**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**TRƯỜNG ĐIỆN - ĐIỆN TỬ**

**A red and yellow logo

Description automatically generated**

**Báo cáo Bài TẬp Lớn**

***Đề tài: Thiết kế hệ thống giám sát môi trường thông minh sử dụng IoT***

**EE4266 - Mạng và các giao thức trong IoT**

**Nhóm 18**

Giảng viên hướng dẫn: PGS.TS Nguyễn Quốc Cường

Sinh viên thực hiện: Tạ Xuân Kiên (Đại diện nhóm) 20202428

Nguyễn Đỗ Hồng Phương 20202490

**Bảng cập nhật báo cáo hàng tuần**

|  |  |  |
| --- | --- | --- |
| **Tuần** | **Mục** | **Nội dung cập nhật** |
| 8 | Kế hoạch thực hiện chung và riêng của từng thành viên trong dự án. | * (Sửa đổi) Phần công việc thực hiện chung của nhóm * (Thêm mới)Thêm phần phân công công việc phụ trách của từng thành viên. |
| Bảng cập nhật báo cáo hàng tuần | * (Thêm mới) Thêm bảng cập nhật báo cáo hàng tuần của nhóm. |
| Trang bìa | * (Thêm mới) Bổ sung trang bìa báo cáo. |
| Mục lục | * (Thêm mới) Bổ sung mục lục báo cáo. |
| Tổng quan dự án | * (Bổ sung) Thông tin về đầu ra và đầu vào của các khối chức năng |
| 9 | Lựa chọn giải pháp và phương án thiết kế | * (Thêm mới)Thêm sơ đồ triển khai của hệ thống |
| 10 | Lựa chọn giải pháp và phương án thiết kế | * (Thêm mới)Thêm phần tìm hiểu lí thuyết và lựa chọn giao thức MQTT, CoAP |
| Kế hoạch thực hiện chung của dự án | * (Sửa đổi) Chỉnh sửa kế hoạch thực hiện |
| 11 | Kế hoạch thực hiện chung của dự án | * (Sửa đổi) Chi tiết các các công việc trong khối công việc lớn |
| Sơ đồ triển khai | * (Sửa đổi) Thay HVAC sử dụng led mô phỏng thàn LCD I2C |
| Lựa chọn và thiết kế phần cứng | * (Thêm mới)Thiết kế khối chức năng phần cứng |
| 12 | Thiết kế thiết bị kết nối MQTT | * (Thêm mới)Thiết kế các cảm biến sử dụng MQTT sử dụng MQTT để truyền thông * (Thêm mới) Thiết kế cơ cấu chấp hành sử dụng MQTT để truyền thông * (Thêm mới)Thiết kế phần gateway sử dụng mqtt để truyền thông |
| 13 | Thiết kế thiết bị kết nối mqtt | * (Thêm mới)Thiết kế thiết bị và phần gateway sử dụng mqtt để truyền thông |
| 14 | Thiết kế thiết bị sử dụng giao thức kết nối CoAP | * (Thêm mới) Thiết kế thiết bị và phần gateway sử dụng CoAP |
| Thiết kế phần cứng | * (Bổ sung) Phần cứng thiết bị |
| 15 | Thiết kế gateway | * (Thêm mới) Thiết kế gateway cho cả 2 giao thức MQTT và CoAP |
| 16 | Lựa chọn nền tảng IoT | * (Thêm mới) Lựa chọn nền tảng IoT Thingsboard |
| 17 | Kết quả đạt được | * (Thêm mới) Hình ảnh kết quả đạt được ( hình ảnh thực tế và giao diện |
| Đầu mục báo cáo | * (Sửa đổi) Sửa đổi một số đầu mục báo cáo |

Mục lục

[**Yêu cầu của dự án** 6](#_Toc169866114)

[**Kế hoạch thực hiện chung của dự án** 7](#_Toc169866115)

[**Đánh giá tỷ lệ đóng góp của từng thành viên trong dự án** 9](#_Toc169866116)

[**CHƯƠNG 1: PHÂN TÍCH YÊU CẦU DỰ ÁN** 10](#_Toc169866117)

[1.1. Tổng quan dự án 10](#_Toc169866118)

[1.2. Tìm hiểu các nghiên cứu, dự án liên quan 12](#_Toc169866119)

[1.2.1. Performance evaluation of CoAP and MQTT with security support for IoT environments 12](#_Toc169866120)

[1.2.2. Indoor Air Quality Monitoring Systems Based on Internet of Things: A Systematic Review 12](#_Toc169866121)

[**CHƯƠNG 2: PHƯƠNG ÁN THIẾT KẾT** 13](#_Toc169866122)

[2.1. Sơ đồ triển khai 13](#_Toc169866123)

[2.2. Tổng quan giao thức và lựa chọn công nghệ 13](#_Toc169866124)

[2.2.1. MQTT 13](#_Toc169866125)

[2.2.2. CoAP 15](#_Toc169866126)

[2.2.3. Nền tảng IoT 16](#_Toc169866127)

[**CHƯƠNG 3: THIẾT KẾ PHẦN CỨNG HỆ THỐNG** 18](#_Toc169866128)

[3.1. Khối điều khiển trung tâm 18](#_Toc169866129)

[3.2. Nút cảm biến 19](#_Toc169866130)

[3.3. Cơ cấu chấp hành HVAC 19](#_Toc169866131)

[3.3.1. Sơ đồ khối chức năng 19](#_Toc169866132)

[3.3.2. Lựa chọn phần cứng 20](#_Toc169866133)

[**CHƯƠNG 4: THIẾT KẾ PHẦN MỀM HỆ THỐNG** 21](#_Toc169866134)

[4.1. Thiết kế thiết bị kết nối MQTT 21](#_Toc169866135)

[4.1.1. Thiết kế cảm biến – DHT11 21](#_Toc169866136)

[4.1.2. Thiết kế cơ cấu chấp hành – HVAC 22](#_Toc169866137)

[4.2. Thiết kế thiết bị kết nối CoAP 23](#_Toc169866138)

[4.2.1. Cảm biến ánh sáng và chất lượng không khí 23](#_Toc169866139)

[4.2.2. Đèn chiếu sáng 23](#_Toc169866140)

[4.3. Thiết kế gateway 24](#_Toc169866141)

[**Kết quả đạt được** 25](#_Toc169866142)

[**Hướng phát triển đề tài** 26](#_Toc169866143)

[Hình 1‑1 Cấu trúc sơ bộ hệ thống 10](#_Toc169866199)

[Hình 2‑1 Sơ đồ triển khai hệ thống 13](#_Toc169866200)

[Hình 2‑2 Tổng quan kiến trúc giao thức MQTT 14](#_Toc169866201)

[Hình 2‑3. Tổng quan về nguyên lý hoạt động CoAP 16](#_Toc169866202)

[Hình 2‑4 Nền tảng IoT mã nguồn mở Thingsboard 17](#_Toc169866203)

[Hình 3‑1. ESP WROOM 32 18](#_Toc169866204)

[Hình 3‑2 Sơ đồ khối chức năng phần cứng cảm biến 19](#_Toc169866205)

[Hình 3‑3 Sơ đồ khối chức năng phầ cứng cơ cấu chấp hành 19](#_Toc169866206)

[Hình 3‑4 Màn hình LCD 1602 20](#_Toc169866207)

[Hình 4‑1 Biểu đồ dữ liệu theo topic của cảm biến 21](#_Toc169866208)

[Hình 4‑2 Biểu đồ tuần tự giao tiếp với cảm biến 21](#_Toc169866209)

[Hình 4‑3 Tổng quan hoạt động kết nối các topic MQTT cho HVAC 22](#_Toc169866210)

[Hình 4‑4 Biểu đồ tuần tự quá trình nhận dữ liệu của HVAC 22](#_Toc169866211)

[Hình 5‑2 Giao diện Thingsboard 25](#_Toc169866212)

[Hình 5‑1 Hình ảnh thực tế kết nối các khối hệ thống 25](#_Toc169866213)

# **Yêu cầu của dự án**

*Ghi chú: 1(cao nhất) – 3 (thấp nhất)*

|  |  |  |
| --- | --- | --- |
| **Yêu cầu** | **Chức năng, Thông số,..** | **Mức độ ưu tiên** |
| Thiết bị IoT:   * Cảm biến * Thiết bị điều khiển * Vi điều khiển | Cảm biến:   * Nhiệt độ( oC) * Độ ẩm (%) * Chất lượng không khí (ppm) * Cường độ ánh sáng (cd hoặc lx)   Thiết bị điều khiên:   * HVAC * Điều khiển ánh sáng   Vi điều khiển: tối thiểu 1 vi điều khiển cho 1 khối chức thiết bị | 2 |
| Môi trường | Trong phòng kín ở các chung cư | 3 |
| Gateway | Làm trung gian giao tiếp giữa thiết bị IoT và Thingsboard  Giao tiếp thiết bị: MQTT và CoAP.  Giao tiếp với nền tảng IoT: MQTT.  Khả năng: Tiền xử lý dữ liệu, quản lý thiết bị và truyền dữ liệu an toàn.  Xử lý và đưa ra tín hiệu điều khiển | 1 |
| Giao thức truyền thông:   * MQTT * CoAP | MQTT: thiết bị cần giám sát liên tục và cập nhật thời gian thực  CoAp: thiết bị cập nhật ít thường xuyên và có lệnh điều khiển đơn giản | 1 |
| Nền tảng IoT | Quản lý và Giám sát thiết bị  Trực quan hóa dữ liệu  - Điều khiển tự động thiết bị dựa trên dữ liệu cảm biến và quy tắc do người dùng định nghĩa.  Phân Tích Dữ Liệu:  - Lưu trữ và phân tích dữ liệu cảm biến lịch sử.  - Giao diện thân thiện với người dùng để giám sát và điều khiển thiết bị IoT.  Kiểm Soát Dựa Trên Web  - Điều khiển từ xa thiết bị IoT thông qua giao diện web.  Xác Thực Người Dùng và Bảo Mật:  - Đăng nhập an toàn, mã hóa dữ liệu và quyền truy cập dựa trên vai trò  Thông Báo và Cảnh Báo:  - Thay đổi ngưỡng cảnh báo | 1 |

# **Kế hoạch thực hiện chung của dự án**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nội dung** | **Mô tả** | **Người thực hiện** | **Thời gian (tuần)** | **Ghi chú** |
| Tìm hiểu tổng quan đề tài | Tìm hiểu được khái niệm, thành phần, phân loại hệ thống giám sát ứng dụng IoT.  Ứng dụng, vấn đề được giải quyết của hệ thống. | Kiên, Phương | T6 |  |
| Phân tích yêu cầu dự án, gán mức ưu tiên cho từng dự án | Đưa ra được bảng phân tích các đầu mục công việc của dự án và mức ưu tiên của các yêu cầu dự án. | Kiên, Phương | T7 |  |
| Tìm hiểu về các bài toán liên quan | Báo cáo tổng quan, đánh giá | Kiên, Phương | T7 |  |
| Thiết kế sơ đồ khối chức năng, sơ đồ triển khai cụ thể | Phân tích đưa ra các khối chức năng cụ thể của dự án và sự kết nối giữa các khối với nhau.  Giải thích các khối chức năng mô tả thông tin đầu vào đầu ra của dự án. | Kiên, Phương | T8 |  |
| Lựa chọn linh kiện, giao thức truyền thông, cổng biên, nền tảng IoT, kết nối mạng.  Xây dựng sơ đồ triển khai | Xác định phần cứng sử dụng.  Tìm hiểu tổng quan về các giao thức MQTT và CoAP.  Lựa chọn công cụ, thư viện cho các giao thức.  Sơ đồ triển khai hệ thống | Kiên, Phương | T9-T10 |  |
| Thiết kế phần mềm hệ thống | Thiết bị cảm biến sử dụng MQTT   * Đọc dữ liệu cảm biến (T11) * Phần mềm truyền đến 1 broker (T12)   Thiết bị cảm biến sử dụng CoAP   * Đọc dữ liệu cảm biến (T12) * Phần mềm truyền dữ liệu qua CoAP (T13)   Truyền dữ liệu lên Thingsboard   * Tạo giao diện (T14) * Kết nối với gateway (T14) * Truyền dữ liệu từ gateway lên Thingsboard (T15) | Phương | T11-T14 |  |
|  | Thiết bị điều khiển sử dụng MQTT   * Kết nối với cơ cấu chấp hành (T11) * Phần mềm nhận dữ liệu từ 1 broker (T12)   Thiết bị điều khiển sử dụng CoAP   * Kết nối với cơ cấu chấp hành (T12) * Phần mềm nhận dữ liệu qua CoAP (T13) * Xử lý tín hiệu nhận được theo ngưỡng. (T13)   Truyền dữ liệu từ Thingsboard   * Tạo giao diện (T14) * Kết nối với gateway (T14) * Nhận dữ liệu từ ngưỡng từ Thingsboard về gateway(T15). | Kiên |  |
| Hoàn thiện hệ thống | Ghép nối các thành phần của các thành viên với nền tảng được cung cấp.   * Đánh giá các công việc khi chuyển sang nền tảng mới (T15) * Phân chia công việc theo hướng làm nền tảng cũ. (T15) * Đánh giá hệ thống (T16) | Kiên, Phương | T15-16 |  |
| Tuần dự trữ | Dự dữ để sửa lỗi. |  | T17 |  |
| Thử nghiệm và đánh giá sản phẩm | Thử nghiệm và đánh giá sản phẩm dựa vào các yêu cầu đặt ra ban đầu. |  | T18 |  |

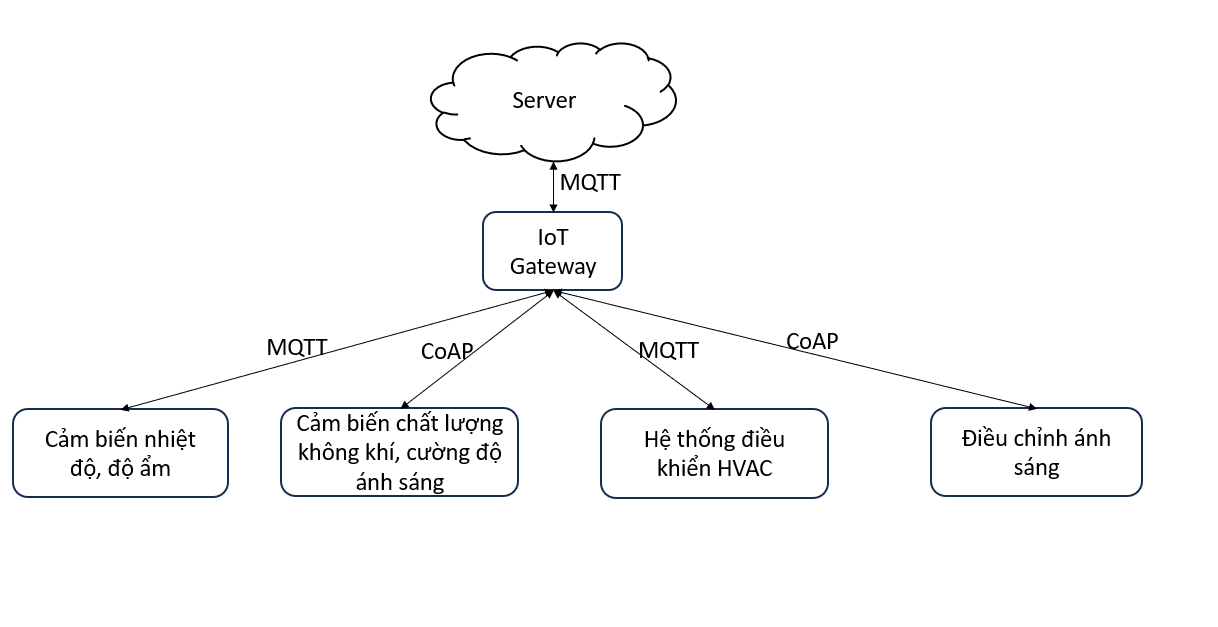
# **Đánh giá tỷ lệ đóng góp của từng thành viên trong dự án**

|  |  |  |
| --- | --- | --- |
| Người thực hiện | Tỷ lệ | Giải quyết được những vấn đề gì của dự án (cần ghi rõ để có cơ sở đánh giá tỷ lệ) |
| Tạ Xuân Kiên | 50% | - Tìm hiểu tổng quan đề tài.  - Phân tích yêu cầu đề tài.  - Tìm hiểu các bài toán liên quan.  - Lựa chọn giao thức, kết nối mạng.  - Lập trình trao đổi dữ liệu dựa trên các giao thức đã chọn  - Nhận dữ liệu từ gateway và Thingsboard để điều khiển các cơ cấu chấp hành |
| Nguyễn Đỗ Hồng Phương | 50% | - Tìm hiểu tổng quan đề tài.  - Phân tích yêu cầu đề tài.  - Thiết kế sơ đồ khối chức năng hệ thống.  - Lựa chọn linh kiện phần cứng.  - Lập trình đọc dữ liệu cảm biến.  - Truyền dữ liệu cảm biến lên gateway và Thingsboard.  - Trực quan dữ liệu |

# **CHƯƠNG 1: PHÂN TÍCH YÊU CẦU DỰ ÁN**

## **1.1. Tổng quan dự án**

1. *Khái niệm*: Hệ thống giám sát môi trường thông minh sử dụng Internet of Things (IoT) là một hệ thống được thiết kế để thu thập, quản lý, và phân tích dữ liệu về môi trường xung quanh như nhiệt độ, độ ẩm, chất lượng không khí, ô nhiễm không khí, độ ồn, ánh sáng,... bằng cách sử dụng các thiết bị cảm biến, dữ liệu được phân tích, xử lý rồi đưa các tin hiệu đến điều khiển cơ cấu chấp hành.
2. *Thành phần* : Hệ thống giám sát ứng dụng IoT bao gồm các thành phần cơ bản sau:
   * Khối cảm biến
   * Gateway
   * Server
   * Cơ cấu chấp hành
3. *Phân loại*: Phân loại hệ thống giám sát môi trường thông minh có thể dựa trên nhiều tiêu chí, bao gồm kích thước (quy mô), phạm vi ứng dụng, công nghệ sử dụng, và mục đích sử dụng. Các hệ thống này có thể được phân loại thành các loại như:
   * Hệ thống giám sát đô thị
   * Hệ thống giám sát nông nghiệp
   * Hệ thống giám sát công nghiệp
4. *Ứng dụng*: Giám sát nhiệt độ môi trường trong phòng ở các tòa nhà chung cư
5. *Vấn đề được giải quyết:*
   * Điều hòa chất lượng không khí, nhiệt độ
   * Cảnh báo ô nhiễm
   * Hỗ trợ quản lý, tiết kiệm tài nguyên: ánh sáng, nhiệt độ
6. *Cấu trúc sơ bộ hệ thống*



Hình 1‑1 Cấu trúc sơ bộ hệ thống

1. *Chức năng chính của các khối chức năng*

* Khối cảm biến:
* Thu thập thông tin môi trường: nhiệt độ, độ ẩm, chất lượng không khí, ánh sáng
* Truyền dữ liệu thu thập lên gateway thông qua giao thức truyền tin phù hợp
* Đầu vào: Dữ liệu mang thông tin về môi tường xung quanh
* Đầu ra: Dạng dữ liệu phù hợp (chuỗi kí tự) được mã hóa để truyền thông
* Khối chấp hành:
* Điều khiển các cơ cấu chấp hành theo tin hiệu điều khiển
* Nhận tín hiệu điều khiển từ gateway thông qua giao thức phù hợp
* Xử lý dữ liệu cần thiết để điều khiển thiết bị
* Đầu vào: Dữ liệu mang tín hiệu điều khiển nhận từ gateway
* Đầu ra: Các biến đổi tác động đến môi trường ứng với các tín hiệu điều khiển
* Gateway:
* Làm trung gian chuyển tiếp dữ liệu giữa cảm biến và cloud
* Cập nhật dữ liệu ngưỡng cảnh báo từ cloud
* Xử lý ngưỡng cảnh báo và đưa ra tín hiệu điều khiển cho khối chấp hành
* Đầu vào:

+ Dữ liệu từ khối cảm biến

+ Tín hiệu tiền xử lý từ server: ngưỡng cảnh báo

* Đầu ra:

+ Dữ liệu phù hợp để truyền thông lên server

+ Tín hiệu điều khiển đến các khối chấp hành

* Máy chủ, giao diện:
* Nhận dữ liệu cảm biến từ gateway
* Trực quan hóa dữ liệu
* Lưu trữ dữ liệu
* Nhập ngưỡng cảnh báo từ người dùng
* Truyền ngưỡng cảnh báo cho gateway.
* Đầu vào:

+ Dữ liệu cảm biến nhận từ gateway

+ Ngưỡng cảnh báo được nhập từ giao diện người dùng

+ Tín hiệu điều khiển nhập từ giao diện người dùng: bật/tắt đèn, thông số cơ cấu chấp hành,...

* Đầu ra:

+ Giao diện người dùng

+ Dữ liệu cập nhật, điều khiển đến gateway

## **1.2. Tìm hiểu các nghiên cứu, dự án liên quan**

### **1.2.1. Performance evaluation of CoAP and MQTT with security support for IoT environments**

Bài báo này phân tích hiệu suất của hai giao thức phổ biến nhất ở tầng ứng dụng: Constrained Application Protocol (CoAP) và Message Queue Telemetry Transport (MQTT).

Vì các thiết bị IoT thường hoạt động dưới ràng buộc về pin, phân tích chủ yếu tập trung vào băng thông, độ trễ và sử dụng CPU.

Các kết luận chính của nghiên cứu là giao thức MQTT đòi hỏi băng thông hơn vì nó thêm vào chi phí của TCP. Ngược lại, nó cung cấp các cơ chế QoS khác nhau và mặc định cung cấp tính tin cậy (cho việc chạy qua TCP), trong khi CoAP chỉ cung cấp một cơ chế QoS đơn giản và tin cậy thông qua các thông điệp có thể xác nhận hoặc không thể xác nhận.

Về đánh giá độ trể, ở điều kiện độ tin cậy cao thì độ trễ của 2 giao thức là tương đương. Tuy nhiên, khi độ tin cậy bi giảm xuống, độ trễ của CoAP sẽ cao hơn MQTT.

Dựa vào bài nghiên cứu, ta xác định được sử dụng 2 giao thức cho từng loại dữ liệu trong hệ thống:

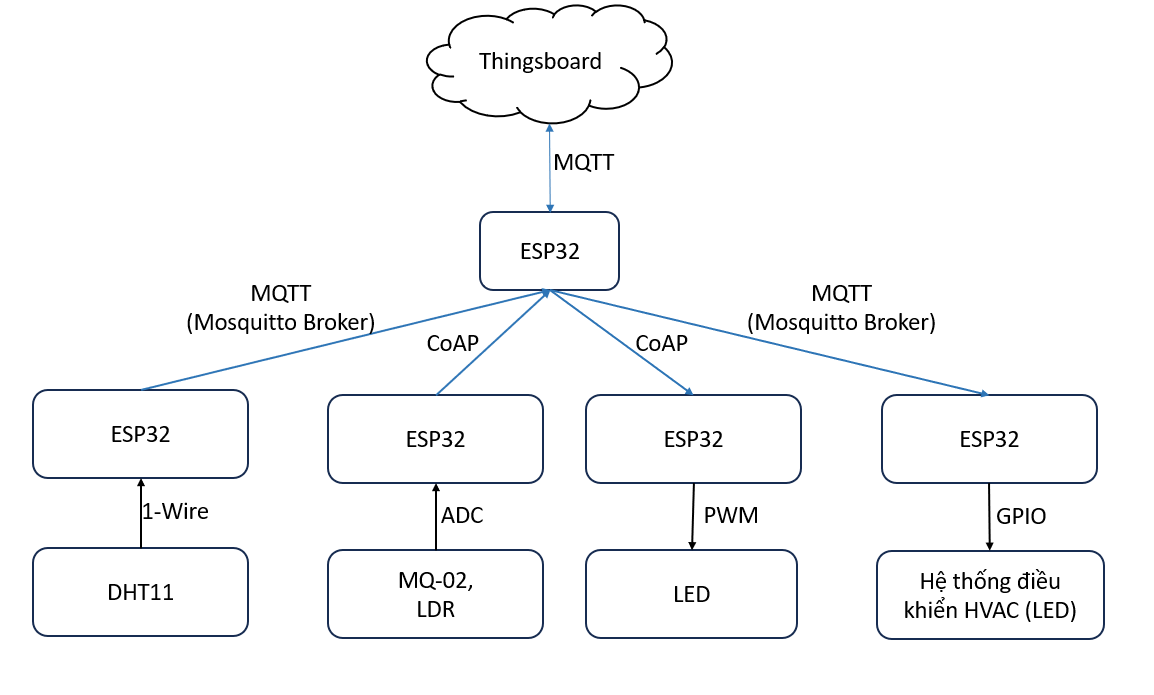
* Với dữ liệu cảm biến cần giám sát và cập nhật thời gian thực thì MQTT được áp dụng
* Với dữ liệu cập nhật ít thường xuyên và không yêu cầu cao về thời gian thực thì CoAP được sử dụng.

### **1.2.2. Indoor Air Quality Monitoring Systems Based on Internet of Things: A Systematic Review**

Mục tiêu chính của công việc này là trình bày một bài đánh giá hệ thống giám sát chất lượng không khí trong nhà dựa trên Internet of Things (IoT). Tài liệu này làm nổi bật các khía cạnh thiết kế cho các hệ thống giám sát, bao gồm loại cảm biến, vi điều khiển, kiến ​​trúc và kết nối cùng với các vấn đề triển khai của các nghiên cứu được công bố trong năm năm trước đó (2015–2020).

# **CHƯƠNG 2: PHƯƠNG ÁN THIẾT KẾT**

## **2.1. Sơ đồ triển khai**

*Hình 2‑1 Sơ đồ triển khai hệ thống*

## **2.2. Tổng quan giao thức và lựa chọn công nghệ**

### **2.2.1. MQTT**

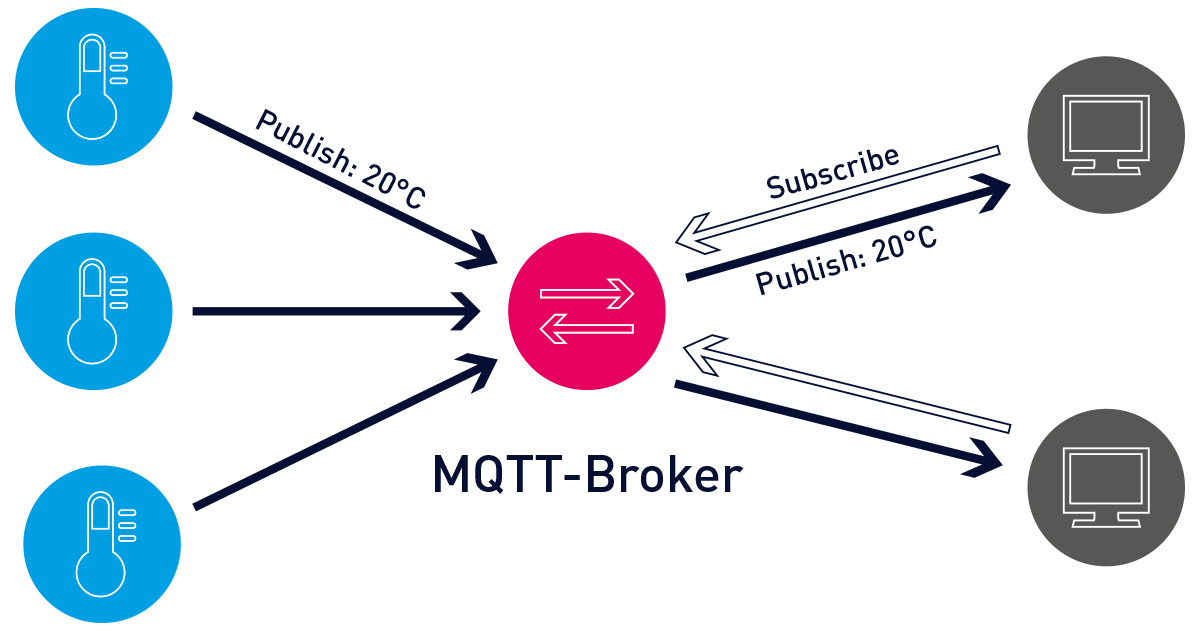
MQTT là một giao thức nhắn tin dựa trên các tiêu chuẩn hoặc một bộ các quy tắc được sử dụng cho việc giao tiếp máy với máy. Cảm biến thông minh, thiết bị đeo trên người và các thiết bị Internet vạn vật (IoT) khác thường phải truyền và nhận dữ liệu qua mạng có tài nguyên và băng thông hạn chế. Các thiết bị IoT này sử dụng MQTT để truyền dữ liệu vì giao thức này dễ triển khai và có thể giao tiếp dữ liệu IoT một cách hiệu quả. MQTT hỗ trợ nhắn tin giữa các thiết bị với đám mây và từ đám mây đến thiết bị.

1. *Nguyên tắc đằng sau MQTT là gì?*

Giao thức MQTT hoạt động trên nguyên tắc của mô hình xuất bản/đăng ký. Trong quá trình giao tiếp mạng truyền thống, máy khách và máy chủ giao tiếp trực tiếp với nhau. Máy khách yêu cầu tài nguyên hoặc dữ liệu từ máy chủ, sau đó máy chủ xử lý và gửi lại một phản hồi. Tuy nhiên, MQTT sử dụng mẫu xuất bản/đăng ký để phân tách bên gửi thông điệp (nơi gửi thông điệp) khỏi bên nhận thông điệp (nơi nhận thông điệp). Thay vào đó, một thành phần thứ ba được gọi là trình truyền tải thông điệp xử lý quá trình giao tiếp giữa nơi gửi thông điệp và nơi nhận thông điệp. Việc của trình truyền tải là lọc tất cả các thông điệp đến từ nơi gửi thông điệp và phân phối các thông điệp đó một cách chính xác đến nơi nhận thông điệp. Trình truyền tải phân tách nơi gửi thông điệp và nơi nhận thông điệp như sau:

* Phân tách không gian: Nơi gửi thông điệp và nơi nhận thông điệp không biết về vị trí của nhau trong mạng và không trao đổi các thông tin như địa chỉ IP và số cổng.
* Phân tách thời gian: Nơi gửi thông điệp và nơi nhận thông điệp không chạy hoặc có kết nối mạng tại cùng một thời điểm.
* Phân tách quá trình đồng bộ: Cả nơi gửi thông điệp và nơi nhận thông điệp có thể gửi hoặc nhận thông điệp mà không làm gián đoạn hoạt động của bên còn lại. Ví dụ: nơi nhận thông điệp không phải đợi nơi gửi thông điệp thực hiện gửi thông điệp.

1. *Các thành phần của MQTT*



Hình 2‑2 Tổng quan kiến trúc giao thức MQTT

MQTT triển khai mô hình xuất bản/đăng ký bằng cách định nghĩa máy chủ và trình truyền tải:

* Máy khách MQTT: Một máy khách MQTT là bất kỳ thiết bị nào từ một máy chủ đến một bộ vi điều khiển có chạy một thư viện MQTT. Nếu máy khách gửi thông điệp, nó hoạt động như một nơi gửi thông điệp và nếu nhận thông điệp, nó hoạt động như bên nhận. Về cơ bản, bất kỳ thiết bị nào giao tiếp bằng MQTT qua một mạng đều có thể được gọi là một thiết bị khách MQTT.
* Trình truyền tải MQTT (broker): Trình truyền tải MQTT là hệ thống backend điều phối thông điệp giữa các máy khách khác nhau. Trách nhiệm của trình truyền tải bao gồm nhận và lọc thông điệp, xác định máy khách đã đăng ký nhận từng thông điệp và gửi thông điệp cho các máy khách đó. Trình trung chuyển cũng chịu trách nhiệm thực hiện các tác vụ khác như:

+ Ủy quyền và xác thực máy khách MQTT

+ Chuyển thông điệp đến các hệ thống khác để phân tích thêm

+ Xử lý các thông điệp bị bỏ lỡ và các phiên trên máy khách

* Kết nối MQTT: Máy khách và máy chủ bắt đầu giao tiếp bằng một kết nối MQTT. Máy khách khởi tạo kết nối bằng cách gửi một thông điệp CONNECT (KẾT NỐI) đến trình truyền tải MQTT. Trình truyền tải xác nhận rằng kết nối đã được thiết lập bằng cách trả lời bằng một thông điệp CONNACK. Cả máy khách MQTT và trình truyền tải đều cần đến ngăn xếp TCP/IP để giao tiếp. Các máy khách không bao giờ kết nối với nhau mà chỉ kết nối với trình truyền tải.

1. *Nguyên lý hoạt động của MQTT*

Dưới đây là tổng quan về cách thức hoạt động của MQTT.

* Máy khách MQTT thiết lập kết nối với trình truyền tải MQTT.
* Sau khi được kết nối, máy khách có thể xuất bản thông điệp, đăng ký nhận các thông điệp cụ thể hoặc thực hiện cả hai.
* Khi trình truyền tải MQTT nhận được một thông điệp, trình truyền tải sẽ chuyển thông điệp đó đến những nơi nhận thông điệp quan tâm.

*Chủ đề MQTT*: Thuật ngữ “chủ đề” đề cập đến các từ khóa trình truyền tải MQTT sử dụng để lọc các thông điệp cho máy khách MQTT. Các chủ đề được tổ chức theo thứ bậc, tương tự như đường dẫn đến tệp hoặc thư mục.

*Quá trình xuất bản qua MQTT*: Máy khách MQTT xuất bản các thông điệp có chứa chủ đề và dữ liệu dưới định dạng byte. Máy khách xác định định dạng dữ liệu như dữ liệu văn bản, dữ liệu nhị phân, tệp XML hoặc JSON. Ví dụ: một cái đèn trong hệ thống nhà ở thông minh có thể xuất bản thông điệp về chủ đề phòng khách/ánh sáng.

*Quá trình đăng ký qua MQTT*: Máy khách MQTT gửi một thông điệp SUBSCRIBE (ĐĂNG KÝ) đến trình truyền tải MQTT để nhận thông điệp về các chủ đề quan tâm. Thông điệp này chứa một mã định danh duy nhất và một danh sách đăng ký. Ví dụ: ứng dụng nhà ở thông minh trên điện thoại của bạn muốn hiển thị số lượng đèn đang bật trong nhà của bạn. Ứng dụng này sẽ đăng ký chủ đề ánh sáng và gia tăng bộ đếm trên tất cả các thông điệp về trạng thái bật.

1. *Lựa chọn MQTT broker*

Như vậy, để kết nối các thiết bị với nhau thông qua giao thức MQTT, chúng ta cần một broker MQTT để trung gian giữa chúng. Chúng ta có thể sử dụng một số dịch vụ MQTT công cộng như Mosquitto hoặc tự triển khai một broker MQTT trên máy tính của mình.

Mosquitto là một MQTT Broker mã nguồn mở cho phép thiết bị truyền nhận dữ liệu theo giao thức MQTT version 5.0, 3.1.1 và 3.1 – Một giao thức nhanh, nhẹ theo mô hình publish/subscribe được sử dụng rất nhiều trong lĩnh vực Internet of Things. Mosquitto cung cấp một thư viện viết bằng ngôn ngữ C để triển khai các MQTT Client và có thể dễ dàng sử dụng bằng dòng lệnh: “mosquitto\_pub” và “mosquitto\_sub”. Ngoài ra, Mosquitto cũng là một phần của Eclipse Foundation, là dự án iot.eclipse.org và được tài trợ bởi cedalo.com

Ưu điểm:

* Ưu điểm nổi bật của Mosquitto là tốc độ truyền nhận và xử lí dữ liệu nhanh, độ ổn định cao, được sử dụng rộng rãi và phù hợp với những ứng dụng embedded.
* Mosquitto rất nhẹ và phù hợp để sử dụng trên tất cả các thiết bị.
* Ngoài ra, Mosquitto cũng được hỗ trợ các giao thức TLS/SSL (các giao thức nhằm xác thực server và client, mã hóa các message để bảo mật dữ liệu).

Nhược điểm:

* Một số nhược điểm của mosquitto là khó thiết kế khi làm những ứng dụng lớn và ít phương thức xác thực thiết bị nên khả năng bảo mật vẫn chưa tối ưu.

=> Do cách thực hiện đơn giản, có sẵn nhóm em lựa chọn sử dụng mã nguồn mở mosquitto là broker trung gian cho giao thức MQTT.

### **2.2.2. CoAP**

1. *Định nghĩa*

CoAP hay giao thức ứng dụng bị ràng buộc là một giao thức truyền web chuyên biệt để sử dụng với các nút bị ràng buộc và mạng bị hạn chế trong IoT.

CoAP được thiết kế để cho phép các thiết bị đơn giản, bị hạn chế tham gia IoT ngay cảthông qua các mạng bị hạn chế với băng thông thấp và tính khả dụng thấp. Nó thường được sử dụng cho các ứng dụng giữa máy và máy (M2M) như năng lượng thông minh và tự động hóa tòa nhà. Giao thức được thiết kế bởi IETF, CoAP được quy định trong IETF RFC 7252.

CoAP là một giao thức truyền tải tài liệu theo mô hình client/server dự trên internet tương tự như giao thức HTTP nhưng được thiết kế cho các thiết bị ràngbuộc. Giao thức này hỗ trợ một giao thức one-to-one để chuyển đổi trạng thái thông tin giữa client và server.

1. *Nguyên lý hoạt động*

Ta có thể nhận thấy, một số tính năng rất giống với HTTP ngay cả khi CoAP không được coi là giao thức HTTP được nén vì CoAP được thiết kế riêng cho IoT và chi tiết hơn cho M2M nên nó được tối ưu hóa cho nhiệm vụ này.

Các gói CoAP nhỏ hơn nhiều so với dòng HTTP TCP. Các trường bit và ánh từ chuỗi các số nguyên được sử dụng rộng rãi để tiết kiệm không gian. Các gói rấtđơn giản có thể được tạo ra và phân tích tại chỗ mà không tốn thêm RAM trongcác thiết bị bị hạn chế.

CoAP sử dụng UDP (User Datagram Protocol), không hỗ trợ TCP, ngoài ra còn hỗ trợ địa chỉ broadcast và multicast, truyền thông CoAP thông qua các datagram phi kết nối (connectionless) có thể được sử dụng trên các giao thứctruyền thông dựa trên các gói.

UDP có thể dễ dàng triển khai trên các vi điều khiển hơn TCP nhưng các công cụ bảo mật như SSL/TSL không có sẵn, tuy nhiên ta có thể sử dụng Datagram Transport Layer Security (DTLS) để thay thế.

CoAP theo mô hình client/server. Client gửi yêu cầu đến máy chủ, sau đó máy chủ gửi lại phản hồi. Client có thể GET, PUT, POST và DELETE các tài nguyên.

A diagram of a client

Description automatically generated

Hình 2‑3. Tổng quan về nguyên lý hoạt động CoAP

CoAP có tính linh động và hỗ trợ đàm phán nội dung. Điều này cho phép client và máy chủ có thể nâng cấp, thêm mới một cách độc lập mà ko ảnh hưởng gì đến phía còn lại.

CoAP đưa ra các yêu cầu quan sát tài nguyên. Cả hai bên đều có thể tác động hoặc xóa các yêu cầu quan sát. Khi cờ quan sát được thiết lập, máy chủ vẫn cóthể tiếp tục hồi đáp sau khi các dữ liệu đã truyền đi.

Trong CoAP một nút cảm biến thường là một máy chủ. Chúng có khả năng nhận các gói tin gửi đến để hoạt động đúng đằng sau NAT, thiết bị đầu tiên phảigửi yêu cầu đến máy chủ, như được thực hiện trong LWM2M, cho phép các router liên kết chúng lại.

### **2.2.3. Nền tảng IoT**

Đáp ứng yêu cầu dự án đưa ra, Thingsboard là nền tảng được lựa chọn làm server và tạo giao diện trực quan hoá dữ liệu.

ThingsBoard là một nền tảng IoT mã nguồn mở để giám sát, xử lý dữ liệu, trực quan hóa dữ liệu cùng với quản lý thiết bị.

Nó hỗ trợ các giao thức IoT tiêu chuẩn công nghiệp - MQTT, CoAP và HTTP. ThingsBoard kết hợp khả năng mở rộng, khả năng chịu lỗi và hiệu suất để thu thập dữ liệu thiết bị để xử lý và giám sát. Nó cung cấp gateway server có thể giao tiếp với các thiết bị đính kèm.

A diagram of a software system

Description automatically generatedXây dựng dựa trên Netty Framework do nó cung cấp hỗ trợ cho nhiều giao thức và ứng dụng. Có thể thêm các giao thức phần cứng mới bằng cách thêm các trình xử lý kênh Inbound và Outbound cho các giao thức mới với netty framework.

Hình 2‑4 Nền tảng IoT mã nguồn mở Thingsboard

Các tính năng của Thingsboard:

* Thu thập dữ liệu từ xa
* Hiển thị trực quan dữ liệu đã thu thập
* Công cụ tạo chuỗi quy tắc kéo thả thân thiện
* Quản lý thiết bị
* Quản lý các báo động
* 100% mã nguồn mở,…

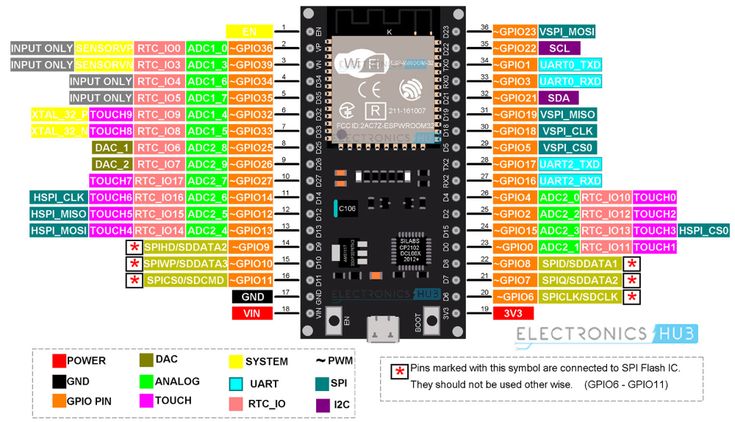
# **CHƯƠNG 3: THIẾT KẾ PHẦN CỨNG HỆ THỐNG**

## **3.1. Khối điều khiển trung tâm**

ESP32-WROOM-32 module là 1 module MCU mạnh mẽ, tích hợp WiFi + Bluetooth + BLE, nhắm đến nhiêu ứng dụng khác nhau, từ mạng cảm biến tiêu thụ ít năng lượng cho đến các nhiêunj vị yêu cầu cao nhất như mã hóa giọng nói, phát nhạc và giải mã MP3

Lõi của module này là chíp ESP32-D0WDQ6. Chip tích hợp này được thiết kế để có khả năng mở rộng và thích ứng. Có hai lõi CPU có thể được điều khiển riêng lẻ, tần số xung nhịp CPU có thể điều chỉnh từ 80MHz đến 240 MHz. ESP32 tích hợp nhiều các loại ngoại vi giúp giao tiếp hiệu quả với các thiết bị cảm biến và cơ cấu chấp hành.

Hệ thống truyền thông triển khai trên cơ sở mạng Wifi ở cả phần kết nối với thiết bị lẫn cloud, vì vậy, esp32 là lựa chọn phù hợp với ưu điểm về kết nối không dây với chi phis thấp. Đồng thời, esp32 tích hợp sẵn các chân giao tiếp ngoại vị giúp kết nối với các cảm biến và cơ cấu chấp hành dễ dàng.

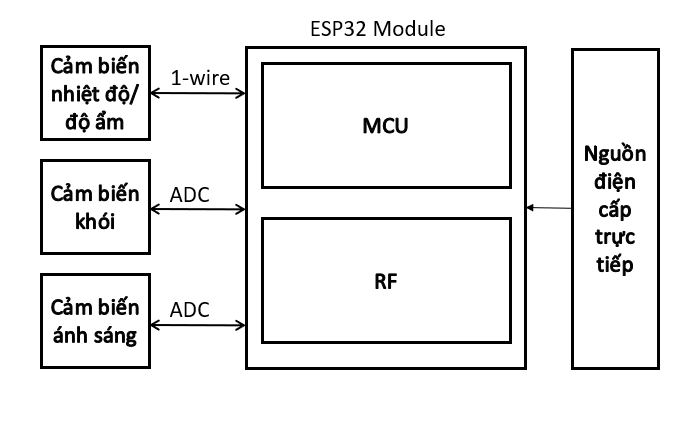


Hình 3‑1. ESP WROOM 32

Thông số kỹ thuật:

* Điện áp hoạt động: 2.2 – 3.6V
* Chip:2x ESP32-D0WDQ6
* Tần số hoạt động: 80 MHz – 240 MHz
* RAM: 520 KB SRAM
* Flash: 4MB
* Wifi: 802.11 b/g/n
* Ngoại vị: GPIO, ADC, DAC, SPI, I2C, ..

## **3.2. Nút cảm biến**



Hình 3‑2 Sơ đồ khối chức năng phần cứng cảm biến

## **3.3. Cơ cấu chấp hành HVAC**

### **3.3.1. Sơ đồ khối chức năng**

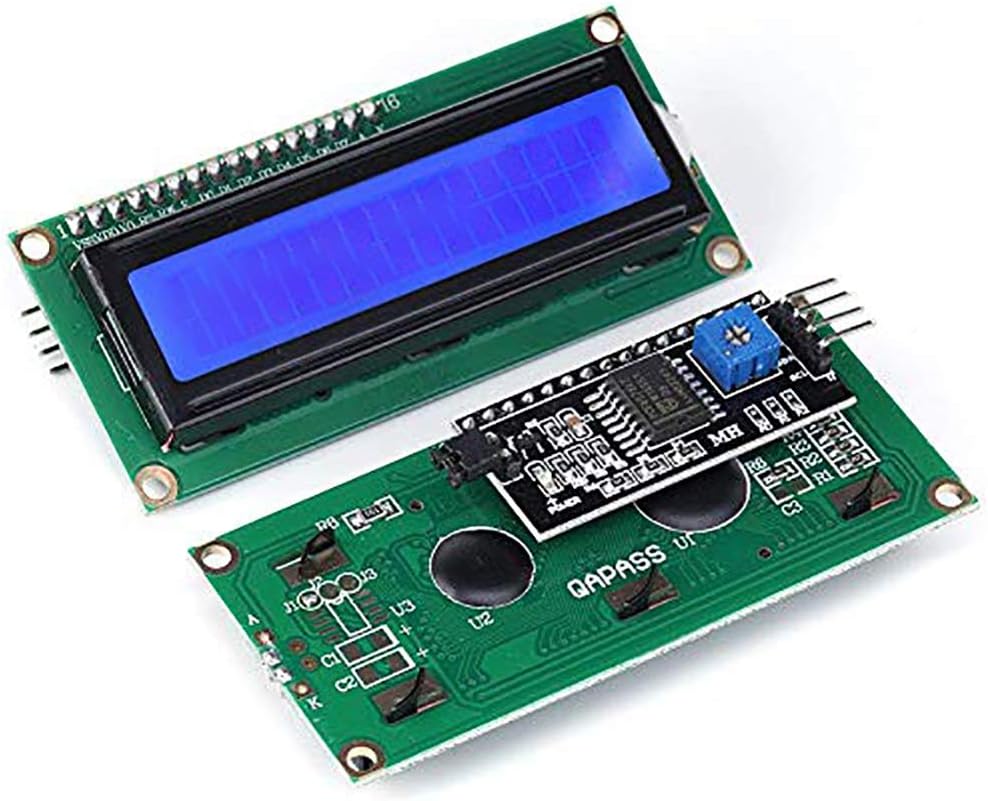
A diagram of a computer system

Description automatically generated

Hình 3‑3 Sơ đồ khối chức năng phầ cứng cơ cấu chấp hành

### **3.3.2. Lựa chọn phần cứng**

Nhóm em chọn màn hình LCD để mô phỏng điều khiển các thông số HVAC.



Hình 3‑4 Màn hình LCD 1602

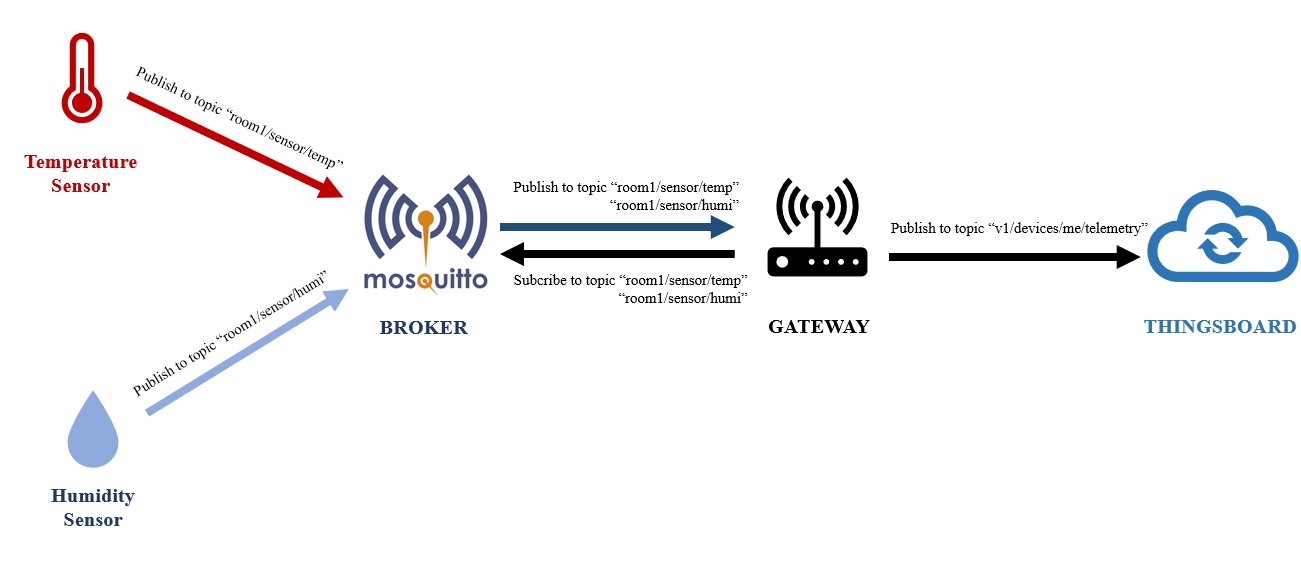
Thông số kỹ thuật:

* Điện áp hoạt động: 2.5-6V DC.
* Hỗ trợ màn hình: LCD1602,1604,2004 (driver HD44780).
* Giao tiếp: I2C.
* Địa chỉ mặc định: 0X27 (có thể điều chỉnh bằng ngắn mạch chân A0/A1/A2).
* Tích hợp Jump chốt để cung cấp đèn cho LCD hoặc ngắt.
* Tích hợp biến trở xoay điều chỉnh độ tương phản cho LCD.

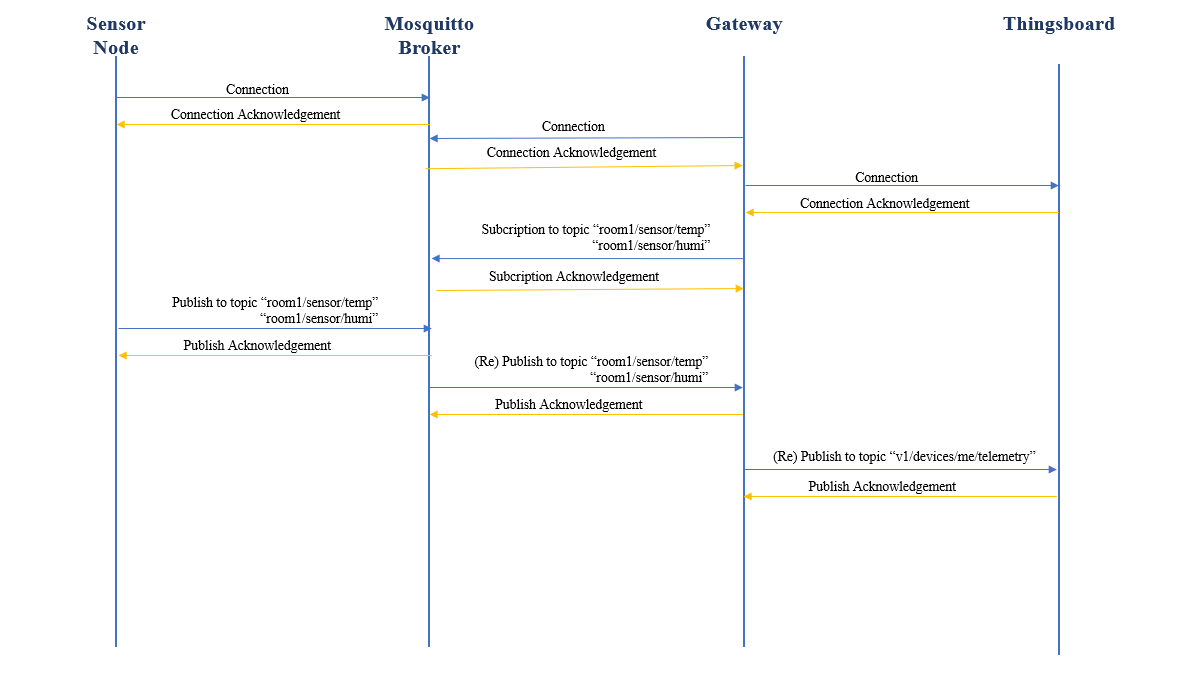
# **CHƯƠNG 4: THIẾT KẾ PHẦN MỀM HỆ THỐNG**

## **4.1. Thiết kế thiết bị kết nối MQTT**

### **4.1.1. Thiết kế cảm biến – DHT11**



Hình 4‑1 Biểu đồ dữ liệu theo topic của cảm biến



Hình 4‑2 Biểu đồ tuần tự giao tiếp với cảm biến

### **4.1.2. Thiết kế cơ cấu chấp hành – HVAC**

A diagram of a router

Description automatically generated

Hình 4‑3 Tổng quan hoạt động kết nối các topic MQTT cho HVAC

A diagram of a computer network

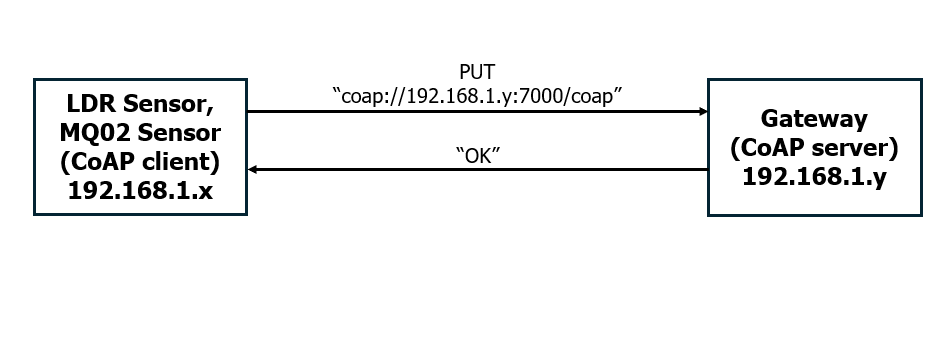
Description automatically generated

Hình 4‑4 Biểu đồ tuần tự quá trình nhận dữ liệu của HVAC

## **4.2. Thiết kế thiết bị kết nối CoAP**

### **4.2.1. Cảm biến ánh sáng và chất lượng không khí**

Cảm biến là bên chủ động trong việc đo dữ liệu và gửi yêu cầu, khởi tạo kết nối, gửi dữ liệu cho gateway nên nó sẽ đóng vai trò là client, còn gateway lắng nghe các yêu cầu, nhận dữ liệu, quản lý tài nguyên và phản hồi lại chúng nên đóng vai trò là server. Cảm biến ánh sáng, chất lượng không khí chủ động đọc dữ liệu và gửi đến gateway, do đó nó là client.



Hình 4‑5 Biểu đồ dữ liệu giữa cảm biến và gateway

### **4.2.2. Đèn chiếu sáng**

Độ sáng của đèn được điều khiển dựa trên xung PWM, là phương pháp điều khiển bằng cách tạo rao các xung điện áp có chu kỳ nhất định và thay đổi độ rộng xung này để điều chỉnh điện áp trung bình cấp cho thiết bị.

Chu kỳ làm việc là tỷ lệ phần trăm thời gian mà xung điện áp ở mức cao so với toàn bộ chu kỳ xung. Chu kỳ làm việc có thể coi là có quan hệ tuyến tính với điện áp trung bình của đèn cũng.

Chính vì vậy, độ sáng của đèn được điều khiển bằng cách thay đổi tỷ lê giữa chu kỳ làm việc của xung pwm với chù kỳ của xung. Điều này cho phép độ sáng được điều chỉnh chính xác từ 0% đến 100%.

Do việc điều khiển ánh sáng ít thường xuyên cùng với gói tin cần trao đổi trong truyền thôn khá đơn giản khi chỉ cần 1 số có giá trị từ 0 đến 100. Vì vậy, giao thức CoAP được sử dụng để truyền nhân dữ liệu giữa gateway và thiết bị đèn chiếu sáng.

Giao thức CoAP sử dụng mô hình client - server, một số phương thức giao tiếp giống với HTTP được sử dụng như: GET, POST, PUT, DELETE. Client và Server trong CoAP giao tiếp với nhau dựa trên các phương thức trên và đặc biệt quá trình giao tiếp luôn được bắt đầu khi có request từ phía client nên server không thể chủ động gửi bản tin đến client trước.

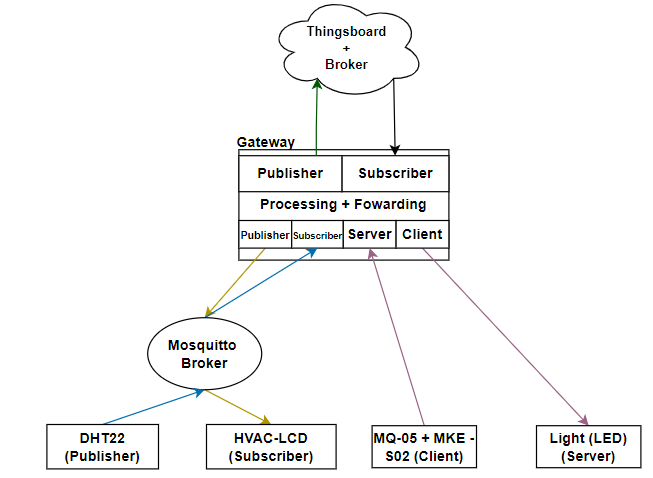
A black text and a rectangular object

Description automatically generated with medium confidence

Hình 4-6 Biểu đồ dữ liệu giữa cảm biến và gateway

Việc điều khiển ánh sáng xảy ra chỉ khi có tín hiệu điều khiển từ gateway và tín hiệu điều khiển này có tính chất hướng sự kiện mà không luân phiên theo chu kỳ. Vì vậy, để truyền nhận dữ liệu, giao diện sử dụng để giao tiếp của thiết bị đèn đóng vai trò là server lắng nghe yêu cầu của gateway, còn phía client hay gateway sẽ gửi tín hiệu điều khiển đến server bằng phương thức PUT. Để đảm bảo độ tin cậy cho truyền nhận dữ liệu, gói tin phàn hồi sẽ được gửi lại gateway sau khi thiết bị đèn đã nhận được tín hiệu điều khiển.

## **4.3. Thiết kế gateway**



Hình 4-7 Sơ đồ kết nối gateway

Yêu cầu thiết kế cho gateway là cần giao tiếp với giao diện Thingsboard qua giao thức mqtt, đồng thời, các thiết bị cảm biến và chấp hành sẽ giao tiếp với gateway qua giao thức mqtt và CoAP.

Với yêu cầu thiết kế và thiết kế đã được triển khai ở phía thiết bị, gateway được xây dựng với các vai trò về mặt logic như trên hình.

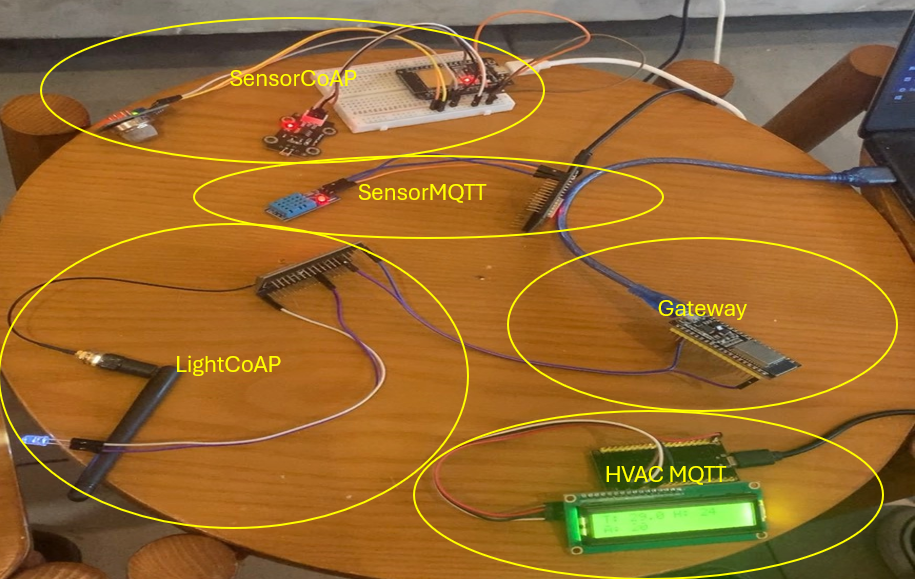
Trong giao thức MQTT, khi truyền dữ liệu cảm biến lên giao diện của Thingsboard, gateway đóng vai trò là subscriber để kết nối đến broker ở mạng local, từ đó nhận dữ liệu mà cảm biến publish đến. Sau khi đã nhận dữ liệu cảm biến, gateway lại đóng vai trò là publisher để truyền dữ liệu lên cho Thingsboard. Quá trình nhận dữ liệu diễn ra tương tự với quy trình ngược lại.

Với dữ liệu từ cảm biến ánh sáng, gateway xây dưng CoAP server để cảm biến (client) truyền dữ liệu đến và khi cần truyền tín hiệu điều khiển đèn, CoAP client được xây dựng đẻ thưc hiện đẩy tín hiệu điều khiển đến đèn.

Ngoài quá trình trình chuyển đổi và luân chuyển dữ liệu, gateway có thể thực hiện các chức năng xử lý dữ liệu như điều khiển đèn, cảnh báo ngưỡng.

# **Kết quả đạt được**

**A screenshot of a computer

Description automatically generated**

Hình 5‑2 Giao diện Thingsboard

Hình 5‑1 Hình ảnh thực tế kết nối các khối hệ thống

# **Hướng phát triển đề tài**

* Thêm phần cảnh báo dựa vào các thông số môi trường đo được để cảnh báo trong các trường hợp bất thường.
* Tự động điều khiển các thông số dựa vào ngưỡng do người dùng cài đặt.