

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

-----🙞🙜🕮🙞🙜-----

Icon

Description automatically generated with medium confidence

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**

**NGHIÊN CỨU, THIẾT KẾ NODE IOT TƯƠNG THÍCH VỚI HOME ASSISTANT**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn** | **: TS. Nguyễn Trung Hiếu** |
| **Họ và tên** | **: Nguyễn Đức Long** |
| **Mã sinh viên** | **: B18DCDT129** |
| **Hệ** | **: Đại học chính quy** |

**Hà Nội - 2022**

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM

**Độc lập – Tự do – Hạnh phúc**

*Hà Nội, ngày 15 tháng 12 năm 2022*

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**

1. Tên đề tài: Nghiên cứu, thiết kế node IoT tương thích với Home Assistant
2. Sinh viên: **Nguyễn Đức Long**            Lớp: **D18DTMT1**
3. Giáo viên hướng dẫn: **TS. Nguyễn Trung Hiếu**
4. Nơi công tác: Khoa Kỹ thuật điện tử 1, Học viện Công nghệ Bưu chính Viễn thông.

**NỘI DUNG NHẬN XÉT**

1. **Đánh giá chung**

Đồ án tốt nghiệp đã hoàn thành đúng nội dung theo đề cương đã được duyệt. Nội dung của đồ án tập trung nghiên cứu Node IoT trong hệ thống IoT ứng dụng trong Smart Home. Sinh viên đã phát triển mã nguồn cho Node IoT có tính bảo mật cao, có khả năng mở rộng, đồng thời áp dụng những công nghệ mới vào quá trình làm đồ án.

1. **Đánh giá chi tiết**

Đồ án gồm 3 chương, đã trình bày tương đối rõ ràng vấn đề được nghiên cứu. Các kết quả chính của đồ án:

* + Hiểu được nguyên lí cũng như cách thức hoạt động của Home Assistant. Tìm hiểu cách tìm
  + Thiết kế được node IoT tự động tích hợp lên nền tảng Home Assistant. Thiết kế được hệ thống hoạt động ổn định, bảo mật, có tính tự động hóa cao.
  + Hình thức trình bày đẹp, bố cục nội dung logic.
  + Kết quả nghiên cứu của đồ án có thể sử dụng làm tài liệu tham khảo tốt trong giảng dạy và nghiên cứu cho sinh viên, kỹ sư ngành điện tử. Mã nguồn có khả năng mở rộng, các sinh viên khóa sau có thể dễ dàng tiếp thu và phát triển tiếp.

1. **Nhận xét về tinh thần, thái độ làm việc**

Sinh viên làm việc nghiêm túc với tinh thần ham học hỏi, chăm chỉ, chịu khó tìm hiểu các công nghệ mới. Trong quá trình làm việc thái độ luôn luôn cầu tiến, tích cực giúp đỡ các bạn, biết lắng nghe ý kiến và tiếp thu.

1. **Kết luận**

Đồ án đảm bảo yêu cầu của một đồ án tốt nghiệp đại học.

Đồng ý cho sinh viên bảo vệ trước Hội đồng chấm bảo vệ tốt nghiệp.

1. **Điểm hướng dẫn:** 10 điểm

|  |  |
| --- | --- |
|  | **GIẢNG VIÊN HƯỚNG DẪN**    **TS. Nguyễn Trung Hiếu** |

NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM

*(Của giảng viên phản biện)*

**Điểm:** *(Bằng chữ:……………………………..)*

**Đồng ý/Không đồng ý** cho sinh viên bảo vệ trước hội đồng chấm đồ án tốt nghiệp…?

*Hà Nội, Ngày….. Tháng…. Năm 2022*

**CÁN BỘ - GIẢNG VIÊN PHẢN BIỆN**

*(ký, ghi rõ họ tên)*

# LỜI CẢM ƠN

Trong quá trình nghiên cứu và thực hiện đề tài nghiên cứu khoa học em đã nhận được sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của thầy cô, gia đình và bạn bè.

Đầu tiên, em xin được gửi lời cảm ơn đến Ban Giám hiệu Học viện Công nghệ Bưu chính Viễn thông đã tạo cho em môi trường rèn luyện tốt để em có thể học tập và tiếp thu được những kiến thức quý báu trong những năm qua.

Em xin cảm ơn tất cả các thầy cô giáo, đặc biệt là các thầy cô trong khoa Kỹ thuật điện tử 1 đã tận tình chỉ dạy những kiến thức quý báu để em có thể hoàn thành được đề tài cũng như những hành trang cần thiết để em có thể bước trên con đường sự nghiệp sau này.

Em xin được gửi lời cảm ơn sâu sắc nhất đến thầy TS. Nguyễn Trung Hiếu và thầy Đinh Sỹ Quảng, những người đã trực tiếp hướng dẫn em thực hiện đề tài này. Mặc dù công việc rất bận rộn nhưng các Thầy vẫn luôn dành thời gian hướng dẫn chỉ bảo tận tình để em có thể hoàn thành tốt đề tài.

Mặc dù đã cố gắng hết sức, song không tránh khỏi những thiếu sót. Em rất mong nhận được sự thông cảm và chỉ bảo tận tình của quý thầy cô và các bạn để em có thể hoàn thành tốt hơn báo cáo này.

Cuối cùng em xin kính chúc các thầy, các cô, gia đình và bạn bè dồi dào sức khỏe và thành công trong sự nghiệp.

*Hà Nội, ngày 15 tháng 12 năm 2022*

***Sinh viên thực hiện***

Nguyễn Đức Long

# 

**MỤC LỤC**

[LỜI CẢM ƠN i](#_Toc122354247)

[DANH MỤC HÌNH ẢNH iii](#_Toc122354248)

[DANH MỤC BẢNG v](#_Toc122354249)

[DANH MỤC CHỮ VIẾT TẮT vi](#_Toc122354250)

[LỜI MỞ ĐẦU viii](#_Toc122354251)

[CHƯƠNG 1. NGHIÊN CỨU TỔNG QUAN 1](#_Toc122354252)

[1.1 Giới thiệu đề tài 1](#_Toc122354253)

[1.1.1 Đặt vấn đề, hiện trạng 1](#_Toc122354254)

[1.1.2 Mục tiêu đề tài 2](#_Toc122354255)

[1.1.3 Mô hình, mô tả hệ thống 2](#_Toc122354256)

[1.2 IoT Gateway 5](#_Toc122354257)

[1.2.1 Máy tính nhúng Orange Pi One 5](#_Toc122354258)

[1.3 Nền tảng Home Assistant 6](#_Toc122354259)

[1.3.1 Giới thiệu 6](#_Toc122354260)

[1.3.2 Kiến trúc của Home Assistant 7](#_Toc122354261)

[1.3.3 Giao tiếp với thiết bị trên Home Assistant 8](#_Toc122354262)

[1.3.4 Ưu điểm của Home Assistant 8](#_Toc122354263)

[Kết luận chương 9](#_Toc122354264)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 10](#_Toc122354265)

[2.1 Mô hình mạng OSI và TCP/IP 10](#_Toc122354266)

[2.1.1 Tầng vật lý 11](#_Toc122354267)

[2.1.2 Tầng liên kết dữ liệu 11](#_Toc122354268)

[2.1.3 Tầng mạng 11](#_Toc122354269)

[2.1.4 Tầng giao vận 12](#_Toc122354270)

[2.1.5 Tầng ứng dụng 12](#_Toc122354271)

[2.2 Công nghệ kết nối wi-fi 13](#_Toc122354272)

[2.2.1 Tổng quan về wi-fi 13](#_Toc122354273)

[2.2.2 Lịch sử ra đời 13](#_Toc122354274)

[2.2.3 Các chuẩn wi-fi 13](#_Toc122354275)

[2.2.4 Bảo mật của wi-fi 15](#_Toc122354276)

[2.3 Các giao thức truyền thông 16](#_Toc122354277)

[2.3.1 Giao thức MQTT 16](#_Toc122354278)

[2.3.2 Giao thức ESP-Touch 29](#_Toc122354279)

[2.3.3 Giao thức One Wire 31](#_Toc122354280)

[2.4 Một số linh kiện sử dụng 34](#_Toc122354281)

[2.4.1 Nguồn HLK-PM01 34](#_Toc122354282)

[2.4.2 Opto quang MOC3021 36](#_Toc122354283)

[2.4.3 Triac BT136E 37](#_Toc122354284)

[Kết luận chương 38](#_Toc122354285)

[CHƯƠNG 3. THIẾT KẾ THIẾT BỊ 39](#_Toc122354286)

[3.1 Thiết kế phần cứng 39](#_Toc122354287)

[3.1.1 Node đèn chiếu sáng 39](#_Toc122354288)

[3.1.2 Node cảm biến 40](#_Toc122354289)

[3.2 Thiết kế Firmware 42](#_Toc122354290)

[3.2.1 Luồng chạy của node IoT 42](#_Toc122354291)

[3.2.2 Lưu đồ thuật toán của node IoT 42](#_Toc122354292)

[3.3 Thử nghiệm, đánh giá 47](#_Toc122354293)

[3.3.1 Thử nghiệm 47](#_Toc122354294)

[3.3.2 Đánh giá 50](#_Toc122354295)

[Kết luận chương 51](#_Toc122354296)

[KẾT LUẬN 52](#_Toc122354297)

[TÀI LIỆU THAM KHẢO 53](#_Toc122354298)

# DANH MỤC HÌNH ẢNH

[*Hình 1. 1. Cấu trúc liên kết mạng đơn bước* 3](#_Toc122188873)

[*Hình 1. 2 Cấu trúc liên kết mạng nhiều chặng* 3](#_Toc122188874)

[*Hình 1. 3. Kiến trúc mạng trong home automation sử dụng mô-đun wifi ESP8266* 4](#_Toc122188875)

[*Hình 1. 4. Mô hình hệ thống home automation với Home Assistant* 4](#_Toc122188876)

[*Hình 1. 5. Mặt trên máy tính nhúng Orange Pi One* 5](#_Toc122188877)

[*Hình 1. 6. Giao diện điều khiển trên Home Assistant Web Client.* 6](#_Toc122188878)

[*Hình 1. 7. Kiến trúc Core của Home Assistant* 7](#_Toc122188879)

[*Hình 1. 8. Chồng giao thức của Home Assistant* 8](#_Toc122188880)

[*Hình 2. 1. Giao thức MQTT trong chồng giao thức TCP/IP. 17*](#_Toc121592688)

[*Hình 2. 2. Kiến trúc cơ bản của MQTT. 17*](#_Toc121592689)

[*Hình 2. 3. Topic trong MQTT. 18*](#_Toc121592690)

[*Hình 2. 4. QoS 0: At Most Once 20*](#_Toc121592691)

[*Hình 2. 5. QoS 1 – At Least Once 20*](#_Toc121592692)

[*Hình 2. 6. QoS 2 – Exactly Once 21*](#_Toc121592693)

[*Hình 2. 7. Định dạng packet của MQTT. 22*](#_Toc121592694)

[*Hình 2. 8. Bản tin CONNECT trong MQTT 22*](#_Toc121592695)

[*Hình 2. 9. Bản tin PUBLISH trong MQTT 23*](#_Toc121592696)

[*Hình 2. 10. Chế độ mã hóa AES-GCM 29*](#_Toc121592697)

[*Hình 2. 11. Mô hình giao thức ESP-Touch 30*](#_Toc121592698)

[*Hình 2. 12. Dạng tín hiệu ghi bit 0/1, đọc bit, reset 32*](#_Toc121592699)

[*Hình 2. 13. Luồng giao tiếp của 1-Wire 33*](#_Toc121592700)

[*Hình 2. 14. Cấp nguồn trong 1-wire 33*](#_Toc121592701)

[*Hình 2. 15. Cấp nguồn vi sai cho 1-wire 34*](#_Toc121592702)

[*Hình 2. 16. Kết nối cảm biến với MCU qua giao thức 1-Wire 34*](#_Toc121592703)

[*Hình 3. 2. Mạch điều khiển triac* 38](#_Toc122350961)

[Hình 3. 1. Sơ đồ nguyên lý của node đèn chiếu sáng 39](#_Toc122350962)

[*Hình 3. 3. Sơ đồ nguyên lý node cảm biến* 40](#_Toc122350963)

[*Hình 3. 4. Lưu đồ thuật toán khối xử lý chính* 42](#_Toc122350964)

[*Hình 3. 5. Lưu đồ thuật toán khối nút nhấn* 43](#_Toc122350965)

[*Hình 3. 6. Lưu đồ thuật toán khối LED báo trạng thái* 44](#_Toc122350966)

[*Hình 3. 7. Lưu đồ thuật toán khối smart config* 45](#_Toc122350967)

[*Hình 3. 8. Lưu đồ thuật toán khối điều khiển MQTT* 46](#_Toc122350968)

[*Hình 3. 9. Mô hình sơ đồ kết nối node IoT với Home Assistant 47*](#_Toc122350969)

[*Hình 3. 10. Hệ thống sản phẩm thực tế* 47](#_Toc122350970)

# DANH MỤC BẢNG

[Bảng 2. 1. Hoạt động của giao thức 1-wire 31](#_Toc122335195)

[Bảng 3. 1. Các chân của triac BT136 37](#_Toc122335213)

[Bảng 3. 2. So sánh các loại cảm biến 41](#_Toc122335214)

[Bảng 3. 3. Luồng hoạt động của node IoT 42](#_Toc122335215)

[Bảng 3. 4. Trạng thái của linh kiện trong thực tế 48](#_Toc122335216)

[Bảng 3. 5. Kết quả kiểm tra chịu tải của node 50](#_Toc122335217)

# DANH MỤC CHỮ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **Từ viết tắt** | **Từ đầy đủ tiếng anh** | **Dịch nghĩa** |
| AES | Advanced Encryption Standard | Tiêu chuẩn mã hóa tiên tiến |
| AES-CTR | Advanced Encryption Standard – Short For Counter | Tiêu chuẩn mã hóa tiên tiến cho bộ đếm ngắn |
| AES-GCM | Advanced Encryption Standard – Galois/Counter Mode | Tiêu chuẩn mã hóa tiên tiến trong trường Galoa |
| ARP | Address Resolution Protocol | Giao thức phân giải địa chỉ |
| CLI | Command Line | Giao diện dòng lệnh |
| CMOS | Complementary Metal-Oxide-Semiconductor | Công nghệ chế tạo mạch tích hợp |
| CoIOT | Communication For Iot | Giao thức cho Internet of Things |
| DHCP | Dynamic Host Configuration Protocol | Giao thức cấu hình máy chủ |
| DNS | Domain Name System | Hệ thống tên miền |
| ECC | Error Correcting Code | Mã sửa lỗi |
| FCC | Federal Communications Commission | Ủy ban Truyền thông liên bang |
| GPIO | General Purpose Input Output Pins | Chân vào ra dùng chung |
| GPU | Graphics Processing Unit | Bộ vị xử lý đồ họa |
| HA/HASS | Home Assistant | Nền tảng Home Assistant |
| HDMI | High-Definition Multimedia Interface | Giao diện đa phương tiện độ phân giải cao |
| HTTPS | Hypertext Transfer Protocol Secure | Giao thức truyền siêu văn bản có bảo mật |
| IEEE | Institute Of Electrical And Electromic Engineers | Hội Kỹ sư Điện và Điện tử |
| IO | Input Output | Vào ra |
| IoT | Internet Of Things | Vạn vật kết nối |
| IP | Internet Protocol | Giao thức mạng |
| LAN | Local Area Network | Mạng cục bộ |
| LSB | Least Significant Bit | Bit trọng số nhỏ nhất |
| MQTT | Message Queuing Telemetry Transport | Giao thức truyền/nhận bản tin |
| MSB | Most Significant Bit | Bit trọng số lớn nhất |
| NVS | Non-volatile Storage | Lưu trữ không mất dữ liệu |
| OS | Operation System | Hệ điều hành |
| OSI | Open Systems Interconnection | Mô hình tham chiếu kết nối các hệ thống mở |
| QoS | Quality Of Service | Chất lượng dịch vụ |
| RAM | Random Access Memory | Bộ nhớ truy cập ngẫu nhiên |
| RF | Radio Frequency | Tần số sóng radio |
| RSA | Rivest-Shamir-Adleman | Tên ba nhà mật mã học, thuật toán mã hóa công khai |
| SBC | Session Border Controller | Thiết bị triển khai trong VoIP |
| SSH | Secure Socket Shell | Giao thức hỗ trợ quản lý, điều chỉnh server từ xa |
| TCP/IP | Transmission Control Protocol/Internet Protocol | Giao thức điều khiển truyền nhận liên mạng |
| TKPIP | Temporal Key Integrity Protocol | Giao thức toàn vẹn khóa tạm thời |
| TLS/SSL | Transport Layer Security | Bảo mật tầng giao vận |
| SSL | Secure Sockets Layer | Tầng socket bảo mật |
| TTL | Transistor-Transistor Logic | Logic transistor lưỡng cực |
| UDP | User Datagram Protocol | Giao thức dữ liệu người dùng |
| USB | Universal Serial Bus | Truyền dữ liệu nối tiếp |
| UTF-8 | 8-Bit Unicode Transformation Format | Định dạng chuyển đổi unicode 8-bit |
| WEP | Wired Equivalent Privacy | Chuẩn bảo mật cho wi-fi |
| WI-FI | Wireless Fidelity | Giao thức mạng không dây |
| WPA | Wi-Fi Protected Access | Giao thức bảo mật wi-fi |
| WPA2 | Wi-Fi Protected Access | Giao thức bảo mật wi-fi |

# LỜI MỞ ĐẦU

Trong cuộc cách mạng công nghiệp 4.0, Internet ngày càng phát triển mở ra sự phát triển về kết nối cho các thiết bị điện tử, đặc biệt là lĩnh vực tự động hóa các thiết bị trong nhà. Home Automation đang dần có mặt khắp mọi nơi, mọi nhà để phục vụ nhu cầu của người dùng.

Cùng với đó, các nền tảng IoT cũng ngày càng phát triển và cho phép người dùng có thể sử dụng để tương tác với các thiết bị đó. Home Assistant là một nền tảng miễn phí cho phép người dùng tạo, tích hợp, điều kiển thiết bị qua web client hoặc ứng dụng. Tuy nhiên, không phải thiết bị nào cũng tương thích với nền tảng Home Assistant. Xuất phát từ vấn đề khó khăn trên, em đã xây dựng các thiết bị tương thích với Home Assistant. Cho phép người dùng dễ dàng tích hợp và thu thập dữ liệu hoặc điều khiển các thiết bị trong nhà.

Đề tài triển khai xây dựng các node IoT để tương thích với Home Assistant. Các node IoT này có thể được sử dụng trong nhà như đèn thông minh, cảm biến quan trắc môi trường trong nhà. Thuật toán sử dụng trong chương trình sẽ giúp tích hợp tự động các thiết bị một cách tự động.

Trong quá trình thực hiện và triển khai đề tài, em đã nỗ lực, cố gắng hết sức và nhận được sự giúp đỡ từ nhiều phía thầy cô và bạn bè. Em xin gửi lời cảm ơn tới thầy TS. Nguyễn Trung Hiếu, người đã tận tâm, đồng hành cùng em trong suốt quá trình làm cho đến khi hoàn thành đề tài. Cuối cùng em xin cám ơn tất cả các thầy cô đã hướng dẫn, đó chính là nền tảng kiến thức quan trọng hỗ trợ em thực hiện đề tài này. Do thời gian có hạn, kiến thức còn hạn chế, đề tài không tránh khỏi những thiếu sót, rất mong sự đóng góp của các thầy cô giáo.



# CHƯƠNG 1. NGHIÊN CỨU TỔNG QUAN

Chương 1 giới thiệu về đề tài, trình bày về mục tiêu của đề tài nghiên cứu. Đồng thời trình bày các kiến thức cơ bản về hệ thống, chi tiết các thành phần của hệ thống như server, phần mềm quản lý. Từ đó làm tiền đề nghiên cứu cho các chương tiếp theo.

## 1.1 Giới thiệu đề tài

### 1.1.1 Đặt vấn đề, hiện trạng

Home automation hay domotics là [tự động hóa tòa](https://en.wikipedia.org/wiki/Building_automation) nhà cho một [ngôi](https://en.wikipedia.org/wiki/Home) nhà, được gọi là Smart Home hoặc Home automation. Hệ thống [tự động hóa](https://en.wikipedia.org/wiki/Automation) gia đình sẽ giám sát và / hoặc kiểm soát các thuộc tính của ngôi nhà như ánh sáng, khí hậu, hệ thống giải trí và thiết bị. Nó cũng có thể bao gồm [an ninh gia đình](https://en.wikipedia.org/wiki/Home_security) như kiểm soát truy cập và hệ thống báo động. Khi được kết nối với Internet, các thiết bị gia đình là một thành phần quan trọng của Internet [of Things](https://en.wikipedia.org/wiki/Internet_of_Things) ("IoT").

Một hệ thống tự động hóa gia đình thường kết nối các thiết bị được điều khiển với một trung tâm [nhà thông minh trung tâm](https://en.wikipedia.org/wiki/Smart_home_hub) (đôi khi được gọi là "Gateway"). Giao diện [người dùng](https://en.wikipedia.org/wiki/User_interface) để kiểm soát hệ thống sử dụng thiết bị đầu cuối treo tường, máy tính bảng hoặc máy tính để bàn, ứng dụng điện thoại di động hoặc giao diện Web cũng có thể truy cập được bên ngoài trang web thông qua Internet.

Mặc dù có nhiều nhà cung cấp cạnh tranh, nhưng ngày càng có nhiều nỗ lực đối với các hệ thống nguồn mở. Tuy nhiên, có những vấn đề với tình trạng tự động hóa gia đình hiện tại bao gồm thiếu các biện pháp bảo mật được tiêu chuẩn hóa và ngừng sử dụng các thiết bị cũ hơn mà không có khả năng tương thích ngược.

Tự động hóa gia đình có tiềm năng cao trong việc chia sẻ dữ liệu giữa các thành viên trong gia đình hoặc các cá nhân đáng tin cậy để đảm bảo an ninh cá nhân và có thể dẫn đến các biện pháp tiết kiệm năng lượng với tác động tích cực đến môi trường trong tương lai.

Thị trường home automation trị giá 64 tỷ đô la Mỹ vào năm 2022 và dự kiến sẽ tăng lên hơn 163 tỷ đô la vào năm 2028.

Bên cạnh những lợi ích của nhà thông minh, cũng có nhiều nhược điểm của những ngôi nhà thông minh. Giống như những nhược điểm của các thiết bị gia dụng, hệ thống nhà thông minh của cũng có một số nhược điểm:

*Giá thành cao:*

Những vấn đề lớn nhất, nhược điểm hay bất lợi của một hệ thống nhà thông minh là chi phí. Có khá nhiều công ty cung cấp hệ thống nhà thông minh, nhưng tất cả chúng đều khá đắt tiền. Đây là điều mà chỉ một số ít người có thể mua được. Bạn sẽ có thể có một khoản tiết kiệm và thu nhập tốt để cài đặt hệ thống này. Nó sẽ tốn kém trong thời gian đầu nhưng sẽ tiết kiệm chi phí trong tương lai.

*Phụ thuộc vào Internet:*

Yêu cầu cơ bản đối với hệ thống nhà thông minh là internet. Nếu không có kết nối internet tốt và mạnh, bạn sẽ không thể kiểm soát điều này. Nếu không có kết nối internet vì một số lý do, không có cách nào khác để bạn có thể truy cập và kiểm soát hệ thống của mình.

***Phụ thuộc vào các chuyên gia:***

Yêu cầu cơ bản đối với hệ thống nhà thông minh là internet. Nếu không có kết nối internet tốt và mạnh, bạn sẽ không thể kiểm soát điều này. Nếu không có kết nối internet vì một số lý do, không có cách nào khác để bạn có thể truy cập và kiểm soát hệ thống của mình. vì vậy điều rất quan trọng là chọn một nhà cung cấp dịch vụ internet chất lượng tốt ở Kozhikode. Airtel, Jio và Asianet là một số nhà cung cấp dịch vụ tốt.

Trong trường hợp có vấn đề với hệ thống nhà thông minh, bạn không thể chỉ cần gọi một người siêng năng hoặc một người tương tự để sửa chữa hoặc quản lý lỗi. Bạn sẽ phải phụ thuộc vào các chuyên gia. Chỉ có các chuyên gia của công ty mới có thể giúp bạn xử lý các vấn đề. Một chuyên gia chỉ có thể chăm sóc các công nghệ nhà thông minh của bạn dẫn đến sự phụ thuộc nhiều vào các chuyên gia.

Đó là một trong những **nhược điểm lớn nhất của Home Automation.**

### 1.1.2 Mục tiêu đề tài

Trong mục trên đã đề cập đến các nhược điểm trong hệ thống Home Automation.

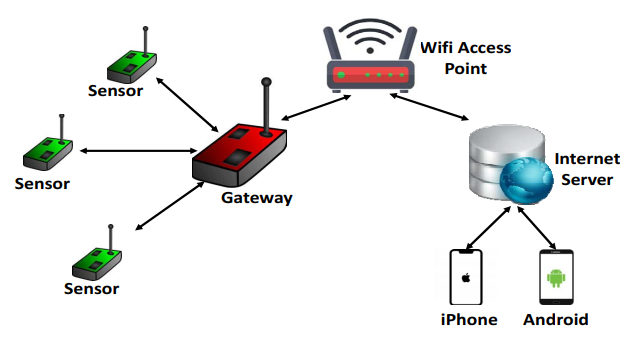
Đề tài sẽ đưa ra một hệ thống điều khiển các thiết bị trong nhà thông qua mạng không dây cục bộ (LAN) với server là Home Assistant được cài đặt trên máy tính nhúng Orange Pi One. Các node IoT với vi điều khiển ESP32, giao tiếp với server sử dụng giao thức MQTT.

Đề tài này sẽ tập trung giải quyết 3 nhược điểm: giá thành cao, phụ thuộc Internet và phụ thuộc công nghệ.

Em sẽ thiết kế các node với chi phí thấp, thông dụng, hoạt động ổn định. Các node có chức năng: tự động kết nối với server, cấu hình tự động, giao tiếp không dây, gửi/nhận lệnh điều khiển/phản hồi.

1.1.3 Mô hình, mô tả hệ thống

Trong đề tài, em đề xuất một mô hình không sử dụng các node relay thay vì sử dụng mô hình truyền thống sử dụng cấu trúc liên kết mạng đơn bước hoặc đa bước.



*Hình 1. 1. Cấu trúc liên kết mạng đơn bước*

Diagram

Description automatically generated

*Hình 1. 2 Cấu trúc liên kết mạng nhiều chặng*

Diagram

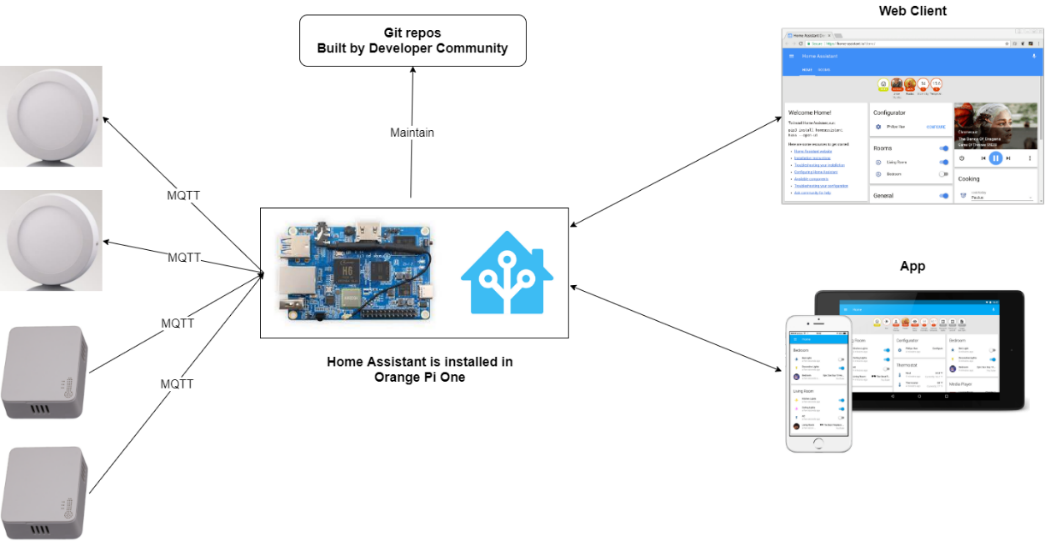
Description automatically generated

*Hình 1. 3. Kiến trúc mạng trong home automation sử dụng mô-đun wifi ESP8266*

Trong hình 1.1 mô tả cấu trúc liên kết mạng đơn bước [1]. Tất cả các nút cảm biến đều có thể giao tiếp trực tiếp với Gateway

Trong hình 1.2 mô tả cấu trúc liên kết mạng nhiều chặng [1]. Trong mô hình này cần có nút chuyển tiếp để chuyển tiếp các gói từ các cảm biến ra khỏi phạm vi vô tuyến của cổng.

Trong hình 1.3 mô tả cấu trúc mạng không yêu cầu cổng và nút chuyển tiếp [1]. Mỗi cảm biến được trang bị ESP8266 tải trực tiếp dữ liệu của nó lên máy chủ.



*Hình 1. 4. Mô hình hệ thống home automation với Home Assistant*

Hệ thống gồm các khối như sau:

*Khối 1: Bao gồm 4 node IOT có 2 node đèn chiếu sáng và 2 node cảm biến.*

*Khối 2: Sử dụng local server là Orange Pi One cài nền tảng Home Assistant*

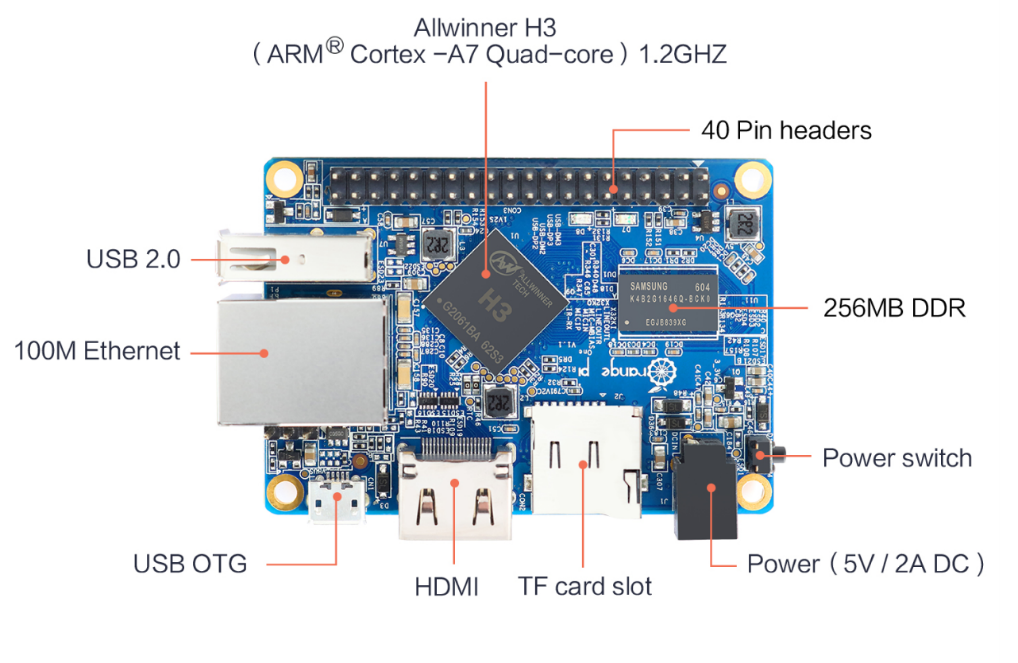
*Khối 3: Home Assistant web client và mobile app.*

Hoạt động của hệ thống: Home Assistant là trung tâm gửi/nhận lệnh điều khiển từ thiết bị và ứng dụng web, app. Node giao tiếp với Home Assistant qua giao thức MQTT sử dụng công nghệ kết nối Wi-Fi. Ứng dụng giao tiếp với Home Assistant qua giao thức HTTPS.

## 1.2 IoT Gateway

### 1.2.1 Máy tính nhúng Orange Pi One

**Orange Pi One là một máy tính nhúng đơn mã nguồn mở. Nó có thể chạy Android 4.4, Ubuntu, Debian, Rasberry Pi Image.** Orange Pi One chạy trên vi xử lý lõi tứ Allwinner H3 tốc độ 1,2Ghz, nhân Cortex A7, GPU Mali-400MP2, RAM 512MB, có kèm khe cắm thẻ nhớ mở rộng cùng nhiều cổng kết nối quan trọng như ethernet, HDMI, USB OTG và USB 2.0.



*Hình 1. 5. Mặt trên máy tính nhúng Orange Pi One*

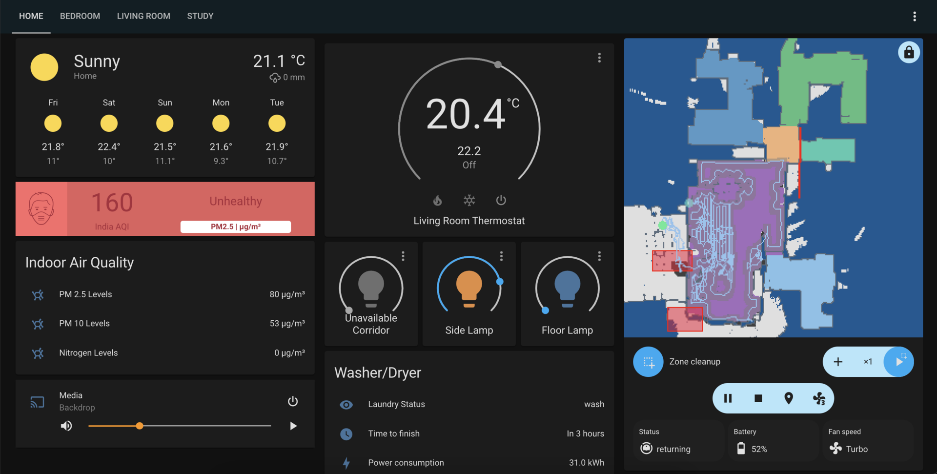
Orange Pi One được tích hợp khá nhiều cổng kết nối trong đó có một cổng HDMI, cổng Ethernet 10/100, USB 2.0, microUSB, cổng kết nối camera, màn hình và đầy đủ chân cắm GPIO như một vi mạch điều khiển thông thường. Máy hỗ trợ sạc thông qua bộ nguồn 5V 2A và không hỗ trợ sạc qua cổng microUSB.

Giống như một số model máy tính Orane Pi khác, model Orange Pi One sẽ có thể chạy được trên nền tảng Android hoặc một số nền tảng khác dựa trên nhân Linux bao gồm: Raspbian, Ubuntu và Debian.

## 1.3 Nền tảng Home Assistant

### 1.3.1 Giới thiệu

Home Assistant còn được gọi là “HA” hay “HASS” là một nền tảng quản lý nhà thông minh được lập trình bằng ngôn ngữ Python. Nó có thể chạy trên mọi nền tảng hệ điều hành và quản lý ngôi nhà thông minh qua giao diện web hay qua ứng dụng trên smartphone. Home Assistant có 2 phiên bản. Phiên bản “Home Assistant” hay "Home Assistant Core" là thành phần cốt lõi nhất, có thể cài đặt lên bất kỳ nền tảng hệ điều hành nào giống như một phần mềm máy tính.



*Hình 1. 6. Giao diện điều khiển trên Home Assistant Web Client.*

Là phần mềm nguồn mở với sự hỗ trợ của đội ngũ kỹ sư, lập trình viên trên khắp thế giới, Home Assistant tương thích với hầu hết mọi thiết bị nhà thông minh, mở ra khả năng làm việc không giới hạn của các thiết bị trong ngôi nhà.

“Home Assistant OS” kết hợp giữa “Home Assistant Core” và các công cụ khác. Phiên bản này có thể cài đặt lên một chiếc máy tính như Raspberry Pi, máy ảo. Hai phiên bản này khi cài lên thiết bị nào đó sẽ biến thiết bị đó thành một hub tổng giúp kết nối và điều khiển thiết bị nhà thông minh, tương tự như Gateway trên nền tảng Xiaomi, Aqara, Hub tổng trên nền tảng Samsung SmartThings.

Dù là sử dụng phiên bản nào, người dùng cũng phải cài đặt Home Assistant trước, sau đó hệ thống sẽ quét qua các thiết bị nhà thông minh hiện có, người dùng sẽ tiến hành cấu hình để các thiết bị làm việc theo nhu cầu.

Như vậy, nếu xét về sự tiện lợi, nhanh chóng, các nền tảng nhà thông minh đã giới thiệu trong các bài viết trước tỏ ra vượt trội, trong khi đó, Home Assistant lại mạnh ở khả năng điều khiển trong mạng nội bộ và tùy biến mạnh mẽ.

### 1.3.2 Kiến trúc của Home Assistant

Home Assistant cung cấp một nền tảng để điều khiển nhà và tự động hóa nhà. Home Assistant không chỉ là một ứng dụng: đó là một hệ thống nhúng cung cấp trải nghiệm giống như các sản phẩm tiêu dùng khác có sẵn: tích hợp, định cấu hình và cập nhật đều được thực hiện thông qua một giao diện dễ sử dụng.

* **Home Assistant Operation System** cung cấp môi trường Linux tối thiểu để chạy Supervisor và Core.
* **Supervisor** quản lý hệ thống điều hành.
* **Core**tương tác với người dùng, người giám sát và các thiết bị & dịch vụ IoT.

Người dùng có các yêu cầu khác nhau cho những gì họ muốn từ một nền tảng tự động hóa gia đình. Đó là lý do tại sao chỉ có thể chạy một phần của ngăn xếp Home Assistant.

**Core Architecture**

Home Assistant Core bao gồm bốn phần chính. Trên hết, nó bao gồm nhiều lớp trợ giúp để xử lý các tình huống phổ biến, như cung cấp một thực thể hoặc xử lý các vị trí.

* **Event Bus:** tạo điều kiện kích hoạt và lắng nghe các sự kiện.
* **State Machine:** theo dõi các trạng thái của sự vật và kích hoạt sự kiện state\_changed khi một trạng thái đã bị thay đổi.
* **Service Registry:** lắng nghe trên bus sự kiện cho các sự kiện call\_service và cho phép các mã khác đăng ký dịch vụ.
* **Timer:** gửi sự kiện time\_changed cứ sau 1 giây trên bus sự kiện.

Graphical user interface, diagram

Description automatically generated

*Hình 1. 7. Kiến trúc Core của Home Assistant*

### 1.3.3 Giao tiếp với thiết bị trên Home Assistant

Các phương pháp xử lý phổ biến giải quyết các khía cạnh của kiến trúc HA đòi hỏi một cách tiếp cận tiêu chuẩn trên toàn bộ hệ thống. Đây là một nhiệm vụ quan trọng vì nó góp phần vào sự gắn kết kỹ thuật tổng thể của hệ thống và làm rõ cách thức và địa điểm xử lý được thực hiện.

Graphical user interface, diagram

Description automatically generated

*Hình 1. 8. Chồng giao thức của Home Assistant*

Có nhiều cách để kết nối với các thiết bị. Nhiều thiết bị kết nối với mạng, và các giao thức khác nhau trên đó (MQTT, HTTP / REST, CoIOT, v.v). những thiết bị khác kết nối qua RF (Zigbee, Zwave, Insteon, Bluetooth, v.v.), có những thiết bị khác kết nối qua USB hoặc các kết nối có dây khác. Các giao thức trên ở tầng Transport của chồng giao thức trong Home Assistant. Trong đề tài này, em sử dụng giao thức MQTT để giao tiếp giữa thiết bị và Home Assistant. Giao thức MQTT sẽ được trình bày chi tiết trong chương 2.

### 1.3.4 Ưu điểm của Home Assistant

Một trung tâm tự động hóa gia đình cục bộ như Trợ lý gia đình có một số lợi ích so với các thiết bị được kết nối với đám mây. Dưới đây là một số ưu điểm của Trợ lý gia đình để tự động hóa gia đình.

*Tốc độ:* Khi bạn kích hoạt một hoạt động hoặc đưa ra lệnh, tất cả quá trình xử lý xảy ra trong nhà của bạn với tư cách là trợ lý gia đình là bộ điều khiển cục bộ. Sẽ không có gì được gửi qua máy chủ web. Ngoài ra, bạn sẽ không có quá nhiều thiết bị cạnh tranh về băng thông của mạng gia đình của mình. Các thiết bị thông minh có thể được giữ trong các mạng con và chỉ kết nối các tiện ích đang tương tác trực tiếp.

*Khả năng tương thích:* Bạn không phải sợ công ty như Amazon cấm các thiết bị Google của bạn nếu bạn không sử dụng dịch vụ đám mây của họ để bắt đầu.

*Bảo mật:* Khi có ít thiết bị được kết nối với internet, sẽ có ít thiết bị hơn cho phép internet vào mạng của bạn. Bạn được phép quyết định những gì có thể truy cập dữ liệu của bạn và cách chúng tương tác.

*Kiểm soát:* Đám mây chỉ cung cấp các dịch vụ hạn chế cho người dùng mà chủ sở hữu của họ muốn. Sử dụng các trung tâm và bộ điều khiển cục bộ, bạn có thể tự do quyết định phải làm gì với nó. Bạn có thể có toàn quyền kiểm soát các thiết bị và tự động hóa và tham gia sâu như bạn muốn.

*Độ tin cậy:* Điều gì sẽ xảy ra nếu dịch vụ đám mây ngừng hoạt động? Hoặc nếu các máy chủ chỉ gặp sự cố khiến hàng nghìn người dùng bị khóa cửa hoặc không có AC vì trung tâm tự động hóa gia đình không hoạt động? Điều gì sẽ xảy ra nếu chủ sở hữu dịch vụ ngừng hỗ trợ thiết bị cụ thể của bạn? Hãy tưởng tượng bạn mất internet? Nhiều trường hợp dẫn đến việc các thiết bị được kết nối với đám mây của bạn ngoại tuyến. Nhưng nếu bạn sử dụng một trung tâm tự động hóa gia đình cục bộ để bảo trì và điều khiển thiết bị của mình, bạn không phải lo lắng về những vấn đề như vậy.

*Miễn phí:* Home Assistant là một nguồn miễn phí.

## Kết luận chương

Chương 1 đã trình bày về mục đích, mục tiêu nghiên cứu của đề tài là giải quyết 3 vấn đề nhược điểm của một hệ thống home automation hiện nay là bảo mật, phức tạp tích hợp và phụ thuộc vào các chuyên gia. Chương này cũng trình bày về hệ thống được triển khai trong đề tài, nền tảng quản lý nhà tự động Home Assistant.

# CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Chương này đồ án nghiên cứu về mô hình mạng OSI và TCP/IP. Đồng thời nghiên cứu hoạt động, bảo mật, ưu điểm, nhược điểm của công nghệ kết nối wi-fi, các giao thức truyền thông có dây và không dây mà node IoT sử dụng để truyền nhận dữ liệu với Home Assistant và các cảm biến. Từ đó, làm tiền đề cho việc thiết kế node IoT được trình bày ở chương 3.

## 2.1 Mô hình mạng OSI và TCP/IP

Mô hình OSI (Open system interconnection – Mô hình kết nối các hệ thống mở) là một cơ sở dành cho việc chuẩn hoá các hệ thống truyền thông, nó được nghiên cứu và xây dựng bởi ISO (International Organization for Standardization). Đến năm 1984, mô hình tham chiếu OSI chính thức được đưa ra giới thiệu, được chia thành 7 lớp:

**Họ giao thức TCP**

TCP/IP Transmission Control Protocol (giao thức điều khiển truyền thông)/ Internet Protocol (Giao thức Internet) là bộ giao thức cho phép kết nối các hệ thống mạng không đồng nhất với nhau. Hai mục tiêu chính cần đạt được trong quá trình thiết kế của giao thức TCP/IP:

Cho phép truyền thông qua các đường dây của mạng rộng 8 (Wide Area Network – WAN).

Cho phép truyền thông giữa các môi trường đa dạng. TCP/IP sử dụng mô hình truyền thông 4 tầng hay gọi là truyền hình DoD (mô hình của bộ quốc phòng Mỹ) TCP/IP được xem là giản lược của mô hình tham chiếu OSI, các tầng trong mô hình này là (theo thứ tự từ trên xuống):

- Tầng ứng dụng (Application Layer).

- Tầng giao vận (Transport Layer).

- Tầng mạng (Internet Layer).

- Tầng liên mạng (Network Interface Layer).

Cũng tương tự như trong mô hình OSI, TCP/IP khi truyền dữ liệu, quá trình tiến hành từ tầng trên xuống tầng dưới, qua mỗi tầng dữ liệu được thêm vào một thông tin điều khiển được gọi là phần header. Khi nhận dữ liệu thì quá trình xảy ra ngược lại, qua mỗi tầng thì phần header tương ứng được lấy đi và khi đến tầng trên cùng thì dữ liệu không còn phần header nữa.

### 2.1.1 Tầng vật lý

Lớp vật lý (viết tắt là PHY) đề cập đến mạch cần thiết trong hệ thống để kết nối thiết bị lớp liên kết (viết tắt là MAC để kiểm soát truy cập trung bình) với vật lý. Môi trường vật lý có thể là cáp đồng, cáp quang hoặc các loại tín hiệu RF trung bình khác.

Đối với thiết kế ngăn xếp UDP / IP, chúng tôi đã sử dụng giao diện độc lập với phương tiện (MII) để kết nối với phương tiện vật lý của chúng tôi, cáp Ethernet. Chúng ta sẽ nói về MIIinterfaces và một số biến thể của nó, tập trung đặc biệt vào giao diện độc lập phương tiện giảm (RGMII) được sử dụng trong Dự án.

### 2.1.2 Tầng liên kết dữ liệu

Lớp này chịu trách nhiệm cung cấp khung hình giữa các nút khác nhau trên cùng một cấp độ của mạng. Lớp liên kết dữ liệu có hai lớp con, điều khiển liên kết logic (LLC) và điều khiển truy cập phương tiện (MAC).

* **Kiểm soát liên kết logic (LLC):** LLC được sử dụng để ghép các giao thức của lớp trên. Nó có thể thực hiện nhiều hành động hơn như kiểm soát luồng, thông báo lỗi, xác nhận, v.v. Nó được định nghĩa trong IEEE 802.2 [8].
* **Kiểm soát truy cập trung bình (MAC):**  Lớp con kiểm soát truy cập trung bình (còn được gọi là lớp con điều khiển truy cập phương tiện) cũng cung cấp tính năng ghép kênh và kiểm soát luồng, nhưng trong trường hợp này là phương tiện truyền vật lý. Như đã được giải thích, khối MAC thường được kết nối với PHY thông qua khối MII. Khi truyền dữ liệu, MAC sửa đổi các khung thành các khung phù hợp với phương tiện vật lý. Nó thêm phần mở đầu cho đồng bộ đồng hồ, đệm, nhận dạng lỗi FCS, v.v. MAC sau đó sẽ gửi dữ liệu khi được phép. Lớp con này cũng chịu trách nhiệm truyền lại các gói nếu phát hiện tắc nghẽn, va chạm hoặc gây nhiễu.

Khi nhận dữ liệu, MAC phải kiểm tra tính toàn vẹn của dữ liệu (với CRC). Nó cũng phải loại bỏ phần mở đầu và đệm, trước khi cho phép dữ liệu đạt được các giao thức cao hơn [2]. Ngăn xếp đã được thiết kế xung quanh giao thức Ethernet và dự án đã được tạo cho kết nối Ethernet có dây. IEEE 802.3 [3] xác định lớp vật lý và điều khiển truy cập phương tiện cho Ethernet có dây.

### 2.1.3 Tầng mạng

Lớp này nhóm tập hợp các giao thức phụ trách các gói mạng truyền tải từ một nguồn cụ thể đến một máy chủ đích. Quá trình này có thể xảy ra mặc dù một mạng duy nhất hoặcmultiple những cái. Một số chức năng của lớp này là:

* **Giao tiếp không kết nối:** Không cần ACKs; kết nối theo định hướng truyền thông được để lại cho các lớp cao hơn.
* **Địa chỉ máy chủ**: Cần có một hệ thống nhận dạng trong mạng (ví dụ: địa chỉ IP).
* **Chuyển tiếp tin nhắn:** Để chuyển tiếp các gói tin giữa các mạng khác nhau. Giao thức mạng chính của ngăn xếp của chúng ta là giao thức IP, cụ thể nó được sử dụng giao thức IPv4. Không cần thiết phải sử dụng IPv6, vì ngăn xếp được thiết kế để sử dụng trong các mạng LAN vừa hoặc nhỏ không yêu cầu số lượng địa chỉ đó và thường vẫn sử dụng IPv4. Giao thức ICMP được gói gọn bên trong IPv4, nhưng nó không được coi là một giao thức truyền tải vì nó hoạt động như một phần bổ sung cho các tin nhắn IP. Cả hai giao thức đều được giải thích bên dưới.

### 2.1.4 Tầng giao vận

Lớp truyền tải được thiết lập để cung cấp giao tiếp giữa máy chủ với máy chủ. Lớp này có thể thực hiện các chức năng, như giao tiếp hướng kết nối, kiểm soát luồng hoặc ghép kênh các giao thức lớp cao hơn (ứng dụng hoặc dịch vụ). Nó là lớp tập trung cao nhất vào độ tin cậy. Các chức năng chính của nó là:

* **Giao tiếp hướng kết nối:** TCP được biết đến là có nó. UDP là giao thức truyền tải thay thế thực sự vẫn không có kết nối.
* **Giao hàng giống nhau:** Lớp mạng không đảm bảo rằng điều này được giải quyết ở đây.
* **Độ tin cậy:** Kỹ thuật phát hiện lỗi, xác nhận, sơ đồ yêu cầu lặp lại tự động, v.v. Để tránh mất các gói trong quá trình networkcongestions hoặc các vấn đề khác.
* **Kiểm soát luồng: Kiểm soát** giữa hai nút để ngăn người gửi quá nhanh so với bộ đệm nhận.
* **Kiểm soát tắc nghẽn:** Để tránh sự sụp đổ của mạng. Những thứ như giảm tốc độ gửi gói và khởi động chậm, v.v.
* **Ghép kênh:** Tùy chọn ghép kênh giữa các nút lớp cao hơn có sẵnnhờ các cổng. Giao thức được sử dụng nhiều nhất ở lớp truyền tải trong bộ Internet là TCP, mặc dù vậy, ngăn xếp đã được thiết kế đặt tốc độ trước độ tin cậy và điều này làm cho nó thích hợp hơn để sử dụng giao thức UDP. Cả hai giao thức sẽ được giải thích tiếp theo, nhưng trọng tâm sẽ nằm ở giao thức UDP, vì đó là nơi chúng tôi thực sự quan tâm.

### 2.1.5 Tầng ứng dụng

Lớp cuối cùng này nhóm các giao thức phụ trách giao tiếp giữa tiến trình với quy trình. Ở đây sẽ mô tả tất cả các dịch vụ sẽ sử dụng ngăn xếp được thiết kế để truyền và nhận dữ liệu. Trong thiết kế, các dịch vụ của lớp ứng dụng được tạo dưới dạng các lõi IP riêng biệt, sau đó sẽ được kết nối với ngăn xếp UDP / IP để sử dụng. Ngoại lệ duy nhất cho điều đó là giao thức DHCP. Giao thức DHCP là cần thiết cho một hiệu suất tốt của ngăn xếp bên trong mạng, giao thức này là một phần không thể thiếu của lõi IP ngăn xếp. Ngăn xếp đã được thiết kế để sử dụng dễ dàng bên trong một mạng không được kiểm soát, nó phải là "plug and play". Điều này làm cho giao thức bắt buộc DHCP, vì không muốn cần phải định cấu hình địa chỉ của các thiết bị đang kết nối.

## 2.2 Công nghệ kết nối wi-fi

### 2.2.1 Tổng quan về wi-fi

Wi-fi (Wireless Fidelity) hay IEEE 802.11 là một tập các chuẩn của tổ chức IEEE (Institute of Electrical and Electromic Engineers) bao gồm các đặc tả kỹ thuật liên quan đến hệ thống mạng không dây. Chuẩn IEEE 802.11 mô tả một giao tiếp truyền qua không khí (Over-The-Air) sử dụng sóng vô tuyến để truyền nhận tín hiệu giữa một thiết bị không dây và tổng đài hoặc điểm truy cập (Access Point), hoặc giữa 2 hay nhiều thiết bị không dây với nhau (mô hình ad-hoc).

### 2.2.2 Lịch sử ra đời

Năm 1985, Ủy ban Liên lạc Liên bang Mỹ FCC (Federal Communications Commission) quyết định mở cửa một số băng tần của giải sóng không dây, cho phép sử dụng chúng mà không cần giấy phép của chính phủ. Các giải sóng này, gọi là các “băng tần rác” (900 MHz, 2.4 GHz, 5.8 GHz), được phân bố cho các thiết bị sử dụng vào các mục đích ngoài liên lạc.

Nǎm 1988, công ty NCR, vì muốn sử dụng dải tần "rác" để liên thông các máy rút tiền qua kết nối không dây, đã yêu cầu một kỹ sư của họ có tên Victor Hayes tìm hiểu việc thiết lập chuẩn chung. Ông này cùng với chuyên gia Bruce Tuch của Trung tâm nghiên cứu Bell Labs đã tiếp cận với tổ chức IEEE, nơi mà một tiểu chuẩn mới tên 802.3 đã xác lập ra chuẩn mạng cục bộ Ethernet phố biến hiên nay. Một tiêu chuẩn mới có tên 802.11 đã ra đời và quá trình thương lượng hợp nhất các tiêu chuẩn bắt đầu.

Nǎm 1997, tiểu ban này đã phê chuẩn một bộ tiêu chí cơ bản, cho phép mức truyền dữ liệu 2Mbps, sử dụng một trong 2 công nghệ dải tần rộng tránh nhiễu bằng cách chuyển đổi liên tục giữa các tần số radio và phát tín hiệu trên một dải tần gồm nhiều tần số. Chuẩn mới chính thức được ban hành năm 1997 và các kỹ sư ngay lập tức bắt đầu nghiên cứu một thiết bị mẫu tương thích với nó và liên tục cải tiến và tạo ra các chuẩn Wi-Fi mới như 802.11b/a/g/n/ac…v.v. Chi tiết về một số chuẩn Wi-Fi như 802.11 b/a/g tham khảo tại tài liệu [8].

### 2.2.3 Các chuẩn wi-fi

#### 2.2.3.a Chuẩn IEEE 802.11b

IEEE đã mở rộng chuẩn 802.11 gốc vào tháng 7 năm 1999, đó chính là chuẩn 802.11b. Chuẩn này hỗ trợ băng thông lên đến 11Mbps, tương quan với Ethernet truyền thống.

802.11b sử dụng tần số vô tuyến 2.4 GHz giống như chuẩn ban đầu 802.11. Các thiết bị 802.11b có thể bị xuyên nhiễu từ các thiết bị điện thoại không dây (kéo dài), hoặc các module Zigbee sử dụng tần số 2.4 GHz – theo tài liệu [9]. Mặc dù vậy, bằng cách cài đặt các thiết bị 802.11b cách xa các thiết bị như vậy có thể giảm được hiện tượng xuyên nhiễu này.

* Ưu điểm của 802.11b: giá thành thấp nhất; phạm vi tín hiệu lớn.
* Nhược điểm của 802.11b: tốc độ tối đa thấp nhất; các ứng dụng của gia đình có thể xuyên nhiễu.

#### 2.2.3.b Chuẩn IEEE 802.11a

Trong khi 802.11b vẫn đang được phát triển, IEEE đã tạo một mở rộng thứ cấp cho chuẩn 802.1 có tên gọi là 802.11a. Do giá thành cao hơn nên 802.11a chỉ được sử dụng trong các mạng doanh nghiệp còn 802.11b thích hợp hơn với thị trường mạng gia đình.

Tiêu chuẩn 802.11a hỗ trợ băng thông lên đến 54Mbps và sử dụng tần số vô tuyến 5GHz. Tần số của 802.11a cao hơn so với 802.11b chính vì vậy đã làm cho phạm vi của hệ thống này hẹp hơn so với các mạng 802.11b do độ suy hao lớn hơn.

* Ưu điểm của 802.11a: tốc độ cao; tần số 5GHz tránh được sự xuyên nhiễu từ các thiết bị khác.
* Nhược điểm của 802.11a: giá thành đắt; phạm vi hẹp.

#### 2.2.3.c Chuẩn IEEE 802.11g

Năm 2003, chuẩn Wi-Fi thế hệ thứ 3 ra đời được đặt là chuẩn 802.11g. chuẩn 802.11g được xem là kết hợp giữa chuẩn a và b trước kia, với giá thành khá rẻ (tuy có phần đắt hơn chuẩn b).

Chuẩn 802.11g hỗ trợ tốc độ đến 54 Mbps như chuẩn a nhưng sử dụng băng tần 2.4 GHz như chuẩn b, vì vậy chuẩn này có tốc độ cao, phạm vi tín hiệu tốt. tuy nhiên chuẩn này cũng có nhược điẻm như chuẩn b là dễ bị nhiễu từ các thiết bị phát sóng khác. Do sự giống nhau về thông số, chuẩn kết nối 802.11g có khả năng tương thích ngược với chuẩn 802.11b và ngược lại.

* Ưu điểm của 802.11g: tốc độ cao, phạm vi tín hiệu rộng.
* Nhược điểm của 802.11g: giá thành đắt hơn 802.11b; các thiết bị có thể bị xuyên nhiễu từ nhiều thiết bị khác sử dụng cùng băng tần 2.4GHz.

#### 2.2.3.d Chuẩn IEEE 802.11n

Đây là chuẩn được thiết kế để cải thiện cho 802.11g trong tổng số băng thông được hỗ trợ bằng cách tận dụng nhiều tín hiệu không dây và các anten (công nghệ MIMO).

Khi chuẩn này được đưa ra, các kết nói 802.11n sẽ hỗ trợ tốc độ dữ liệu lên đến 100 Mbps. 802.11n cũng cung cấp phạm vi bao phủ tốt hơn so với các chuẩn Wi-Fi trước đó nhờ cường độ tín hiệu mạnh của nó. Thiết bị 802.11n tương thích với các thiết bị 802.11g.

* Ưu điểm của 802.11n là tốc độ nhanh hơn chuẩn b, g. Vùng phủ sóng tốt nhất, trở kháng lớn hơn để chống nhiễu từ các tác động của môi trường.
* Nhược điểm của 802.11n là giá cao hơn 802.11g, sử dụng nhiều luồng tín hiệu có thể gây nhiễu với các thiết bị 802.11b/g kế cận.

### 2.2.4 Bảo mật của wi-fi

Không giống như mạng có dây, đối với mạng không dây đặc biệt là Wi-Fi vấn đề bảo mật là vấn đề hết sức quan trọng khi mà dữ liệu hay thông tin được phát ra môi trường và bất cứ ai trong vùng phủ sóng cũng có khả năng truy cập được thông tin.

#### 2.2.4.a Phương thức bảo mật WEP

WEP (wired equivalent privacy) nghĩa là bảo mật tương đương với mạng có dây (Wirec LAN). Khái niệm này là một phần trong chuẩn IEEE 802.11. theo định nghĩa, WEP được thiết kế để đảm bảo tính bảo mật cho mạng không dây đạt mức độ nhưu mạng nối cáp truyền thống.

WEP cung cấp bảo mật cho dữ liệu trên mạng không dây qua phương thức mã hóa sử dụng thuật toán đỗi xứng RC4, được Ron Rivest - thuộc hãng RSA Security Inc phát triển. thuật toán RC4 cho phép chiều dài của khóa thay đổi và có thể lên đến 256 bit. Chuẩn 802.11 đòi hỏi bắt buộc các thiết bị WEP phải hỗ trợ được chiều dài khóa tối thiểu là 40 bit, đồng thời đảm bảo tùy chọn hỗ trợ cho các khóa dài hơn. Đa số các thiết bị không dây hỗ trợ wep với ba chiều dài khóa: 40-bit, 64-bit và 128 bits.

Nhược điểm lớn nhất của wep là sử dụng các khóa mã hóa tĩnh. Khi thiết lập cơ chế wep cho router, một khóa được sử dụng cho mọi thiết bị trên mạng để mã hóa tất cả gói tin truyền tải. nhưng các gói đã mã hóa này không tránh được hiện tượng bị chặn lại, nếu có thể chặn đủ số lượng gói tin đã mã hóa sẽ tìm ra được khóa giải mã là gì.

#### 2.2.4.b Phương thức bảo mật WPA

Nhận thấy được những nhược điểm không thể khắc phục của WEP, Wi-Fì Alliance đã đưa ra giải pháp khác gọi là Wi-Fi Protected Access (WPA) vào năm 2003, một năm tuóc khi WEP bị loại bỏ.

Một trong những đổi thay lớn được tích hợp vào WPA bao gồm khả năng kiểm tra tính toàn vẹn của gói tin (message integrity check) để xem liệu gói tin có bị thu thập hay thay đổi gói tin truyền qua lại giữa điểm truy cậpvà thiết bị dùng wi-fi hay không. Ngoài ra còn có giao thức khóa toàn ven thời gian (Temporal Key Integrity Protocol - TKIP). TKIP sừ dụng hệ thống mã hóa thay đổi cho từng gói, an toàn hơn rất nhiều so với mã hóa tĩnh – một khóa cho tất cả các gói của wep. Wpa cũng sử dụng thuật toán RC4 như WEP, nhưng mã hóa đầy đủ 128 bit.

Có một lỗ hổng trong wpa và lỗi này chỉ xảy ra với wpa personal. Khi mà sử dụng hàm thay đổi khóa TKIP được sử dụng để tạo ra các khóa mã hóa bị phát hiện, nếu hacker có thể đoán được khóa khởi tạo hoặc một phần của mật khẩu, họ có thể xác định được toàn bộ mật khẩu, do đó có thể giải mã được dữ liệu. tuy nhiên, lỗ hổng này cũng sẽ bị loại bỏ bằng cách sử dụng những khóa khởi tạo không dễ đoán (ví dụ như “PASSWORD” hoặc “12345678”).

Điều này cũng có nghĩa rằng kĩ thuật TKIP của WPA chỉ là giải pháp tạm thời, chưa cung cấp một phương thức bảo mật cao nhất. WPA chỉ thích hợp với những công ty mà không truyền dữ liệu “mật” về thương mại, hay các thông tin nhạy cảm… WPA cũng thích hợp với những hoạt động hàng ngày và mang tính thử nghiệm công nghệ.

#### 2.2.4.c Phương thức bảo mật WPA2

Đến năm 2006, WPA được thay thế bằng chuẩn mưới là WPA2. Những thay đổi đáng kể nhất của WPA2 so với các phương thức bảo mật trước là WPA2 sử dụng 1 thành phần mới thay thế cho TKIP là có tên CCMP, đồng thời WPA2 yêu cầu phải sử dụng thuật toán mã hóa mạnh mẽ được gọi là chuẩn mã hóa nâng cao hay AES. AES mã hóa đối xứng theo khối Rijindael, sử dụng khối mã hóa 128 bits, 192 bits hoặc 256 bits. Quá trình mã hóa được thực hiện trong các phần cứng.

Tuy nhiên TKIP vẫn còn lưu giữ trong WPA2 như một hệ thống dự phòng và khả năng tương tác với WPA.

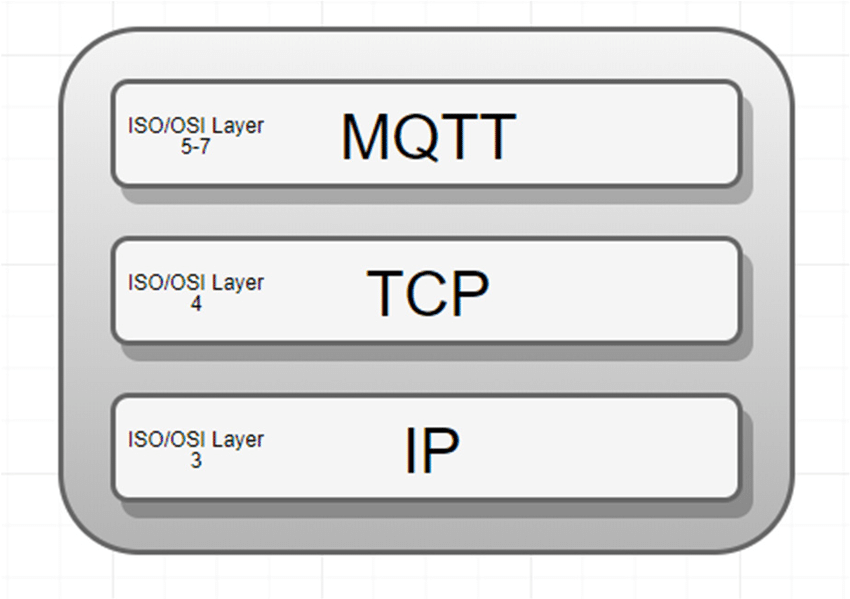
## 2.3 Các giao thức truyền thông

### 2.3.1 Giao thức MQTT

Phần này trình bày giao thức MQTT là giao thức truyền thông IoT được quyết định sử dụng trong giải pháp của em.

#### 2.3.1.a Giới thiệu

MQTT hoặc Message Queuing Telemetry Transport, được phát minh bởi DrAndy Stanford-Clark của IBM và Arlen Nipper của Arcom (nay là Eurotech), vào năm 1999 và bây giờ nó là một tiêu chuẩn mở. Kể từ đó, nó đã trở thành một trong những giao thức tiêu chuẩn trong Internet of Things (IoT) và nó được áp dụng rộng rãi bởi nhiều ngành công nghiệp. Như đã đề cập trước đây, giao thức IoT này được thiết kế đặc biệt cho các thiết bị bị hạn chế như cảm biến. Do đó, MQTT có dấu chân cực kỳ nhỏ và băng thông tối thiểu, điều này làm cho nó rất thuận tiện cho các ứng dụng IoT.



*Hình 2. 1. Giao thức MQTT trong chồng giao thức TCP/IP.*

MQTT là một giao thức nhắn tin dựa trên đăng ký xuất bản, rất đơn giản và nhẹ. Nó là một giao thức lớp ứng dụng hoạt động trên cùng của ngăn xếp TCP / IP, như nó được thể hiện trong hình 2.1. Các ý tưởng chính của thiết kế của nó là giảm băng thông mạng cũng như các yêu cầu về tài nguyên thiết bị, đồng thời cố gắng cung cấp độ tin cậy và mức độ đảm bảo phân phối [6].

Diagram, schematic

Description automatically generated

Hình 2. . *Kiến trúc cơ bản của MQTT.*

MQTT có một mô hình máy chủ khách hàng trong đó mọi thiết bị IoT là một máy kháchmà kết nối với một máy chủ, tức là nhà môi giới MQTT. Một khách hàng, tức là thiết bị IoT, sẽ kết nối với nhà môi giới và xuất bản dữ liệu cho một chủ đề. Một thuê bao sẽ nhận được tin nhắn được xuất bản bởi một khách hàng nếu nó được đăng ký vào chủ đề đó. Một đơn giản của kiến trúc này có thể được nhìn thấy trong hình 2.2. Trong hình đó, hai khách hàng, tức là Máy tính xách tay và Thiết bị di động, được đăng ký vào các chủ đề khác nhau. Một máy khách, tức là thiết bị IoT, đang xuất bản thông báo "Hello" cho topic "Message". Nhà môi giới sẽ có trách nhiệm gửi tin nhắn " Hello " đến thuê bao đã đăng ký topic "Message".

#### 2.3.1.b MQTT Broker

Do đó, nhà môi giới MQTT có thể được coi là máy chủ. Máy chủ này, quản lý tất cả các tin nhắn được xuất bản từ khách hàng. Nói cách khác, việc gửi tin nhắn đến những người đăng ký [6] là có trách nhiệm. Do đó, broker inMQTT là một yếu tố quan trọng để giao thức nhắn tin này hoạt động.

**Topic**

Một chủ đề là một chuỗi UTF-8 được nhà môi giới sử dụng để xử lý tin nhắn cho mỗi kết nối thiết bị. Chúng thường bao gồm các cấp độ khác nhau được phân tách bằng một dấu gạch chéo. Cấu trúc giống như hệ thống phân cấp này được thể hiện trong hình 2.3.

Diagram

Description automatically generated with medium confidence

*Hình 2. 3. Topic trong MQTT.*

Các chủ đề không phải là vĩnh viễn và được tạo bởi khách hàng hoặc người đăng ký. Một chủ đề chỉ tồn tại nếu một người đăng ký đã đăng ký vào nó hoặc người môi giới có một thông báo di chúc được giữ lại hoặc cuối cùng được lưu trữ cho chủ đề đó. Khi tạo một chủ đề mới, một khách hàng thực tế được tự do trong việc lựa chọn tên đó mặc dù có một số ngoại lệ như được giải thích trong phần sau.

**Special topics**

Có một chủ đề đặc biệt trong MQTT được gọi là chủ đề $SYS. Điều này được tạo bởidefault bởi nhà môi giới MQTT và nó là một chủ đề dành riêng được sử dụng bởi hầu hết các nhà môi giới MQTT để cung cấp thông tin về nhà môi giới. Các chủ đề chỉ dành cho người đọc này cho các máy khách MQTT [7]. Đây là lý do tại sao khi tạo dị ứng, nếu nó bắt đầu bằng biểu tượng $ thì nó sẽ được xử lý đặc biệt.

**Wildcards**

Như đã đề cập ở trên, một thiết bị có thể đăng ký một hoặc nhiều chủ đề. Có một chủ đề đặc biệt được gọi là ký tự đại diện được đại diện bởi #or bằng +. Biểu tượng đầu tiên có nghĩa là một ký tự đại diện đa cấp và ký tự đại diện cấp asingle sau. Điều này có thể dễ dàng hiểu được với ví dụ sau:

* /blastingsensor/iot1/#/
* /blastingsensor/+/movement

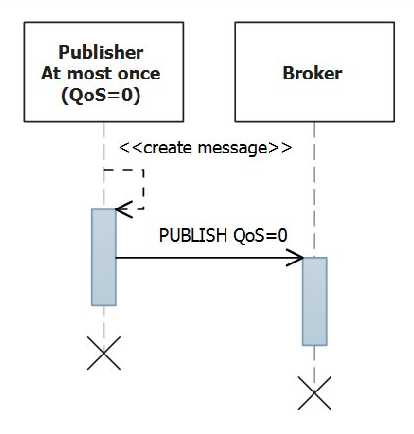
Trong trường hợp đầu tiên, tất cả các tính năng từ iot1 sẽ được cung cấp trong trường hợp thứ hai chỉ có chuyển động tính năng sẽ được đưa ra nhưng từ bất kỳ thiết bị đơn lẻ nào, tức là IoTx. Cần phải đề cập rằng ký tự đại diện chỉ có thể thực hiện được đối với các đăng ký chứ không phải trong xuất bản từ máy khách.

#### 2.3.1.c Quality of Service

Chất lượng dịch vụ là một khái niệm phổ biến trong mạng nói về việc gửi tin nhắn ngụ ý một thỏa thuận phải được thực hiện giữa người gửi và người nhận. MQTT hỗ trợ ba mức chất lượng dịch vụ khác nhau [6]. Đây là những điều sau đây:

* QoS 0: Nhiều nhất một lần giao hàng cũng được gọi là "fire and forget".
* QoS 1: Ít nhất một lần giao hàng còn được gọi là "acknowledged delivery".
* QoS 2: Chính xác một lần giao hàng còn được gọi là "assured delivery".

Sơ đồ dòng chảy của các khả năng khác nhau về chất lượng dịch vụ có thể được nhìn thấy hình 2.4, hình 2.5 và 2.6 tương ứng.



*Hình 2. 4. QoS 0: Almost Once*

Diagram

Description automatically generated

*Hình 2. 5. QoS 1 – At Least Once*

QoS 0 hoặc 'nhiều nhất một lần' có nghĩa là khách hàng xuất bản, ví dụ: thiết bị IoT, sẽ gửi tin nhắn đến nhà môi giới và nó sẽ xóa nó sau đó. Không có xác nhận nào sẽ được trả lại cho khách hàng có nghĩa là thisentity sẽ không biết liệu tin nhắn có đến nhà môi giới hay không. Mức độ dịch vụ chất lượng này cung cấp sự đảm bảo giống như giao thức TCP. Kiến trúc luồng của chất lượng dịch vụ này có thể được nhìn thấy trong 2.4.

QoS 1 hoặc 'ít nhất một lần' có nghĩa là khách hàng sau khi đã gửi hoặc xuất bản đến một chủ đề, nó sẽ lưu trữ tin nhắn vừa được xuất bản và sẽ tiếp tục nhận được xác nhận từ nhà môi giới MQTT. Nếu điều này không xảy ra sau một khoảng thời gian, khách hàng sẽ thử lại để xuất bản lạimessage. Khi nó nhận được ack, tức là xác nhận, nó sẽ xóa tin nhắn. Vì vậy, nói cách khác, có một sự đảm bảo rằng tin nhắn sẽ được gửi đến người nhận ít nhất một lần. Kiến trúc dòng chảy của chất lượng dịch vụ này có thể được nhìn thấy trong 2.5.

Diagram

Description automatically generated

*Hình 2. 6. QoS 2 – Exactly Once*

QoS 2 hoặc 'chính xác một lần' có nghĩa là khách hàng sẽ lưu trữ thông báo để xuất bản nó lên nhà môi giới. Điều này cũng sẽ lưu trữ nó cho đến khi một người đăng ký mà chủ đề đó kết nối. Sau khi đã xuất bản tin nhắn, nhà môi giới sẽ gửi cho khách hàng một Xuất bản tin nhắn nhận được, hoặc PUBREC. Theclient sau đó sẽ trả lời bằng một bản phát hành đã xuất bản, tức là PUBREL. Sau đó, broker sẽ xóa tin nhắn đã lưu trữ trước đó và gửi một Publishcomplete, PUBCOMP, cho khách hàng để nó biết rằng tin nhắn đã đến và có thể bị xóa. Đây là mức chất lượng dịch vụ chậm nhất mà MQTT cung cấp. Kiến trúc dòng chảy của chất lượng dịch vụ này có thể được nhìn thấy trong 2.6.A điều cuối cùng và rất quan trọng cần được chỉ ra, đó làQoS được lựa chọn nghiêm ngặt giữa khách hàng và nhà môi giới. Điều này có nghĩa là khách hàng xuất bản cho một nhà môi giới có thể đã chọn một QoS khác với khách hàng đăng ký chủ đề đó và nhà môi giới.

#### 2.3.1.d Session and Messages

Trong mục này, các đặc điểm cụ thể của định dạng messages và session sẽ được đề cập [9].

Định dạng của gói MQTT có thể được nhìn thấy trong hình.

Timeline

Description automatically generated

*Hình 2. 7. Định dạng packet của MQTT.*

Timeline

Description automatically generated

*Hình 2. 8. Bản tin CONNECT trong MQTT*

Khi một khách hàng kết nối với một nhà môi giới, nó cần phải đăng ký một hoặc một số chủ đề để nhận được tin nhắn. Một khía cạnh quan trọng cần được nhắc nhở là xuất bản máy khách và máy khách đăng ký không cần phải kết nối cùng một lúc. Tin nhắn có thể được lưu trữ hoặc nói tốt hơn, giữ lại, trong nhà môi giới. Điều này là do cái gọi là "Tin nhắn được giữ lại". Khi một khách hàng xuất bản một tin nhắn đến một chủ đề, các loại cờ và tùy chọn khác nhau cũng có thể được gửi cùng với như nó có thể được nhìn thấy trong hình 2.7. Một cái rất quan trọng được gọi là 'Giữ lại' là bit cuối cùng của byte đầu tiên của tin nhắn. Nếu bit đó được đặt thành '1', tin nhắn được xuất bản bởi khách hàng sẽ được giữ trong nhà môi giới cho đến khi một tin nhắn mới được gửi đi.

Điều này có nghĩa là bất cứ khi nào một thuê bao kết nối, nó sẽ nhận được thông báo cuối cùng được xuất bản cho chủ đề.

Graphical user interface, application, table, timeline

Description automatically generated

*Hình 2. 9. Bản tin PUBLISH trong MQTT*

#### 2.3.1.e Bảo mật trong MQTT

Trong phần này, các khía cạnh bảo mật MQTT sẽ được giải thích. Đây là những vấn đề bảo mật trong MQTT và các khả năng khác nhau có thể được thực hiện trên giao thức này.

**Vi phạm bảo mật và các cuộc tấn công chống lại MQTT**

Trong phần này, các vi phạm bảo mật và các cuộc tấn công khác nhau chống lại MQTT asit đã được thiết kế, tức là MQTT qua TCP, được giải thích [10].

***Subscriber cho tất cả các topic***

Kẻ tấn công có thể đăng ký chủ đề ký tự đại diện được giải thích trong phần trước, tức là #. Điều này sẽ cho phép kẻ tấn công được đăng ký alltopics và theo cách đó xem tất cả các chủ đề và dữ liệu. Nó phá vỡ chống lại tính bảo mật.

***Publish unauthorized data to the broker***

Nếu broker không có bất kỳ loại cơ chế xác thực nào, mọi thiết bị, bao gồm cả kẻ tấn công, đều có thể xuất bản dữ liệu cho nhà môi giới và do đó thực hiện tấn công từ chối dịch vụ cho nhà môi giới MQTT. Theaim là những loại từ chối dịch vụ này, tức là các cuộc tấn công DoS là để cạn kiệt tài nguyên của máy chủ, trong trường hợp này là nhà môi giới MQTT.

***Quyền riêng tư của dữ liệu***

Dữ liệu MQTT không được mã hóa nên nó thực sự là một vấn đề trong nhiều tình huống khác nhau. Không có vấn đề gì nếu nhà môi giới sử dụng cơ chế xác thực hay không vì dữ liệu trong quá trình chuyển tiếp có thể dễ dàng bị đánh hơi bởi kẻ tấn công. Vấn đề này một lần nữa là một vấn đề bảo mật có nghĩa là dữ liệu có thể được nhìn thấy bởi bất kỳ ai có kỹ năng kết nối mạng cơ bản.

***Xác thực***

Như đã nêu trong phần trước, MQTT cung cấp một số tính năng bảo mật trong lớp ứng dụng. Ví dụ: nếu nhà môi giới sử dụng cơ chế xác thực máy khách bằng cách chỉ cho phép người dùng bằng tên người dùng và mật khẩu, kẻ tấn công sẽ bị chặn trong bất kỳ cơ hội nào xuất bản dữ liệu trái phép. Điều này là bằng cách giả sử rằng kẻ tấn công không biết tên người dùng cũng như mật khẩu để kết nối với nhà môi giới. Tuy nhiên, tên người dùng và mật khẩu được gửi dưới dạng văn bản thuần túy trong MQTT trong lệnh CONNECT. Vấn đề tương tự phát sinh ở đây, kẻ tấn công có thể đánh hơi dữ liệu và lấy những thông tin đăng nhập đó và sử dụng chúng để chống lại the broker để xuất bản dữ liệu hoặc nhận nó.

***Tính toàn vẹn dữ liệu***

Bây giờ rõ ràng là kẻ tấn công có thể đánh hơi dữ liệu mà không gặp quá nhiều rắc rối. Không thực sự quan trọng rằng tên người dùng và mật khẩu được sử dụngnhư một cơ chế ủy quyền để kết nối với nhà môi giới MQTT. Một vấn đề lớn hơn xuất hiện ở đây liên quan đến tính toàn vẹn dữ liệu. Một kẻ tấn công, havingeavesdropd cuộc trò chuyện có thể nắm bắt các gói và sửa đổithem để sau đó gửi chúng đến nhà môi giới MQTT. Điều này có thể gây ra sự cố nghiêm trọng cho các thiết bị IoT khác nhau đã đăng ký các chủ đề đónơi thông báo đang bị thay đổi. Hành vi của các thiết bị IoT đó có thể được thay đổi do những thông báo sửa đổi đó. Đã nêu các loại tấn công và vi phạm khác nhau liên quan đến bảo mật trong MQTT, một số khả năng bảo mật cần thực hiện sẽ được giải thích ở đây.

**Khả năng bảo mật trong MQTT**

Như đã ngụ ý, bản thân giao thức MQTT không được thiết kế với mục đích bảo mật. MQTT là một giao thức truyền thông thực sự nhẹ và rất đơn giản và nó chỉ có một vài cơ chế bảo mật được chỉ định trong đó. Bên cạnh đó và như đã đề cập trước đây, MQTT dựa vào TCP làm giao thức truyền tải của nó để không có giao tiếp được mã hóa nào được sử dụng.

Trong mỗi lớp của ngăn xếp giao thức TCP / IP, khả năng chống lại các loại tấn công khác nhau có thể được cung cấp.

***Network Level***

Ở cấp độ mạng, một giải pháp an toàn sẽ là sử dụng mã hóa tải trọng để làm cho thông tin đã xuất bản được mã hóa ngay cả khi không sử dụng mã hóa truyền tải.

***Transport Level***

Giao thức TLS / SSL được biết là được sử dụng ngày nay trong nhiều loại mạng khác nhau và trên nhiều giao thức khác nhau. Các giao thức TLS nhằm mục đích cung cấp không chỉ quyền riêng tư trong giao tiếp mà còn toàn vẹn dữ liệu giữa các bên giao tiếp. Trong cổng MQTT, 8883 được chuẩn hóa cho MQTT qua TLS / SSL. Việc triển khai này được khuyến khích mạnh mẽ bởi tiêu chuẩn OASIS [11]. Cổng 1883 được đăng ký trong IANA cho kết nối không phải TLS, tức là TCP không được mã hóa.

***Application Level***

Ở cấp độ này, các thuộc tính bảo mật MQTT là tùy chọn để triển khai. Trong gói kết nối MQTT được hiển thị trong 2.8, có hai trường tùy chọn có thể được sử dụng để xác thực máy khách MQTT cho trình chuyển tiếp. Các trường đó có thể được nhìn thấy trong hình 2.8 là một phần của tải trọng tùy chọn trong gói MQTT CONNECT. Các trường này là:

* Mã định danh máy khách
* Thông tin đăng nhập tên người dùng và mật khẩu

Ngoài tất cả các giải thích ở trên, một số triển khai bảo mật bổ sung có thể được áp dụng trên nhà môi giới MQTT tùy thuộc vào công nghệ được sử dụng để thực hiện điều đó. Một trong những triển khai này là liên quan đến kiểm soát truy cập.

Kiểm soát truy cập này đối với nhà môi giới dựa trên một số danh sách được gọi là Danh sách kiểm soát truy cập, tức là ACL, về cơ bản cho phép kết nối với nhà môi giới từ các máy khách cụ thể hoặc thậm chí có thể chỉ định các hạn chế đối với một số chủ đề.

#### 2.3.1.f Mã hóa

Do đó, mật mã là một khía cạnh chính để xem xét cải thiện tính bảo mật trong các ứng dụng triển khai MQTT làm giao thức. Trong phần này, các khía cạnh khác nhau của mật mã được giải thích. Mật mã học hiện đại có thể được chia thành hai lĩnh vực lớn: Mật mã đối xứng và Mật mã khóa không đối xứng.

**Asymmetric key**

Năm 1976, Whitfield Diffie và Martin Hellman đã đề xuất khái niệm mật mã khóa công khai này, nơi hai khóa khác nhau nhưng có liên quan đến toán học được sử dụng. Trong các hệ thống mật mã này, người gửi và người nhận sử dụng các khóa khác nhau. Mỗi người trong số họ có một cặp khóa cụ thể là khóa riêng tư và khóa công khai. Khóa công khai đúng như tên gọi của nó là công khai, vì vậy nó được phân phối tự do trong khi khóa riêng tư phải được giữ như một bí mật. Mật mã khóa công khai là một hoạt động tốn kém và đó là lý do tại sao nó được biết là chậm hơn mật mã khóa đối xứng. Khái niệm được thêm vào trong mật mã khóa công khai này là mỗi thực thể có thể nói là có danh tính riêng và hai khóa khác nhau.

**RSA**

RSA, tức là, Rivest-Shamir-Adleman là một thuật toán mật mã, một trong những hệ thống mật mã khóa bất đối xứng đầu tiên. Trong thuật toán này, hai số mồi lớn là cần thiết để được tính toán để sau này lưu tính toán đó dưới dạng khóa bí mật. RSA được nhúng trong giao thức TLS / SSL được sử dụng để thiết lập giao tiếp an toàn.

Người gửi có thể mã hóa tin nhắn bằng khóa công khai của người nhận và người nhận sau đó sẽ giải mã tin nhắn đó bằng khóa riêng.

**ECC**

Mật mã đường cong Elliptic, tức là ECC, là một thuật toán bất đối xứng mới hơn sử dụng các đường cong elip trên các trường hữu hạn. Đây là một trong những giải pháp được cho là có thể thay thế RSA.

|  |  |
| --- | --- |
| **Kích thước khóa RSA (bit)** | **Kích thước khóa ECC (bit)** |
| 1024 | 160 |
| 2048 | 224 |
| 3072 | 256 |
| 7680 | 384 |
| 15360 | 512 |

*Comparison between ECC and RSA key sizes.*

Nghiên cứu trước đây đã nghiên cứu hành vi và đã so sánh giữa RSA và ECC [2] cho thấy mật mã đường cong Elliptic nhanh hơn nhiều so với RSA và do đó nó đòi hỏi ít sức mạnh tính toán hơn. Điều này chủ yếu là do kích thước khóa nhỏ hơn cần thiết trong ECC để đạt được mức độ bảo mật tương tự so với RSA. Điều này có thể được nhìn thấy trong bảng 2.10.

**Symmetric key**

Loại mã hóa này là loại mã hóa duy nhất được biết đến cho đến tháng 1976 năm XNUMX khi một loại mã hóa mới được phát hành vì nó sẽ được giải thích trong phần sau. Các hệ thống mật mã sử dụng khóa đối xứng đề cập đến các phương pháp mã hóa trong đó cùng một khóa được chia sẻ giữa cả người gửi và người nhận. Đó là lý do tại sao nó còn được gọi là hệ thống mật mã khóa bí mật. Các khóa này thường được phân phối giữa các bên bằng cách sử dụng một kênh an toàn. Mật mã khóa đối xứng được triển khai dưới dạng mật mã khối hoặc mật mã luồng. Có một số ví dụ về thuật toán mật mã khóa đối xứng như Tiêu chuẩn mã hóa dữ liệu, tức là DES, Mã hóa nâng cao Tiêu chuẩn, tức là, AES hoặc Blowfish. Cần phải đề cập rằng các hệ thống mật mã đối xứng được biết là nhanh hơn về thời gian tính toán so với các hệ thống bất đối xứng sẽ được giải thích trong phần sau.

**Stream Ciphers**

Mật mã luồng là mật mã khóa đối xứng trong đó luồng chữ số mật mã giả được kết hợp với các chữ số văn bản thuần túy. Chúng hoạt động với sự chuyển đổi thay đổi theo thời gian trên các chữ số văn bản thuần túy riêng lẻ. Đây là các chữ số văn bản thuần túy được mã hóa từng chữ một với luồng khóa để sau đó đưa ra một chữ số của luồng bản mã. Mã hóa sau đó phụ thuộc vào trạng thái hiện tại của mật mã, vì vậy mật mã luồng này còn được gọi là mật mã trạng thái.

**Block Ciphers**

Mật mã khối được sử dụng để mã hóa hoặc giải mã dữ liệu được vận hành trên các đơn vị bit được gọi là khối. Ví dụ về mật mã khối là DES, AES hoặc Blowfish. AES sẽ được giải thích sau vì nó là một mật mã khối được xem xét theo công việc này. Mật mã khối này có một số chế độ hoạt động vì mọi thứ phụ thuộc vào cách đầu vào và đầu ra của khối được xâu chuỗi với các chế độ sau.

Tiêu chuẩn mã hóa tiên tiến quảng cáo, tức là AES hoặc Blowfish. Cần phải đề cập rằng các hệ thống mật mã đối xứng được biết là nhanh hơn về thời gian tính toán so với các hệ thống bất đối xứng sẽ được giải thích trong phần sau.

**Chế độ mã hóa của AES**

Mật mã khối này, tức là AES, có một số chế độ hoạt động khác nhau như đã được giải thích ở trên mà mật mã khối làm. Trong phần này của báo cáo, hai trong số các chế độ mã hóa của AES sẽ được giải thích, những chế độ được sử dụng trong công việc này.

Tiêu chuẩn mã hóa nâng cao, tức là AES là một mật mã khối được phát triển bởi Joan Daemen và Vincent Rijmen. AES có thể hoạt động với các kích thước khóa khác nhau, tức là 128 bit, 256 bit và 512 bit, và nó có từ 10 đến 14 vòng mật mã khối đơn giản. Mật mã khối này thành công do tính đơn giản trong thiết kế.

**AES-CTR**

Chế độ truy cập là một thuật toán mã hóa đối xứng phổ biến. Nó biến một mật mã khối thành một mật mã luồng. Keystream tiếp theo được tạo ra bằng cách mã hóa các giá trị liên tiếp của một 'bộ đếm'. Chế độ này cũng được giới thiệu cùng lúc với mã hóa khóa công khai, năm 1979 bởi Whitfield Diffie và Martin Hellman. Chế độ mã hóa này được hiển thị trong hình

**AES-GCM**

Galois Counter Mode, tức là GCM, là một chế độ hoạt động cho các mật mã khối mật mã khóa đối xứng [13]. GCM là một phương thức hoạt động của thuật toán AES [14]. Chế độ này thuận tiện để bảo vệ các gói dữ liệu vì nó có độ trễ thấp và chi phí hoạt động tối thiểu. Nó được biết đến và áp dụng rộng rãi do hiệu quả và hiệu suất của nó. Nó cung cấp sự đảm bảo về tính bảo mật của dữ liệu bằng cách sử dụng một biến thể của chế độ AES-CTR để mã hóa. Đây là một chế độ mã hóa xác thực đạt được điều này do chức năng băm phổ quát được cung cấp. Nói cách khác, nó cung cấp cả tính xác thực của dữ liệu, tức là tính toàn vẹn và tính bảo mật.

**Security in AES-GCM**

Chế độ mã hóa AES-GCM, đã được chứng minh là an toàn khi nó được sử dụng với mật mã khối rằng nó không chỉ là một hoán vị ngẫu nhiên. Chế độ Galois Counter (GCM) được sử dụng trong các bộ mật mã khác nhau trong TLS [13] làm thuật toán mã hóa và đó là lý do chính tại sao em quyết định đánh giá nó với MQTT trong luận án này. Khía cạnh quan trọng nhất để đảm bảo an ninh với AES-GCM là chọn một vectơ khởi tạo duy nhất cho mọi mã hóa được thực hiện với cùng một khóa. GCM là một chế độ bảo mật hữu ích để giao tiếp một cách an toàn với các thiết bị nhúng nhỏ [15]. Như đã đề cập trước đây, vectơ khởi tạo, tức là IV, là một trong những khía cạnh quan trọng nhất cần quan tâm trong chế độ mã hóa này. AESGCM được sử dụng để mã hóa được xác thực với dữ liệu liên quan. GCM cung cấp tính bảo mật và tính xác thực cho dữ liệu được mã hóa và tính xác thực cho dữ liệu được xác thực bổ sung (AAD). AAD không được mã hóa. Chế độ GCM yêu cầu IV là một nonce, tức là, IV phải là duy nhất cho mỗi lần thực hiện nếu cùng một khóa được sử dụng [16]. Các bước để mã hóa GCM có thể được nhìn thấy trong hình 2.11. Đó là những điều sau đây:

* Khóa con băm cho hàm GHASH được tạo bằng cách áp dụng mật mã khối cho khối "không".
* Khối tiền truy cập được gọi là J0 được tạo ra từ IV. Đặc biệt, khi độ dài của IV là 96 bit, thì chuỗi đệm 031|| 1 được nối vào IV để tạo thành khối tiền truy cập. Mặt khác, IV được đệm với số lượng bit '0' tối thiểu, có thể không có, do đó độ dài của chuỗi kết quả là bội số của 128 bit (kích thước khối); chuỗi này lần lượt được nối với 64 bit '0' bổ sung, tiếp theo là biểu diễn 64 bit về độ dài của IV và hàm GHASH được áp dụng cho chuỗi kết quả để tạo thành khối tiền truy cập.
* Hàm gia số 32 bit được áp dụng cho khối bộ đếm trước để tạo khối bộ đếm ban đầu cho việc gọi hàm GCTR trên bản rõ. Đầu ra của lệnh gọi này của hàm GCTR là bản mã.
* AAD và bản mã đều được nối với số lượng bit null tối thiểu, có thể không có, do đó độ dài bit của các chuỗi kết quả là bội số của kích thước khối. Sự ghép nối của các chuỗi này được nối với các biểu diễn 64 bit về độ dài của AAD và bản mã để tạo ra khối u.
* Hàm GHASH được áp dụng để chặn bạn để tạo ra một khối đầu ra duy nhất.
* Khối đầu ra này được mã hóa bằng hàm GCTR với khối bộ đếm trước được tạo ở Bước 2 và kết quả được cắt bớt đến độ dài thẻ được chỉ định để tạo thành thẻ xác thực.

Diagram

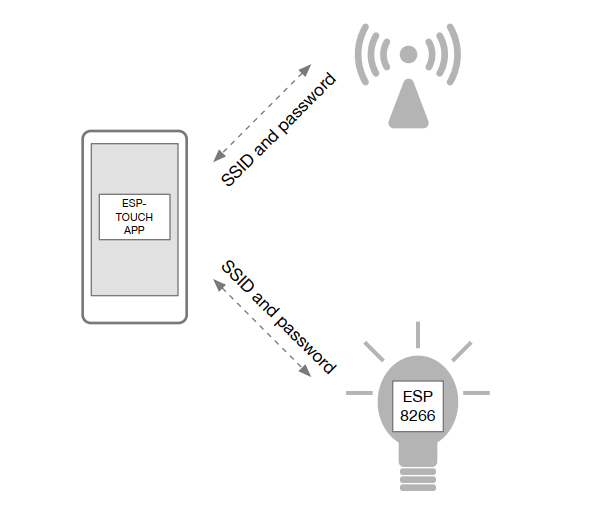
Description automatically generated

*Hình 2. 10. Chế độ mã hóa AES-GCM*

### 2.3.2 Giao thức ESP-Touch

#### 2.3.2.a Giới thiệu

ESP-TOUCH Smart Config quá trình được bắt đầu sau khi chức năng Cấu hình thông minh được bật trong thiết bị hỗ trợ ESP-TOUCH. Khi điều kiện này được đáp ứng, cấu hình sử dụng thiết bị hỗ trợ Wi-Fi có thể bắt đầu [3]. Thiết bị hỗ trợ Wi-Fi như máy tính bảng hoặc điện thoại thông minh kết nối và gửi các gói UDP đến Wi-Fi AP thông qua ứng dụng di động chuyên dụng đặc biệt. Ngoài ra, nó sẽ mã hóa SSID và mật khẩu vào trường độ dài của một chuỗi các gói UDP. Từ đó thiết bị ESP tiếp cận và giải mã thông tin để kết nối với AP. Sau khi hoàn tất quá trình cấu hình, thiết bị sẽ trả về IP của phía máy phát và phía máy phát sẽ nhận được IP của thiết bị. Khái niệm cơ bản của quy trình được mô tả trong Hình 3 [3]



*Hình 2. 11. Mô hình giao thức ESP-Touch*

Nếu thiết bị không thể kết nối với bộ định tuyến trong một thời gian xác định, ứng dụng sẽ trả về thông báo lỗi. Thông báo lỗi và vòng tiếp theo của quy trình SmartConfig sẽ bắt đầu nếu thiết bị không thể lấy SSID và PASSWORD trong một thời gian nhất định.

#### 2.3.2.b Lợi thế

Ưu điểm của ESP-TOUCH được liệt kê như sau:

* thân thiện với người dùng.
* cấu hình tương đối nhanh.
* Quá trình cấu hình có thể được thực hiện xa thiết bị.
* cho phép cấu hình nhiều thiết bị cùng một lúc.

#### 2.3.2.c Hạn chế

Giới hạn của ESP-TOUCH được liệt kê như sau:

* Phải sử dụng thiết bị ESP để thiết lập hệ thống.
* Quá trình cấu hình sử dụng Wi-Fi để truyền dữ liệu.
* Các thiết bị mới trong hệ thống yêu cầu quá trình cấu hình bổ sung.

### 2.3.3 Giao thức One Wire

#### 2.3.3.a Giới thiệu về giao thức

Giao thức 1-Wire sử dụng các mức logic CMOS / TTL thông thường (tối đa 0,8V cho logic "0" và tối thiểu 2.2V cho logic "1") với hoạt động được chỉ định trên dải điện áp cung cấp từ 2.8V đến 6V. Cả master và slave đều được cấu hình làm bộ thu phát cho phép dữ liệu tuần tự bit chảy theo một trong hai hướng, nhưng chỉ có một hướng tại một thời điểm (halfduplex); Master khởi tạo và điều khiển tất cả các thiết bị

Dữ liệu được tuần tự byte và tuần tự bit với dữ liệu đọc và ghi bit ít quan trọng nhất (LSB) đầu tiên tín hiệu được truyền trong các khe thời gian đồng hồ hệ thống là không cần thiết; mỗi phần 1 dây được tự đồng hồ bởi một bộ dao động bên trong được đồng bộ hóa với cạnh rơi của bộ chính

#### 2.3.3.b Hoạt động

|  |  |  |
| --- | --- | --- |
| **Hoạt động** | **Mô tả** | **Thực thi** |
| Ghi bit 1 | Gửi một bit 1 tới slave | Bus điều khiển mức thấp, delay 6us  Giải phóng bus, delay 64us |
| Ghi bit 0 | Gửi môt bit 1 tới slave | Bus điều khiển mức thấp, delay 60us  Giải phóng bus, delay 10us |
| Đọc bit | Đọc 1-bit từ slave | Kéo mức thấp trong 6us  Kéo mức cao trong 9us  Lấy mẫu để đọc bit từ slave, trong 55us |
| Reset | Reset bus và đợi sẵn sàng gửi lệnh | Kéo mức thấp trong 480us  Giải phóng bus trong 70us  Lấy mẫu, delay 410us |

*Bảng 2. 1. Hoạt động của giao thức 1-wire*

#### 2.3.3.c Tính năng/Lợi ích

* Mỗi 1-Wire slave đã lưu trữ trong ROM một số sê-ri 64 bit duy nhất hoạt động như thiết bị địa chỉ nút của nó để được chọn riêng trong số nhiều thiết bị có thể được kết nối với cùng một dây bus.
* Địa chỉ duy nhất trên toàn cầu này bao gồm tám byte được chia thành ba phần chính. Bắt đầu với LSB, byte đầu tiên lưu trữ các mã 8 bit xác định loại thiết bị. Sáu byte tiếp theo lưu trữ một địa chỉ riêng lẻ 48 bit có thể tùy chỉnh. Byte cuối cùng, byte quan trọng nhất (MSB), chứa kiểm tra dự phòng theo chu kỳ (CRC) với giá trị dựa trên dữ liệu chứa trong bảy byte đầu tiên. Điều này cho phép chủ xác định xem một địa chỉ đã được đọc mà không có lỗi. Với nhóm số sê-ri 248, các địa chỉ nút xung đột hoặc trùng lặp trên mạng không bao giờ là vấn đề.
* Các thiết bị 1-Wire có thể được định dạng với một thư mục tệp như các tệp đĩa mềm có thể được truy cập và thay đổi ngẫu nhiên mà không làm phiền các bản ghi khác
* Bảo mật dữ liệu tối đa có thể được cung cấp bằng cách triển khai chip 1 dây của Thuật toán băm an toàn (SHA-1) được chính phủ Hoa Kỳ chứng nhận
* sử dụng một dây duy nhất (cộng với mặt đất) để thực hiện cả giao tiếp và truyền tải điện.
* Một bus master duy nhất có thể cung cấp nhiều nô lệ trên một cáp xoắn đôi đơn giản hóa thiết kế với một giao thức giao diện cung cấp điều khiển, báo hiệu và cấp nguồn qua kết nối một dây.

Diagram, engineering drawing

Description automatically generated

*Hình 2. 12. Dạng tín hiệu ghi bit 0/1, đọc bit, reset*

Table

Description automatically generated

*Hình 2. 13. Luồng giao tiếp của 1-Wire*

Phần đầu tiên của bất kỳ giao tiếp nào liên quan đến việc chủ xe buýt đưa ra "thiết lập lại" để đồng bộ hóa toàn bộ xe buýt. Một thiết bị nô lệ sau đó được chọn cho các liên lạc tiếp theo. Điều này có thể được thực hiện bằng cách chọn tất cả nô lệ, chọn một nô lệ cụ thể (sử dụng số sê-ri của thiết bị) hoặc bằng cách phát hiện ra nô lệ tiếp theo trên xe buýt bằng thuật toán tìm kiếm nhị phân. Các lệnh này được gọi chung là lệnh "network" hoặc ROM (ReadOnly-Memory). Khi một thiết bị cụ thể đã được chọn, tất cả các thiết bị khác sẽ bỏ qua và bỏ qua các liên lạc tiếp theo cho đến khi thiết lập lại tiếp theo được đưa ra. Khi một thiết bị được cách ly để liên lạc với xe buýt, chủ có thể đưa ra các lệnh dành riêng cho thiết bị cho thiết bị đó, gửi dữ liệu đến thiết bị hoặc đọc dữ liệu từ thiết bị đó. Bởi vì mỗi loại thiết bị thực hiện các chức năng khác nhau và phục vụ một mục đích khác nhau, mỗi loại có một giao thức duy nhất khi nó đã được chọn. Mặc dù mỗi loại thiết bị có thể có các giao thức và tính năng khác nhau, nhưng tất cả chúng đều có cùng một quy trình lựa chọn và tuân theo luồng lệnh như trong Hình 1 (phía trên). Cách cung cấp điện trong các trường hợp phổ biến nhất (w / sơ đồ). Ba cách chính:

* Tìm nguồn cung ứng điện bất cứ khi nào đường dây trên 3.5V.
* Vì các thiết bị 1 dây có thể hoạt động với nguồn cung cấp 3V, năng lượng có sẵn giữa các mức cung cấp xe buýt là 3.5V và 5V có thể được khai thác.
* Tương đương với việc vận hành tải ở chế độ shunt (vĩnh viễn trên xe buýt, hoặc tốt nhất là dưới sự kiểm soát của bus master)

Diagram, schematic

Description automatically generated

*Hình 2. 14. Cấp nguồn trong 1-wire*

* Tìm nguồn cung ứng điện khi đường dây cao bằng cách chuyển điện tích đến tụ điện (hoặc pin sạc) thông qua một diode chặn
* Trong thời gian nhàn rỗi khi xe buýt ở 5V, mạch 'đánh cắp' nguồn điện từ đường dây để sạc tụ điện và cấp nguồn cho tải.
* Disadv: thêm cả tải điện dung và rò rỉ làm giảm phạm vi và khả năng của mạng 1 dây. Có thể sửa chữa bằng cách có thể cô lập nó giữa hai công tắc địa chỉ được điều khiển bởi chủ đường dây.
* Adv: đơn giản và tiết kiệm; được sử dụng nội bộ bởi các thiết bị 1-Wire

Diagram, schematic

Description automatically generated

*Hình 2. 15. Cấp nguồn vi sai cho 1-wire*

* Tìm nguồn cung ứng năng lượng với lực kéo mạnh mẽ trong thời gian giao tiếp nhàn rỗi.
* Có thể được thực hiện bằng cách sử dụng MOSFET để kéo xe buýt trực tiếp vào đường ray
* Ví dụ: DS18S20-PAR (nhiệt kế kỹ thuật số, trong quá trình chuyển đổi nhiệt độ)

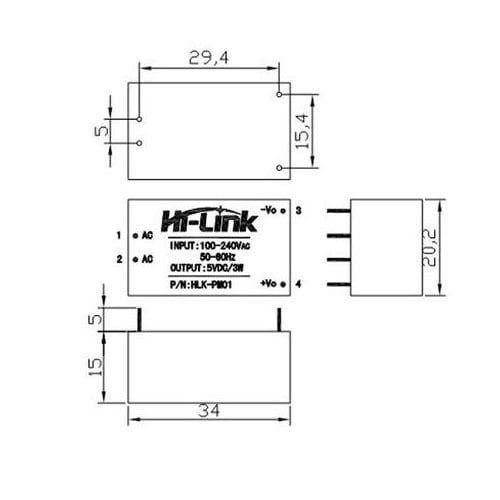
Diagram, schematic

Description automatically generated

*Hình 2. 16. Kết nối cảm biến với MCU qua giao thức 1-Wire*

## 2.4 Một số linh kiện sử dụng

### 2.4.1 Nguồn HLK-PM01



Hình 2. 17. Kích thước của module HLK-PM01

Diagram, schematic

Description automatically generated

Hình 2. 18. Sơ đồ nguyên lý của module nguồn HLK-PM01

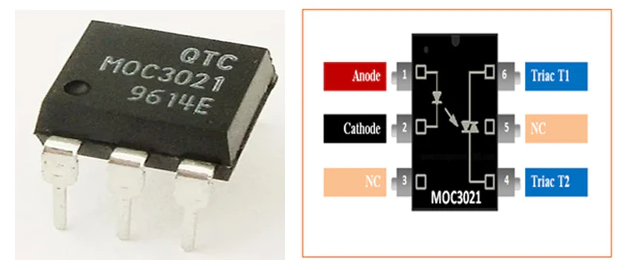
Module nguồn AC-DC Hi-Link HLK-PM01 5VDC 3W có thiết kế nhỏ gọn với vỏ bọc nhựa an toàn, chuyên nghiệp, được sử dụng để chuyển nguồn xoay chiều AC sang 5VDC công suất tối đa 3W cấp cho thiết bị, module được sản xuất bởi hãng Hi-Link chuyên về các module nguồn được sử dụng trong công nghiệp với độ bền, chống nhiễu tốt và độ an toàn cao.

Thông số kỹ thuật:

* Điện áp đầu vào: 100V ~ 240VAC / 50~60Hz
* Điện áp đầu ra: 5VDC
* Công suất trung bình: 3W
* Đáp ứng yêu cầu UL, CE
* Kích thước 20.2 mm x 34mm x 15 mm
* Bảo vệ quá tải và ngắn mạch
* Hiêu năng cao
* Sản phẩm được thiết kế đáp ứng các yêu cầu về tiêu chuẩn EMC và Safe Test
* Tiêu thụ năng lượng thấp, bảo vệ môi trường, tổn thất công suất không tải < 0.1W

### 2.4.2 Opto quang MOC3021

MOC3021 là optocoupler hay optoisolator điều khiển TRIAC phát hiện điểm 0. Như đã biết thuật ngữ optocoupler hay optoisolater có nghĩa là sử dụng ánh sáng để ghép nối gián tiếp vào các bộ mạch. Điểm đặc biệt của MOC3021 là nó có khả năng phát hiện điểm zero và được điều khiển bởi một Triac.



Hình 2. 19. Bộ ghép quang MOC3021

Vì đầu ra được điều khiển bởi TRIAC, có thể điều khiển tải lên đến 400V và triac có thể dẫn theo cả hai hướng do đó việc kiểm soát tải AC sẽ không thành vấn đề. Cũng vì nó có khả năng phát hiện điểm 0 nên khi bật tải AC, triac sẽ bắt đầu dẫn điện sau khi sóng AC đạt đến 0V, theo cách này chúng ta có thể tránh điện áp đỉnh trực tiếp đến tải và do đó ngăn nó bị hỏng. Nó cũng có thời gian tăng và giảm khá và do đó có thể được sử dụng để kiểm soát điện áp đầu ra.

Các tính năng này của MOC3021 làm cho nó trở thành một lựa chọn lý tưởng để điều khiển tải AC điện áp cao thông qua các bộ điều khiển kỹ thuật số như MPU/ MCU. Vì đầu ra được kiểm soát, chúng ta có thể kiểm soát cường độ của ánh sáng hoặc tốc độ của động cơ AC. Vì vậy, nếu bạn đang tìm kiếm một bộ cách ly quang để điều khiển một ứng dụng AC thông qua DC thì IC này là lựa chọn phù hợp cho bạn.

* Optoisolator với điều khiển triac phát hiện điểm không
* Điện áp thuận Diode LED đầu vào: 1.15V
* Dòng chốt thuận LED: 15mA
* Điện áp chân đầu ra TRIAC: 400V (tối đa)
* Dòng đầu ra đỉnh TRIAC: 1A
* Có dạng PDIP 6 chân có và không có hậu tố M

### 2.4.3 Triac BT136E

BT136 là TRIAC với dòng điện đầu cuối tối đa 4A. Điện áp ngưỡng cổng của BT136 cũng rất ít nên có thể được điều khiển bởi các mạch kỹ thuật số.

Vì TRIAC là thiết bị chuyển mạch hai chiều, chúng thường được sử dụng để chuyển đổi các ứng dụng AC. Vì vậy, nếu chuyển đổi điều khiển (mờ, điều khiển tốc độ) một tải AC tiêu thụ ít hơn 6A với một thiết bị kỹ thuật số như vi điều khiển hoặc bộ vi xử lý thì BT136 có thể phù hợp với bạn.

**Cấu hình chân**

|  |  |  |
| --- | --- | --- |
| **Pin number** | **Pin name** | **Description** |
| 1 | Main Terminal 1 | Connected to Phase or neutral of AC mains |
| 2 | Main Terminal 2 | Connected to Phase or neutral of AC mains |
| 3 | Gate | Used to trigger the SCR |

*Bảng 3. 1. Các chân của triac BT136*

**Tính năng**

* Thiết bị đầu cuối tối đa hiện tại: 4A
* Điện áp cổng on-state: 1.4V
* Cổng kích hoạt hiện tại: 10mA
* Điện áp đầu cuối tối đa là 600 V
* Giữ hiện tại: 2.2mA
* Chốt hiện tại: 4mA
* Có sẵn trong gói To-220

**Cách sử dụng BT136**

Diagram

Description automatically generated

*Hình 3. 1. Mạch điều khiển triac*

Có nhiều cách để sử dụng TRIAC, vì thiết bị là hai chiều, cổng TRIAC có thể được kích hoạt với điện áp dương hoặc điện áp âm. Vì vậy, điều này cho phép TIRAC được vận hành ở bốn chế độ khác nhau. Một mạch chuyển mạch TRIAC đơn giản được hiển thị bên dưới.

Trong mạch như hình 3.2, TRIAC có thể được xoay bằng công tắc, khi nhấn công tắc, TRIAC sẽ đóng kết nối cho bóng đèn AC mặc dù nguồn điện AC. Để điều này xảy ra, chân cổng của TRIAC sẽ nhận được điện áp lớn hơn điện áp cổng ngưỡng và cũng sẽ nhận được dòng điện lớn hơn dòng kích hoạt cổng. Điều này sẽ làm cho TRIAC bật.

Vì TRIAC và SCR có hầu hết các đặc điểm giống nhau, giống như SCR, TRIAC cũng sẽ không tắt khi điện áp cổng được loại bỏ. Chúng ta cần loại mạch đặc biệt được gọi là mạch giao hoán để quay lại SCR một lần nữa. Việc chuyển mạch này thường được thực hiện bằng cách giảm dòng tải (giao hoán cưỡng bức) ít hơn dòng điện giữ. Nói một cách đơn giản, TRIAC sẽ chỉ bật cho đến khi dòng tải lớn hơn dòng giữ của TRIAC.

Ngoài việc điều khiển thông qua chuyển đổi, BT136 cũng có thể được điều khiển thông qua bộ vi điều khiển hoặc bộ vi xử lý. Để làm điều này, chúng ta cần một bộ cách ly Opto như [MOC3021](https://components101.com/ics/moc3021-triac-driven-optoisolator-pinout-equivalent-datasheet) để cách ly mạch AC dạng điện tử kỹ thuật số. Bằng cách này, tải không chỉ có thể được chuyển đổi mà còn có thể điều khiển điện áp đầu ra bằng cách sử dụng tín hiệu PWM để chuyển đổi nhanh chóng.

## Kết luận chương

Chương 2 đã trình bày chi tiết về công nghệ kết nối wi-fi để truyền thông tin dữ liệu giữa các thiết bị. Các giao thức truyền thông như MQTT quy định định dạng đóng gói dữ liệu , ESP-Touch bản chất là giao thức UDP để broadcast các bản tin UDP để cấu hình wi-fi tự động cho node IoT. Giao thức 1-wire để giao tiếp giữa cảm biến DHT22 và vi điều khiển ESP32 của node.

# CHƯƠNG 3. THIẾT KẾ THIẾT BỊ

Dựa trên các kiến thức được nghiên cứu, tổng hợp ở các chương trước, chương 3 sẽ trình bày chi tiết về các thiết kế node IoT dựa trên các nghiên cứu đó. Sau đó là phần thử nghiệm, đánh giá sản phẩm của đề tài.

## 3.1 Thiết kế phần cứng

### 3.1.1 Node đèn chiếu sáng

#### 3.1.1.a Giới thiệu node đèn chiếu sáng

Sơ đồ nguyên lý

Diagram

Description automatically generated

Hình 3. 2. Sơ đồ nguyên lý của node đèn chiếu sáng

Nguyên lý hoạt động của mạch:

1. Khối nguồn sử dụng module HLK-PM01 biến đổi điện áp đầu vào xoay chiều 220V thành điện áp đầu ra một chiều 5V. Điện áp đầu ra cung cấp cho vi điều khiển ESP32.
2. Vi điều khiển ESP32 là bộ xử lý tín hiệu điều khiển. Trong thiết kế node đèn 220V, ESP32 sử dụng chân IO D4 để xuất tín hiệu mức cao (3.3V) hoặc mức thấp (0V) để điều khiển bật hoặc tắt đèn 220V. Trong thiết kế node cảm biến nhiệt độ độ ẩm, ESP32 sử dụng chân IO D4 để giao tiếp với cảm biến DHT22 qua giao thức 1-wire.
3. IC so quang MOC3021 được dùng để cách ly dòng điện xoay chiều và 1 chiều nhằm mục đích bảo vệ khối xử lý ESP32 và các linh kiện sử dụng điện 1 chiều. Khi ESP32 xuất mức tín hiệu cao 3.3V thì MOC3021 dẫn điện, xuất mức tín hiệu thấp thì MOC3021 ngắt.
4. Triac BT136-E dùng để đóng ngắt mạch, tác dụng như một công tắc xoay chiều. Khi MOC3021 thông thì có dòng điện đi vào cực G của triac, khi này cực MT1và MT2 thông mạch, vậy có dòng điện chạy qua đầu ra đến bóng đèn, bóng đèn khi này sáng. Ngược lại, khi MOC3021 ngắt, triac ngắt, không có dòng điện đến bóng đèn, đèn tắt.
5. Điện trở R2 để hạn dòng đến cực G của triac, mục đích chống sốc điện đầu vào cho cực G, vì dòng điện lớn nhất đến cực G của triac là 30mA [3]. Đèn LED1 đồng bộ trạng thái bật tắt với đèn 220V, mục đích để kiểm tra trạng thái đèn 220V.

### 3.1.2 Node cảm biến

#### 3.1.2.a Sơ đồ nguyên lý

Diagram

Description automatically generated

*Hình 3. 3. Sơ đồ nguyên lý node cảm biến*

Node cảm biến thực hiện chức năng thu thập dữ liệu nhiệt độ, độ ẩm môi trường trong nhà. Trên thị trường có nhiều loại cảm biến nhiệt độ, độ ẩm như DHT11, DHT22, SHT10, SHT40, v.v. Sau khi so sánh về giá thành, độ chính xác, khả năng ứng dụng, em quyết định sử dụng cảm biến DHT22 sử dụng cho node cảm biến.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Loại** | **Giá thành  (VND)** | **Điện áp sử dụng (V)** | **Chuẩn giao tiếp** | **Dải đo nhiệt độ (Độ C)** | **Dải đo độ ẩm (%)** | **Sai số nhiệt độ (Độ C)** | **Sai số độ ẩm (%)** | **Tốc độ lấy mẫu (Hz)** |
| **1** | **DHT11** | 40,000 | 3 - 5 | 1-wire | 0 - 50 | 20 - 80 | 2 | 5 | 1 |
| **2** | **DHT21** | 115,000 | 3 - 5 | 1-wire | -40 - 80 | 0 - 100 | 1 | 3 | 0.5 |
| **3** | **DHT22** | 95,000 | 3 - 5 | 1-wire | -40 - 125 | 0 - 100 | 0.5 | 2 - 5 | 0.5 |
| **4** | **HDC1080** | 180,000 | 2.7 - 5.5 | I2C | -40 - 125 | 0 - 100 | 0.2 | 2 |  |
| **5** | **CJMCU-SHT10** | 200,000 | 2.7 - 5.5 | I2C | -40 - 120 | 0 - 100 | 0.1 | 0.1 |  |

Bảng 3. 2. So sánh các loại cảm biến

DHT22 đầu ra hiệu chỉnh tín hiệu kỹ thuật số. Nó sử dụng kỹ thuật thu thập tín hiệu kỹ thuật số và công nghệ cảm biến độ ẩm độc quyền, đảm bảo độ tin cậy và ổn định của nó. Các yếu tố cảm biến của nó được kết nối với máy tính chip đơn 8 bit. Mọi cảm biến của model này đều được bù nhiệt độ và hiệu chuẩn trong buồng hiệu chuẩn chính xác và hệ số hiệu chuẩn được lưu trong loại chương trình trong bộ nhớ OTP, khi cảm biến phát hiện, nó sẽ trích dẫn hệ số từ bộ nhớ. Kích thước nhỏ và mức tiêu thụ thấp và khoảng cách truyền dài (20m) cho phép DHT22 phù hợp trong tất cả các loại dịp ứng dụng khắc nghiệt. Hàng đơn được đóng gói với bốn chân, giúp kết nối rất thuận tiện.

Thông số kỹ thuật hoạt động:

(1) Điện áp của Nguồn và chân Nguồn phải là 3.3-6V DC. Khi nguồn điện được cung cấp cho cảm biến, không gửi bất kỳ hướng dẫn nào đến cảm biến trong vòng một giây để vượt qua trạng thái không ổn định. Một tụ điện có giá trị 100nF có thể được thêm vào giữa VDD và GND để lọc sóng.

(2) Truyền thông và tín hiệu Dữ liệu một bus được sử dụng để liên lạc giữa MCU và DHT22, thời gian 5 ms cho một lần giao tiếp. Dữ liệu bao gồm phần nguyên và phần thập phân

(3) Công thức cho dữ liệu DHT22 gửi bit dữ liệu cao hơn:

Trước tiên, DATA = 8 bit tích phân RH dữ liệu + dữ liệu RH thập phân 8 bit + dữ liệu T tích phân 8 bit + dữ liệu T thập phân 8 bit + tổng kiểm tra 8 bit. Nếu truyền dữ liệu đúng, tổng kiểm tra phải là 8 bit cuối cùng của "dữ liệu RH tích phân 8 bit + dữ liệu RH thập phân 8 bit + dữ liệu T tích phân 8 bit + dữ liệu T thập phân 8 bit".

(4) Khi MCU gửi tín hiệu bắt đầu, DHT22 thay đổi từ chế độ tiêu thụ điện năng thấp sang chế độ chạy. Khi MCU hoàn tất việc gửi tín hiệu bắt đầu, DHT22 sẽ gửi tín hiệu phản hồi của dữ liệu 40 bit phản ánh thông tin về độ ẩm và nhiệt độ tương đối đến MCU. Nếu không có tín hiệu khởi động từ MCU, DHT22 sẽ không cung cấp tín hiệu phản hồi cho MCU. Một tín hiệu bắt đầu cho dữ liệu phản hồi của một lần phản ánh thông tin về độ ẩm và nhiệt độ tương đối từ DHT22. DHT22 sẽ chuyển sang chế độ tiêu thụ điện năng thấp khi thu thập dữ liệu kết thúc nếu nó không nhận được tín hiệu bắt đầu từ MCU một lần nữa.

## 3.2 Thiết kế Firmware

### 3.2.1 Luồng chạy của node IoT

|  |  |  |
| --- | --- | --- |
| **STT** | **Luồng** | **Mô tả** |
| **1** | Nhấn nút 5 lần trên Node | Node vào chế độ smart config |
| **2** | Ứng dụng ESP Touch gửi thông tin wi-fi đến router | Ứng dụng trên điện thoại gửi thông tin wi-fi: ssid và password đã được mã hóa đến Router |
| **3** | Node | Node chạy chế độ wi-fi sniffer, quét và nhận bản tin chứa ssid và password để giải mã. |
| **4** | Node | Kết nối wi-fi sử dụng ssid và password đã giải mã được. |
| **5** | Node gửi thông tin đến ứng dụng ESP Touch | Gửi gói tin UDP chứa thông tin địa chỉ IP của node và ACK xác nhận đã nhận được thông tin wi-fi, giải mã và kết nối thành công. |
| **6** | Node | Kết nối đến broker MQTT với username và password đã cấu hình sẵn trong node  Thông tin MQTT Broker: mqtt://ptitteam.local   * Subcribe vào các topic để nhận bản tin điều khiển * Publish bản tin phản hồi |

*Bảng 3. 3. Luồng hoạt động của node IoT*

### 3.2.2 Lưu đồ thuật toán của node IoT

Diagram

Description automatically generated

*Hình 3. 4. Lưu đồ thuật toán khối xử lý chính*

Trong khối xử lý chính, lưu đồ thuật toán xây dựng trên luồng chạy của thiết bị node IoT. Khi khởi động, node IoT kiểm tra và đọc thông tin wi-fi trong phân vùng nhớ NVS của vi điều khiển ESP32 để kết nối wi-fi. Sau đó, kết nối tới broker MQTT và tạo các tác vụ cần thiết: tác vụ kiểm tra nút nhấn, tác vụ đèn led báo trạng thái hoạt động của node IoT, tác vụ gửi/nhận bản tin MQTT. Trong trường hợp không tìm thấy thông tin wi-fi trong phân vùng NVS thì node IoT ở trạng thái không xác định.

Diagram

Description automatically generated

*Hình 3. 5. Lưu đồ thuật toán khối nút nhấn*

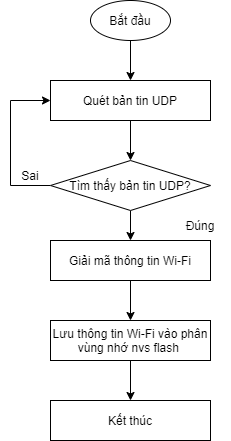
Theo lưu đồ thuật toán khối nút nhấn hình 3.5, khi nhấn nút 5 lần trên node IoT mỗi lần nhấn nút không chậm hơn 1s thì node IoT sẽ chạy vào chế độ smart config để lắng nghe thông tin wi-fi gồm ssid và password từ ứng dụng ESP Touch broadcast. Nếu số lần nhấn nút không đủ 5 lần thì node vẫn tiếp tục ở trạng thái cũ.



*Hình 3. 6. Lưu đồ thuật toán khối LED báo trạng thái*

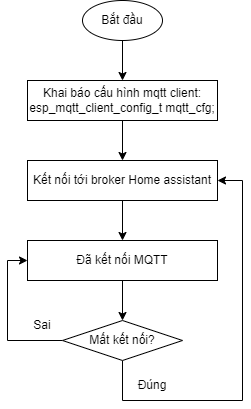
Lưu đồ thuật toán khối đèn led báo trạng thái hoạt động của Node IoT được mô tả trong hình 3.5. Em cấu hình cho node IoT có 5 trạng thái bao gồm:

* **UNKNOW:** Trạng thái không xác định.
* **NORMAL:** Trạng thái hoạt động bình thường, khi đó, node IoT kết nối MQTT ổn định và truyền nhận bản tin.
* **LOCAL MODE:** Trạng thái node IoT có kết nối wi-fi nhưng không có Internet.
* **WIFI DISCONNECTED:** Trạng thái node IoT không có kết nối wi-fi.
* **SMARTCONFIG MODE:** Trạng thái node IoT đang ở chế độ nhận thông tin wi-fi.



*Hình 3. 7. Lưu đồ thuật toán khối smart config*

Trong khối điều khiển smart config lưu đồ thuật toán như hình 3.6. Node IoT vào chế độ wi-fi sniffer, khi này Node IoT sẽ đọc các bản tin UDP có chứa thông tin ssid, password đã được mã hóa. Sau đó, nó sẽ giải mã các bản tin này và lưu vào phân vùng nhớ nvs flash.



*Hình 3. 8. Lưu đồ thuật toán khối điều khiển MQTT*

Lưu đồ thuật toán khối điều khiển kết nối MQTT như trong hình 3.7, node IoT kết nối với broker MQTT với thông tin đã được cấu hình sẵn. URL broker là mqtt://ptitteam.local:1883, username: nmtam, password: 221220.

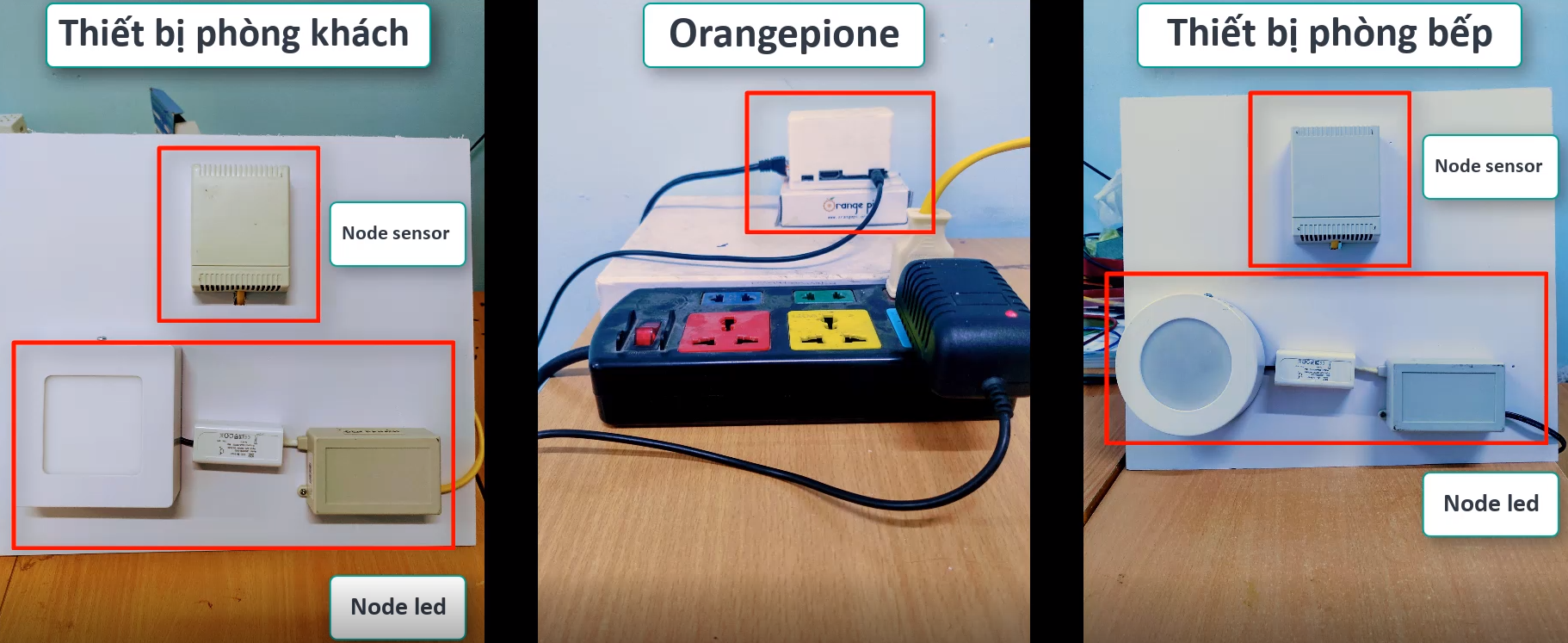
## 3.3 Thử nghiệm, đánh giá

### 3.3.1 Thử nghiệm

Diagram

Description automatically generated

Hình 3. 9. Mô hình sơ đồ kết nối node IoT với Home Assistant



*Hình 3. 10. Hệ thống sản phẩm thực tế*

Trong hình 3.9 là sơ đồ thiết lập các thiết bị trong hệ thống nhà tự động, trong hình 3.10 là hệ thống các thiết bị thực tế. Ở đây, em thiết lập mô phỏng thiết bị trong phòng khách và thiết bị phòng bếp, mỗi phòng gồm 2 thiết bị node là node bật tắt đèn 220V và node cảm biến DHT22 thu nhận dữ liệu nhiệt độ, độ ẩm trong phòng.

#### 3.3.1.a Thử nghiệm hoạt động của node điều khiển đèn 220V

Trước khi thực hiện thử nghiệm, em kết nối Gateway chạy Home Assistant với router wi-fi qua cổng Ethernet và Kết nối wi-fi điện thoại đã tải ứng dụng ESP Touch.

Để kiểm tra hoạt động của node điều khiển đèn, em đã tiến hành lắp đặt node kết nối với đèn Led Maxwin sử dụng điện áp đầu vào 220V như hình 3.10.

Theo lý thuyết nguyên lý mạch điện trong hình 3.1, đèn bật khi chân D4 có mức logic cao, đèn tắt khi chân D4 có mức logic thấp. Trong bảng 3.4 là trạng thái hoạt động của các linh kiện trong thực tế, kết quả trong bảng cho thấy node đèn hoạt động đúng so với lý thuyết.

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Mức logic chân D4** | **Trạng thái triac BT136 - E** | **Trạng thái đèn Led Maxwin** |
| 1 | Cao | Thông | Sáng |
| 2 | Thấp | Ngắt | Tắt |

Bảng 3. 4. Trạng thái của linh kiện trong thực tế

#### 3.3.1.b Thử nghiệm hệ thống với các trường hợp trong thực tế

**Bài 1: Kiểm tra hoạt động chung của hệ thống**

* **Bước 1:**
* Hành động: Bật 4 node, router wi-fi, Gateway
* Kiểm tra các thiết bị có khởi động không?
* **Bước 2:**
* Hành động: Mở Home Assistant web hoặc ứng dụng với đường dẫn: *http://ptitteam:8123*
* Kiểm tra các thiết bị node đã được tự động tích hợp lên Home Assistant không?
* **Bước 3:**
* Hành động: Rút nguồn điện toàn bộ hệ thống. Sau khi ngắt điện hoàn toàn thì cắm điện lại hệ thống.
* Kiểm tra hệ thống có tự hoạt động bình thường trở lại không?
* **Kết quả:**
* 4 node được tự động tích hợp lên Home Assistant
* Sau khi có điện trở lại, hệ thống hoạt động bình thường sau 3-5 phút.

**Bài 2: Kiểm tra khả năng tự kết nối wi-fi của node**

* **Bước 1:**
* Hành động: Bật router wi-fi và 4 node gồm đèn và cảm biến.
* Kiểm tra các node có kết nối được với router không?
* **Bước 2:**
* Hành động: Sau khi các node đã kết nối được, tắt router wi-fi sau đó bật lại.
* Kiểm tra các node có kết nối lại được sau khi router khởi động xong không?
* **Kết quả:**

Sau khi router khởi động xong, các node mất khoảng 2-3 giây để tự động kết nối lại.

**Bài 3: Kiểm tra khả năng tự kết nối MQTT của node**

* **Bước 1:**
* Hành động: Bật router, Gateway và node. Kiểm tra node đã kết nối tới router
* Kiểm tra node có kết nối được với Gateway không?
* **Bước 2:**
* Hành động: Tắt Gateway. Đợi 5 giây sau đó bật lại.
* Kiểm tra node có kết nối lại được với Gateway không?
* **Kết quả:**

Sau khi Gateway khởi động thành công, node mất khoảng 4 giây để tự động kết nối lại tới broker.

**Bài 4: Kiểm tra khả năng tự khởi động lại của node**

* **Bước 1:**
* Hành động: Sau khi node kết nối được MQTT, ra lệnh điều khiển node
* Kiểm tra node có thực hiện đúng yêu cầu không?
* **Bước 2:**
* Hành động: Tắt node. Sau đó bật node.
* Kiểm tra node có tự kết nối lại và nhận lệnh điều khiển không?
* **Kết quả:**

Sau khi bật lại node, node tự khởi động lại sau 5 giây và nhận lệnh điều khiển chính xác.

**Bài 5: Kiểm tra khả năng chịu tải bản tin của node**

* **Bước 1:**
* Hành động: Subscribe một topic bất kì cho node. Gửi một bản tin bất kì
* Kiểm tra node có nhận bản tin đó không?
* **Bước 2:**
* Hành động: Sử dụng một ESP32 khác gửi 10000 bản tin giống nhau đến topic mà node đã subscribe theo chu kì 1000 ms, 500 ms, 100 ms, 10 ms.
* Kiểm tra số lượng bản tin node nhận được có đủ 10000 bản tin không?
* Kiểm tra node có bị reset hoặc mất kết nối với broker không?
* **Kết quả:**
* Các trường hợp gửi bản tin với chu kì trên không làm node bị ngắt kết nối hoặc reset
* Đối với trường hợp gửi với chu kì 10ms thì số lượng bản tin bị mất là 7 bản tin

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Chu kì (ms) | Số bản tin gửi | Số bản tin nhận | Tỉ lệ nhận bản tin (%) | Bị reset | Bị treo | Bị mất kết nối |
| 10 | 10000 | 9993 | 99.93 | Không | Không | Không |
| 100 | 10000 | 10000 | 100 | Không | Không | Không |
| 500 | 10000 | 10000 | 100 | Không | Không | Không |
| 1000 | 10000 | 10000 | 100 | Không | Không | Không |

Bảng 3. 5. Kết quả kiểm tra chịu tải của node

**Bài 6: Kiểm tra thiết lập ngữ cảnh điều khiển**

* **Bước 1:**
* Hành động: Bật hệ thống hoạt động ổn định mô phỏng thiết bị phòng khách và phòng bếp.
* Kiểm tra các thiết bị có hoạt động đúng không?
* **Bước 2:**
* Hành động: Mở ứng dụng Home Assistant, cài đặt các ngữ cảnh sau:
* Hẹn giờ tắt đèn, bật đèn
* Cảnh báo trên màn hình nếu nhiệt độ phòng bếp cao hơn 60oC
* Tắt đèn phòng bếp khi đèn phòng khách tắt
* Kiểm tra các ngữ cảnh trên có hoạt động đúng không
* **Kết quả:**

Các node kết hợp với nhau thực hiện đúng các ngữ cảnh đã cài đặt.

### 3.3.2 Đánh giá

#### 3.3.2.a Kết quả

* Các node tự động tích hợp lên Home Assistant
* Hệ thống hoạt động chính xác, ổn định
* Độ trễ thời gian điều khiển nhỏ hơn 200ms
* Dựa vào đồ thị dòng điện tiêu thụ của node cảm biến, mức tiêu thụ năng lượng của node ở chế độ hoạt động bình thường mức tiêu thụ năng lượng của node là khoảng 95 mA, ở chế độ ngủ là khoảng 1mA.

#### 3.3.2.b Ưu điểm

* Việc điều khiển phản hồi nhanh, chính xác, ổn định
* Tính tương thích cao, dễ tích hợp
* Độ bảo mật cao do không có bản tin đi ra ngoài Internet
* Giá thành thấp hơn so với các sản phẩm trên thị trường

#### 3.3.2.c Nhược điểm

* Điều khiển cục bộ trong nhà
* Quá nhiều thiết bị sẽ tăng tải cho Router Wi-Fi

## Kết luận chương

Chương này trình bày về thiết kế phần cứng, phần mềm cũng như cách thức hoạt động và các chức năng thành phần của thiết bị. Để từ đó có thể giúp người đọc hiều được thêm về ứng dụng của thiết bị trong cuộc sống. Một kịch bản thử nghiệm cũng đã được triển khai, đánh giá chi tiết về mức độ tự động hóa, độ ổn định và hiệu năng của thiết bị.

# KẾT LUẬN

Sau thời gian gần 3 tháng nghiên cứu với sự nỗ lực của bản thân và sự hướng dẫn tận tình của Thầy TS. Nguyễn Trung Hiếu, em đã hoàn thành đồ án “Nghiên cứu, thiết kế thiết bị node IoT tương thích với Home Assistant”.

**Kết quả đồ án đã đạt được những nội dung cụ thể sau đây:**

* Nghiên cứu tổng quan về home automation, ưu nhược điểm, các giải pháp tích hợp trong home automation.
* Tìm hiểu về nền tảng quản lý, điều khiển thiết bị Home Assistant.
* Hiểu được cách hoạt động của công nghệ kết nối wi-fi và các giao thức MQTT, ESP Touch, One-wire.
* Thiết kế thành công thiết bị, thử nghiệm, đánh giá chi tiết.

**Hướng phát triển của đồ án:**

* Tiếp tục hoàn thiện giải pháp tự động hóa, tích hợp thêm tiện ích cho hệ thống.
* Do việc sử dụng công nghệ kết nối wi-fi nên khi số lượng thiết bị tăng thì có thể làm cho router bị quá tải. Cần nghiên cứu các cơ chế để giúp giảm tải đường truyền đến router.
* Việc điều khiển thiết bị chỉ điều khiển được cục bộ trong một mạng WLAN. Vì vậy, cần tích hợp thêm điều khiển thiết bị từ xa như thuê dịch vụ cloud của Home Assistant, sử dụng dịch vụ Duck DNS, hoặc sử dụng chatbot của các nền tảng chat như Telegram, Messenger, Whats App, v.v.
* Tăng cường cơ chế bảo mật: Để tăng độ tin cậy, an toàn cho hệ thống, cần có thêm cơ chế để bảo mật như mã hóa đầu cuối bản tin gửi đi. Điều này cũng đồng thời làm tăng thời gian xử lý của hệ thống khi nhiều thiết bị cùng kết nối.

Qua quá trình thực hiện đề tài nghiên cứu, được sự hướng dẫn tận tình của các Thầy Cô, em đã được làm việc và nghiên cứu một cách khoa học hơn, hiệu quả hơn. Qua đó, phần nào thể hiện được niềm đam mê và khát khao làm chủ công nghệ của mình. Những kiến thức đạt được sau quá trình học tập, nghiên cứu tại Học viện Công nghệ Bưu chính Viễn thông và nhất là những hiểu biết thu được sau khi hoàn thiện đồ án tốt nghiệp, sẽ rất có ích cho em trong quá trình làm việc sau này.

Mặc dù đã có nhiều cố gắng nhưng đề tài khó tránh khỏi những sai sót, em rất mong nhận được sự góp ý từ các Thầy Cô giáo để có thể hoàn thiện hơn nữa.

Em xin chân thành cảm ơn!

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | N. T. L. T. P. V. Minh Anh Tuan Tran, "Smart-Config Wifi Technology Using ESP8266 for Low-Cost Wireless Sensor Networks," *International Conference on Advanced Computing and Applications (ACOMP),* 2018. |
| [2] | I. R. 1122, Requirements for Internet Host – Communication Layers, Sept. 1989. |
| [3] | P. Semiconductors, Datasheet BT136 Series. |
| [4] | P.Newnam, "The Internet of Things 2017 Report," *BI Intelligence,* 2017. |
| [5] | T. I. Incorporated, "CC3000 SmartConfig Getting Starter," Texas Instruments Incorporated, 24 September 2013. [Online]. Available: http://processors.wiki.ti.com/index.php/CC3000\_SmartConfig\_Getting\_Started. [Accessed 19 September 2022]. |
| [6] | E. Incorporated, "ESP-TOUCH User Guide," ESP-TOUCH User Guide, 12 April 2016. [Online]. Available: http://www.espressif.com/sites/default/files/30b-esptouch\_user\_guide\_en\_v1.1\_20160412\_0.pdf. [Accessed 25 September 2022]. |
| [7] | D. A. a. D. Biswajit, "A Study on Energy Consumption of Different," 28 December 2012. [Online]. Available: http://www.ijert.org/view-pdf/1847/a-study-on-energy-consumption-of-differentwireless-devices. [Accessed 30 Semptember 2022]. |
| [8] | E. Incorporated, "ESP32 Datasheet," 4 April 2017. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32\_datasheet\_en.pdf. [Accessed 20 September 2022]. |
| [9] | N. Neshenko, E. Bou-Harb and e. al, "Neshenko, N.; Bou-Harb, E.; et al," *IEEE Communications Surveys & Tutorials,* vol. 21, no. 3, pp. 2702-2733, 2019. |
| [10] | E. Y. Vasserman and N. V. Hopper, "Draining life from wireless ad hoc sensor networks," *IEEE transactions on mobile computing,* vol. 12, no. 2, pp. 318-332, 2011. |
| [11] | P. J. C. Gomez, "Wireless home automation networks: A survey of architectures and technologies," *of architectures and technologies,* vol. 48, no. 6, pp. 92-101, 2010. |