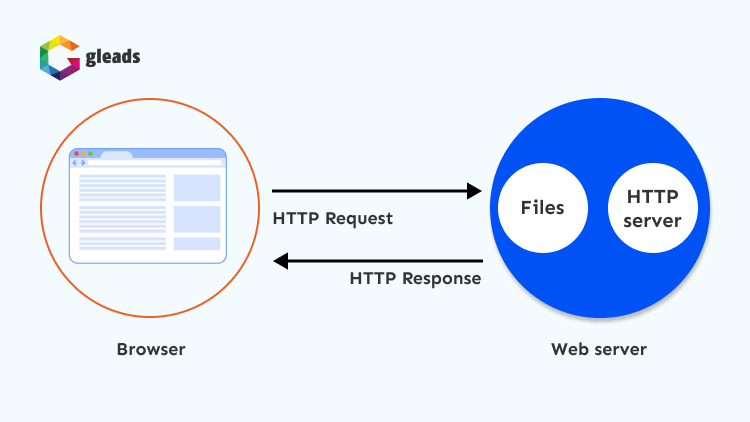
Web server có thể là phần cứng hoặc phần mềm, hoặc cả hai.

Ở khía cạnh phần cứng, web server là một máy tính lưu trữ các file thành phần của một website (các tài liệu, hình ảnh,...) và có thể phân phát chúng tới thiết bị của người dùng. Web server kết nối tới Internet và có thể truy cập tới thông qua một tên miền.

Ở khía cạnh phần mềm, web server điều khiển người sử dụng web truy cập tới các file được lưu trữ trên một HTTP server (máy chủ HTTP). HTTP server là một phần mềm hiểu được các địa chỉ web (URL) và giao thức trình duyệt web (HTTP).

2.5.2 Giao tiếp với Webserver

Khi một trình duyệt cần một file lưu trữ trên một web server, trình duyệt sẽ yêu cầu (request) file đó thông qua HTTP. Khi một yêu cầu gửi tới đúng web server (phần cứng), HTTP server (phần mềm) sẽ gửi file được yêu cầu cũng thông qua HTTP.



Hình 2. 12 Giao tiếp với Webserver

Web server hỗ trợ giao thức HTTP (Giao thức truyền phát siêu văn bản).

HTTP là cách truyền các siêu văn bản giữa hai máy tính. HTTP cung cấp các quy tắc rõ ràng, về cách client và server giao tiếp với nhau

2.6 FreeRTOS

2.6.1 Tổng quan về FreeRTOS

FreeRTOS là một hệ điều hành thời gian thực (RTOS - Real-Time Operating System) nguồn mở, được thiết kế đặc biệt cho các hệ thống vi điều khiển và các ứng dụng nhúng. FreeRTOS được phát triển bởi Richard Barry và hiện nay thuộc sở hữu của Amazon Web Services (AWS).



Hình 2. 13 FreeRTOS

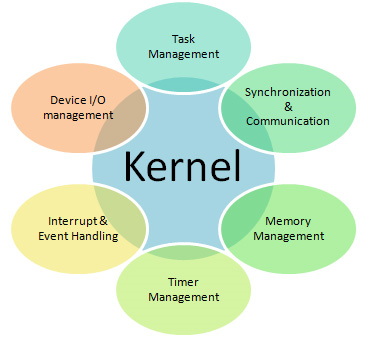
FreeRTOS được phân phối theo giấy phép MIT, cho phép sử dụng trong cả các dự án thương mại và phi thương mại mà không yêu cầu mở mã nguồn của ứng dụng. FreeRTOS đã trở thành một trong những RTOS phổ biến nhất cho các hệ thống nhúng nhờ vào những đặc điểm sau:

* Nhỏ gọn và hiệu quả, chiếm ít tài nguyên hệ thống
* Dễ dàng tích hợp với nhiều vi điều khiển khác nhau
* Hỗ trợ đa nền tảng
* Tính khả chuyển cao giữa các vi điều khiển khác nhau
* Cộng đồng hỗ trợ rộng lớn và tài liệu phong phú

2.6.2 Kiến trúc FreeRTOS

FreeRTOS có kiến trúc đơn giản với ba thành phần chính:

* Kernel (nhân): Quản lý và lập lịch các task, cung cấp các dịch vụ hệ thống cơ bản
* Tác vụ (Tasks): Đại diện cho các tác vụ chương trình độc lập
* Các cơ chế đồng bộ hóa và giao tiếp: Semaphores, mutexes, queues và event groups



Hình 2. 14 Kiến trúc FreeRTOS

FreeRTOS sử dụng cơ chế lập lịch ưu tiên (priority-based scheduling), trong đó mỗi task được gán một mức độ ưu tiên. Task có mức ưu tiên cao nhất sẽ được thực thi trước. FreeRTOS cũng hỗ trợ chế độ round-robin giữa các task có cùng mức ưu tiên.

2.6.3 Lợi ích của FreeRTOS trong phát triển ứng dụng ESP32

* Cấu trúc hóa ứng dụng: FreeRTOS cho phép chia nhỏ logic phức tạp của đồ án thành các task độc lập, mỗi task chịu trách nhiệm cho một chức năng cụ thể (ví dụ: Task đọc cảm biến A, Task đọc cảm biến B, Task xử lý dữ liệu, Task giao tiếp WiFi/Bluetooth, Task điều khiển động cơ, Task cập nhật hiển thị LCD). Điều này làm cho mã nguồn trở nên rõ ràng, dễ hiểu, dễ quản lý và gỡ lỗi hơn so với một cấu trúc super-loop lớn.
* Đảm bảo tính đáp ứng: Bằng cách gán ưu tiên phù hợp cho các task, có thể đảm bảo rằng các hoạt động quan trọng (ví dụ: xử lý tín hiệu cảnh báo khẩn cấp, phản hồi lệnh điều khiển tức thời) được ưu tiên thực thi, ngay cả khi các task khác đang bận rộn. Điều này rất quan trọng trong các ứng dụng cần phản ứng nhanh với môi trường hoặc người dùng.
* Xử lý đồng thời hiệu quả: Trong nhiều đồ án ESP32, cần thực hiện nhiều việc "cùng lúc" - ví dụ, vừa liên tục đọc dữ liệu từ cảm biến, vừa xử lý tính toán, vừa duy trì kết nối mạng và gửi/nhận dữ liệu. FreeRTOS cung cấp cơ chế lập lịch và chuyển đổi ngữ cảnh cho phép các hoạt động này diễn ra xen kẽ một cách hiệu quả, tạo ảo giác về sự đồng thời thực sự.
* Tận dụng kiến trúc Dual-Core của ESP32: FreeRTOS trong ESP-IDF cho phép gán các task chạy trên từng lõi CPU cụ thể thông qua hàm xTaskCreatePinnedToCore. Nhờ đó, công việc có thể được phân chia một cách hợp lý — ví dụ, một lõi có thể đảm nhận các tác vụ yêu cầu xử lý tính toán cao hoặc yêu cầu thời gian thực nghiêm ngặt (như xử lý tín hiệu), trong khi lõi còn lại có thể đảm nhiệm các tác vụ như giao tiếp mạng hoặc các tác vụ ít nhạy cảm về thời gian. Cách tiếp cận này giúp cải thiện hiệu suất tổng thể của hệ thống một cách đáng kể.
* Đơn giản hóa xử lý I/O Blocking: Nhiều hoạt động I/O (như đọc/ghi file, giao tiếp mạng) có thể mất thời gian chờ đợi. Trong môi trường không có RTOS, việc chờ đợi này sẽ chặn toàn bộ hệ thống. Với FreeRTOS, task thực hiện I/O có thể bị block trong khi chờ, cho phép các task khác tiếp tục chạy, cải thiện hiệu quả sử dụng CPU. Các driver của ESP-IDF thường tích hợp sẵn với FreeRTOS để hỗ trợ hoạt động non-blocking hoặc blocking hiệu quả.
* Cộng đồng hỗ trợ lớn và tài liệu phong phú: FreeRTOS là một dự án mã nguồn mở rất phổ biến, có cộng đồng người dùng đông đảo và rất nhiều tài liệu hướng dẫn, ví dụ mã nguồn, diễn đàn thảo luận. Khi gặp vấn đề trong quá trình phát triển đồ án, khả năng tìm kiếm được sự trợ giúp là rất cao.

2.7 Node-RED

2.7.1 Giới thiệu

Node-RED là một công cụ phát triển đồ họa giúp kết nối các thiết bị phần cứng, API và dịch vụ trực tuyến với nhau một cách dễ dàng.

Cung cấp một trình soạn thảo trực quan cho phép nhà phát triển có thể cấu hình tùy chỉnh các chức năng bằng cách sử dụng các (node) từ bất kỳ trình duyệt nào trên máy tính.

Mỗi ứng dụng Node-RED bao gồm các node được liên kết với nhau dưới dạng input, operation và output.

A close-up of a band aid

Description automatically generated

Hình 2. 15 Node trong Node – RED cơ bản

Node-RED ban đầu được IBM phát triển để sử dụng nội bộ nhưng đã được mở mã nguồn vào năm 2016.

Kể từ đó, ngành công nghiệp tự động hóa đã áp dụng công nghệ này như một cách dễ dàng để tạo ra các ứng dụng IoT công nghiệp nhằm thu thập, xử lý và chia sẻ dữ liệu qua Internet.

2.7.2 Node-RED trong IoT

Node-RED giúp phát triển các ứng dụng IoT công nghiệp dễ dàng theo hai cách:

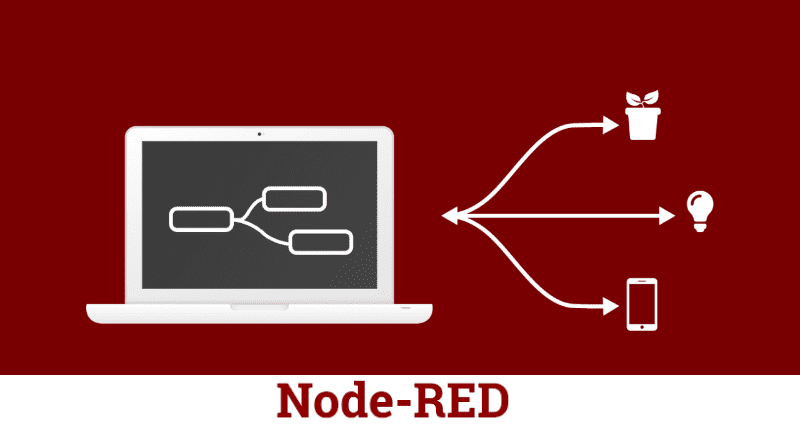
- Đầu tiên, Node-RED được tích hợp sẵn một số lượng lớn các nút được dựng sẵn. Là một nhà phát triển, bạn có thể sử dụng các nút này trong ứng dụng của mình để dễ dàng thực hiện các tác vụ phức tạp như gửi dữ liệu bằng giao thức MQTT , giao thức Modbus/TCP hoặc qua email.

Thậm chí còn có các nút cho phép bạn xuất bản dữ liệu lên các dịch vụ trực tuyến như Dropbox và Google Drive.

Hiện tại, có hơn hai nghìn nút có sẵn để bạn sử dụng trong các ứng dụng IoT công nghiệp của mình và nhờ vào nỗ lực không ngừng nghỉ của cộng đồng nguồn mở, con số đó đang không ngừng tăng lên.

- Thứ hai, Node-RED là một công cụ phát triển đồ họa. Mỗi nút trong ứng dụng được đặt trên một canvas và kết nối với các nút khác. Mỗi nút trong ứng dụng thực hiện một tác vụ cụ thể như thu thập dữ liệu, xử lý dữ liệu hoặc gửi dữ liệu.

Hai cách trên làm cho Node-RED trở thành một công cụ mạnh mẽ để phát triển các ứng dụng IoT, khiến nó có thể là một trong số các công cụ chính trong số các công cụ dành cho phát triển IoT



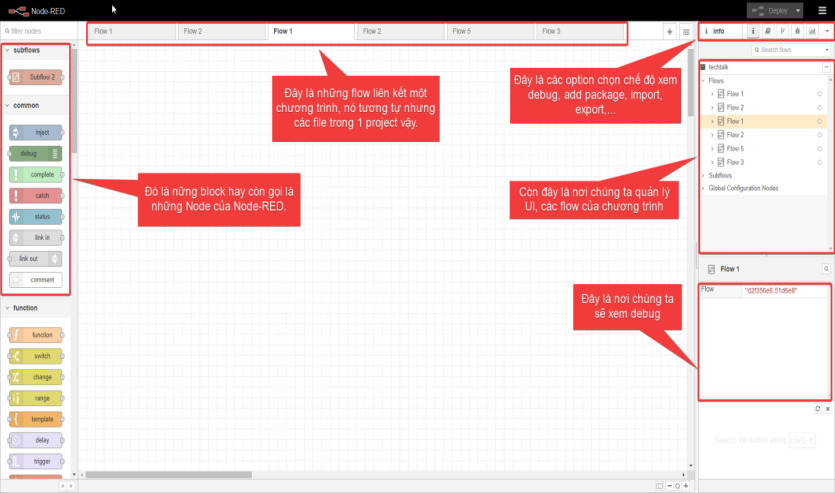
Hình 2. 16 Node - RED và IoT

2.7.3 Tính năng

Được xây dựng trên Node.js, hỗ trợ môi trường thời gian chạy nhẹ.

Nhanh chóng trong việc phát triển các ứng dụng IoT.

Lấy dữ liệu từ các cảm biến và thiết bị. Phù hợp với hầu hết các thiết bị, được sử dụng rộng rãi trên Raspberry, Arduino, các thiết bị Android.



Hình 2. 17 Giao diện Node - RED

Giao diện lập trình gồm 3 thành phần:

* Bên trái: tập hợp các nút có sẵn để xây dựng ứng dụng.
* Trung tâm: nơi thực hiện ứng dụng, kết nối các nút.

2.8 MySQL

2.8.1 Giới thiệu

MySQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. Người dùng có thể tải về MySQL miễn phí từ trang chủ. MySQL có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS,.. MySQL là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL). MySQL được sử dụng cho việc bổ trợ NodeJs, PHP, Perl, và nhiều ngôn ngữ khác, làm nơi lưu trữ những thông tin trên các trang web viết bằng NodeJs, PHP hay Perl,...



Hình 2.18Cơ sở dữ liệu MySQL

2.8.2 Các tính năng của MySQL

Tạo và quản lý bảng: MySQL cho phép tạo ra các bảng để lưu trữ dữ liệu. Mỗi bảng chứa các cột và hàng, trong đó mỗi cột đại diện cho một thuộc tính và mỗi hàng đại diện cho một bản ghi trong cơ sở dữ liệu. Người dùng có thể định nghĩa các ràng buộc, quan hệ và chỉ mục trong bảng để xác định sự liên quan giữa các bảng.

Gửi yêu cầu SQL: Người dùng hoặc Client có thể gửi các yêu cầu SQL đặc biệt tới MySQL. Yêu cầu này có thể bao gồm các câu truy vấn SELECT, INSERT, UPDATE, DELETE và nhiều loại câu lệnh khác.

Xử lý yêu cầu: MySQL Server nhận và xử lý các yêu cầu SQL từ Client. Công cụ sẽ kiểm tra cú pháp và tính hợp lệ của yêu cầu, sau đó thực hiện các thao tác tương ứng trên cơ sở dữ liệu. MySQL sử dụng bộ trình diễn truy vấn (query optimizer) để tối ưu hóa hiệu suất truy vấn.

Trả kết quả: Sau khi xử lý yêu cầu, MySQL Server sẽ trả về kết quả cho Client. Kết quả có thể là dữ liệu truy vấn từ cơ sở dữ liệu hoặc thông báo về việc cập nhật, thêm, xóa dữ liệu.

2.8.3 Ưu điểm của MySQL

Được tối ưu hóa để đạt hiệu suất cao, cho phép thực thi truy vấn nhanh chóng và xử lý số lượng lớn dữ liệu một cách hiệu quả.

Có khả năng mở rộng linh hoạt, cho phép mở rộng cơ sở dữ liệu dựa trên nhu cầu. MySQL hỗ trợ cơ chế nhân rộng (scalability) để thêm các máy chủ hoặc node để xử lý tải lớn và đảm bảo sự ổn định của hệ thống.

Cung cấp nhiều tính năng phong phú để làm việc với cơ sở dữ liệu. Công cụ hỗ trợ các chuẩn SQL phổ biến và thêm những chức năng như khóa ngoại, xử lý giao dịch, truy vấn phức tạp, truy vấn văn bản đầy đủ,... Điều này giúp lập trình viên có thể thực hiện các hoạt động phức tạp trên cơ sở dữ liệu một cách thuận tiện.

Cung cấp các tính năng bảo mật mạnh mẽ để bảo vệ dữ liệu. Công cụ hỗ trợ xác thực người dùng, quản lý quyền truy cập, mã hóa dữ liệu và các biện pháp bảo mật khác.

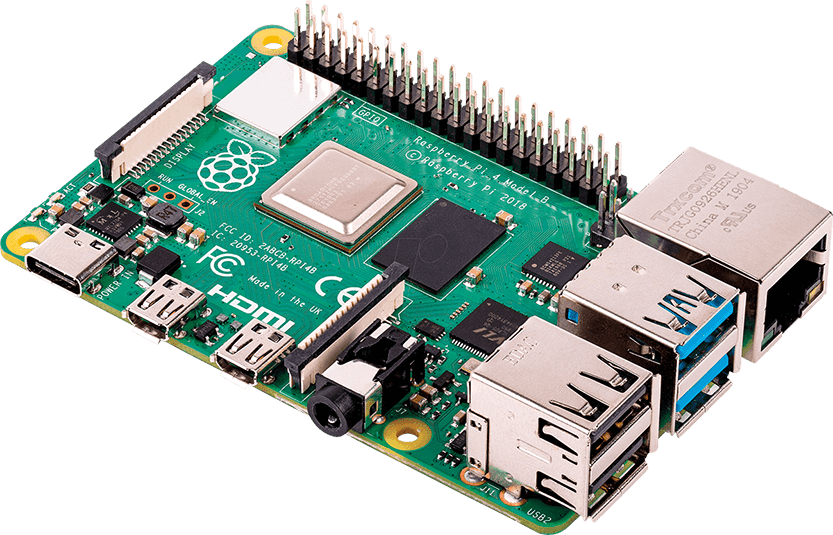
2.9 Raspberry Pi 4

2.9.1 Giới thiệu

Raspberry Pi 4 là một máy tính bảng đơn (Single Board Computer - SBC) được phát triển bởi Raspberry Pi Foundation, ra mắt vào tháng 6/2019. Nó tích hợp bộ vi xử lý Broadcom BCM2711 với bốn nhân Cortex-A72, bộ nhớ RAM tùy chọn từ 2GB đến 8GB, hỗ trợ Wi-Fi, Bluetooth, và nhiều giao diện ngoại vi như GPIO, USB, HDMI, và Ethernet. Với hiệu năng vượt trội so với các phiên bản trước, Raspberry Pi 4 là nền tảng lý tưởng cho các ứng dụng giáo dục, phát triển dự án IoT, máy chủ cá nhân, và giải trí.

Raspberry Pi 4 cung cấp khả năng kết nối mạng qua Wi-Fi 802.11ac băng tần kép và Bluetooth 5.0, cho phép tương tác với các thiết bị không dây và tích hợp vào các hệ thống IoT. Hỗ trợ nhiều giao thức như TCP/IP, HTTP, và MQTT, thiết bị này phù hợp cho các ứng dụng giám sát, điều khiển từ xa, và thu thập dữ liệu. Với cộng đồng người dùng lớn và giá thành hợp lý (từ 35 USD), Raspberry Pi 4 được sử dụng rộng rãi trong học tập, nghiên cứu, và các dự án công nghệ sáng tạo.

Các ứng dụng phổ biến bao gồm: học lập trình (Python, Scratch), xây dựng hệ thống nhà thông minh, phát triển máy chủ web, media center (phát video 4K), và máy chơi game cổ điển. Raspberry Pi 4 có các phiên bản khác nhau dựa trên dung lượng RAM, đáp ứng nhu cầu đa dạng từ các dự án đơn giản đến phức tạp.



Hình 2. 19 Raspberry Pi 4

2.9.2 Các loại Raspberry Pi 4

Raspberry Pi 4 có các phiên bản khác nhau, chủ yếu dựa trên dung lượng RAM và một số cải tiến nhỏ:

* Raspberry Pi 4 Model B (2GB): Phiên bản cơ bản với 2GB RAM LPDDR4, phù hợp cho các ứng dụng nhẹ như học lập trình, dự án IoT đơn giản, hoặc máy chủ nhỏ. Raspberry Pi 4 Model B (4GB): Phiên bản phổ biến nhất, với 4GB RAM, lý tưởng cho đa nhiệm, trình duyệt web, và các dự án yêu cầu hiệu năng trung bình như media center hoặc retro gaming.
* Raspberry Pi 4 Model B (8GB): Phiên bản cao cấp với 8GB RAM, hỗ trợ các ứng dụng đòi hỏi nhiều tài nguyên như máy tính để bàn, máy chủ web lớn, hoặc xử lý dữ liệu phức tạp.
* Raspberry Pi 4 (phiên bản cải tiến): Một số bo mạch được cập nhật từ năm 2020 với CPU xung nhịp 1.8GHz (thay vì 1.5GHz) và cải thiện khả năng quản lý nhiệt, nhưng vẫn giữ nguyên các thông số khác.

2.9.3 Tính năng

Raspberry Pi 4 được trang bị các tính năng kỹ thuật nổi bật, đáp ứng nhu cầu phát triển đa dạng:

* Wi-Fi: Hỗ trợ Wi-Fi 802.11ac băng tần kép (2.4GHz và 5GHz), đảm bảo kết nối không dây tốc độ cao và ổn định.
* Bluetooth: Tích hợp Bluetooth 5.0, hỗ trợ kết nối với các thiết bị như bàn phím, chuột, loa, hoặc cảm biến không dây.
* Giao diện ngoại vi đa dạng: Cung cấp GPIO (40 chân), USB (2 x USB 3.0, 2 x USB 2.0), micro-HDMI, Ethernet, và cổng âm thanh/video analog, cho phép kết nối với nhiều thiết bị và cảm biến.
* Bộ nhớ flash: Sử dụng thẻ microSD làm bộ nhớ chính (khuyến nghị Class 10, từ 16GB trở lên) để lưu trữ hệ điều hành và dữ liệu.
* Bộ nhớ RAM: Tùy chọn 2GB, 4GB, hoặc 8GB LPDDR4-3200, hỗ trợ đa nhiệm và các ứng dụng yêu cầu tài nguyên lớn.
* Hỗ trợ giao thức mạng: Tích hợp TCP/IP, HTTP, MQTT, và các giao thức IoT khác, giúp dễ dàng tích hợp vào các hệ thống mạng.
* Hỗ trợ video 4K: Hai cổng micro-HDMI cho phép xuất video 4K 60Hz, phù hợp cho media center hoặc hiển thị dữ liệu chất lượng cao.
* Quản lý năng lượng: Sử dụng nguồn USB-C 5V/3A, với khả năng tối ưu tiêu thụ năng lượng trong các ứng dụng nhẹ.
* Hỗ trợ OTA (Over-The-Air): Cho phép cập nhật hệ điều hành và phần mềm từ xa thông qua Wi-Fi, đảm bảo thiết bị luôn được nâng cấp dễ dàng.

2.10 ESP32

2.10.1 Giới thiệu

ESP32 kết hợp một bộ vi xử lý hai nhân Tensilica LX6, bộ nhớ flash tích hợp, Wi-Fi, Bluetooth và nhiều giao diện ngoại vi khác nhau như GPIO, UART, I2C và SPI. Điều này khiến nó trở thành một nền tảng phát triển mạnh mẽ cho việc xây dựng các ứng dụng IoT đa dạng, từ các dự án nhỏ đến các hệ thống phức tạp.

ESP32 cung cấp khả năng kết nối mạng với Wi-Fi và Bluetooth, cho phép thiết bị được kết nối với các mạng không dây và tương tác với các thiết bị khác. Nó cũng hỗ trợ nhiều giao thức và tiện ích như TCP/IP, HTTP, MQTT và nhiều giao thức khác, giúp việc giao tiếp và truyền thông dễ dàng trong các ứng dụng IoT.

ESP32 được sử dụng phổ biến trong việc phát triển các ứng dụng như hệ thống giám sát, điều khiển thiết bị từ xa, đo lường và cảm biến, thiết bị định vị và nhiều ứng dụng IoT khác. Nó có sẵn trong các phiên bản khác nhau với các tính năng và giao diện ngoại vi khác nhau, giúp phù hợp với nhu cầu cụ thể của từng dự án.

A computer chip with many different colored buttons

Description automatically generated

Hình 2. 20 Module ESP32-WROOM-32D

2.11 BMP280+AHT20

2.11.1 Giới thiệu

BMP280+ATH20 là cảm biến đo áp suất khí quyển và nhiệt độ được phát triển bởi Bosch Sensortec. Đây là phiên bản cải tiến của cảm biến BMP180 với hiệu suất cao hơn, tiêu thụ năng lượng thấp hơn và độ chính xác cao hơn. Cảm biến này thường được sử dụng trong các ứng dụng thời tiết, đo độ cao, thiết bị IoT và di động.

Đặc điểm chính:

* Điện áp hoạt động: 1.71V đến 3.6V.
* Giao tiếp: I2C và SPI.
* Phạm vi đo áp suất: 300 hPa đến 1100 hPa (độ cao từ -500 m đến 9000 m).
* Độ chính xác nhiệt độ: ±1°C.
* Kích thước nhỏ gọn và dễ tích hợp.

A close-up of a purple circuit board

AI-generated content may be incorrect.

Hình 2. 21 Module cảm biến BMP280+AHT20

2.11.2 Cấu tạo

Module bao gồm hai vi mạch cảm biến độc lập:

BMP280:

* Bộ cảm biến MEMS đo áp suất.
* Cảm biến nhiệt độ tích hợp.
* IC xử lý tín hiệu và giao tiếp I2C/SPI.

AHT20:

* Cảm biến điện dung đo độ ẩm.
* Cảm biến nhiệt độ tích hợp.
* IC điều khiển tích hợp giao tiếp I2C.

2.11.3 Nguyên lý hoạt động

Đo áp suất (BMP280):

* Màng cảm biến MEMS biến dạng dưới áp suất khí quyển.
* Biến dạng này được chuyển thành tín hiệu điện, rồi đưa vào bộ chuyển đổi ADC tích hợp.
* Kết quả được xử lý và gửi về qua giao tiếp I2C/SPI.

Đo nhiệt độ và độ ẩm (AHT20):

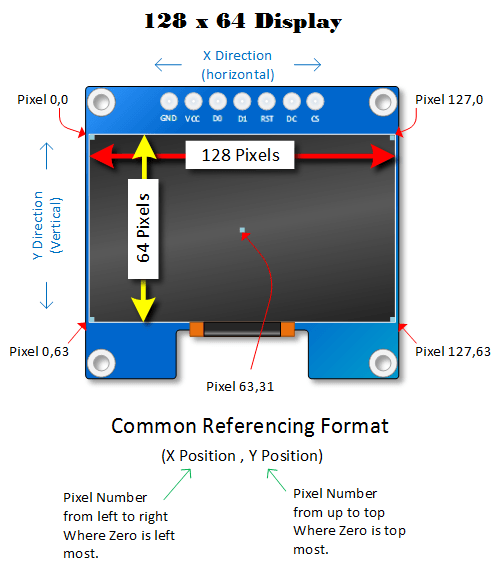
* Cảm biến điện dung đo độ ẩm dựa trên thay đổi điện dung của vật liệu cảm ứng khi hấp thụ hơi nước
* Nhiệt độ được đo bằng cảm biến điện trở nhiệt bên trong.
* Dữ liệu được xử lý và trả về qua I2C.
* Giao tiếp và cấu hình:
* Người dùng có thể cấu hình BMP280 qua I2C hoặc SPI.
* AHT20 chỉ hỗ trợ I2C.

Các cảm biến có thể hoạt động ở chế độ chu kỳ, tiết kiệm năng lượng hoặc liên tục tùy cấu hình phần mềm

2.12 OLED

2.12.1 Giới thiệu

Màn hình OLED 0.96 inch I2C là một loại màn hình hiển thị sử dụng công nghệ OLED (Organic Light Emitting Diodes), có kích thước 0.96 inch (khoảng 2.44 cm). Nó sử dụng giao thức truyền thông I2C (Inter-Integrated Circuit) để giao tiếp với các vi điều khiển như Arduino, Raspberry Pi và các mạch điện tử khác..



Hình 2. 22 Màn hình OLED

2.12.3 Nguyên lý hoạt động

Màn hình OLED 0.96 inch hoạt động dựa trên công nghệ phát sáng của các điểm ảnh được làm từ vật liệu hữu cơ. Khi có dòng điện chạy qua, các electron và lỗ trống tái hợp tại lớp phát sáng, tạo ra ánh sáng mà không cần đèn nền. Điều này không chỉ giúp tiết kiệm năng lượng mà còn mang lại độ tương phản cao, cho màu đen sâu và hình ảnh rõ nét.

Khi sử dụng với vi điều khiển như ESP32, màn hình giao tiếp qua giao thức I2C với hai dây tín hiệu chính: SCL (đồng hồ) và SDA (dữ liệu). Vi điều khiển gửi lệnh và dữ liệu đến driver IC, phổ biến là SSD1306, để điều khiển màn hình.

2.13 Relay 2 chanel

2.13.1 giới thiệu

Module Relay 2 kênh là một module điện tử được sử dụng để điều khiển các thiết bị hoặc mạch điện thông qua tín hiệu điều khiển từ một nguồn bên ngoài như Arduino, Raspberry Pi… Nó cung cấp hai kênh relay độc lập, cho phép người dùng điều khiển hai thiết bị riêng biệt. Nó bao gồm 2 Relay, mỗi Relay có dòng điện là 10A và hoạt động ở điện áp 250VAC hoặc 30VDC.

A red and blue circuit board with black text

Description automatically generated

Hình 2. 23 Relay 2 Chanel

2.13.3 Nguyên Lý hoạt động

Khi dòng điện chạy qua mạch thứ nhất (1), nó sẽ kích hoạt nam châm điện. Từ đó tạo ra từ trường để thu hút một tiếp điểm (màu đỏ). Sau đó sẽ kích hoạt mạch thứ hai (2). Khi tắt nguồn, một lò xo được lắp trước vào tiếp điểm sẽ có nhiệm vụ là kéo tiếp điểm trở lại vị trí ban đầu, tắt mạch thứ hai lại một lần nữa.

Bên dưới là một hình ảnh cho thấy cách một relay liên kết hai mạch với nhau. Ở bên trái, có mạch đầu vào được cung cấp bởi một công tắc hoặc loại cảm biến nào đó.

A diagram of a wire and a coil

Description automatically generated with medium confidence

Hình 2. 24 Nguyên lý hoạt động của relay

Khi mạch này được kích hoạt thì nó cung cấp dòng điện cho một nam châm điện, sau đó sẽ kéo công tắc kim loại đóng lại. Từ đó kích hoạt mạch đầu ra thứ hai (ở phía bên phải).

CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

## 3.1 Tổng quan về hệ thống

**Giới thiệu chung**

Hệ thống IoT được xây dựng nhằm thu thập dữ liệu từ các cảm biến môi trường, đồng thời cung cấp khả năng điều khiển thiết bị từ xa thông qua giao thức MQTT. Hệ thống gồm hai node chính: Node và node master, giao tiếp với nhau thông qua MQTT broker.

**Mô tả hệ thống**

* Node: Node được trang bị các cảm biến môi trường như BMP280+AHT20. Node này có nhiệm vụ:
* Gửi dữ liệu lên MQTT broker: Dữ liệu từ cảm biến được định dạng và gửi qua các topic MQTT.
* Nhận lệnh điều khiển relay: Lắng nghe lệnh từ MQTT broker qua topic điều khiển và thực hiện đóng/mở relay tương ứng.
* MQTT broker: MQTT broker đóng vai trò trung gian trong hệ thống, đảm bảo các node có thể giao tiếp với nhau một cách hiệu quả:
* Xử lý dữ liệu: Nhận dữ liệu từ Node và phân phối cho các client subscribe topic tương ứng (bao gồm node master).
* Điều khiển relay: Nhận lệnh từ node master và gửi lệnh đó đến Node qua topic MQTT.

**Kết nối và giao tiếp**

Hệ thống hoạt động trên giao thức MQTT với các topic được định nghĩa như sau:

* Topic dữ liệu cảm biến: “slave/sensors/sensor\_data” Node gửi dữ liệu cảm biến.
* Topic điều khiển relay: “master/control/mode\_relay” Node master gửi lệnh điều khiển relay, Node lắng nghe và thực hiện.

**Sơ lược quy trình hoạt động**

Node đọc dữ liệu từ cảm biến, gửi dữ liệu lên MQTT broker qua topic node/slave/sensor\_data.

Node master subscribe topic này để hiển thị dữ liệu lên giao diện web.

Khi người dùng nhấn nút vật lý hoặc điều khiển relay từ web server, node master gửi lệnh qua topic node/relay/control.

Node nhận lệnh từ topic này và thực hiện đóng/mở relay tương ứng.

## 3.2 Sơ đồ khối chức năng

**Sơ đồ khối hệ thống tổng quát**

* Node có chức năng gửi dữ liệu lên MQTT broker và nhận lệnh điều khiển chế độ từ MQTT broker.
* Node master có chức năng điều khiển relay qua MQTT broker thông qua nút nhấn hoặc giao diện web
* MQTT broker đóng vai trò trung gian trao đổi dữ liệu giữa Node và node master

**A white cloud with black text

AI-generated content may be incorrect.**

Hình 3. 1 Sơ đồ khối tổng quát

**Sơ đồ khối node**

* Khối nguồn là khối cơ bản nhất nó cung cấp dòng nuôi ổn định cho toàn bộ linh kiện trong mạch.
* Khối ESP32: đây là khối xử lý trung tâm, có nhiệm vụ
* Đọc dữ liệu từ khối cảm biến.
* Giao tiếp trao đổi dữ liệu với MQTT broker qua WIFI.
* Xuất tín hiệu điều khiển khối hiển thị để hiển thị thông tin.
* Xuất tín hiệu điều khiển khối relay để điều khiển relay.
* Khối WIFI: kết nối WIFI và thực hiện các nhiệm vụ sau
* Nhận dữ liệu từ MQTT broker và điều khiển trực tiếp các thiết bị ngoại vi.
* Nhận dữ liệu từ khối xử lý trung tâm để gửi lên MQTT broker.
* Khối cảm biến: bao gồm
* Cảm biến BMP280+ATH20 để thu thập dữ liệu áp suất trong môi trường.
* Khối hiển thị: hiển thị thông tin nhiệt độ, độ ẩm, áp xuất , chế độ relay
* Khối relay: nhận lệnh từ khối xử lý trung tâm để bật tắt theo chế độ tương ứng

A diagram of a computer network

Description automatically generated

Hình 3. 2 Sơ đồ khối Node

## 3.3 Lưu đồ thuật toán

a) Lưu đồ chương trình node

A diagram of a flowchart

AI-generated content may be incorrect.

Hình 3. 3Lưu đồ chương trình chính

Lưu đồ chương trình StartAPMode

A diagram of a computer program

AI-generated content may be incorrect.

Hình 3. 4 Lưu đồ chương trình StartAPMode

Lưu đồ chương trình TaskControl

A diagram of a flowchart

AI-generated content may be incorrect.

Hình 3. 5 Lưu đồ chương trình TaskControl

Lưu đồ chương trình TaskDisplay

A diagram of a software project

AI-generated content may be incorrect.

Hình 3. 6 Lưu đồ chương trình TaskDisplay

Lưu đồ chương trình TaskMQTT

A diagram of a company

AI-generated content may be incorrect.

Hình 3. 7 Lưu đồ chương trình TaskMQTT

Lưu đồ chương trình TaskMQTTSubcribe

A diagram of a flowchart

AI-generated content may be incorrect.

Hình 3. 8 Lưu đồ chương trình TaskMQTTSubcribe

Lưu đồ chương trình TaskPrintData

A diagram of a flowchart

AI-generated content may be incorrect.

Hình 3. 9 Lưu đồ chương trình TaskPrintData

Lưu đồ chương trình TaskReadSensor

A diagram of a process

AI-generated content may be incorrect.

Hình 3. 10 Lưu đồ chương trình TaskReadSensor

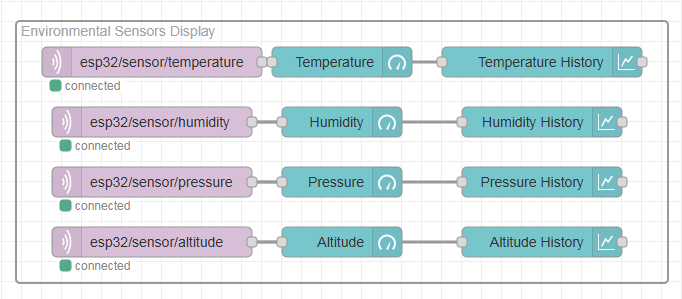
Lưu đồ chương trình TaskResetTimeout

A diagram of a software flowchart

AI-generated content may be incorrect.

Hình 3. 11 Lưu đồ chương trình TaskResetTimeout

## 3.4 Flow node red broker



Hình 3. 12 Flow View Dashboard Node - RED

A diagram of a computer

AI-generated content may be incorrect.

Hình 3. 13 Flow Control Dashboard Node - RED

A diagram of a computer program

AI-generated content may be incorrect.

Hình 3.13 Flow Control mode device and thresholds Web

A diagram of a data storage handler

AI-generated content may be incorrect.

Hình 3. 14 Flow Save database MySQL

## 3.5 Phần mềm và ngôn ngữ lập trình

a) giới thiệu phần mềm lập trình Arduino IDE

**Arduino IDE** là một phần mềm mã nguồn mở được sử dụng để viết, biên dịch và nạp chương trình vào các vi điều khiển của nền tảng Arduino. Phần mềm này hỗ trợ nhiều hệ điều hành như **Windows, macOS, và Linux**, đồng thời cung cấp một giao diện thân thiện với người mới bắt đầu, rất phù hợp cho các ứng dụng về điều khiển nhúng và IoT.

**Arduino IDE** sử dụng ngôn ngữ lập trình dựa trên **C/C++**, kết hợp với các thư viện đơn giản để người dùng có thể dễ dàng giao tiếp với phần cứng như cảm biến, động cơ, LED, v.v.



Hình 3. 15 Arduino IDE

CHƯƠNG 4 : THIẾT KẾ & ĐI DÂY

## 4.1 Sản phẩm hoàn thiện

A circuit board with wires and lights

AI-generated content may be incorrect.

Hình 4. 1 đi dây hệ thống

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4. 2Webserver Node-red

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4. 3 biểu đồ node-red

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4. 4 Web server

A screenshot of a graph

AI-generated content may be incorrect.

Hình 4. 5 biểu dồ nhiệt độ độ ẩm

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

**Kết luận**

Đề tài "HỆ THỐNG IOT GIÁM SÁT NHIỆT DỘ ĐỘ ẨM ÁP XUẤT MÔI TRƯỜNG " đã hoàn thành mục tiêu đề ra, góp phần giải quyết vấn đề ô nhiễm không khí trong không gian kín. Hệ thống được thiết kế tích hợp nhiều cảm biến , giúp đo lường các thông số môi trường bao gồm nhiệt độ, độ ẩm, áp xuất Kết quả nghiên cứu cho thấy:

* Tính khả thi cao: Hệ thống đã được thử nghiệm và hoạt động ổn định trong môi trường thực tế, cung cấp dữ liệu thời gian thực với độ chính xác tương đối tốt.
* Ứng dụng công nghệ IoT hiệu quả: Hệ thống sử dụng giao thức MQTT và cơ sở dữ liệu DB để truyền tải và lưu trữ dữ liệu, giúp dễ dàng quản lý và theo dõi thông tin qua giao diện web trực quan.
* Tính năng tự động hóa: Mặc dù chức năng tự động điều chỉnh thiết bị còn đơn giản, hệ thống đã bước đầu hỗ trợ điều khiển thiết bị dựa trên dữ liệu môi trường đo được.

Tuy nhiên, đề tài vẫn còn một số hạn chế cần cải thiện, như khả năng chống nhiễu dữ liệu từ cảm biến và độ chính xác trong điều kiện môi trường khắc nghiệt. Đây là cơ sở để tiếp tục phát triển và hoàn thiện trong tương lai.

**Hạn chế và khó khăn**

Trong quá trình thực hiện đề tài đã gặp phải một số khó khăn và hạn chế:

* Độ chính xác của cảm biến:
* Cần thêm thời gian và tài nguyên để hiệu chỉnh và thử nghiệm dữ liệu trong các điều kiện môi trường đa dạng.
* Nguồn lực hạn chế:
* Thời gian thực hiện đề tài ngắn khiến nhóm không thể triển khai hết các tính năng mong muốn, như tích hợp AI để phân tích dữ liệu sâu hơn hoặc phát triển ứng dụng di động..
* Khả năng mở rộng:
* Hệ thống hiện tại chỉ được thử nghiệm trên không gian nhỏ và điều kiện đơn giản. Khi áp dụng vào thực tế tại các không gian lớn hoặc môi trường khắc nghiệt hơn, cần thực hiện thêm nhiều thử nghiệm và điều chỉnh.

# TÀI LIỆU THAM KHẢO

[1] Rui Santos, Sara Santos - MicroPython Programming with ESP32 and ESP8266 (2019)

[2] Shiro , Akira - EXPLORE ESP32 MICROPYTHON\_ Python Coding, Arduino Coding, Raspberry Pi, ESP8266, IoT Projects, Android Application Projects (2021)  
[3] Dr V M Senthil kumar, Professor Mr.M.Anantha Guptha, Assistant Professor - INTERNET OF THINGS & ITS APPLICATIONS  
[4] Achim Pieters - ESP32 PinOut